



Delinea

Secret Server

Documentation © 11.2.0



Table of Contents

Secret Server Documentation	75
Introduction	75
Documentation	75
Primary Documentation	75
Getting Started	76
Best Practices	76
Security Whitepapers	76
Help	76
Download Secret Server	76
Release Notes	76
Secret Society (Forum)	76
Video Tutorials	76
Developer Resources	77
Getting Started Tutorial	78
Step 1: Trial Installation Prerequisites	79
<i>System Requirements</i>	79
<i>Hardware Requirements</i>	79
<i>Software Requirements</i>	79
Checklist	79
SQL Server	79
Application Server	79
<i>Application Configuration</i>	79
Service Account	79
Active Directory Group Sync	80
Discovery	80
Test Accounts	80
Email Notifications	80
SSL Certificate	80
Firewalls and Ports	80
Step 2: Installation	81
<i>Process</i>	81
<i>Licenses</i>	81
Step 3: Secret Server Dashboard	82
Step 4: Security Best Practices	83
<i>Local Admin Account Best Practices</i>	83
<i>SSL (HTTPS) Best Practice</i>	83

Step 5: Backups	84
Step 6: Active Directory Integration	85
<i>Setting up Active Directory</i>	85
<i>Enabling Active Directory Users</i>	85
<i>Managing Active Directory Users via a Distributed Engine</i>	85
Step 7: Secret Server Framework	86
Step 8: Discovery	87
Step 9: Remote Password Changing	88
<i>Enabling Remote Password Changing</i>	88
<i>Performing a Manual RPC</i>	88
<i>Common RPC Error Codes</i>	88
Step 10: Heartbeats	89
<i>Enabling Heartbeat</i>	89
<i>Running Heartbeat</i>	89
Step 11: Audits and Reports	90
Step 12: Secret Access and Workflow	91
Step 13: Secret Launchers	92
Step 14: Recording Sessions	93
Step 15: Secret Server APIs and CLI	94
Step 16: Additional Resources for Secret Server	96
Help	97
Document Conventions	98
<i>Capitalization</i>	98
<i>Code and Command Line Text</i>	98
<i>Keyboard Shortcuts</i>	98
<i>Notes</i>	98
<i>Other Special Text</i>	98
<i>Screen Components and Attentional Targets</i>	99
Secret Server Glossary	100
Self-Help Resources	105
<i>Forums</i>	105
<i>Delinea Blog</i>	105
Technical Support	106
<i>Technical Support Coverage</i>	106
Accessing Upgrades	106
Requesting New Features	106
<i>Getting Technical Support</i>	106
Step One: Gather Information You May Need	106
Step Two: Get a Mandatory Support PIN	106
<i>Secret Server</i>	106

<i>Secret Server Cloud</i>	107
Step Three: Choose a Support Method	107
Step Four: Contact Support	107
<i>Phone Support</i>	107
<i>Email Support</i>	107
<i>Ticketing System Support</i>	107
Access Requests	108
Approving a Request	109
Customizing Access Request Emails	110
Duo Push Notifications	111
<i>Prerequisites</i>	111
<i>Assigning the Duo Approval Permission</i>	111
Requesting Access After Approval Is Granted	112
Setting up Access Requests for Secrets	113
Secret Server Administration	114
Administration Side Panel and Page	115
<i>Administration Side Panel</i>	116
Navigation	116
Pinning Commonly Used Items	117
<i>Administration Page</i>	117
Application Dashboard	121
<i>Dashboard Components</i>	122
Overview Tab	122
Customized Tabs	122
<i>Tab Management</i>	122
<i>Dashboard Widgets</i>	122
<i>Widget Types</i>	122
<i>Managing Widgets</i>	123
<i>Dashboard Tools and Help Menu</i>	124
Tool Section	124
Help Section	124
<i>Running Dashboard Bulk Operations</i>	125
<i>User Interfaces, Themes, and Color Modes</i>	127
Overview	127
<i>Terms</i>	127
<i>Best Practices</i>	127
<i>For Users</i>	127
<i>For Admins</i>	127
Procedures for Users	128
<i>Switching to the New UI from the Classic UI</i>	128

<i>Switching to the Classic UI from the New UI</i>	128
<i>Setting Your Default Classic UI Theme</i>	128
<i>Setting Your Default Color Mode</i>	129
Procedures for Admins	130
<i>Choosing the Default Classic UI and Theme for New Users</i>	130
<i>Choosing the Default New UI Color Mode for New Users</i>	131
Configuration Overview Page	132
Enabling a Company Policy Login Banner	134
Enabling and Disabling Maintenance Mode	136
Encryption and Security	138
<i>Advanced Encryption Standard</i>	139
<i>Restricting IP Addresses</i>	140
Creating IP Address Ranges	140
Editing and Deleting IP Address Ranges	141
Assigning an IP Address Range	142
<i>Secret Key Rotation</i>	144
Overview	144
How to Perform Secret Key Rotation	144
Estimated Processing Time	144
<i>Security Compliance Standards</i>	145
FIPS Compliance	145
PCI Datacenter Compliance	145
<i>SSH Key Rotation</i>	146
Basic SSH Key Rotation	147
<i>Introduction</i>	147
<i>Requirements</i>	147
<i>Configuring a Secret for SSH Key Rotation</i>	147
<i>SSH Key Rotation Using the Secret's Credentials</i>	147
<i>Creating the Secret</i>	147
<i>Rotating the Key</i>	148
<i>SSH Key Rotation Using a Privileged Account</i>	148
<i>Creating the Secret</i>	148
<i>Rotating the Key</i>	149
<i>Troubleshooting</i>	149
Custom SSH Key Rotation	150
<i>Introduction</i>	150
<i>Requirements</i>	150
<i>Secret Templates</i>	150
<i>Creating a New SSH Key Rotation Secret</i>	150
<i>Editing the SSH Key Rotation Templates</i>	152

<i>Password Changers</i>	152
<i>Authentication</i>	153
<i>Command Sets</i>	153
<i>Overview</i>	153
<i>Password Change Command Set</i>	154
<i>Verify Password Changed (Heartbeat) Command Set</i>	154
<i>Post Successful Change Command Set</i>	154
<i>Post Fail Change Command Set</i>	154
<i>Notes</i>	154
<i>Troubleshooting</i>	155
<i>SSL Certificates</i>	156
Exporting and Importing Secret Server Settings	157
<i>Overview</i>	157
<i>Prerequisites</i>	157
Required General Permissions	157
Required Additional Permissions	157
Required Licenses	157
<i>Advanced Auditing License</i>	157
<i>Enterprise Edition</i>	158
<i>Pro Edition</i>	158
<i>Platinum Edition</i>	158
<i>Procedures</i>	158
Exporting Settings	158
Importing Settings	162
<i>Setting Category Reference</i>	168
Application Settings	168
Advanced Settings	169
Launcher Settings (Runtime)	169
Email	169
Folder Settings	169
Licenses	169
Local User Passwords	169
Login	169
Permission Options	170
Protocol Handler Settings (Install-Time)	170
SAML	170
Security	170
Session Recording	170
SSH Commands	171
Ticket System	171

User Experience	171
User Interface	171
<i>JSON Export File</i>	171
External Instance ID	171
Configuration Version	171
<i>JSON Import File</i>	171
<i>API Calls Filter</i>	172
<i>Audits</i>	177
<i>Events</i>	178
<i>Logs</i>	178
System Logs or CEF Example	178
SS.log Examples	178
<i>Errors and Resolutions</i>	178
Maintenance Mode FAQ	180
<i>What is Maintenance Mode?</i>	180
<i>Why do we need Maintenance Mode?</i>	180
<i>Can I still access my Secrets when Maintenance Mode is turned on?</i>	180
<i>How long does Maintenance Mode last?</i>	180
<i>How do you enable and disable Maintenance Mode?</i>	180
Secret Server Object Metadata	181
<i>Overview</i>	181
<i>Features</i>	181
<i>Example Use Cases</i>	181
<i>Adding Object Metadata</i>	181
<i>Best Practices</i>	186
Syncing with DevOps Secret Vault	187
<i>Overview</i>	187
<i>Behavior Test</i>	187
<i>Fields</i>	187
<i>Setup in Secret Server</i>	187
<i>API Examples</i>	189
Creating a DevOps Secret Vault Tenant	189
Creating a Sync Map	190
Manually Syncing a Secret	190
Listing DevOps Secret Vault Tenants	190
Getting a DevOps Secret Vault Tenant's Details	191
Getting the Status of a Secret's Synchronization	191
Getting a List of Secret Synchronization Statuses	191
APIs and Scripting	192
Configuring WinRM for PowerShell	193

<i>Overview</i>	193
<i>Configuration (Domain Joined)</i>	193
<i>Verifying Listeners</i>	193
<i>Additional Considerations</i>	193
<i>Configuration (Standalone)</i>	193
Creating and Using PowerShell Scripts	195
<i>Overview</i>	195
<i>Creating a PowerShell Script</i>	195
<i>Best Practices</i>	197
Debugging	197
Script Arguments	197
Exceptions	198
Creating and Using SQL Scripts	199
<i>Creating a SQL Script</i>	199
<i>Using Parameters</i>	199
Examples	199
SQL	199
MySQL	199
PostgreSQL	199
ODBC	199
Basic Oracle PL/SQL	199
Advanced Oracle PL/SQL	199
<i>Returning Errors</i>	200
Examples	200
SQL	200
MySQL	200
PostgreSQL	200
ODBC	200
<i>SQL Example</i>	200
Example	200
SQL	200
Parameters	200
Creating and Using SSH Scripts	202
<i>Creating an SSH Script</i>	202
<i>Use Case</i>	202
<i>Adding an SSH Script as a Dependency</i>	203
Dependency Token List	205
<i>Overview</i>	205
<i>Best Practices</i>	205
Wrap Fields in Quotation Marks	205

Avoid Using Spaces in Field Names	205
<i>Token List</i>	205
Downloads for Secret Server Software Development Kit for DevOps	208
<i>Overview</i>	208
<i>SDK Client version 1.5.3</i>	208
Release Notes	208
Downloads	208
Download File Hashes	208
<i>Windows x64</i>	208
<i>MacOS x64</i>	208
<i>Linux x64</i>	208
<i>Red Hat Enterprise Linux 6 x64</i>	208
<i>Legacy Releases</i>	208
SDK Client version 1.5.0	208
Release Notes	209
Downloads	209
Download File Hashes	209
<i>Windows x64</i>	209
<i>MacOS x64</i>	209
<i>Linux x64</i>	209
<i>Red Hat Enterprise Linux 6 x64</i>	209
SDK Client version 1.4.1	209
Release Notes	209
Downloads	209
SDK Client version 1.3.0	210
Release Notes	210
Downloads	210
SDK Client version 1.0.0	210
Release Notes	210
Downloads	210
<i>SDK NuGet Packages (Optional)</i>	210
REST API Client Generation with OpenAPI Swagger	211
<i>Generating Clients</i>	211
C# Client Using NSwagStudio	211
C# or .NET Core Client Using OpenAPI Generator	211
Java Client Using OpenAPI Generator	212
PowerShell Using OpenAPI Generator	213
Postman Workspace	214
Insomnia Workspace	214
<i>Getting OpenAPI Generator</i>	214

<i>Self-Signed or Other Invalid Certificates</i>	214
<i>Swagger 2.0 Notes</i>	214
REST API Perl Scripts	216
REST API PowerShell Script Examples	225
<i>Authentication</i>	225
<i>Searching Secrets</i>	225
<i>Creating a Secret</i>	226
<i>Editing a Secret</i>	227
<i>Checking in a Secret</i>	228
<i>Deleting a Secret</i>	228
<i>Creating a User</i>	229
<i>Field Update, Get, File Upload, and Download</i>	230
<i>Expiring a Token</i>	231
<i>Managing Folders (Basic)</i>	231
REST API Python Scripts	235
REST Web Services API Reference and Download	237
<i>Overview</i>	237
<i>Accessing the Guides</i>	237
<i>Downloading the Guides</i>	237
Current Version	237
Version Archive	237
Script Authentication Using Tokens	239
<i>Overview</i>	239
<i>Best Practices</i>	239
<i>Authentication Methods</i>	239
Scripts Run by Automated Processes	239
Specific User Use Cases with Script Augmentation	239
<i>Example Credential-Access Scripts</i>	239
Recommended Methods	239
<i>Get-Credential with 2FA</i>	239
<i>Prompting for Credentials with 2FA</i>	240
<i>Integrated Windows Authentication</i>	241
Deprecated Method	241
<i>API token from the UI with 2FA</i>	241
<i>Downloading Example Scripts</i>	242
Secret Server Software Development Kit for DevOps	243
<i>Overview</i>	243
<i>How It Works</i>	243
<i>Configuration Overview</i>	244
<i>Required Roles and Permissions</i>	244

<i>Setup Procedure</i>	244
Task 1: Configuring Secret Server	244
Task 2: Installing the SDK Client	250
Task 3: Connecting to Secret Server	250
<i>Usage Examples</i>	250
<i>SDK Client Management</i>	251
<i>SDK Client Caching</i>	251
Overview	251
Examples	251
SSH Command Restrictions	253
<i>SSH Blocked Command Lists</i>	254
Overview	254
Requirements	254
<i>Creating SSH Blocked Command Lists</i>	254
<i>Applying SSH Command Blocked Lists in Secret Settings</i>	257
<i>SSH Command Restrictions via a Secret Policy</i>	258
<i>SSH Command Menus</i>	259
Using Secret Fields in Scripts	260
Using Webservices with IWA via PowerShell	261
<i>Overview</i>	261
<i>Procedure</i>	261
<i>Access Examples</i>	261
SOAP	261
REST	262
Using Webservices with IWA via Perl	263
<i>Overview</i>	263
<i>Procedure</i>	263
<i>Example</i>	263
Overview	263
SecretServerGetSecret.ps1	264
SecretServer.pm	264
Sample.pl	264
Secret Server Authentication, Encryption, and Security	266
Configuring CredSSP for WinRM with PowerShell	267
<i>Introduction</i>	267
<i>Enabling CredSSP for WinRM in Secret Server</i>	267
<i>Configuring CredSSP for WinRM on the Secret Server Machine</i>	268
<i>Configuring CredSSP for WinRM on a Distributed Engine</i>	268
<i>Enabling CredSSP on Secret Server Agents for PowerShell Script Dependencies</i>	271
Configuring SAML OneLogin	273

<i>Step One: OneLogin</i>	273
<i>Step Two: Secret Server</i>	275
Configuring SAML Single Sign-on	278
<i>SAML Overview</i>	278
<i>Prerequisites</i>	278
Licensing and Version	278
.NET Framework 4.6.2+	279
Administer Configuration SAML Role Permission	279
<i>Setting up Secret Server</i>	281
<i>Setting up IDPs</i>	283
<i>Lockout Workaround</i>	284
Delinea One and Secret Server	285
<i>Overview</i>	285
<i>Cloud versus On-Premise</i>	285
<i>Procedures</i>	285
Logging in with Delinea One	285
Configuring Delinea One Delinea	286
<i>Secret Server Cloud</i>	286
<i>Secret Server On-Premise</i>	287
Generating a Delinea One Credential	288
Enabling FIPS Compliance	291
<i>Overview</i>	291
<i>Procedure</i>	291
Task 1: Enable FIPS in Secret Server	291
Task 2: Enable FIPS in Windows	291
Task 3: Reset the IIS Server	292
<i>Related Information</i>	292
Enabling Refresh Tokens for Web Services	293
<i>Overview</i>	293
<i>How to Enable Refresh Tokens in Secret Server</i>	293
Procedure	293
Example	297
Installing a Self-Signed SSL Certificate	298
<i>Overview</i>	298
<i>Obtaining an SSL Certificate</i>	298
<i>Installing a Self-Signed Certificate</i>	298
Task One: Generate an IIS Self-Signed Certificate	298
Task Two: Bind the Self-Signed Certificate to the IIS Site	298
Task Three: Test the Self-Signed Certificate	299
OpenID Connect Integration	300

<i>Introduction</i>	300
OpenID Connect	300
OpenID Connect Support in Secret Server	300
<i>Prerequisites</i>	300
General	300
Permissions	300
Configuration	300
<i>Task One: Acquire and Configure an OpenID Connect Provider</i>	300
<i>Task Two: Configure Secret Server</i>	301
<i>Task Three: Matching External Accounts to Secret Server Users</i>	301
<i>Task Four: Logging on with OpenID Connect</i>	301
SAML	303
Server SSH Key Verification	304
<i>How to Map a Server SHA1 Digest to a Secret</i>	304
<i>Heartbeat</i>	304
<i>Password Changing</i>	304
<i>Non-Proxied Launcher</i>	304
<i>Proxied Launcher</i>	304
<i>SSH Script Dependencies</i>	304
<i>Unix Account Discovery</i>	304
X.509 Certificate Security Chain Options	306
<i>Setting the Certificate Verification Policy</i>	306
<i>Certificate Validation Options</i>	307
X509RevocationMode	307
X509RevocationFlag	308
X509VerificationFlags	308
<i>Troubleshooting</i>	309
Integrated Windows Authentication	310
<i>Configuring Integrated Windows Authentication</i>	311
Introduction	311
Setting Up Windows Authentication	311
<i>Task 1: Configuring Secret Server</i>	311
<i>Task 2: Configuring IIS</i>	314
<i>Task 3: Configuring Secret Server Launchers</i>	316
<i>Task 5: Configuring Client Certificates</i>	323
Troubleshooting	325
<i>AD User Prompted for Credentials Even Though IWA Is Active</i>	325
<i>Logging in as a Local Account Is Not Available</i>	325
<i>Installing Windows Authentication in Windows Server 2012 Manager</i>	326
Secret-based Credentials for PowerShell Scripts	327

<i>Overview</i>	327
<i>RunAs Secret Precedence</i>	327
Remote Password Changing	327
Secret Dependencies	327
Checkout Hooks	327
<i>Procedures</i>	327
Setting the Default PowerShell Credential for a Site	327
Using the Site PowerShell Credentials for Discovery	328
Trusting an SSL Certificate on a Client Machine	329
<i>Step 1: Compare Host Names</i>	329
<i>Step 2: Transfer a copy from your server to the client computer</i>	329
<i>Step 3: Install the certificate on the client computer</i>	329
Two-Factor Authentication	331
<i>Duo Security Authentication</i>	332
Task 1: Create a Duo Application Representing Your Secret Server (Admin)	332
Task 2: Configure Secret Server to Use Duo (Admin)	332
Task 3: Setting up Duo (User)	333
<i>Applications for Soft Token Two-Factor Authentication</i>	334
<i>Email Two-Factor Authentication</i>	335
<i>FIDO2 (YubiKey) Two-Factor Authentication Configuration</i>	336
Overview	336
<i>FIDO2</i>	336
<i>YubiKey</i>	336
Configuration	336
<i>Prerequisites</i>	336
<i>Enabling FIDO2 for a Single User</i>	336
<i>Enabling FIDO2 for Multiple Users</i>	338
<i>Disabling FIDO2 for Users</i>	339
<i>Unregistering Users from FIDO2</i>	340
<i>Registering FIDO2 Devices (End User Operation)</i>	340
<i>Auditing and Security</i>	340
<i>Troubleshooting and Issues</i>	341
<i>RADIUS User Authentication</i>	342
Configuring RADIUS	342
Enabling RADIUS for a User	342
Enabling RADIUS Two-Factor Authentication	343
<i>TOTP</i>	344
Disabling TOTP for Users	345
Enabling TOTP for Secret Server Users	346
Enabling TOTP for Launchers	347

<i>Secret Template Setup</i>	347
<i>TOTP Secret Setup</i>	347
Resetting TOTP for Secret Server Users	350
Viewing a TOTP for a Web Secret	351
<i>Enabling Two-Factor Authentication in Delinea One</i>	353
TOTP Two-Factor Authentication	353
SMS Two-Factor Authentication	354
Backup and Disaster Recovery	356
Backing up Secret Server to a Network Share	357
Backup Folder Permissions	359
Backup Settings	360
<i>Overview</i>	360
<i>File Path Settings</i>	360
Common Backup Errors	361
Disaster Recovery	362
<i>Overview</i>	362
<i>Setup</i>	362
Role	362
Data Source DR configuration	362
Replica DR configuration	362
Finishing up	362
<i>Replication</i>	363
Replication Scope	363
Replica Mode	363
<i>Recommendations</i>	363
File Attachment Backups	364
Secret Server Geo Replication	365
<i>Overview</i>	365
<i>SQL Server Replication</i>	365
Benefits of Replication	365
<i>High Availability</i>	365
<i>Architecture</i>	366
<i>Data Synchronization</i>	366
<i>Data Conflicts</i>	366
<i>SQL Server Replication Monitor and Conflict Viewer</i>	366
<i>Tracking level</i>	366
<i>Conflict Resolvers</i>	366
Secret Server and SQL Replication	367
Using a Subscriber When the Publisher Is Offline	367
Secret Server Distributed Engine	368

Secret Server Replication Settings	369
Publications	369
Tracking Level and Resolvers	369
Conflict Auditing	370
Operational Latency	370
Article Settings	370
Compensate for Errors	371
Identity Range	371
Variations	371
<i>Installing and Configuring SQL Server Replication</i>	371
Installation	371
Troubleshooting the Installation	372
<i>Replication Setup Scripts Fail</i>	372
<i>SQL Replication Job Fails</i>	372
<i>Removing SQL Server Replication</i>	372
On Each Subscriber	372
On the Publisher	373
<i>Managing SQL Server Replication</i>	373
<i>Web Server Nodes</i>	374
<i>Secret Server Upgrade Scenario</i>	374
<i>Other Information about SQL Server Replication</i>	375
Manually Backing up Secret Server	376
Restoring Secret Server from a Backup	377
<i>Restoring the Application</i>	377
<i>Restoring the SQL Server Database</i>	377
Scenario One: Database and Secret Server Are in the Same Location	377
Scenario Two: The Database and Secret Server Are in Different Locations	378
Scheduled Backups	380
Server Clustering	381
SQL Server Mirroring	382
<i>Introduction</i>	382
<i>Procedures</i>	382
Setting up Databases for Mirroring	382
SQL Server Configuration	382
Configuring Mirroring	382
Configuring Secret Server for Mirroring	383
Testing Mirroring	385
Database SSL Configuration	385
Unlimited Administration Mode	387
Best Practices	388

Getting Started	388
<i>Overview</i>	388
<i>Terminology</i>	388
Administrator	388
Basic User Role	388
Folder	388
Role Based Access Control (RBAC)	388
Secret	388
Site	389
User	389
<i>Know Your Edition</i>	389
Installation and Configuration	389
<i>Installation</i>	389
<i>Basic Configuration</i>	389
<i>Advanced Configuration</i>	389
Architectural and Design Considerations	391
<i>Session Recording</i>	391
<i>Discovery</i>	391
<i>API Use Case</i>	392
<i>Remote Password Changes and Heartbeats</i>	392
<i>Proxying</i>	392
<i>General On-Premise Considerations</i>	392
Securing the encryption.config File	393
<i>Secret Server On-Premises</i>	393
<i>Secret Server Cloud</i>	393
Privileged Account Management Strategy	393
<i>Identify Data at Risk</i>	394
<i>Who Accesses Secret Server?</i>	394
<i>What Privilege Levels Are Necessary?</i>	394
<i>What are your Password Requirements?</i>	394
<i>Evaluate your Existing Setup</i>	394
<i>Define Your Core PAM Strategy</i>	395
<i>Individual Privileged Domain Accounts</i>	395
<i>Shared Privileged Domain Accounts</i>	395
<i>Hybrid of Individual and Shared Accounts</i>	395
<i>What Is the Highest Risk?</i>	396
Users and Groups	396
<i>Local Secret Server Accounts</i>	396
<i>Active Directory Accounts</i>	396
<i>Local or Active Directory Accounts?</i>	396

Only Local Users and Groups	397
Only AD Users and Groups	397
Hybrid of AD Users and Local Groups	397
<i>Business Users</i>	397
Authentication Strategy	398
<i>Strong Authentication</i>	398
SAML	398
<i>Directory Services</i>	398
Roles	399
<i>Role Definition and Assignment</i>	399
<i>Group Assignment</i>	399
Permissions	399
Folder Structure	400
<i>Using Folders to Control Access (Inherit Permission)</i>	400
<i>Deciding on your Folder Structure</i>	400
Secret Policy	401
Discovery	401
<i>Discovery Workflow</i>	401
<i>Enterprise Deployment Considerations</i>	402
Cloud Accounts	402
Local Windows Accounts	402
Find Backdoor Accounts	402
Service Accounts	402
Unix Accounts	402
ESX/ESXi accounts	402
Workflow Security	402
<i>Hide Launcher Password</i>	403
<i>Require Approval</i>	403
<i>Require Comments</i>	403
<i>Check Out</i>	404
<i>Session Monitoring</i>	404
Secret Templates	404
<i>Configuring Templates</i>	405
<i>File Attachments</i>	405
<i>Naming Patterns</i>	405
<i>Password History</i>	405
<i>Password Requirements</i>	405
<i>Secret Expiration</i>	406
<i>Session Launcher</i>	406
Template Management	406

<i>Basic Configuration</i>	406
<i>Deactivate Unused or Retired Templates</i>	406
<i>Limit Secret Template Administrators</i>	406
<i>Override Settings at the Secret</i>	406
Alerting and Reporting	407
Data Retention and Database Size Management	407
API and Extensibility	407
<i>Running PowerShell with Secret Server</i>	407
PowerShell Runspaces	408
CredSSP	408
API	408
API Authentication	408
<i>Software Development Kit</i>	408
<i>Integrated Windows Authentication</i>	408
<i>Event Pipelines</i>	409
Developer Resources	410
Custom Reports	410
General Scripting	410
REST API	410
Scripting Dependencies	410
Scripting Tools and CLI	410
SOAP API	410
Directory Services	411
Active Directory	412
<i>Active Directory Rights for Synchronization Account</i>	413
Recommended Permissions	413
<i>Object Tab</i>	413
Minimum Required Permissions	413
<i>Object Tab</i>	413
<i>Properties Tab</i>	413
<i>Active Directory Credential Caching</i>	414
Overview	414
AD Caching Configuration	414
Auditing	414
<i>ADFS Custom Rules</i>	415
Overview	415
Change the SAML Username Attribute	415
Create Three Rules	416
<i>Rule 1: Query AD for UPN and sAMaccountname Attributes</i>	416
<i>Rule 2: Obtain the Domain from the UPN</i>	416

<i>Rule 3: Combine the sAMAccountname with the Domain</i>	416
<i>Configuration Parameters</i>	418
<i>Configuring Active Directory</i>	419
Step 1: Enabling Active Directory Integration	419
Step 2: Adding a Domain	419
Step 3: Setting Up Synchronization Groups	419
Step 4: Adding or Removing Groups	419
Step 5: Enabling Active Directory Synchronization	420
Step 6: Running Active Directory Synchronization	420
<i>Converting Local Users to Domain Users</i>	421
<i>Creating Active Directory Users</i>	422
<i>Enabling and Disabling Active Directory Users</i>	423
<i>Setting up SAML SSO for Active Directory</i>	424
ADFS Server	424
Secret Server	424
Adding Users to ADFS	425
Common Errors	426
<i>Syncing and Authenticating AD Users via a Distributed Engine</i>	427
Local Versus Distributed Engine Sites	427
<i>Understanding Active Directory Automatic User Management</i>	429
Overview	429
Examples	429
<i>Example One</i>	429
<i>Example Two</i>	429
<i>Example Three</i>	430
<i>Integrate Secret Server with Azure Active Directory</i>	431
<i>Configuration Parameters</i>	432
<i>Setting Up Azure AD for SAML</i>	433
Configuration Steps	433
Adding Users to Single Sign-On in Azure AD	434
Advanced Settings	435
<i>Setting Up SAML SSO for Azure Active Directory</i>	436
Azure AD Configuration	436
Secret Server Configuration	436
Syncing Usernames in Azure AD and Secret Server	437
Advanced Certificate Signing Settings	438
<i>Create Azure App Registration</i>	439
Azure Portal Method	439
<i>Create the Application Registration</i>	439
<i>Add Client Secret to the Application Registration</i>	439

<i>Add API Permissions to the Application Registration</i>	439
Script Method	440
<i>Configure Azure Active Directory Domain</i>	441
Add Azure Active Directory Domain	441
Lightweight Directory Access Protocol (LDAP)	442
<i>Secure LDAP</i>	443
Overview	443
Troubleshooting LDAPS Connection Issues	443
<i>Syncing with OpenLDAP Directory Service</i>	444
Introduction	444
Unsupported Use Cases	444
<i>Anonymous User Authentication</i>	444
<i>Duplicate User Attributes</i>	444
Procedure	444
Discovery	447
Overview	447
In a Hurry?	447
Discovery Benefits	447
<i>Quick Initial and Ongoing Importation of Network Credentials</i>	447
<i>Protection Against Backdoor Accounts</i>	447
Discovery Types	447
<i>Active Directory Discovery</i>	447
<i>ESX/ESXi Discovery</i>	447
<i>AWS Discovery</i>	448
<i>Google Cloud Platform Discovery</i>	448
<i>Unix Discovery</i>	448
<i>Extensible Discovery</i>	448
Discovery Performance	448
How Discovery Works	449
<i>Automated Discovery</i>	449
<i>Automated Discovery Terms</i>	449
Discovery Source	449
Discovery Scanner	449
Discovery Input Template	449
Discovery Output Template	449
Discovery Rule	449
<i>Example Automated Discovery Process</i>	449
Manual Discovery	450
Discovery Platform Specifics	451
<i>Active Directory Discovery</i>	452

Active Directory Local Account Discovery Methods	453
<i>Remote Procedure Calls (RPC)</i>	453
<i>Windows Management Instrumentation (WMI)</i>	453
<i>Attempt WMI First, and Failover to RPC if Needed</i>	453
Creating an Active Directory Discovery Source	454
Setting Permissions for Active Directory Scans	459
<i>Local Windows Accounts</i>	459
<i>Windows Services, Scheduled Tasks, App Pools, and COM+ Applications</i>	459
Running and Interpreting Active Directory Discovery	461
<i>Step One: Discovery Configuration</i>	461
<i>Step Two: Discovery Scan</i>	461
<i>Step Three: Computer Scan</i>	461
<i>Step Four: Viewing Discovery Results</i>	462
<i>Browsing Discovery Results</i>	462
<i>Searching Discovery Results</i>	464
<i>Understanding Discovery Results</i>	464
AWS Account Discovery	466
AWS Instance Discovery	467
Enabling AWS Discovery	472
Password Management in AWS	473
<i>Amazon IAM Keys</i>	473
<i>Amazon IAM Console Password</i>	473
<i>Permissions Required for Secret Key Changes</i>	473
<i>Permissions Required for Changing the Amazon IAM Console Password</i>	474
Viewing AWS Discovery Source Scanners	475
Google Cloud Platform Discovery	478
Overview	478
Configuration	478
<i>Task 1: Creating GCP Service Accounts</i>	478
<i>Task 2: Setting GCP Permissions</i>	481
<i>Discovery</i>	481
<i>RPC/Heartbeat</i>	481
<i>Task 3: Creating a GCP IAM Service-Account Secret</i>	482
<i>Task 4: Creating an RPC/Heartbeat Password Changer</i>	483
<i>Task 5: Creating a GCP Discovery Source</i>	489
Viewing Discovery Scanners for the GCP Discovery Source	494
Instance Custom Filter	494
Importing Service Accounts	494
GCP APIs	496
<i>Overview</i>	496

<i>Enabling GCP APIs</i>	497
Errors and Solutions	498
<i>Create Keys Failed: Access Denied</i>	498
<i>Error</i>	498
<i>Likely Cause</i>	498
<i>Solution</i>	498
<i>Create Keys Failed: Maximum Number of Keys on Account Reached</i>	499
<i>Error</i>	499
<i>Likely Cause</i>	499
<i>Solution</i>	499
<i>Discovery Consumer: Syncing OUs Failed</i>	499
<i>Error</i>	499
<i>Likely Cause</i>	499
<i>Solution</i>	499
<i>Discovery Consumer: Syncing Machines Failed</i>	500
<i>Error</i>	500
<i>Likely Cause</i>	500
<i>Solution</i>	500
<i>Discovery Consumer: Machine Scan Completed but Computers Failed Authentication</i>	500
<i>Error</i>	500
<i>Likely Cause</i>	500
<i>Solution</i>	500
<i>Invalid Grant: Account Not Found</i>	500
<i>Error</i>	500
<i>Likely Cause</i>	500
<i>Solution</i>	501
<i>Request Error: Caller Does Not Have Permission</i>	501
<i>Error</i>	501
<i>Likely Cause</i>	501
<i>Solution</i>	501
Unix Account Discovery	502
Creating a Unix Discovery Source	503
<i>Creating the Discovery Source</i>	503
<i>Editing the Unix Discovery Source Scanners</i>	507
Discovering SSH Public Keys	514
<i>Task 1: Viewing Discovery Scanners for the Unix Discovery Source</i>	514
<i>Task 2: Adding the SSH Public Key Scanner for the Unix Discovery Source</i>	516
<i>Task 3: Importing SSH Public Keys</i>	517
<i>Task 4: Creating SSH Public Key Import Rules</i>	522
VMware ESX/ESXi Account Discovery	529

Creating an ESX/ESXi Discovery Source	530
VMware ESX/ESXi Account Discovery and RPC Configuration	534
<i>Overview</i>	534
<i>Details</i>	534
<i>Download Locations</i>	535
<i>Troubleshooting and Issues</i>	535
<i>ESXi Certificate Settings</i>	535
General Information	537
<i>Account Permissions for Discovery</i>	538
Unix	538
ESXi	538
Local Windows Accounts	538
Windows Services, Scheduled Tasks, App Pools, and COM+ Applications	539
<i>Discovery Best Practices</i>	540
Overview	540
Global Settings	540
<i>Enabling Port Scanning</i>	540
<i>Introduction</i>	540
<i>Accessing Port Scanning</i>	541
<i>Additional Reasons to Consider Discovery Port Scanning</i>	541
<i>Lowering the Discovery Scanner Timeout May Cause Issues</i>	541
<i>Secrets with Multiple Dependencies May Create Especially Long Timeouts</i>	541
<i>When to Run Discovery</i>	541
<i>Discovery Settings</i>	542
Environment-Specific Considerations	543
<i>Discovery Scan Offset Hours</i>	543
<i>Advanced Settings</i>	544
<i>Run Secret Computer Matcher Once per Discovery</i>	544
<i>Limit the Network Traffic Caused by Nested Organizational Units</i>	544
Engines and Engine Workers	545
<i>Discovery Error Messages</i>	546
<i>Discovery on Non-Domain Joined or Unix Targets</i>	547
Overview	547
Setting Credentials on a Discovery Scanner	547
Creating a Secret Search Filter	547
<i>Introduction to Discovery Sources, Scanners, and Templates</i>	549
Discovery Source	549
Discovery Scanner	549
Discovery Input Template	550
Discovery Output Template	551

Example	551
Editing and Adding Discovery Scanners	552
<i>Enabling Specific OU Domain Discovery</i>	555
<i>Creating Discovery Rules</i>	561
Creating Local Account Rules	561
Creating Dependency Rules	569
Discovery and Sites—Where Does Secret Server Run Discovery Scans?	574
Extensible Discovery	575
<i>Overview</i>	575
<i>When to Use Extensible Discovery</i>	575
<i>Extensible Discovery Tutorial</i>	575
Task One: Understanding the Process	576
Task Two: Creating the Scripts for Each Discovery Step	576
<i>Script Name: Host Range Scanner</i>	578
<i>Script Name: Machine Scanner</i>	578
<i>Script Name: Local Account Scanner</i>	579
<i>Script Name: Windows Service Dependency Scanner</i>	579
Task Three: Creating Scan Templates	580
<i>Host Range</i>	581
<i>Machines</i>	584
<i>Local Accounts</i>	585
<i>Dependencies Scan Template</i>	587
Task Four: Setting up Discovery Scanners and Sources	589
<i>Discovery Scanners</i>	589
<i>Host Ranges</i>	590
<i>Machines</i>	593
<i>Local Accounts</i>	595
<i>Dependencies</i>	596
<i>Discovery Sources</i>	597
Alerts, Auditing, Events and Logs	605
Audit Data Retention	606
<i>In This Section</i>	606
<i>Overview</i>	606
<i>Data Retention Policies</i>	606
<i>Permissions</i>	606
<i>Procedures</i>	606
Viewing the Status and History of Audit-Data Retention Policies	606
Editing Audit Data Policies	608
Running an Old Audit-Data Purge Right Now	609
Preventing PII Export in Audit Reports	611

<i>Overview</i>	611
<i>Procedure</i>	611
Enabling Debug Mode in Distributed Engine Log Files	614
<i>Overview</i>	614
<i>Procedure</i>	614
<i>Verbose Mode</i>	614
Enabling Debug Mode in System Log Files	615
<i>Overview</i>	615
<i>Procedure</i>	615
<i>Verbose Mode</i>	615
Event Pipelines	616
<i>Overview</i>	616
<i>Event Pipeline Components</i>	616
Definitions	616
<i>Event Pipeline</i>	616
<i>Event Pipeline Policy</i>	616
<i>Event Pipeline Filter</i>	616
<i>Secret Policy Filters</i>	616
<i>User Policy Filters</i>	617
<i>Event Pipeline Policy Target</i>	617
<i>Event Pipeline Task</i>	617
<i>Secret Tasks</i>	618
<i>User Tasks</i>	619
<i>Event User</i>	620
<i>Event Variable</i>	620
<i>Target User</i>	620
<i>Triggers</i>	620
<i>Secret Triggers</i>	620
<i>User Triggers</i>	621
Component Relationships	622
<i>Event Variables</i>	622
Secret Field Tokens	622
Event Setting Tokens	622
Secret Setting Tokens	623
Additional Tokens	625
Secret	625
Folder	625
Event User	625
Metadata	625
Target User	626

Custom Task Variables	626
Global Variable	626
Item Variable	626
<i>Permissions</i>	626
<i>Procedures</i>	626
Event Pipelines	626
<i>Activating or Deactivating Event Pipelines</i>	626
<i>Creating New Event Pipelines</i>	627
<i>Step One: Create EP</i>	627
<i>Step Two: Add Triggers</i>	628
<i>Step Three: Add Filters</i>	629
<i>Step Four: Choose Tasks</i>	630
<i>Editing Existing Event Pipelines</i>	631
<i>Viewing Event Pipelines</i>	631
Event Pipeline Policies	631
<i>Activating or Deactivating Event Pipeline Policies</i>	631
<i>Adding an Existing Event Pipeline</i>	632
<i>Assigning Folders and Secret Policies to Event Policy Targets</i>	632
<i>Folders</i>	632
<i>Secret Policies</i>	632
<i>Creating, Importing, and Duplicating Event Pipeline Policies</i>	633
<i>Monitoring Event Pipeline Policies</i>	633
<i>Ordering Event Pipelines in Event Pipeline Policies</i>	633
<i>Removing Event Pipelines from Event Pipeline Policies</i>	633
<i>Advanced Settings and Troubleshooting</i>	634
Configuring Advanced Settings	634
Infinite Loops	634
Event Subscriptions	635
<i>Creating Event Subscriptions</i>	637
Task 1: Creating an Event Subscription	637
Task 2: Adding Events	638
Task 3: Adding Subscribers	639
Task 4: Associating Inbox Rules	640
<i>Deleting an Event Subscription</i>	642
<i>Editing an Event Subscription</i>	643
Event List	644
Giving Application Pools Event Log Access	648
Overview	648
Required Registry Permissions	648
Applying Windows Event Log Permissions	648

Inbox	650
<i>Marking Alerts as Viewed</i>	651
<i>Using Inbox Rules</i>	653
Overview	653
<i>Inbox Rule Components</i>	653
<i>Message (Notification) Types</i>	653
<i>Rule Conditions</i>	653
<i>Predefined System Rules</i>	654
<i>Example Rule Diagram</i>	654
Procedures	655
<i>Creating a Rule from Scratch</i>	655
<i>Task 1: Create the Inbox Rule</i>	655
<i>Task 2: Add Rule Conditions</i>	657
<i>Task 3: Set up an Email Digest</i>	659
<i>Task 4: Add Subscribers to the Email or Slack Message</i>	661
<i>Creating an Inbox Rule from a Notification</i>	662
<i>Using Inbox Templates</i>	666
Overview	666
Managing Full SQL Server Transaction Logs	676
<i>Potential Solutions</i>	676
Report Auditing	677
Secret Audit Log	678
Secret Server Log List	679
<i>Secret Server Logs</i>	679
SS log	679
SS-BSSR log	679
SS-BWSR log	679
SS-EWSR log	679
<i>SS-MMSR log</i>	679
SS-SRWSR log	679
<i>Protocol Handler Log</i>	679
SS-RDPWin log	679
Enabling Debug Logging for the Protocol Handler Log	680
<i>Distributed Engine Log</i>	681
SSDE log	681
Enabling Debug Mode in Distributed Engine Log Files	681
Enabling Discovery Logging	681
<i>RabbitMQ Log</i>	681
Secure Syslog/CEF Logging	682
Overview	682

<i>Configuring a Secure TCP Syslog/CEF External Audit Server in Secret Server</i>	682
Compatible Audit Servers	682
Configuring an External Audit Server	682
<i>Caching Syslog Audits</i>	683
<i>Configure Auditing for TLS Connections</i>	683
<i>Adding Client Certificate Thumbprints</i>	684
<i>Determining the Status of a Remote Audit Server</i>	684
<i>Compatibility Notes for Client Certificates</i>	684
IIS Application Pool Certificate Permissions	685
AlienVault	685
Setting the Logging Type	686
Overview	686
Setting the Logging Level	686
Secret Server Web Servers	686
Distributed Engines	686
Viewing a User Audit Report	689
System Log	690
Viewing Event Subscription Logs	691
Folders	692
Folder Permissions	693
Personal Folders	693
Required Role Permissions for Managing Folders	693
Folder Synchronization	694
Synchronizing with the ConnectWise API	694
Synchronizing with a Database (Advanced)	696
Managing Folders	697
Assigning Secret Policies to Folders	698
Creating Folders	699
Deleting Folders	700
Editing Folder Permissions	701
Enabling Personal Folders	704
Modifying Folders with Secret Policies	706
Moving Folders	708
Moving Secrets Between Folders	709
Pinning Folders	711
Pinning Folders	711
Unpinning Folders	711
Custom Naming Pinned Folders	712
Secret Folder-Tree Panel	713
Mobile Computing	714

Setting Maximum Time for Offline Caching	715
<i>Overview</i>	715
<i>Procedure</i>	715
<i>Example</i>	716
Secret Server Networking	717
Change SQL Service Account Passwords without Restarting the SQL Service	718
<i>Requirements</i>	718
<i>Create a PowerShell Script in Secret Server</i>	718
<i>Create a New Dependency Changer</i>	719
<i>Discovery and Importing to the Right Template</i>	722
Changing SQL Server Connection Parameters	723
Checking Secret Server Site Status	725
Distributed Engines	726
<i>Overview</i>	726
<i>Architecture and Workflow</i>	726
Main Components	726
Ports	727
Security	728
Engine Workflow	728
<i>Configuring Distributed Engines</i>	728
FAQ	728
<i>Configuration and Sizing</i>	730
Requirements	730
<i>Windows Server 2012</i>	730
<i>Distributed Engine Offline and Online Events</i>	731
<i>Internal Site Connector</i>	732
<i>RabbitMQ Durable Exchanges</i>	733
Overview	733
Manually Creating Durable RabbitMQ Exchanges	734
Creating Durable RabbitMQ Exchanges with a PowerShell Script	734
<i>Using the Script</i>	734
<i>Script</i>	735
<i>RabbitMQ Naming Conventions for Queues</i>	744
Introduction	744
Secret Server Roles	744
Queue Names	744
<i>Section1</i>	744
<i>Section2</i>	745
<i>Section3</i>	745
Secret Server Roles and Queues	745

<i>Background Worker Role Queues</i>	745
<i>Active Directory Synchronization</i>	746
<i>Bulk Operation</i>	746
<i>ConnectWise Integration</i>	746
<i>Discovery</i>	746
<i>Duo Integration</i>	746
<i>Email Processing</i>	746
<i>Event Pipelines</i>	747
<i>Heartbeat and Remote Password Change</i>	747
<i>Import</i>	748
<i>Management: Backup, and Cleanup</i>	748
<i>Distributed Engine Management</i>	748
<i>Password Generation</i>	748
<i>Reports</i>	749
<i>Run Now</i>	749
<i>Scheduled Tasks</i>	749
<i>Search</i>	750
<i>SSH Terminal</i>	750
<i>Delinea Privilege Behavior Analytics Integration</i>	750
<i>Delinea Privilege Manager Integration</i>	751
<i>Delinea Telemetry</i>	751
<i>Delinea One Identify Provider Integration</i>	751
<i>Engine Role Queues</i>	751
<i>Active Directory Synchronization</i>	751
<i>Discovery</i>	752
<i>Heartbeat, Remote Password Change, and Dependency</i>	752
<i>Management</i>	753
<i>Proxy</i>	753
<i>Scripting</i>	753
<i>Syslog Integration</i>	753
<i>Delinea Privilege Behavior Analytics Integration</i>	753
<i>Ticketing System Integration</i>	754
<i>Engine Worker Role Queues</i>	754
<i>Active Directory Synchronization</i>	754
<i>Discovery</i>	754
<i>RDP Proxy, SSH Proxy, and SSH Terminal</i>	754
<i>Syslog Integration</i>	755
<i>Heartbeat, Remote Password Change, and Dependency</i>	755
<i>Delinea Privilege Behavior Analytics Integration</i>	756
<i>Distributed Engine Management</i>	756

<i>Session Recording Worker</i>	756
<i>Post Recording</i>	756
<i>Video Conversion</i>	756
<i>Post Recording (Legacy)</i>	757
<i>Management</i>	757
Security	758
Internal Site Connector	759
Ports Used by Secret Server	760
<i>Overview</i>	760
<i>Port Listing</i>	760
<i>Related Articles and Resources</i>	762
RDP Proxy Configuration	763
<i>Overview</i>	763
<i>Recommended Method</i>	763
How It Works	763
Configuration	763
Configuration Settings	764
<i>Alternative Method</i>	764
How It Works	764
Configuration	765
<i>Known Issues</i>	765
"Could not load file or assembly..." Error	765
RDP Proxy Does Not Work with FIPS Validation	765
Secret Server Clustering	767
<i>Overview</i>	767
Clustering and Background Thread Changes in 10.7.	767
Clustering Overview	767
<i>Nodes</i>	767
<i>Backbone Bus</i>	767
<i>Engine Response Bus</i>	767
<i>Worker Roles</i>	767
<i>Component Communication</i>	768
<i>Server Node Configurations</i>	768
<i>Scheduled Background Operations</i>	769
<i>Procedures</i>	770
Markdig.Syntax.Inlines.EmphasisInline	770
Upgrading Secret Server in a Clustered Environment	771
<i>Overview</i>	771
<i>Procedure</i>	772
Upgrading Database Mirroring	773

Upgrading Disaster Recovery Installations	773
Load Balancing Secret Server Clusters	773
<i>Custom URL Configuration</i>	773
<i>SSL Recommendations</i>	774
<i>Configuring Client's IP Address (X-Forwarded-For)</i>	774
<i>Clustering Errors</i>	774
Secret Server Support for HTTP/2	776
Securing Traffic with HTTP Strict Transport Security	777
SSH Cipher Support	778
<i>Secret Server with FIPS Enabled</i>	778
Default Encryption Algorithms,	778
Default Key Exchange Algorithms,	778
Default MAC Algorithms,	779
<i>Secret Server with FIPS Disabled</i>	779
Default Encryption Algorithms, Non-FIPS	779
Default Key Exchange Algorithms, Non-FIPS	781
Default MAC Algorithms, Non-FIPS	781
SSH Command Restrictions	783
<i>SSH Blocked Command Lists</i>	784
Overview	784
Requirements	784
<i>Creating SSH Blocked Command Lists</i>	784
<i>Applying SSH Command Blocked Lists in Secret Settings</i>	787
<i>SSH Command Restrictions via a Secret Policy</i>	788
<i>SSH Command Menus</i>	789
SSH Jumpbox Routes	790
<i>Best Practices for Jumpbox Routes</i>	791
Overview	791
Assumptions	791
Best Practices	791
Example Setup	791
<i>Task 1: Launching a New Linux Instance</i>	791
<i>Task 2: Configuring OpenSSH</i>	791
<i>Task 3: Configuring Public Key Authentication and Disabling Root Logins</i>	792
<i>Configuring the sshd_config File</i>	792
<i>Setting Key Exchange Algorithms, Ciphers, and MACs</i>	792
<i>Task 4: Disabling Forwarding</i>	792
<i>Task 5: Changing the Default SSH Port</i>	793
<i>Task 6: Testing Configuration and Restarting SSHD Service</i>	793
<i>Task 7: Send Your Users' SSH Keys to the Jumpbox</i>	793

Using Bastion Hosts with Secret Server and Jumpbox Routes	793
<i>References</i>	793
<i>Creating and Testing Secrets for Jumpbox Routes</i>	794
Overview	794
Task 1: Creating Jumpbox and Target Host Secrets	794
Task 2: Creating a Jumpbox Route with Secret Levels	795
Task 3: Assigning a Jumpbox Route on the Target Secret	795
Task 4: Test Launching the Target Secret with PuTTY Launcher over SSH Proxy	795
Task 5: Test Launching the Target Secret with SSH Terminal	796
<i>Creating and Editing SSH Jumpbox Routes</i>	798
Introduction	798
Prerequisites	798
Permissions	798
<i>Creating, Editing, and Deactivating Jumpbox Routes</i>	798
<i>Assigning a Jumpbox Route to a Secret</i>	801
<i>Global Jumpbox Settings</i>	802
SSH Proxy Configuration	803
<i>Enabling Proxy</i>	803
<i>Web Application Proxy Performance</i>	803
Minimum Hardware	803
Session Activity	803
<i>Proxy Connections</i>	804
<i>SSH Proxy with Multiple Nodes</i>	805
SSH Terminal Administration	806
<i>Introduction</i>	806
<i>Feature Summary</i>	806
<i>Requirements</i>	806
System Requirements	806
Recommended	806
Secret Server Permission Requirements	806
<i>Configuring SSH Terminal</i>	807
Enabling SSH Terminal on Secret Server	807
Enabling Terminal on Secret Server Distributed Engine	808
<i>Logging into the SSH Terminal</i>	808
<i>Increasing Maximum Concurrent Logins for Users</i>	809
<i>SSH Terminal Login with Two Factor Authentication</i>	809
<i>Escaping Special Characters</i>	810
<i>Terminal Commands</i>	810
man	810
Syntax	810

<i>Description</i>	810
<i>Examples</i>	810
search	810
<i>Syntax</i>	811
<i>Description</i>	811
<i>Parameters</i>	811
<i>Examples</i>	812
cat	812
<i>Syntax</i>	812
<i>Description</i>	812
<i>Parameters</i>	813
<i>Examples</i>	813
launch	813
<i>Syntax</i>	813
<i>Description</i>	813
<i>Parameters</i>	814
<i>Examples</i>	814
<i>Launching a Secret with the SSH Terminal</i>	814
Launching a Secret on a Local Site	814
Launching a Secret on a Distributed Engine Site	816
Launching a Secret upon Terminal Connection	818
<i>SSH Terminal Launching with a Custom SSH Command Allowlist</i>	818
<i>SSH Terminal Launching with Session Recording</i>	821
<i>SSH Key Pairs for Terminal</i>	822
Overview	822
Limitations	822
Enabling Users to use SSH Key Pairs to Authenticate	822
Creating SSH Key Pairs	823
Administering Public SSH Keys	823
Using SSH Keys for Authentication (PuTTY Example)	823
Remote Password Changing	824
Assigning a Password Changer to a Secret Template	825
Automatic Remote Password Changing	829
<i>Auto Change Schedule</i>	829
<i>Understanding Expiration, Auto Change and Auto Change Schedules</i>	831
Definition	831
Examples	831
<i>Scenario One: Expiration with Auto Change and No Auto Change Schedule</i>	831
<i>Scenario Two: Expiration with Weekly Auto Change</i>	831
<i>Scenario Three: Expiration with No Auto Change</i>	831

Important Considerations and Best Practices	831
Configuring Secret Dependencies for RPC	833
<i>COM+ Dependency Scanner</i>	834
Requirements for Discovery	834
<i>Windows Services</i>	834
<i>Component Services</i>	834
<i>COM+ Network Access</i>	834
Versions Supported	834
Versions Not Supported	835
Configuring COM+ Discovery for a New Domain	835
Configuring COM+ Discovery for an Existing Domain	835
<i>Creating Custom Dependencies</i>	837
<i>Dependency Groups</i>	838
<i>Dependency Settings and Information</i>	839
<i>Manually Adding Dependencies</i>	840
<i>Secret Dependency Status</i>	843
Account Dependence	843
Account Clusters	843
Viewing Dependency Status	844
<i>Secret Dependency Failures</i>	844
<i>Secret Dependency Not Run</i>	844
<i>Secret Dependency Overview</i>	845
<i>Secret Dependency Status</i>	846
<i>Using Regex with Dependencies</i>	847
Overview	847
UNC Names	847
Examples	848
<i>XML Configuration Files</i>	848
<i>Example One</i>	848
Source	848
Regex	848
<i>Example Two</i>	848
Source	848
Regex	848
<i>Windows Initialization (.ini) Files</i>	848
Source	848
Regex	848
<i>SQL Server Connection Strings</i>	848
Source	848
Regex	848

<i>Oracle Connection Strings</i>	849
<i>Example One</i>	849
<i>Source</i>	849
<i>Regex</i>	849
<i>Example Two</i>	849
<i>Source</i>	849
<i>Regex</i>	849
<i>YAML</i>	849
<i>Source</i>	849
<i>Regex</i>	849
Create a New Dependency Changer for Synchronizing Passwords During RPC	850
<i>Requirements</i>	850
<i>PowerShell Script</i>	856
Custom Password Changers	857
<i>Changing Ports and Line Endings</i>	858
<i>Creating a Custom Password Changer for IBM AS/400</i>	859
<i>Creating a Custom Password Changer for IBM AS/400 in Secret Server 10.5.</i>	859
<i>Configuration</i>	859
<i>Create an AS/400 password changer from an existing z/OS Mainframe password changer:</i>	859
<i>Modify the AS/400 IBM iSystem password changer commands:</i>	859
<i>Modify the AS/400 password changer for 5250 emulation and commands:</i>	861
<i>Create an AS/400 template from the z/OS Secret Template:</i>	861
<i>Modify the AS/400 Secret Template to use the AS/400 Password Changer:</i>	862
<i>Creating a Custom Password Changer</i>	865
<i>Advanced Post Change Commands</i>	866
<i>Advanced Settings</i>	867
<i>A Note About Commands</i>	868
<i>Creating a Custom Password Changer for Cisco ASA</i>	869
<i>Authenticate As</i>	869
<i>Commands</i>	869
<i>Creating a Custom SSH Password Changer</i>	870
<i>Deactivating Password Changers</i>	871
<i>Distributed Engines and RPC</i>	872
<i>Editing Custom Commands</i>	873
<i>RPC-Mapped Text-Entry Fields</i>	873
<i>Associated Reset Secrets</i>	873
<i>Check-Result Commands</i>	873
<i>Enabling RPC</i>	875
<i>Mapping Account Fields for RPC</i>	876
<i>Mapping an SSH Key or Private Key Passphrase for Authentication</i>	877

<i>Minimum Requirements for Windows Local Accounts</i>	879
<i>Modifying Password Changers</i>	880
<i>Password Changing Scripts</i>	881
Creating Scripts	881
Testing Scripts	881
Using Scripts	881
Viewing Audits	881
<i>RPC Error Codes</i>	882
<i>RPC for Service Accounts and SSH Keys</i>	883
Service Accounts	883
SSH Keys	883
<i>RPC Logs</i>	884
<i>Running a Manual RPC</i>	885
<i>Treating Specific Heartbeat "Unknown Errors" as Connection Failures</i>	886
Procedure	886
<i>Triggering an RPC When Defined Errors Occur</i>	889
Procedure	889
Minimum Permissions for Active Directory Remote Password Changing	893
<i>Overview</i>	893
<i>Setting Permissions</i>	893
Configuring Oracle DB 19c for Heartbeat and RPC	895
<i>Introduction</i>	895
<i>Procedure</i>	895
Task One: Installing the Oracle Database Access Components	895
Task Two: Configuring Secret Server	895
Task Three: Configuring a Secret Server Distributed Engine	896
<i>Troubleshooting</i>	896
Log Files	896
Errors	897
Password Changer List	898
<i>Overview</i>	898
<i>List</i>	898
PostgreSQL and ODBC Remote Password Changing	900
<i>Overview</i>	900
<i>Create an ODBC Password Changer</i>	900
<i>Example Reset Commands</i>	900
<i>Adding Connection Strings</i>	900
Adding Connection Strings to Password Changer Settings	900
Adding Connection Strings to Secrets	901
<i>Troubleshooting</i>	901

<i>PostgreSQL with Distributed Engines</i>	901
Privileged Accounts and Reset Secrets	902
Remote Password Changing Errors	903
<i>Overview</i>	903
<i>Errors</i>	903
The user name cannot be found	903
Change password failed: Unknown. (ERROR_CANT_ACCESS_DOMAIN_INFO)	903
Change password failed: Unknown. (NERR_PasswordPolicySettings)	903
Change password failed: Unknown. (ERROR_ACCESS_DENIED)	904
Change password failed: Unknown. (ERROR_INVALID_PASSWORD)	904
Change password failed: Unknown. (ERROR_ACCOUNT_LOCKED_OUT)	904
ExpiredSecretMonitor - Unspecified error	904
DirectoryEntry.Invoke SetPassword - The network path was not found.	904
Secret '	904
Error changing password - Check Out is enabled on associated Secret.	904
Remote Password Changing on SQL Server Accounts	905
<i>Overview</i>	905
<i>Creating the Account</i>	905
<i>Assign Permissions</i>	905
<i>Using the Account</i>	905
Remote Password Changing with PowerShell	907
<i>Overview</i>	907
<i>Procedure</i>	907
Task 1: Creating the Active Directory Verify Password Script	907
Task 2: Creating the Active Directory Change Script	907
Task 3: Testing the Scripts	908
Task 4: Configuring a Password Changer for Secret Server Version 10.0.000006 and Later	908
Task 5: Creating a Secret Template	909
Task 6a: Finishing the Secret Template Configuration for Secret Server 10.0.000006 and later	909
Task 6b: Finishing the Secret Template Configuration for Secret Server 8.8.000000 to 10.0.000000	910
Task 7: Creating Secrets Using PowerShell Remote Password Changing	910
<i>Errors</i>	911
Running Heartbeat and RPC for Office 365 Accounts with a PowerShell Module	913
<i>Procedure</i>	913
<i>Troubleshooting</i>	913
Salesforce.com Password Changer	914
SAP Heartbeat and Password Changing	915
<i>For Secret Server 8.8.000000 and Higher</i>	915
<i>For Secret Server 8.7.000000 and Below</i>	915
Using SQL Privileged Account for RPC	916

<i>Overview</i>	916
<i>Task 1: Creating an Account</i>	916
<i>Task 2: Assigning Permissions</i>	916
<i>Task 3: Using the Account</i>	916
Reports	917
Built-in Reports	919
<i>Activity</i>	919
<i>Discovery Scan</i>	919
<i>Folders</i>	919
<i>Groups</i>	919
<i>Legacy Reports</i>	920
<i>Password Compliance</i>	920
<i>Report Schedules</i>	920
<i>Roles and Permissions</i>	920
<i>Secret Policy</i>	920
<i>Secrets</i>	920
<i>System Reports</i>	921
<i>User</i>	921
Creating and Editing Reports	922
<i>Creating a Custom Report</i>	922
<i>Editing Reports</i>	925
<i>Report SQL Scripts</i>	926
<i>Overview</i>	926
<i>Dynamic Parameters</i>	926
<i>Viewing Secret Server SQL Database Information</i>	926
<i>Database Paging</i>	926
Deleting or Undeleting Reports	927
Modifying Report Categories	928
Report Page	929
<i>Reports General Tab</i>	929
<i>Reports Security Hardening Tab</i>	929
<i>Configuration Section</i>	929
<i>Database Section</i>	930
<i>Environment Section</i>	930
<i>SSL Section</i>	931
<i>Reports User Audit Tab</i>	931
Reporting and Dual Controls	932
Saving Reports to File	935
Scheduled Reports	937
<i>Creating New Schedules for Reports</i>	937

<i>Viewing Existing Report Schedules</i>	939
<i>Editing Schedule Settings</i>	939
Using Dynamic Parameters in Reports	941
<i>Primary Parameters</i>	941
#STARTDATE	941
#ENDDATE	941
#USER	941
#ORGANIZATION	941
#GROUP	942
#FOLDERID	942
#FOLDERPATH	942
#CUSTOMTEXT	942
<i>Additional Parameters</i>	942
Parameters	942
Example	943
<i>Coloring Your Reports</i>	943
Viewing Auditing for a Report	944
Viewing Reports	945
Roles	946
Assigning Roles to a User	947
Creating Roles	948
Editing Role Permissions	949
Secret Server Role Permissions List	950
<i>Overview</i>	950
<i>Complete List</i>	950
Access Offline Secrets on Mobile	950
Add Secret	950
Add Secret Custom Audit	950
Administer Active Directory	950
Administer Automatic Export	950
Administer Backup	950
Administer Configuration	950
Administer Configuration Proxying	950
Administer Configuration SAML	950
Administer Configuration Security	951
Administer Configuration Session Recording	951
Administer Configuration Two Factor	951
Administer Configuration Unlimited Admin	951
Administer ConnectWise Integration	951
Administer Create Application Accounts	951

Administer Create Users	951
Administer Custom Password Requirements	951
Administer Data Retention	951
Administer DevOps Secrets Vault Tenants	951
Administer Disaster Recovery	951
Administer Discovery	952
Administer Distributed Engine Configuration	952
Administer DoubleLock Keys	952
Administer Dual Control	952
Administer Event Subscriptions	952
Administer Export	952
Administer Folders	952
Administer Inbox	952
Administer Groups	952
Administer HSM	952
Administer IP Addresses	952
Administer Jumpbox Route	952
Administer Key Management	952
Administer Languages	953
Administer Licenses	953
Administer Lists	953
Administer Metadata	953
Administer Nodes	953
Administer OpenID Connect	953
Administer Password Requirements	953
Administer Pipelines	953
Administer Platform Integration	953
Administer Remote Password Changing	953
Administer Reports	953
Administer Role Assignment	953
Administer Role Permissions	953
Administer Scripts	953
Administer Search Indexer	954
Administer Secret Policy	954
Administer Secret Templates	954
Administer Security Analytics	954
Administer Session Monitoring	954
Administer SSH Menus	954
Administer System Log	954
Administer Teams	954

Administer Template Custom Columns	954
Administer Users	954
Administer Workflows	954
Advanced Import	954
Allow Access Challenge	955
Allow List Secret Access for Assigning Policy	955
Approve Via Duo Push	955
Assign Pipelines	955
Assign Secret Policy	955
Browse Reports	955
Bypass Direct API Authentication Restriction	955
Bypass SAML Login	955
Copy Secret	955
Create Root Folders	955
Deactivate Secret	955
Delete Secrets from Reports	955
Download Automatic Export	955
Edit Secret	956
Erase Secret	956
Expire Secrets from Reports	956
Force Check In	956
Own Group	956
Own Secret	956
Own User	956
Personal Folders	956
Privilege Manager Administrator	956
Privilege Manager Helpdesk User	956
Privilege Manager MacOS Admin	956
Privilege Manager Windows Administrator	956
Privilege Manager User	956
Rotate Encryption Keys	957
Run Automatic Export	957
Run Disaster Recovery Data Replication	957
Session Recording Auditor	957
Unlimited Administrator	957
Unrestricted by Teams	957
User Audit Expire Secrets	957
View About	957
View Active Directory	957
View Advanced Dashboard	957

View Advanced Secret Options	957
View Backup	958
View Configuration	958
View Configuration Proxying	958
View Configuration SAML	958
View Configuration Security	958
View Configuration Session Recording	958
View Configuration Two Factor	958
View Configuration Unlimited Admin	958
View ConnectWise Integration	958
View Data Retention	958
View Deleted Secrets	958
View DevOps Secrets Vault Tenants	958
View Disaster Recovery	958
View Discovery	959
View Distributed Engine Configuration	959
View DoubleLock Keys	959
View Dual Control	959
View Event Subscriptions	959
View Export	959
View Folders	959
View Group Roles	959
View Groups	959
View HSM	959
View IP Addresses	959
View Jumpbox Route	959
View Key Management	959
View Launcher Password	959
View Licenses	960
View Lists	960
View Nodes	960
View OpenID Connect	960
View Password Requirements	960
View Pipelines	960
View Platform Integration	960
View Remote Password Changing	960
View Reports	960
View Roles	960
View Scripts	960
View Search Indexer	960

View Secret	960
View Secret Audit	961
View Secret Password and Private Key History	961
View Secret Policy	961
View Secret Templates	961
View Security Analytics	961
View Security Hardening Report	961
View Session Monitoring	961
View Session Recording	961
View SSH Menus	961
View System Log	961
View Teams	961
View User Audit Report	961
View Users	961
View Workflows	961
Web Services Impersonate	962
Secret Checkout	963
Checking out Secrets	964
Checkout Hooks	965
<i>Overview</i>	965
<i>Checkout User Variables for Scripts</i>	965
Configuring Password Changing on Check-in	966
Exclusive Access	967
Secret Checkout Expiration	968
Secret DoubleLocks	970
Assigning a DoubleLock to a Secret	971
Assigning a User a DoubleLock Password	973
Assigning Users to Existing DoubleLocks	975
Creating a DoubleLock and a DoubleLock Password	979
DoubleLock Objects and Relationships	984
Password Loss and Assignment	985
Resetting a DoubleLock Password	986
Using a DoubleLock	988
Secret Heartbeats	989
Alerts on Heartbeat Failure	990
Configuring Heartbeat	991
Enabling Heartbeat in RPC	992
Heartbeat Logs	993
Heartbeat Status Codes	994
Remote Accounts Supported	995

Running Heartbeat for a Secret	996
Secret Import and Export	997
Overview	997
What Gets Imported or Exported	997
Migrating to and from Secret Server Cloud	998
Automatic Secret Export	999
<i>Export Process</i>	999
<i>Considerations and Settings</i>	999
Configurations and Returned Data	999
<i>Export Storage</i>	1000
<i>Security</i>	1000
<i>Permissions</i>	1000
<i>Event Subscriptions</i>	1000
Setting up Automatic Exports	1001
Automatic Secret Export REST API	1005
<i>Overview</i>	1005
<i>Viewing the Storage List</i>	1005
Sample Request	1005
Sample Response	1005
<i>Downloading Secret Exports</i>	1005
Sample Request	1005
Sample Response	1006
Exporting Secrets	1007
Importing Secrets	1009
<i>Importing CSV Data</i>	1009
<i>Importing Secrets with XML</i>	1011
Procedure	1011
Example XML File	1012
Notes	1012
Sample XML	1012
Secret Server Migration Tool	1016
Secret Launchers and Protocol Handlers	1017
Automatic Sudo or Su Privilege Elevation	1018
Built-In Launcher Types	1019
Configure RDP Launcher Domain for Windows Account Template	1020
<i>Problem</i>	1020
<i>Solution</i>	1020
Custom Launcher for SecureCRT (SSH)	1021
<i>Step 1: Creating the Custom Launcher</i>	1021
<i>Step 2: Creating a Custom Secret Template (optional)</i>	1024

<i>Step 3: Associating the Launcher with a Secret Template</i>	1024
Custom Launchers	1026
<i>Creating Custom Launchers</i>	1027
Procedure	1027
Settings	1030
<i>General Settings</i>	1030
<i>Windows Settings</i>	1031
<i>Mac Settings</i>	1031
<i>Custom Launcher Errors</i>	1032
<i>Custom Launcher Process Arguments</i>	1033
Syntax	1033
Examples	1033
<i>Creating a Custom TOAD Launcher</i>	1034
<i>Creating and Implementing an Ultra VNC Custom Connection Launcher</i>	1036
Create an Ultra VNC Custom Connection Launcher	1036
Assign the Launcher to a Template	1038
Enabling CAC/PIV Smart Cards for Secret Launchers	1039
<i>Overview</i>	1039
<i>Enabling Globally with User Settings</i>	1039
<i>Enabling on a Specific Secret</i>	1039
Enabling Launchers	1040
<i>Introduction</i>	1040
<i>MSI Installer</i>	1040
<i>Installing by Group Policy</i>	1041
Launcher Configuration and Support	1042
<i>Adding a Program Folder to the Windows PATH</i>	1043
<i>Common Launcher Errors</i>	1044
<i>Configuring Launchers on the Secret</i>	1045
<i>Configuring SSH Proxies for Launchers</i>	1046
<i>Default Launcher Requirements</i>	1049
<i>Managing Superuser Privilege</i>	1050
<i>Session Recording and Launchers</i>	1053
Launching Sessions	1054
Limiting Launcher Domains	1055
Managing Multiple Secret Server Instances with Protocol Handlers and Launchers	1056
<i>Prerequisites</i>	1056
<i>Setup Steps and Configuration</i>	1056
<i>Manually Updating Protocol Handler</i>	1057
Protocol Handler Administrative Settings	1058
<i>Available Settings</i>	1058

Allowed Secret Server Domains	1058
Disable Auto-Update	1058
<i>Configuration Methods</i>	1058
Choosing the Configuration Method	1058
Configuring GPOs	1058
Configuring Settings During Secret Server Installation	1059
Remote Desktop Launchers	1060
<i>Adding Remote Desktop Launchers</i>	1061
<i>Browser Configuration</i>	1062
<i>Editing RD Launchers</i>	1063
<i>Enforcing a Secret's RDP Launcher Policy</i>	1064
<i>Setting Up Secret Templates for RD Launchers</i>	1067
Removing the Mac Launcher	1068
Secret Server Session Connector	1069
<i>Overview</i>	1069
<i>Connection Sequences</i>	1069
<i>Download</i>	1071
<i>Configuration</i>	1071
Task 1: Reviewing RDS Server Prerequisites	1071
Task 2: Setting up RDS Services	1072
<i>Step 2.1: Installing Remote Desktop Services—Remote Desktop Session Host</i>	1072
<i>Step 2.2: Setting up RDS in \$1Secret Server\$2</i>	1074
<i>Step 2.3: Configuring Session Connector Settings</i>	1075
Task 3: Setting up RDS	1075
<i>Step 3.1: Installing the Secret Server RDS Protocol Handler</i>	1075
<i>Step 3.2: Adding the Remote Desktop Collection and Application</i>	1075
<i>Step 3.3: Configuring RDS-related Group Policy Settings</i>	1083
Task 4: Installing the Secret Server Session Connector	1083
Task 5: Updating API Credentials	1084
Task 6: Launching Session Connector Sessions	1084
<i>Subprocedures</i>	1084
Creating RDS Application Accounts	1084
Enabling Application Account RDS Credential Sharing	1086
Configuring Session Connector Custom Launchers	1087
Assigning Session Connector Custom Launchers to Secret Templates	1088
Troubleshooting Session Connector	1091
Uninstalling Session Connector	1091
Session Connector Downloads	1092
Using Connect As Command and SSH Proxy with a PuTTY Launcher	1093
<i>Overview</i>	1093

<i>Setting up SSH Proxy to Use the Connect As Feature</i>	1093
Web Launchers	1095
<i>Configuring Web Launchers for Secrets</i>	1096
<i>Creating a Configuration</i>	1097
<i>Launching to a Website</i>	1098
Secret Management	1099
All Secrets Page	1100
<i>Secret Columns</i>	1100
<i>Quick Launch and Actions</i>	1101
<i>Other Features</i>	1101
Procedures	1102
<i>Creating Secret Policies</i>	1103
<i>Creating Secrets</i>	1105
<i>Customizing the All-Secrets Page</i>	1108
Customizing Visible Columns	1108
Filtering Search Results	1108
Sizing Columns	1108
<i>Deactivating and Reactivating Secrets</i>	1109
<i>Duplicating Secrets</i>	1110
<i>Editing Secrets</i>	1111
<i>Erasing Secrets</i>	1112
Task 1: Configuring Secret Erase	1112
Task 2: Erasing a Secret	1117
<i>Overriding the Secret Template's Password Requirements</i>	1120
<i>Setting Up Password Masking</i>	1121
<i>Sharing Secrets</i>	1122
Permissions	1122
Password Visibility	1122
Procedure	1123
<i>Viewing Secrets</i>	1125
Searching and Search Indexer	1126
<i>Searching for Secrets</i>	1126
<i>Search Indexer</i>	1126
Secret Configuration Options	1129
<i>Common Configuration Options</i>	1129
<i>Advanced Configuration Options</i>	1129
Secret Expiration	1130
<i>Forcing Expirations</i>	1131
<i>Resetting Expired Secrets</i>	1132
<i>Setting up Secret Templates for Secret Expiration</i>	1133

<i>Setting up Secrets</i>	1134
Secret Navigation Slideout	1135
<i>Favorites</i>	1135
Dropdown Quick List	1135
Favorites Page	1136
<i>Recent</i>	1136
<i>Most Used Secrets</i>	1136
<i>Shared with Me</i>	1136
Secret Permissions	1137
Secret Tabs	1138
<i>Secret Dependencies Tab</i>	1139
<i>Secret Expiration Tab</i>	1140
<i>Secret Launcher Tab</i>	1141
<i>Secret Personalize Tab</i>	1142
<i>Secret RPC Tab</i>	1143
<i>Secret Security Tab</i>	1144
Checkout	1144
Approval	1144
Password Requirements	1144
Other Security	1144
Secret Server Cloud	1145
AWS Key Management in Secret Server Cloud	1146
<i>Introduction</i>	1146
<i>Configuring Key Management</i>	1146
Overview	1146
<i>Key Management Providers</i>	1146
AWS Key Management Services Pricing	1146
<i>Procedure</i>	1146
Task 1: Setting up the Encryption Key and IAM User in AWS	1147
Task 2: Adding Encryption Key and User Details in Secret Server	1148
Task 3: Secret Key Rotation	1148
<i>Secret Server Key Management via the REST API</i>	1148
Migrating Secret Server Online to Secret Server Cloud	1149
<i>Introduction</i>	1149
<i>What Is Migrated</i>	1149
<i>Procedure</i>	1149
Task One: Exporting from Secret Server Online	1149
Task Two: Importing to Secret Server Cloud	1151
Task Three: Finishing up	1152
Secret Server Cloud Quick Start	1153

<i>Overview</i>	1153
<i>Cloud Versus On-Premise Secret Server</i>	1153
<i>Getting Started</i>	1153
System Requirements	1153
Engine Connectivity	1153
Initial Setup	1153
Distributed Engine	1154
Configure Active Directory Integration	1155
Test Heartbeat and Remote Password Changing	1155
<i>Next Steps</i>	1156
<i>Troubleshooting and Resources</i>	1156
Get Error: "Site (Default) engines are not currently online" When Saving Domain	1156
Secret Server End User Guide	1158
What Is Secret Server?	1158
What Is the Purpose of the End User Guide?	1158
Getting Help	1158
Logging on Secret Server	1158
Secrets	1159
Secret Folders	1160
Using Secrets on Websites (Web Password Filler)	1160
Checking out Secrets	1160
Getting Notified of Secret Events	1161
Learning More About Secret Server—the Getting Started Tutorial	1161
Secret Templates	1162
Creating and Editing Custom Password-Exclusion Dictionaries	1163
<i>Creating a Custom Dictionary</i>	1163
<i>Editing a Custom Password-Exclusion Dictionary</i>	1167
How to Use the Privileged Password Security Policy Template	1171
List of Built-in Secret Server Templates Secret Server	1172
<i>Built-in Secret Templates Available Out-of-the-box</i>	1172
Managing Secret Templates	1174
<i>Activating and Deactivating Templates</i>	1175
<i>Changing a Secret's Template</i>	1176
<i>Configuring Secret Template Permissions</i>	1177
<i>Create and Customize an IBM iSystem (AS/400) Template to use the new IBM iSeries (AS/400) Password Changer</i>	1181
Create an AS/400 Secret Template	1181
Modify Your AS/400 Secret Template to use the AS/400 Password Changer	1182
Customize Your AS/400 Password Changer for Your Environment	1183
Additional Functions, Adjustments, and Parameters	1184
<i>Creating a Unix Account Secret Template that Uses Key Authentication Instead of a Password</i>	1186

Create the New Template	1186
Disable	1187
<i>Creating or Editing Secret Templates</i>	1189
Editing Secret Templates	1189
Creating or Importing a New Template	1190
<i>Task 1: Creating the Template</i>	1190
<i>Task 2: Adding General Settings and Setting an Expiration or One-Time Password</i>	1192
<i>Task 3: Defining Fields for the Template</i>	1194
<i>Task 4: Mapping Launchers and RPC Type</i>	1195
<i>Task 5: Adding Permissions</i>	1196
<i>Task 6: Viewing the Template's Audit Trail</i>	1198
Settings for Specific Template Types	1199
<i>Oracle Account as SYS</i>	1199
<i>SQL Windows Authentication Account Secret Template and Launcher</i>	1199
Configuring SAP SNC Account Secret Templates	1200
<i>Introduction</i>	1200
<i>New Template Fields</i>	1200
<i>Server-Side Setup</i>	1200
Prerequisites	1200
<i>SAP Server Setup</i>	1200
<i>SAP NCO Files</i>	1201
<i>SAP Cryptographic Library</i>	1201
<i>SAP Server Certificate</i>	1201
Personal Security Environment Setup	1201
Importing PSE information to the SAP GUI	1202
<i>Creating an SAP SNC Secret in Secret Server</i>	1209
<i>Troubleshooting</i>	1209
Secret Template Settings	1210
<i>Field Slug Names</i>	1211
<i>Secret Template Fields</i>	1212
Field Types	1212
Editing Fields	1212
Text-Entry Field and Control Settings	1212
<i>Secret Template List Fields</i>	1214
Overview	1214
Adding a New List Field	1214
<i>Task 1: Create the List</i>	1214
<i>Task 2: Create a Template Using the List</i>	1218
<i>Task 3: Create a Secret</i>	1222
SSH Authentication Templates	1225

Template Character Sets	1226
Template Naming Patterns	1227
Template Password Requirements	1228
<i>Overview</i>	1228
<i>Creating a Custom Password Requirement</i>	1228
Security and Hardening	1234
Accessing MS SQL Server with IWA	1235
<i>Introduction</i>	1235
<i>Creating a Domain Service Account</i>	1235
<i>Granting Access to SQL Server database</i>	1235
<i>Assigning Account as Identity of Application Pool</i>	1235
Introduction	1237
<i>The Common Criteria Standard</i>	1237
<i>Common Criteria for Secret Server</i>	1237
<i>Audience</i>	1237
<i>Enabling Common Criteria Security Hardening</i>	1238
Security Hardening Checklist	1238
Evaluated Configuration	1238
Assumptions and Operational Environment	1238
Configuring TLS	1239
<i>Manually Disabling TLS Version 1.0</i>	1239
<i>TLS Diffie-Hellman Hardening Overview</i>	1239
<i>Restricting Server Cipher Suites for TLS</i>	1239
<i>Allowed Suites</i>	1239
<i>Changing Cipher Suites with the IIS Crypto Tool</i>	1239
<i>Configuring TLS with IIS</i>	1241
<i>Enabling TLS Auditing</i>	1241
<i>Configuring TLS with Active Directory</i>	1241
<i>Configuring TLS with Syslog</i>	1241
Additional Common Criteria Configurations	1242
<i>Configuring X.509v3 Certificates</i>	1242
<i>Enabling DPAPI</i>	1242
<i>Enabling FIPS Mode</i>	1242
<i>Ensuring Zero Information Disclosure</i>	1242
Configuring Custom Error Messages	1242
Hiding the Application Version Number	1242
Configuring the Login Banner	1243
Configuring Account Lockout	1243
Disabling "Remember Me" Logins	1243
Configuring SQL Server	1244

Running the IIS Application Pool with a Service Account	1244
Assigning Common Criteria Roles and Permissions	1247
Managing User Passwords	1247
Configuring Secret Templates	1247
Setting Authentication Strength for Non-Password Credentials	1247
Configuring Remote Password Changing for SSH Key Rotation	1247
Configuring External Auditing	1247
<i>Connecting to an External Audit Server</i>	1247
<i>Configuring Local Windows Event Log Auditing</i>	1247
Installing Your System	1248
Prerequisites	1248
<i>Minimum Hardware</i>	1248
<i>Recommended Software</i>	1248
<i>Operational Environment</i>	1248
SQL Installation and Configuration	1248
<i>Installing SQL Server 2016 Standard Edition</i>	1248
<i>Installing SQL Server Management Studio</i>	1260
<i>Configuring SQL Server for Secret Server Installation</i>	1263
<i>Create the SQL Server User Account</i>	1263
<i>Add the User Account to SQL Server</i>	1266
<i>Additional Pre-Requisites for Secret Server</i>	1268
<i>To run the Secret Server IIS Application Pool with a Service Account</i>	1268
Secret Server Installer	1269
<i>Downloading and Verifying the Installer</i>	1270
<i>Installer Verification and Authentication</i>	1270
<i>Running the Installer</i>	1274
<i>Additional Permissions for Secret Server Folder</i>	1283
Accessing Your System	1284
Browser Compatibility	1284
Web Interface	1284
Local Admin Account	1286
Installing License Keys	1286
Local Admin Account	1288
Installing License Keys	1289
Updating Secret Server	1290
Checking Software Version	1290
<i>Verify Software Version and Components</i>	1290
<i>Downloading Software Updates</i>	1290
Configuring Authentication and Login	1291
Local Authentication	1291

Domain Authentication	1291
Account Lockout Configuration	1291
<i>Lockout Window</i>	1291
<i>How to Disable "Allow Remember Me" during Logins</i>	1291
Configuring the Login Banner	1291
<i>How to Modify the Login Banner Messaging</i>	1292
Configuring Session Timeouts	1292
Configuring IP Address Restrictions	1292
Managing Local Users	1293
Create New Local Users	1293
Manage Local Users	1293
Managing Domain Users	1294
Configuring Active Directory Sync	1294
<i>Create a Sync Secret</i>	1294
<i>Specify the Domain & Enable Active Directory Integration</i>	1295
<i>Configuring TLS with Active Directory</i>	1297
<i>Setup Synchronization Groups</i>	1298
Managing Domain Credentials	1298
<i>Updating Domain Credentials</i>	1298
<i>Failed Domain Authentication</i>	1298
Common Criteria Roles and Permissions	1300
Assigning Roles to Users	1300
Management Functions Based On Role	1301
Managing User Passwords	1304
Password Requirements for User Authentication	1304
Resetting User Authentication	1305
Setting Local User Password History Requirements	1305
Managing Secrets	1307
Creating Secrets	1307
Configuring Secret Templates	1309
<i>Object</i>	1310
<i>Attribute</i>	1310
<i>Secret</i>	1310
<i>Attributes Inherited from Template</i>	1310
<i>Command Restrictions</i>	1310
<i>Field Data</i>	1310
<i>Folder</i>	1310
<i>Password Requirements Rule Override</i>	1310
<i>Policy Identifier</i>	1310
<i>Subject Identifier</i>	1310

<i>Secret Name</i>	1310
<i>Template</i>	1311
<i>Field Parameters</i>	1311
<i>Password Change Policy</i>	1311
<i>Password Strength Policy</i>	1311
<i>Secret Expiration Policy</i>	1311
<i>Secret Name Pattern</i>	1311
<i>Template Description</i>	1311
<i>Template Name</i>	1311
<i>Template Status</i>	1311
<i>Secret Access Policy</i>	1311
<i>Secret Modification Policy</i>	1311
Configuring Password Policy for Secret Templates	1312
Authentication Strength for Non-Password Credentials	1314
Configuring Remote Password Changing for SSH Key Rotation	1315
<i>Security Overview for SSH Key Rotation and PuTTY Launcher</i>	1315
<i>Creating a Unix Account (SSH Key Rotation) Secret</i>	1315
<i>Enabling the Launcher</i>	1316
<i>Using the Launcher</i>	1316
<i>Enable Remote Password Changing</i>	1317
<i>Rotate SSH Key Remotely</i>	1317
Secret Expiration	1319
<i>Setting up Secret Expiration for the Secret template</i>	1320
<i>Setting up Secret Expiration for the Secret</i>	1320
<i>Forcing Expiration</i>	1320
<i>Resetting an Expired Secret</i>	1320
<i>AutoChanging an Expired Secret</i>	1320
<i>Setting the Password Change Retry Interval</i>	1320
Configuring Common Criteria	1323
Verify Settings in Government Edition	1323
<i>Verify DPAPI Setting is Enabled</i>	1323
<i>Verify FIPS Mode is Enabled</i>	1323
<i>Manually Enabling FIPS on Windows Local Server</i>	1323
<i>Manually Disabling TLS v1.0</i>	1324
<i>Microsoft Cryptographic Engine</i>	1325
Configuring X.509v3 Certificates	1325
<i>Installing Certificate in Trust Store</i>	1325
<i>Configuring Web Certificate</i>	1326
<i>Configuring Client Certificates</i>	1328
<i>Certificate Identifiers Overview</i>	1330

Configuring Transport Layer Security (TLS)	1330
<i>TLS Configuration with AD</i>	1330
<i>TLS Configuration with Syslog</i>	1330
<i>TLS Configuration with IIS</i>	1331
Local Auditing	1332
Understanding Local Audit Records and Reports	1332
Accessing Local Reports	1332
Configuring Local Windows Event Log Auditing	1334
<i>Granting Application Pool Access to Windows Event Log</i>	1334
<i>Required Registry Permissions</i>	1334
<i>How to Apply Windows Event Log Permissions</i>	1335
Accessing Windows Event Logs	1337
External Auditing	1338
Security - Connecting to External Audit Server	1338
Configuring Syslog/CEF External Audit Server	1338
<i>Compatible Audit Servers</i>	1338
<i>Configuring Your External Audit Server</i>	1338
<i>Caching Syslog Audits</i>	1339
<i>Compatibility Notes Related to Using Client Certificates</i>	1339
Configuring Auditing for TLS Connections	1339
Determining Status of a Remote Audit Server	1340
Management Functions	1341
Roles and Management Functions	1341
Considerations for an Externally Accessible Secret Server	1343
<i>Limiting the Attack Surface</i>	1343
<i>Using Secure Connections</i>	1343
<i>Setting Up Remote Password Changing</i>	1343
Enabling Common Criteria Security Hardening	1344
Introduction	1344
Overview	1344
Audience	1344
What Is Common Criteria?	1344
Procedures	1344
Security Hardening Checklist	1344
Configuring TLS	1344
<i>Manually Disabling TLS Version 1.0</i>	1345
<i>TLS Diffie-Hellman Hardening Overview</i>	1345
<i>Restricting Server Cipher Suites for TLS</i>	1345
<i>Allowed Suites</i>	1345
<i>Changing Cipher Suites with the IIS Crypto Tool</i>	1345

<i>Configuring TLS with IIS</i>	1347
<i>Enabling TLS Auditing</i>	1347
<i>Configuring TLS with Active Directory</i>	1347
<i>Configuring TLS with Syslog</i>	1347
Additional Common Criteria Configurations	1348
<i>Configuring X.509v3 Certificates</i>	1348
<i>Enabling DPAPI</i>	1348
<i>Enabling FIPS Mode</i>	1348
<i>Ensuring Zero Information Disclosure</i>	1348
Configuring Custom Error Messages	1348
Hiding the Application Version Number	1348
Configuring the Login Banner	1349
Configuring Account Lockout	1349
Disabling "Remember Me" Logins	1349
Configuring SQL Server	1350
Running the IIS Application Pool with a Service Account	1350
Assigning Common Criteria Roles and Permissions	1353
Managing User Passwords	1353
Configuring Secret Templates	1353
Setting Authentication Strength for Non-Password Credentials	1353
Configuring Remote Password Changing for SSH Key Rotation	1353
Configuring External Auditing	1353
<i>Connecting to an External Audit Server</i>	1353
<i>Configuring Local Windows Event Log Auditing</i>	1353
Hardware Security Modules	1354
<i>Introduction</i>	1354
<i>HSM Requirements</i>	1354
<i>Silent HSM Operation</i>	1354
<i>Configuring HSM Integration</i>	1355
<i>Rotating the HSM Key</i>	1355
<i>Securing HSM Integration</i>	1355
<i>HSM Redundancy</i>	1356
<i>Testing HSM CNG Configuration</i>	1356
Master Encryption Key Rotation	1357
<i>Overview</i>	1357
<i>Rotation Procedure</i>	1357
Secret Server Telemetry	1361
<i>Overview</i>	1361
<i>Checking for and Downloading Updates</i>	1361
<i>License Activation</i>	1361

<i>Reporting Anonymized Usage Metrics</i>	1361
<i>Setting and Viewing Secret Server Telemetry</i>	1362
Securing ASP Cookies	1364
Securing IIS Server	1365
<i>Accounts</i>	1365
<i>Auditing and Logging</i>	1365
<i>Code Access Security</i>	1365
<i>Files and Directories</i>	1365
<i>IIS Metabase</i>	1365
<i>ISAPI Filters</i>	1365
<i>Machine.config</i>	1366
<i>Patches and Updates</i>	1366
<i>Ports</i>	1366
<i>Protocols</i>	1366
<i>Registry</i>	1366
<i>Script Mappings</i>	1366
<i>Server Certificates</i>	1366
<i>Services</i>	1366
<i>Shares</i>	1367
<i>Sites and Virtual Directories</i>	1367
<i>Other Considerations</i>	1367
<i>Hiding HTTP Header Information</i>	1368
Hide the IIS Version	1368
Hide the ASP.NET Version	1368
Hide the Server Type	1368
Security Hardening Guide	1369
<i>Introduction</i>	1369
<i>Overview</i>	1369
<i>Best Practices</i>	1369
General	1369
Active Directory	1369
Database	1369
Application Server	1370
Application Settings	1370
<i>Security Hardening Report</i>	1370
Configuration Section	1371
<i>Allow Approval for Access from Email</i>	1371
<i>File Attachment Restrictions</i>	1372
<i>Frame Blocking</i>	1372
<i>Force Password Masking</i>	1372

<i>Login Password Requirements</i>	1372
<i>Maximum Login Failures</i>	1373
<i>Remember Me</i>	1373
<i>Secure Session and Forms Auth Cookies</i>	1373
<i>Markdig.Syntax.Inlines.EmphasisInline</i>	1373
<i>Zero Information Disclosure Error Message</i>	1373
Database Section	1374
<i>SQL Account Using Least Permissions</i>	1374
<i>SQL Server Authentication Password Strength and Username</i>	1374
<i>Windows Authentication to Database</i>	1374
Environment Section	1374
<i>Application Pool Identity</i>	1374
<i>DPAPI or HSM Encryption of Encryption Key</i>	1375
SSL Section	1375
<i>Require SMTP SSL</i>	1375
<i>Require SSL</i>	1375
<i>SSL/TLS Hash</i>	1375
<i>SSL/TLS Key</i>	1376
<i>SSL/TLS Protocols</i>	1376
<i>Using HTTP Strict Transport Security</i>	1376
Security Settings Not in the Hardening Report	1376
<i>Apply TLS Certificate Chain Policy and Error Auditing</i>	1376
<i>Enable FIPS Compliance</i>	1377
<i>Key Rotation</i>	1377
Two-Factor Authentication	1377
<i>SAML</i>	1377
<i>Email</i>	1377
<i>Soft Tokens</i>	1377
<i>RADIUS</i>	1377
<i>Duo Security</i>	1378
<i>Enabling Two-Factor Authentication</i>	1378
<i>Enabling for Users</i>	1378
<i>Enabling per Domain</i>	1378
Roles	1378
<i>Controlling Access to Features Using Roles</i>	1378
<i>Limiting Role Access to the Export Permission</i>	1378
<i>Unlimited Administration Mode</i>	1378
<i>Limiting Role Access to Secret Templates</i>	1378
<i>Monitoring Roles with Event Subscriptions</i>	1379
<i>Using Two Roles for Access to Unlimited Administration Mode</i>	1379

<i>Encryption</i>	1379
DPAPI Encryption	1379
<i>Overview</i>	1380
<i>Enabling and Disabling DPAPI</i>	1380
<i>Using Clustering with DPAPI</i>	1380
Protecting Your Encryption Key Using EFS	1380
SSL (TLS) and HSTS	1381
SSH Key Validation	1381
Mapping an SHA1 Digest to Secrets	1381
Validating SHA1 Digests for Unix Account Discovery	1382
<i>Disabling IIS HTTP Headers</i>	1382
Introduction	1382
Procedure	1382
<i>Adjusting CORS Policy Headers</i>	1382
<i>Additional Resources</i>	1383
Session Recording	1384
Basic Session Recording	1384
Advanced Session Recording	1386
Session Recording Tab	1387
Caveats and Recommendations	1388
<i>General</i>	1388
<i>Database</i>	1388
<i>Network Bandwidth and Video</i>	1388
<i>Session Recording</i>	1388
<i>macOS Catalina Security</i>	1389
Configuring Session Recording	1391
<i>Overview</i>	1391
<i>Configuration</i>	1391
Using Legacy Video Codecs	1391
Enabling Session Recording on Secrets	1391
Extending Session Recording with Custom Launchers	1392
<i>Record Multiple Windows Option</i>	1392
<i>Record Additional Processes Option</i>	1393
<i>Example</i>	1393
Advanced Session Recording	1393
<i>Metadata Recording</i>	1393
<i>Record All Sessions</i>	1393
Session Recording Settings	1394
<i>Hide Recording Indicator</i>	1394
<i>Enable On-Demand Video Processing</i>	1394

<i>Enable Inactivity Timeout (Minutes)</i>	1394
<i>Max Session Length (Hours)</i>	1394
<i>Use Hardware Acceleration</i>	1394
<i>Save Videos to</i>	1395
<i>Archive Location Dependent on Site</i>	1395
<i>Folder Path</i>	1395
<i>Encrypt Archive on Disk</i>	1395
<i>Enable Archiving to Disk</i>	1395
<i>Enable Deleting</i>	1395
<i>Setting Notes</i>	1395
<i>Using Network Share Path</i>	1395
<i>Configuring the Maximum Concurrent Recording Sessions per Web Node</i>	1397
Installing the Advanced Session-Recording Agent	1398
<i>Overview</i>	1398
<i>How Advanced Session Recording Agents Work</i>	1398
Record All Sessions	1399
<i>Secret Server Configuration</i>	1399
SSH Metadata	1399
Remote Desktop Metadata	1399
Session Recording Worker Role	1400
Advanced Session Recording Agent	1400
<i>Agent Manual Installation</i>	1400
<i>Agent Updates</i>	1400
<i>Agent Uninstallation</i>	1400
<i>Agent Group Policy Installation</i>	1400
Task 1: Review the Prerequisites	1400
Task 2: Download the Advanced Session Recording Agent Installer	1400
Task 3: Customize the Installer	1401
Task 4: Set up a Network Share	1403
Task 5: Create a Group Policy with Software Installation to install the MSI	1403
Task 6: Link your Group Policy Object to an OU	1404
Task 7: Verify Configuration at the Domain Level	1404
Task 8: Verify the Configuration of a Domain Member	1404
Session Recording Requirements	1406
<i>Advanced Session Recording</i>	1407
<i>Basic Session Recording</i>	1408
Stability and Compatibility with Older ASRAs	1409
<i>Enabling Inactivity Timeout</i>	1410
<i>Enabling On-Demand Video Processing</i>	1411
<i>Record All Sessions</i>	1412

<i>Recording Metadata</i>	1413
System Capacity Specifications	1414
Secret Server Setup	1415
Installation	1416
<i>Advanced (Manual) Installation</i>	1417
Procedure	1417
<i>Step 1: Downloading the Secret Server Application Files</i>	1417
<i>Step 2: Creating Folders and Extracting Contents</i>	1417
<i>Step 3: Configuring IIS</i>	1417
<i>Step 4a: Installing Secret Server as a Virtual Directory</i>	1417
<i>Step 4b: Installing Secret Server as a Website</i>	1418
<i>Step 5: Completing Secret Server Installation from the Website</i>	1418
Troubleshooting Notes	1419
<i>Basic (Automatic) Installation</i>	1420
Introduction	1420
<i>Secret Server Is an ASP.NET Website</i>	1420
<i>SQL Server Is Usually Required</i>	1420
<i>Administrative Access</i>	1420
<i>Review the Prerequisites</i>	1420
<i>System Requirements Overview</i>	1420
<i>Additional Recommendations</i>	1420
Procedure	1420
<i>Step 1: Downloading the Latest Version of Secret Server</i>	1420
<i>Step 2: Running the Installer</i>	1421
<i>Welcome Page</i>	1421
<i>Database Page</i>	1421
<i>Pre-Requisites Page</i>	1421
<i>Database Connection Page</i>	1421
<i>Create User Page</i>	1421
<i>Email Server Page</i>	1421
<i>Review Page</i>	1421
<i>Install Page</i>	1421
<i>Step 3: Reviewing the Log Files (Optional)</i>	1421
<i>Step 4: Opening Secret Server</i>	1421
<i>Step 5: Learning Secret Server</i>	1422
<i>Choosing a SQL Server Edition to Use with Secret Server</i>	1423
SQL Server Express Edition	1423
SQL Server Standard Edition	1423
SQL Server Enterprise Edition	1423
<i>Creating and Using a SQL Server Privileged Account</i>	1424

Overview	1424
Procedure	1424
<i>Task 1: Creating an Account</i>	1424
<i>Task 2: Assigning Permissions</i>	1424
<i>Step 3: Using the Account</i>	1424
Enabling SQL Server Encryption	1426
Manual IIS Installation	1428
Roles and Features	1428
<i>Roles</i>	1428
<i>Features</i>	1429
Step One: Windows Server 2012–2019 IIS Installation	1429
Step Two: Configure the IIS Website	1430
Step Three: Ensure IIS Does Not Stop the Worker Process	1431
Step Four: Ensure the User Profile Always Loads	1432
Installing and Configuring SQL Server	1433
Creating a SQL Account	1433
<i>SQL Authentication</i>	1433
<i>Windows Authentication</i>	1433
Configuring Database Access in Secret Server	1433
<i>SQL Location</i>	1433
<i>SQL Authentication</i>	1433
Installing RabbitMQ	1435
Overview	1435
<i>What is RabbitMQ?</i>	1435
<i>Why Install It?</i>	1435
<i>RabbitMQ and Encryption</i>	1435
Downloading Delinea's RabbitMQ Helper	1435
Prerequisites	1435
<i>General</i>	1435
<i>SSL Certificate</i>	1435
Installation	1436
<i>Task 1: Secret Server</i>	1436
<i>Task 2: RabbitMQ Host</i>	1437
Troubleshooting	1438
Clearing RabbitMQ Message Queues	1439
Installing Secret Server via the Command Line	1440
Overview	1440
Install Prerequisites	1440
<i>Parameters</i>	1440
<i>Prerequisites</i>	1441

<i>Single-Line Example</i>	1441
Installing Applications	1441
<i>Secret Server Parameters</i>	1441
<i>Privilege Manager Parameters</i>	1441
<i>Required Database Parameters</i>	1442
<i>Email Parameters</i>	1442
<i>Single-Line Example</i>	1442
Moving Secret Server to Another Machine	1443
Moving the Microsoft SQL Server Database to Another Machine	1444
Task 1: Backing up and Restoring the Database	1444
Task 2: Connecting Secret Server to the New Database	1444
Running the IIS Application Pool As a Service Account	1446
Overview	1446
Procedure	1446
<i>Task 1: Creating a Domain Service Account</i>	1446
<i>Task 2: Granting Access to the SQL Database</i>	1446
<i>Task 3: Assigning the Identity of Application Pools</i>	1447
<i>Task 4: Granting Folder Permissions</i>	1448
<i>Task 5: Configuring User Rights</i>	1448
<i>Option 1: Setting User Rights Assignment on the Domain</i>	1449
<i>Option 2: Setting User Rights Assignment Locally</i>	1449
SQL Server 2016 Standard Edition Installation	1451
Overview	1451
Procedures	1451
<i>Installing SQL Server 2016</i>	1451
<i>Installing SQL Server Management Studio</i>	1456
<i>Creating the SQL Server Database</i>	1458
<i>Adding a SQL Server User</i>	1458
SQL Server Authentication Configuration	1460
Enabling Mixed Mode	1460
Enabling Named Pipes and SQL Browser	1460
SQL Server 2014 Express Edition Installation	1461
Overview	1461
Procedures	1461
<i>Downloading SQL Server Express with Tools</i>	1461
<i>Installing SQL Server Express 2014</i>	1462
<i>Creating the SQL Server Database</i>	1469
<i>Adding a SQL Server User</i>	1469
Licensing	1471
<i>Understanding Licenses</i>	1471

<i>Activating Licenses</i>	1471
<i>Licensing Limited Mode</i>	1471
<i>Adding, Activating, Converting, and Deleting Licenses</i>	1472
Adding and Activating Secret Server Licenses Online or Offline	1472
Converting Evaluation Licenses	1476
Deleting Secret Server Licenses	1476
<i>License Activation FAQ</i>	1477
Secret Server Download Hashes	1479
11.2.000003 (GA Version)	1479
<i>Earlier Versions</i>	1479
11.2.000002 (GA Version)	1479
11.2.000000 (EA Version)	1479
11.1.000012 (Security Release)	1479
11.1.000007 (GA Version)	1479
11.1.000006	1480
11.1.000000 (EA Version)	1480
11.0.000008	1480
11.0.000007	1480
11.0.000006	1481
11.0.000005	1481
10.9.000064	1481
10.9.000063	1482
10.9.000033	1482
<i>Downloaded February 24th 2021 or Later</i>	1482
<i>Downloaded Before February 24th 2021</i>	1482
10.9.000005	1482
10.9.000032	1483
10.9.000002	1483
10.9.000000	1483
10.8.000004	1483
10.8.000000	1484
10.7.000059	1484
9.1.000001	1484
8.4.000004	1484
System Requirements for Secret Server	1485
<i>Minimum Requirements for Basic Deployments</i>	1485
<i>Recommended Requirements for Basic Deployments</i>	1485
<i>Minimum Requirements for Advanced Deployments</i>	1485
<i>Recommended Requirements for Specific Features</i>	1486
<i>Notes</i>	1486

Upgrading	1488
<i>Ensuring the Upgrade Is Secure</i>	1488
<i>Minimizing Upgrade Downtime</i>	1489
Introduction	1489
Procedures	1489
<i>Load Balanced Configuration Upgrade</i>	1489
<i>Prerequisites</i>	1489
<i>Procedure</i>	1489
<i>Manual Rolling Upgrade</i>	1491
Introduction	1491
Prerequisites	1491
Procedure	1491
Task One: Uploading the Upgrade	1491
Task Two: Verifying SQL Changes (Wizard Step One)	1494
Task Three: Generating the Upgrade File (Wizard Step Two)	1494
Task Four: Generating the SQL Script (Wizard Step Three)	1495
Task Five: Backing up and Staging (Wizard Step Four)	1496
Task Six: Starting Upgrade Mode (Wizard Step Five)	1498
Task Seven: Upgrading Web Nodes (Wizard Step Six)	1499
Task Eight: Finishing up (Wizard Step Seven)	1500
Troubleshooting and Notes	1500
<i>Rolling Back to the Previous Version</i>	1500
<i>Version Guard</i>	1501
<i>New Advanced Configuration Setting</i>	1502
<i>New Audit Type</i>	1502
<i>Secret Server and Secret Server Cloud .NET Framework 4.8 Mandatory Upgrade</i>	1503
Introduction	1503
Preparing Secret Server for the December Release	1503
<i>Secret Server Web Nodes</i>	1503
<i>Distributed Engines</i>	1503
<i>Protocol Handler</i>	1504
<i>Advanced Session Recording Agent (ASRA)</i>	1504
<i>Session Connector</i>	1505
Effects on Connection Manager	1505
Identifying Distributed Engine Servers	1505
Unaffected Secret Server Components	1506
Determining Your .NET Framework Version	1506
Installing .NET Framework 4.8	1507
<i>Secret Server Cloud IP Address Change for March to May 2021</i>	1508
Overview	1508

FAQ	1508
<i>Who Is Affected?</i>	1508
<i>What Other Delinea Products Are Affected?</i>	1508
<i>What Delinea Domains Are Affected?</i>	1508
<i>How Do I Verify My Domain Is Affected?</i>	1508
<i>What Do I Need to Change?</i>	1508
When Are The IP Addresses and Hostnames Changing?	1509
<i>Will I Lose Connectivity During the Change?</i>	1509
<i>How Will I Know When the Change Is Complete?</i>	1509
<i>Who Do I Contact If I Have Issues After the Change?</i>	1509
Delinea Support	1509
<i>Support Portal</i>	1509
<i>Telephone</i>	1509
<i>Americas</i>	1509
<i>EMEA</i>	1509
<i>APAC</i>	1509
IP Addresses and Hostnames	1509
<i>New IP Addresses and Hostnames</i>	1509
<i>Old IP Addresses and Hostnames</i>	1511
<i>Upgrading Secret Server</i>	1512
How Upgrades Work	1512
Before You Begin	1512
How to Upgrade	1512
<i>Upgrading Secret Server with Web Clustering</i>	1516
Introduction	1516
Before Beginning	1516
Upgrading a Clustered Environment	1516
EFS and DPAPI Encryption	1516
Upgrading Database Mirroring	1517
Upgrading Remote DR Instances	1517
Error Conditions	1517
<i>Upgrading Secret Server Without Outbound Access</i>	1518
How Upgrades Work	1518
Procedure	1518
<i>Step 1: Open the Upgrade Secret Server Wizard</i>	1518
<i>Step 2: Get and Upload the Latest .zip File</i>	1519
<i>Step 3: Upgrade Secret Server</i>	1519
Offline Installation Download Files	1519
<i>Upgrading to 10.9.000005/33</i>	1521
How Upgrades Work	1521

Before You Begin	1521
How to Upgrade	1521
Prerequisites	1527
<i>System Requirements</i>	1527
<i>Hardware Requirements</i>	1527
<i>Software Requirements</i>	1527
Checklist	1527
SQL Server	1527
Application Server	1527
<i>Secret Server Major Browser Support Secret Server</i>	1528
Using Chrome to Access Secret Server	1528
Uninstalling Secret Server	1529
<i>Task 1: Deleting the Database</i>	1529
<i>Task 2: Deleting the Virtual Directory</i>	1530
<i>Task 3: Deleting Secret Server Files</i>	1531
<i>Decommissioning a Secret Server Node</i>	1532
Ticketing System Integration	1533
Introduction	1533
Ticket System Tab	1533
Ticket Number Validation	1533
<i>Overview</i>	1533
<i>Configurable Settings</i>	1533
<i>Setting a Ticket Pattern Regex</i>	1534
Third-Party Integrations	1534
Atlassian JIRA Integration (PowerShell)	1535
<i>Requirements</i>	1535
<i>Ticket Number Validation Pattern (Regex)</i>	1535
<i>Validating Ticket Status</i>	1535
<i>Adding Comments to Tickets</i>	1536
BMC Remedy Integration	1538
<i>Overview</i>	1538
<i>Requirements</i>	1538
<i>Configurable Settings</i>	1538
Validating Ticket Status	1538
View Ticket URL Template	1538
Ticket Number Format Pattern (Regex)	1538
Ticket Number Validation Error Message	1539
Service Endpoint URL	1539
System Credentials	1539
Authentication	1539

Add Comments to Ticket	1539
Comment Work Type	1539
<i>Testing Your Integration Setup</i>	1539
<i>BMC Remedy Error Messages</i>	1539
ManageEngine ServiceDesk Plus Integration (PowerShell)	1541
<i>Requirements</i>	1541
<i>Ticket Number Validation Pattern (Regex)</i>	1541
<i>Validating Ticket Status</i>	1541
<i>Adding Comments (Notes) to Tickets</i>	1542
PowerShell Ticketing Integration	1543
<i>Configurable Settings</i>	1543
View Ticket URL Template	1543
Ticket Number Validation Pattern (Regex)	1543
Ticket Number Validation Error Message	1543
The PowerShell RunAs Credentials	1543
System Credentials	1543
<i>Validating Ticket Status</i>	1543
Overview	1543
Sample Script	1543
<i>Adding Comments to Tickets</i>	1544
<i>Adding Comments to a General Audit Log</i>	1544
ServiceNow Integration	1545
<i>Introduction</i>	1545
<i>Requirements</i>	1545
<i>Configurable Settings</i>	1545
View Ticket URL Template	1545
Ticket Number Format Pattern (Regex)	1545
Ticket Number Validation Error Message	1545
Domain Name	1545
System Credentials	1546
Add Comments to Ticket	1546
<i>Testing your Integration Setup</i>	1546
Troubleshooting and Notices	1547
Application Pool Discovery Over Distributed Engines	1548
<i>80070005 Error</i>	1548
Application Pool Load User Profile Setting Must Be Enabled	1550
Changing IIS to Not Stop Worker Process in IIS 7.0 and Later	1551
<i>Overview</i>	1551
<i>Procedure</i>	1551
Finding the Version Number of Your Secret Server Release	1553

<i>Through Secret Server</i>	1553
<i>Through Windows File Explorer</i>	1553
<i>Through SQL</i>	1553
<i>Through the API</i>	1553
HTTP 404.2 Error ISAPI/CGI Restrictions Stopping .NET Framework 4.5.1	1554
Error	1554
<i>Resolution</i>	1554
HTTP Error 404.17 - Not Found After Upgrading .NET Framework Version	1555
<i>Error</i>	1555
<i>Resolution</i>	1555
Windows Server 2012 or 2012 R2	1555
Invalid Oracle Data Source Length	1556
<i>Error</i>	1556
Error Text	1556
Scenario	1556
<i>Resolution</i>	1556
Notice: jQuery CVE-2019-11358	1557
<i>Relevance</i>	1557
<i>Technical Issue</i>	1557
<i>Resolution</i>	1557
<i>Related Articles and Resources</i>	1557
Notice: jQuery CVE-2020-11022	1558
<i>Relevance</i>	1558
<i>Technical Issue</i>	1558
<i>Resolution</i>	1558
<i>Related Articles and Resources</i>	1558
Security Advisory 2019	1559
<i>Detection</i>	1559
<i>The Security Issue</i>	1559
<i>Common Vulnerability Scoring System Version 3.0</i>	1559
<i>Products Affected</i>	1559
<i>Recommended Actions</i>	1559
Troubleshooting SAML Configuration Errors After Upgrading	1560
<i>Initial Troubleshooting</i>	1560
<i>Additional Troubleshooting</i>	1560
Troubleshooting Google Authenticator	1561
<i>Solution A (preferred)</i>	1561
<i>Solution B</i>	1561
Troubleshooting Heartbeat and RPC Errors for Linux Secrets	1562
<i>Step 1: Verifying Ports and Connectivity</i>	1562

<i>Step 2: Testing Heartbeat and RPC in Secret Server</i>	1563
<i>Step 3: Troubleshooting Heartbeat or RPC Outside of Secret Server</i>	1570
Troubleshooting Invalid Domain Errors	1578
<i>Troubleshooting Procedure</i>	1578
<i>Configuring the DNS Record on Your Server</i>	1579
<i>Resolving Other DNS Issues</i>	1579
Troubleshooting Quartz Trigger Jobs	1581
<i>Quartz-Related Tables</i>	1581
<i>Trigger Job Components</i>	1581
<i>Simple Trigger Job</i>	1581
<i>Cron Trigger Job</i>	1582
<i>Procedures</i>	1582
Viewing Trigger Job State (QRTZ_TRIGGERS Table)	1582
Viewing Trigger Firing Times	1582
Adjusting Trigger Job Frequency	1583
Troubleshooting SSH Issues	1584
<i>Local Servers with Direct Access</i>	1584
<i>Remote Servers</i>	1584
<i>Logging from the Client Perspective</i>	1584
<i>Understanding SSH Logging</i>	1584
Example	1584
Confirming Proper Operation	1585
VMware Issues	1587
Windows Local-Account Access-Denied Error Workaround PowerShell Scripts	1588
<i>Overview</i>	1588
<i>Additional Requirements</i>	1588
<i>Remediation Options</i>	1588
<i>Option 3: Modifying the Default GPO</i>	1588
PowerShell Script Description	1588
Download	1588
Script Argument Help	1589
<i>Command Prompt Help</i>	1589
<i>Parameters</i>	1589
-ComputerNames (string)	1589
-Username (string)	1589
-GroupName (string)	1589
-ForceGPUUpdate	1589
<i>Examples</i>	1589
Related Articles and Resources	1590
<i>Option 4: Creating a Heartbeat GPO Workaround</i>	1590

User Groups	1593
Assigning Group Owners	1594
Assigning Users to Groups	1598
Creating User Groups	1601
User Teams	1602
What Are Secret Server Teams for?	1602
Team-Related Permissions	1602
Configuring Teams Management	1603
Creating Teams	1604
Deactivating Teams	1609
Editing Teams	1611
Troubleshooting Teams	1616
Viewing a User's Teams	1617
Users	1620
Bulk Operations on Users	1621
Configuring Users	1622
Creating Users	1623
Deleting Users	1624
Password Settings	1625
Removing Deactivated User PII	1626
<i>Overview</i>	1626
<i>Removing the PII</i>	1626
<i>Active Directory Considerations</i>	1626
Unlocking Local Accounts	1627
User Login Settings	1628
User Owners	1629
User Preferences	1630
<i>General Tab</i>	1630
<i>Launcher Tab</i>	1630
User Restriction Settings	1631
User Settings	1632
Webservices	1633
Enabling Webservices	1634
Integrated Windows Authentication Webservice	1635
Secret Workflows	1636
Multi-Level Workflow	1636
Multiple Approvers to Advance	1636
Approval Process Workflow	1636
Workflow Versus Basic Access Requests	1637
Workflow Step Timeout	1637

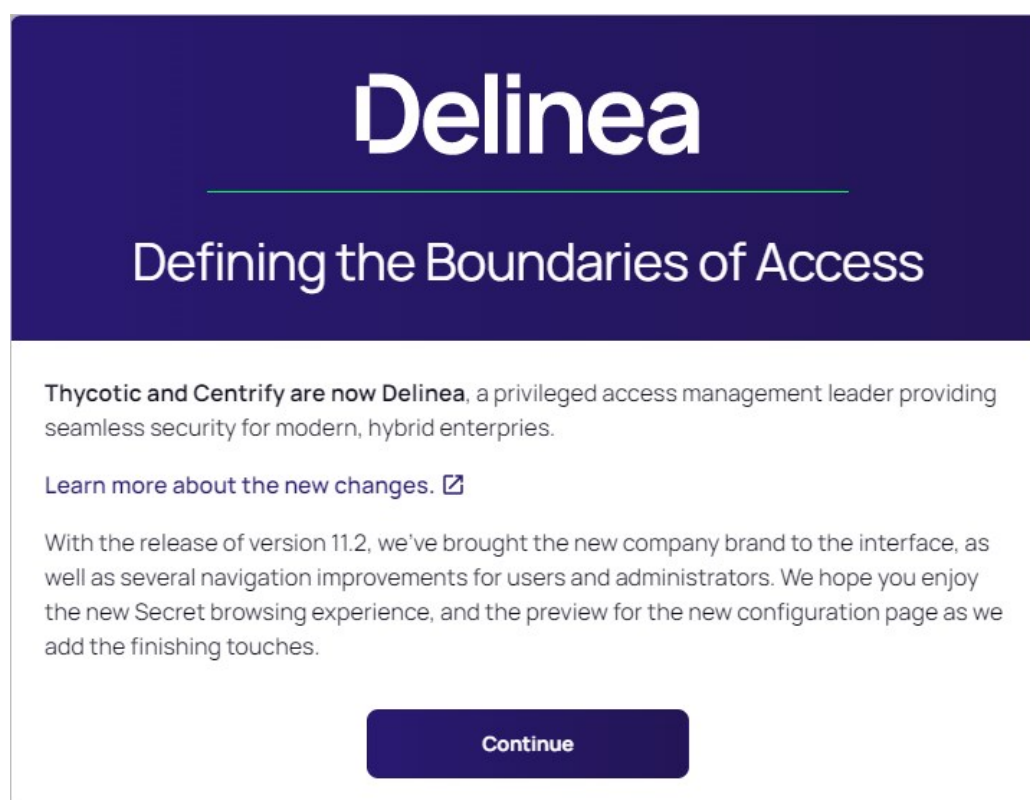
Accessing the Workflow Designer	1639
Assigning Workflows to Secret Policies	1640
Creating New Workflows	1643
<i>Markdig.Syntax.Inlines.EmphasisInline</i>	1643
<i>Markdig.Syntax.Inlines.EmphasisInline</i>	1644
<i>Markdig.Syntax.Inlines.EmphasisInline</i>	1645
Deleting Workflows	1647
Duplicating Workflows	1648
Editing Workflows	1650
Understanding Workflow Design Best Practices	1651
Secret Server Release Notes	1652
Current	1652
Secret Server On-Premises Legacy	1652
Secret Server Cloud Legacy	1653
Documentation Releases	1653

Introduction

Delinea Secret Server is an enterprise-grade, privileged access management solution that is quickly deployable and easily managed. With Secret Server, you can automatically discover and manage your privileged accounts through an intuitive interface, protecting against malicious activity, enterprise-wide. This section of the Delinea Document Portal (TDP) supports Secret Server.

Note: Navigate using the dynamic table of contents on the left, the page contents on the right, or by entering a search term above. Many pages in this documentation have sub-pages. The container (parent) pages can have introductory text or simply a heading with no text. Please click the table of contents on the left to see any sub-pages it might have.

Figure: Introductory Popup for Secret Server 11.2



Documentation

- [Delinea Documentation Portal](#): You are at the home page of the current Delinea Document Portal for Secret Server. It contains:
 - New material.
 - Converted knowledge base articles. These are usually marked as *deprecated* in the legacy knowledge base. They also may be missing all together because they were archived.
 - Links to legacy knowledge bases article that have yet to be converted or archived. There are very few of these left.
 - Links to legacy PDF documentation. These are also rare, and if you do find one, its target is very likely out of date.

- [Knowledge Base Articles](#): Use the Search text box at the top of the page. This is the legacy platform that almost completely replaced with the Delinea Documentation Portal (where you are right now). There might still be a few locations here where this online documentation links to legacy documentation.

- [End User Guide](#) (for non-technical users)
- [Getting Started Tutorial](#) (for technical users)
- [Installation Guides](#)
- [System Requirements](#)

- [Best Practices](#)
- [Discovery Best Practices](#)
- [Secret Server Government Edition—Common Criteria Hardening Guide](#) (PDF)
- [Security Hardening Guide](#)

- [Distributed Engine Security](#) (PDF)
- [Launcher Security](#) (PDF)
- [Meltdown and Spectre Security Information](#) (PDF)
- [*nix Management](#) (PDF)
- [Security Model](#) (PDF)
- [Web Services Security](#) (PDF)

Help

- [Document Conventions](#)
- [Secret Server Glossary](#)
- [Self-Help Resources](#)
- [Technical Support](#)

[Product Downloads](#)

[Release Notes](#) (On-Premises and Cloud)

[Delinea Secret Society](#): An educational community, replacing the Forum.

Note: Some of these tutorials feature legacy versions of Secret Server.

[Video Tutorials](#)

Developer Resources

[Developer Resources](#)

Getting Started Tutorial

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server is a powerful application with many facets. As such, approaching it for the first time can be daunting. To counter that, we created this section, which is an introductory guided tutorial, for new users. The tutorial suggest an order to learn topics and points to specific sections of documentation for details.

Important: This tutorial is oriented toward system administrators and other technical professionals. We recommend that non-technical users start with our [End User Guide](#). For Secret Server Cloud, see the [Secret Server Cloud Quick Start](#).

Below are our suggested guidelines for preparing to run a trial or proof-of-concept (POC) of Secret Server.

System Requirements

Please review the detailed [System Requirements for Secret Server](#). The *Minimum Requirements* are for trial, sandbox, and POC environments. The *Recommended Requirements* are for production deployments.

Hardware Requirements

Secret Server can be installed on a physical server or virtual machine.

If you would like to set up front-end (application) clustering, you need to have two or more servers available.

For testing of high availability for the SQL Server, you can use either existing Microsoft AlwaysOn infrastructure or database mirroring. If you choose to test this, this is something your database team needs to prepare in advance.

Software Requirements

Checklist

- Windows Server 2012 or newer (recommended) (one server, minimum)
- SQL Server (one instance, minimum)
- Application server prerequisites
- SSL certificate

SQL Server

You can create the SQL database in an existing SQL instance, or a new installation of SQL Server. For high availability, this needs to be a paid edition of SQL Server (not SQL Express). If you are using a new installation of SQL Server, please have this installed beforehand.

Detailed instructions for installation and configuration of SQL Server are included in one of the installation guides below (choose the guide matching the OS that SQL server will be installed on).

Application Server

We recommend installing Secret Server on Windows Server 2012 or greater. Include IIS, ASP.NET and .NET Framework. Refer to the System Requirements KB above to view prerequisite details.

Application Configuration

Service Account

Set up a service account:

1. Log on as a batch job (on the server that Secret Server runs on)
2. Modify permissions to the Secret Server application directory (typically C:\inetpub\wwwroot) and C:\Windows\temp.
3. Provide access to your SQL Server instance by adding the db_owner permission to the Secret Server database.

For detailed instructions on how to configure the permissions for the service account, see [Running Secret Server IIS Application Pool with a Service Account](#). The installation guides include instructions for assigning db_owner permission to the service account in SQL Server.

If you would like to test features that rely on Active Directory, such as AD group sync or discovery, you should also have accounts available with the appropriate permissions (described below). One option is to use the same account for both features.

Active Directory Group Sync

Active Directory group synchronization means that Secret Server can automatically add users and enable or disable them to log into Secret Server based off of their Active Directory group membership. You can choose which groups to sync. When configuring AD group sync in Secret Server, you are required to specify an account that can read the properties of users and groups. See [AD Synchronization Rights for Synchronization Account](#) for a detailed list of required permissions.

Discovery

To test discovery, please have some machines available for Secret Server to connect to for discovering accounts. An account is required to sync with AD and also scan the machines found for Windows local account and service account discovery. [Account Permissions for Discovery](#) describes the permissions required for an AD account to be used for discovery.

Test Accounts

We recommend having a few test accounts available to represent the types of accounts you want to manage using Secret Server. These could be local Windows accounts, service accounts running scheduled tasks or services, SQL server accounts, and others.

Email Notifications

To test email notifications, which can be used for event subscription notifications or requests for approval to passwords, you need configuration information for the company SMTP server:

- Service account to run the application and connect to SQL
- Domain (test or production)
- Domain account to be used for AD sync and discovery
- Test machines (if testing discovery)
- Test accounts
- SMTP server settings

SSL Certificate

We recommend setting up SSL (or https) for access to Secret Server. To do so, you will need an SSL certificate. You may use an existing wildcard certificate, create your own domain certificate, or purchase a third-party SSL certificate for the Secret Server.

Firewalls and Ports

Secret Server must connect directly to a target system to change its password. For devices that are firewalled off from Secret Server, remote agent can provide connectivity to them, but they also require connectivity from them to the target systems for password changing.

Please see [Ports Used by Secret Server](#) for a list of ports needed by Secret Server for password changing, discovery, and other features.

Process

Run the Installer: Secret Server comes with an installer that walks you through the entire process from start to finish. Once you have the prerequisites ready to go, download and run your installer, and the wizard will take you through the installation process. Please see our [installation articles](#).

Licenses

See the [Licensing](#) section.

The Secret Server Dashboard is the main page for searching and viewing secrets. Nearly everything you do in Secret Server starts with the Dashboard. See [Secret Server Dashboard](#) for details.

As you start using Secret Server, we strongly recommend configuring the following security settings. While these are optional, setting them is a best practice.

Local Admin Account Best Practices

Even if you plan to [integrate with Active Directory](#) to log into Secret Server, chances are you will need to use this account again. This is the first account you created during the installation process. Keep this account secure and avoid being locked out of Secret Server by following these suggestions:

- Store the credentials in a secure location that you can access if you lose all access to Secret Server.
- Enable the **Allow Users to Reset Forgotten Passwords** setting to provide a way of resetting the password if account is locked out or if the password is forgotten:
 1. Select **Admin > Configuration**. The Configuration page appears.
 2. Click the **Local User Passwords** tab to locate the setting.
 3. Click the **Edit** button to edit the setting.
 4. Click the **Save** button when finished.

Note: This requires having an [SMTP server configured](#) (KB).

- Configure the other **Local User Passwords** settings to enforce your password requirements, expiration, password history, and other password policies.

SSL (HTTPS) Best Practice

We recommend requiring SSL access to Secret Server. This requires setting up an SSL certificate for the website, preferably with a domain certificate. However, if you don't have a certificate, see [Installing a Self-Signed Certificate](#) (KB). Once you have your certificate:

1. Configure the HTTPS binding for your Secret Server website using the certificate you choose.
2. Ensure your certificate is trusted on the Secret Server users' machines. See [Trusting an SSL Certificate on a Client Machine](#) (KB) for instructions.
3. Enable **Force HTTPS/SSL** on the **Security** tab of the Secret Server **Configuration** settings.

Configure backups to avoid losing your data. Secret Server provides the option to automatically take a backup on the interval you specify, sending the backups to a local or network location. There are two components of an entire backup of Secret Server: the Web application files and the database. Find these settings by selecting **Backup** from the **Admin** menu. See [Backup and Disaster Recovery](#) for more information.

To configure the backup paths, see [Backup Configuration File Path Settings](#) (KB).

Note The file paths configured on this page by default need to be either changed or created on each server that the Secret Server application and database reside on.

To allow users to log in with their Active Directory (AD) credentials, you can configure your AD domain settings in Secret Server and then add users either individually or by group.

Setting up Active Directory

See [Configuring Active Directory](#).

Enabling Active Directory Users

See [Enabling and Disabling Active Directory Users](#).

Managing Active Directory Users via a Distributed Engine

See [Syncing and Authenticating AD Users via a Distributed Engine](#).

To try out Secret Server, you must have folders, roles, users, and secrets to operate on:

1. Setup some folders and roles: We encourage is for you to setup a folder structure and a few roles. The folder structure is how you will keep your secrets organized, and provide access to shared secrets. Additionally, roles ensure you are able to control access to different parts of Secret Server and assign permissions to view certain folders and secrets. See [Secret Folders](#) and [Roles](#).
2. Add users if you have not already from AD. See [Creating Users](#) and [Creating User Groups](#).
3. Add an Active Directory or other secrets. If you plan on using discovery, the account will also need permissions to scan computers on the network for accounts. See [Managing Secrets](#).

Secret Server has a discovery feature that can automatically find local Windows accounts, Active Directory service, Unix, VMware ESX/ESXi, and Active Directory domain accounts. Account and dependency types not supported out-of-the-box in Secret Server can still be discovered by writing PowerShell scripts that can be run as custom scanners. This allows administrators to quickly import accounts found by Secret Server on specified domains or IP addresses.

Note: Please see the [Discovery Topic](#) for a comprehensive guide to configuring and using discovery.

To run discovery on a domain, IP address range, or a custom source, you need to first enable the discovery feature for Secret Server. Second, you must enable discovery for each discovery source you would like to be scanned.

Secret Server remote password changing (RPC) provides the ability to either start a password change manually or schedule automatic password changes to occur at a regular interval.

Enabling Remote Password Changing

See [Enabling RPC](#).

Performing a Manual RPC

See [Run a Manual RPC](#).

Common RPC Error Codes

See [RPC Error Codes](#).

Heartbeat allows you to determine from Secret Server whether the credentials in a secret authenticate successfully with their target system. By default, heartbeat is turned off in Secret Server. See [Heartbeats: Automatically Testing Secret Credentials](#) for general information.

Enabling Heartbeat

See [Enabling Heartbeat in RPC](#).

Running Heartbeat

See [Running Heartbeat for a Secret](#).

Before running reports and audits, you must create something to report on—to that end:

- Import a few accounts or create secrets manually
- Rotate passwords a few times
- View a couple of your secrets

This generates enough audit logs to provide meaningful outputs in your reports:

- Security Hardening Report
- What secrets have been accessed
- What secrets failed heartbeat
- Failed login attempts
- Secret activity

See [List of Built-In Reports](#) for the most up-to-date list of reports included.

For details on using reports, see:

- [Creating and Editing Reports](#)
- [Viewing Reports](#)

Sometimes, depending on your scenario, you want to add extra protections to highly sensitive secrets. Secret Server has a access request and workflow features:

- [Secret Check-Outs](#): Grant access to a single user
- [Basic Secret-Access Requests](#): Require approval prior to accessing a secret for a defined time period
- [Advanced Secret-Access Requests with Workflow Templates](#): Require multi-level and multi-user approval prior to accessing a secret for a defined time period
- [Secret DoubleLocks](#): Add another security layer by encrypting secret data with a supplemental custom encryption key that is only accessible with an additional password, regardless of regular permissions.

A secret *launcher* opens a connection to the remote computer or device or logs into a website using the secret's credentials directly from the Web page. While this provides a convenient method of opening RDP and PuTTY connections, it also circumvents users being required to know their passwords. A user can still gain access to a needed machine, but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser. SS launchers, also called protocol handlers, come in three primary types:

- **Remote Desktop:** Launches a Windows Remote Desktop session and automatically authenticates the user to the machine.
- **PuTTY:** Opens a PuTTY session and authenticates the user to a Unix system.
- **Web Password Filler:** Uses a Chrome extension to automatically log the user into a website with secret credentials. See our separate documentation for Web Password Filler.
- **Web Launcher:** An alternative method to automatically log on websites. See [Web Launcher](#).

See [Secret Launchers](#) for more information.

Session recording provides an additional level of security by recording a user's actions after a launcher is used. Session recording works for any launcher, including PuTTY and SSH, Windows Remote Desktop, Microsoft SQL Management Studio, and custom executables. The resulting movie is viewable from the secret audit. There are two types of session recording:

- [Basic Session Recording](#)
- [Advanced Session Recording](#)

You can access Secret Server without using the user interface for automation and integration purposes. Currently, there are two APIs:

- An asynchronous REST (representational state transfer) API for Web services, which is based on JSON (JavaScript Object Notation). This is the preferred method. It is faster and easier to read than the SOAP API and is still actively updated.
- A synchronous SOAP (Simple Object Access Protocol) for Web services, which is based on XML. This method is deprecated, but we still support it. It is based on an older technology, which has largely been replaced in recent years. There will be no enhancements to this API. There are, however, a few, rarely used capabilities that only our SOAP API has.

We offer a software development kit (SDK) that contains a .NET framework and a command line interface (CLI) for accessing the REST API with Windows applications or scripting languages.

Both APIs, the .NET framework, and the CLI support:

- GET Requests: Retrieve information from Secret Server, including entire secrets, individual secret fields, and security tokens
- POST Requests: Create Secret Server data
- PUT Requests: Update Secret Server data
- DELETE Requests: Remove Secret Server data
- Once-per-session permissions (tested once and then based on the IP address), administered with a Secret Server rule

SDK Documentation:

- [Secret Server SDK Guide](#): Includes these topics:
 - Secret Server configuration
 - Roles and permissions
 - SDK client installation
 - Connecting to Secret Server
 - SDK client caching
 - Examples
- [Secret Server SDK Downloads](#): Includes these topics:
 - SDK downloads
 - Download
 - SDK release notes
 - NuGet packages
- [SDK Integration Document](#): Includes these topics:
 - Integrating using C#
 - Integrating using the web.config file
 - Methods of the SecretServerClient() class

REST API Documentation:

- [REST Web Services API - Secret Server](#): Links to online reference guides (by Secret Server release)
- [REST API PowerShell Scripts - Getting Started](#)
- [REST API Perl Examples](#)
- [REST API Java Examples](#): Downloadable Zip file

SOAP API Documentation:

- [SOAP Web Services API - Secret Server](#): Reference guide in a downloadable PDF
- [SOAP-based Web services API - Getting Started](#)

You have finished this "Getting Started" introduction to Secret Server. There is much more to explore within Secret Server, such as scripting, third-party Integrations (SIEM, CRM, HSM, and more), and connecting to Privilege Manager to monitor and protect endpoints. We look forward to working with you!

See [Additional Resources](#) to learn more about Secret Server and other Delinea products.

Help

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Capitalization

Technical writing is typically so awash in capitalization that it often denotes nothing and harms legibility. To counter that, in general, this document follows the IBM Style Guide rule:

"Do not capitalize the names of features and components unless they are sold separately or are trademarked."

More specifically, the only things capitalized in this document are:

- Company, person, country, geographic place, or organization names
- Official or trademarked products or services, unless they officially have atypical capitalization, for instance *iPod*.
- Acronyms and initializations
- When referring to any UI labels that are capitalized
- When the word begins a sentence or phrase

Code and Command Line Text

Variable text in literal typed-in text and command-line parameters follow these industry-wide standards:

- All code and command-line interface text appears in monospaced text.
- Required parameters appear in angle brackets: `ping <hostname>`
- Optional parameters appear in square brackets: `mkdir [-p] <dirname>`
- Repeated parameters are followed by ellipses: `cp <source1> [source2...] <dest>`
- Multiple choice items are separated by vertical bars and grouped by curly brackets: `netstat {-tl-u}`

Keyboard Shortcuts

- Keyboard keys are bolded and surrounded with square brackets: **[Enter]**
- Concurrent key presses are denoted with plus signs: **[Ctrl]+[Alt]+[Del]**
- Sequential key presses are denoted by commas: **[Page Down], [Enter]**

Notes

There are three types of notes: *regular*, *important* and *warning*.

Note: Regular notes have a title, either "Note" or something custom, which appears as a phrase followed by a colon at the beginning of the note. A note contains tangential (an aside) or supplemental information (a tip or clarification).

Important: Important notes contain substantive information that should be heeded, or negative consequences can occur, involving frustration, wasted time, or minor data loss.

Warning: Warning notes contain substantive information that should be heeded, or negative consequences can occur, involving injury, major data loss, or equipment damage.

Other Special Text

- Email addresses and URLs are usually denoted by a colored underline: support@thycotic.com.
- When URLs are part of the instruction, as opposed to clickable link, they appear in monospaced text: Type `https://www.somewhere.com` OR click <https://www.somewhere.com>.
- Cross-references to headings are hyperlinks: See [\[Booting a Server\]](#)[].
- Document or article names (not sections) appear in italics: See the *Server Administration Guide*. They may or may not be hyperlinks.
- All file and folder paths appear in monospaced text: `app\bin\web_config.xml`

- File names by themselves do *not* appear in monospaced text: `web_config.xml`. If the file name contains spaces, the name is surrounded by quotation marks: "web config.xml".

Note: Ending punctuation may be omitted for clarity when following typed-in text, including URLs.

Screen Components and Attentional Targets

- Mouse-click, keyboard, and other attentional targets (anything a looks for) are denoted by bold type: **OK** button or **Login** link.
- Attentional Targets and screen component names in system *responses* are not bolded: "The OK button appears" verses "Click the **OK** button."
- Names of screen components, such as tabs, buttons, and text boxes, are corrected for spelling and capitalization. The component type appears in lowercase. Example: **SEARVER CONFIGURATION** window becomes **Server Configuration** window.

Table: Terms and Definitions

2FA	<i>Two-Factor Authentication</i>
AD	<i>Active Directory</i>
Administrator	<i>Administrator</i> is a default role that comes preconfigured with Secret Server. Roles control access to features within Secret Server. This role can be customized to have different permissions. In this guide, administrator (lowercase) is used when referring to users who manage the system and have control over global security and configuration settings. Note that administrators in Secret Server do not automatically have access to all data stored in the system—access to data is still controlled by explicit permissions on that data.
AES	<i>Advanced Encryption Standard</i>
All Secrets	<i>All Secrets</i> is a master table of the secrets stored on Secret Server. It is a one-stop, searchable location for examining the status and properties of secrets. It is a supplement to, not a replacement for, the secret folder tree. It lists and you can sort by secret template, heartbeat status, sync status, machine, access date, username, and much more. You can customize which characteristics are displayed.
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ASP	<i>Advanced Server Pages</i>
AWS	<i>Amazon Web Services</i>
CAC	<i>Common Access Card</i>
CEF	<i>Common Event Format</i>
CHG	<i>Change</i>
CIDR	<i>Classless Inter-Domain Routing</i>
CRM	<i>Customer Relationship Management</i>
CSV	<i>Comma-Separated Values</i>
DBA	<i>Database Administrator</i>
DE	<i>Distributed Engine (Secret Server)</i>
DES	<i>Data Encryption Standard</i>
Directory Services	<i>Directory services</i> are components of network operating systems that map the names of network resources to their network addresses. Their shared information infrastructure locates, manages, and organizes network resources, which can include volumes, folders, files, users, groups, devices, and much more. Active Directory is Secret Server's native

	directory service.
Discovery	<i>Discovery</i> is the process where Secret Server scans an environment to find accounts and associated resources called dependencies. Once accounts are found, you can use them to create associated new secrets in Secret Server. Users with the "administer discovery" role permission can either manually import accounts or can create an automated process, called a discovery rule, to do so. Using discovery does not stop users from manually creating their own secrets.
Distributed Engine	For smaller enterprises, Secret Server performs all functions on the Web server it is installed on. Secret Server is also scalable for large enterprises and scenarios demanding higher performance. We use remote <i>distributed engines</i> to accomplish this. You route high-demand processing and traffic operations through one or more of these to enhance Secret Server's capacity. For example, distributed engines can synchronize and authenticate for Active Directory. They can also perform remote password changing, heartbeat, discovery and more, all controlled by a single Secret Server installation.
DPAPI	<i>Data Protection Application Programming Interface</i>
DSS	<i>Data Security Standard</i>
EC2	<i>Elastic Compute Cloud</i>
ESX	<i>Elastic Sky X</i>
FIPS	<i>Federal Information Processing Standard</i>
FQDN	<i>Fully Qualified Domain Name</i>
GDPR	<i>General Data Protection Regulation</i>
Group	You can manage users with user <i>groups</i> . Users belonging to a group receive roles (and by extension permissions) attributed to that group. This simplifies the management of the permissions and roles assigned to users. Additionally, groups can be synchronized with Active Directory to further simplify management. A user can belong to multiple groups.
HSM	<i>Hardware Security Module</i>
HSTS	<i>HTTP Strict Transport Security</i>
IAM	<i>Identity and Access Management</i>
IIS	<i>Internet Information Services</i>
IP	<i>Internet Protocol</i>
ITSM	<i>Information Technology Service Management</i>
KB	<i>Kilobyte or Knowledge Base</i>
KBA	<i>Knowledge Base Article</i>
LDAP	<i>Lightweight Directory Access Protocol</i>

NAT	<i>Network Address Translation</i>
NATO	<i>North Atlantic Treaty Organization</i>
NIST	<i>National Institute of Standards and Technology</i>
NSA	<i>National Security Agency</i>
NTLM	<i>NT LAN Manager</i>
OATH	<i>Open Authentication</i>
OS	<i>Operating System</i>
OTP	<i>One-Time Password</i>
OU	<i>Organizational Unit</i>
PBA	<i>Privileged Behavior Analytics</i>
PCI	<i>Payment Card Industry</i>
PDF	<i>Portable Document Format</i>
PII	<i>Personally Identifiable Information</i>
PIV	<i>Personal Identity Verification</i>
PuTTY	<i>Popular SSH and Telnet Client</i>
QR	<i>Quick Response (code)</i>
RADIUS	<i>Remote Authentication Dial-In User Service</i>
RBAC	<i>Role-Based Access Control</i>
RBS	<i>Role-Based-Security</i>
RD	<i>Remote Desktop</i>
RDP	<i>Remote Desktop Protocol</i>
Remote Password Changing	Secret Server can automatically change passwords on remote devices and various platforms, including the following: Windows accounts, database logins, Active Directory accounts, Unix and Unix-like accounts (including root passwords), network appliances or devices and more.
REST	<i>Representational State Transfer</i>
Role	Every user and group must be assigned to a <i>role</i> . Secret Server uses role-based access control to provide very granular system access. Secret Server ships with three roles: Administrator, User, and Read-Only User. Each role contains a set of

	permissions to match the job function of users with that role. See the Secret Server Role Permissions List for details.
Role-based Security	Secret Server uses role-based access control, which provides the ability to set strict, granular permissions for each user. All features in Secret Server are available to users based on permissions, which collectively make up roles.
RPC	See <i>Remote Password Changing</i>
SAML	<i>Security Assertions Markup Language</i>
SEC	<i>Security and Exchange Commission</i>
Secret	A piece of information that is stored and managed within Secret Server is referred to as a secret. Secrets are derived from secret templates. Typical secrets include, but are not limited to, privileged passwords on routers, servers, applications, and devices. Files can also be stored in secrets, allowing for storage of private key files, SSL certificates, license keys, network documentation, Microsoft Word or Excel documents and more.
Secret Template	Secret templates are used to create secrets and allow customization of the format and content of secrets to meet company needs and standards. Examples include: local administrator account, SQL Server account, Oracle account, credit card and Web password. Templates can contain passwords, usernames, notes, uploaded files, and drop-down list values. New secret templates can be created, and all existing templates can be modified.
SHA1	<i>Secure Hashing Algorithm 1</i>
SIEM	<i>Security Information Event Management</i>
SMS	<i>Short Message Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SP	<i>Service Pack</i>
SQL	<i>Structured Query Language</i>
Secret Server	<i>Secret Server</i>
SSH	<i>Secure SHell</i>
SSL	<i>Secure Socket Layer</i>
TCP	<i>Transmission Control Protocol</i>
TOTP	<i>Time-Based One-Time Password</i>
UDP	<i>User Datagram Protocol</i>
UI	<i>User Interface</i>
UNC	<i>Universal Naming Convention</i>

Unlimited Administration Mode	An emergency, break-the-glass mode that gives administrators access to all content within the system, regardless of explicit permissions. Access to unlimited administration mode is controlled using role permissions.
URL	<i>Uniform Resource Locator</i>
User	<i>Users are Secret Server's representation of people—one person per user. Each user has a unique username, as well as other attributes. Users are assigned to groups, and roles are assigned to them, either directly or via groups.</i>
VM	<i>Virtual Machine</i>
VPN	<i>Virtual Private Network</i>
WS	<i>Web Services</i>
XML	<i>eXtensible Markup Language</i>

Forums

[Forums](#). Forums are oriented toward admins and other technical users.

Delinea Blog

[Delinea Blog](#)

To have access to Delinea Technical Support, you must have an equal number of unexpired user and support licenses. All support licenses expire 365 days after they are issued.

Technical Support Coverage

Please see our [Getting Technical Support](#) section below.

Note: Please see the **Support** link on the menu above for details about our support policies.

Accessing Upgrades

Supported customers have access to all new releases (both minor and major). See [Secret Server Installation and Upgrade Guides](#).

Requesting New Features

We encourage customers with active support licensing to participate on feedback.thycotic.com where you can discuss and vote on new features.

Getting Technical Support

Important: Please see the **Support** link on the menu above for details about our support policies.

Step One: Gather Information You May Need

Before you contact Support, gather the following information:

- Your Delinea Support username and password
- The email account already associated with your account (if using email)
- Your company name
- The technical contact name
- The technical contact phone number
- The product name
- Issue symptoms and details
- Any other relevant details, such as hours the technical contact is present

Step Two: Get a Mandatory Support PIN

Secret Server

The support PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for Delinea Support to locate your customer records and give you better support.

To get your PIN:

1. Get the log on the credentials you received when you became a Delinea customer.
2. Log on the [Support Portal](#) using your credentials.
3. On the main page, click the large blue **PIN** bar to get your PIN. The PIN appears on the button.
4. Record your PIN.
5. If you want to use our ticketing system for support, leave the browser tab open, and return for step four.

Secret Server Cloud

The support PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for Delinea Support to locate your customer records and give you better support. In addition, there is an additional "privileged PIN" for accessing your cloud instance.

To get your PIN:

1. Log on the Cloud Manager Dashboard at <https://portal.thycotic.com>.
2. Click the **Generate Tech Support** PIN button. A Tech Support PIN popup appears.
3. Record your PIN.
4. Click the Generate Privilege PIN button. Another Tech Support PIN popup appears.
5. Record your privileged PIN. Note that privileged PINs begin with "p"

Important: Providing us a privileged PIN gives Delinea Technical Support write access to your cloud database for one day. Secrets and other sensitive data remain encrypted and unreadable.

Step Three: Choose a Support Method

Delinea customers have access to support by phone, email, and our support ticketing system (best for issue tracking). In all cases, **you must first obtain a support PIN**.

Important: For Severity 1 issues you **must** use phone support. Otherwise, use the method you prefer. Severity 1 means a critical problem that has caused *complete loss of service* and work cannot reasonably continue at your worksite.

Step Four: Contact Support

Click the **Support** link on the menu above for Support email addresses and phone numbers.

Using one of the below methods, contact Delinea Support.

Phone Support

Delinea delivers support by phone worldwide. Select the applicable number from the list on the **Support** link in the menu above.

Email Support

Send your email to support@thycotic.com **with the PIN number as part of the subject line** of your email. For example: PIN 345 Workflow Stopped Unexpectedly. Include all the information listed in step one.

Important: You must send your email using an email address already noted in your account with Delinea. Otherwise, it might delay our response.

Ticketing System Support

Open a support ticket and track your issue to resolution.

- Visit the [Support Portal Login Page](#) using the credentials you received when you became a customer.
- After logging on, click the **Cases** tab, and then click **Create a Case**.
- Follow the instructions to complete your case.

Access Requests

The access request feature allows a secret to require approval prior to accessing the secret. Note the following:

- Establishing a workflow model, the user must request access from the approval group or groups.
- An email is sent to everyone in the approval groups, notifying them of the request.
- The request can be approved or denied by any members of the approval groups.
- Access is granted for a set time period.
- If **Owners and Approvers also Require Approval** is enabled, then even owners or those in an approval group needs to request access.

Once a request for access to a secret has been made, approvers receive an email.

The email contains one link to the secret **Access Request Approval** page for that request in Secret Server, and five additional links to approve or deny the request if the **Allow Approval for Access from Email** configuration setting is enabled.

The approver can either click one of the links contained in the email or navigate to the **Notification Center** in the user menu within Secret Server.

Alert Notification Center

FILTER

Notification Type

- Event Subscription
- Secret Access Requests
- Application Access Requests
- System Alerts
- Include Archived

Priority

- Requires Interaction
- Critical
- Informational

PRIORITY	NAME	DESCRIPTION
	i	Pending Engine

If choosing the latter, in the displayed grid click the access request name. This takes you to the secret's Access Request Approval page.

From here, you can accept or deny the request as well as set an expiration date.

The requestor has access to the secret until the specified date.

Selecting the current date is the smallest window of time allowed and grants access to the end of the day.

With **Allow Approval for Access from Email** enabled, clicking one of the five additional links in the email allows access for 1, 2, 4, or 8 hours or deny the request, per the link description in the email.

Note: The expiration date referred to in approval requests is **not** the same as secret expiration.

When implementing the "Requires Approval For Access" feature to force a user to request access to a Secret and "Allow Approval For Access from Email" is turned on, an email with approval links is sent out to all the approvers. To customize this email, follow the steps below:

1. Find the Secret Server directory, which is typically C:\inetpub\wwwroot\SecretServer
2. Under the customresources directory:
 - If it does not already exist, create a directory named by a digit corresponding to your language Locale ID (LCID). For English, use the digit 9.
 - If it does not already exist, create a pages.xml file.
 - Insert the code below into pages.xml The hours value can be any integer greater than 0:

```
<resources>
<page name="legacyhome.aspx">
<control controlId="SearchTermTextBox">
<property name="ToolTip" value="This is a customer set resource." />
</control>
</page>
<messages>

<message name="SecretAccessRequestEmailSubject">Application : {0}, Secret : {1}, Requestor : {2}</message>
<message name="SecretAccessRequestEmailMessage">Requestor : {0}{2}Secret : {1}{2}Beginning : {6}{2}Ending : {7}{2}Reason : {3}{2}Link : {4}{2}Ticket Number :
{5}{2}Secret Field : #${DOMAIN#}</message>

<message name="SecretAccessRequestEmailMessageApprovalPart">
<![CDATA[
Approve for 3 hours from start date: #URL#RequestApproval.aspx?op=approve&id=#UNIQUEID#&hours=3

Approve for 24 hours from start date: #URL#RequestApproval.aspx?op=approve&id=#UNIQUEID#&hours=24

Deny Request: #URL#RequestApproval.aspx?op=deny&id=#UNIQUEID#
]]>
</message>
<message name="SecretAccessRequestEmailMessageTicketReferencePart">{0}{1}</message>
</messages>
</resources>
```

#URL# = the URL of your Secret Server instance

#UNIQUEID# = the approver's unique ID

hours = X is the number of hours approval will apply

##\$SECRETFIELD# = the name of a Secret Field

For example #MACHINE# will be replaced with the value from the machine field on the Secret. If no field with that name is present, it will be blanked out. Please note the location of the field name in the xml. The fields will not resolve in the CDATA section.

3. Perform IIS reset

Users can now approve secret access requests and workflows using Duo push notifications. The push notification includes information, displayed on the user's screen, that helps the approver make the access decision.

Prerequisites

To use Duo push notifications:

- Duo must be set up for Secret Server. See [Duo Security Authentication](#).
- Duo user must be set up for Duo two-factor authentication. See [Setting up Duo \(User\)](#).
- The permission "Approve via DUO" must be granted to a role that is assigned to a group that includes all who will be approving requests via Duo. This allows enough flexibility so that those not wanting Duo push approvals can be configured to not receive them.

Assigning the Duo Approval Permission

To associate the permission with users:

1. Go to **Admin > Roles**.
2. Click the **Create New** button to create a new role. Name it "Duo Push Approver" or another name of your choosing.
3. Assign the **Approve Via DUO Push** permission to the new role.
4. Click the **Save** button.
5. If you choose to create a separate group for approvers, do this by navigating to **Admin > Groups**.
6. Click the **Create New** button to create a new group.
7. Add the desired users (chosen approvers) to that group.

Note: You can also assign users to the group later. This method is a shortcut when creating a group.

8. Click the **Save** button.
9. Go to **Admin > Roles**.
10. Click the **Assign Roles** button. The View Role Assignment page appears.
11. Click the **Role** dropdown list to select the role you created. Note that there are no groups or users.
12. Click the **Edit** button. The Role Assignment page appears.
13. Assign the **Approve via DUO Push** role to the **Assigned** list box.
14. Click the **Save Changes** button. Setup is now complete.

Note: In addition to having the role you created, the user must be properly set up to receive Duo push notifications. See [Setting up Duo \(User\)](#).

Note: Any notifications will all be sent out at the same time, and the first response (approve or deny) will be the determinant response. A non-response will not result in either an approve or deny response.

To start the request process for access to a secret, the user must simply attempt to view the secret. The user is then sent to the Request page. In there, the user can explain the reason for the request and then click **Request Access** to submit the request.

If a member of the Approval Group either approves or denies the request (see below for details), the requestor is sent an email with the details. If approved, the requestor can access the secret via the link contained in the email.

To enable Access Request for a secret, navigate to the **Secret View** page for the secret:

1. Go into the **Security** tab and click the **Edit** button.
2. Check the **Enable Requires Approval for Access** checkbox to enable the setting.
3. Once enabled, select users or groups as approvers for the secret. Unless the **Owners and Approvers also Require Approval** option is turned on, owners or users that are members of the Approvers group do not need to request access to view the secret.

Note: Users need at least view access to the secret to be able to access the secret even with **Access Request** enabled. If the users do not have view permission they are unable to find the secret with search or browse.

Note: The email configuration settings need setting up, including valid email addresses, for the users in the approval group for emailing to work.

Secret Server Administration

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server is highly customizable. Administrators can increase site security through various configuration settings such as force inactivity timeouts and specifying a SMTP server. This level of configuration allows Secret Server to be altered to meet the needed requirements for the instance. The settings are explained in this section.

The administration side panel and page is a control panel for administering Secret Server. You access it by clicking or hovering over the **Administration** main menu item and clicking the >> icon. Clicking the << icon closes the side panel.

Note: The most commonly sought items appear in the same list, so you can go directly to them without having to go to the Admin page.

Figure: Administration Page (alphabetized view with side panel closed)

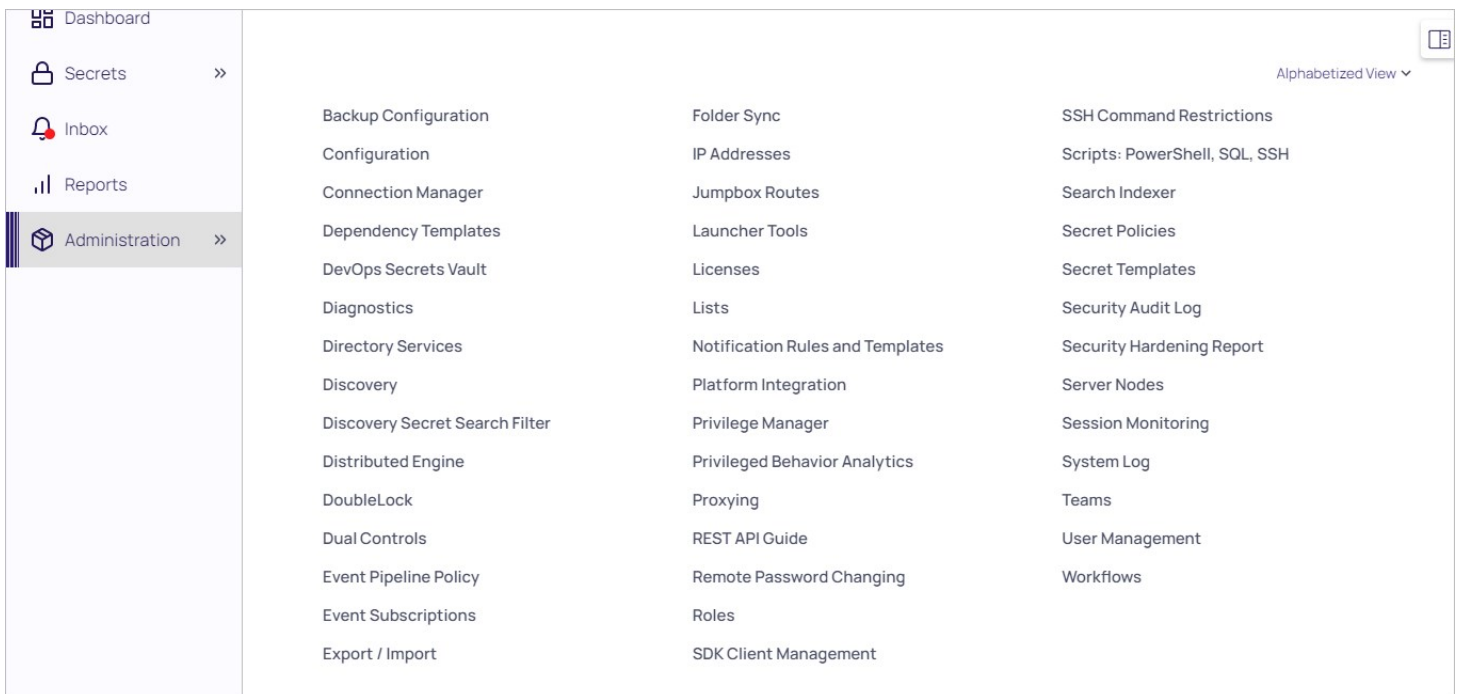
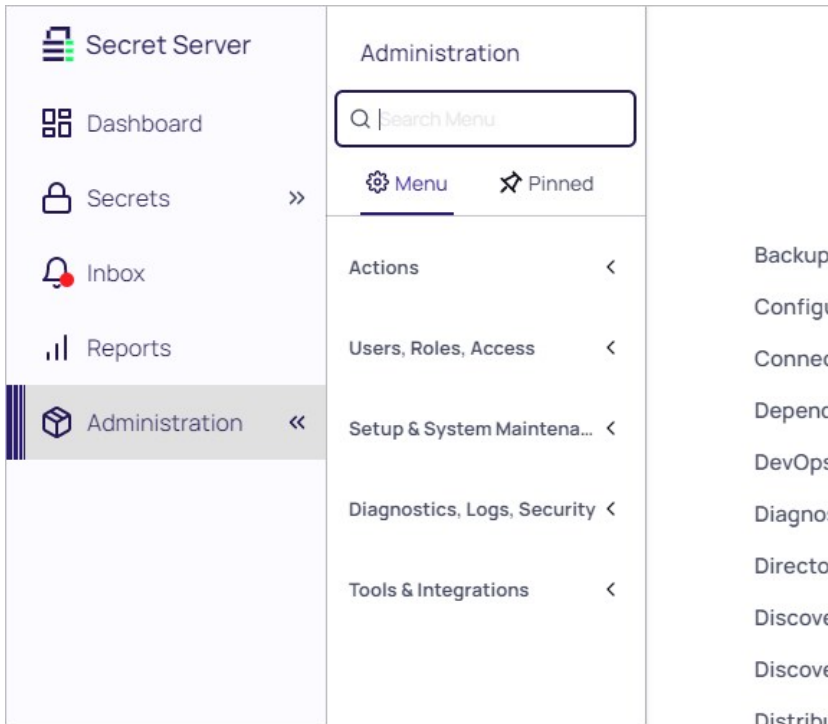


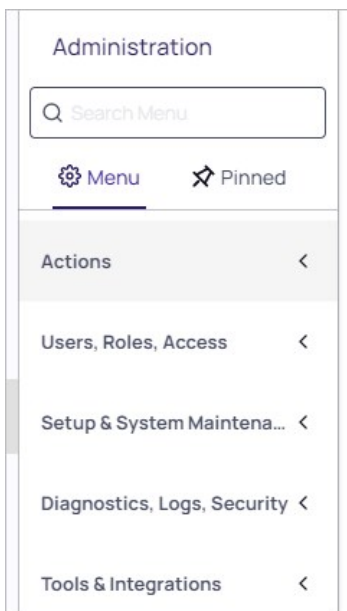
Figure: Administration Page Side Panel



With the page and side panel, you can quickly and easily find administration controls in several ways:

Administration Side Panel

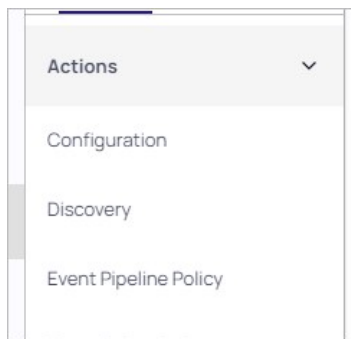
Figure: Administration Side Panel



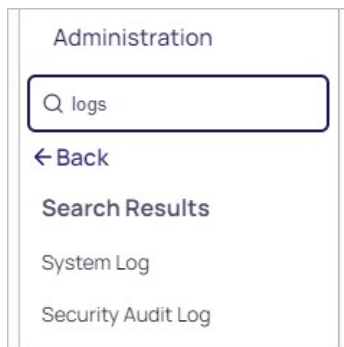
Navigation

Click the double chevron icon to view the side panel. You can find admin controls in two ways:

- **Menu Navigation:** Click the admin menu to drill down to a topic of your choice. As you click each top-level choice, subtopics appear. Once you see the item you desire, click and you are brought to that page.



- **Dynamic Search:** You can search for a concept, configuration, or component by typing a search term in the search text box on the side panel. The text box automatically suggests items as you type. Once you see the item you desire, click and you are brought to that page.



Pinning Commonly Used Items

You can build your own admin panel on the Pinned tab by hovering the mouse pointer over a side panel item and clicking the pin icon. Pinned items remain on the Pinned tab until you remove them.

Administration Page

The administration page is an alternative to the side panel. You have three views to choose from, which you set by clicking the unlabeled view dropdown list. The link text states the current view. The views are:

- **Simple View:** Hover over or click the large panel buttons to see a menu of topics.

Note: The legacy search feature is essentially the same as that in the administration side panel.

What are you looking for?

Search for an admin option



Simple View 



Actions

Features that perform important jobs



Setup & System Maintenance

Setup your system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find tools and other product integrations here

- **Alphabetized View:** A text list of the available items:

Alphabetized View ▾

Backup Configuration	Platform Integration
Configuration	Privilege Manager
Connection Manager	Privileged Behavior Analytics
Dependency Templates	Proxying
DevOps Secrets Vault	REST API Guide
Diagnostics	Remote Password Changing
Directory Services	Roles
Discovery	SDK Client Management
Discovery Secret Search Filter	SSH Command Restrictions
Distributed Engine	Scripts: PowerShell, SQL, SSH
DoubleLock	Search Indexer
Dual Controls	Secret Policies
Event Pipeline Policy	Secret Templates
Event Subscriptions	Security Audit Log
Export / Import	Security Hardening Report
Folder Sync	Server Nodes
IP Addresses	Session Monitoring
Jumpbox Routes	System Log
Launcher Tools	Teams
Licenses	User Management
Lists	Workflows
Notification Rules and Templates	

- **Categorized View:** A text list of the available items bunched by category:

Categorized View ▾

CORE ACTIONS

Configuration

Discovery

Dependency Templates

Discovery Secret Search Filter

Event Pipeline Policy

Event Subscriptions

Notification Rules and Templates

Lists

Remote Password Changing

Session Monitoring

Secret Policies

Secret Templates

IMPORT, EXPORT, SYNC

Export / Import

Folder Sync

NETWORKING

Distributed Engine

Server Nodes

USERS, ROLES, ACCESS MANAGEMENT

Directory Services

User Management

Roles

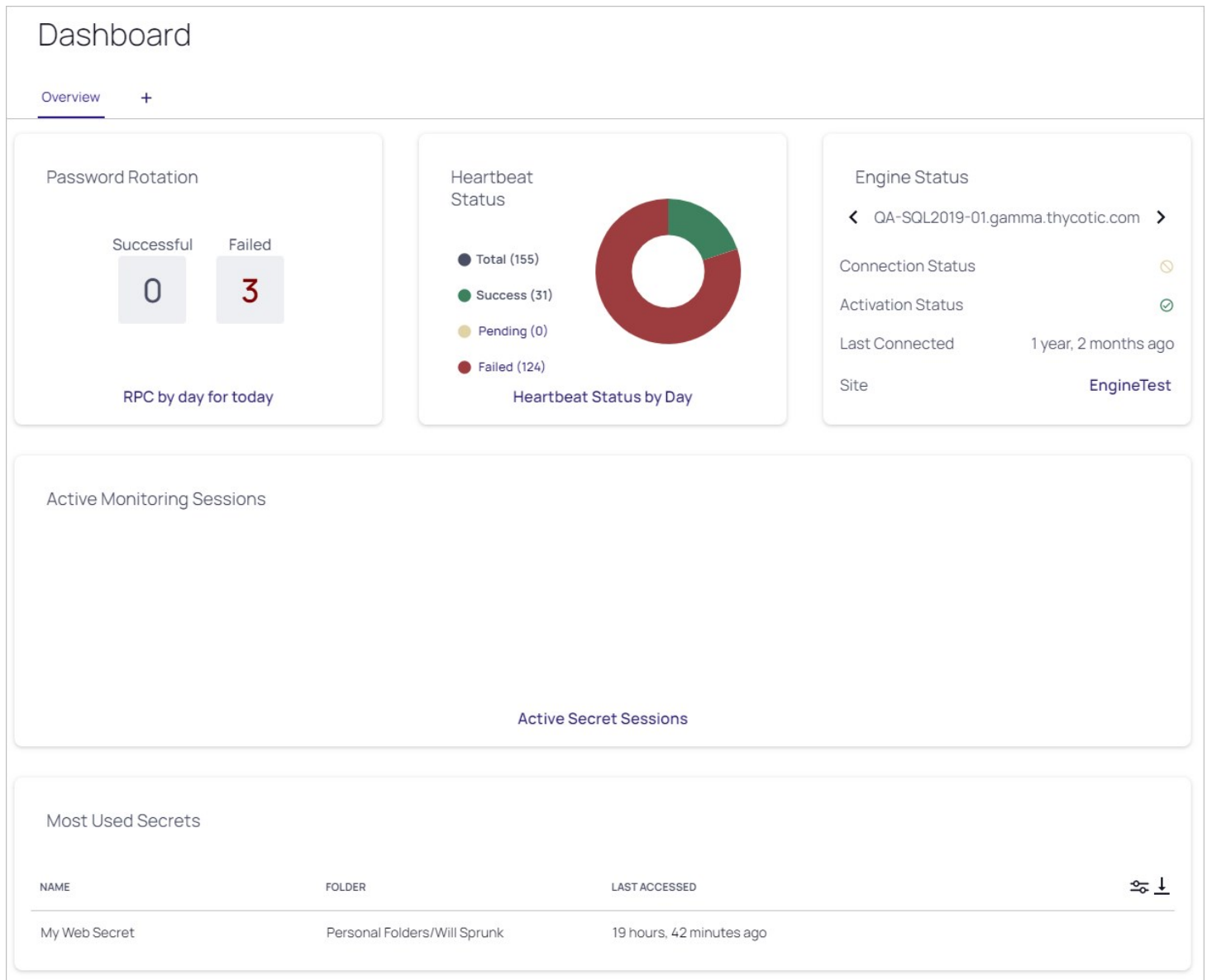
Teams

IP Addresses

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The Secret Server dashboard is the main page for searching and viewing secrets. You access it by clicking the Dashboard menu item.

Figure: Secret Server Dashboard



Dashboard Components

Overview Tab

The Overview tab provides several fixed widgets for getting a quick understanding of your Secret Server installation:

- **Active Monitoring Sessions:** Your current monitored sessions. See [Session Recording](#).
- **Approvals:** Your current in-process approvals. See [Secret Access Requests](#).
- **Engine Status:** A scrollable list of distributed engine connection and activation statuses.
- **Heartbeat Status:** A graphic of the current status of your heartbeats: success, pending, or failed. When you click on one of the statuses, you are brought to a report page for that status. For example, **Reports > Secrets Failing Heartbeat**. When you click the **Current** link, you are brought to the **Reports > Heartbeat Status by Day** page. See [Secret Heartbeats](#).
- **Most Used Secrets:** A table of the most recently accessed secrets, listed by date and folder.
- **Password Rotation:** The state of your current password rotations. When you click the **Today** link you are brought to the **Reports > RPC by Day** report page. See [Remote Password Changing](#).

Note: To see an overview of incoming system and subscription alerts, see the [Alert Notification Center \(Inbox\)](#).

Customized Tabs

Tab Management

The following operations are available for creating custom tabs:

- **Create:** Click the **+** to the right of the tabs to create a new empty tab.
- **Delete:** Click the **▼** icon on a tab and select **Delete** to delete a tab. You can cancel changes by clicking the **Cancel** button. A confirmation pop up page appears.
- **Rename:** Click the **▼** icon on a tab and select **Rename** to change the tab name. You can cancel changes by clicking the **Cancel** button.
- **Sort:** Click and drag a tab to the left or right of an existing tab.

Dashboard Widgets

By default, the Overview tab contains the Favorite Secrets, Recent Secrets, and + Add Widget widgets (function boxes). You can add these widgets on custom tabs:

- Expired Secrets
- Out-of-Sync Secrets
- Reports
- Request Management

Widget Types

Table: Dashboard Widgets

Expired Secrets	Displays expired secrets.
Favorite Secrets	Displays secrets marked as favorites. Also appears on the Secret Navigation Slideout.

Out-of-Sync Secrets	Displays secrets that are out-of-sync—the heartbeat or RPC have failed.
Recent Secrets	Displays the secrets viewed most recently. Also appears on the Secret Navigation Slideout.
Report	Displays a report. Click the Report Category list to select a report from the drop-down menu. One report can be displayed per widget. Click the title of the report to navigate to the Report View page.
Request Management	Displays any requests pending for the logged in user.
+ Add Widget	Default widget that appears on an empty custom tab. When clicked, adds a widget that is not currently displayed to the Dashboard. This widget's function is duplicated automatically when you add a new Dashboard tab. You cannot remove this widget.

Note: The Search and Browse widgets cannot be rearranged. They always remain in the top left region of the tab.

Managing Widgets

The following operations are available (by clicking the  icon) for managing widgets:

- **Delete:** Delete the widget from this custom tab.
- **Refresh:** Update the information in the widget. This is not available for all widgets.

Dashboard Tools and Help Menu

The Dashboard Tools Menu is available via the  button on the Dashboard. It includes links to:

Tool Section

- Connection Manager
- Importing Secrets
- Exporting Secrets
- Manage Secret Access Request
- Launcher Tools
- Privilege Manager
- Privilege Behavior Analytics

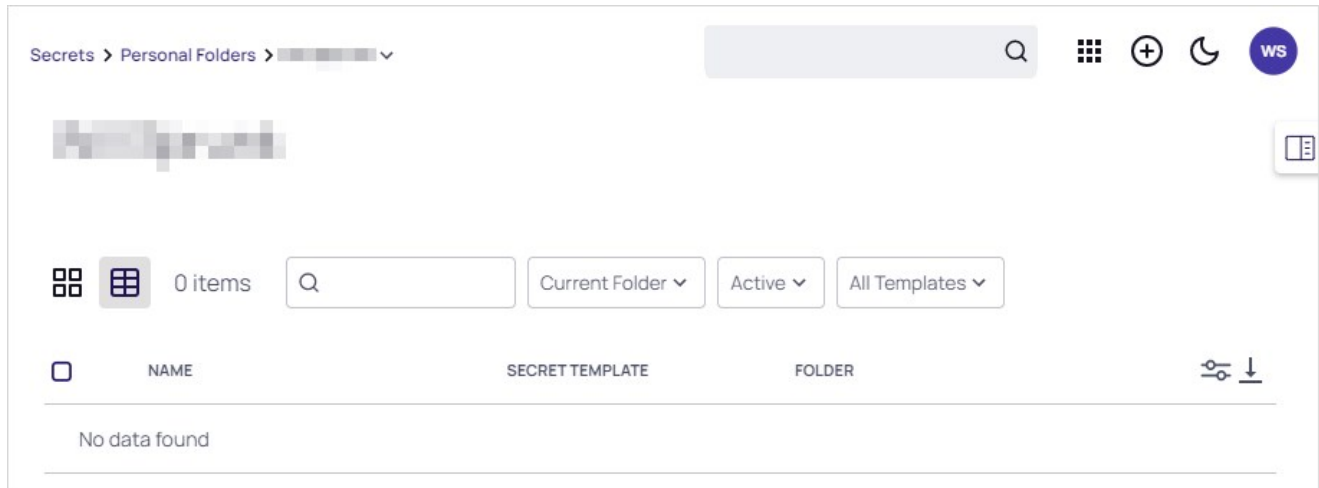
Help Section

- About
- Help
- Secret Server REST API Guide
- User Guide

Running Dashboard Bulk Operations

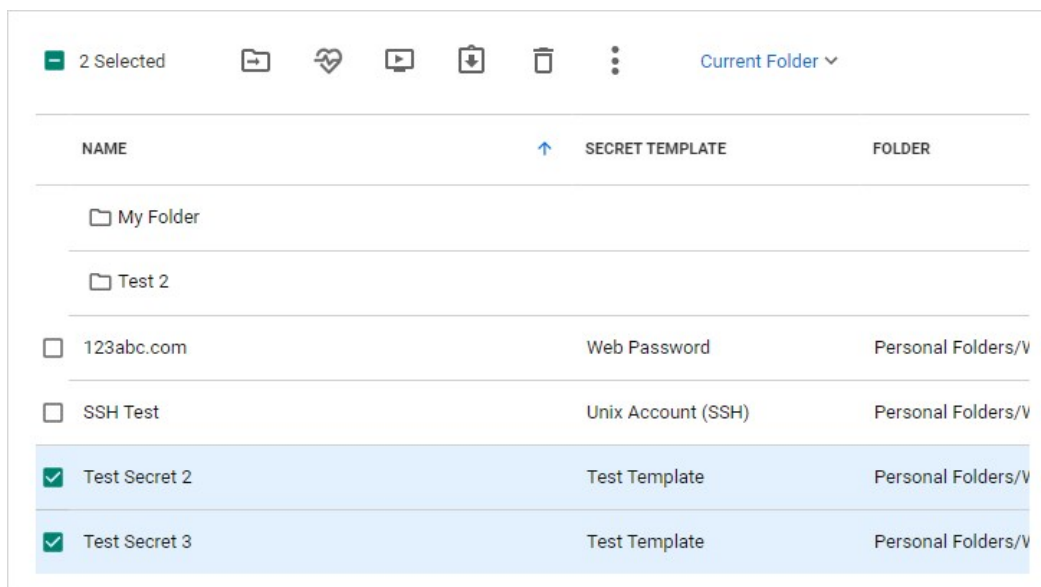
You can perform bulk operations from the Dashboard on multiple secrets:

1. Navigate to the folder containing the secrets you wish to perform a bulk operation on:

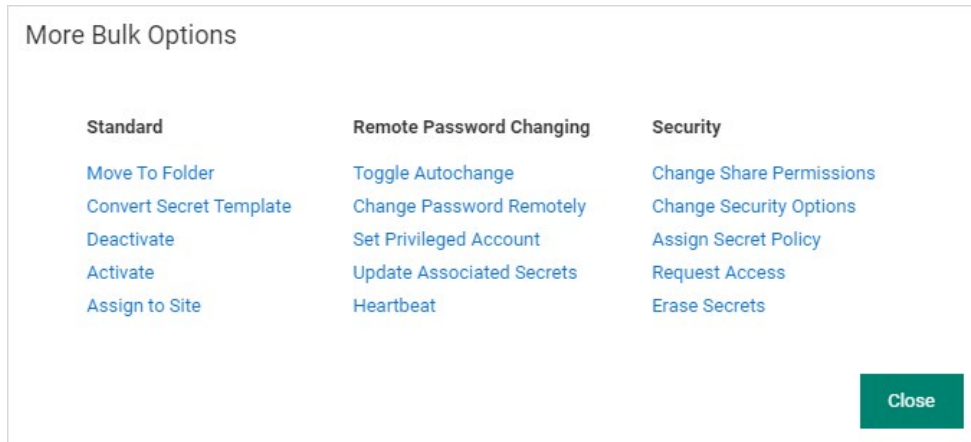


Note: You can also run a bulk operation on the All Secrets page.

2. Click the **Current Folder** link if you want to display the secrets in subfolders too. If so, the link changes to Include Subfolders.
3. Click to select the secrets you wish to include. To check them all, check the check box in the column header row. The bulk operations toolbar appears at the top:



The most common operations have icons. Hover over each to see a text label. The sideways ellipsis (three stacked dots) icon opens a text menu of all the bulk operations:



Note: Many of the available operations (below) are accessed via menus that appear when you click the links. For example, to disable check out, you click Change Security Options.

4. Click to select the bulk operation via either an icon or the **More Bulk Operations** menu . Available bulk operations include:

- Add share
- Assign secret policy
- Assign to site
- Change password remotely
- Change to inherit permissions
- Convert secret template
- Deactivate (was "Delete" in earlier versions)
- Disable autochange
- Disable check out
- Disable comment on view
- Disable heartbeat
- Edit share
- Enable autochange
- Enable check out
- Enable comment on view
- Enable heartbeat
- Erase secrets
- Hide launcher password
- Move to folder
- Run heartbeat
- Set privileged account
- Undelete
- Unhide launcher password

Note: Bulk operations differ by Secret Server version.

User Interfaces, Themes, and Color Modes

Overview

Secret Server has two categories of user interface (UI) "skins"—the "new UI" and the "classic UI."

Important: The availability of these features is up to your Secret Server admin. It is possible that you do not have access to the classic UI at all. At your Secret Server admin's discretion, your Secret Server can default to either UI when you first open it.

Terms

Settings and terminology for the two UIs are intertwined in Secret Server, so some clarification is in order—some definitions:

- **Classic UI:** The original, utilitarian, "90s-looking" interface, which many still prefer. It is sometimes called Secret Server classic.
- **New UI:** The refined, modern-looking interface with enhanced usability and aesthetics.
- **Theme or Classic Theme:** A color scheme (skin) for the classic UI. The "real" themes are Secret Server Classic - Blue, Secret Server Classic - Dark, Secret Server Classic - Default (Delinea green), Secret Server Classic - Gray, and Secret Server Classic - Green. The new UI masquerading as a classic UI theme is called "Secret Server New" (see [Best Practices](#)).
- **Color Mode:** A color scheme (skin) for the new UI. The color modes are System Default, Light, and Dark. System default means whatever color was chosen on your system for the Windows application default.
- **Secret Server Classic:** An alternate name for the classic UI. This appears at times in Secret Server.

Best Practices

For Users

It is easy to get befuddled about settings and terms in the two UIs because:

- Two different terms are used to denote essentially the same thing—*themes* and *color mode*. The former is used for the classic UI and the latter for the new UI.
- You can configure settings for the new UI while in the classic UI. That is, you can set the default color mode while in the classic UI, which does not use color modes. Presumably, this was intended to make it easier to switch back and forth between the two UIs. It also makes it easier to confuse which setting goes with which UI.
- The "My Theme" dropdown list in the "Edit My Preferences" section of the classic UI lists the new UI as one of the themes, which it is not—it is an entirely new UI. This means that *can* use that dropdown to set your default UI to the new UI and switch to it right away (as soon as you click the Save button), but we suggest avoiding that. The dropdown's stated purpose is setting the theme for the classic UI. We recommend not using it to switch from the classic UI to the new UI or to set the new UI color mode—do that from the new UI.

To combat that we recommend:

- Most importantly, just remember that themes are for the classic UI and color modes are for the new UI. They both refer to color (and icon) schemes (skins) for their respective interfaces.
- Pick one UI and stick with it. We recommend the new UI.
- If you want to use the classic UI, set your user default theme right after reading this, while this topic is fresh in your mind, and then leave it be.

For Admins

You can set the default theme for new users. With the correct settings chosen, users can override the Secret Server defaults.

We recommend encouraging your users to use the new UI. As an admin, you can limit your users using two settings on the **Admin > Configuration** page:

- **Enable New User Interface As Default for New Users** check box.

- **Allow Users to Select Classic Theme** check box.

You can also control the ability for users to set themes at the role level using the "Allow User to Select Themes" role permission.

Procedures for Users

Important: The availability of these features is up to your Secret Server admin.

Switching to the New UI from the Classic UI

Hover the mouse pointer over the user icon at the top right of any Secret Server page and select **View in New UI**. The interface changes to the new UI (set to the default color mode for the logged on user).

Switching to the Classic UI from the New UI

Click the user icon at the top right of any Secret Server page and select **View in Classic UI**. The interface changes to the classic interface (set to the default theme for the logged on user).

Setting Your Default Classic UI Theme

1. If necessary, switch to the classic UI: Click the user icon in the top right of any page and select **View in Classic UI**. The user interface changes to the classic UI.
2. Hover the mouse pointer over the user icon at the top right of any Secret Server page and select **Account Settings**. The General tab of the My Preferences page appears:

My Preferences

General Launcher

- Clear the Search field when changing folders (classic UI)
- Mask passwords when viewing Secrets
- Send email when dependencies fail to update
- Send email when Secrets are changed
- Send email when Heartbeat fails for Secrets
- Send email when Secrets are viewed
- Show the full folder path on search results

Language: Browser Language

My Theme: Secret Server Classic - Blue

Color Mode: System Default

Date Format: M/d/yyyy

Time Format: hh:mm tt

[Back](#) [Edit](#)

3. Click the **Edit** button. The page becomes editable.
4. Click the **My Theme** dropdown list to select the desired theme (skin). Do **not** select Secret Server New (see the note below). The default setting simply means you want to use your Secret Server admin's choice, not yours.
5. Leave the **Color Mode** dropdown list alone—it only applies to the new UI (see the note below).
6. Click the **Save** button. The current theme changes to your choice, and your default is set.

Note: You *can* select Secret Server (New) to change your default to the new UI. As soon as you click the Save button, the interface changes to the new UI until you change it, even if you log off. If you choose Secret Server (New), the Color Mode dropdown list becomes relevant, so you should set it before clicking the Save button. For clarity, we recommend configuring new UI settings from the new UI and using the user icon to switch between UIs, but it is your choice (see [Best Practices](#)).

Setting Your Default Color Mode

Note: While it is possible to set this while in the classic UI, we recommend changing it while in the new UI (see [Best Practices](#)).

1. If necessary, switch the new UI: Hover the mouse pointer over the user icon at the top right of any Secret Server page and select **View in New UI**. The interface changes to the new UI (set to the default color mode for the logged on user).
2. Click the user icon at the top right of any page and select **User Preferences**. The User Preferences Page appears:

User Preferences

[General](#) [Settings](#)

General Information

This section contains general information about your user account within your organization's framework. The information in this section cannot be edited on this page. To update your display name, username, or email, please contact your Secret Server administrator.

Display Name	
Username	
Email	
Last Login	8/13/2020 03:15 pm

Password Settings

Change your local user password when logging into Secret Server, or make changes to your user account's DoubleLock password, here.

[Change Password](#)
[Change DoubleLock Password](#)
[Reset DoubleLock Password](#)

3. Click the **Settings** tab:

User Preferences

+
WS

General
Settings

General Information

This section contains general information about your user account within your organization's framework. The information in this section cannot be edited on this page. To update your display name, username, or email, please contact your Secret Server administrator.

Language	Browser Language
Color Mode	System Default
Date Format	M/d/yyyy (M/d/yyyy - 1/31/1980)
Time Format	hh:mm tt (hh:mm tt - 09:09 PM)
Login Home	All Secrets

Email Settings

Users are able to choose what you want to be notified about, based on Secret events in Secret Server. Enabling these email notifications will apply to all

Send Email When Dependencies Fail to Update	<input type="checkbox"/>
Send Email When Secrets are Changed	<input type="checkbox"/>

4. Click the **Color Mode** dropdown list to select the desire color mode. Your choices are:

- o Light
- o Dark
- o System Default: The color chosen for your Windows default application color.

Procedures for Admins

Choosing the Default Classic UI and Theme for New Users

This is a procedure for admin users that determines what all newly created users default to.

Note: This instruction assumes you are using the new user interface. The method for the classic user interface is nearly identical.

1. Go to **Admin > Configuration**.
2. Ensure the **General** tab is selected.
3. Click the **Edit** button at the bottom of the page. The page becomes editable.
4. Go to the **User Interface** section.
5. Click the **Select Default Classic Theme** dropdown list to select the theme. Your choices are:

- Secret Server Classic - Blue Secret Server
- Secret Server Classic - Dark
- Secret Server Classic - Default
- Secret Server Classic - Gray
- Secret Server Classic - Green

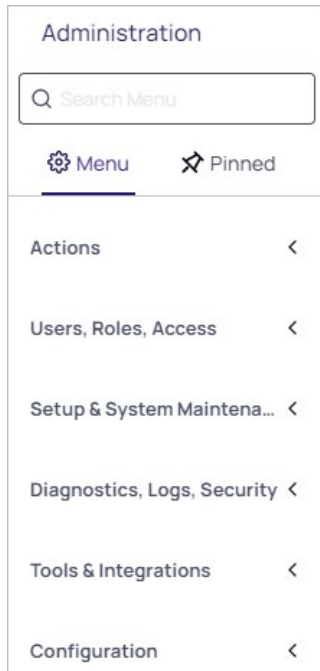
6. Click the **Save** button. The classic user interface theme for new users is now set.

Choosing the Default New UI Color Mode for New Users

This is not configurable by design. Users default to the system default (Windows application) color mode.

The Configuration Overview page (also called the "configuration preview") is a dynamic, one-stop locator for Secret Server configurations. To use the page:

1. Click the **Administration** menu item.
2. Click the >> button to view the admin side panel:




3. Click **Configuration > Configuration Overview**. The configuration overview page appears:

Admin > Configuration

Configuration

Search configuration section names, labels, and actual configuration values.



Security Hardening

The security hardening report identifies preferred configuration settings or possible security issues.

CONFIGURATION

- Configuration Overview
- Application

Permission

- Remote Passwo
- Security

4. Either choose a configuration section from the links at the bottom or type a configuration name in the dynamic search box. Either way, the matching configuration sections appear at the bottom (replacing the original links).
5. Refer to the documentation topics for the specific configuration you navigated to.

If your company requires the login banner for usage agreements and conditions to be visible when users log into Secret Server:

To enable the login banner, follow the procedure below:

1. Click the **Administration** menu item and search for and select **Configuration**.
2. In the **Configuration** window, select the **Login** tab.

Configuration

General **Login** SAML Folders Local User f




Allow Remember Me

Allow Two Factor Remember Me

Allow AutoComplete

3. On the **Login** tab, scroll down to the bottom options and click **Login Policy Agreement**.

Key Integration (for SSH Terminal)	Yes
Authentication Method	Pl
FA Validation Bypass	No
Expiration in	8 h

 Edit  Login Policy Agreement  Test Radius

4. In the **Administration Login Policy** window, click **Edit**.

Administration Login Policy

Note: You may change the contents of the Login Policy Statement by editing the file "policy.txt". Your changes will ***NOT*** be lost during upgrades.

Enable Login Policy No

Force Login Policy No

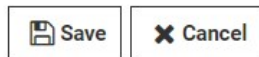


5. Select the checkbox for **Enable Login Policy** and click **Save**.

Administration Login Policy

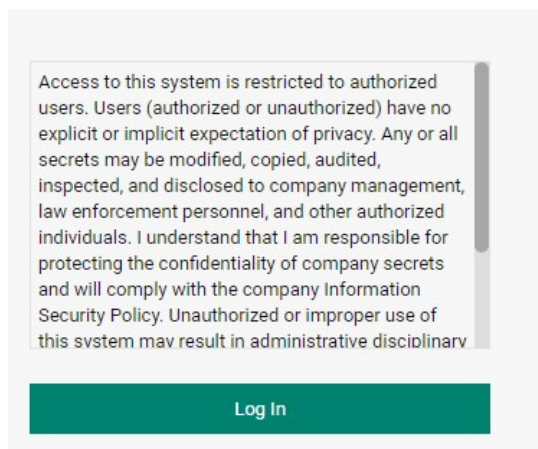
Enable Login Policy

Force Login Policy



6. Log out of Secret Server and re-try logging in.

On the Secret Server Login page, users logging into Secret Server will see the default message provided in the login policy box.



Note: The login policy is not applicable to the mobile app and will not be shown.

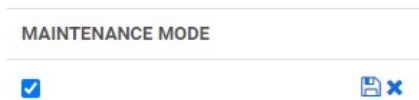
Turning on maintenance mode allows you to temporarily prevent users from changing roles, secrets, or secret-related data such as dependencies, templates, and password requirements. For example, you would want to enable Maintenance Mode while migrating the Secret Server application to a new server with a different domain.

To turn on Maintenance Mode:

1. On the **Admin** page, click **Setup & System Maintenance**.
2. Click **Server Nodes**.
3. In the **Maintenance Mode** column, click the edit icon next to Disabled.



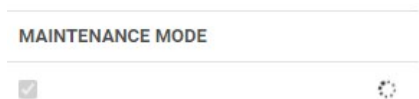
4. Check the box that appears and click the Save icon. If you change your mind, you can click the **X** next to the Save icon.



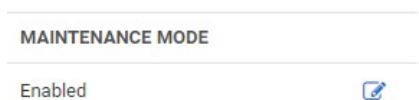
A notice appears stating, "Enabling Maintenance Mode will take 5 minutes. Are you sure you want to proceed?"

5. Click **OK**.

While Maintenance Mode is setting up, a spinner appears.



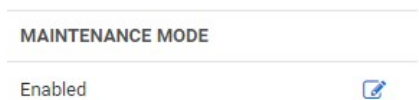
When Maintenance Mode is enabled, "Enabled" appears in the Maintenance Mode column.



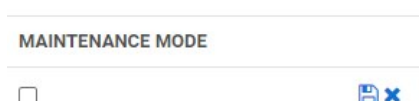
Note: When Secret Server is in Maintenance Mode; a notification bar is displayed to alert users.

To return Secret Server from Maintenance Mode to normal operation, return to the Server Nodes window, then follow these steps:

1. In the Maintenance Mode column, click the edit icon next to Enabled.



2. Uncheck the box that appears and click the Save icon. If you change your mind, you can click the **X** next to the Save icon.



Note: When Secret Server is in its normal running mode, the Maintenance Mode notification bar is no longer displayed to users.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This section discusses built-in security features of the Secret Server application, including encryption and compliance standards.

Advanced Encryption Standard

Secret Server uses different types of encryption to ensure data security. Every text-entry field, except name, on a secret is encrypted at the database level with the Advanced Encryption Standard (AES) 256-bit algorithm. Database encryption prevents unauthorized access of sensitive data on the server.

The AES encryption algorithm provides a high security level for sensitive data. The National Institute of Standards and Technology (NIST) and National Security Agency (NSA) search for a replacement for the Data Encryption Standard (DES), which had numerous issues, namely small key size and efficiency, and finally settled on AES.

Note: Encryption algorithms use keys to obfuscate the data. While DES only had a key size of 56 bits, AES can have a key size of 128, 192 or 256 bits. Larger keys provide more security as their size makes brute force attacks infeasible.

Note: To address concerns from the cryptographic community, NIST embarked on a transparent selection process. During the selection process NIST solicited designs from the global cryptographic community and voted for a winner from within fifteen finalists. The eventual winner was a team of Belgian cryptographers with their submission of the Rijndael encryption method, which became AES. For more information about the technical specifications of AES, please see the official standard.

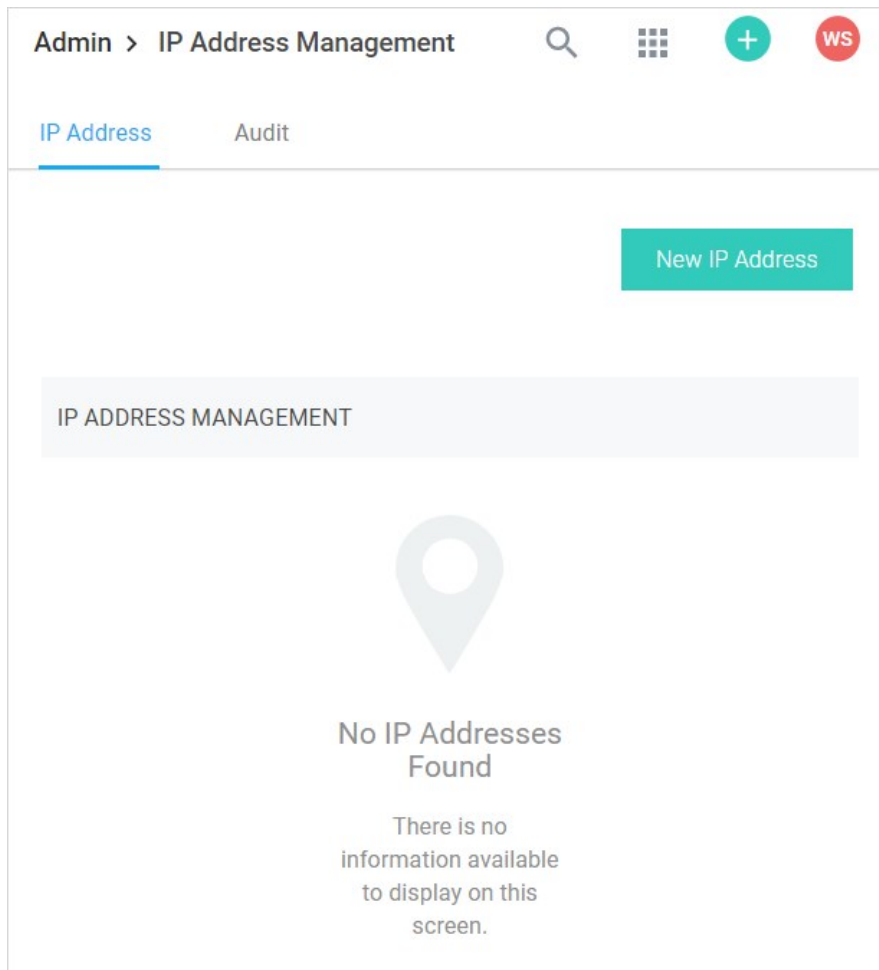
Restricting IP Addresses

IP address restrictions allow you to control which IP address ranges users can use to log in to Secret Server.

Creating IP Address Ranges

To create an IP address range:

1. Go to **Admin > IP Addresses** under Administration. The IP Address Management page appears:



2. Click the **New IP Address** button. The Add New IP Address Range popup page appears:

Add New IP Address Range

IP Address User/Network Name *

IP Address Range *

192.168.3.12
 192.168.42.147-192.168.42.194
 192.168.3.52/22

Cancel Save

3. In the **IP Address User/Network Name** text box, type a descriptive name for your range.
4. In the **IP Address Range** text box, enter an IP Address or IP Address range. Secret Server supports single IP Addresses (10.0.0.4), a range separated by a hyphen (10.0.0.1-10.0.0.255), and CIDR notation (10.0.0.0/24).
5. Click the **Save** button. The new address or range appears in the IP Address Management table:

New IP Address

IP ADDRESS MANAGEMENT

NAME	↑
IP ADDRESS RANGE	⚙️
Generic Internal 192.168.1.1	Edit Delete

Note: You can show or hide columns in the table by clicking the button.

Editing and Deleting IP Address Ranges

To edit an IP address range, go to the **IP Address Management** page, click on a range, and click **Edit**. To delete a range, click on the range and click the **Delete** button.

Assigning an IP Address Range

1. To assign a range to a user:
2. Go to **Admin > Users** page. The View User page appears:

View User

User Name	wsprunk
Display Name	Will Sprunk
Email Address	
Domain	gamma.thycotic.com
Two Factor	< None >
Enabled	Yes
Locked Out	No
Application Account	No
IP Address Restrictions	
None	
Restricted By Team	No

3. Scroll to the bottom of the page and click the **Change IP Restrictions** button. The Edit IP Address Restrictions Page appears:

Edit IP Address Restrictions

RESTRICTED	NAME	IP ADDRESS RANGE
<input type="checkbox"/>	Generic Internal	192.168.1.1

4. Click to select or deselect check boxes next to the ranges to choose which IP Addresses a user can use to access Secret Server. If no boxes are checked, the user can access Secret Server through any IP Address.
5. Click the **Save** button.

Note: Regardless of the restrictions, users can always log in when accessing Secret Server on the server using a local IP address (127.0.0.1). This prevents total lockout from Secret Server.

Secret Key Rotation

Overview

Secret key rotation is a somewhat similar process to RPC by which the encryption key, used for securing secret data, is changed and that secret data is re-encrypted. Each secret receives a new, unique AES-256 key. Secret key rotation can be used to meet compliance requirements that mandate encryption keys be changed on a regular basis.

How to Perform Secret Key Rotation

Note: Secret key rotation requires the Rotate Encryption Keys permission.

1. Go to **Admin > Configuration > Security**.
2. In the **Key Rotation** section, click the **Rotate Secret Keys** button.

Secret key rotation begins as soon as Secret Server enters maintenance mode. Because maintenance mode disables various functionality (such as secrets cannot be updated), the timing of secret key rotation merits consideration of Secret Server processing time. We recommend running secret key rotation during off-peak or non-business hours.

Note: To learn more about maintenance mode, see the [Maintenance Mode FAQ](#).

Estimated Processing Time

Maintenance mode takes five minutes to enable before secret key rotation is started. The processing time for secret key rotation varies greatly, depending on the following factors:

- Total number of secrets
- Total number of secrets with file attachments and the size of those file attachments
- Hardware configuration:
 - Number of CPUs and cores
 - Memory size
 - Network latency
- HSM key size, if applicable

As a general guideline, use the following:

Table: Secret Key Rotation Processing Time

Without HSM (default)	2,000-12,000 secrets per minute
HSM with a 2048-bit key	240-600 secrets per minute
HSM with a 4096-bit key	120-300 Secrets per minute

Security Compliance Standards

FIPS Compliance

The Federal Information Processing Standard 140-1 (FIPS 140-1) and its successor (FIPS 140-2) are United States Government standards that provide a benchmark for implementing cryptographic software. Secret Server has been tested within environments that are FIPS compliant. For instructions to enabling FIPS in Secret Server, see the [Enabling FIPS Compliance in Secret Server](#) KB article.

PCI Datacenter Compliance

Secret Server can make it easier to comply with PCI-DSS requirements:

- **Requirement 8:** Assign a unique ID to each person with computer access: Secret Server helps you comply with Requirement 8 by providing a secure repository for you to maintain an automated password changing schedule, forcing each user to have a unique, secured password. Secret Server's Web-based access makes it easy to access these passwords.
- **Requirement 10:** Track and monitor all access to network resources and cardholder data: Secret Server can monitor all access to network resources. By employing remote password changing to force password changes, administrators can monitor and update network resources on a customized schedule. You can create a password changing schedule that best suits your environment.
- **Requirement 11:** Regularly test security systems and processes.
- **Requirement 12:** Maintain a policy that addresses information security: You can optimize Secret Server's software's global configuration and template-driven data structure to fit the requirements of your current information security policy or assist in creating a policy based on Secret Server. Configuration options include:
 - Applying two-factor authentication
 - Enabling launchers
 - Enabling Web services
 - Enforcing local-user password requirements
 - Forcing HTTPS/SSL
 - Requiring folders for secrets (for uniform permissions)

SSH Key Rotation

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

SSH Key Rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair will be regenerated and the private key encrypted using a new passphrase. The public key will then be updated on the Unix machine referenced on the secret.

There are two topics addressed here:

- [Basic SSH Key Rotation](#): A step-by-step tutorial on quickly getting started with the default SSH key rotation secret types and password changers.
- [Custom SSH Key Rotation](#): Provides additional information for users who need to customize the default commands for their environment.

Basic SSH Key Rotation

This topic is a tutorial on how to quickly get started using SSH key rotation to change a Unix account's public and private key and automatically update a remote machine using Remote Password Changing (RPC).

Introduction

SSH key rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair is regenerated and the private key encrypted using a new passphrase. The public key is then be updated on the Unix machine referenced on the secret.

This document is a tutorial showing you how to quickly get started using SSH key rotation using our default key rotation password changers. For an in-depth description of SSH key rotation including modifying the command sets for your environment, see our [Custom SSH Key Rotation](#) topic.

Requirements

To use our default SSH key rotation commands, the following minimum requirements must be met on the machine being managed:

- SSH key logins should be enabled on the target using keys in OpenSSH format. A secret can be created with keys in PuTTY format but they will be converted to OpenSSH when the key is rotated.
- Public keys should be stored in `[-userhome]/.ssh/authorized_keys` (not `authorized_keys2`).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH key rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password and the permissions to edit the user's `authorized_keys` file with sudo.

If a system does not meet these requirements it may still be possible to do key rotation by modifying the key rotation command sets. Our [Custom SSH Key Rotation](#) topic describes how to do this.

Configuring a Secret for SSH Key Rotation

Secret Server comes with two secret templates for SSH key rotation: **Unix Account (SSH key rotation)** and **Unix Account (Privileged Account SSH Key Rotation)**.

Use **Unix Account (SSH Key Rotation)** if:

- The account is able to change its own password and modify its own `authorized_keys` file.
- The account password and key should only be changed by Secret Server (Secret Server will always have the current password and keys).

Use **Unix Account (Privileged Account SSH Key Rotation)** if:

- The account is not able to change its own password or modify its own `authorized_keys` file.
- The account password and key may be changed outside of Secret Server, and Secret Server may not have the current account credentials. A privileged account that is able to change the password and `authorized_keys` files of other users will still be able to change the account credentials.

SSH Key Rotation Using the Secret's Credentials

Creating the Secret

1. Create a new secret in Secret Server using the **Unix Account (SSH Key Rotation)** template.

2. Enter the account user name and password.
3. Upload the private key file.
4. If the private key is encrypted using a passphrase, enter the passphrase.
5. Uploading a public key is optional but recommended. If a public key is not provided, Secret Server will regenerate it from the private key during key rotation, but if the key in `authorized_keys` is not in the same format as the generated key or does not match exactly (including comments), the rotation will fail because it could not find the public key that needs to be removed.
6. After the secret is created you should see a successful heartbeat status. If heartbeat isn't running, make sure heartbeat and RPC are enabled under **Admin > Remote Password Changing**.

Rotating the Key

1. Go to the **Remote Password Changing** tab and click **Change Password Remotely**.
2. Enter the new password or click **Generate** next to the **Next Password** field to generate a random password.
3. Click to select **Generate New SSH Key** to create a new, random SSH key. If you want to supply your own private key, uncheck this option and paste the key into the **Next Private Key** text box that appears.
4. If you have unchecked **Generate New SSH Key** you must enter the passphrase that was used to encrypt the private key at the time it was created. Leave this field blank if the private key was not encrypted with a passphrase. If you have checked **Generate New SSH Key** you have the option to enter your own passphrase, leave it blank (for an unencrypted private key), or click the **Generate** button next to the field to create a new, random passphrase. If you want to change the key without changing the passphrase, you must put the current passphrase in the **Next Private Key Passphrase** text box.
5. Click **Change** to start the key rotation and a password change. After you start the change, you can check the status either in **Admin > Remote Password Changing** or on the **Remote Password Changing** tab of the secret.
6. Once the password change / key rotation is complete the heartbeat status should be successful. You can check the audit log to see notes that the key was rotated and start a session using the key with the PuTTY Launcher.

SSH Key Rotation Using a Privileged Account

To use **Unix Account (Privileged Account SSH Key Rotation)**, you must have a secret that is able to use the `sudo` command to access other accounts' `authorized_keys` files and change their passwords. This can be any type of Unix secret and can use a password and/or private key to authenticate. If you have a secret that meets these requirements, you can set up SSH key rotation using a privileged account as follows.

Creating the Secret

1. Create a new secret in Secret Server using the **Unix Account (Privileged Account SSH key rotation)** template.
2. Enter the account user name and password.
3. Upload the private key file.
4. If the private key is encrypted using a passphrase, enter the passphrase.
5. Uploading a public key is optional, but recommended. If it is not provided, Secret Server will regenerate it from the private key during key rotation, but if the key in `authorized_keys` is not in the same format as the generated key or does not match exactly (including comments), the rotation will fail because it could not find the public key that needs to be removed.
6. After the Secret is created you should see a successful heartbeat status. If heartbeat is not running, make sure that heartbeat and RPC

are enabled under **Admin > Remote Password Changing**.

7. Next go to the **Remote Password Changing** tab and choose the privileged secret that can authenticate to the machine and modify the user's `authorized_keys` file.
8. Click the **Back** button after adding the associated secret.

Rotating the Key

1. Go to the **Remote Password Changing** tab and click **Change Password Remotely**.
2. Enter the new password or click **Generate** next to the **Next Password** field to generate a random password.
3. Click to select **Generate New SSH Key** to create a new, random SSH key. If you want to supply your own private key, uncheck this option and paste the key into the **Next Private Key** text box that appears.
4. If you have unchecked **Generate New SSH Key** you must enter the passphrase that was used to encrypt the private key at the time it was created. Leave this text box blank if the private key was not encrypted with a passphrase. If you have checked **Generate New SSH Key** you have the option to enter your own passphrase, leave it blank (for an unencrypted private key), or click the **Generate** button next to the field to create a new, random passphrase. If you want to change the key without changing the passphrase, you must put the current passphrase in the **Next Private Key Passphrase** text box.
5. Click **Change** to start the key rotation and a password change. After you start the change, you can check the status either in **Admin > Remote Password Changing** or on the **Remote Password Changing** tab of the secret.
6. Once the password change / key rotation is complete the heartbeat status should be successful. You can check the audit log to see notes that the key was rotated and start a session using the key with the PuTTY Launcher.

Troubleshooting

- The SSH Password Changers are targeted to OpenSSH. If using a different SSH library or if the user keys are not in the user's `/.ssh/authorized_keys` file you can check the commands used and modify them as appropriate under **Admin > Remote Password Changing** and clicking **Configure Password Changers**. The password changers used are **SSH Key Rotation** and **SSH Key Rotation Privileged Account**.
- Errors are logged to **Admin > Remote Password Changing**. Additional logs can be found in the Secret Server directory in the log subfolder. That is: `C:\inetpub\wwwroot\secretserver\log`.
- A change was made to how SSH script variables are named to differentiate them from tokens when testing command sets on the Configure Password Changers page. Non-token script variables should begin with an underscore. Anything in the script beginning with a dollar sign not followed by an underscore will be treated as a token and displayed as a field in the test dialog. For example:
 - `$USERNAME` References the username from the secret.
 - `$_1$USERNAME` References the username from the first linked secret.
 - `$_USERNAME` References a bash variable defined in the script.
- The default command set for the SSH key rotation privileged account password changer assumes that the `sudo` command will not prompt for a password. If your environment prompts for a password when using `sudo` the command sets will need to be modified to supply the password. If your environment caches the `sudo` credentials, the easiest way to handle this is to add the following two lines at the top of each command set on the SSH key rotation privileged account password changer:

```
sudo -i echo
$_1$PASSWORD
```

This caches the credentials for the rest of the script.

Custom SSH Key Rotation

This topic discusses how to change public keys for Unix accounts using Remote Password Changing (RPC) in Secret Server. For a step-by-step tutorial on quickly getting started with the default SSH key rotation secret types and password changers, see our [Basic SSH Key Rotation](#). The current topic provides additional information for users who need to customize the default commands for their environment.

Introduction

SSH key rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair is regenerated and the private key encrypted using a new passphrase. The public key will then be updated on the Unix machine referenced on the secret.

Secret Server provides secret templates and password changers for SSH key rotation.

Requirements

To use our default SSH key rotation commands, the following minimum requirements must be met on the machine being managed:

- SSH key logins should be enabled on the target using keys in OpenSSH format. A secret can be created with keys in PuTTY format but they will be converted to OpenSSH when the key is rotated.
- Public keys should be stored in `[-userhome]/.ssh/authorized_keys` (not `authorized_keys2`).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH key rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password and the permissions to edit the user's `authorized_keys` file with sudo.

The default command sets have been tested in the following Linux environments:

- CentOS Linux release 7.0.1406
- FreeBSD bsdRadiusServer 9.3-RELEASE-p5 i386
- Linux ubuntu 3.13.0-32-generic

If a system does not meet these requirements or has a different configuration than the tested Linux environments, it may still be possible to do key rotation by modifying the key rotation command sets. The command sets that may need to be edited and the process for doing so are described later in this topic.

Secret Templates

Secret Server includes two secret templates for SSH key rotation: **Unix Account (SSH Key Rotation)** and **Unix Account (Privileged Account SSH Key Rotation)**. The first template changes the password and key on the account using the account's credentials. Use this template if both of the following conditions apply:

- The account is able to change its own password and modify its own `authorized_keys` file.
- The account password and key should only be changed by Secret Server, which will always have the current password and keys.

Unix Account (Privileged Account SSH Key Rotation) uses an additional secret to provide the credentials for the connection that performs the password change and key rotation commands. You should use this template if either of the following conditions apply:

- The account is not able to change its own password or modify its own `authorized_keys` file.
- The account password and key may be changed outside of Secret Server, and Secret Server may not have the current account credentials. A privileged account that is able to change the password and `authorized_keys` files of other users will still be able to change the account credentials.

Creating a New SSH Key Rotation Secret

When creating a new secret based on either of these templates you will see the following form:

Create New Secret

Secret Template	Unix Account (SSH Key Rotation) Change
Folder	No Folder Selected
Secret Name *	<input type="text"/>
Machine *	<input type="text"/>
Username *	<input type="text"/>
Password *	<input type="password"/> Show Generate
Generate SSH Key	<input type="checkbox"/>
Private Key	Change
Private Key Passphrase	<input type="password"/> Show Generate
Public Key	Change
Notes	<input type="text"/>

[Cancel](#) [Create Secret](#)

1. Type the secret name in the **Secret Name** text box.
2. Type the machine in the **Machine** text box.
3. Type the username in the **Username** text box.
4. Click the **Generate** button to create a user password.
5. Click **Private Key** link to upload a file containing the private key.
6. If you are creating a new secret and want to generate a new, random private key, click to select the **Generate New SSH Key** check box. This disables the "Change" links for both the private and public keys.
7. If your uploaded private key was encrypted with a passphrase or you are generating a new key and wish to encrypt it with a passphrase, type that passphrase in the **Private Key Passphrase** text box. Otherwise, leave it empty.
8. If you are creating a new key and want to create a random passphrase for it, click this **Generate** button.

9. If you are uploading a private key, click the **Public Key Change** link to upload the corresponding public key. Uploading a public key is optional, but recommended. If not provided, Secret Server regenerates it from the private key during key rotation, but if the key in the `authorized_keys` file is not in the same format as the generated key, the old key will not be removed when the new key is added.

If neither private key nor public key is attached to the secret, a key rotation creates a new key pair, attaches them to the secret, and adds the new public key to `authorized_keys`.

After the secret is created, you should see a successful heartbeat status. If heartbeat is not running, make sure that heartbeat and RPC are enabled under **Admin > Remote Password Changing**.

Note: Heartbeat status when either the private key/passphrase or password are incorrect is indeterminate and based on the host configuration. If the system allows log on as long as one of the two is correct, it will return a successful heartbeat when the password is wrong but the key is valid and vice-versa.

If you are adding a secret using the **Unix Account (Privileged Account SSH Key Rotation)** you will also need to specify which privileged account to use during key rotation. To do this:

1. Switch to the **Remote Password Changing** tab.
2. Click the **Edit** button.
3. Click the **No selected secret** link.
4. Choose a privileged Secret that can authenticate to the machine and use the `sudo` command to access other accounts' `authorized_keys` files and change their passwords. This can be any type of Unix secret and can use a password and/or private key to authenticate.
5. Click the **Back** button to exit edit mode.

Editing the SSH Key Rotation Templates

To edit a template, go to **Admin > Secret Templates**, choose the template you want to edit in the dropdown list of templates, then click **Edit**.

You can add, remove, or edit any fields you like, but if you change or replace any of the following fields you will need to update the password changer mapping for the template:

- Machine
- Username
- Password
- Private Key
- Private Key Passphrase
- Public Key

Note: Private key and public key must remain field type "File".

If you change any of the fields listed above, click the **Configure Password Changing** button to map the fields to the password changer. Click the **Edit** button and assign all the password changer fields to the corresponding fields on the secret template.

Password Changers

Secret Server includes two password changers for SSH key rotation: **SSH Key Rotation** and **SSH Key Rotation Privileged Account**. The **Unix Account (SSH Key Rotation)** secret template uses the **SSH Key Rotation** password changer and the **Unix Account (Privileged Account SSH Key Rotation)** secret template uses the **SSH Key Rotation Privileged Account** password changer. Each of these password changers includes a set of command sets designed to change the password and public key on an account using the

secret's credentials and using sudo with a privileged account, respectively.

Although you can edit these password changers through **Admin > Remote Password Changing > Configure Password Changers**, clicking on the password changer and then **Edit** and **Edit Commands**, the recommended practice is to copy the existing password changers and then modify the copies. To do this:

1. Go to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. Scroll down to the bottom of the page and click the **New** button.
4. Select the password changer you want to copy in the **Base Password Changer** list and give the new password changer a name.
5. Click **Save**. This creates a new password changer and copies the command sets from the original password changer.
6. Enter the authentication information (see below for more information), clicking the **Save** button in each authentication field set.
7. Make the required changes to the command sets for your environment by editing existing lines, deleting lines, adding new lines, and rearranging lines.

Authentication

The authentication section defines the credentials that will be used to connect to the machine and run the command set. These can be either the credentials of the secret or credentials from an associated secret. The command sets on the **SSH Key Rotation** password changer are designed to be run with the secret's credentials. Any customized password changer based on this password changer should use the secret's credentials in the authentication section.

For more information about tokens beginning with a dollar sign used in the above screenshot, see the [Dependency Token List](#).

The command sets on the **SSH Key Rotation Privileged Account** password changer are designed to be run with the credentials of an account that can change the password and public key on behalf of other users. Any customized password changer based on **SSH Key Rotation Privileged Account** should use the credentials off one of the secret's associated secrets. This is typically the first associated secret but can be any associated secret if your password changer requires more than one associated secret. The exception to this is validation which does not run a command set by default and uses the secret's credentials. If you modify validation to use a command set you will need to change the default authentication for validation if the command set uses sudo.

Here is a typical authentication for **SSH Key Rotation Privileged Account** (except validation, which is identical to the authentication block used by **SSH Key Rotation**):

Username: \${1}\$USERNAME

Password: \${1}\$PASSWORD

Key: \${1}\$PRIVATE KEY

Passphrase: \${1}\$PRIVATE KEY PASSPHRASE

Command Sets

Overview

To handle cleanup and error-handling for key rotation two command set types were added: **Post Successful Change** and **Post Fail Change**. A password change using one of the two new password types will execute as follows:

1. Connect to the box using the specified credentials (secret credentials or an alternate secret's credentials).

2. Run the Password Change command set. This will add the new public key to the authorized_keys file. It does not change the password yet nor does it remove the old public key.
3. Verify using the new public key and old password. This verifies that the new key was added correctly.
4. If the verify check is successful, run the Post Success Change command set. This changes the password and, if successful, removes the old public key from authorized_keys if present.
5. If the verify check is unsuccessful, run the Post Failure Change command set. This removes the new public key from authorized_keys and does not change the password.
6. Return the success or failure of the overall process.

Post Successful Change and **Post Failure Change** are advanced command sets that are hidden by default. To see them, scroll to the bottom of the page and click the **Advanced Post Change Commands** link.

Password Change Command Set

In other password types the Password Change command set is only responsible for changing the password. For the two key rotation password changers, the default Password Change command set checks for the existing public key on the secret, and if found, will append a new public key to authorized_keys (the old key is then removed in the Post-Reset Command Set). The password change is done in the Post Successful Change command set only after the key change has been validated.

Verify Password Changed (Heartbeat) Command Set

Following this Reset Command Set, a Verify Password Changed is performed by attempting to connect to the host using the credentials on a secret to validate the new public key that was added to authorized_keys. If a command set is present, those commands are then run after connecting and the validation is a success only if both the connection and the command set are successful. (The command set is normally used when a secret uses alternate secrets as credentials in which case the alternate credentials are specified for authentication and command set does the actual validation of the secret.)

In the case of SSH key rotation, this validation heartbeat is run immediately after the reset command set using the current username, current password, new private key and new passphrase to connect for validation. If connection is successful, the validation is considered successful and Post Successful Change command set is run next to remove the old public key (if current private/public keys exist on the secret) from authorized_keys). If validation is not successful, the Post Fail Change command set is run to remove the new public key added during the reset.

Post Successful Change Command Set

If the Password Change command set and Verify Password Changed are both successful, the Post Successful Change command set is run. This command set finalizes the key rotation by changing the password on the account and removing the old public key from authorized_keys.

Post Fail Change Command Set

If the Password Change command set is successful but the Verify Password Changed fails, the Post Fail Change command set is run. This command set rolls back the changes made in the Password Change command set by removing the new public key from authorized_keys.

For more information about customizing SSH command sets, see [How to Create a Custom SSH Password Changer](#).

Notes

SSH Key Rotation scripts will typically be more complex than password change command sets that do not do key rotation. These scripts will often include tokens representing values from the secret and associated secrets as well as commands to verify success or failure of

previous commands using `$$CHECKFOR` and `$$CHECKCONTAINS`. For more information about these features see the **Editing a Custom Command** topic in the [Secret Server TDP](#).

The default command sets for **SSH Key Rotation Privileged Account** use `sudo` to execute several commands. These command sets assume that the `sudo` command will not prompt for a password. If your environment prompts for a password when using `sudo` the command sets will need to be modified to supply the password. If your environment caches the `sudo` credentials, the easiest way to handle this is to add the following two lines at the top of each command set on the SSH key rotation Privileged Account password changer:

```
sudo -i echo
${1}$PASSWORD
```

This will pass the credentials from the first associated secret when prompted by `sudo` and cache the credentials for the rest of the script.

Troubleshooting

- The SSH Password Changers are targeted to OpenSSH. If using a different SSH library or if the user keys are not in the users `/.ssh/authorized_keys` file you can check the commands used and modify them as appropriate under **Admin > Remote Password Changing** and clicking **Configure Password Changers**. The password changers used are **SSH Key Rotation** and **SSH Key Rotation Privileged Account**.
- Errors are logged to **Admin > Remote Password Changing**. Additional logs can be found in the Secret Server directory in the log subfolder. For example: `C:\inetpub\wwwroot\secretserver\log`.
- A change was made to how SSH script variables are named in order to differentiate them from tokens when testing command sets on the Configure Password Changers page. Non-token script variables should begin with an underscore. Anything in the script beginning with a dollar sign not followed by an underscore will be treated as a token and displayed as a field in the test dialog. For example:
 - `$_USERNAME` – References the username from the Secret.
 - `${1}$USERNAME` – References the username from the first linked Secret.
 - `$_USERNAME` – References a bash variable defined in the script.

SSL Certificates

Secret Server can be configured to run using Secure Sockets Layer (SSL) certificates. We strongly recommend that Secret Server installations run using SSL. Not using SSL significantly reduces the security of the contents of Secret Server since browsers viewing the site are not using an encrypted connection.

Important: This feature is part of the early release of Secret Server 10.11. The general release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version.

Overview

Secret Server can now export and import Secret Server settings as a JavaScript Object Notation (JSON) file. With this, you can more easily move settings from an existing Secret Server environment to another.

Prerequisites

Required General Permissions

There are the permissions required to access and perform the process. These are:

To view the Export/Import page or menu link:

- Administer Export or View Export
- Administer Import

To view audits (at least one is required):

- Administer Users
- Own User
- View Users

Exporting to a JSON file:

- Administer Configuration
- Administer Export

Importing from a JSON file:

- Administer Configuration
- Advanced Import

Required Additional Permissions

Some of the settings require additional permissions to export or import:

Table: Required Additional Permissions

| Setting | Permission | -- | -- | | OpenID Log on | Administer Delinea One | | SAML | Administer Configuration SAML | | Security | Administer Configuration Security | | Session Recording | Administer Configuration Session Recording | | SSH Commands | Administer SSH Menus | | Delinea One Log on | Administer Delinea One | | Two Factor Log on | Administer Configuration Two Factor | [Unexpected Link Text](#)

Required Licenses

Additional licenses may be required to import or export some settings.

Advanced Auditing License

This license is required for these settings in the Application Settings category:

- SyslogCefLogSite
- SyslogCefPort
- SyslogCefProtocol
- SyslogCefServer
- SyslogCefTimeZone

Enterprise Edition

Note: these settings are also available with the Professional Edition and Approval Workflow Add-on licenses.

The Enterprise Edition license is required for these settings in the Launcher Settings category:

- CheckInSecretOnLastLauncherClose
- CloseLauncherOnCheckInSecret

It is required for these settings in the Permission Options category:

- EnableApprovalFromEmail
- ForceSecretApproval

It is required for the TicketSystems category.

Pro Edition

The Pro Edition license is required for these setting categories:

- SAML
- Session Recording

Platinum Edition

The Platinum Edition license (or Pro Edition and Unix SUPM licenses) is required for the SSH Commands setting category.

Procedures

Exporting Settings

To export Secret Server settings:

1. Go to **Admin > All**. The Admin page appears:

What are you looking for?

Search for an admin option



Simple View ▾



Actions

Secret Server features that perform important jobs



Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

TOOLS & INTEGRATIONS

Launcher Tools

Connection Manager

SDK Client Management


Privilege Manager

Privileged Behavior Analytics

DevOps Secrets Vault


Slack Integration

2. Click the **Setup & System Maintenance** button. A menu appears alongside the button:




Actions

Secret Server features that perform important jobs




Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more




Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

SETUP & SYSTEM UPKEEP

- Backup Configuration
- Database
- Email Server
- Licenses
- Mobile Onboarding
- Search Indexer
- Upgrade Secret Server

IMPORT, EXPORT, SYNC

- Export / Import
- Folder Sync

NETWORKING

- Distributed Engine
- Server Nodes
- Internal Site Connector

3. Click the **Export / Import** menu item. The Secrets tab of the Export / Import page appears:

Admin > Export / Import 🔍 🏠 + WS

Secrets Secret Server Settings

72 Items

Export

Import

DATE RECORDED	USER	ACTION	NOTES
2/19/2021 02:08 pm	admin	EXPORT	Exported Secret Count...
2/19/2021 02:05 pm	admin	IMPORT	Imported 1 secrets

4. Click the **Secret Server Settings** tab:

Admin > Export / Import

Secrets Secret Server Settings

2 Items Export Import

DATE RECORDED	USER	ACTION	NOTES
2/24/2021 10:08 am	[REDACTED]	SECRET SERVER S...	Application Settings
2/24/2021 10:07 am	[REDACTED]	SECRET SERVER S...	Application Setting...

5. Click the **Export** button. The Settings Export page appears:

Admin > Export / Import > Settings Export

Settings Export

This feature allows you to export pre-selected settings, which can be imported to other Secret Server environments for streamlined setup.

To achieve this, select which setting categories you wish to export.
[KB Link](#)

Setting Categories *

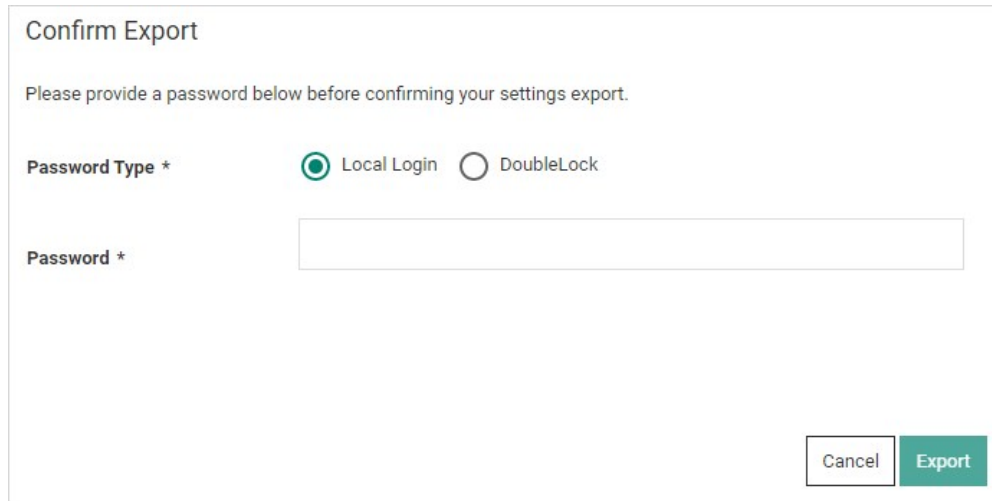
- Configuration
 - Application Settings
 - Launcher Settings (Runtime)
 - Protocol Handler Settings (Install-Time)
 - Permission Options
 - User Experience
 - User Interface
 - Advanced Settings
 - Login
 - Folder Settings
 - Local User Passwords
 - Security
 - Email
 - Ticket System
 - Session Recording
 - SAML
 - SSH Commands
 - Licenses

Cancel Export Settings

- Click to select the check boxes for the settings categories you wish to include. Clicking the **Configuration** check box selects all available categories.

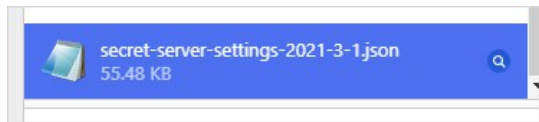
Note: See the [Setting Category Reference](#) section for details on the settings in each category.

- Click the **Export** Settings button. A Confirm Export popup appears:



The image shows a 'Confirm Export' dialog box. At the top, it says 'Please provide a password below before confirming your settings export.' Below this, there are two radio button options for 'Password Type *': 'Local Login' (which is selected) and 'DoubleLock'. Underneath is a text input field for 'Password *'. At the bottom right, there are two buttons: 'Cancel' and 'Export'.

- Click the **Passport** selection button to choose **Local Login** or **DoubleLock** if that applies.
- Type your password in the **Password** text box.
- Click the **Export** button. The JSON file appears in your browser's downloads:



Note: This example is for the Vivaldi Chrome browser—yours will likely look different.

Importing Settings

To Import Secret Server settings:

- Go to **Admin > All**. The Admin page appears:

What are you looking for?

Search for an admin option



Simple View ▾



Actions

Secret Server features that perform important jobs



Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

TOOLS & INTEGRATIONS

Launcher Tools

Connection Manager

SDK Client Management


Privilege Manager

Privileged Behavior Analytics

DevOps Secrets Vault


Slack Integration

2. Click the **Setup & System Maintenance** button. A menu appears alongside the button:




Actions

Secret Server features that perform important jobs




Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more




Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

SETUP & SYSTEM UPKEEP

- Backup Configuration
- Database
- Email Server
- Licenses
- Mobile Onboarding
- Search Indexer
- Upgrade Secret Server

IMPORT, EXPORT, SYNC

- Export / Import
- Folder Sync

NETWORKING

- Distributed Engine
- Server Nodes
- Internal Site Connector

3. Click the **Export / Import** menu item. The Secrets tab of the Export / Import page appears:

Admin > Export / Import 🔍 🏠 + WS

Secrets Secret Server Settings

72 Items

Export
Import

DATE RECORDED	USER	ACTION	NOTES
2/19/2021 02:08 pm	admin	EXPORT	Exported Secret Count...
2/19/2021 02:05 pm	admin	IMPORT	Imported 1 secrets

4. Click the **Secret Server Settings** tab:

Admin > Export / Import

Secrets [Secret Server Settings](#)

2 Items Export Import

DATE RECORDED	USER	ACTION	NOTES
2/24/2021 10:08 am	[REDACTED]	SECRET SERVER S...	Application Settings
2/24/2021 10:07 am	[REDACTED]	SECRET SERVER S...	Application Setting...

5. Click the **Import** button. The Settings Import page appears:

Admin > Export / Import > Settings Import

Settings Import

This feature allows you to import settings from an existing Secret Server environment to aid in gaining congruently configured environments.

To achieve this, select for import a JSON file that was exported from the desired Secret Server environment configuration. Executing this import will override existing setting configurations of this environment.
[KB Link](#)

Import File * [Change](#)

Setting Categories *

- Configuration
 - Application Settings
 - Launcher Settings (Runtime)
 - Protocol Handler Settings (Install-Time)
 - Permission Options
 - User Experience
 - User Interface
 - Advanced Settings
- Login
- Folder Settings
- Local User Passwords
- Security
- Email
- Ticket System
- Session Recording
- SAML
- SSH Commands
- Licenses

- Click the **Change** link, and navigate to and select the JSON file you want to import. The name of the file you chose appears above the Change link:

secret-server-settings-2021-3-1.json (55.48 KB)
[Change](#)[Clear](#)

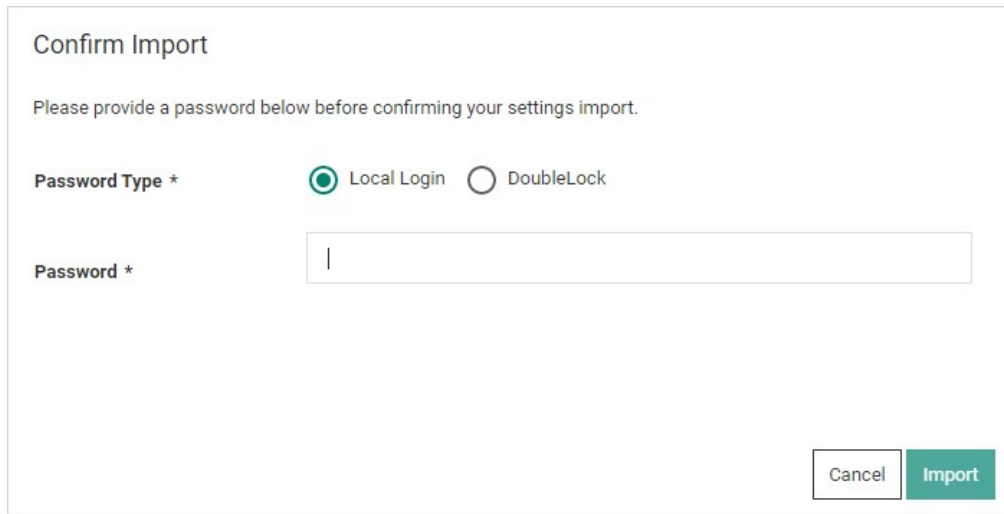
Configuration

- Click to select the check boxes for the settings categories you wish to include. Clicking the **Configuration** check box selects all available categories.

Note: Some settings may not allow you to select them, based on your permissions and licenses. Another possibility is the category was not included in the original export. Hover the mouse pointer over any of these settings to view a hint of what is

likely causing it. **Note:** See the [Setting Category Reference](#) section for details on the settings in each category.

8. Click the **Import Settings** button. A Confirm Import popup appears:

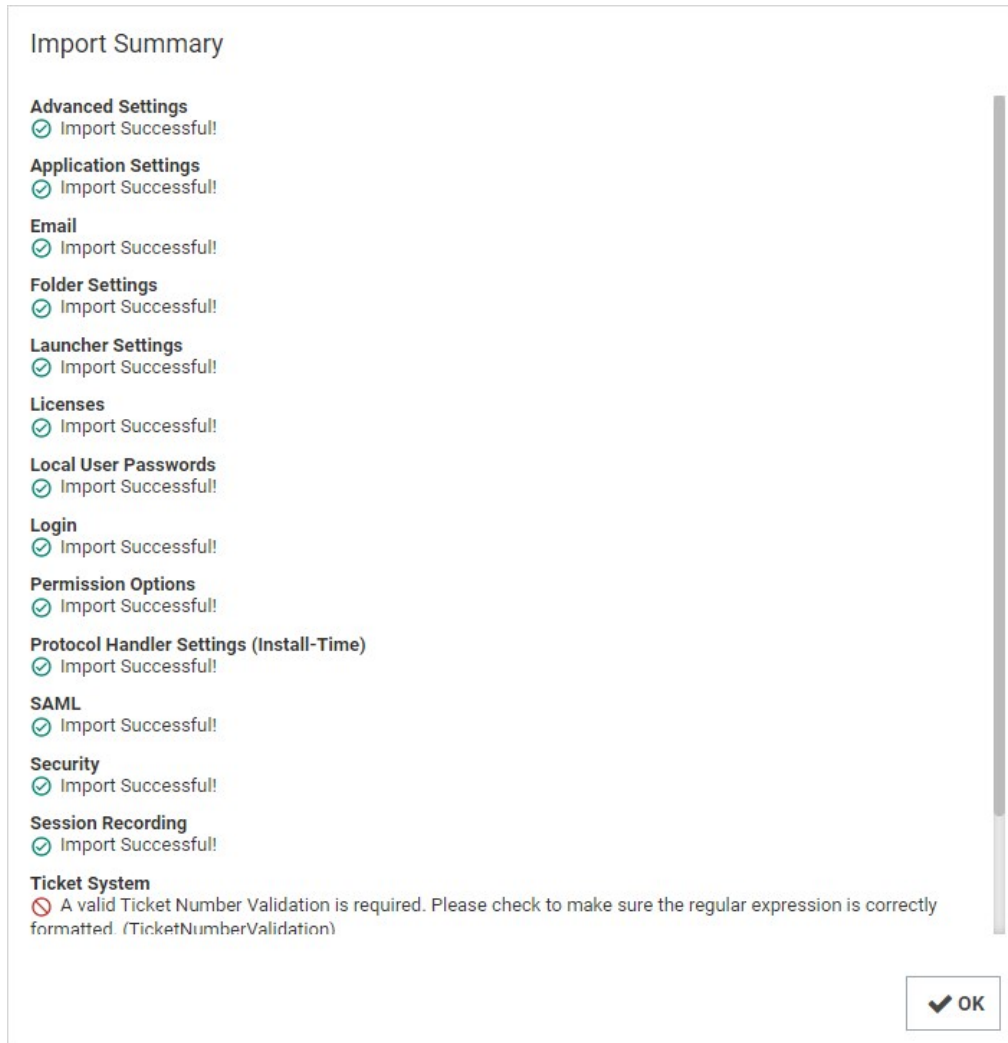


The image shows a 'Confirm Import' dialog box. At the top, it says 'Confirm Import'. Below that, it says 'Please provide a password below before confirming your settings import.' There are two radio button options for 'Password Type *': 'Local Login' (which is selected) and 'DoubleLock'. Below the radio buttons is a text input field for 'Password *' with a vertical cursor. At the bottom right, there are two buttons: 'Cancel' and 'Import'.

9. Click the **Password** selection button to choose **Local Login** or **DoubleLock** if that applies.

10. Type your password in the **Password** text box.

11. Click the **Import** button. An Import Summary popup appears:



12. Click the **OK** button. The importation appears in the log that you saw earlier on the Export / Import page.

Setting Category Reference

This section details what settings are contained in the following settings categories:

Note: Some settings are unavailable in certain environments or if requiring a license or permission.

Application Settings

These settings correspond to the Application Settings section on the Configuration General page.

This setting is unavailable in an on-premise environment:

- DisplayDowntimeMessageToAdminsOnly

These settings are unavailable in a cloud environment:

- AllowSoftwareUpdateChecks
- CustomURL

- EnableKeepAliveThread
- TmsRootUrl
- WriteSyslogToEventLog

This setting is unavailable in an IBM environment:

- AllowSendTelemetry

Advanced Settings

These settings correspond to the Advanced Settings section on the Configuration Advanced page.

Launcher Settings (Runtime)

These settings correspond to the Launcher Settings (Runtime) section on the Configuration General page.

Launcher Deployment Type setting can be one of the following:

- 0: Click Once
- 1: Protocol Handler

This setting is unavailable in a cloud environment:

- LauncherDeploymentType

Email

These settings correspond to the Email tab on the Configuration Email page.

These settings are unavailable in a cloud environment:

- SmtDomain
- SmtPassword
- SmtPort
- SmtServer
- SmtUseCredentials
- SmtUseImplicitSSL
- SmtUserName
- SmtUseSSL

Folder Settings

These settings correspond to the Folders tab on the Configuration Folders page.

Licenses

These settings correspond to the licenses listed on the Licenses page.

Local User Passwords

These settings correspond to the Local User Passwords tab on the Configuration Local User Passwords page.

Login

These settings correspond to the Login tab on the Configuration Login page.

These settings unavailable in a cloud environment:

- CacheAdCredentials
- TwoFactor.Radius.ClientPortRange

Permission Options

These settings correspond to the Permission Options section on the Configuration General page.

The Default Secret Permissions setting can be one of the following:

- 0: Secrets inherit permissions from folder
- 1: New Secrets copy permissions from folder
- 2: Only creator has permissions to new Secrets

Protocol Handler Settings (Install-Time)

These settings correspond to the Launcher Settings (Runtime) section on the Configuration General page.

SAML

These settings correspond to the SAML tab on the Configuration SAML page. To insert a new identity provider, in the same instance the export file came from, the IdentityProviderId setting must be set to 0. Otherwise, it will treat it as an update (see External Instance Id). The identity provider name cannot match another already in the database.

Security

These settings correspond to the Security tab on the Configuration Security page.

These settings are unavailable in a cloud environment:

- DatabaseIntegrityMonitoringSymmetricKey
- EnableDatabaseIntegrityMonitoring
- EnableHSTS
- FipsEnabled
- ForceHttps
- HSTSMaxAge

Session Recording

These settings correspond to the Session Recording tab on the Configuration Session Record page. The launcher must be enabled, and a valid license for Session Monitoring is required to export and import this feature.

These settings are unavailable in a cloud environment:

- ArchiveLocationBySite
- ArchivePath
- DaysUntilArchive
- EnableArchive
- EnableHardwareAcceleration
- StoreInDatabase
- VideoCodeId

To update SSHProxyRecordVideo or SSHProxyRecordKeyStrokes, SSH Proxy must be enabled.

To update RDPProxyRecordVideo or RDPProxyRecordKeyStrokes, RDP Proxy must be enabled.

SSH Commands

These settings correspond to the SSH command restrictions, the SSH commands, allowed command menus, and blocked command lists.

Ticket System

These settings correspond to the Ticket System tab on the Configuration Ticket System page. To insert a new ticket system in the same instance the export file came from, TicketSystemId must be set to 0. Otherwise, it will treat it as an update (see External Instance Id). The ticket system name cannot match another already in the database.

User Experience

These settings correspond to the User Experience section on the Configuration General page.

- Application Language
- Default Date Formats can be found in the tbDateOptions table.
- Default Time Formats can be found in the tbTimeOptions table.
- Default New User Roles can be found in the tbRoles table
- Server Time Zones can be found in the server registry: Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones

User Interface

These settings correspond to the User Interface section on the Configuration General page.

These settings are unavailable in an IBM environment:

- AllowUserToSelectTheme
- CustomLogoCollapsed
- CustomLogoFullSize

JSON Export File

In addition to the setting categories, here are a few components of the JSON export file that you should be aware of.

External Instance ID

```
"externalInstanceid": "95931fb9-02b0-47a5-a59d-69d6543a192d",
```

The external instance ID is an identifier for the Secret Server instance the settings were exported from. If you change this ID, Secret Server will assume the export came from another database and will insert new records for the ticket system (TicketSystemId) or SAML (Identity Providers—IdentityProviderId) categories. To add a new record in the same instance, set the ID to 0 and it will be treated as a new item.

Configuration Version

```
"configurationVersion": "1.0.0",
```

This is the configuration version the settings were exported from. In the future when other settings are added, it will help Secret Server determine which settings are available and which are not in the database.

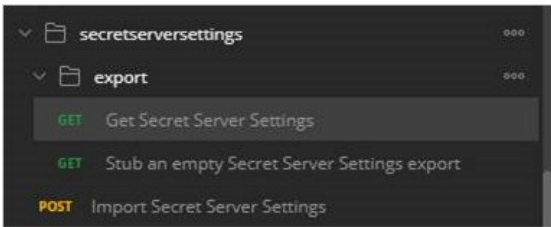
JSON Import File

In the UI, the exported JSON file can be easily modified and used as the import JSON file. For the API, the exported JSON must be added to the

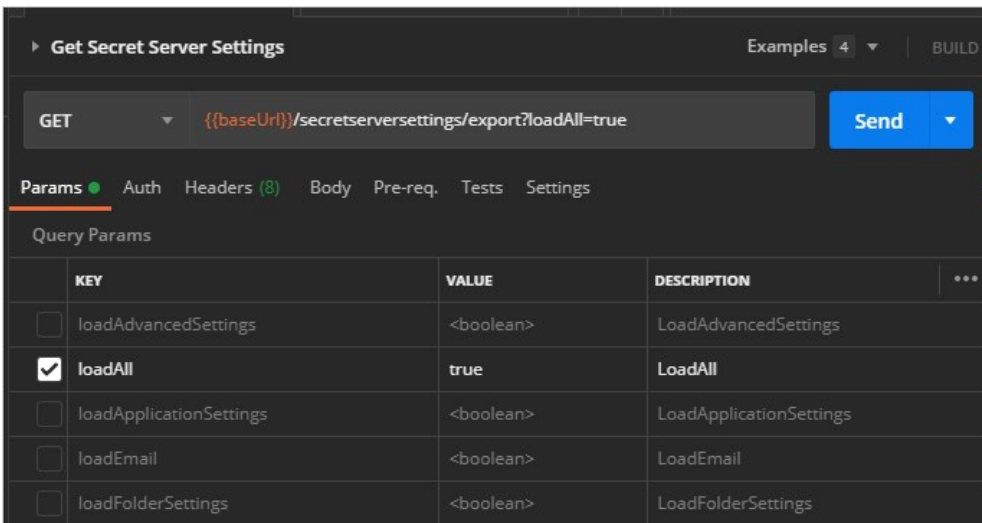
data object. Then manually update the desired filter category load to true to import.

API Calls Filter

Secret Server has settings import/export endpoints for the API to manipulate. Opening Postman and going to **secretserversettings > export > GET Get Secret Server Settings**, you would see:



Looking at the query parameters for that endpoint, we see:



The keys are equivalent to those on the user interface or those in the JSON file.

The `loadAll` key tells Secret Server to update all the available settings. These include application settings, launcher settings, protocol handler settings, permission options, user experience settings, and user interface settings.

If you click the **Body** tab below, you can see what JSON code represents the key you chose for the export:

The screenshot shows a REST client interface for a GET request. The URL is `{{baseUrl}}/secretserversettings/export?loadAll=true`. The response status is 200 OK, with a response time of 3.96s and a size of 59.27 KB. The response body is displayed in JSON format, showing configuration details for the secret server.

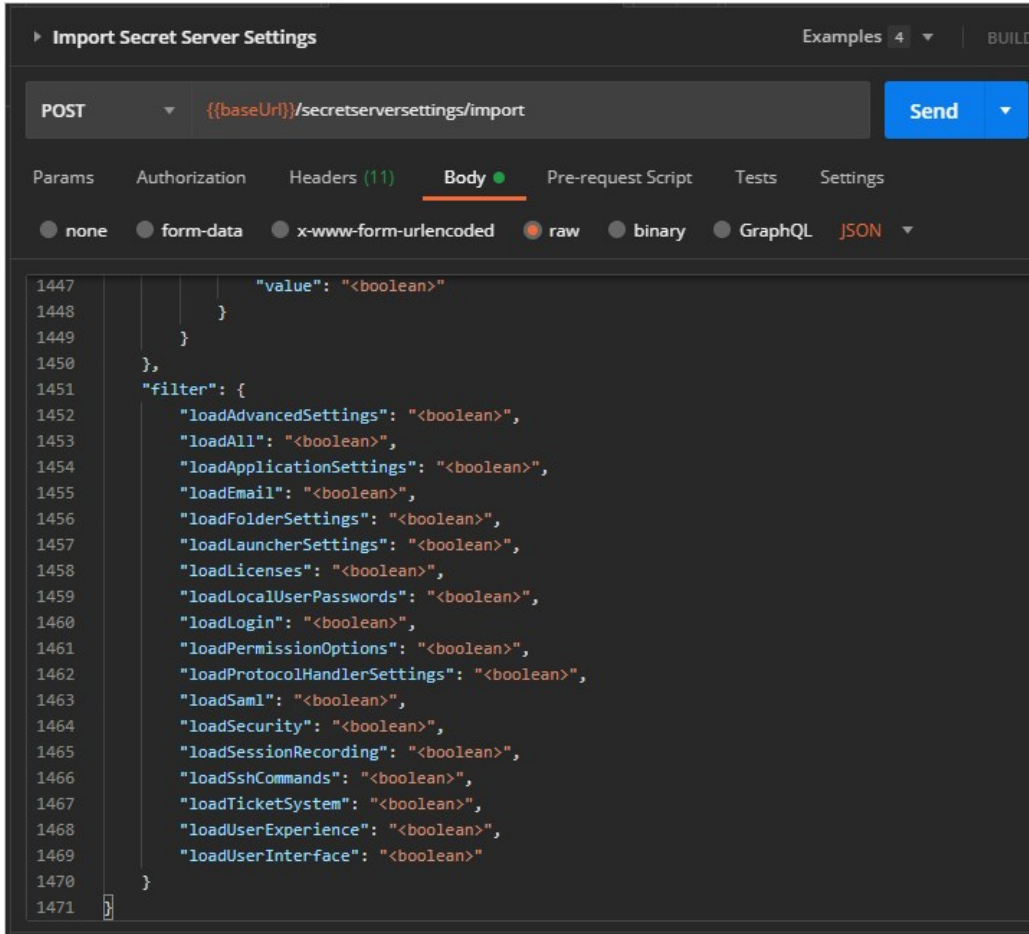
Param	Value	Type	Label
<input type="checkbox"/> loadAdvancedSettings		<boolean>	LoadAdvancedSettings
<input checked="" type="checkbox"/> loadAll	true		LoadAll

```

1  {
2    "externalInstanceId": "bf05836e-e992-41ce-85f9-0699fc643661",
3    "configurationVersion": "1.0.0",
4    "applicationSettings": {
5      "allowSoftwareUpdateChecks": false,
6      "configurationEarlyAdopterEnabled": false,
7      "allowSendTelemetry": true,
8      "enableWebServices": true,
9      "mobileMaxOfflineDays": 30,
10     "mobileMaxOfflineHours": 0,
11     "apiSessionTimeoutUnlimited": false,
12     "apiSessionTimeoutDays": 0,
13     "apiSessionTimeoutHours": 0,
14     "apiSessionTimeoutMinutes": 20,
15     "apiRefreshTokensEnabled": true,
16     "maximumTokenRefreshesAllowed": 3,
17     "preventApplicationFromSleeping": true,
18     "enableSyslogCefLogging": true,
19     "syslogCefServer": "127.0.0.1",
20     "syslogCefPort": 514,
21     "syslogCefProtocol": 1,
22     "syslogCefTimeZone": 2,
23     "syslogCefLogSite": 1,
24     "writeSyslogToEventLog": false,
25     "winRmEndpointUrl": "http://THY-01-0079-LT.testparent.thycotic.com:5985/wsman",
26     "enableCredSsp": false,
27     "maxSecretLogLength": 100000,
28     "customUrl": "",
29     "tmsInstallationPath": null,

```

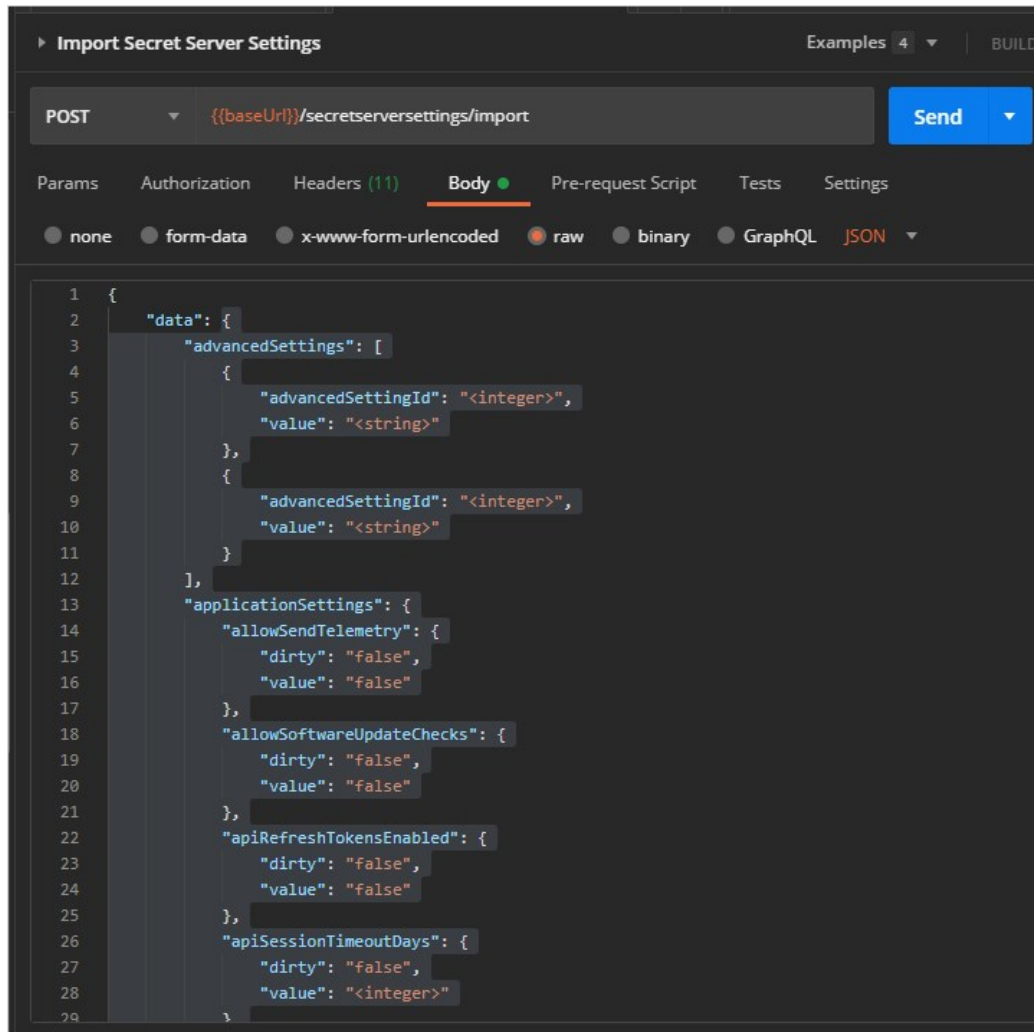
When using the POST Import Secret Server Settings command, you will see a filter object at the bottom of the code stipulating what to update:



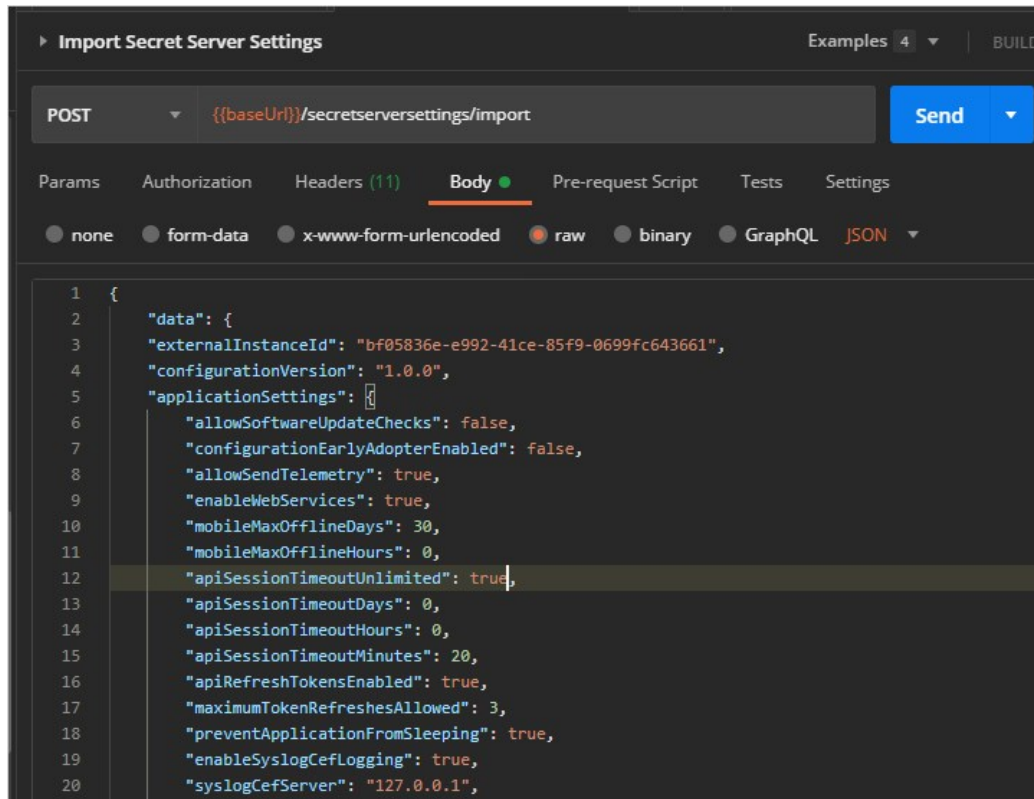
For example, if you set `loadApplicationSettings` to `true`, only the application settings are updated, assuming the objects stipulated were sent with the request. Similarly, included objects that are disallowed by the filter are ignored.

To make a GET call to update a single setting:

1. Import the category it belongs to. For example, if you want to update `apiSessionTimeoutUnlimited` to `true`, you would copy the entire `applicationSettings` result (the category and all of its settings).
2. For the POST Import Secret Server Settings call, remove the settings in the data section, leaving the filter section as is:



3. Paste the settings you copied earlier in its place.
4. Change the `apiSessionTimeoutUnlimited` setting to `true`:



5. Scroll down to the filter section and remove the filters you do not want to update. Alternatively, you can replace all the `<boolean>` settings with `false` for the filters you do not want.

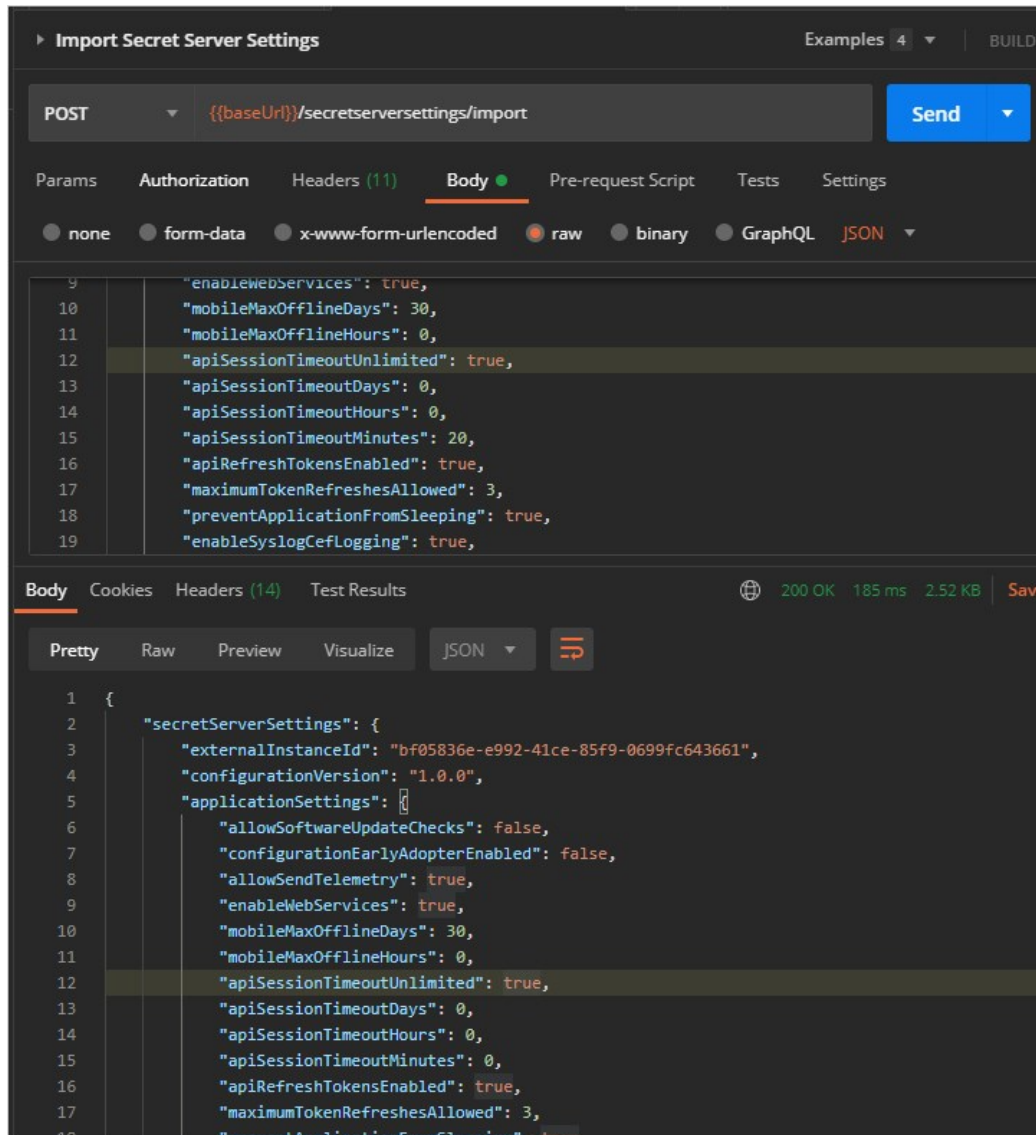
6. If you want to set a nullable field back to null, set the dirty flag and the value to null. For example:

```

"siteId": {
  "dirty": "true",
  "value": null
}

```

7. Click the **Send** button. If all goes well, Postman will return the updated category object:



Note: In this example, we copied the whole data object, but you do not have to. For a quick update, you can Import with just the settings you want to update. Anything not sent is ignored. This is the reason a nullable setting has to be explicitly set to null, along with setting the dirty flag—everything set to null is ignored, the same as if you did not send the setting at all.

8. If something went wrong, you will see an error section at the bottom of the results:



Audits

An audit is recorded for each setting category that was exported or imported by user. The individual setting audits can be viewed on the Configuration Audit page.

Figure: Audits on the Export / Import page:

DATE RECORDED	USER	ACTION	NOTES
02/18/2021 9:51 am	admin	SECRET SERVER SETTINGS IMPORT	Ticket System - Completed with errors. Please check the System Log for details.
02/18/2021 9:41 am	admin	SECRET SERVER SETTINGS EXPORT	Advanced Settings, Application Settings, Permission Options, Launcher Settings, Protocol Handl...
02/18/2021 9:13 am	user	SECRET SERVER SETTINGS IMPORT	Application Settings, SAML, Ticket System - Completed with errors. Please check the System Lo...
02/18/2021 9:11 am	user	SECRET SERVER SETTINGS IMPORT	Ticket System - Completed with errors. Please check the System Log for details.

If there are errors, a system log entry will also be saved with details:

- Email
- Login
- Security
- Ticket System
- SAML - Completed with errors

Events

When Secret Server settings are exported or imported, an SECRETSERVERSETTINGS event is logged.

Logs

When Secret Server settings are exported or imported or validation errors occur, a new log entry will appear in the ss.log file.

Note: <USERNAME> and <USERID> are replaced with your values. The items in the parentheses are the errant category settings.

System Logs or CEF Example

<USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import SAML for the following reason(s): TicketSystem=Only one ticket system can be default. (IsDefault);SAML=Identity Provider Id was not found in the database. Check that it was not modified after export. (IdentityProviderId)

SS.log Examples

- ERROR Delinea.Logging.ILogWriter - <USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import SAML for the following reason(s): TicketSystem=Only one ticket system can be default. (IsDefault);SAML=Identity Provider Id was not found in the database. Check that it was not modified after export. (IdentityProviderId) Secret Server.
- ERROR Thycotic.Logging.ILogWriter - <USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import some settings due to the following reason(s): Security=Access Denied;Login=Insufficient permissions to edit Radius settings. (Radius),Insufficient permissions to edit Delinea One or OpenId settings. (OpenIdConnect),Insufficient permissions to edit Duo settings. (Duo);TicketSystem=Only one ticket system can be default. (IsDefault);SAML=Access Denied

Errors and Resolutions

SAML=Access Denied	Need Administer Configuration SAML permission to update SAML settings.
SAML=Identity Provider Id was not found in the database. Check that it was not modified after export. (IdentityProviderId)	For SAML, the IdentityProviderId provided in the import file was not found in the database. If intending to add a new one, set this to 0.
TicketSystem=Only one ticket system can be default. (IsDefault)	For TicketSystem, IsDefault is set to true when there is already one set to true in the database. If intending to set it to true, set the other one to false.
Insufficient permissions to edit Radius settings. (Radius) Insufficient permissions to edit Duo settings. (Duo)	Need Administer Configuration Two Factor permission to update Radius or Duo settings.
Insufficient permissions to edit Delinea One or OpenId settings. (OpenIdConnect)	Need Administer OpenID Connect permission to update Open ID Connect settings.

What is Maintenance Mode?

Maintenance mode prevents users from changing secrets or secret-related data such as dependencies, secret templates, and password requirements.

Why do we need Maintenance Mode?

When secret key rotation takes place, or the HSM configuration is changed, Secret Server needs to ensure that no data corruption occurs. To mitigate this, these operations turn on maintenance mode, which puts Secret Server into read-only mode. We also recommend manually enabling maintenance mode before performing upgrades.

Can I still access my Secrets when Maintenance Mode is turned on?

Yes. Secrets will be read-only, but you can still view them, including secrets that are double-locked or protected by "require approval for access." You are unable to change the checkout status of a secret during maintenance mode. This means if the secret is currently checked-in, you will be unable to check it out. If the secret is currently checked out, it cannot be checked in until the system leaves maintenance mode.

How long does Maintenance Mode last?

Maintenance mode lasts until the operation triggering it is completed. The time required will vary based on the operation and the number of secrets in the system. Typically, maintenance mode lasts less than 30 minutes.

How do you enable and disable Maintenance Mode?

To enable and disable Maintenance Mode, see [Enabling and Disabling Maintenance Mode](#).

Important: This feature is part of the early release of Secret Server 10.11. The general release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version.

Overview

Object metadata allows you to store extended information on several Secret Server objects including users, groups, folders, dates, or secrets via the user interface or REST API. You can store most data types, including strings, Boolean values, numbers, dates, and users. You can combine this metadata into sections containing named fields of your defined types.

Unlike preexisting object fields, this metadata is flexible and dynamic. No coding, structural changes, or database schema changes are required. The only constraint is a role permission that controls who can add metadata fields or sections, which is granular down to the sections level on a given entity. For example, users that can view a user might be allowed to edit the values in the "public" metadata section and users that can edit a user might be allowed to edit all the sections for that user.

Features

Secret Server object metadata features:

- You can store user-defined metadata sections and field values on users, groups, folders, or secrets.
- Sections and fields are defined once and can be used across any applicable object. This allows for a common description across all objects. For example, all the objects could have metadata fields for business owner, source system, and corporate department name.
- Metadata fields are grouped or organized into sections.
- When viewing metadata, only populated fields appear. Field names with blank values are never present.
- You can define who can edit which sections via a role permission and by view or edit permissions the object.
- Each object maintains an audit history for all metadata fields, including previous values and who defined them.
- Audit history is viewable as a basic line chart for metadata fields stored as numbers. This provides a historical value table, as opposed to an audit log.

Example Use Cases

There are many ways to use Secret Server object metadata. You could:

- Define common attributes from a corporate directory that are not available on a standard user, such as manager, hire date, or department.
- Allow defined users to add data to an object without allowing that user to edit the object itself. For example, the user could not edit a secret but can add notations on the secret that are useful to others accessing the object.
- Store external system link identifiers for integrations. For example, the employee ID from the HR system could be stored in the metadata for users. Integration jobs could then query this ID from metadata and use it for synchronization.
- Add a "department owner" field on a folder to store which department owns it. For instance, the folder contains secrets for marketing as defined by the metadata field.
- Avoid users putting numerous items into the notes field on a secret, resulting in a disorganized mess. With metadata, those items could be stored in properly named fields. This organizes the notes and allows them to be easily searched without having to parse a block of text.

Adding Object Metadata

Note: This instruction is on a user object. The process for folders, groups, and secrets is very similar.

1. Go to **Admin > Users**. The User Management page appears:

Admin > User Management

Groups Users Audit

There are currently 101 enabled user(s) out of a total licensed 101 user(s).

Migrate to AD Create User

103 Items All Domains Include Disabled

USERNAME	NAME	EMAIL	ENABLED	DOMAIN	LAST LOGIN
admin	admin	admin@gamma.thy...	Yes		3 hours, 52 ...
AdminC	Alan Carol	Alan.Carol@gamma.thy...	Yes	gamma.thy...	
AdminL	Alan Linda	Alan.Linda@gamma.thy...	Yes	gamma.thy...	5 months, ...
AdminP	Alan Paul	Alan.Paul@gamma.thy...	Yes	gamma.thy...	5 months, ...
AdminD	Alan David	Alan.David@gamma.thy...	Yes	gamma.thy...	1 year, 6 m...
AdminA	Antonio	Antonio@gamma.thy...	Yes	gamma.thy...	

2. Click on the desired user. The user's page appears:

The screenshot shows the 'User Management > Users' page in the Delinea Admin console. The breadcrumb trail is 'Admin > User Management > Users > jdoe'. The 'General' tab is selected, with other tabs for 'Groups', 'Roles', 'Teams', 'Metadata', and 'Audit'. A search icon, a grid icon, a green plus icon, and a red 'WS' icon are in the top right. Below the tabs, there is an 'Options' dropdown menu. The main content area is titled 'User Details' with an 'Edit' link. A note states: 'Specific login and user detail information for a single user. [KB Link](#)'. The user details are as follows:

Username	jdoe
Display Name *	J. Doe
Domain	Local
Email	jdoe@domain.com
Application Account	No
Multifactor Authentication	< None >
Enabled	Yes
Locked Out	No
Restricted By Team	No

3. Click the **Metadata** tab.
4. Click the **Add Metadata** button. The Add Metadata popup appears:

Add Metadata

Section Name *

Cancel Save

5. Click the **Section Name** dropdown list and select **Add New Section**. Additional controls appear:

Add Metadata

Section Name *

New Section Name *

Section Description

Metadata Field *

Cancel Save

6. Type the section name in the **New Section Name** text box.
7. (Optional) Type a description of the field in the **Section Description** text box.
8. Click the **Metadata Field** dropdown list and select **Add New Field**. Still more controls appear:

Add Metadata

Section Name *

Metadata Field *

Metadata Field Name *

Field Type *

9. Type the field's name in the **Metadata Field Name** text box.
10. Click the **Field Type** dropdown list and select the desired data type. We chose Boolean.
11. (Optional) Click to select the **Value** check box if you want the field to be prepopulated to true. This only applies to the Boolean data type.

Note: Boolean fields always appear later because they always have a value.

12. Click the **Save** button. The new section and metadata field appears:

Brains [Edit](#)
Add Metadata

	Big Brain	false	Edit
--	-----------	-------	----------------------

The section appears in the top-left (*Brains*), and the field appears in the bottom right (*Big Brain*). We now have true or false field denoting if the user has a big brain.

13. Click the **Add Metadata** button again to add another field to the section. The Add Metadata popup returns.
14. This time, click the **Section Name** dropdown list and select the section that you just created.
15. Click the **Metadata Field** dropdown list box and select **Add New Field**. More controls appear:
16. Type the name of the new field in the **Metadata Field Name** text box. We chose *Brain Last Used*.
17. Click the **Field Type** dropdown list to select Date / Time. Date and time text boxes appear.
18. Add a date and time. You must add a value—otherwise there is no point in adding the field—blank fields are invisible.
19. Click the **Save** button. The new field appears on the Metadata tab:

Brain Last Used 3/4/2021 12:00 am

[Edit](#)

20. Add additional fields as desired.

21. The **Brains** section, **Big Brain** Boolean value, and **Brain Last Used** date / time fields are now available for use across all Secret Server users.

Best Practices

How your organization uses object metadata requires some forethought, including:

- Will you allow anyone to add metadata or only a specific set of individuals? This is controlled by applying the above mentioned role.
- How do you want to standardize the naming of sections and fields? One user might call the same field *business owner* and another might call it *subject expert* if you do not establish the field nomenclature up front.
- Do you want to create a "public" field section that is available to all users to edit, even those with read only permission on the object?

Overview

Secret Server can push its secrets to DevOps Secret Vault by creating a secret based on the "DevOps Secret Vault Client Credentials" template, which holds the client credentials for a DevOps Secret Vault tenant. Using the REST API, you can then register a DevOps Secret Vault tenant in Secret Server. That tenant references that secret to push secrets to DevOps Secret Vault at a set sync interval.

Behavior Test

You can manually push secrets to the DSV tenant, in addition to Secret Server checking for secrets to push to tenants on a timer. Secret Server will check for if a tenant needs updating every 30 minutes on the cloud or 10 minutes for an on-premises installation. Users are prevented from setting a tenant's sync interval to less than Secret Server's timed iteration because there would be no benefit to doing so. When Secret Server checks for secrets to be pushed to DSV, it only pushes secrets that have been changed since the last time they were updated in DSV. When a secret is pushed to DSV, its sync time is updated.

Fields

All secret fields are copied to DSV except for fields that are marked as "Hide On View." The notes field of a secret maps to the secret description in DSV. Files are Base64 Encoded, then sent to DSV. They are stored as encoded, and need decoding for use.

Setup in Secret Server

To configure pushing secrets to DSV:

1. Create a client in DSV. Save the client ID and secret that are generated when you created it. A DSV client is a container for a password.

Note: Please see the DSV documentation for details.

2. Create a secret to connect to DSV:

1. [Create a new secret](#) based on the DevOps Secrets Vault Client Credentials template:

Create New Secret

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Secret Template DevOps Secrets Vault Client Credentials [Change](#)

Folder Personal Folders [Clear](#)

Secret Name *

Client ID *

Client Secret * [Show](#) [Generate](#)

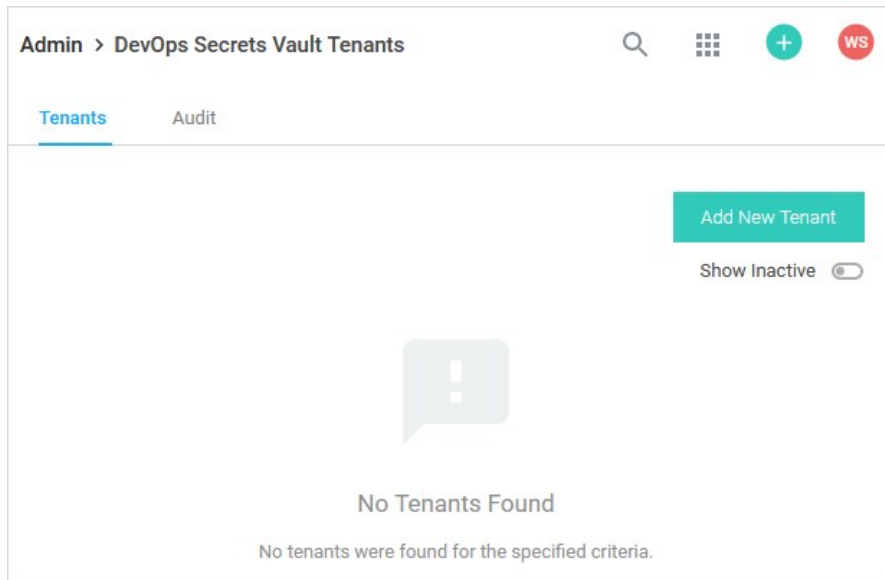
Tenant *

Notes

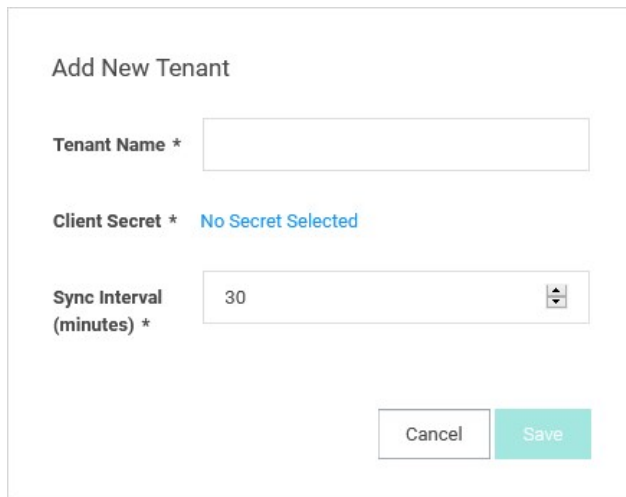
Cancel
Create Secret

2. Type the name for the new secret in the **Secret Name** text box.
3. Type the DSV client ID in the **Client ID** text box.
4. Type the DSV password for authentication in the **Client Secret** text box. If you do not have one, you can create a new here by clicking the **Generate** button. Then, create or configure a client in DSV using the password.
5. Type the DSV tenant to connect to in the **Tenant** text box. A DSV tenant is your DSV cloud account and the rights to access it. Use the format: `https://<tenantname>.secretsvaultcloud.<region>` with the region being one of the following:
 - U.S. region: com
 - E.U. region: eu
 - APAC region: au
6. Click the **Site** dropdown list to select your Secret Server site.
7. Click the **Create Secret** button.

3. Go to **Admin** > **See All**. The Admin Menu page appears.
4. Click the **DevOps Secrets Vault** link. The DevOps Secrets Vault Tenants page appears:



5. Click the **Add New Tenant** button. The Add New Tenant popup appears:

A screenshot of the 'Add New Tenant' popup form. The form has a title 'Add New Tenant' at the top. Below the title are three fields: 'Tenant Name *' with an empty text input box; 'Client Secret *' with a blue link text 'No Secret Selected'; and 'Sync Interval (minutes) *' with a dropdown menu showing '30'. At the bottom of the form are two buttons: 'Cancel' and 'Save'.

6. Type a descriptive name for the tenant in the **Tenant Name** text box. This can be anything you wish.
7. Click the **Client Secret** link to select the secret you created earlier in this instruction.
8. Click the **Sync Interval** list box to select how often you want Secret Server to push secrets to DSV for this tenant.
9. Click the **Save** button.

API Examples

Creating a DevOps Secret Vault Tenant

Use a POST to `/api/v1/devops-secrets-vault/tenant` using the body below to create a tenant in Secret Server.

```
{
  "Data": {
    "secretId": { "value": 79, "dirty": true },
    "tenantName": { "value": "LJDevTenant", "dirty": true },
    "syncInterval": { "value": 60, "dirty": true },
    "active": { "value": true, "dirty": true }
  }
}
```

The secret ID is the client ID for the secret based on the DSV Client Credentials template. The Sync Interval is how often Secret Server checks if secrets needs to be pushed to DSV. Only secrets associated with active tenants are pushed to DSV. You are returned the tenant ID if the POST is successful.

Creating a Sync Map

Use a POST to `/api/v1/devops-secrets-vault/add-sync` using this body to map a secret to a DSV tenant:

```
{
  "data": {
    "secretId": {
      "dirty": true,
      "value": 60
    },
    "dsvTenantId": {
      "dirty": true,
      "value": 1
    },
    "active": {
      "dirty": true,
      "value": true
    },
    "fieldNamesPath": {
      "dirty": true,
      "value": [
        "Demo","\$domain","qagreentest"
      ]
    }
  }
}
```

When the secret is mapped to a tenant, an initial sync immediately occurs. Following the initial sync, the secret is checked to determine if updates have been made when the sync Interval expires (making it "dirty") for the mapped tenant. If no changes have been made to the secret, then the secret is not pushed to DSV. You can reference fields from the secret to create the path in DSV. Secret Server will look for a \$, then search for the following string as the [field slug names](#) for the secret's template. The path in DSV follows this format:

```
/secrets/<DSV_secret_name>.
```

Manually Syncing a Secret

Use a POST to `/api/v1/devops-secrets-vault/sync` to manually trigger a push to DSV for existing sync maps. The list of integers contains the SyncMapIds of the secret to tenant mapping, so you can control which secret is pushed to which tenant.

```
{
  "data": [
    3, 4, 5
  ]
}
```

Listing DevOps Secret Vault Tenants

List DSV tenants registered to Secret Server by running a GET to `/api/v1/devops-secrets-vault/tenant`. Query parameters accepted:

- filter.nameSearch=
- filter.includeInactive=

Getting a DevOps Secret Vault Tenant's Details

View the details of a single tenant by specifying a tenant ID in a GET to `/api/v1/devops-secrets-vault/tenant/{tenantId}`.

Getting the Status of a Secret's Synchronization

View a secret's sync status by running a GET to `/api/v1/devops-secrets-vault/sync/status/{syncMapId}`.

Getting a List of Secret Synchronization Statuses

View a list of secret sync statuses by running a GET to `/api/v1/devops-secrets-vault/sync/status`. Query parameters accepted:

- filter.secretId=
- filter.includeInactive=
- filter.tenantId=

APIs and Scripting

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: This section contains general material on API access and scripting. APIs, scripts and techniques may appear elsewhere in the documentation for specific aspects of Secret Server. For example, Automatic Secret Export REST API is in the Secret Import and Export section.

Overview

Secret Server relies on [Windows Remote Management](#) (WinRM) components to run PowerShell scripts. This requires configuration on the Secret Server Web server and any distributed engines. By default, Secret Server uses `http://localhost:5985/wsman` as the WinRM endpoint. The endpoint URI can be seen under **Admin > Configuration** or **Admin > Distributed Engine > Manage Sites > Local** if using distributed engines. At the moment, we only support running PowerShell scripts on localhost. If you are new to PowerShell Remoting please review [PowerShell Remoting Security Considerations](#).

Configuration (Domain Joined)

Run PowerShell as an Administrator and execute:

Enable-PSRemoting

See [Enable-PSRemoting](#) for more information.

This command performs the following steps:

- Runs the [Set-WSManQuickConfig](#) cmdlet, which performs the following tasks:
 - Starts the WinRM service
 - Sets the startup type on the WinRM service to Automatic
 - Creates a listener to accept requests on any IP address
 - Enables a firewall exception for WS-Management communications
 - Registers the Microsoft.PowerShell and Microsoft.PowerShell.Workflow session configurations, if it they are not already registered
 - Registers the Microsoft.PowerShell32 session configuration on 64-bit computers, if it is not already registered
 - Enables all session configurations
 - Changes the security descriptor of all session configurations to allow remote access.
- Restarts the WinRM service to make the preceding changes effective.

Verifying Listeners

Confirm the listener:

```
Get-WSManInstance -ResourceURI winrm/config/listener -SelectorSet @{Address="*";Transport="http"}
```

The output should reflect the newly configured listener with **Enabled : true**

Additional Considerations

- By default two BUILTIN groups are allowed to use PowerShell Remoting as of v4.0: Administrators and Remote Management Users. (`Get-PSSessionConfiguration -Name Microsoft.PowerShell).Permission`)
- Sessions are launched by Secret Server under the user's context, which means all the same security controls and policies apply within the session.
- See [Investigating PowerShell Attacks](#) by FireEye for additional security considerations.
- Your environment may already be configured for WinRM. If your server is already configured for WinRM but it is not using the default configuration, you can change the URI to use a custom port or URLPrefix.

Configuration (Standalone)

By default WinRM uses Kerberos for authentication. Since Kerberos is not available on machines that are not joined to the domain, HTTPS is

required for secured transport of the password. Only use this method if you are going to run scripts from a Secret Server Web server or distributed engine that is **not** joined to the domain.

Note: WinRM HTTPS requires a local computer "Server Authentication" certificate with a CN matching the hostname that is not expired, revoked, or self-signed. A certificate would need to be installed on each endpoint Secret Server or the engine manages.

Create the new listener:

```
New-WSManInstance - ResourceURI winrm/config/Listener -SelectorSet @{Transport=HTTPS} -ValueSet @{{Hostname="HOST";CertificateThumbprint="XXXXXXXXXX"}}
```

Overview

You can use PowerShell scripts in Secret Server to automate specific tasks. These scripts are useful in several places in Secret Server, such as in creating custom remote password changers, custom dependency changers, discovery scanners, and custom ticket system integration.

Creating a PowerShell Script

1. Develop your script. See:

- [REST API PowerShell Script Examples](#)
- [Using Secret Fields in Scripts](#)
- [Dependency Token List](#)


Important: Do not edit the script with Windows Notepad. Instead, use Notepad++, Visual Studio Code, or Atom. Windows Notepad can add invisible characters that can cause issues.

Note: Using PowerShell in Secret Server involves passing a parameter string to the script. This string can contain literal values as well as tokens that represent values on the object in Secret Server that the script is attached to. For example, when creating a custom password changer, you pass in values such as the user name, old password, and new password using tokens that represent these values for whichever secret is running the password change script. Similarly, dependencies have a set of tokens that represent values on the dependency and its associated secret.

















































2. Go to **Admin > Scripts**.

Scripts


PowerShell SQL SSH

 Script testing will execute from the Web Server you are connected to but scripts may run on other nodes outside of testing

[+ Create New](#) [Information on PowerShell Scripts](#)

NAME	DESCRIPTION	CATEGORY	ACTIVE	USAGE COUNT	
Create OU Placeholder	Discovery	Discovery Scanner	Yes	0	   
Find Users in AD	Users	Discovery Scanner	Yes	0	   
Find Host Ranges	HostRange	Discovery Scanner	Yes	0	   
Find Machines	Machine	Discovery Scanner	Yes	0	   
Find Local Accounts	local Acc	Discovery Scanner	Yes	0	   
Find Local Accounts - with custom field names	Local Accounn	Discovery Scanner	Yes	0	   
Script to find AD Users who have administrator rights on a computer	AD users	Discovery Scanner	Yes	0	   
Find Dependencies	Depen	Discovery Scanner	Yes	0	   
Create a Domain Controller in a OU	Control Domain	Discovery Scanner	Yes	0	   
Find Users in OU	OU	Discovery Scanner	Yes	1	   
Service Account Dummy Code	Service Account Dummy Code	Discovery Scanner	Yes	0	   
Task Account Dummy	Task Account Dummy	Discovery Scanner	Yes	0	   

Show Inactive

Filter by category: 

Note: In some older versions of Secret Server, you can find the **Scripts** option under **Remote Password Changing** on the **Administration** menu bar.

3. On the **PowerShell** tab, click **+Create New**. The New PowerShell Script popup appears:

New PowerShell Script

Name *

Description *

Category Untyped ▾

Script

1

✓ OK ✕ Cancel

4. Type the script name in the **Name** text box.
5. Type a description in the **Description** text box.
6. Click the **Category** dropdown list to select the type of script. This will determine where the script resides in Secret Server and more. For instance, the Dependency choice ensures that Remote Password Changing is turned on by enabling it on the Remote Password Changing page.
7. Paste your script into the **Script** text box.
8. Click the **OK** button. The new script appears in the table on the Scripts page.

Best Practices

Debugging

Scripts may contain debug lines to help you test the script. Debug statements use the Write-Debug command. For example: Write-Debug "The users name is \$Username"

Script Arguments

There are input boxes for specifying arguments in places where the PowerShell scripts are used. Argument values are specified on a single line separated by a space. Values containing spaces should be enclosed in quotes. Parameters to PowerShell scripts are referenced through the zero-based `Args` array. It is often beneficial to assign `Args` variables to other more-meaningful named variables. For example:

Arguments: "Welcome back" and "Joe"

Script:

```
$greeting = $Args[0]
```

```
$name = $Args[1]
Write-Debug "$greeting $name"
```

Output: "Welcome back Joe"

Exceptions

In situations where the script should fail, given a specific set of conditions, an exception should be explicitly thrown. When an exception is thrown, the script stops running and the failure is logged in the system log. The script is considered to have successfully run if no errors or exceptions occur while processing. For example:

```
if ($meetsCondition -eq $false)
{
    throw "Did not meet condition"
}
```

You can use SQL scripts in Secret Server to automate specific tasks. You can configure a SQL script as a dependency of a secret and run after the password is successfully changed on the secret.

Creating a SQL Script

1. From the Administration Menu, click **Scripts**.
2. Click the **SQL** tab on the Scripts page.
3. Click the **Create New** button.
4. Type the name, description, and script in the dialog box.
5. Click the **OK** button.
6. Using SQL scripts as dependencies requires that Remote Password Changing is turned on, so ensure that this is enabled on the Remote Password Changing page.

Using Parameters

Where SQL scripts are used, we provide an arguments text box. It is often beneficial to assign variables to other more meaningful variables.

Examples

SQL

```
UPDATE TABLE cmsuser  
SET password = PWDCRYPT(@Password) WHERE username = @Username;
```

MySQL

```
UPDATE TABLE cmsuser  
SET password = PASSWORD(?) WHERE username = ?;
```

PostgreSQL

```
UPDATE TABLE cmsuser  
SET passwd = CRYPT(?, GEN_SALT('sha256')) WHERE username = ?;
```

ODBC

```
UPDATE TABLE cmsuser  
SET passwd = $Password WHERE username = $Username;
```

Basic Oracle PL/SQL

```
INSERT INTO cmuser values (:username, :password);
```

Advanced Oracle PL/SQL

```
EXECUTE IMMEDIATE 'alter user || :username || identified by "" || :password || ""';
```

Returning Errors

In situations where the script should fail given specific conditions, the script should explicitly throw an exception. When an exception is thrown, the script stops running and the failure is recorded in the system Log. The script is considered to have successfully run if no errors or exceptions occur while processing.

Examples

SQL

```
RAISERROR(N'ERROR: %s', 14, 1, N'Failure');
```

MySQL

```
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT = 'ERROR: Failure';
```

PostgreSQL

```
DO language plpgsql $$  
BEGIN  
RAISE EXCEPTION 'ERROR (14)';  
END  
$$;
```

ODBC

```
RAISERROR(N'ERROR: %s', 14, 1, N'Failure');
```

SQL Example

An issue that occurs in SQL Server database environments is when a linked database is set up with a credential and that credential's password changes. To counter this, you can set up a SQL script to run as a dependency after the password change occurs to drop and recreate the link. Note that you may need to edit the option depending on the desired linked server configuration.

Example

SQL

```
EXEC master.dbo.sp_dropserver @server=@MACHINE, @droplogins='droplogins'  
EXEC master.dbo.sp_addlinkedserver @server = @MACHINE, @srvproduct='SQL Server'  
EXEC master.dbo.sp_addlinkedsrvlogin @rmtsrsvname=@MACHINE, @useself=N'True', @locallogin=NULL, @rmtuser=NULL, @rmtpassword=NULL  
EXEC master.dbo.sp_addlinkedsrvlogin  
@rmtsrsvname=@MACHINE, @useself=N'False', @locallogin=@LOCALUSERNAME, @rmtuser=@REMOTEUSERNAME, @rmtpassword=@PASSWORD
```

Parameters

- @MACHINE The machine or instance of the server where the linked database exists. For instance, SERVER\SQL2014.
- @LOCALUSERNAME The local login on the server where the linked database is configured.

- @REMOTEUSERNAME The username that is set in the linked database's security info for connecting to the linked database. This should be the username of the secret that the dependency is on.
- @PASSWORD This will be the new password after the SQL account's password is changed.

SSH scripts can be used in Secret Server to automate specific tasks. An SSH script can be configured as a dependency of a secret and run after the password is successfully changed on the secret.

Creating an SSH Script

1. From the **Admin** menu, click **Scripts**.
2. Click the **SSH** tab on the Scripts page.
3. Click the **Create New** button.
4. Type the name and description in their text boxes.
5. Paste the script in the text box.
6. Click the **OK** button.

Using SSH scripts as dependencies requires that Remote Password Changing is turned on, so ensure that this is enabled on the Remote Password Changing page.

Scripts return all output results, which allows you to debug your script using the echo command (or any command that will write to standard output).

For example:

```
echo The users name is $USERNAME
```

An input box for specifying arguments will exist in places where the SSH scripts are used. Argument values are specified next to their associated argument name. Parameters to SSH scripts are referenced through named variables. It is often beneficial to assign parameter variables to other more meaningful variables.

Another example:

```
$GREETING = Welcome back
```

```
$NAME = Joe
```

```
echo $GREETING $NAME
```

Output: Welcome back, Joe

In situations where the script should fail given a specific set of conditions, exit the script with a non-zero status code. When scripts exit with a non-zero status, the failure is logged in the System Log. The script is considered to have successfully run if it does not exit with a non-zero status code.

Example:

```
if [ ! $MEETS_CONDITION -eq 0 ]; then
```

```
echo Failure
```

```
exit 1;
```

```
fi
```

```
echo Success
```

Use Case

Update a user's password in the Apache user configuration file when it is changed in Secret Server:

```
cd $FILEPATH;
rm -f tomcat-users.xml.bak;
sed -i.bak 's/username="$USERNAME".*password="[^\"]*/username="$USERNAME" password="$NEWPASSWORD"/' tomcat-users.xml;
if grep -q $NEWPASSWORD "tomcat-users.xml"; then
exit 0;
fi
exit 5;
```

Adding an SSH Script as a Dependency

1. Navigate to **Admin > Scripts**.
2. Select the **SSH** tab.
3. Click **Create New** to create a script.
4. Fill out the required fields **Name**, **Description**, and **Script**.
5. (Optional) Change the setting for the environmental line ending in case your environment experiences issues with line termination.
6. (Optional) Specify the port that the script will connect to.

When filling out the script there is some important information to remember. If you would like to use one of the Secret Server variables such as \$USERNAME, \$NEWPASSWORD, or \$CURRENTPASSWORD, you need to specify that they are parameters. For example, take the following script from our use case:

Edit SSH Script

Name: Update Tomcat Config *

Description: Updates the user's password in Tomcat. *

Script:

```
1 cd $FILEPATH;
2 rm -f tomcat-users.xml.bak;
3 sed -i.bak 's/username="$USERNAME".*password="[^\"]*/username="$USERNAME" password='
4
5
6 if grep -q $NEWPASSWORD "tomcat-users.xml"; then
7   exit 0;
8 fi
9 exit 5;
```

Settings: New Line (n) Port

Params:

USERNAME	Literal	✕
NEWPASSWORD	Literal	✕
Add New...	Interpreted	+

OK Cancel

Within the script, parameters are specified with the dollar sign prefixed to the name. In the list of parameters, you simply specify their name without the dollar sign. This is so you can fill their values in later when implementing the script as a secret dependency.

When adding the SSH Script as a dependency to a secret, specify the built-in Secret Server variables with the dollar-sign prefix. For example, using our use case:

New Dependency

Explain

Type: SSH Script

Script: Update Tomcat Config *

Name: Change Tomcat Password *

Server Name: centos.example.org *

Port: 22 (default)

Server Key Digest:

Description:

Wait (s): 0 *

Enabled:

Privileged Account: [No Secret Selected](#)

SSH Key Secret: [No Secret Selected](#)

Parameters

USERNAME: \$USERNAME

NEWPASSWORD: \$PASSWORD

If a parameter in the script is not specified in the list of parameters, then they will use environmental variables on the remote host.

Overview

A list of all valid tokens that can be used in PowerShell, SSH, and SQL dependency script arguments.

PowerShell, SSH, and SQL dependencies can have script arguments that derive their values from values on the dependency, the secret it belongs to, or any other secrets associated for remote password changing. Starting with Secret Server 10.0, tokens can also be used in ODBC connection string arguments. Script arguments are defined on dependency changers in Secret Server 10.0 and above and on the dependency in earlier versions of Secret Server. The following table lists the tokens available to dependency scripts.

Best Practices

Wrap Fields in Quotation Marks

When specifying PowerShell arguments, the `$fieldname` argument is replaced with the information in the field, and *then* the whole string is passed to PowerShell. Thus, if the string returned has spaces, PowerShell parses the spaces as separate arguments, which causes errors.

Therefore, we recommend you wrap your fields in two quotation marks. For example: `"$fieldname"`, which is then parsed by PowerShell as one string.

Avoid Using Spaces in Field Names

We recommend not using spaces within your field names. In many cases, that would not cause a problem, but if the first word in the field name is also a reserved word, it will cause an error because the reserved word is parsed separately from the rest of the field name.

For example, a field named:

```
$machine name
```

Because `$machine` is a reserved word, the parser would separate the reserved `$machine` value, followed by `name` as a separate argument.

Note: Surrounding the field name in quotation marks would *not* fix this—Secret Server makes the substitution prior to PowerShell parsing the string.

Token List

Table: Dependency Tokens

<code>\([x]\)</code>	pre-10.0	The value of the field with the same name on the xth secret on the RPC tab for use in custom password changing commands and scripts.
<code>\${x}\$PASSWORD</code>	pre-10.0	The password on the xth secret on the RPC tab for use in custom password changing commands and scripts.
<code>\${x}\$USERNAME</code>	pre-10.0	The username on the xth secret on the RPC tab for use in custom password changing commands and scripts. Example: <code>\${1}\$USERNAME</code> is the username of the first associated secret.
<code>\$</code>	pre-10.0	The value of the field on the secret with the same name. Examples: <code>\$DOMAIN</code> matches the secret's "Domain" field, and <code>\$NOTES</code> matches the "Notes" field.

\$	10.0	The name of any scan item field (defined on the scan item template) that is visible in the dependency edit dialog. If a scan item field is derived from a parent field, you may also use the parent field name as a parameter that translates to this field's value.
\$CURRENTPASSWORD	pre-10.0	The password currently on the secret (context-sensitive to whether script is run before or after password change).
\$CURRENTPUBLICKEY	10.2	The public key currently on the secret (context-sensitive to whether script is run before or after key rotation).
\$DATABASE	pre-10.0	The value of the Database field from the dependency. Only valid for SQL dependencies unless added as a field by the scan item template (see below).
\$DEPENDENCYPRIVILEGEDPASSPHRASE	10.3	The private key passphrase on the privileged account assigned to the dependency.
\$DEPENDENCYPRIVILEGEDPASSWORD	10.3	The password on the privileged account assigned to the dependency.
\$DEPENDENCYPRIVILEGEDPRIVATEKEY	10.3	The private key on the privileged account assigned to the dependency.
\$DEPENDENCYPRIVILEGEDUSERNAME	10.3	The user name on the privileged account assigned to the dependency.
\$DEPENDENCYSSHKEY	10.3	The new SSH key to set on the dependency.
\$DEPENDENCYSSHKEYPASSPHRASE	10.3	The new passphrase of the SSH key to be set on the dependency.
\$MACHINE	10.0	The value of the Machine Name field on the dependency. This is always the second part of the dependency title (<service name> on <machine>).
\$NEWPASSWORD	10.2	This is not available to the dependency at the time of the password change. It will no longer be set as the NewPassword gets set to the Password.
\$NEWPUBLICKEY	10.2	The new public key that is being set on the secret.
\$PASSPHRASE	10.2	The passphrase used to encrypt the private key in a public/private key pair on this secret.
\$PASSWORD	pre-10.0	The password on the secret.
\$PORT	pre-10.0	The value of the Port field from the dependency. Only valid for SQL and SSH dependencies unless added as a field by the scan item template (see below).
\$PRIORPASSPHRASE	10.2	The passphrase that was set on the secret before the current passphrase rotation.
\$PRIORPASSWORD	10.2	In the context of a password change, this is the password at the beginning of the password change. The password that was set on the secret before the current password change. In the context of run dependency from the Dependency tab, this is the prior value of the password before the last password change.

\$PRIORPUBLICKEY	10.2	The public key that was set on the secret before the current key rotation.
\$PUBLICKEY	10.2	The public key on the secret.
\$SERVICENAME	10.0	The value of the Service Name field on the dependency. Service Name may have a different name based on the dependency type, but it is always the first part of the dependency title (<service name> on <machine>).
\$USERNAME	pre-10.0	The username on the secret.

Note: The mappings of these parameters come from the remote password changer, so you can check the mapping by editing the secret template and selecting configure password changing.

Note: Some of these tokens, such as \$PASSWORD, \$CURRENTPASSWORD, \$NEWPASSWORD, and \$PRIORPASSWORD, may seem to duplicate each other, but they have distinctions based on the context as described above.

Note: In some cases, whether or not the dependency is being changed locally or through a distributed engine may have an impact on what these tokens return. This is due to the asynchronous nature of distributed engines. The newer tokens, such as \$NEWPASSWORD and \$PRIORPASSWORD, were created to address this issue. If you are using older tokens, such as \$PASSWORD and \$CURRENTPASSWORD, and seeing unexpected results, try using \$PRIORPASSWORD and \$NEWPASSWORD.

Overview

The Secret Server SDK replaces and improves upon the existing functionality of the Java API and .NET/Application API. Users can leverage this SDK to tokenize credentials in scripts and configuration files for .NET web applications. The SDK can also call for a REST Web services authentication token for added functionality. Finally, the SDK has a local encrypted cache for every location it is installed in to allow for quicker transit times and resiliency in case communication with Secret Server is lost.

To download the SDK command line tool, choose a version and OS platform below.

See the [Secret Server Software Development Kit for DevOps](#) topic for how to use the SDK.

SDK Client version 1.5.3

Release Notes

Added a "multi" option to the CLI. You can now select one field from multiple secrets in a single call by passing in a comma-separated list of multiple Secret IDs and a field name.

Downloads

- [Windows x64](#)
- [MacOS x64](#)
- [Linux x64](#) (including RHEL 7+)
- [Red Hat Enterprise Linux 6 x64](#)

Download File Hashes

Windows x64

- File: secretserver-sdk-1.5.3-win-x64.zip
- SHA256: 231F88106CC03434AC7E107579BB1B7686B10011F208068913F5F3762A4A5B64

MacOS x64

- File: secretserver-sdk-1.5.3-osx-x64.zip
- SHA256: E13657A9E80235F2404F398254E537185619DC921546C03D28BDF17C75E1694

Linux x64

- File: secretserver-sdk-1.5.3-linux-x64.zip
- SHA256: 8AFAB562C35EDC4AFD5EB6B6275635A06C319FD0275318B5DE6ED61D8678FCC4

Red Hat Enterprise Linux 6 x64

- File: secretserver-sdk-1.5.3-rhel.6-x64.zip
- SHA256: E4E83E632FFA640A47C70A880E53DBDE38F12F2CAF6FE4441D18ECEB7D716904

Legacy Releases

SDK Client version 1.5.0

Release Notes

- Cache improvements
- Stability improvements

Downloads

- [Windows x64](#)
- [MacOS x64](#)
- [Linux x64](#) (including RHEL 7+)
- [Red Hat Enterprise Linux 6 x64](#)

Download File Hashes

Windows x64

- File: secretserver-sdk-1.5.0-win-x64.zip
- SHA256: 4B3246470E4EA87190CE3B511151E93E04E7363EADBD145BA0EE8AA63BC1378B

MacOS x64

- File: secretserver-sdk-1.5.0-osx-x64.zip
- SHA256: 492D177CF86554EC22B947957328D833471E2DECFA7AE95FC0D2B3FFB1B24E37

Linux x64

- File: secretserver-sdk-1.5.0-linux-x64.zip
- SHA256: 94672CA26C438301A070C020FFAEE2932250F3358C91B30BC4F9B8F0E0A1210B

Red Hat Enterprise Linux 6 x64

- File: secretserver-sdk-1.5.0-rhel.6-x64.zip
- SHA256: BF0BA52C7BA0838E56C4C0C44FC29794223B04336458D342F14872FE3914E30B

SDK Client version 1.4.1

Release Notes

- New option to specify a configuration and cache directory other than the default
- SDK CLI (tss) now has a --configure-directory option.
- Nuget SDK packages now check for a SecretServerSdkKeyDirectory AppSetting.
- Updated the Nuget SDK packages to support customizing the key storage directory. They now check for a SecretServerSdkKeyDirectory AppSetting.
- Updated the Nuget SDK packages to have a SecretServerClient GetAccessToken() and GetAccessTokenAsync() method, to get a REST or SOAP API token. This is equivalent to the existing SDK CLI command token.

Downloads

- [Windows x64](#)
- [MacOS x64](#)
- [Linux x64](#) (including RHEL 7+)
- [Red Hat Enterprise Linux 6 x64](#)

SDK Client version 1.3.0

Release Notes

- Upgrade to .NET Core 2.1 runtime
- Encrypt configuration with DPAPI on Windows by default
- New commands to display connection status and version
- New option to specify a key storage directory other than the default
- Target multiple OS versions with a single build
- New build for Red Hat Enterprise Linux 6

Downloads

- [Windows x64](#)
- [MacOS x64](#)
- [Linux x64](#) (including RHEL 7+)
- [Red Hat Enterprise Linux 6 x64](#)

SDK Client version 1.0.0

Release Notes

Initial release of the SDK command line tool

Downloads

- [Windows 10 x64](#)
- [CentOS 7 x64](#)
- [Red Hat Enterprise Linux 7 x64](#)
- [Ubuntu 16.10 x64](#)
- [MacOS 10.12 x64](#)

SDK NuGet Packages (Optional)

To get the SDK NuGet packages, check out the documentation on our [GitHub page](#). This package is not required to use the SDK Client.

Note: The SDK Client and SDK NuGet packages are two separate projects with separate versioning. We actively work to maintain feature parity between the two projects, but at times their features may differ.

Secret Server contains an OpenAPI Swagger specification file that describes the REST API endpoints. There are several Swagger-based tools available to generate clients. This document describes using some popular ones to generate clients for different languages including C#, Java, and PowerShell. The generated client facilitates calling the Secret Server REST API and indicates API changes with new Secret Server versions.

Note: Please use the article quick links on the right to jump to the section you are interested in.

Generating Clients

C# Client Using NSwagStudio

1. Download and install NSwagStudio (<https://github.com/RicoSuter/NSwag/wiki/NSwagStudio>)
2. Copy and paste the `swagger.json` file for "Documentation for token authentication." This is located at: {Your SecretServer Base Url}\Documents\restapi\OAuth\swagger.json
3. Suggested settings:
 - o Namespace: SecretServerAuthentication
 - o Inject HttpClient via constructor
 - o Generate default values for properties
4. Generate output.
5. Copy output into a C# file in solution.
6. Copy and paste the `swagger.json` for "Documentation for REST API using bearer token authentication." It is located at {Your SecretServer Base Url}\Documents\restapi\TokenAuth\swagger.json.
7. Suggested Settings:
 - o Namespace: SecretServerRestClient
 - o Inject HttpClient via constructor
 - o Generate default values for properties
8. Generate output.
9. Copy output into a C# file in solution

See [Authenticating](#) for creating a token to use in the examples below.

```
// Authenticate:
var httpClient = new HttpClient();
var token = "<TOKEN>";

// Set credentials (token):
httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token);

// Call REST API:
var client = new SecretServerRestClient.SecretsServiceClient("https://secretserver.url.local/ss/api/v1", httpClient);
var search = client.SearchAsync(sortBy0_name: "lastHeartBeatStatus", sortBy0_direction:"asc");
var results = search.Result;
```

C# or .NET Core Client Using OpenAPI Generator

Note: These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. Future versions may fix issues that necessitated some workarounds. If you are using a newer version, you may need to make adjustments.

Note: These instructions assume that you have Java and .NET Core installed on your machine. They should work on all systems, but the syntax will need to be tweaked for other shells.

1. Follow the steps in [Getting OpenAPI Generator](#).
2. Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

```
$oauthSwagger = 'OAuth/swagger.json' $tokenAuthSwagger = 'TokenAuth/swagger.json'
```

3. Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the .NET Core generator are described on [Config Options for csharp-netcore](#) page.

```
java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g csharp-netcore --skip-validate-spec --package-name SecretServerOAuth --remove-operation-id-prefix --additional-properties=netCoreProjectFile=true -o oauth-csharp java -jar openapi-generator-cli.jar generate -i $tokenAuthSwagger -g csharp-netcore --skip-validate-spec --package-name SecretServerTokenAuth --remove-operation-id-prefix --additional-properties=netCoreProjectFile=true -o tokenauth-csharp
```

4. Build the packages in release mode:

```
dotnet build -c Release oauth-csharp/src/SecretServerOAuth dotnet publish -c Release tokenauth-csharp/src/SecretServerTokenAuth
```

5. Add the necessary `.dll` files to your project. The DLLs are located at the paths below:

```
oauth-csharp/src/SecretServerOAuth/bin/Release/netstandard2.0/SecretServerOAuth.dll tokenauth-csharp/src/SecretServerTokenAuth/bin/Release/netstandard2.0/publish/*.dll
```

6. Test the API. For example:

See [Authenticating](#) for creating a token to use in the examples below.

```
var token = "<TOKEN>";  
// Do not use tokenAuthConfig.AccessToken  
// This method can be called repeatedly as the access token expires, old values will be overwritten  
tokenAuthConfig.AddApiKey("Authorization", $"{token.TokenType} {token.AccessToken}");  
  
var foldersApi = new FoldersApi(tokenAuthConfig);  
var folder = await foldersApi.GetAsync(11);  
  
Console.WriteLine(folder);  
Console.ReadKey();
```

Java Client Using OpenAPI Generator

Note: These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. If you are using a newer version, you may need to make adjustments.

Note: These instructions assume that you have Java configured on your machine. JDK 8 or greater is required. These instructions should work on all systems, but the syntax will need to be tweaked for other shells.

1. Follow the steps in [Getting OpenAPI Generator](#).
2. Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

```
$oauthSwagger = 'OAuth/swagger.json' $tokenAuthSwagger = 'TokenAuth/swagger.json'
```

3. Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the Java generator are described on the [Config Options for java](#) page.

```
java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g java --skip-validate-spec --invoker-package secretserver.oauth.client --api-package secretserver.oauth.api --model-package secretserver.oauth.model --group-id secretserver --artifact-id secretserver.oauth --remove-operation-id-prefix -o oauth-java java -jar openapi-generator-cli.jar generate -i $tokenAuthSwagger -g java --skip-validate-spec --invoker-package secretserver.tokenauth.client --api-package
```

```
secretserver.tokenauth.api --model-package secretserver.tokenauth.model --group-id secretserver --artifact-id secretserver.tokenauth --remove-operation-id-prefix -o tokenauth-java
```

- Follow the instructions in `oauth-java/README.md` to include the package in your build process. Alternatively, follow the steps below to generate JAR files and test.

```
cd oauth-java mvn clean install -DskipTests cd ../tokenauth-java mvn clean install -DskipTests
```

- Include the following JARs in your project (replace version numbers as necessary):

```
oauth-java/target/secretserver.oauth-10.9.9.jar tokenauth-java/target/secretserver.tokenauth-10.9.9.jar tokenauth-java/target/lib/*.jar
```

- Test the JAR files:

See [Authenticating](#) for creating a token to use in the examples below.

```
// set to the root of the Secret Server instance with no trailing slash
String basePath = "https://thycotic.com/SecretServer";
```

```
secretserver.oauth.client.ApiClient oauthClient = secretserver.oauth.client.Configuration.getDefaultApiClient();
secretserver.tokenauth.client.ApiClient tokenAuthClient = secretserver.tokenauth.client.Configuration.getDefaultApiClient();
```

```
String token = "<TOKEN>";
```

```
tokenAuthClient.setBasePath(basePath + "/api");
tokenAuthClient.setApiKey(tokenResponse.getTokenType() + " " + token);
FoldersApi foldersApi = new FoldersApi(tokenAuthClient);
FolderModel folderModel = foldersApi.get(11, false, false);
System.out.println(folderModel);
```

PowerShell Using OpenAPI Generator

Note: These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. If you are using a newer version, you may need to make adjustments. The PowerShell generator is in beta.

- Follow the steps in [Getting OpenAPI Generator](#).
- Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

```
$oauthSwagger = 'OAuth/swagger.json' $tokenAuthSwagger = 'TokenAuth/swagger.json'
```

- Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the PowerShell generator are described on [Config Options for powershell](#) page.

Note: These commands set `disallowAdditionalPropertiesIfNotPresent` to `false` to work around [a bug in the generator](#).

```
java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g powershell --skip-validate-spec --package-name SecretServerOAuth --remove-operation-id-prefix -o oauth-ps java -jar openapi-generator-cli.jar generate -i $tokenAuthSwagger -g powershell --skip-validate-spec --package-name SecretServerTokenAuth --remove-operation-id-prefix -o tokenauth-ps
```

- Perform a find/replace to work around a bug in the generator:

```
Get-ChildItem ".\tokenauth-ps\src\SecretServerTokenAuth\Model\*.ps1" | ForEach-Object { (Get-Content $_) | ForEach-Object { $_ -replace "$AllProperties = \()", "$AllProperties = @()" } | Set-Content $_ }
```

- Run the following commands to build the PowerShell modules. `-Verbose` can be specified if desired.

```
.\oauth-ps\Build.ps1 .\tokenauth-ps\Build.ps1
```

- Import both modules:

```
Import-Module .\oauth-ps\src\SecretServerOAuth Import-Module .\tokenauth-ps\src\SecretServerTokenAuth
```

7. Set the base URL of each module. Only edit the first line below. The value must be the full URL to the root of Secret Server **with no trailing slash**, for example `https://thycotic.com/SecretServer`.

```
$baseUrl = "https://thycotic.com/SecretServer" SecretServerOAuth\Set-Configuration -BaseUrl $baseUrl SecretServerTokenAuth\Set-Configuration -BaseUrl $baseUrl
```

8. Store a token - See [Authenticating](#) for creating a token.

9. Set the TokenAuth module to use the token: `SecretServerTokenAuth\Set-Configuration -DefaultHeaders @{ Authorization = $token.token_type + " " + $token.access_token }`

10. Test the folder service Get endpoint (replace the ID as necessary):

```
Get-FolderDetail -id 11
```

Postman Workspace

[Postman](#) is a REST client with a GUI to easily test API requests. To automatically generate a workspace with Secret Server API calls, click the **Import** button and choose whether to use a local file or a URL. Ensure the **Generate a Postman Collection** check box is selected. The program appears to be doing nothing while importing, but will inform you of success shortly.

Insomnia Workspace

[Insomnia](#) is a REST client similar to Postman. To automatically generate a workspace with Secret Server API calls, first click the arrow to open the dropdown, then click **Import/Export**. Click **Import Data**, then **From File** or **From URL** as desired.

Getting OpenAPI Generator

Note: These client-generation instructions were written with OpenAPI Generator version 4.3.0, which was the latest version at the time. Future versions may fix issues that necessitated some workarounds. If you are using a newer version, you may need to make adjustments.

OpenAPI Generator supports several installation methods, described on their [CLI Installation](#) page. You can pick the method most suitable for your environment. The commands in this article use the JAR method for simplicity, but other installation methods use mostly the same command format so only small tweaking should be required.

Note: All methods require the Java runtime (version 8+).

Self-Signed or Other Invalid Certificates

If you use an SSL certificate that is self-signed or otherwise not technically valid, OpenAPI Generator throws an error if you try to use a URL to `swagger.json` instead of a local file (or when using the Java client). To fix this, you need to import the certificate into Java's certificate store. The following example commands are for Windows, but the same concept applies to Mac and Linux as well.

Note: The default password for the cacerts keystore is `changeit`.

Java 8 stores the cacerts keystore in `JAVA_HOME` unless it was explicitly changed. `keytool` should also be on your `PATH` if `java` itself is.

```
keytool -import -keystore "%JAVA_HOME%\jre\lib\security\cacerts" -file "path\to\cert.pem"
```

Java 11 provides a flag on `keytool` to find the cacerts keystore automatically.

```
keytool -import -cacerts -file "path\to\cert.pem"
```

Swagger 2.0 Notes

Some features in our REST API are not currently supported in Swagger 2.0. For example, the sort is an array of a complex objects. The actual API will accept multiple sort levels by passing this query string, `paging.sortBy[0].name=secretname&paging.sortBy[1].name=folderid`. The auto-generated

client only allows you to specify the first sort by default. To specify multiple sorts, the serialization needs to be customized. We assumed that multiple sorting levels is probably an advanced feature and or choice will work for most.

Secret Server web services can be called using scripts. This example demonstrates how to authenticate and retrieve a Secret programmatically in Perl. If connecting to an Installed instance, change the URL to match your site and specify the application name if configured.

This example runs using standard Perl modules and Strawberry Perl v5.24.1.1.

You may need to install REST::Client from CPAN which can be done by running:

```
cpanm REST::Client
```

This module makes interacting with REST-ful APIs much easier.

The TSS script below leverages the API created in the TSS with .pm extension file.

The TokenResponse with .pm extension file defines an object that stores the authentication credentials for the session.

The ErrorResponse with .pm extension file defines an object that stores error information should a REST request fail.

The SecretModel with .pm extension file defines an object that stores the Secret information when requesting a particular Secret from the REST API.

TSS.pl:

```
use REST::Client;
use JSON;

use TSS;

# Enter your Secret Server web site here.
my $host = 'http://localhost';

# Enter the application name, should it be defined.
my $applicationPath = '/secretserver';

my $false = 0;
my $true = 1;
my $usingSelfSignedCertificate = $true;

if ($usingSelfSignedCertificate) {
    # Disable hostname validation since this will fail with self-singed certs.
    $ENV{'PERL_LWP_SSL_VERIFY_HOSTNAME'} = $false;
}

# Initialize the TSS Perl API with the connection information provided above.
my $tss = TSS->new(
    secretServerUrl => $host,
    applicationPath => $applicationPath
) ;

print "Username: ";
my $input = <STDIN>;
chomp $input;
my $username = $input;

print "Password: ";
$input = <STDIN>;
chomp $input;
my $password = $input;

# Authenticate to Secret Server
my $response = $tss->getToken($username, $password);
if (ref $response eq 'ErrorResponse') {
    $tss->handleError($response);
    exit();
}

# Persist the API token in our TSS API session.
```



```
$tss->tokenResponse($response);

$password = undef;

print "Secret ID to retrieve: ";
$input = <STDIN>;
chomp $input;
my $secretId = $input;

# Retrieve a Secret from Secret Server by Secret ID.
$response = $tss->getSecret($secretId);
if (ref $response eq 'ErrorResponse') {
    $tss->handleError($response);
    exit();
}

# Read the SecretModel object
print 'Secret ' . $response->id . ' retrieved successfully. Secret name: ' . $response->name . "\r\n";
```

TSS.pm:

```
package TSS;

use REST::Client;
use JSON;

use TokenResponse;
use ErrorResponse;
use SecretModel;

use strict;

sub new {
    my($class, %args) = @_ ;

    my $self = {
        client => REST::Client->new(),
        secretServerUrl => undef,
        applicationPath => undef,
        tokenResponse => undef,
        %args
    };
    bless $self, $class;

    $self->client->setHost($self->secretServerUrl);

    return($self);
}

sub client {
    my $self = shift;
    return $self->{client};
}

sub secretServerUrl {
    my $self = shift;
    return $self->{secretServerUrl};
}

sub applicationPath {
    my $self = shift;
    return $self->{applicationPath};
}

# Getter and setter so that API tokens can be reused through a session.
sub tokenResponse {
    my $self = shift;
    my $tokenResponse = shift;

    if ($tokenResponse ne undef) {
        $self->{tokenResponse} = $tokenResponse;
        return $tokenResponse;
    }
}
```

```
return $self->{tokenResponse};
}

# Crafts a HTTP header for providing authentication to Secret Server.
sub getHeaders {
my $self = shift;

my $headers = {
  Accept => 'application/json',
  Authorization => 'Bearer ' . $self->tokenResponse->access_token
} ;

return $headers;
}

sub handleError {
my $self = shift;
my $errorResponse = shift;

print 'ERROR!' . $errorResponse->message . ' (Error Code: ' . $errorResponse->errorCode . ') . "\r\n";
}

# Authenticates to Secret Server by POST'ing a Form URL Encoded content body containing sensitive credentials.
sub getToken {
my $self = shift;
my $username = shift;
my $password = shift;

my $postData = $self->client->buildQuery([
  username => $username,
  password => $password,
  grant_type => "password"
] );

my $response = $self->client->POST($self->applicationPath . '/oauth2/token', substr($postData, 1), {'Content-Type' => 'application/x-www-form-urlencoded'});

if ($self->client->responseCode() ne 200) {
my $errorResponse = ErrorResponse->new(
  errorCode => $self->client->responseCode(),
  message => "Unable to get token."
);

return $errorResponse;
}

my $responseJson = from_json($response->responseContent);

if (exists $responseJson->{'errorCode'}) {
my $errorResponse = ErrorResponse->new(
  errorCode => $responseJson->{'errorCode'},
  message => $responseJson->{'message'}
);

return $errorResponse;
}

my $tokenResponse = TokenResponse->new(
  access_token => $responseJson->{'access_token'},
  token_type => $responseJson->{'token_type'},
  expires_in => $responseJson->{'expires_in'}
);

return $tokenResponse;
}

# Gets a Secret from Secret Server by GET'ing a specified Secret by ID and passing an authentication header containing our API token.
sub getSecret {
my $self = shift;
my @params = @_;
my $secretId = $params[0];

my $headers = $self->getHeaders();

$self->client->GET($self->applicationPath . '/api/v1/secrets/' . $secretId, $headers);
my $responseJson = from_json($self->client->responseContent());
```

```

if (exists $responseJson->{'errorCode'}) {
    my $errorResponse = ErrorResponse->new(
        errorCode => $responseJson->{'errorCode'},
        message => $responseJson->{'message'}
    ) ;

    return $errorResponse;
}

my $secretModel = SecretModel->new(
    id => $responseJson->{'id'},
    name => $responseJson->{'name'},
    secretTemplateId => $responseJson->{'secretTemplateId'},
    folderId => $responseJson->{'folderId'},
    active => $responseJson->{'active'},
    items => $responseJson->{'items'},
    launcherConnectAsSecretId => $responseJson->{'launcherConnectAsSecretId'},
    checkOutMinutesRemaining => $responseJson->{'checkOutMinutesRemaining'},
    checkedOut => $responseJson->{'checkedOut'},
    checkOutUserDisplayName => $responseJson->{'checkOutUserDisplayName'},
    checkOutUserId => $responseJson->{'checkOutUserId'},
    isRestricted => $responseJson->{'isRestricted'},
    isOutOfSync => $responseJson->{'isOutOfSync'},
    outOfSyncReason => $responseJson->{'outOfSyncReason'},
    autoChangeEnabled => $responseJson->{'autoChangeEnabled'},
    autoChangeNextPassword => $responseJson->{'autoChangeNextPassword'},
    requiresApprovalForAccess => $responseJson->{'requiresApprovalForAccess'},
    requiresComment => $responseJson->{'requiresComment'},
    checkOutEnabled => $responseJson->{'checkOutEnabled'},
    checkOutIntervalMinutes => $responseJson->{'checkOutIntervalMinutes'},
    checkOutChangePasswordEnabled => $responseJson->{'checkOutChangePasswordEnabled'},
    proxyEnabled => $responseJson->{'proxyEnabled'},
    sessionRecordingEnabled => $responseJson->{'sessionRecordingEnabled'},
    restrictSshCommands => $responseJson->{'restrictSshCommands'},
    allowOwnersUnrestrictedSshCommands => $responseJson->{'allowOwnersUnrestrictedSshCommands'},
    isDoubleLock => $responseJson->{'isDoubleLock'},
    doubleLockId => $responseJson->{'doubleLockId'},
    enableInheritPermissions => $responseJson->{'enableInheritPermissions'},
    passwordTypeWebScriptId => $responseJson->{'passwordTypeWebScriptId'},
    siteId => $responseJson->{'siteId'},
    enableInheritSecretPolicy => $responseJson->{'enableInheritSecretPolicy'},
    secretPolicyId => $responseJson->{'secretPolicyId'},
    lastHeartBeatStatus => $responseJson->{'lastHeartBeatStatus'},
    lastHeartBeatCheck => $responseJson->{'lastHeartBeatCheck'},
    failedPasswordChangeAttempts => $responseJson->{'failedPasswordChangeAttempts'},
    lastPasswordChangeAttempt => $responseJson->{'lastPasswordChangeAttempt'},
    secretTemplateName => $responseJson->{'secretTemplateName'},
    responseCodes => $responseJson->{'responseCodes'}
) ;

return $secretModel;
}

1;

```

TokenResponse.pm:

```

package TokenResponse;

use warnings;
use strict;

sub new {
    my $class = shift;
    my %options = @_;

    my $self = {
        access_token => undef,
        token_type => undef,
        expires_in => undef,
        %options
    } ;
    bless $self, $class;
}

```

```
    return $self;
}

sub access_token {
    my $self = shift;
    return $self->{access_token};
}

sub token_type {
    my $self = shift;
    return $self->{token_type};
}

sub expires_in {
    my $self = shift;
    return $self->{expires_in};
}

1;
```

ErrorResponse.pm:

```
package ErrorResponse;

use warnings;
use strict;

sub new {
    my $class = shift;
    my %options = @_;

    my $self = {
        errorCode => undef,
        message => undef,
        %options
    };
    bless $self, $class;

    return $self;
}

sub errorCode {
    my $self = shift;
    return $self->{errorCode};
}

sub message {
    my $self = shift;
    return $self->{message};
}

1;
```

SecretModel.pm:

```
package SecretModel;

use warnings;
use strict;

sub new {
    my $class = shift;
    my %options = @_;

    my $self = {
        id => undef,
        name => undef,
        secretTemplateId => undef,
        folderId => undef,
        active => undef,
        items => undef,
        launcherConnectAsSecretId => undef,
    };
    bless $self, $class;

    return $self;
}

1;
```

```
checkOutMinutesRemaining => undef,
checkedOut => undef,
checkOutUserDisplayName => undef,
checkOutUserId => undef,
isRestricted => undef,
isOutOfSync => undef,
outOfSyncReason => undef,
autoChangeEnabled => undef,
autoChangeNextPassword => undef,
requiresApprovalForAccess => undef,
requiresComment => undef,
checkOutEnabled => undef,
checkOutIntervalMinutes => undef,
checkOutChangePasswordEnabled => undef,
proxyEnabled => undef,
sessionRecordingEnabled => undef,
restrictSshCommands => undef,
allowOwnersUnrestrictedSshCommands => undef,
isDoubleLock => undef,
doubleLockId => undef,
enableInheritPermissions => undef,
passwordTypeWebScriptId => undef,
siteId => undef,
enableInheritSecretPolicy => undef,
secretPolicyId => undef,
lastHeartBeatStatus => undef,
lastHeartBeatCheck => undef,
failedPasswordChangeAttempts => undef,
lastPasswordChangeAttempt => undef,
secretTemplateName => undef,
responseCodes => undef,
% o p t i o n s
} ;
bless $self, $class;

return $self;
}

sub id {
    my $self = shift;
    return $self->{id};
}

sub name {
    my $self = shift;
    return $self->{name};
}

sub secretTemplateId {
    my $self = shift;
    return $self->{secretTemplateId};
}

sub folderId {
    my $self = shift;
    return $self->{folderId};
}

sub active {
    my $self = shift;
    return $self->{active};
}

sub items {
    my $self = shift;
    return $self->{items};
}

sub launcherConnectAsSecretId {
    my $self = shift;
    return $self->{launcherConnectAsSecretId};
}

sub checkOutMinutesRemaining {
    my $self = shift;
    return $self->{checkOutMinutesRemaining};
}
```

```
}

sub checkedOut {
  my $self = shift;
  return $self->{checkedOut};
}

sub checkOutUserDisplayName {
  my $self = shift;
  return $self->{checkOutUserDisplayName};
}

sub checkOutUserId {
  my $self = shift;
  return $self->{checkOutUserId};
}

sub isRestricted {
  my $self = shift;
  return $self->{isRestricted};
}

sub isOutOfSync {
  my $self = shift;
  return $self->{isOutOfSync};
}

sub outOfSyncReason {
  my $self = shift;
  return $self->{outOfSyncReason};
}

sub autoChangeEnabled {
  my $self = shift;
  return $self->{autoChangeEnabled};
}

sub autoChangeNextPassword {
  my $self = shift;
  return $self->{autoChangeNextPassword};
}

sub requiresApprovalForAccess {
  my $self = shift;
  return $self->{requiresApprovalForAccess};
}

sub requiresComment {
  my $self = shift;
  return $self->{requiresComment};
}

sub checkOutEnabled {
  my $self = shift;
  return $self->{checkOutEnabled};
}

sub checkOutIntervalMinutes {
  my $self = shift;
  return $self->{checkOutIntervalMinutes};
}

sub checkOutChangePasswordEnabled {
  my $self = shift;
  return $self->{checkOutChangePasswordEnabled};
}

sub proxyEnabled {
  my $self = shift;
  return $self->{proxyEnabled};
}

sub sessionRecordingEnabled {
  my $self = shift;
  return $self->{sessionRecordingEnabled};
}
```

```
sub restrictSshCommands {
  my $self = shift;
  return $self->{restrictSshCommands};
}

sub allowOwnersUnrestrictedSshCommands {
  my $self = shift;
  return $self->{allowOwnersUnrestrictedSshCommands};
}

sub isDoubleLock {
  my $self = shift;
  return $self->{isDoubleLock};
}

sub doubleLockId {
  my $self = shift;
  return $self->{doubleLockId};
}

sub enableInheritPermissions {
  my $self = shift;
  return $self->{enableInheritPermissions};
}

sub passwordTypeWebScriptId {
  my $self = shift;
  return $self->{passwordTypeWebScriptId};
}

sub siteId {
  my $self = shift;
  return $self->{siteId};
}

sub enableInheritSecretPolicy {
  my $self = shift;
  return $self->{enableInheritSecretPolicy};
}

sub secretPolicyId {
  my $self = shift;
  return $self->{secretPolicyId};
}

sub lastHeartBeatStatus {
  my $self = shift;
  return $self->{lastHeartBeatStatus};
}

sub lastHeartBeatCheck {
  my $self = shift;
  return $self->{lastHeartBeatCheck};
}

sub failedPasswordChangeAttempts {
  my $self = shift;
  return $self->{failedPasswordChangeAttempts};
}

sub lastPasswordChangeAttempt {
  my $self = shift;
  return $self->{lastPasswordChangeAttempt};
}

sub secretTemplateName {
  my $self = shift;
  return $self->{secretTemplateName};
}

sub responseCodes {
  my $self = shift;
  return $self->{responseCodes};
}

1;
```


Note: The REST API is available as of Secret Server 9.1.

Note: To use the REST API you first must enable Webservices. To do so, go to Admin > Configuration general tab. Enabling Webservices simply makes the ASP.NET REST Webservices built into Secret Server available.

Note: For a full reference of the REST endpoints and parameters, see the [REST API Guide](#).

Note: When using the API to search secrets, the account used must have at least "view" permissions on the full folder path to find the secret.

Note: Secret Server Cloud exclusively supports TLS 1.2. This version includes fixes for known vulnerabilities in older TLS versions and will eventually be required for PCI compliance. The following will need to be added to the top of your scripts because PowerShell defaults to TLS 1.0: [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

Note: Secret Server Cloud API users: If Delinea One is enabled, *and* an API user logging on has an email address set in Secret Server, the user *must* use the Delinea One password, instead of the local Secret Server password. Using a local account password will fail. Use one of these workarounds:

- Disable Delinea One, if not needed
- Use the Delinea One password instead
- Use a user account that does not have an email address in Secret Server

Authentication

See [Authenticating](#) for creating a token to use in the script examples below.

Searching Secrets

```
try
{
    $api = "https://<Secret Server URL>/api/v1"
    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    $filters = "?filter.HeartbeatStatus=1&filter.includeRestricted=true&filter.searchtext=< mySearchText >"

    Write-Host "-----"
    Write-Host "---- Secret Search Values ----"
    Write-Host "-----"

    #?filter.searchfield=username&filter.searchfield=displayname&filter.searchfield=filter.searchText=mister&filter.includeinactive=true" -Headers $headers

    $result = Invoke-RestMethod "$api/secrets/$filters" -Headers $headers

    Write-Host $result.filter.searchField
    Write-Host $result.total

    foreach($secret1 in $result.records)
    {
        Write-Host $secret1.id - "$secret1.name" - "$secret1.folderId - $secret1.lastHeartBeatStatus"
    }

    Write-Host "-----"
    Write-Host "---- Secret Lookup Values ----"
    Write-Host "-----"

    #?filter.searchfield=username&filter.searchfield=displayname&filter.searchfield=filter.searchText=mister&filter.includeinactive=true" -Headers $headers

    $result = Invoke-RestMethod "$api/secrets/lookup/$filters" -Headers $headers

    Write-Host $result.filter.searchField
```

```

Write-Host $result.total

foreach($secret in $result.records)
{
    Write-Host $secret.id" - "$secret.value
}
Write-Host "-----"
}
catch [System.Net.WebException]
{
    Write-Host "---- Exception ----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host $responseBody.errorCode " - " $responseBody.message
    foreach($ModelState in $responseBody.modelState)
    {
        $ModelState
    }
}
}

```

Creating a Secret

```

try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    #stub
    $templateId = 6001
    $secret = Invoke-RestMethod $api/secrets/stub?filter.secrettemplateid=$templateId -Headers $headers

    #modify
    $timestamp = Get-Date

    $secret.name = "$timestamp"
    $secret.secretTemplateId = $templateId
    $secret.AutoChangeEnabled = $true
    $secret.autoChangeNextPassword = "NextPA$$w0rd"
    $secret.SiteId = 1

    $secret.IsDoubleLock = $true

    foreach($item in $secret.items)
    {
        if($item.fieldName -eq "Domain")
        {
            $item.itemValue = "theDomain"
        }
        if($item.fieldName -eq "Username")
        {
            $item.itemValue = "myaccountname"
        }
        if($item.fieldName -eq "Password")
        {
            $item.itemValue = "!@#ssword1"
        }
        if($item.fieldName -eq "Notes")
        {
            $item.itemValue = "TheNotes"
        }
    }
}

$secretArgs = $secret | ConvertTo-Json

```

```
#create
Write-Host ""
Write-Host "-----Create secret -----"

$secret = Invoke-RestMethod $api"/secrets/" -Method Post -Body $secretArgs -Headers $headers -ContentType "application/json"

$secret1 = $secret | ConvertTo-Json
Write-Host $secret1
Write-Host $secret.id
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()

    Write-Host $responseBody
}
}
```

Editing a Secret

```
try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    #get
    $secretId = 45
    $secret = Invoke-RestMethod $api"/secrets/$secretId/" -Headers $headers

    #modify
    $timestamp = Get-Date

    $secret.Active = $true
    $secret.Name = "Updated Secret$timestamp.Ticks"
    $secret.folderId = 1101
    $secret.LauncherConnectAsSecretId = -1
    $secret.AutoChangeEnabled = $false
    $secret.RequiresComment = $false
    $secret.CheckOutEnabled = $false
    $secret.CheckOutIntervalMinutes = 17
    $secret.CheckOutChangePasswordEnabled = $true
    $secret.ProxyEnabled = $true
    $secret.SessionRecordingEnabled = $true
    $secret.PasswordTypeWebScriptId = -1
    $secret.SiteId = 1
    $secret.EnableInheritSecretPolicy = $false
    $secret.SecretPolicyId = -1
    $secret.AutoChangeNextPassword = "pass1232"

    $secretArgs = $secret | ConvertTo-Json

    #update
    Write-Host ""
    Write-Host "-----Update secret -----"

    $secret = Invoke-RestMethod $api"/secrets/$secretId/" -Method Put -Body $secretArgs -Headers $headers -ContentType "application/json"

    $secretUpdate = $secret | ConvertTo-Json
    Write-Host $secretUpdate
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
}
```

```

Write-Host $_.Exception
Write-Host $_.Exception.Response.StatusCode
Write-Host $_.Exception.Response.StatusDescription
$result = $_.Exception.Response.GetResponseStream()
$reader = New-Object System.IO.StreamReader($result)
$reader.BaseStream.Position = 0
$reader.DiscardBufferedData()
$responseBody = $reader.ReadToEnd()

Write-Host $responseBody
}

```

Checking in a Secret

```

try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    $secretId = 3803
    #SecretArgs are only required as needed for the REST endpoint. ForceCheckIn may be relevant if the secret has been checked out by another user since you will need to
    force checkin to use the secret. This removes the other user's checkin and rotates the password if the secret is configured to do so.
    $secretArgs = @{
        #DoubleLockPassword
        #TicketNumber
        #TicketSystemId
        #Comment ="Passing a comment"
        #ForceCheckIn = $false
        #ExposeFieldValues
        #IncludeInactive
    } ConvertTo-Json

    $secret = Invoke-RestMethod $api"/secrets/$secretId/check-in" -Method Post -Body $secretArgs -Headers $headers -ContentType "application/json"

    Write-Host $secret
    Write-Host $secret[0].message
    Write-Host $secret[0].code

    Write-Host "Name: " = $secret.name
    Write-Host "Active: " = $secret.active
    Write-Host "Template: " = $secret.secretTemplateName
    Write-Host "Heartbeat Status: " = $secret.lastHeartBeatStatus
    Write-Host "RPC" = $secret.lastPasswordChangeAttempt
    Write-Host "Checkout" = $secret.CheckedOut
    Write-Host "Checkout User" = $secret.CheckOutUserDisplayName
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()

    Write-Host $responseBody
}
}

```

Deleting a Secret

```

try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

```

```

$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("Authorization", "Bearer $token")

$secretId = 11231

Write-Host "----- Delete a Secret -----"

$timestamp = Get-Date

$deletemodel = Invoke-RestMethod "$api/secrets/$secretId" -Headers $headers -Method DELETE -ContentType "application/json"
Write-Host $deletemodel
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host $responseBody.errorCode " - " $responseBody.message
    foreach($ModelState in $responseBody.modelState)
    {
        $ModelState
    }
}
}

```

Creating a User

```

try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    # create user
    Write-Host ""
    Write-Host "----- Create a User -----"

    $timestamp = Get-Date
    $userCreateArgs = @{
        userName = "newuser$timestamp"
        password = "faW1fs@3S@"
        DisplayName = "new user$timestamp"
        enabled = $true
        emailAddress = "newuser$timestamp@v.com"
        twoFactor = $false
        radiusUserName = "aRadiusName"
        radiusTwoFactor = $false
        DomainId = 1
        isApplicationAccount = $true
    } | ConvertTo-Json

    $user = Invoke-RestMethod "$api/users" -Headers $headers -Method Post -ContentType "application/json" -Body $userCreateArgs
    Write-Host "New User ID : " $user.id
}
catch
{
    Write-Debug "----- Exception -----"
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host $responseBody.errorCode " - " $responseBody.message
    foreach($ModelState in $responseBody.modelState)

```

```
{
  $ModelState
}
```

Field Update, Get, File Upload, and Download

Note: This script requires Secret Server 10.1 or later.

```
try
{
  $site = "https://<Secret Server URL>"

  #secret id and field to test against
  $secretId = 283;
  $fieldToUpdate = "machine"

  #secretid and file to upload
  $fileSecretId = 283;
  $fileFieldToUpdate = "private-key"

  $token = "<TOKEN>"

  $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
  $headers.Add("Authorization", "Bearer $token")

  $endpoint = "$site/api/v1/secrets/$secretId/fields/$fieldToUpdate"

  $body = @{
    value = "newhost"
  } | ConvertTo-Json

  echo $endpoint
  echo -----

  echo -----
  echo "Updating Field $fieldToUpdate"

  $response = $null
  $response = Invoke-RestMethod -Method Put -Uri $endpoint -Headers $headers -ContentType "application/json" -Body $body
  echo $response;

  echo -----
  echo "Getting value from $fieldToUpdate"

  $response = $null
  $response = Invoke-RestMethod -Method Get -Uri $endpoint -Headers $headers
  echo $response;

  echo -----
  echo "Uploading file from $fileFieldToUpdate"

  $endpoint = "$site/api/v1/secrets/$fileSecretId/fields/$fileFieldToUpdate"
  echo $endpoint

  $secretArgs = @{
    fileName = "test.pem"
    fileAttachment = [IO.File]::ReadAllBytes("C:\temp\test.pem")
  } | ConvertTo-Json

  $response = $null
  $response = Invoke-RestMethod -Method Put -Uri $endpoint -Headers $headers -Body $secretArgs -ContentType "application/json"

  echo $response

  echo -----
  echo "Downloading file from $fileFieldToUpdate"
  echo $endpoint

  $response = $null
  $response = Invoke-RestMethod -Method Get -Uri $endpoint -Headers $headers #-OutFile "c:\temp\outfile2.pem"

  Write-Host $response.Length
  Write-Host $response
}
```

```

echo -----
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
}

```

Expiring a Token

Note: This script requires Secret Server 10.1 or later.

```

try
{
    $site = "https://<Secret Server URL>"
    $api = "$site/api/v1"

    $token = "<TOKEN>"

    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")

    foreach($user in $pagedUsers.records)
    {
        Write-Host $user.userName
    }

    # expire token
    Write-Host ""
    Write-Host "----- Expire Token -----"

    $expireToken = Invoke-RestMethod "$site/api/v1/oauth-expiration" -Headers $headers -Method Post

    # This part should fail with a 403 Forbidden
    Write-Host ""
    Write-Host "----- Expect an error -----"

    $secrets = Invoke-RestMethod "$site/api/v1/secrets" -Headers $headers
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host $_.Exception
    Write-Host $_.Exception.Response.StatusCode
    Write-Host $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
}

```

Managing Folders (Basic)

Note: This script requires Secret Server 10.1 or later.

```
$uri = "https://<Secret Server URL>"
$sapi = "$uri/api/v1"

$token = "<TOKEN>"

$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("Authorization", "Bearer $token")

#Get Folder Stub
$folderStub = Invoke-RestMethod $sapi"/folders/stub" -Method GET -Headers $headers -ContentType "application/json"

##### Add Folder #####

$timeStamp = Get-Date
$folderStub.folderName = "Rest Folder" + $timeStamp
$folderStub.folderTypeId = 1
$folderStub.inheritPermissions = $false
$folderStub.inheritSecretPolicy = $false

$folderArgs = $folderStub | ConvertTo-Json

$folderAddResult = Invoke-RestMethod $sapi"/folders" -Method POST -Body $folderArgs -Headers $headers -ContentType "application/json"
$folderId = $folderAddResult.id

if($folderId -gt 1)
{
    echo ""
    echo "-----"
    echo "--Add Folder Successful--"
    echo "-----"
    echo ""
    echo $folderAddResult | ConvertTo-Json
}
else
{
    Write-Error "ERROR: Failed to Add a folder."
    return
}

##### Delete Folder #####

#DeleteFolder is not implemented. Must be done through the UI.
echo ""
echo "-----"
echo "--Delete Folder Not Implemented--"
echo "-----"
echo ""

##### Get Folder #####

$folderGetResult = Invoke-RestMethod $sapi"/folders/$folderid" -Method GET -Headers $headers -ContentType "application/json"

if($folderGetResult.id -eq $folderId)
{
    echo ""
    echo "-----"
    echo "--Get Folder Successful--"
    echo "-----"
    echo ""
    echo $folderGetResult | ConvertTo-Json
}
else
{
    Write-Error "ERROR: Failed to Get a folder."
    return
}

##### Add Child Folder #####

$timeStamp = Get-Date
$folderStub.folderName = "Rest Child Folder" + $timeStamp
$folderStub.folderTypeId = 1
$folderStub.inheritPermissions = $false
$folderStub.inheritSecretPolicy = $false
$folderStub.parentFolderId = $folderGetResult.id
```



```
$folderArgs = $folderStub | ConvertTo-Json

$folderChildAddResult = Invoke-RestMethod $api"/folders" -Method POST -Body $folderArgs -Headers $headers -ContentType "application/json"
$childFolderId = $folderChildAddResult.id

if($childFolderId -gt 1)
{
    echo ""
    echo "-----"
    echo "--Add Child Folder Successful--"
    echo "-----"
    echo ""
    echo $folderChildAddResult | ConvertTo-Json
}
else
{
    Write-Error "ERROR: Failed to Add a Child folder."
    return
}

##### Update Folder #####

$childFolderId = $folderChildAddResult.id
$childFolderGetResult = Invoke-RestMethod $api"/folders/$childFolderId" -Method GET -Headers $headers -ContentType "application/json"

$foldername = "UpdatedFolderNAME"
$childFolderGetResult.folderName = $foldername
$childFolderGetResult.folderTypeId = 1
$childFolderGetResult.inheritPermissions = $true
$childFolderGetResult.inheritSecretPolicy = $false

$childFolderUpdateArgs = $childFolderGetResult | ConvertTo-Json

$childFolderUpdateResult = Invoke-RestMethod $api"/folders/$childFolderId" -Method PUT -Body $childFolderUpdateArgs -Headers $headers -ContentType
"application/json"

if($childFolderUpdateResult.folderName -eq $foldername)
{
    echo ""
    echo "-----"
    echo "--Update Folder Successful--"
    echo "-----"
    echo ""
    echo $childFolderUpdateResult | ConvertTo-Json
}
else
{
    Write-Error "ERROR: Failed to Update a folder."
    return
}

##### Search Folders #####

$searchFilter = "?filter.searchText=UpdatedFolderNAME"

$searchResults = Invoke-RestMethod $api"/folders$searchFilter" -Method GET -Headers $headers -ContentType "application/json"
$folder = $searchResults.records[0]
echo $searchResults
echo $folder

$name = "UpdatedFolderNAME"
if($searchResults.total -gt 0 -and $folder.folderName -eq $name)
{
    echo ""
    echo "-----"
    echo "--Search Folder Successful--"
    echo "-----"
    echo ""
    echo $group
}
else
{
    Write-Error "ERROR: Failed to Search Folders."
    return
}
```

```
##### Lookup Groups #####

$lookupFilter = "?filter.searchText=Updated"

$lookupResults = Invoke-RestMethod $api"/folders/lookup$lookupFilter" -Method GET -Headers $headers -ContentType "application/json"
$folder = $lookupResults.records[0]
echo $lookupResults
echo $folder

if($searchResults.total -gt 0 -and $folder.value -eq $name)
{
    echo ""
    echo "-----"
    echo "--Lookup Folder Successful--"
    echo "-----"
    echo ""
    echo $folder
}
else
{
    Write-Error "ERROR: Failed to Lookup Folders."
    return
}

##### Add Group To Folder #####

$folderPermissionCreateArgs = Invoke-RestMethod $api"/folder-permissions/stub?filter.folderId=$folderId" -Method GET -Headers $headers -ContentType "application/json"
#To give permissions to a group, populate the GroupId variable and leave UserId $null.
#To give permissions to a user, populate the UserId variable and leave GroupId $null.
$folderPermissionCreateArgs.GroupId = 3
$folderPermissionCreateArgs.UserId = $null
$folderPermissionCreateArgs.FolderAccessRoleName = "Owner"
$folderPermissionCreateArgs.SecretAccessRoleName = "View"

$permissionArgs = $folderPermissionCreateArgs | ConvertTo-Json

$permissionResults = Invoke-RestMethod $api"/folder-permissions" -Method POST -Headers $headers -Body $permissionArgs -ContentType "application/json"
if($permissionResults.FolderId -eq $folderId)
{
    echo ""
    echo "-----"
    echo "--Add Folder Permissions Successful--"
    echo "-----"
    echo ""
    echo $permissionResults
}
else
{
    Write-Error "ERROR: Failed to Add Folder Permissions."
    return
}
$folderPermissionId = $permissionResults.id

##### Remove Group From Folder #####

$permissionDeleteResult = Invoke-RestMethod $api"/folder-permissions/$folderPermissionId" -Method DELETE -Headers $headers -ContentType "application/json"
if($permissionDeleteResult.id -eq $folderPermissionId)
{
    echo ""
    echo "-----"
    echo "--Remove Folder Permissions Successful--"
    echo "-----"
    echo ""
}
else
{
    Write-Error "ERROR: Failed to Remove Folder Permissions."
    return
}
```

This code sample demonstrates how to use Python 3 to authenticate to Secret Server, retrieve a secret, and update secret fields and items.

This code uses the "requests" module, which makes it simple to communicate via REST. You can install the requests module by running: `pip install requests`

A token is returned from username and password authentication that is used for all subsequent API calls.

```
import http.client
import urllib
import json
import requests

site = '[Your Secret Server Site]' #ex: http://domain.com/SecretServer
authApi = '/oauth2/token'
api = site + '/api/v1'
token = "<TOKEN>"

#REST call to retrieve a secret by ID
def GetSecret(token, secretId):
    headers = {'Authorization': 'Bearer ' + token, 'content-type': 'application/json'}
    resp = requests.get(api + '/secrets/' + str(secretId), headers=headers)

    if resp.status_code not in (200, 304):
        raise Exception("Error retrieving Secret. %s %s" % (resp.status_code, resp))
    return resp.json()

#REST call method to update the secret on the server
def UpdateSecret(token, secret):
    headers = {'Authorization': 'Bearer ' + token, 'content-type': 'application/json'}
    secretId = secret['id']
    resp = requests.put(api + '/secrets/' + str(secretId), json=secret, headers=headers)

    if resp.status_code not in (200, 304):
        raise Exception("Error updating Secret. %s %s" % (resp.status_code, resp))
    return resp.json()

#Retrieves the secret item by its "slug" value
def GetItemBySlug(secretItems, slug):
    for x in secret['items']:
        if x['slug'] == slug:
            return x
    raise Exception("Item not found for slug: %s" % slug)

#Updates the secret item on the secret with the updated secret item
def UpdateSecretItem(secret, updatedItem):
    secretItems = secret['items']
    for x in secretItems:
        if x['itemId'] == updatedItem['itemId']:
            x.update(updatedItem)
            return
    raise Exception("Secret item not found for item id: %s" % str(updatedItem['itemId']))

#Get secret with ID = 1
print("Retrieving Secret with id: 1...")
secret = GetSecret(token, 1)
print("Secret Name: " + secret['name'])
print("Secret ID: " + str(secret['id']))
print("Active: " + str(secret['active']))

#Get the "Notes" secret item
notesItem = GetItemBySlug(secret, 'notes')
print("Notes secret field value: %s" % notesItem['itemValue'])
print()

#Change value of "Notes" secret item
print("Updating secret...")
notesItem.update({'itemValue': 'New Notes Value'})
UpdateSecretItem(secret, notesItem)
print("Secret updated.")
print()
```


```
#Change secret values
updateValues = {'name':'Updated Secret Name' }
secret.update(updateValues)
updatedSecret = UpdateSecret(token, secret)
notesItem = GetItemBySlug(updatedSecret, 'notes')
print("Updated Secret Name: " + updatedSecret['name'])
print("Notes secret field value: %s" % notesItem['itemValue'])
```

Overview

The Secret Server REST API guides are version specific. In fact, they are automatically generated when a Secret Server version is created. Thus, to ensure you have the correct guides *for* your Secret Server version, it is best (but not required) to access the documentation *from* that version of Secret Server; however there are many reasons why this might not be practical, so we provide download links below.

Accessing the Guides

To access the guides:

1. For Secret Server versions 10.5.3 and later with the new user interface, click the  icon in the top right of the dashboard and click **Secret Server REST API Guide**. The Secret Server REST API Guide page appears.
2. Accessing Secret Server through the API and scripts can be powerful mechanism but ensure you are following best practices with the account and access for any script connecting to Secret Server. The Secret Server Software Development Kit for DevOps is recommended for all automated or machine-to-machine scripts. If not using the SDK, creating an application account for the script access and limiting the permissions both from Role perspective and on Secret themselves is security best practice.
3. On the Secret Server REST API page of your instance, click the link for one of the three guides:
 - o **Documentation for REST API using bearer token authentication:** Hyperlinked documentation for REST API access using token authentication.
 - o **Documentation for token authentication:** Instructions for getting an authentication token.
 - o **Documentation for REST API using Windows integrated authentication:** Hyperlinked documentation for REST API access using Integrated Windows Authentication (IWA).

Downloading the Guides

Note: Only use the following links if you cannot easily access a guide from a version of Secret Server. There is a possibility the guides below are not the most current versions.

Current Version

Click the  icon in the top right of the dashboard and click **Secret Server REST API Guide**. The Secret Server REST API Guide page appears.

Note: You can also go to **Admin > REST API Guide**.

Version Archive

- [Secret Server 11.1.000007 REST API Guide](#)
- [Secret Server 11.0.000008 REST API Guide](#)
- [Secret Server 11.0.000007 REST API Guide](#)
- [Secret Server 11.0.000006 REST API Guide](#)
- [Secret Server 10.9.000064 REST API Guide](#)
- [Secret Server 10.9.000033 REST API Guide](#)
- [Secret Server 10.8.000000 REST API Guide](#)
- [Secret Server 10.7.000000 REST API Guide](#)
- [Secret Server 10.6.000000 REST API Guide](#)
- [Secret Server 10.5.000000 REST API Guide](#)
- [Secret Server 10.4.000000 REST API Guide](#)

- [Secret Server 10.3.000000 REST API Guide](#)
- [Secret Server 10.2.000000 REST API Guide](#)
- [Secret Server 10.1.000000 REST API Guide](#)

Overview

Manipulating the Secret Server API via scripts is very handy and powerful but also potentially dangerous, so it is critical to follow best practices to ensure robust security. This topic discusses these best practices. We recommend the Secret Server Software Development Kit for DevOps for all automated or machine-to-machine scripts. If not using the SDK, we strongly recommend creating an application account for script access and limiting permissions via both roles and the secrets themselves.

Best Practices

For security, it is critical that you:

- Do not use admin accounts.
- Use a least-privilege account, limiting role and secret access permissions.
- Never store passwords or tokens in plain text. Use secure credentials instead—see below for details.
- Use a least-privilege application account. These are restricted from logging in the UI.
- Never use domain admin credentials in an automated script.
- Create a specific service account with only the permissions needed to get the task done.
- Always expire tokens. To expire tokens, navigate to the **Admin > Configuration > Application** and enable **Session Timeout for Webservices**, setting the timeout to a short period, the default is 20 mins.

Authentication Methods

Scripts Run by Automated Processes

Recommended methods:

- Secret Server Software Development Kit for DevOps (SDK). The SDK is a great way to limit access to a specific client with encrypted credentials and IP restrictions. See [Secret Server Software Development Kit for DevOps](#) for more information.
- Integrated Windows Authentication (IWA).

Specific User Use Cases with Script Augmentation

Recommended methods:

- Get credentials with two-factor authentication (2FA). This is a secure way to prompt users for credentials for manually run scripts.
- Prompt with 2FA. This prompts the user for script credentials.
- IWA.

Not Recommended method:

- API token from the UI with 2FA.

Note: If your account is secured with 2FA, use the `-UseTwoFactor` switch to be prompted for your two-factor OTP code.

Example Credential-Access Scripts

Recommended Methods

Get-Credential with 2FA

```
$Application = "https://<Secret Server URL>"

function Get-Token
{
[CmdletBinding()]
param (
    $credentials,
    [Switch] $UseTwoFactor
)

$creds = @{
    username = $credentials.UserName
    password = $credentials.GetNetworkCredential().Password
    grant_type = "password"
} ;

$headers = $null
If ($UseTwoFactor) {
    $headers = @{
        "OTP" = (Read-Host -Prompt "Enter your OTP for 2FA: ")
    }
}

try
{
$response = Invoke-RestMethod "$Application/oauth2/token" -Method Post -Body $creds -Headers $headers;
$token = $response.access_token;
return $token;
}
catch
{
$result = $_.Exception.Response.GetResponseStream();
$reader = New-Object System.IO.StreamReader($result);
$reader.BaseStream.Position = 0;
$reader.DiscardBufferedData();
$responseBody = $reader.ReadToEnd() | ConvertFrom-Json
Write-Host "ERROR: $($responseBody.error)"
return ;
}
}
```

```
$Token = Get-Token -credentials (Get-Credential) -UseTwoFactor
```

Prompting for Credentials with 2FA

```
$Application = "https://<Secret Server URL>"

function Get-Token
{
[CmdletBinding()]
param (
    [Switch] $UseTwoFactor
)

$creds = @{
    username = Read-Host -Prompt "Enter your username: "
    password = Read-Host -Prompt "Enter your password: "
    grant_type = "password"
} ;

$headers = $null
If ($UseTwoFactor) {
    $headers = @{
        "OTP" = (Read-Host -Prompt "Enter your OTP for 2FA: ")
    }
}

try
{
$response = Invoke-RestMethod "$Application/oauth2/token" -Method Post -Body $creds -Headers $headers;
$token = $response.access_token;
return $token;
}
catch
{
}
```



```
$result = $_.Exception.Response.GetResponseStream();  
$reader = New-Object System.IO.StreamReader($result);  
$reader.BaseStream.Position = 0;  
$reader.DiscardBufferedData();  
$responseBody = $reader.ReadToEnd() | ConvertFrom-Json  
Write-Host "ERROR: $($responseBody.error)"  
    r e t u r n ;  
    }  
}
```

```
$token = Get-Token -UseTwoFactor
```

Integrated Windows Authentication

Note: Secret Server Cloud does not support IWA.

```
$api = "https://<Secret Server URL>/winauthwebservice/api/v1"  
$endpoint = "$api/secrets/3844"  
$secret = Invoke-RestMethod $endpoint -UseDefaultCredentials
```

Note: See [Using Webservices with IWA via PowerShell](#) for more information.

Deprecated Method

API token from the UI with 2FA

Note: The user must first be set up with 2FA.

```
$application = "https://<Secret Server URL>"  
$token = "<TOKEN>"
```

```
$headers = @{}  
$headers.Add("Authorization", "Bearer $token")
```

1. From the Secret Server REST API page of your instance, select one of the following token authentication methods for user scripts:
 - Bearer token authentication
 - Token authentication
 - Windows integrated Authentication
2. Generate a token from your user preferences in Secret Server. Use this as a bearer token in user scripts.
3. Click the user-initials button on the top right of the page and select **User Preferences**. The User Preferences page appears:

User Preferences

General Settings Security

General Information

This section contains general information about your user account within your organization's framework. The information in this section cannot be edited on this page. To update your display name, username, or email, please contact your administrator.

Display Name	Will Sprunk
Username	wsprunk@gamma.thycotic.com
Email	None
Domain	gamma.thycotic.com
Last Login	15 minutes ago

API Token

Create an API Token for use in scripts.

[Generate API Token and Copy to Clipboard](#)

Password Settings

Change your local user password used to log in to Secret Server, or make changes to your user account's DoubleLock password.

[Create DoubleLock Password](#)

4. In the **API Token** section, click on **Generate API Token and Copy to Clipboard**.
5. Paste the token text into the script.

Downloading Example Scripts

Example Scripts:

- [PowerShell Examples](#)

Note: For REST API Client Generation (Advanced), please see [REST API Client Generation with OpenAPI Swagger](#)

Overview

This Secret Server Software Development Kit (SDK) for DevOps tool, or simply SDK, was created for securing and streamlining DevOps processes with regard to Secret Server. The SDK for DevOps tool allows you to more efficiently engage Secret Server via a Command Line Interface (CLI) without compromising security. It allows you to securely retrieve credentials from and track access to a secure vault.

The SDK uses the [Secret Server REST API](#). The SDK is a .NET library (available via a NuGet package), which you can use in a custom application. The SDK .NET library exposes a limited subset of the REST API.

Note: You can download the current and legacy versions of the API on the [Downloads for Secret Server Software Development Kit for DevOps](#) page.

Note: The SDK can run on any version of .NET that supports .NET Standard 2.0 apps including .NET Core 2.0 plus, .NET Framework 4.6.1 plus, .NET 5 plus, and .NET 6 plus.

Note: See the [Delinea SDK Integration Doc](#) on GitHub for more information.

There is also a .NET Core CLI SDK Client that uses the SDK .NET library. The SDK Client was created to allow customers to write automation scripts to access secrets without having to write code to directly access the REST API.

The .NET SDK library and the .NET Core CLI client both:

- Automatically store the credentials and remote server in an encrypted file used to acquire an OAUTH token. The token is then used to make subsequent API calls. OAUTH tokens have an expiration time, which is configurable in the UI on the configuration page via the "Session Timeout for Webservices" value.
- Get the contents of a secret.
- Provide client-side caching (SDK client caching)

Secret Server has user and application accounts. Both types can access Secret Server via the REST API. Application accounts are not counted for licensing purposes. Application account can *only* access Secret Server via the REST API. Both account types never expire. Secret Server provides security for automated clients. SDK rules manage permissions. Client IDs are created when `SecretServerClient.Configure()` or `tss init` is called. The client ID is used to reference SDK client instances.

Note: Do not give an Application Account the Administrator Role or give it all of the Role permissions. It is recommended to only assign the minimum roles needed to complete the task, that means create a new Role with only the permissions it needs and only share the secrets needed.

Note: For REST API Client Generation (Advanced), please see [REST API Client Generation with OpenAPI Swagger](#)

Note: We have a [Python SDK](#) that is independent of the SDK .NET library. It allows a Python script to access secrets without requiring REST knowledge. It has access to a small subset of the REST API.

How It Works

The SDK is a console application written in .NET Core that wires up its own credentials based on the machine it is installed on. Those credentials, called "DevOps Users", do not have any rights in Secret Server but can be assigned to other Secret Server users or application user accounts, mimicking permissions to access secrets.

This removes the widespread DevOps issues with hard coding credentials into scripts and configuration files. Instead, the target system is registered via IP address that is white listed, which provides REST authentication without entering user credentials. You can use the SDK to retrieve a REST user token for our REST API, or you can use the SDK to perform direct queries on Secret Server.

In summary, the SDK establishes secure access points so that power users can employ the Secret Server API directly from the CLI, without

wasting time entering privileged account passwords.

Configuration Overview

Secret Server exposes a REST API interface that is used by the SDK client. When the SDK sends a REST request, Secret Server determines where the request came from (via IP address) and what permissions it should be granted (via a rule set in Secret Server). Thus, once the client is initialized with the Secret Server and the rule name, that configuration is stored and encrypted on the client, ready for subsequent calls.

Out of the box, the SDK offers:

- token retrieval
- secret retrieval
- secret field retrieval

Note: We expect to expand the SDK's capabilities over time to allow even greater access to the REST API.

The SDK requires setup in two areas: Secret Server configuration and SDK installation on the DevOps system.

Required Roles and Permissions

- **Administer Configuration:** Allows a user to enable SDK functionality in Secret Server, that is, to enable webservice and enable the SDK itself.
- **Administer Users:** Allows a user to use the SDK to retrieve account credentials on client machines. Alternatively, you can be the owner of the application account used by the SDK.
- **Administer Create Users:** Allows a user to access the **Admin > SDK Client Management** page in Secret Server.

Setup Procedure

Task 1: Configuring Secret Server

Configure Secret Server for communication with the SDK:

1. Navigate to **Admin > Configuration**.
2. Click the **General** tab.
3. Click the **Edit** button.
4. Click to select the **Enable Webservice** check box in the **Application Settings** section.
5. Click the **Save** button.
6. Select and setup any application accounts that you want for use by SDK clients. Make sure these application accounts have appropriate permissions to access any secrets or execute any operations the client host needs to perform. To create a new account with the needed permissions:
 1. Go to **Admin > Users**.
 2. Click the **Create New** button. The Edit User page appears:

Edit User

User Name

Display Name

Email Address

Domain

Password

Confirm

Two Factor

Enabled

Locked Out

[Advanced](#)

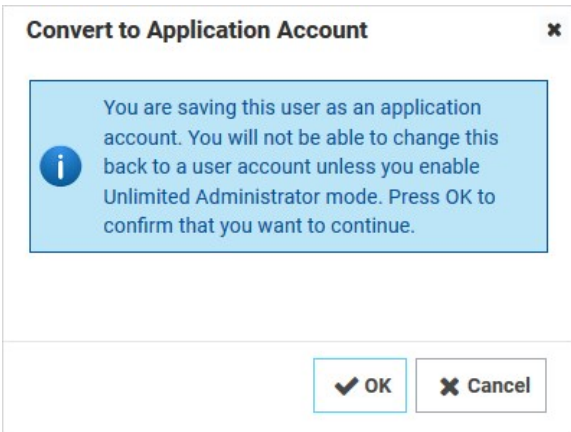
3. Type the account name in the **User Name** and **Display Name** text boxes.
4. Type a password in the **Password** and **Confirm** text boxes. Record the password for future use.
5. Click the **Advanced** link. The Advance section appears:

[Advanced](#)

Application Account *As an application account, the user will only be able to log in through the Application Account API and will not require a separate user license. (See [KB Article](#))*

Managed By

6. Click to select the **Application Account** check box.
7. Click the **Save** button. A confirmation popup appears.



8. Click the **OK** button. The View User page appears:

View User

User Name	DevOps1
Display Name	DevOps1
Email Address	
Domain	Local
Two Factor	< None >
Enabled	Yes
Locked Out	No
Application Account	Yes

IP Address Restrictions
None

Restricted By Team No

GROUPS FOR USER

There are no groups for this user.

ROLES FOR USER

Save To File < 1 to 1 of 1 >

ROLE NAME

User

[Back](#) [Edit](#) [View Audit](#) [Change IP Restrictions](#) [Assign Groups](#)

[Assign Roles](#) [Configure Secret Template Permissions](#)

7. Create a new role:

1. Goto **Admin > Roles**.
2. Click the **Create New** button. The Role Edit page appears:

Role Edit

Role Name *

Enabled

Created

Permissions Assigned

Permissions Unassigned

- Access Offline Secrets on Mobile
- Add Secret
- Add Secret Custom Audit
- Administer Active Directory
- Administer Backup
- Administer Configuration
- Administer Configuration Proxying
- Administer Configuration SAML
- Administer Configuration Security
- Administer Configuration Session Recording
- Administer Configuration Two Factor
- Administer Configuration Unlimited Admin
- Administer ConnectWise Integration
- Administer Create Application Accounts
- Administer Create Users

Save Cancel

3. Type the new role name in the **Role Name** text box.

4. Assign the **View Secret** permission to that role. The permission appears in the Permissions Assigned text box. You can add additional permissions later as needed.

5. Click the **Save** button. The new role appears in the Roles table.

8. Enable SDK management:

1. Go **Admin > See All**. The admin panel appears.

2. Type SDK in the **Search** text box and select **SDK Client Management**. The SDK Client Management page appears:

SDK Client Management

Accounts Client Onboarding Audit Disabled

Search 10 All Assignments

Records: 0 Page: 1 / 0 < Prev Next >

NAME	DETAILS	ASSIGNMENT
------	---------	------------

3. Click the **Disabled** toggle button to change it to **Enabled**.

9. Set up a SDK client rule:

1. Click the **Client Onboarding** tab.

SDK Client Management

Accounts Client Onboarding Audit Enabled

Search 10 All Assignments Records: 0 Page: 1 / 0 « Prev Next »

NAME	DETAILS	ASSIGNMENT	REQUIRE THIS GENERATED ONBOARDING KEY
------	---------	------------	---------------------------------------

+ Rule

2. Click the **+ Rule** link. A new rule appears:

SDK Client Management

Accounts Client Onboarding Audit Enabled

Search 10 All Assignments Records: 0 Page: 1 / 0 « Prev Next »

NAME	DETAILS	ASSIGNMENT	REQUIRE THIS GENERATED ONBOARDING KEY
Rule Name	IPV4 Address, or CIDR notation	Not Selected	<input type="checkbox"/> Require this generated onboarding key

Save Cancel

3. Type a short, unique name in the **Rule Name** text box. Clients must provide a valid rule name when connecting. For example: ProductionWebApp.
4. Type an IPV 4 address or address range (in CIDR notation) in the **Details** text box. Secret Server will only allow clients to use this rule if they connect from a valid IP address. If not provided, Secret Server will not enforce IP address restrictions on this rule. We strongly recommend using this feature.

Note: There is a 250-character limit, so you can only add a few dozen IP addresses unless you use CIDR notation.
5. Click to select the application account you created earlier in the **Assignment** dropdown list. Clients are granted the same permissions as this account within Secret Server. If not provided, an account will be automatically created for clients, but will have no default permissions. You must use an application account (the one you created) for a rule. Application accounts are restricted from logging into the system through the normal user interface and do not count towards your license quota.
6. Click to select the **Require this generated onboarding key** check box. Clients must provide a generated additional key string when authenticating. If not provided, Secret Server allow any client to use the rule if its IP address is within the specified range. We strongly recommend using this feature.
7. Click the **Save** button. The Show Key link appears.
8. Click the **Show Key** link to save the generated onboarding key (something like TFyORLL1teQmD8OAMstqKGWkJGksFRtaelY0b2NnhsM=) for future use. It will not be visible again.

Note: If you cannot copy the key text after selecting it, you probably need to add the Secret Server Utilities extension for your

browser. For now, just manually copy it.

Task 2: Installing the SDK Client

Note: This requires Secret Server version 10.4+ or Secret Server Cloud

Note: IWA is not supported by the SDK.

Note: The SDK can run on any version of .NET that supports .NET Standard 2.0 apps including .NET Core 2.0 plus, .NET Framework 4.6.1 plus, .NET 5 plus, and .NET 6 plus.

1. [Download the SDK](#) for your platform.
2. Unzip the SDK zip file you downloaded.
3. To get the SDK Nuget packages, see our [documentation on GitHub](#).
4. On non-Windows systems, you must make the tss program executable by running `chmod u+x tss`.
5. On Linux systems, you must install libunwind as follows:
 - On Red Hat or Centos, run `sudo yum install libunwind libicu`
 - On Ubuntu, run `sudo apt-get install libunwind-dev`
6. To confirm the SDK client tool is installed and working, run the help:
 - On Windows, run `tss --help`
 - On non-Windows systems, run `./tss --help`

Task 3: Connecting to Secret Server

Before the client can retrieve data from Secret Server, it must be initialized to connect to the Secret Server instance. This is a one-time operation on the client machine:

Run `tss init --url <url> -r <rule> -k <key>` using the parameters you recorded earlier for your Secret Server instance.

For example, if your parameters are:

- Secret Server is hosted at `https://myserver/SecretServer/`
- You created a rule named `ProductionWebApp`
- Your onboarding key is `CNrQwRBscnq4qAZ6v3EIAcE27vQuLiz6KSpfRjHryA=`

Then you would run:

```
tss init --url https://myserver/SecretServer/ -r ProductionWebApp -k CNrQwRBscnq4qAZ6v3EIAcE27vQuLiz6KSpfRjHryA=
```

Note: You can also start the SDK client in interactive mode (`tss -i`) and follow the prompts to run the `init` command.

Secret Server will validate that the client-provided information is correct and that the IP address matches the configured restrictions. If successful, the client and server have established the connection, and the client can now be used to retrieve Secret Server data.

Usage Examples

- Retrieving a secret by ID (returns a JSON structure describing the entire secret record): `tss secret -s 4`
- Retrieving all secret field values for a secret by ID: `tss secret -s 4 -ad`
- Retrieving only the value of a particular secret field by secret ID: `tss secret -s 4 -f password`
- Writing a secret field value to a file: `tss secret -s 4 -f password -o passwordfile.txt`

- Retrieving an access token for use in other REST API requests: `tss token`

The SDK client also includes an interactive mode (`tss -i`) that allows you to input multiple commands into a series of prompts. To exit interactive mode, run the `exit` command.

SDK Client Management

To view and manage a list of connected SDK clients from within Secret Server:

1. Go **Admin > See All**. The admin panel appears.
2. Type SDK in the **Search** text box and select **SDK Client Management**. The SDK Client Management page appears.
3. Click the **Accounts** tab. A list of connected SDK clients SDK appears. You can remove or rename them. You can use the **Enable/Disable** button at the top of the page to disable and re-enable all SDK client activity.
4. Click the **Client Onboarding** tab to manage the onboarding rules.
5. Click the **Audit** tab. A list of SDK client activity appears.

Note: If you remove a connected client, it may be able to reconnect unless you alter or remove the rule that it used. You can use the button above the grid to disable and re-enable all SDK client activity.

6. To remove the Secret Server connection from a client machine, run the `tss remove` command. This deletes the connection, and the client can no longer retrieve Secret Server data without being re-initialized.

SDK Client Caching

Overview

To increase performance and reliability, you can configure the SDK client to cache values retrieved from Secret Server on the client machine. Cached values are stored encrypted on disk. You can configure client caching in one of four ways:

- **Never (0):** With this default setting, the client never caches Secret Server data. All data requests result in a query directly against the SSS instance. If the instance is unavailable, the requests fail. Secret Server
- **Server Then Cache (1):** With this setting, the client first attempts to retrieve the value from the server. If the server is unavailable, it checks to see if the same value has been previously fetched within a given period, and if so, it will return the cached value. Use this setting to guard against losing the connection to SS.
- **Cache Then Server (2):** With this setting, the client first checks to see if the same value has been previously fetched within a given period. If so, it returns the value without consulting the server. If not, it fetches, caches, and returns the value from the server. Use this setting for increased performance by reducing requests sent to SS.
- **Cache Then Server Fallback on Expired Cache (3):** This strategy operates similarly to "Cache Then Server," but if the server is unavailable and an expired value exists in the cache, the client returns that value as a last resort. Use this strategy for increased performance and reliability.

All these cache strategies have a configurable age, in minutes, after which the value is considered expired and not used (except in "Cache Then Server Fallback" mode). Cache settings are set using the client application. See the examples below.

Examples

- Turn off caching: `tss cache --strategy 0`
- Turn on "Cache Then Server" setting with a cache age of five minutes: `tss cache --strategy 2 --age 5`
- Immediately clear all cached values: `tss cache --bust`
- Show the current cache settings: `tss cache --current`

Important: Anytime you use a cached value, recent changes made to Secret Server may not be applied, including changes to the value itself, permissions, or other access control settings. Examine your organization's security policies and application requirements to determine the best cache settings to use.

Note: Please click the table of contents on the left to see the sub-pages to this one. Click the table of contents on the right to see headings on this page.

SSH Blocked Command Lists

Overview

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that you can apply to secrets with SSH launchers over SSH proxy. With privilege management, you can grant users access to a machine to block specific commands that a user may run as root or any other privileged account.

Note: To use command restrictions, Secret Server must have SSH Proxy and Enable Block Listing enabled.

With SSH blocked command lists, you can define disallowed commands when connecting as a privileged account. The blocked command list is defined by a series of regular expressions.

Upon launching a secret with an assigned SSH command blocklist, each command sent to the target is evaluated for a match on the blocklist. If the command is found to match a list entry, that command is blocked from execution. Blocked output is shown to the user at the terminal.

Requirements

System requirements:

- Secret Server 10.9.000064 or later Secret Server
- Secret Server Platinum Edition license or Secret Server Professional and Unix SUPM license
- SSH proxy must be enabled

Creating SSH Blocked Command Lists

The format for specifying a blocked command follows a regular expression syntax that is typical to most scripting languages. The blocked commands are surrounded with `\b` word boundary anchors. For example, to block the `sudo` command, you use:

```
\bsudo\b
```

This expression blocks the execution of any command with the `sudo` string in it (as a separate word), for example, these are blocked:

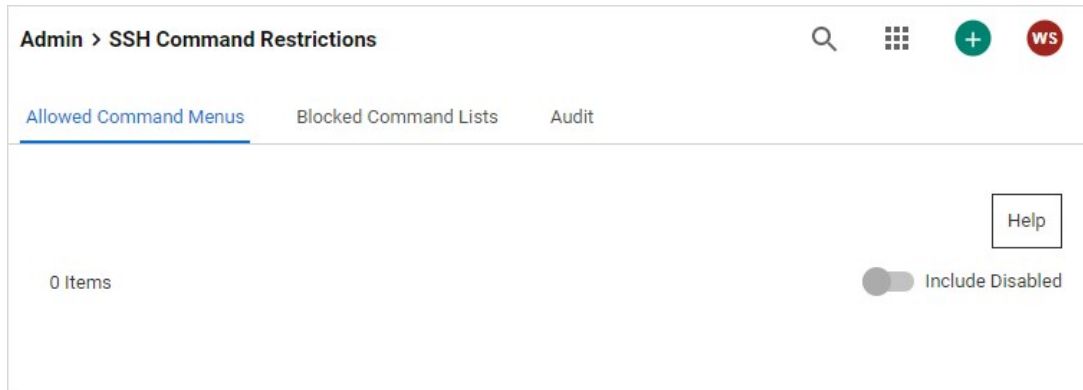
- `sudo`
- `sudo root`
- `sudo ls /usr/local/protected`
- `sudo shutdown -r +15 "quick reboot"`

And these are not:

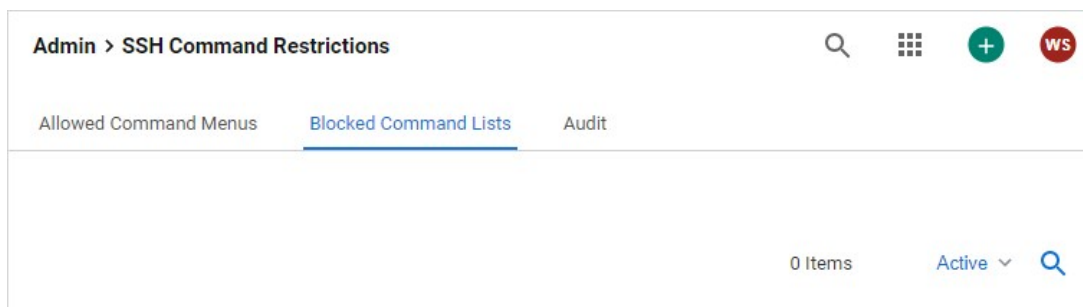
- `cat sudoku`
- `echo "sudo"`

To create a list of blocked commands:

1. Go to **Admin > See All**.
2. Hover the mouse pointer over the **Actions** menu item and select **SSH Command Restrictions**. The SSH Command Restrictions page appears:



3. Click the **Blocked Command List** tab:



4. Click the **Create Blocked Command List** button. The New Blocked Command List page appears:

New Blocked Command List

Name *

Enabled *

Description

Blocked Commands *

Prevent superuser commands

Prevents superuser commands

Selected Commands (2)

Blocks su commands : \bsu\b	Remove
Blocks sudo commands : \bsudo\b	Remove

Add Commands to Blocked Command List

- +
Blocks copy commands : \wcp\b
- +
Blocks move commands : \bmv\b
- +
Blocks reboot commands : \breboot\b
- +
Blocks remove commands : \brm\b
- +
Blocks shred commands : \bshred\b
- +
Blocks wget commands : \bwget\b

Cancel

Create Blocked Command List

5. Type the list name in the **Name** text box.
6. Ensure the **Enabled** check box is selected.
7. **Either** select a predefined regex from the **Add Commands to Blocked Command List** dropdown list. **Or** add a custom regex of your own:
 1. Scroll to the bottom of the dropdown list.
 2. Click the **Create New Command** link. The Create Command and Add to Blocked Command List popup appears:

Create Command and add to Blocked Command List

Command Name *

Command Pattern(regex) *

1	\bls\b
---	--------

3. Type the name for the command in the **Command Name** text box.
4. Type or paste the regex in the **Command Pattern** text box.
5. Click the **Create and Add** button. The command is added to the dropdown list.
8. Click the **Create Blocked Command List** button.

Applying SSH Command Blocked Lists in Secret Settings

To enable privilege management for an account:

1. Navigate to a PuTTY secret's **Settings** tab.
2. Go to the **SSH Launcher** section.
3. Click the **Edit** link.
4. Click the **Connect Using** dropdown list and select **Credentials on Another Secret**.
5. Click the **No Secret Selected** link to choose a secret containing your log on credentials, which the launcher uses when logging on the SSH service.
6. Enable command restrictions:
 1. Click the **Security** tab.
 2. Go to the **Other Security** section.
 3. Click the **Edit** link to set **Enable Proxy** to Yes.
 4. Click the **Edit** link for the **Enable SSH Command Restrictions**. The Edit SSH Command Restrictions popup appears:

Edit SSH Command Restrictions

Restrict SSH Commands

Allowed Command Menus
 Blocked Command Lists

Owner Permission Unrestricted

Edit Permission Admin Command Blocklist

View Permission User Command Blocklist

Cancel
Save

5. Ensure the Restrict SSH Commands check box is selected.
6. Click to select the Blocked Command Lists selection button.
7. Click the **Owner**, **Edit**, and **View Permission** dropdown lists to map the blocked command lists to users via those permissions. You can also leave them as unrestricted.
8. Click the **Save** button.

SSH Command Restrictions via a Secret Policy

You can apply SSH command restrictions to a secret policy for ease of management. You can apply secret policies to secret folders or directly to a secret itself. To apply command restrictions, set a policy as follows:

Table: Secret Policy Security Settings for SSH Command Restrictions

Security Settings	Enable Proxy	Enforced	Checked
Security Settings	Enable SSH Command Restrictions	Enforced	Checked
Security Settings	SSH Command Restriction Type	Enforced	Blocked List
Security Settings	SSH Command Blocklist for Secret Owners	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Editors	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Viewers	Enforced or Not Set	Desired Block Command List or Not Set

SSH Command Menus

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that can be applied to secrets with SSH Launchers over SSH Proxy. Privilege management gives the ability to grant users access to a machine with specific command restrictions to define the available commands that a user may run as root or another privileged account.

Note: To use command restrictions, Secret Server must have SSH Proxy enabled.

With command menus, you can configure predefined commands that users or groups will be able to access when connecting as a privileged account. A command menu is a list of command names mapped to system commands. The format for specifying a command is to separate a name and command with an equals symbol. For example:

```
restart_apache = /usr/sbin/service apache restart
```

You may also use parameters in commands so users can execute more complex commands. For example:

```
move_file = /bin/mv $src $dst
```

You can specify environmental variables by escaping dollar signs in commands. For example:

```
go_home = cd $$HOME
```

Command restrictions currently do not support complex commands, such as multiple commands on one line, piping, or output redirection. To support these functions, you may add a script to the system that has these capabilities and point map the command to that script. We highly recommend that generated scripts have proper user permissions and that the absolute path is used. The absolute path ensures that the correct script is being executed.

Commands may not be named as numbers or one of the following predefined commands:

```
..
up
-help
?
-more
logout
exit
```

To enable privilege management for an account, navigate to a PuTTY Secret's launcher tab and specify the "Connect As" secret that you wish to connect as. When launching this secret, the launcher uses it as credentials to log into the SSH service and then log into the credentials specified on the secret.

To enable command restrictions, navigate to a PuTTY secret's security tab and specify "Enable Proxy" and "Enable SSH Command Restrictions." This gives you the ability to map users and groups to command menus. When the unrestricted command menu is specified for a user, the user is launched into a normal shell environment without command restrictions. Likewise, if the "Allow Owners Unrestricted SSH Commands" option is enabled, the owners of the secret are also launched into a normal shell environment without command restrictions.

When specifying command menus on a secret, at least one command menu must be selected unless "Allow Owners Unrestricted SSH Commands" is enabled.

You can apply command restrictions to a secret policy for ease of management.

Secret Server supports using PowerShell, SSH, and SQL scripts as dependencies on a secret. These scripts can use information on the secret through the field name prepended with a \$. For example, \$DOMAIN, \$PASSWORD, or \$USERNAME. Linked secrets are accessible by \${1}\$FIELDNAME for the first linked secret, \${2}\$FIELDNAME for the second, and so on.

There are two contexts in which script dependencies run:

- As part of the RPC process. See [Password Changing Scripts](#).
- When run manually from the Dependencies tab on the secret.

For a complete list of tokens that are available to script dependencies, see [List of Dependency Tokens](#).

Overview

You can enable webservices at **Admin > Configuration** on the **General** tab. Checking the **Enable Webservices** check box makes the ASP.NET SOAP and REST webservices built into Secret Server available for use. Additional steps are needed in IIS to ensure proper access.

Note: Integrated Windows Authentication (IWA) does **not** work on Secret Server Cloud.

Note: This procedure only works if Secret Server on-premises is installed on IIS 7 or greater.

Procedure

To enable IWA for webservices in IIS:

1. Open IIS Manager (inetmgr).
2. Expand the **Sites** node until you locate your Secret Server application or website
3. Expand the **Secret Server** node to locate the **winauthwebservices** folder.
4. Click on the **winauthwebservices** folder.
5. Click on **authentication** in the **Security** section.
6. Disable **Anonymous Authentication**.
7. Enable **Windows Authentication**.

Note: If you are using IIS7 or greater and do not see this option, the option will need to be added through the server roles (webserver). IIS may give an alert about using both challenge and redirect-based authentication, which you can ignore.)

8. Open Windows Explorer.
9. Navigate to the **winauthwebservices** folder.
10. Give **read access** to the **winauthwebservices** folder to the domain users and groups that will be using IWA to access the webservices.

Access Examples

SOAP

The SOAP web service URL for IWA is <Secret Server URL>/winauthwebservices/sswinauthwebservice.asmx.

Example script:

```
# Sample Powershell Script
# demonstrating retrieval of a Secret from Secret Server
# via web service protected by IWA

$where = 'http://mysecretserver/winauthwebservices/sswinauthwebservice.asmx';
$secretId = 1
$sws = New-WebServiceProxy -uri $where -UseDefaultCredential
$swsResult = $sws.GetSecret($secretId, $false, $null)
if ($swsResult.Errors.length -gt 0){
    $swsResult.Errors[0]
}
else
{
    $swsResult.Secret
```

```
}
```

REST

REST web service references the same winauthwebservices folder as SOAP when doing IWA, but in code the URL endpoint does not need to change.

Example script:

```
# Sample Powershell Script  
# demonstrating authentication via web service protected by IWA  
  
$api = 'http://mysecretserver/winauthwebservices/api/v1';  
$endpoint = $api/secrets/3844'  
$secret = Invoke-RestMethod $endpoint -UseDefaultCredentials
```

Overview

You can enable webservices at **Admin > Configuration** on the **General** tab. Checking the **Enable Webservices** check box makes the ASP.NET SOAP and REST webservices built into Secret Server available for use. Additional steps are needed in IIS to ensure proper access.

Note: Integrated Windows Authentication (IWA) does **not** work on Secret Server Cloud.

Note: This procedure only works if Secret Server on-premises is installed on IIS 7 or greater.

Procedure

To enable IWA for webservices in IIS:

1. Open IIS Manager (inetmgr).
2. Expand the **Sites** node until you locate your Secret Server application or website
3. Expand the **Secret Server** node to locate the **winauthwebservices** folder.
4. Click on the **winauthwebservices** folder.
5. Click on **authentication** in the **Security** section.
6. Disable **Anonymous Authentication**.
7. Enable **Windows Authentication**.

Note: If you are using IIS7 or greater and do not see this option, the option will need to be added through the server roles (webserver). IIS may give an alert about using both challenge and redirect-based authentication, which you can ignore.)

8. Open Windows Explorer.
9. Navigate to the **winauthwebservices** folder.
10. Give **read access** to the **winauthwebservices** folder to the domain users and groups that will be using IWA to access the webservices.

Example

Overview

The SOAP web service URL for IWA is <Secret Server URL>/winauthwebservices/sswinauthwebservice.asmx.

The method below uses the SecretServerGetSecret.ps1 PowerShell script to make the SecretGet WebService call, exposing it through the SecretServer.pm Perl package. The Sample.pl file uses the SecretServer.pm package to retrieve specific fields from the result.

The flow is as follows:

1. Your Perl script (sample.pl) makes a request to the SecretServer.pm package.
2. The SecretServer.pmpackage passes the request on to the SecretServerGetSecret.ps1 PowerShell script.
3. The SecretServerGetSecret.ps1 PowerShell script calls the Secret Server web services and authenticates using the service account that sample.pl is running under.

4. The results are passed back to SecretServer.pm and then on to your Perl script (Sample.pl)

5. Create the following three files:

SecretServerGetSecret.ps1

```
# Sample Powershell Script
# demonstrating retrieval of a Secret from Secret Server
# via web service protected by Windows Authentication
# returned as Xml

$where = $args[0]
$secretid = $args[1]
$ws = New-WebServiceProxy -uri $where -UseDefaultCredential
$wsResult = $ws.GetSecret($secretid)
$res = convertto-xml $wsResult.Secret -As string -Depth 20
$res
```

SecretServer.pm

```
package SecretServer;
use strict;

sub usage {
    print "\nUsage: GetSecret [webservice url] [secretid]\n";
}

sub new {
    my($class, %args) = @_;
    my $self = bless({}, $class);
    return($self);
}

sub get_secret {

    my($self, $url, $secretid) = @_;
    my $result = `powershell.exe .\SecretServerGetSecret.ps1 $url $secretid`;
    return($result);
}

sub get_field_from_result {

    my($self, $result, $field) = @_;
    $result =~ /<Property Name="Value" Type="System.String">([^\<]+<\/Property>(?:\s*<Property Name="(?!FieldName)[^"]+"[^\>]+<\/Property>)*<Property Name="FieldName"[^\>]+>$field<\/Property>/gsi;

    return("$1");
}

1;

# this is if you want to execute the Get Secret call manually from the command line
# if (@ARGV != 2)
# {
#     usage(); # Call subroutine usage()
#     exit(); # When usage() has completed execution,
#             ## exit the program.
# }

# my $url = $ARGV[0];
# my $secretid = $ARGV[1];
# my $result = `powershell.exe .\SecretServerGetSecret.ps1 $url $secretid`;

# print $result;
```

Sample.pl

```
use lib 'C:/Path to the SecretServer.pm file>';
use SecretServer;

my $x = SecretServer->new();

#Change this value to match your URL
```



```
my $url = 'http://<Your Secret Server Url>/winauthwebservices/sswinauthwebservice.asmx';
```

```
#Change this value to match your desired Secret Id
```

```
my $secretid = 17;
```

```
my $result = $x->get_secret($url, $secretid);
```

```
my $username = $x->get_field_from_result($result, 'UserName');
```

```
my $password = $x->get_field_from_result($result, 'Password');
```

```
print "$username : $password";
```

Secret Server Authentication, Encryption, and Security

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server provides integration options for Windows authentication and SAML to automatically authenticate users to the application when they browse to Secret Server on their workstations. Secret Server also allows you encrypt data at various locations.

Introduction

In some cases, a PowerShell script may need to access resources outside of a Secret Server machine. This requires that any credentials are delegated to the target machine. Secret Server runs PowerShell scripts using Windows Remote Management (WinRM), which does not allow credential delegation by default. To allow credential delegation, the Secret Server machine must have Credential Security Support Provider (CredSSP) enabled. CredSSP is a security support provider that allows a client to delegate credentials to a target server.

Some scenarios requiring CredSSP:

- The script needs to query or update a value in Active Directory.
- The script needs to query or update a value in a SQL Server instance.
- The script is used as part of extensible discovery for locating accounts or machines on a different domain or non-domain joined environment.

Enabling CredSSP for WinRM in Secret Server

1. Go to **Administration > Configuration**. The General tab of the Configuration page appears:

APPLICATION SETTINGS	
Allow Automatic Checks for Software Updates	Yes
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	No View Metric Data
View Webservices	
Enable Webservices	Yes
Maximum Time for Offline Access on Mobile Devices	30 days
Session Timeout for Webservices	20 minutes
Enable Refresh Tokens for Web Services	No
Prevent Application from Sleeping When Idle	Yes
Syslog/CEF Logging Advanced Settings Information	
Enable Syslog/CEF Logging	No
Test PowerShell with WinRM	
WinRM Endpoint URL	http://localhost:5985/wsman
How do I configure CredSSP for WinRM?	
Enable CredSSP Authentication for WinRM	Yes

2. Click **Edit** button at the bottom of the page.
3. Click to select the **Enable CredSSP Authentication for WinRM** checkbox.

4. Click the **Save** button.

Note: This is the global CredSSP settings and by default will configure CredSSP and connections to come *from* the Web server. This is used when **not using distributed engines**.

Note: If you are using distributed engines and you enable CredSSP at the site-specific level, these settings take precedence over this global CredSSP setting. Secrets will prioritize these site-specific settings. Therefore, if you plan on using CredSSP through a distributed engine, you should consider disabling the global setting seen below and only configure it at the site-specific level.

Configuring CredSSP for WinRM on the Secret Server Machine

1. Log on to the machine running Secret Server.
2. Run Windows PowerShell as an administrator.
3. Enable client-side CredSSP by running:

```
Enable-WSManCredSSP -Role Client -DelegateComputer <Secret Server fully qualified machine name>
```

For example:

```
Enable-WSManCredSSP -Role Client -DelegateComputer <localhost>
```

Note: localhost is the actual string that Secret Server uses to generate the PowerShell run space. Sometimes customers need both localhost and FQDN entries. *In theory*, those entries should be the same, thus not needing a second one.

4. Enable server-side CredSSP by running:

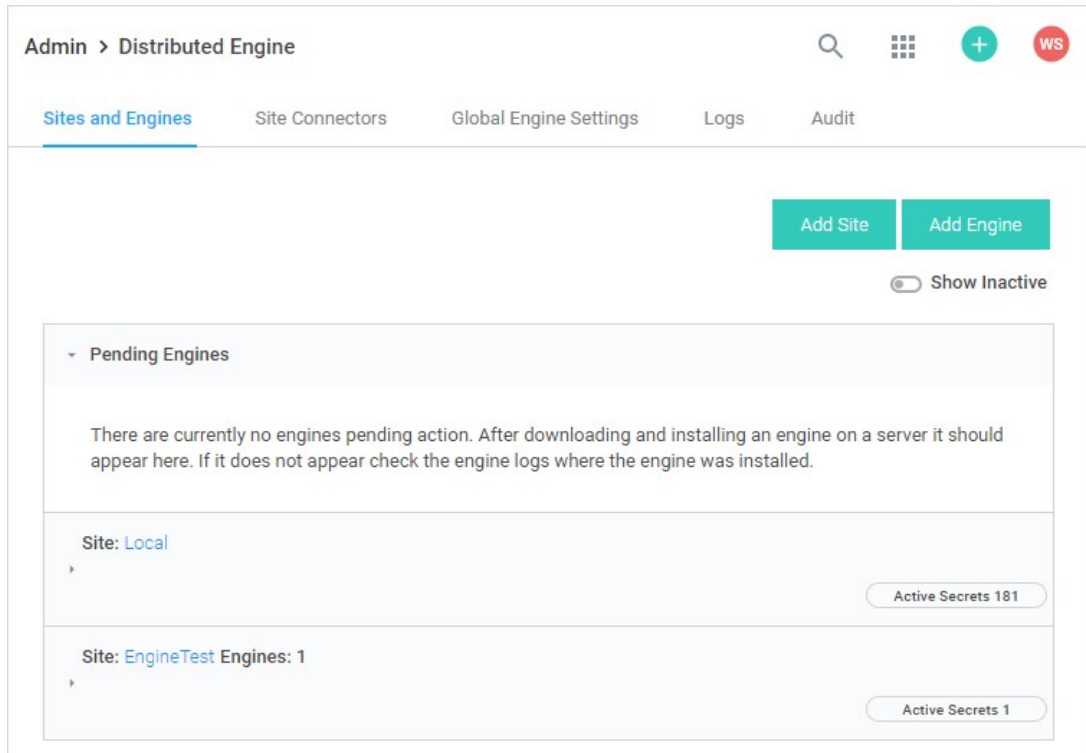
```
Enable-WSManCredSSP -Role Server
```

5. The Web server always uses a specified account to run the PowerShell scripts. Considerations:
 - Ensure that account is added to the "Remote Management Users" local group on each Web server.
 - For RPCs with custom password changers, this would be "Change Password Using," and then select "Privileged Account."
 - For PowerShell password changers in the classic UI, this would be "Run PowerShell Using" and can alternatively be configured as the "Default Privileged Account" at the template level.
 - For custom dependencies using PowerShell scripts, this would be the "Run As" secret.
 - If you use any form of extensible discovery, this account needs to be the first secret that is linked to the scanner. Any additional secrets linked to the scanner are typically associated with authentication to the destination system.

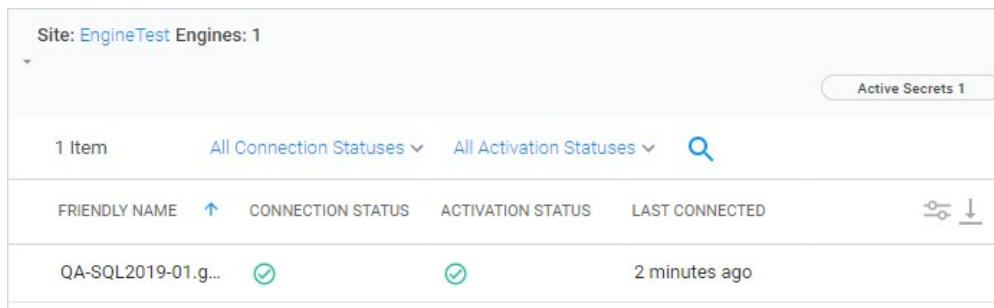
Configuring CredSSP for WinRM on a Distributed Engine

You can alternatively configure CredSSP and the credential delegation to occur from your distributed engines by changing this setting at the site level:

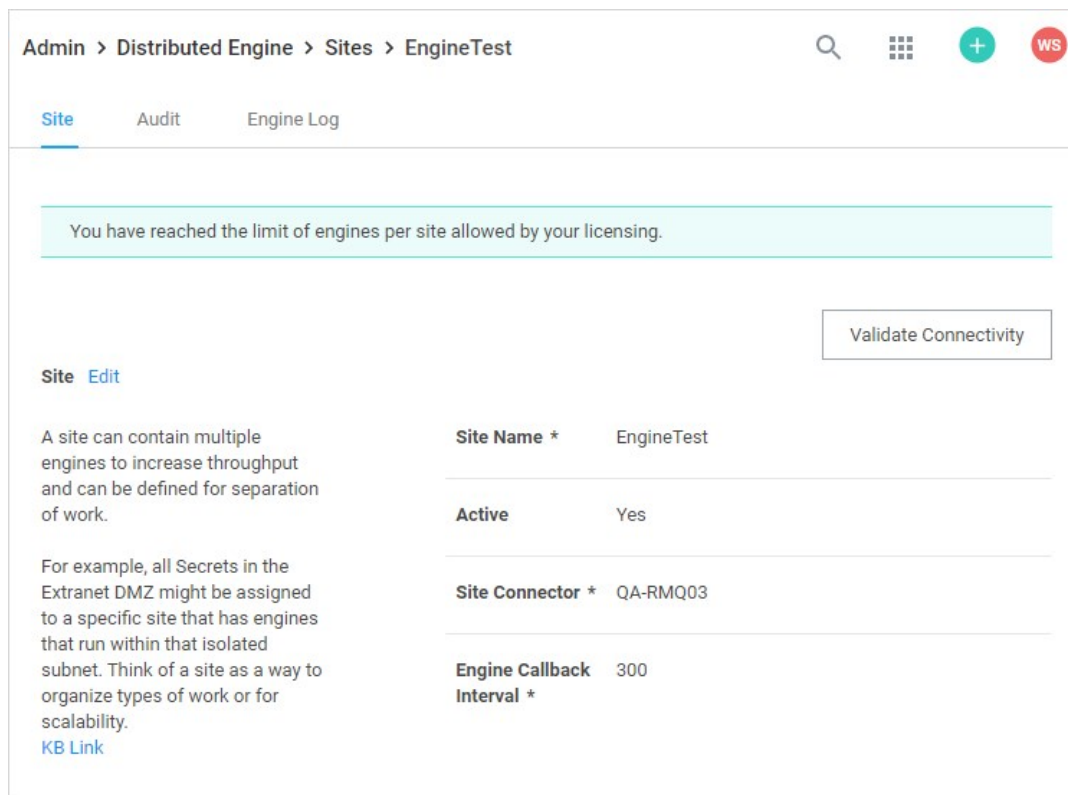
1. Go to **Admin > Distributed Engine**. The Distribute Engine Configuration page appears:



2. Click the site panel button for the desired DE. The panel expands, displaying the DEs for that site:



3. Click the site name link at the top of the panel. The site's page appears:



4. Scroll down to see the **Enable CredSSP Authentication for WinRM** listing in the **Advanced Site Configuration** section.
5. If it is not enabled, log on to each of your distributed engines where CredSSP is enabled.
6. Run Windows PowerShell as an administrator.
7. Enable client-side CredSSP by running:


```
Enable-WSManCredSSP -Role Client -DelegateComputer <distributed engine fully qualified machine name>
```

```
Enable-WSManCredSSP -Role Client -DelegateComputer <localhost>
```

Note: localhost is the actual string that the Distributed Engine is using to generate the run space. Some customers need to have both the localhost and FQDN entry. *In theory*, both entries above should be the same, thus not needing a second entry.
8. Enable server-side CredSSP by running:


```
Enable-WSManCredSSP -Role Server
```
9. The distribute engine will always use a specified account to run the PowerShell scripts. Considerations:
 - o Ensure that account is added to the "Remote Management Users" local group on each engine where CredSSP is enabled.
 - o For RPCs with custom password changers, this would be "Change Password Using," and then select "Privileged Account".
 - o For PowerShell password changers in the classic UI, this would be "Run PowerShell Using" and can alternatively be configured as the "Default Privileged Account" at the template level.
 - o For custom dependencies using PowerShell scripts, this would be the "Run As" secret.
 - o If you use any form of extensible discovery, this account needs to be the first secret that is linked to the scanner. Any additional secrets linked to the scanner are typically associated with authentication to the destination system.
10. Ensure that the "Allow Delegating Fresh Credentials" group policy setting is enabled and is not disabled by a domain policy.

1. Open the gpedit.msc file on your Secret Server machine or distributed engine, depending on where CredSSP is enabled
 2. Navigate to **Computer Settings > Administrative Templates > System > Credentials Delegation**.
 3. Edit the "Allow Delegating Fresh Credentials" setting.
 4. Verify that it is Enabled.
 5. Click "Show..."
 6. Verify that the list contains an entry that begins with "wsman/" and ends with the fully qualified machine name of the Secret Server machine or distributed engine.
 7. If destination systems are non-domain joined or on another domain without a trust, it may be required for you to add in an entry for **each** destination system you wish to run the script or do discovery on (as examples). Consider collecting a list of all destination FQDNs for your specific use case and adding them all in one go.
11. Depending on where CredSSP is configured (Web server or distributed engine), run the following commands:
- o View existing entries: `Get-Item WSMAN:\localhost\Client\TrustedHosts`
 - o Adding computers if your TrustedHosts list is empty: `Set-Item WSMAN:\localhost\Client\TrustedHosts *-Value* <ComputerName>, [<ComputerName>]`
 - o Adding computers to your existing TrustedHosts list: `$curList = (Get-Item WSMAN:\localhost\Client\TrustedHosts).value Set-Item WSMAN:\localhost\Client\TrustedHosts -Value "$curList, Server01"`
12. On the destination system, if it is on a separate domain without a trust or non-domain joined, add the reverse WSman entries so the destination system trusts either Secret Server or your engines. Run one of the following commands:
- Web server:
- ```
Set-Item WSMAN:\localhost\Client\TrustedHosts *-Value* <Web Server 1 FQDN>,<Web Server 2 FQDN>
```
- Engine:
- ```
Set-Item WSMAN:\localhost\Client\TrustedHosts *-Value* <Distributed Engine 1 FQDN>, [<Distributed Engine 2 FQDN>]
```
13. Restart either Secret Server or the engine you just trusted.

Enabling CredSSP on Secret Server Agents for PowerShell Script Dependencies

Note: Remote agents were upgraded to distributed engines in Secret Server version 8.9. This section only applies to Secret Server versions 8.8.000020 and earlier.

Note: Remote Agents are only needed for networks that are not directly connected to the network that Secret Server is installed on. If you are not using remote agents, disregard this section.

By default, Secret Server agents inherit the "Enable CredSSP Authentication for WinRM" setting from Secret Server; however, you can override this in the agent configuration file as follows:

1. On the machine running the agent, locate the the agent program files. By default, they are at C:\Program Files (x86)\Thycotic Software Ltd\Secret Server Agent.
2. Edit the SecretServerAgentService.exe.Config file in a text editor.
3. Locate the "UnencryptedSettings" section.
4. Add a new key to that section for EnableCredSSPForWinRM and set it to true. For example:

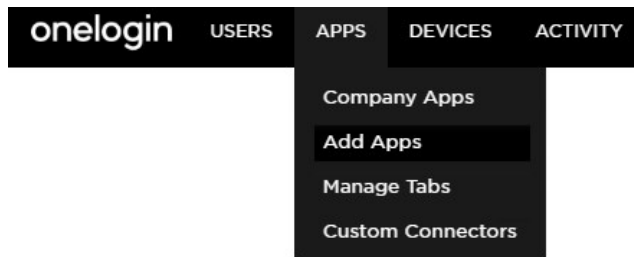
```
<add key="EnableCredSSPForWinRM" value="true" />
```

5. Restart the "Secret Server Agent" service to apply the setting.

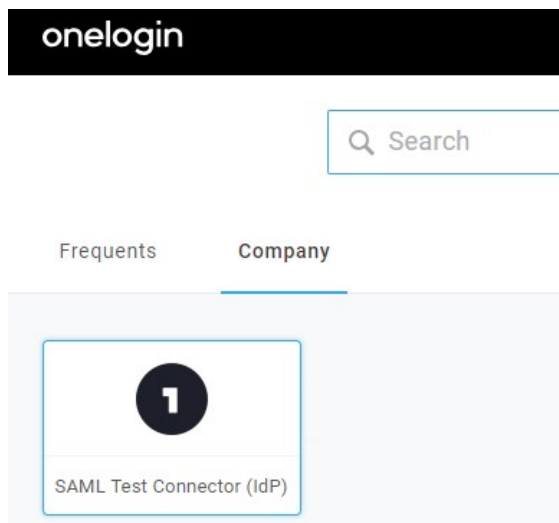
To access Secret Server using OneLogin for SAML, follow the steps below for OneLogin, then follow the steps for Secret Server.

Step One: OneLogin

1. Navigate to your OneLogin instance and log in as an administrator.
2. Select **Administration > Apps > Add Apps**.



3. Search for **SAML Test Connector (IdP)** and select it, then click **Save**.



4. Click on the **Configuration** tab and fill out the details as described below:

RelayState

Audience

Recipient

ACS (Consumer) URL Validator*

*Required. Regular expression - Validates the ACS URL when in

ACS (Consumer) URL*

*Required

Single Logout URL

- o **RelayState** can be left blank.
- o **Audience** is the name of the Service Provider configured in Secret Server (for instance "SecretServerServiceProvider").
- o **Recipient** can be left blank.
- o **ACS (Consumer) URL Validator** a required field that needs to be a valid RegEx of the ACS (Consumer) URL.

Modify the text in the example below according to the URL string of your Secret Server instance:

`https://instance.example.com/saml/vasertionconsumerservice.aspx$`

- o **ACS (Consumer) URL** like the step above, but no longer in RegEx format.

Modify the text in the example below according to the URL of your Secret Server:

`https://instance.example.com/saml/assertionconsumerservice.aspx`

- o **Single Logout URL** the Secret Server URL for SLO (Single Logout):

`https://instance.example.com/saml/sloconst.aspx`

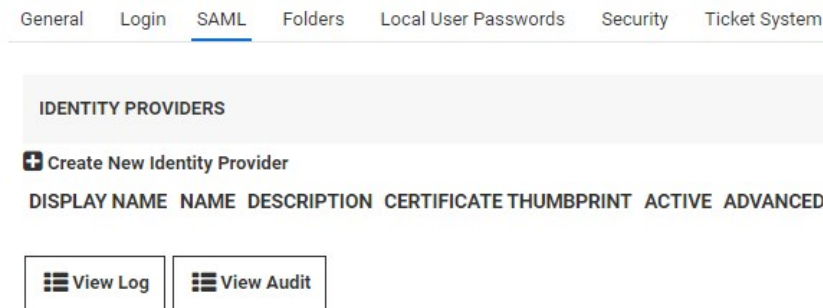
5. Click **Save** when done.

6. Click **More Actions** and **SAML Metadata** to download the metadata for OneLogin.

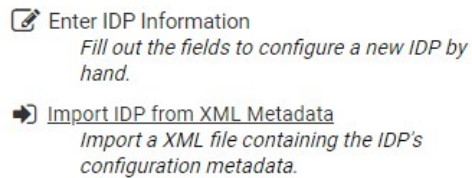


Step Two: Secret Server

1. Log into your Secret Server instance, then go to **Admin > Configuration > SAML** tab and click **Create New Identity Provider**.



2. Click **Import IDP from XML Metadata** and select the OneLogin metadata you saved previously. If you don't see the file, you may need to change the metadata filetype to .xml



3. To add users to OneLogin, navigate to OneLogin and log in as an administrator once more, then click **Administration > Users > New User**.

User Info Authentication Applications Activity

Active

First Name *

Last Name *

Email

Username

Phone Number

Manager

Company

Department

Title

4. Fill out the required information and click **Save** when finished.

Note: If you are using a Secret Server local account or Secret Server Cloud, the username will be in email format and it must be identical on OneLogin and Secret Server. For an Active Directory account, it should be the samAccountName.

5. Click on the **Applications** tab, then click the plus sign (+).

User Info Authentication Applications Activity

Applications +

6. Select **SAML Test Connector (IdP)**, then click **Continue**.

Assign New Login To Test Monster

This login will override any apps assigned via roles.

Select Application

SAML Test Connector (IdP)

7. Enter the user's Secret Server username (email format) then click **Save**.

Edit SAML Test Connector (IdP) Login For Test Monster

Enabled Allow users to sign in

NameID (fka Email)

NameID Format: Email

8. Mouse over **More Actions** and click **Change Password** to give the user a login password.

MORE ACTIONS ▾

- Assume User
- Change Password
- Force Logout
- Send Invitation
- Show User Details
- Reapply Mappings
- Delete
- Unlicense
- Download PKI cert
- Create New User
- Create New Sub User

9. In another browser or in incognito mode, log into your OneLogin instance as the user you just created. If prompted to add OneLogin to your browser, click **Skip**.
10. You should see the **SAML Test Connector (IdP)**. Click on it to authenticate into Secret Server using the SAML workflow.

The screenshot shows the OneLogin dashboard. At the top left is the "onelogin" logo. Below it is a search bar with a magnifying glass icon and the text "Search". There are two tabs: "Frequents" and "Company", with "Company" being the active tab. Under the "Company" tab, there is a card with a large black circle containing the number "1" and the text "SAML Test Connector (IdP)" below it.

Important: This topic is for Secret Server v10.5 and later and assumes you have a running Identity Service Provider (IDP) with a signed certificate.

Note: Secret Server does not support using SAML when Integrated Windows Authentication (IWA) is enabled.

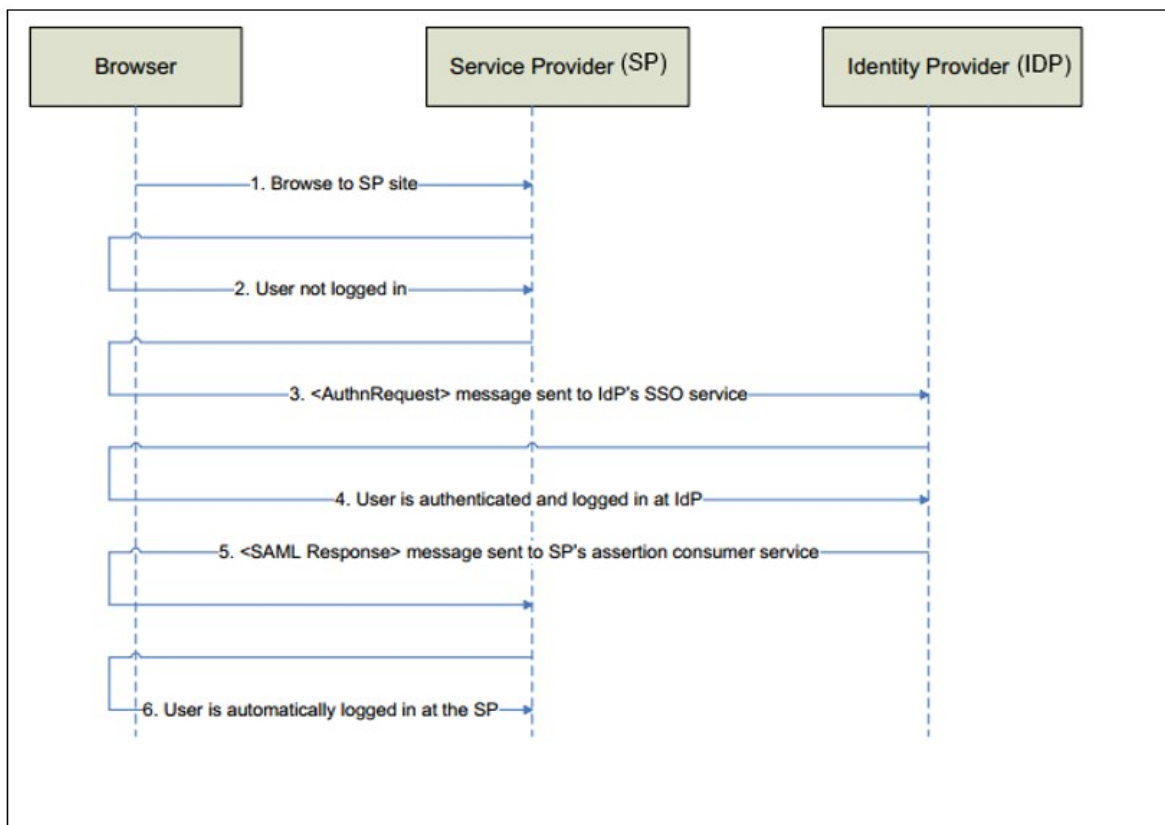
Note: This topic applies to Secret Server 10.5 and later.

SAML Overview

Secret Server allows the use of SAML Identity Provider (IDP) authentication instead of the normal authentication process for single sign-on (SSO). To do this, Secret Server acts as a SAML Service Provider (SP) that can communicate with any configured SAML IDP.

In the diagram below, Secret Server acts as the service provider. Any configured SAML IDP can be used for this process and there are several well tested providers, including OKTA, OneLogin, Azure ADFS, and Microsoft ADFS.

Figure: Secret Server as a SAML Identity Provider



Prerequisites

Licensing and Version

Secret Server Professional Edition or higher, upgraded to version 10.5 or later. To install a new SAML license, go to **Admin > Licenses > Install New License**.

.NET Framework 4.6.2+

To use SAML 2.0, you must install .NET Framework 4.6.2 or higher on your Web server. This allows Secret Server to use Microsoft's "next generation" CryptoNG API for signing SAML requests, instead of being limited to the much older CryptoAPI. This is often necessary to use modern SSL certificates and is strongly recommended as a security best practice.

To download and install the latest version of .NET Framework: See [Microsoft .NET Framework 4.8 offline Installer for Windows](#) for the latest version as of when this topic was written. If you have already installed Secret Server on the same Web server, you have already done this.

Administer Configuration SAML Role Permission

The "Administer Configuration SAML" role permission is required to use SAML to access Secret Server. To grant a user this permission from an administrator account:

1. Go to **Admin > Roles**. The Roles page appears.
2. Click the **Create New** button. The Role Edit page appears:

Role Edit

Role Name *

Enabled

Created

Permissions Assigned

Permissions Unassigned

- Access Offline Secrets on Mobile
- Add Secret
- Add Secret Custom Audit
- Administer Active Directory
- Administer Backup
- Administer Configuration
- Administer Configuration Proxying
- Administer Configuration SAML
- Administer Configuration Security
- Administer Configuration Session Recording
- Administer Configuration Two Factor
- Administer Configuration Unlimited Admin
- Administer ConnectWise Integration
- Administer Create Application Accounts
- Administer Create Users

Save Cancel

3. Type the name, such as SAML, in the **Role Name** text box.
4. Click to select the **Enabled** check box.
5. Click **Administer Configuration SAML** in the right side **Permissions Unassigned** list box.
6. Click the **<** button to move the permission to the other side.
7. Click the **Save** button. The Roles page returns.

8. Click the Assign Roles link of the newly created role. The View Role Assignment page appears:

View Role Assignment

[By Role](#) [By User Or Group](#)

Role Administrator

Save To File < 1 to 15 of 15 >

NAME	TYPE	CREATED
admin	User	5/15/2019
gamma.thycotic.com\...	User	3/30/2020
gamma.thycotic.com\...	User	3/26/2020
Developers	Group	8/9/2019
gamma.thycotic.com\...	User	4/9/2020
gamma.thycotic.com\...	User	8/9/2019
gamma.thycotic.com\...	User	8/9/2019

9. Click the **Role** dropdown list to select the role you just created.

View Role Assignment

[By Role](#) [By User Or Group](#)

Role SAML

There are no Groups or Users.

10. Click the **Edit** button. The Role Assignment page appears:

Role Assignment

i Please note that changing role assignment could remove your access to Role Administration.

By Role | By User Or Group

Role: SAML

Assigned

Unassigned

- admin
- appaccount1
- DevOps1
- Duo Approvers
- Everyone
- gamma.thycotic.com\Access Control Assistance Operators
- gamma.thycotic.com\Account Operators
- gamma.thycotic.com\Administrators
- gamma.thycotic.com
- gamma.thycotic.com
- gamma.thycotic.com
- gamma.thycotic.com Password Replication Group
- gamma.thycotic.com
- gamma.thycotic.com
- gamma.thycotic.com

[← Back](#)

11. Move the desired users to the **Assigned** list using the same method as before.
12. Click the **Save Changes** button.

Setting up Secret Server

1. Navigate to **Admin > Configuration**.
2. Click the **SAML** tab:

SAML Configuration

General Login **SAML** Folders Local User Passwords Security Ticket System Email Session Recording HSM

i SAML instructs Secret Server to trust a separate server as its Identity Provider. When SAML is enabled, the Identity Provider is responsible for asking the user for their username or password, but Secret Server will still ask the user for any configured 2-factor.

SAML GENERAL SETTINGS

SAML Enabled	No
Use Legacy SAML	No

[Edit](#)

SAML SERVICE PROVIDER SETTINGS

Name SecretServerServiceProvider
Certificate

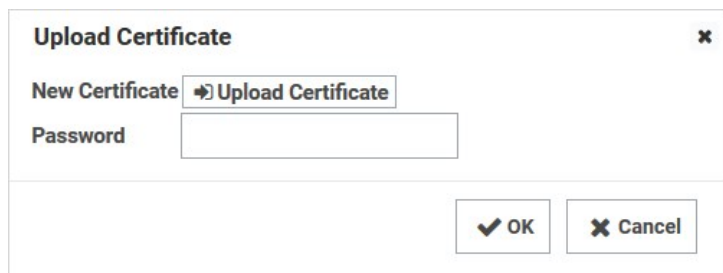
[Edit](#) [Download Service Provider Metadata \(XML\)](#)

IDENTITY PROVIDERS

[+ Create New Identity Provider](#)

DISPLAY NAME	NAME	DESCRIPTION	CERTIFICATE	THUMBPRINT	ACTIVE	ADVANCED
View Log View Audit						

3. Click the **Edit** button in the **SAML General Settings** section.
4. Click to select the **SAML Enabled** check box.
5. Click the **Save** button.
6. Under General Settings, click **Edit**, then check the **SAML Enabled** checkbox. **Save** changes.
7. Click the **Edit** button in the **SAML Service Providers** section.
8. Type a name for your Secret Server service provider, such as SecretServerServiceProvider, in the **Name** text box.
9. Click the **Select Certificate** link. The Upload Certificate popup appears:



- Click the **Upload Certificate** button to upload the certificate used for Secret Server's HTTPS configuration.

What type of certificate can be used?

- The uploaded SAML certificate requires a .pfx file format.
- For on-premises instances: The uploaded certificate should match the one used for Secret Server's HTTPS configuration, **or** it can be created as a self-signed certificate using the PowerShell script [here](#).
- For Secret Server Cloud users: Generate your own certificate using the same PowerShell script.

Note: Run the referenced PowerShell script as an administrator on a machine with .NET 4.5 or above and replace the variables in the script as directed. Your certificate is created in the directory from which you run the script. The subject name on the certificate is irrelevant, though for on-premises instances it typically matches the URL of the instance.

- Locate your certificate .pfx file and select it.
- Click the **Open** button. The new certificate appears.
- Type the access password for the private key of the certificate in the **Password** text box.
- Click the **OK** button. The certificate is uploaded and tested, and the popup disappears. The certificate now appears in the SAML Service Provider Settings section.

Note: If you have an outdated version .NET Framework (earlier than 4.6.2), you may see an error recommending you upgrade to fix the error. Reload the certificate after you do so.

- Click the **Save** button.
- Click the Create New Identity Provider link. An Identity Provider popup appears.
- Click the **Import IDP from XML Metadata** link.
- Navigate to your SecretServerSAMLMetadata.xml file and select it. This is used for uploading into your IDP, which varies by provider. Follow instructions in the following section..
- Click the Open button.

Setting up IDPs

IDP setup varies by provider. Go to the [TDP Integration site](#) for instructions for your provider.

Note: The username returned from the IDP to Secret Server within the SAML Response/Assertion's subject statement must match the desired format. The format of the username passed depends upon how the user was created within Secret Server.

Note: If AD Sync was used to create Secret Server users, the username returned from the IDP must match this format:

SecretServerUsername@ADsyncDomain OrADsyncDomain\SecretServerUsername. If using SLO, ensure that the NameID is set correctly in the IDP as an outgoing claim for the Secret Server Service Provider. If a user has different sAMAccountName and userPrincipalName in Active Directory, custom rules in the IDP can be created.

Lockout Workaround

Locked Out? Here's how you get around SSO. If during the configuration process for SAML you lock yourself (as an administrator or a user) out of Secret Server, you can log on Secret Server without using the SSO workflow by using this URL string:

[YourSecretServerInstanceName]/login.aspx?preventautologin=true

The role permission needed for this is "Bypass SAML Login," which admins have by default.

Overview

Delinea One is the single-sign-on provider for Delinea applications. With Delinea One, one user account can be granted access to multiple Delinea products, such as Secret Server, Privilege Manager, DevOps Secrets Vault, and Account Lifecycle Manager. Delinea One enables login integration using the OpenID Connect protocol, an industry standard single-sign-on method.

This article describes the Delinea One configuration options available in Secret Server.

Cloud versus On-Premise

For Secret Server Cloud, Delinea One is the default identity provider. When you set up the cloud instance, it will already be configured and ready to use Delinea One. The initial admin user will log in with their Delinea One account, and optionally, all newly created Secret Server accounts can be synchronized with Delinea One, so they can log in that way as well.

For the on-premise version of #, #Thycotic One integration is off by default, but it is supported. You can turn on Delinea One integration and configure it. For example you might want to share an identity provider between your on-premise instance, and one or more other cloud products.

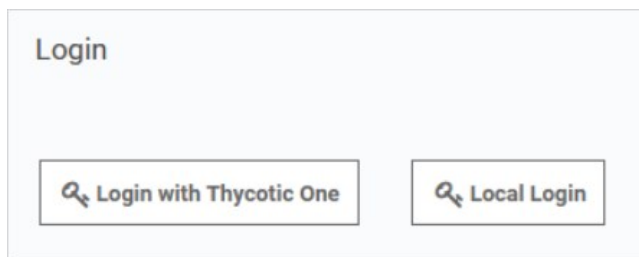
Procedures

Logging in with Delinea One

When Delinea One integration is turned on, all Secret Server users can log in either with their local passwords or with Delinea One. All Secret Server permissions and configuration will apply to that user regardless of how they logged in.

However, the local username and password and the Delinea One username and password are not necessarily the same thing. In Delinea One, you'll log in with your email address rather than your username, and the password you use may very well be different from the Secret Server password.

You'll see this on the login screen:



Clicking **Local Login** will bypass Delinea One and allow the user to log in with their local Secret Server password. Clicking **Login with Delinea One** will redirect the user to Delinea One to authenticate. Once that is successfully done, the user will be redirected back to Secret Server.

After clicking **Login with Delinea One**, users will type their email address and password:

Sign In

Email address

Next

[Create New Account](#) [Reset My Password](#)

And then be redirected back to their dashboard in Secret Server.

Configuring Delinea One Delinea

Thycotic One integration is configured on the **Admin > Configuration** page, under the **Login** tab. You can view the configuration there:

Enable Thycotic One Integration	Yes	Sync Now
Thycotic One Server URL	https://login.thycotic.com/	
Add New Users to Thycotic One	Yes	
Use Thycotic One authentication as the default	Yes	

The **Sync Now** button provides a way for you to trigger a synchronization of your Secret Server accounts with Delinea One. In most cases, you will not need to use this, as synchronization will happen on a schedule or whenever a relevant event happens, such as enabling a user or performing an Active Directory synchronization. Only active user accounts with email addresses will be synchronized.

Click **Edit** at the bottom of the page to change the configuration. The available options are slightly different between the cloud and on-premise versions of Secret Server.

Secret Server Cloud

When editing the options in Secret Server Cloud, you'll see something like this:

Enable Thycotic One Integration	<input checked="" type="checkbox"/>
Secret Server Redirect URI	https://yourcloudinstance.secretservercloud.com/signin-oidc
Thycotic One Server URL	https://login.thycotic.com/
Client Id	d9f43331-09d3-41b1-82ca-326c9c6dd419
Client Secret	< Saved >
Add New Users to Thycotic One	<input checked="" type="checkbox"/>
Use Thycotic One authentication as the default	<input checked="" type="checkbox"/>

Here are the available options:

- **Enable Delinea One Integration:** Turn on to enable Delinea One functionality. Turn off to completely disable Delinea One logins and synchronization. Make sure you have an admin account with a working local password.
- **Secret Server Redirect URI:** For informational purposes, this shows the page address to which you are redirected after you have logged in with Delinea One.
- **Thycotic One Server URL:** The Delinea One server you have connected to. There is one separate Delinea One instance in each Secret Server Cloud region.
- **Client ID:** The client ID portion of the Delinea One server credentials.
- **Client Secret:** Not shown, the client password portion of the credentials.
- **Add New Users to Delinea One:** When checked, Secret Server accounts will be synchronized with Delinea One. Adding a user will send them a welcome email, where they can set up their Delinea One account password and log into Secret Server. When unchecked, users will not be synchronized and no email will be sent. New users will not be able to log in with Delinea One, unless you click **Sync Now** on the **Admin > Configuration > Login** page, which will synchronize all active users.
- **Use Delinea One authentication as the default:** When checked, Delinea One authentication is used for the REST and SOAP APIs and mobile apps. Users who have logged in with Delinea One use their Delinea One account passwords for those activities, rather than their local Secret Server account passwords. When unchecked, they will use their local Secret Server account passwords for those activities.

In Cloud, the server URL, client ID, and client secret cannot be edited—they are set up for you when the instance is provisioned and cannot be changed.

Secret Server On-Premise

When editing the options in Secret Server on-premise, you'll see something like this:

Enable Thycotic One Integration	<input checked="" type="checkbox"/>
Secret Server Redirect URI	<code>https://mysecretserverinstance.example.com/SecretServer/signin-oidc</code>
Thycotic One Server URL	<input type="text"/>
Client Id	<input type="text"/>
Client Secret	<input type="text"/>
Add New Users to Thycotic One	<input type="checkbox"/>
Use Thycotic One authentication as the default	<input type="checkbox"/>

Unlike in Cloud, the server URL, client ID, and client secret can be edited in an on-premise instance. You can generate Delinea One credentials using Delinea's cloud management portal, Cloud Manager. Otherwise, the configuration options behave the same as in Cloud.

Generating a Delinea One Credential

To generate a credential for use in an on-premise Secret Server instance, follow the steps below:

1. From Cloud Manager, choose a Delinea One region under Other Login Options.
2. Log into Delinea One as a user that will be managing your organization's credentials. Create an account if you have not yet done so.
3. Go to Cloud Manager at <https://portal.thycotic.com/>.
4. Click **Sign In**. You are redirected to our tech support portal login.
5. Click the button for the Delinea One region you chose. Since you are already logged in to Delinea One, this will redirect you back to Cloud Manager.
6. Next, choose a team: In the menu, go to **Manage > Teams**. You may already have one if you have an existing cloud product. If not, create one. Each team can handle multiple Delinea One credentials.
7. Having selected your team, go to **Organizations**. Again, if you already have an organization, you can use it; if not, you can create one. An organization provides a way to manage the global login policies for all users.
8. Go to **Credentials**. Click **Add**. An Organization Credential dialog box appears:

Organization Credential
✕

Name

Post-Login Redirect URIs +

https://mysecretserverinstance.example.com/SecretServer/signin-oidc
✖

Post-Logout Redirect URIs +

✖

Credentials

These credentials must be added to your application (for example, Secret Server) to connect to Thycotic One.

Endpoint

Client Id

c36b17ca-3e20-438c-bfa4-f0903ea54fcf
♻️

Client Secret

806f0ddc33d46994313b857468c97318d6e1fadf73efaa00b4dc05dec31c48cc

Make a note of this value, as it cannot be retrieved once it is saved.

💾 Save

✕ Cancel

9. The available fields are as follows:

- **Name:** A description of the application using this credential, for informational purposes.
- **Post-Login Redirect URIs:** A list of valid URIs that will be allowed to authenticate with this credential. The value of "Secret Server Redirect URI" from your on-premise instance should go here. If users access your instance with more than one URI, you may want to add all of them here by clicking the + button to create additional fields. Unless an application supplies a URI that is an exact match to one of these, Delinea One will not complete the authentication.
- **Post-Logout Redirect URIs:** Secret Server does not support this feature, so this may be left blank.
- **Credentials:** The fields in this area contain the values you need to put into the Delinea One configuration in SS. Copy and paste them into the corresponding fields.

10. Once you capture all the values, click **Save**, and then save the configuration in Secret Server as well. Your instance is now fully integrated with Delinea One. If you selected the synchronization option, Secret Server will immediately sync your active users with

Delinea One, and they'll receive welcome emails describing how to continue the process.

Overview

The Federal Information Processing Standard 140-1 (FIPS 140-1) and its successor FIPS 140-2 are United States Government standards that provide a benchmark for implementing cryptographic software. Secret Server was tested and operates correctly in FIPS-compliant environments.

Note: The Microsoft .NET implementations of AES and SHA are not FIPS certified so Secret Server uses the Windows API versions for encryption functionality which *are* FIPS certified.

See [FIPS 140-2 Validation](#) for the FIPS certificate numbers for the Windows operating systems, including the algorithm implementations that we use. Supported operating systems include Windows Server 2008 R2 and above.

Procedure

To enable FIPS compliance:

Task 1: Enable FIPS in Secret Server

1. Ensure Secret Server is already installed.

Important: Secret Server is unavailable and may give errors (such as "Parser Error Message: This implementation is not part of the Windows Platform FIPS validated cryptographic algorithms") until all the steps are completed.

Important: During Secret Server installation, if FIPS compliance for Windows has already been enabled 'InvalidOperationException' error messages may result. To resolve the issue, please contact support for assistance.

Important: If FIPS is enabled as part of a domain group policy, it must be disabled before the option can be enabled in Secret Server, otherwise an error may occur. It can be re-enabled using group policy once the feature has been enabled in the application.

2. In Secret Server, go to **Admin > Configuration**.
3. Click the **Security** tab.
4. Click the **Edit** button at the bottom of the page.
5. Click to enable the **Enable FIPS Compliance** check box in the **FIPS Compliance** section.
6. Click the **Save** button.

Task 2: Enable FIPS in Windows

1. At the Windows command prompt, run `secpol.msc`. The Local Security Policy application appears.
2. In the left pane, drill down to **Security Settings > Local Policies > Security Options**.
3. In the right pane double-click the **System Cryptography: Use FIPS Compliant algorithms for encryption, hashing, and signing** policy. Its properties appear.
4. Click to enable the **Enabled** selection button on the **Local Security Setting** tab.
5. Click the **OK** button.
6. Close the **Local Security Policy** application.

Task 3: Reset the IIS Server

Run `iisreset` from the Windows command prompt. IIS resets.

Note: When using FIPS compliance mode in Secret Server, we use the NIST-certified encryption algorithms within the Windows Operating System.

Note: There should be no need to enable FIPS on the database server operating system because the encryption applies between the application and the database, not between the operating systems. Data is encrypted before it reaches the database.

Related Information

- [NIST Cryptographic Module Validation Program Information](#)
- [FIPS information for Windows](#)

Overview

Many modern secure applications use access tokens to ensure that users have access to the resources appropriate for them. Access tokens typically have a limited lifetime to ensure that information they contain or reference doesn't become stale, and to limit the time available for an attacker to use a stolen token.

When an access token expires or becomes invalid but the application still needs to access a protected resource, the application must use a new access token. To provide a new access token without requiring the user to grant permission a second time, OAuth 2.0 introduced an artifact called a *refresh token*.

Note the following:

- You cannot use a refresh token more than once.
- You cannot use a refresh token if your API Session Timeout is set to *unlimited*.
- In Secret Server, the refresh token "Time to live" equals the APISessionTimeout plus 15 minutes.
- Access tokens retrieved from REST can also be used for SOAP.

How to Enable Refresh Tokens in Secret Server

Procedure

You will receive a refresh token only if the option is enabled in **Admin > Configuration** as described below.

1. Click **Admin > Configuration** > then click the **General** tab.

The **Enable Web Services** field is visible but not editable.

Configuration

[General](#)
[Login](#)
[SAML](#)
[Folders](#)
[Local User Passwords](#)
[Security](#)
[Ticket System](#)

APPLICATION SETTINGS

Allow Automatic Checks for Software Updates Yes

Early Adopter No

[Anonymized System Metrics Information](#)

Send Anonymized System Metrics to Thycotic No [View Metric Data](#)

[View Webservices](#)

Enable Webservices No

Prevent Application from Sleeping When Idle Yes

[Syslog/CEF Logging Advanced Settings Information](#)

Enable Syslog/CEF Log Output No

[Test PowerShell with WinRM](#)

WinRM Endpoint URL http://localhost:5985/wsman

[How do I configure CredSSP for WinRM?](#)

Enable CredSSP Authentication for WinRM Yes

Secret Server Custom URL https://qa-cust-01.gamma.thy

Privilege Manager Installation URL ~/../TMS

2. Scroll to the bottom of the window, click the **Edit** button, and scroll back up. The window title changes from **Configuration** to **Edit Configuration** and the **Enable Web Services** field is now editable.

Edit Configuration

[General](#) [Login](#) [SAML](#) [Folders](#) [Local User Passwords](#) [Security](#) [Ticket System](#)

APPLICATION SETTINGS

Allow Automatic Checks for Software Updates

Early Adopter

[Anonymized System Metrics Information](#)

Send Anonymized System Metrics to Thycotic

[View Metric Data](#)

[View Webservices](#)

Enable Webservices

Prevent Application from Sleeping When Idle

[Syslog/CEF Logging Advanced Settings Information](#)

Enable Syslog/CEF Log Output

[Windows Remote Management Explanation](#)

WinRM Endpoint URL

[How do I configure CredSSP for WinRM?](#)

Enable CredSSP Authentication for WinRM

Secret View Interval Minutes

3. Check the box next to **Enable Web Services**. The menu expands and the **Enable Refresh Tokens for Web Services** field is now visible.

[View Webservices](#)

Enable Webservices

[Maximum Time Offline Explanation](#)

Maximum Time for Offline Access on Mobile Devices

Days

Hours

Session Timeout for Webservices

Unlimited

Days

Hours

Minutes

Enable Refresh Tokens for Web Services

Prevent Application from Sleeping When Idle

4. Click to enable the check box next to **Enable Refresh Tokens for Web Services**. The menu expands and the **Maximum Token Refreshes Allowed** field is now visible.

[View Webservices](#)

Enable Webservices

[Maximum Time Offline Explanation](#)

Maximum Time for Offline Access on Mobile Devices

Days

Hours

Session Timeout for Webservices

Unlimited

Days

Hours

Minutes

Enable Refresh Tokens for Web Services

Maximum Token Refreshes Allowed

5. Enter a numeral in the box next to **Maximum Token Refreshes Allowed**.
6. Scroll to the bottom of the page and click **Save**.
7. Authenticate with REST. You should receive both an access_token and a refresh_token.
8. Use the access token until it expires.

9. When the access token expires, POST to the same endpoint for authentication ("oauth2/token") with the body containing the following:

```
grant_type = "refresh_token"
```

```
Set refresh_token = <YOUR REFRESH TOKEN>
```

10. You should receive a refresh_token and a new access_token.

Example

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$uri = "https:// <yoursecretserverinstance >"
$sapi = "$uri/api/v1"

function Authenticate {
    $args= @{
        username = "username"
        password = "password"
        grant_type = "password"
    }

    echo "-----"
    echo "--Authenticate--"
    echo "-----"
    $response = Invoke-RestMethod "$uri/oauth2/token" -Method Post -Body $args -ContentType "application/json"
    $global:token = $response.access_token
    $global:refreshToken = $response.refresh_token

    $global:headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $global:headers.Add("Authorization", "Bearer $token")
}

function Refresh {
    $args= @{
        grant_type = "refresh_token"
        refresh_token = $refreshToken
    }

    echo "-----"
    echo "----Refresh----"
    echo "-----"
    echo "--Sending Refresh Token"
    echo $refreshToken

    $response = Invoke-RestMethod "$uri/oauth2/token" -Method Post -Body $args -ContentType "application/json"

    if($response.access_token){
        $global:token = $response.access_token
        $global:refreshToken = $response.refresh_token
        $global:headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
        $global:headers.Add("Authorization", "Bearer $token")
    }
}
```

Overview

An SSL (Secure Sockets Layer) certificate greatly enhances the security between the user's browser and the server your Secret Server is installed on. It encrypts all data between the server and the client's browser so if an attacker were to look at the data being transmitted between the two, they would not be able to decipher it.

Note: SSL is required when using Integrated Windows Authentication.

Obtaining an SSL Certificate

You can get a certificate from various companies such as [Thawte](#) or [VeriSign](#). If you already obtained a certificate from one of them, please follow their instructions for installing their certificates.

Note: Delinea does **not** provide certificates.

Installing a Self-Signed Certificate

You can create your own certificate for trial or sandbox environments:

Note: This requires IIS 7 or later.

Task One: Generate an IIS Self-Signed Certificate

1. Open IIS manager (**inetmgr**) on your Web server.
2. Click on the server node (one of the root nodes) in the left panel.
3. Double-click the **Server certificates** icon.
4. Click the **Create Self-Signed Certificate** link in the **Actions** panel. The Specify Friendly Name dialog box appears.
5. Type any name you desire in the **Specify a Friendly name for the certificate** text box.
6. Click the **OK** button. You now have an IIS self-signed certificate that is valid for one year. It appears under the Server Certificates panel. The certificate common name (Issued To column) is the host name of the machine running the site.

Task Two: Bind the Self-Signed Certificate to the IIS Site

1. In IIS Manager, click the server you want to bind to on the **Connections** panel tree.
2. Drill down to **Sites > Default Web Site**.
3. Click the **Bindings...** link in the **Actions** panel. The Site Bindings dialog box appears.
4. Click the **Add...** button. The Edit Site Binding dialog box appears.
5. Click the **Type** dropdown list and select **https**.
6. Click the **SSL certificate** dropdown list to select the certificate you just created.
7. Click the **OK** button. You return to the Site Bindings dialog box, where the HTTPS binding now appears.
8. Click the **Close** button. The dialog box closes.

Task Three: Test the Self-Signed Certificate

1. In a browser, go to the Website using the certificate. You should see a warning that there is an issue with the site's security certificate—specifically, the security certificate was issued for a different website's address. This occurs because IIS uses the server's name as the common name when using a self-signed certificate, which usually does not match the hostname to access the site in your browser.
2. To access the website, click the "continue to the website" link or button. You will have to do this each time you access the site. Because this is a test environment, this should not be an issue.

Note: It is possible to remove the warning by adding the self-signed certificate to the trusted root certificate authorities, but that is beyond the scope of this instruction.

Note: This applies to Secret Server Version 10.7 SP2+.

Introduction

OpenID Connect

OpenID Connect is an industry-standard single-sign-on (SSO) protocol. An identity provider implementing OpenID Connect can be used as an identity source for Secret Server, allowing users to log in with external credentials.

OpenID Connect Support in Secret Server

Secret Server

Secret Server implements OpenID Connect authorization code flow, allowing any standards-compliant provider to be used as an identity source. To use OpenID Connect, a Secret Server administrator must configure the login integration. Additionally, user accounts must be created in Secret Server that correspond to the external accounts.

Prerequisites

General

- Secret Server version 10.7 SP2+
- An OpenID-Connect-compatible identity provider, such as Delinea One, Azure AD, Auth0, or Okta.

Permissions

- To configure the login integration, a Secret Server user must have Administer OpenID Connect permissions. Secret Server
- To add user accounts that can be used with OpenID Connect, a Secret Server user must have administer user permissions.

Task One: Acquire and Configure an OpenID Connect Provider

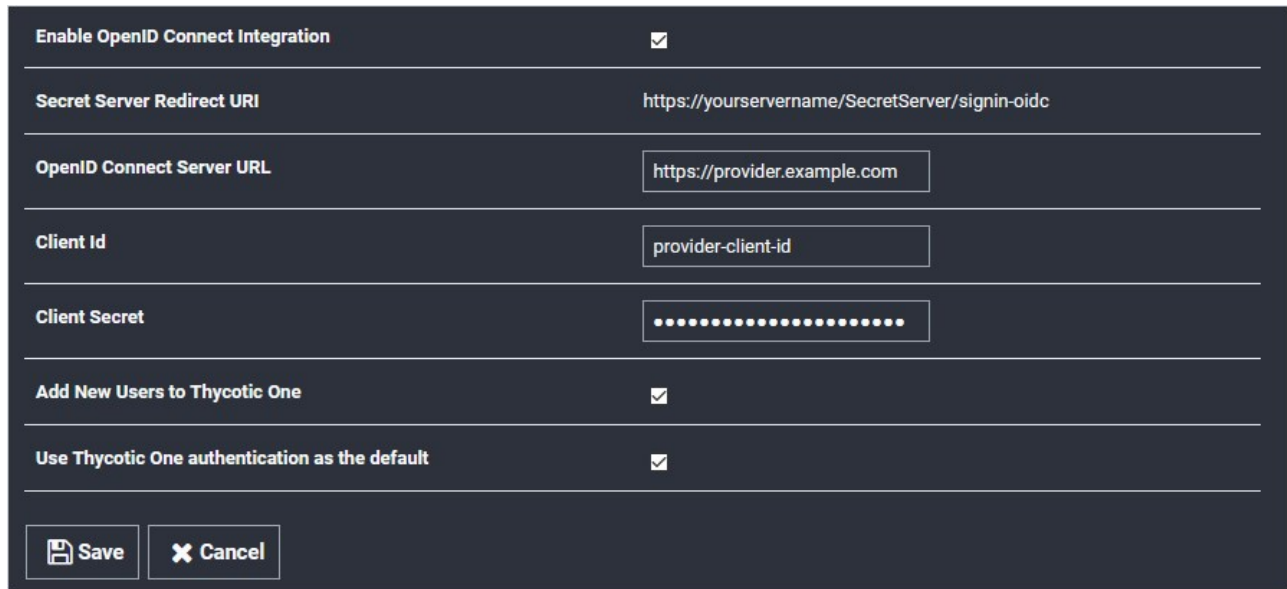
1. Follow your OpenID Connect provider's setup instructions for configuring a new identity client.
2. Gather the configuration data from the provider. To configure an OpenID Connect login provider, the provider must supply these:
 - **Provider URL:** An HTTPS URL, acting as an authentication endpoint. Secret Server expects the provider URL to be the "issuer" URL of the provider. For example, if the OpenID Connect configuration for the provider is accessible at
`https://example.com/.well-known/openid-configuration,`
then the URL used in the Secret Server configuration should be
`https://example.com.`
 - **Client ID:** The ID portion of the credentials used to interact with the provider.
 - **Client Secret:** The password portion of the credentials used to interact with the provider.

The process for determining this information varies by provider and you can usually find it by following the provider-specific documentation for configuring an OpenID Connect client.

3. Configure the provider with the Secret Server Redirect URI, as shown below, when configuring OpenID Connect integration in Secret Server. It is usually `https://YourWebServer/SecretServer/signin-oidc`. The Secret Server Redirect URI value must be added to the provider's configuration, so that the Secret Server instance can perform the authentication handshake. This process varies by provider, but it is usually known as a post-login redirect URI or callback URI.

Task Two: Configure Secret Server

1. Locate the Secret Server OpenID Connect configuration at **Admin > Configuration > Login**:



The screenshot shows a configuration form for OpenID Connect integration. The form is dark-themed with white text and input fields. It includes the following fields and controls:

- Enable OpenID Connect Integration**: A checked checkbox.
- Secret Server Redirect URI**: A text field containing the value `https://yourservername/SecretServer/signin-oidc`.
- OpenID Connect Server URL**: A text input field containing `https://provider.example.com`.
- Client Id**: A text input field containing `provider-client-id`.
- Client Secret**: A text input field filled with 15 dots, representing a masked secret.
- Add New Users to Thycotic One**: A checked checkbox.
- Use Thycotic One authentication as the default**: A checked checkbox.
- At the bottom, there are two buttons: **Save** (with a floppy disk icon) and **Cancel** (with an 'X' icon).

2. Click to select the **Enable OpenID Connect Integration** check box.
3. Fill in the other values using the what you gathered earlier.

Note: The two Delinea One options are not relevant unless you use Delinea One as your OpenID Connect provider. They have no effect for other providers.

4. Click the **Save** button.

Task Three: Matching External Accounts to Secret Server Users


Ensure a matching user already exists in Secret Server, which is required when logging on an OpenID Connect account. OpenID provides for no user list synchronization or on-the-fly account creation. Users are matched according to the claims provided in their authentication ticket. Matching occurs using the following criteria:


- If the name identifier value matches an active user that has already logged in with OpenID Connect, then this user will be logged on.
- If not, then if the email or UPN values match an existing, unique, active Secret Server user, then this user will be logged on.

Otherwise, the login attempt will fail, and information about the failure will be added to the system log.

Task Four: Logging on with OpenID Connect

Finally, confirm the configuration by logging on with OpenID Connect. Once Open ID Connect has been configured, the Secret Server login page will have a new **Login with OpenID Connect** button:

 Login with OpenID Connect

 Local Login

Clicking this button initiates the OpenID Connect log on process with the external provider. Depending on provider settings, you may be asked to approve the login request or grant access to specific profile info. Once you have approved the request and logged in to the external provider, you will be redirected back to Secret Server and logged in as the corresponding local user.

Secret Server provides the option to integrate your SAML implementation to automatically authenticate users to the application:

- To configure SAML for versions 10.5+, see the [SAML 2.0 Configuration Guide](#).
- To configure SAML for versions 10.2-10.4, see the [SAML Configuration Guide for Secret Server 10.2-10.4](#).

Host SSH key verification is supported for use with heartbeat, proxied launchers, password changers, and discovery. Host SSH key verification can be used to ensure that the machine you are connecting to is a trusted host. Host SSH key verification will not pass credentials to the target machine unless the public key digest matches the SHA1 digest that Secret Server has on file. This helps prevent man-in-the-middle attacks.

How to Map a Server SHA1 Digest to a Secret

To configure host SSH key verification:

1. go to Secret Templates and add a field for the host's SSH key digest.
2. Click **Configure Extended Mappings**.
3. Add a "Server SSH Key" mapping to your newly created SSH key digest field.
4. On your secrets, add the SSH Key digest of the hosts to your digest field. Verification takes effect the next time you connect to the host.

Heartbeat

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then heartbeat will fail with a "UnableToValidateServerPublicKey" error. The heartbeat log will show the expected and actual values for the SHA1 digest.

Password Changing

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then the password change will fail. The Remote Password Changing log will show the expected and actual values for the SHA1 digest.

Non-Proxied Launcher

When launching PuTTY, it displays a message if the server's public key digest is not yet trusted.

Proxied Launcher

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then a message will be written to PuTTY displaying the expected and actual values for the SHA1 digest. The credentials from the secret will not be passed to the target machine.

SSH Script Dependencies

SSH Script dependencies now have a "Server Key Digest" field. When this field is blank, the server's digest will not be checked. When it is filled in, if it does not match, an error is returned indicating the expected and actual values from that server. No credentials are passed to the target machine unless the digest matches.

Unix Account Discovery

To validate SHA1 server digests for Unix account discovery, create a file named `KeyDigests.txt` in the root of the Secret Server website. Each line should contain an IP address or other computer identifier, a comma, and then the SHA1 digest (see example below). When the file exists and has data, all machines to be scanned must match one of the SHA1 hashes in the file. Any computers that do not match will still show up on the Discovery Network View page, but authenticated scanning will not take place (no credentials will be passed to the machine, and

accounts will not be retrieved from the machine).

Sample KeyDigests.txt:

```
192.168.1.5:7E:24:0D:E7:4F:B1:ED:08:FA:08:D3:80:63:F6:A6:A9:14:62:A8:15  
apollo,7A:25:AB:38:3C:DD:32:D1:EA:86:6E:1C:A8:C8:37:8C:A6:48:F9:7B
```

Be sure that your digest value which you input into KeyDigests.txt is not an MD5 value. MD5 values are 32 bytes, whereas a SHA1 is 40. So the above is correct whereas if you obtain a digest of your public key file and the result has 32 bytes, this is likely an MD5. One command which could be used to obtain the SHA1 digest of the public key file for use in KeyDigests.txt is:

```
awk '{$print $2}' id_XXX.pub | openssl base64 -d -A | openssl sha1
```

After which, you should add : to every two characters to the output such that it matches the above. We will try to connect to the keys which we have the strongest preference for first in the event of multiple keys and it is legal to have multiple digests for the same IP address or hostname in the file. In the event of multiple keys on your system, it is usually correct to get the digest from /etc/ssh/ssh_host_rsa_key.pub and put it into KeyDigests.txt if this public key exists.

Starting with version 10.4, Secret Server allows you to define a policy for validating X509 Certificates. This applies to all Active Directory domains using LDAPS. It also applies to any connections to syslog servers over TLS. Certificates that do not meet the policies specified in Secret Server are rejected, denying connections to the server. All certificate validation failures are logged in the security audit log, which is available by going to **Admin > See All** and then **Security Audit Log**.

Setting the Certificate Verification Policy

To set a verification policy:

1. Go to **Admin > Configuration**.
 2. Click the **Security** tab.
 3. Click the **Edit** button
 4. Click to select the **Apply TLS Certificate Chain Policy and Error Auditing** check box. The TLS Auditing options appear:
-

TLS AUDITING

Apply TLS Certificate Chain Policy and Error Auditing

Ignore Certificate Revocation Failures

Additional Certificate Chain Policy Options ?
[What are X509 Certificate Chain Policy Options?](#)

X509RevocationMode.NoCheck

Enable TLS Debugging and Connection Tracking

[Advanced \(not required\)](#)

i Secret Server's IIS AppPool must be granted permission to use the Client Certificate, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). Example usage:
`winhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "Certificate Subject" -a "HOSTNAME\IIS APPPOOL\SecretServer"`
[Download WinHttpCertCfg - Official WinHttpCertCfg documentation](#)

Client Certificate Thumbprint(s) ?

Enter Client Certificate Thumbprint Ids ...

5. To change the policy, type a semi-colon delimited list of policy options in the **Additional Certificate Chain Policy Options** text box. To use a policy, enter the <full_enumeration_name>.<enumeration_item>. For example, to validate the entire certificate chain, add X509RevocationFlag.EntireChain to the semi-colon delimited list of options. See [Certificate Validation Options](#) for details.
6. If you wish to ignore certificate revocation warnings and allow revoked certificates, click to select the **Ignore Certificate Revocation Failures** check box.

Certificate Validation Options

The following Microsoft enumerations are the available certificate chain policy options. For detailed descriptions of each option, see the linked documentation.

X509RevocationMode

Specifies the mode used to check for X.509 certificate revocation.

NoCheck	0	No revocation check is performed on the certificate.
Offline	2	A revocation check is made using a cached certificate revocation list (CRL).
Online	1	A revocation check is made using an online certificate revocation list (CRL).

[Unexpected Link Text](#)

See [X509RevocationMode Enum](#) for details.

X509RevocationFlag

Specifies which X.509 certificates in the chain should be checked for revocation.

EndCertificateOnly	0	Only the end certificate is checked for revocation.
EntireChain	1	The entire chain of certificates is checked for revocation.
ExcludeRoot	2	The entire chain, except the root certificate, is checked for revocation.

[Unexpected Link Text](#)

See [X509RevocationFlag Enum](#) for details.

X509VerificationFlags

Specifies conditions under which verification of certificates in the X.509 chain should be conducted. These values can be bitwise combined to indicate multiple flags.

AllFlags	4095	All flags pertaining to verification are included.
AllowUnknownCertificateAuthority	16	Ignore that the chain cannot be verified due to an unknown certificate authority (CA).
IgnoreCertificateAuthorityRevocationUnknown	1024	Ignore that the certificate authority revocation is unknown when determining certificate verification.
IgnoreCtlNotTimeValid	2	Ignore that the certificate trust list (CTL) is not valid, for reasons such as the CTL has expired, when determining certificate verification.
IgnoreCtlSignerRevocationUnknown	512	Ignore that the certificate trust list (CTL) signer revocation is unknown when determining certificate verification.
IgnoreEndRevocationUnknown	256	Ignore that the end certificate (the user certificate) revocation is unknown when determining certificate verification.

IgnoreInvalidBasicConstraints	8	Ignore that the basic constraints are not valid when determining certificate verification.
IgnoreInvalidName	64	Ignore that the certificate has an invalid name when determining certificate verification.
IgnoreInvalidPolicy	128	Ignore that the certificate has invalid policy when determining certificate verification.
IgnoreNotTimeNested	4	Ignore that the CA (certificate authority) certificate and the issued certificate have validity periods that are not nested when verifying the certificate. For example, the CA cert can be valid from January 1 to December 1 and the issued certificate from January 2 to December 2, which would mean the validity periods are not nested.
IgnoreNotTimeValid	1	Ignore certificates in the chain that are not valid either because they have expired or they are not yet in effect when determining certificate validity.
IgnoreRootRevocationUnknown	2048	Ignore that the root revocation is unknown when determining certificate verification.
IgnoreWrongUsage	32	Ignore that the certificate was not issued for the current use when determining certificate verification.
NoFlag	0	No flags pertaining to verification are included.

[Unexpected Link Text](#)

See [X509VerificationFlags Enum](#) for details.

Troubleshooting

If you enable certificate policy validation and logging, you may have server connections rejected due to certificates that violate the set policies. These errors are recorded in the security audit log. If the information logged there is not enough to determine why a certificate was rejected, you can get additional log details by enabling TLS Debugging. This adds detailed information to the logs about each certificate checked.

Due to the possibility of exposing sensitive information in the logs, TLS debugging requires two steps to enable:

1. Click to select the **Enable TLS Debugging and Connection Tracking** check box.
2. Change the global logging level to DEBUG. To do this, edit the `web-log4net.config` file in the root folder of your Web application. Follow the comments in the file to comment out the current log level line (the default is INFO), and uncomment the line that sets the value to DEBUG.

Important: Only enable TLS debugging when you are actively troubleshooting a certificate validation issue. Disable this option when you are not to prevent logging of certificate details.

Note: Please click the table of contents on the left to see any sub-pages to this one.

Windows integrated authentication allows Active Directory users that are synced with Secret Server to log into workstations and be automatically authenticated to the application. A user's Active Directory credentials are automatically passed through to IIS, logging them into the site.

Configuring Integrated Windows Authentication

Note: This article applies to Secret Server On-Premises only.

Introduction

Integrated Windows Authentication (IWA) allows users to log into Secret Server automatically if they are logged into a workstation with their Active Directory credentials.

Note: When using IWA, see [Using Mobile Devices with Windows Authentication Enabled](#) to connect mobile applications to Secret Server.

Note: [Secure LDAP](#) only works with Integrated Windows Authentication in Server 2008 R2 and later.

Setting Up Windows Authentication

Task 1: Configuring Secret Server

1. Log into Secret Server as a user with Active Directory administration privileges.
2. Navigate to **Administration > Active Directory**:

Active Directory Integration	
Enable Active Directory Integration	Yes
Enable Integrated Windows Authentication	No

Active Directory User Synchronization	
Enable Synchronization of Active Directory	Yes
Synchronization Interval for Active Directory	1 hour
User Account Options	User status mirrors Active Directory (Automatic)

Buttons: Back, Edit, Edit Domains, Edit Synchronization, View Audit

3. Click the **Edit** button. The Edit Active Directory Configuration page appears:

Edit Active Directory Configuration

Active Directory Integration

Enable Active Directory Integration

Enable Integrated Windows Authentication *Requires advanced IIS settings (See [KB Article](#))*

Active Directory User Synchronization

Enable Synchronization of Active Directory *Enable to synchronize users by Active Directory Group*

Synchronization Interval for Active Directory

Days:

Hours:

Minutes:

User Account Options:

4. If necessary, click to select the following check boxes:

- Enable Active Directory Integration
- Enable Synchronization of Active Directory
- Enable Integrated Windows Authentication.

5. Select your desired option from the **User Account Options** dropdown list.

6. Type the in the **Days**, **Hours**, and **Minutes** text boxes to choose a synchronization interval, which is how often Secret Server pulls in users from AD.

7. Click the **Save** button. The Active Directory Configuration page reappears:

Active Directory Configuration

Active Directory Integration

Enable Active Directory Integration Yes

Enable Integrated Windows Authentication Yes

Active Directory User Synchronization

Enable Synchronization of Active Directory Yes

Synchronization Interval for Active Directory 1 hour

User Account Options User status mirrors Active Directory (Automatic)

8. Click the **Edit Domains** button. The Active Directory Domains page appears:

Active Directory Domains

Save To File < 1 to 1 of 1 >

Domain	Friendly Name	Active	Login Enabled	Use LDAPS
testparent.thycotic.com	TestParent	Yes	Yes	No

[Back](#)
[+ Create New](#)

- Click the **Create New** button. The Active Directory Domain page appears:

Active Directory Domain

Credentials

Fully Qualified Domain Name *

Friendly Name *

Active

Allow Logins From Domain

Sync Secret [No Selected Secret](#) [Create New Secret](#)

Site

Enable Discovery Discovery is not enabled and will not be run. To enable it, go to the Discovery section under Administration.

[Advanced \(not required\)](#)

[Save And Validate](#)
[Cancel](#)

- Type the domain name for single-sign-on in the **Fully Qualified Domain Name** text box.
- Type the human-friendly name in the **Friendly Name** text box.
- Click the **Save and Validate** button. The Active Directory Configuration page reappears:

Active Directory Configuration

Active Directory Integration

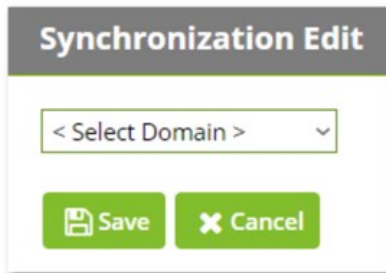
Enable Active Directory Integration Yes
Enable Integrated Windows Authentication Yes

Active Directory User Synchronization

Enable Synchronization of Active Directory Yes
Synchronization Interval for Active Directory 1 hour
User Account Options User status mirrors Active Directory (Automatic)

[Back](#)
[Edit](#)
[Edit Domains](#)
[Edit Synchronization](#)
[View Audit](#)

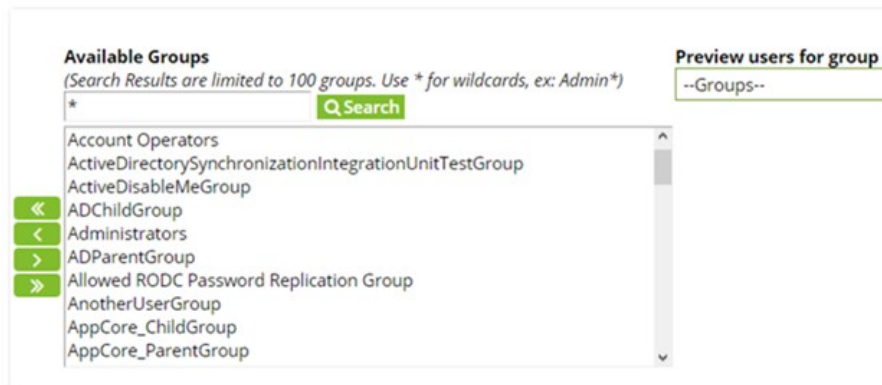
- Click the **Edit Synchronization** button. The Synchronization Edit page appears:



- Click the dropdown list to select the desired domain. The page changes to show the groups for that domain:



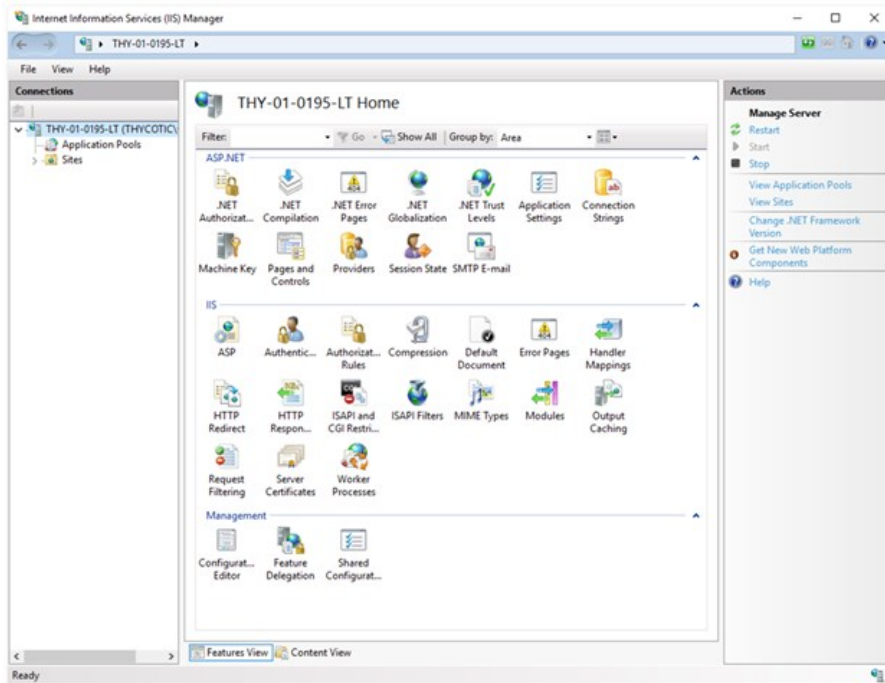
- Use the **Available Groups** text box and **Search** button to locate your desired groups. The matching groups appear in the list:



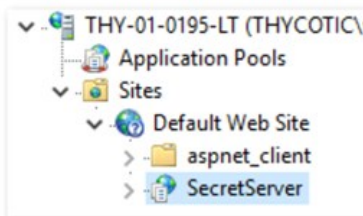
- Select the desired groups and click the << button to move them into the **Synchronized Groups** list.
- Click the **Save** button. The Active Directory Configuration page reappears.
- Click the **Synchronize Now** button in the **Messages** section. This pulls all the users of the specified groups into Secret Server.

Task 2: Configuring IIS

- Start the Internet Information Services (IIS) Manager:



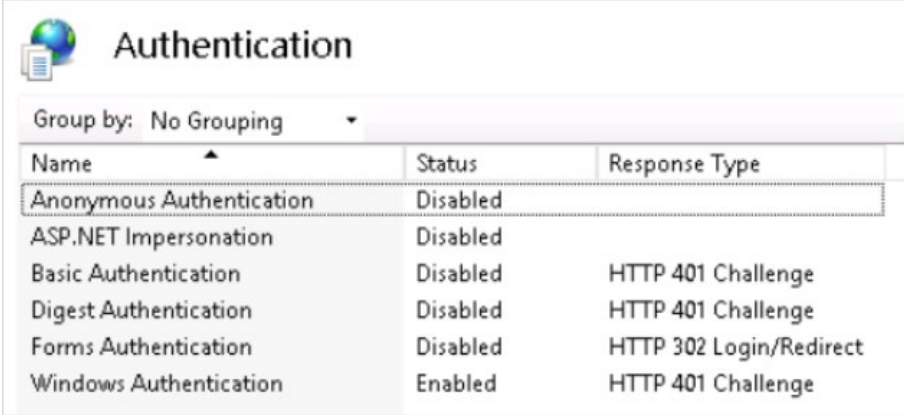
2. Navigate to and select your Secret Server website in the **Connections** tree:



3. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
4. Enable the **Windows Authentication** parameter by right-clicking it and selecting **Enable**. For now, ignore the alert if it appears in the Alert section.

Note: If Windows Authentication is not visible, ensure that the Windows Authentication Role service is enabled in Windows. This is different than earlier versions.

5. Disable the **Anonymous Authentication**.
6. Disable the **Forms Authentication**. The alert in the Alert section should disappear.
7. When finished, the Authentication settings should look like this:



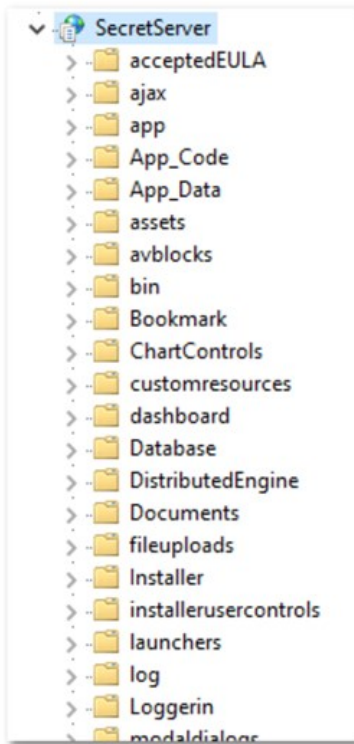
Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

- Restart your IIS server with an `iisreset` command.
- On the Secret Server folder, ensure users have read or higher permission, and ensure the security settings are set to be inherited by child objects. Because Secret Server impersonates those users, they require access to Secret Server files.
- Log in to the Secret Server site from an authenticated workstation.

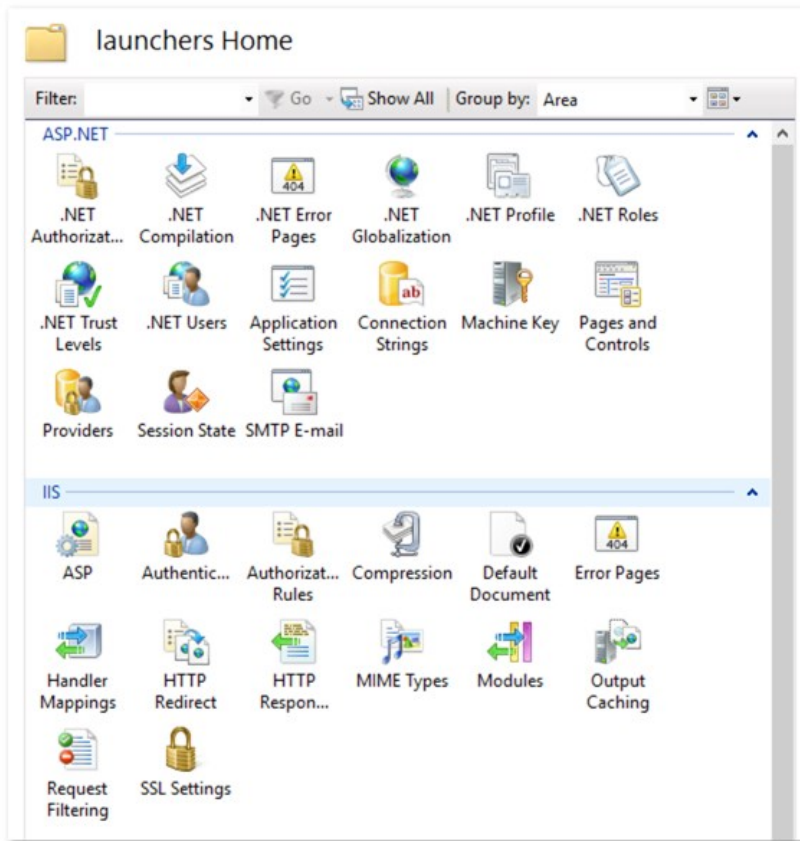
Task 3: Configuring Secret Server Launchers

By default, a launcher will not work when using IWA, resulting in an HTTP 401: Unauthorized error. If this is an issue, ensure Secret Server is on Windows Server 2008 or later and complete the following steps:

- Open IIS and browse to your Secret Server application.
- Click the **>** to see the application's folders:



3. Click to select the **launchers** folder. The launchers Home panel appears:



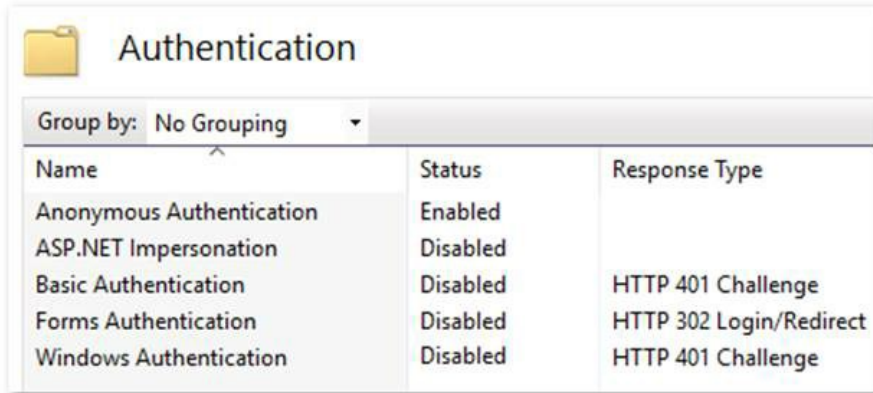
4. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
5. Ensure the **Anonymous Authentication** is set to **Enabled**.
6. Ensure the **Windows Authentication** is set to **Disabled**.
7. Ensure all others are disabled. When you are finished, the settings should look like this:

The screenshot shows the 'Authentication' pane in Windows. The 'Group by' is set to 'No Grouping'. The table below shows the status of various authentication methods.

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

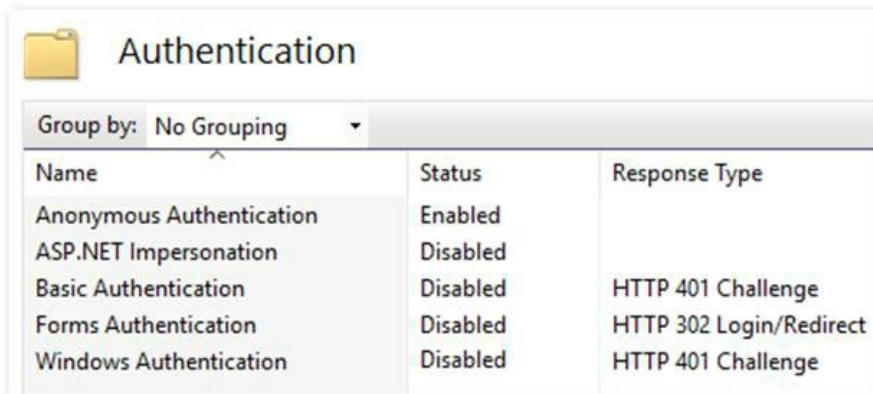
8. Click the **webservices** folder.
9. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.

10. Ensure the **Anonymous Authentication** is set to **Enabled**.
11. Ensure the **Windows Authentication** is set to **Disabled**.
12. Ensure all others are disabled. When you are finished, the settings should look like this:



Authentication		
Group by: No Grouping		
Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

13. Click the **rdp** folder.
14. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
15. Ensure the **Anonymous Authentication** is set to **Enabled**.
16. Ensure the **Windows Authentication** is set to **Disabled**.
17. Ensure all others are disabled. When you are finished, the settings should look like this:

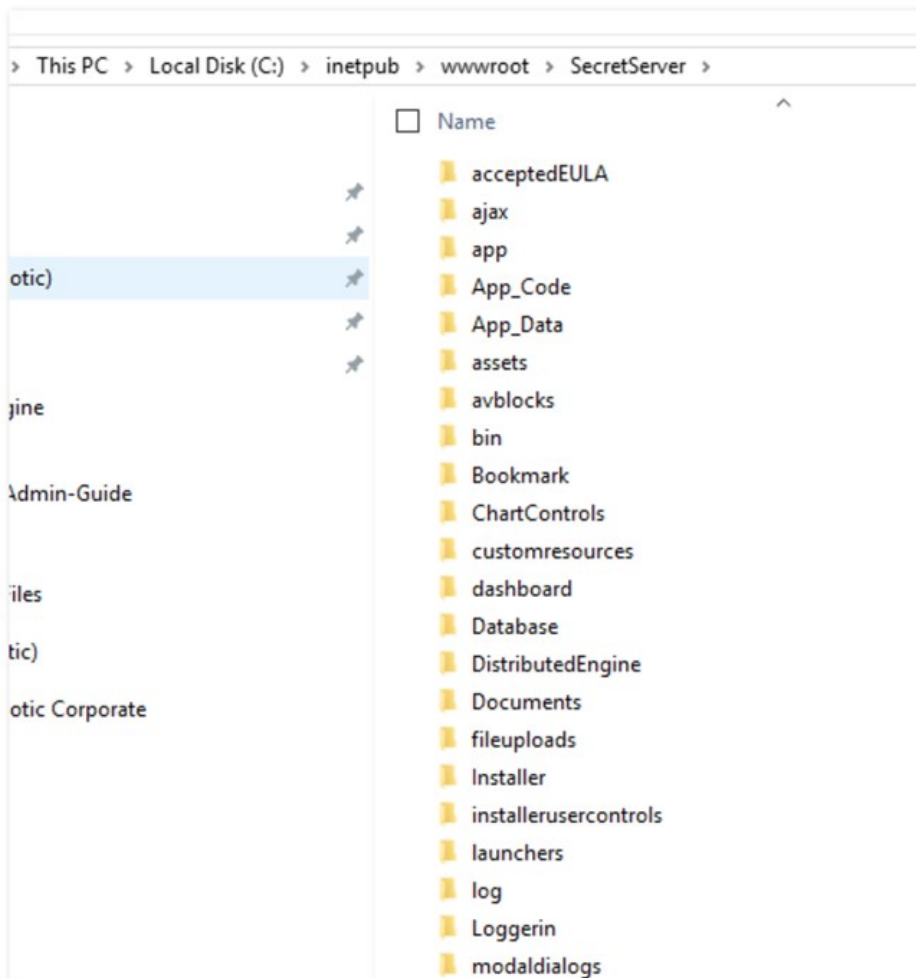


Authentication		
Group by: No Grouping		
Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

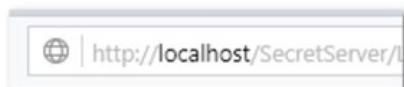
Task 4: Configuring Distributed Engines

Similarly, Secret Server with distributed engines will not work with IWA by default. If this is an issue, complete the following:

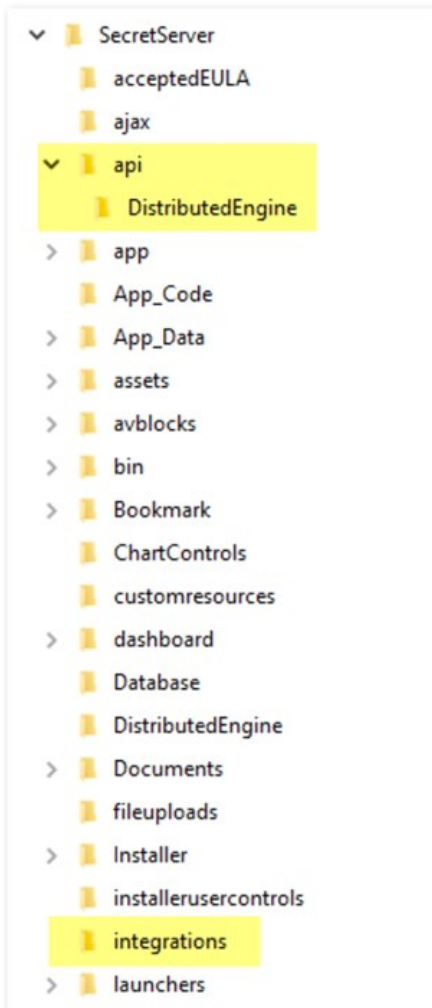
1. In Windows Explorer, navigate to the ...\\SecretServer\ folder:



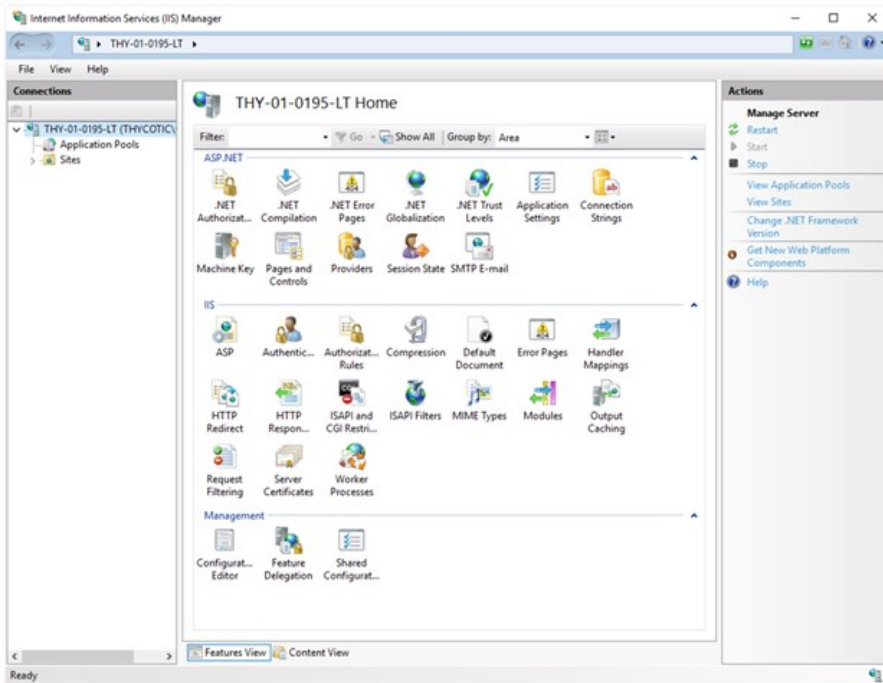
This folder is mapped to your SecretServer folder in your webserver:



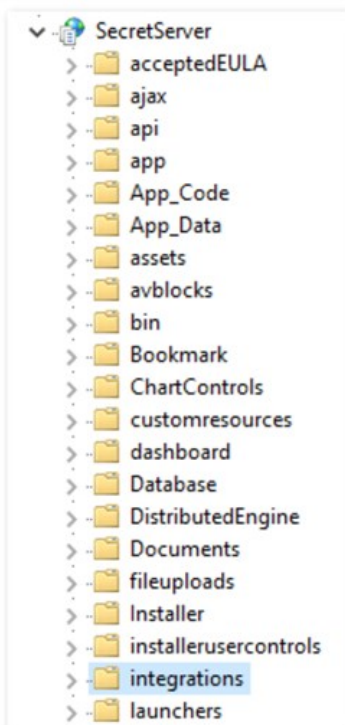
2. Create a subfolder named ...\SecretServer\integrations.
3. Create a subfolder called ...\SecretServer\api in the same location.
4. In your ...\SecretServer\api folder, create a subfolder named ...\SecretServer\api\DistributedEngine.
5. When you are finished, the new folders appear as follows:



6. Start IIS Manager:



7. Navigate the **Connections** tree back to **integrations** folder in the **SecretServer** node:



8. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.

9. Ensure the **Anonymous Authentication** is set to **Enabled**.

10. Ensure the **Windows Authentication** is set to **Enabled**.
11. Ensure all others are disabled. When you are finished, the settings should look like this:

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

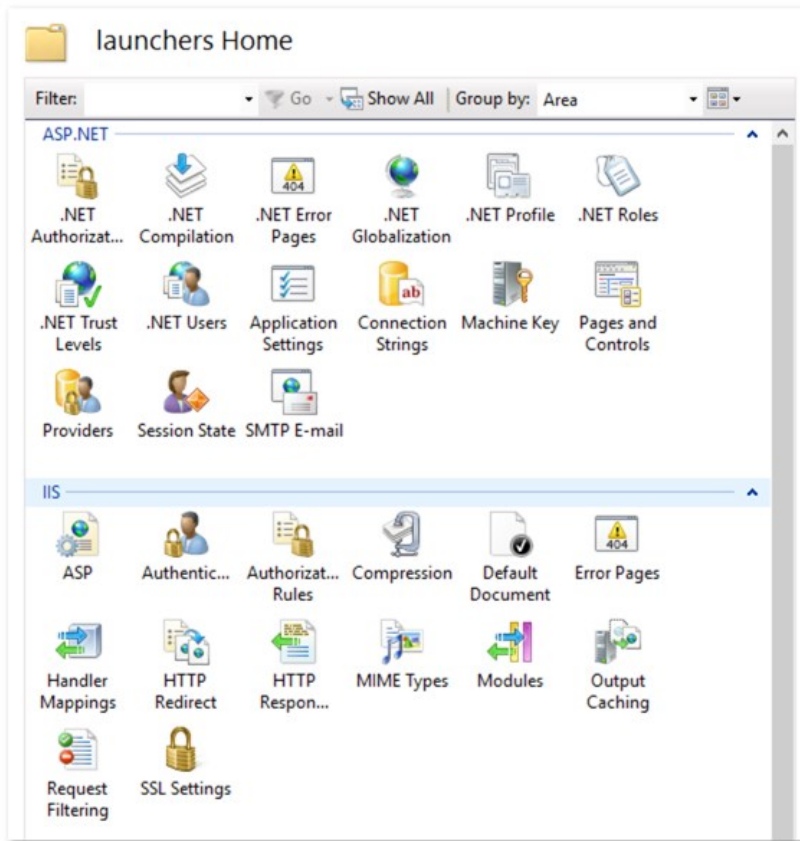
12. Navigate to the ...\\SecretServer\\api\\DistributedEngine folder.
13. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
14. Ensure the **Anonymous Authentication** is set to **Enabled**.
15. Ensure the **Windows Authentication** is set to **Disabled**.
16. Ensure all others are disabled. When you are finished, the settings should look like this:

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

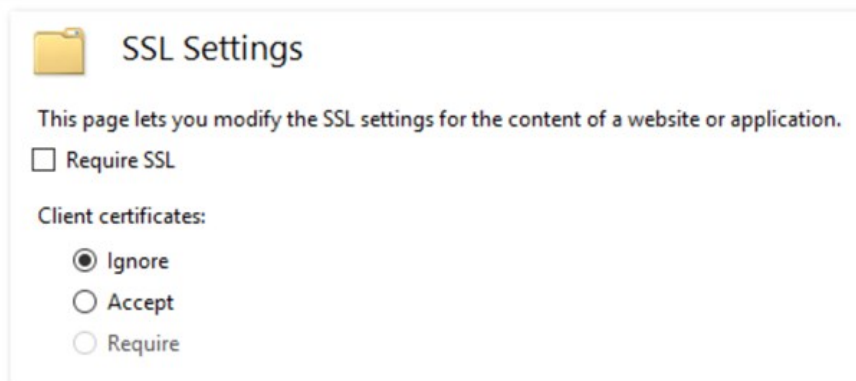
Task 5: Configuring Client Certificates

If you are using client certificates, configure the following in IIS for launchers to work:

1. Click to select the **launchers** folder. The launchers Home panel appears:



2. Double-click the **SSL Settings** icon. The settings panel appears:



3. Click to set the **Client Certificates** selection button to **Accept**.
4. Click to select the **Webservices** folder.
5. Once again, double-click the **SSL Settings** icon.
6. This time, set the **Client Certificates** selection button to **Ignore**.

Note: If you are not automatically logged in to Secret Server after setting up IWA, IIS may not be handling the credentials correctly. To fix this, recreate the web site in IIS.

Note: When testing IWA, keep in mind the requirements at [Internet Explorer May Prompt You for a Password](#).

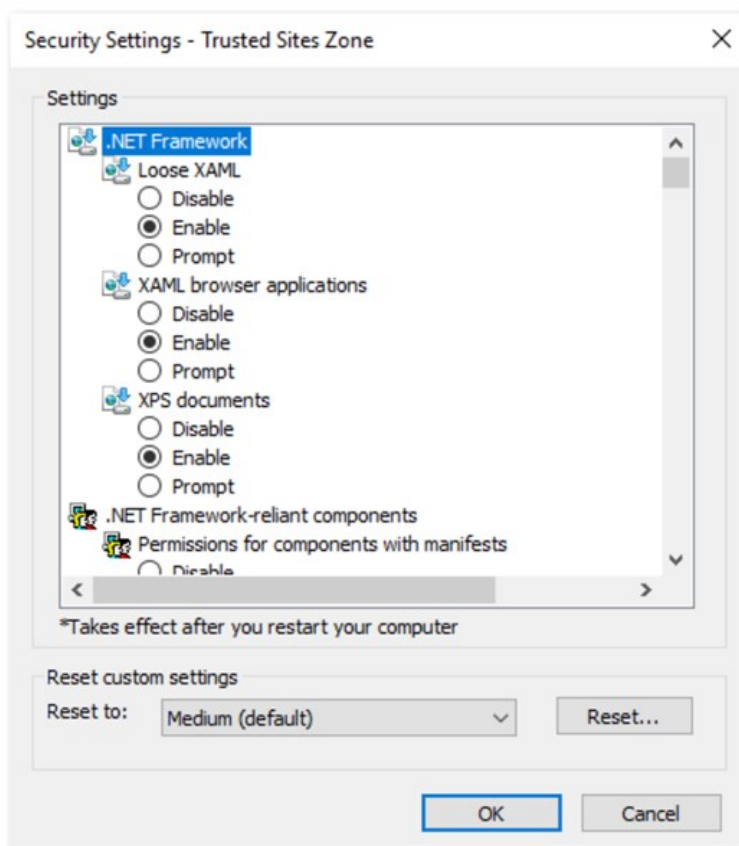
Note: You may not be able to log in using IWA on the server running Secret Server for Server 2008 or later because of security settings.

Troubleshooting

AD User Prompted for Credentials Even Though IWA Is Active

A user is logged onto their machine with the same Active Directory credentials they can log into Secret Server with, but the browser still prompts them for their credentials to reach the site. Ensure your Secret Server site is included in a security zone that allows for automatic logon:

1. In Internet Explorer, go to Internet **Options** > **Security**.
2. Click the **Trusted Sites** security zone.
3. Click the **Custom Level** button. The Security Settings – Trusted Sites Zone dialog box appears:



4. Scroll down to **User Authentication**.
5. Click to select the **Automatic logon with current user name and password** selection button.
6. Click the **OK** button.

Logging in as a Local Account Is Not Available

In Secret Server 10.0 and later, Secret Server requires Integrated Mode in IIS. The Integrated Mode can only support either Window Authentication or Forms Authentication (used for local account authentication), not both. Because of this limitation, Forms Authentication must be disabled for the site when using Integrated Windows Authentication. Thus, logging in as Secret Server local account is not available when IWA is enabled.

Installing Windows Authentication in Windows Server 2012 Manager

1. In Server Manager, click the **Manage** menu and select **Add Roles and Features**. The Add Roles and Features wizard appears.
2. Click the **Next** button. The Select installation type window appears.
3. Select the installation type.
4. Click the **Next** button. The Server selection window appears.
5. Select the destination server.
6. Click the **Next** button. The Server roles window appears.
7. Click to expand **Web Server (IIS) > Web Server > Security**.
8. Click to select **Windows Authentication**.
9. Click the **Next** button. The Select features window appears.
10. Click the **Next** button. The Confirmation window appears.
11. Click the **Install** button. The Results window appears.
12. Click the **Close** button.

Overview

You can specify a secret to provide the default credentials for running all PowerShell scripts on a site. This allows sites in different data centers to have different default credentials. This applies to remote password changing, checkout hooks, and account discovery PowerShell scripts.

Note: If you want a specific secret checkout hook, secret password changer, or account discovery scanner to use different credentials you can still provide credentials in those areas, which will take precedence over the one set on the site.

RunAs Secret Precedence

Remote Password Changing

The precedence order for which RunAs secret to use for remote password changing is:

1. Privileged account on the secret RPC tab
2. Secret site's RunAs secret
3. Secret

Secret Dependencies

The precedence order for which RunAs secret to use for PowerShell Secret dependencies is:

1. Privileged account on the dependency
2. Run As secret on the dependency group's site
3. Secret site's RunAs secret
4. Secret

Checkout Hooks

The precedence order for which RunAs secret to use for checkout hooks is:

1. Privileged account on the hook
2. Secret site's RunAs secret
3. Secret

Procedures

Setting the Default PowerShell Credential for a Site

To set a default PowerShell credential for a site:

1. Go to **Admin > Distributed Engines > Manage Sites**.
2. Select the desired site.
3. Click **Edit**.

4. Click the secret picker link on the **Default PowerShell RunAs Secret** field.
5. Click **Save**.

Using the Site PowerShell Credentials for Discovery

To use the site PowerShell credentials on a discovery scanner:

1. Add a PowerShell scanner to a discovery source or edit an existing scanner.
2. In the **Edit** dialog for the scanner, click to select the **Use Site RunAs Secret** checkbox.
3. Click **Save**.

Note: If no RunAs secret is set on the site, you will get an error message when you try to save.

For public websites, only SSL certificates issued by trusted authorities are recognized as valid. Self-signed certificates used only within a company or domain might generate security warnings but these can be ignored. The same is true of self-signed certificates installed on a server for the Secret Server website. However, these security warnings can also interfere with the use of the Secret Server Launcher and Web Password Filler. To resolve these issues, install the certificate on the client machine, either through Internet Explorer or Certificates snap-in.

To enable trust in the Secret Server self signed certificates, following these steps:

Step 1: Compare Host Names

Make sure that the host to which the certificate is issued is the same as the host name for your Secret Server website:

1. Open Internet Explorer and navigate to Secret Server.
2. Click **Continue to this website** if you are prompted.
3. Click the **Certificate Error** icon next to the navigation bar.
4. Click the **View certificate** button. The value next to **Issued to** should match the host name for your website. For example, if your website is `https://www.mydomain.local/SecretServer`, it should say **Issued to:** `www.mydomain.local`. If these fields do not match, the client will not be able to fully trust the certificate.

Step 2: Transfer a copy from your server to the client computer

Obtain a copy of the certificate file and transfer it to the client computer:

1. On the server where Secret Server is installed, find **Run** from the start menu or screen and type in `mmc`, then click the **Enter** button.
2. From the **File** menu, select **Add/Remove Snap-in**.
3. Select the **Certificates** snap-in, then click the right arrow button to add it.
4. In the window that appears, select **Computer Account**.
5. Select **Local Computer**.
6. Click **Finish**. You should now see the **Certificates (Local Computer)** node.
7. Expand the **Personal** folder and then the **Certificates** folder under it.
8. Right-click the certificate that Secret Server uses.
9. Click **All tasks**.
10. Select **Export**.
11. Keep clicking the **Next** button to accept defaults in the wizard.
12. Type in a filename.
13. Click the **Finish** button. The certificate has now been exported.

Note: If you have Firefox, the certificate can be saved to your client computer by viewing and exporting it after navigating to the website.

Step 3: Install the certificate on the client computer

1. On the client computer, find **Run** from the start menu or screen and type in `mmc`, then hit the **Enter** button.
2. From the **File** menu, select **Add/Remove Snap-in**.
3. Select the **Certificates** snap-in, then click the right arrow button to add it.
4. In the window that appears, select **My user account**.
5. Click the **Finish** button.
6. Expand the **Trusted Root Certification Authorities** folder.
7. Right-click the **Certificates** folder and select **All Tasks > Import**.

8. Click **Next** and **Yes** to accept default settings for all steps of the wizard.
9. When prompted for the certificate file, select the file you saved in the previous Step 2.

Note: You may need to re-open Internet Explorer and browse to Secret Server once more to see the change reflected on the client machine.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server supports a second layer of authentication, called multi-factor authentication (MFA) or two-factor authentication (2FA), for added security. This section discusses several options.

Duo Security Authentication

Note: Using this method of two-factor authentication requires that you have an active account for Duo Security.

Note: Secret Server supports using Duo Security as a second factor of authentication. See below for setup instructions.

Note: For more information on Duo and Secret Server, see the [Delinea Secret Server and Duo](#) page.

Task 1: Create a Duo Application Representing Your Secret Server (Admin)

1. Sign up for a new Duo account, or log in to an existing one at [Duo Security](#).
2. Under **Applications**, create a new application of the **Delinea Secret Server** type. Name the application as you wish.
3. Record the API hostname, integration key, and secret key from the new Duo application you just created.

Task 2: Configure Secret Server to Use Duo (Admin)

Note: Because Duo is a service, the Secret Server instance must have outbound access (TCP port 443) to reach the API host to work. If there is a firewall rule preventing access to Duo's servers, two factor authentication will not work.

1. Open Secret Server.
2. From the **Admin** menu, select **Configuration**.
3. Click the **Login** tab, and then click **Edit**.
4. Select the **Enable Duo Integration** check box.
5. Enter the **API Hostname**, **Integration Key**, and **Secret Key** values.
6. Click the **Save** button.
7. Go to **Admin > Users** to create a test user. The Users page appears.
8. Click the **Create New** button. The **Edit User** page appears:

Edit User

User Name

Display Name

Email Address

Domain

Password

Confirm

Two Factor

Enabled

Locked Out

[Advanced](#)

9. Click the **Two Factor** dropdown list and select **Duo**.
10. Type or select the other parameters for the new user. See [Users](#).
11. Log on as the test user. If there are multiple two-factor devices available, you will be prompted to select one. If you are un-enrolled you will be given a link to perform self-enrollment. You are contacted via the Duo app, SMS, or a phone call for the second factor.
12. Add or configure actual users one at a time or by using bulk operations.

Task 3: Setting up Duo (User)

1. Log on to Secret Server.
2. After successful authentication, a new screen appears with the option to select a method to authenticate with.
3. Select one of the options (**Duo Push**, **Send SMS**, or **Phone**), depending on your setup with Duo) and complete the selected authentication process to log in.

Applications for Soft Token Two-Factor Authentication

The name of the Secret Server soft token two-factor setting for Time-based One Time Password (TOTP) is "TOTP Authenticator." Any TOTP application that uses the TOTP RFC6238 algorithm, such as such as Microsoft [Authenticator](#), will work with the Secret Server TOTP Authenticator. Details on the TOTP RFC6238 standard can be found at [TOTP: Time-Based One-Time Password Algorithm](#).

Note: Google Authenticator support started in Secret Server version 8.6.

Email Two-Factor Authentication

Secret Server requires that a connection to a SMTP server be properly configured to send out confirmation code emails. Enter the SMTP server information and an email address that is used to send notifications:

1. Click **Admin > Configuration**.
2. Click the **Email** tab.
3. Verify SMTP server availability with telnet using the command `telnet <your server name> 25`.

Note: If virus protection is running, you may need to add a firewall rule to allow aspnet_wp.exe to send e-mails.

FIDO2 (YubiKey) Two-Factor Authentication Configuration

Overview

FIDO2

FIDO2 (Fast Identity Online, second edition) is an open authentication standard that uses physical devices for authentication. Delinea uses it for two factor authentication (2FA) with FIDO2 providing the second authentication after a normal password entry—any FIDO2-enabled user attempting access to a Secret Server account **must** have a FIDO2 device in hand. The device eliminates many password-related issues, such as phishing and man-in-the-middle attacks. It also speeds up the long on process over callback or texting 2FA.

YubiKey

YubiKey is a FIDO2-compliant product series from Yubico, a commercial company. We recommend two of their devices--YubiKey 5 Series and Security Key by Yubico.

Configuration

FIDO2 configuration follows these steps, which we cover in detail in this section:

1. Enable FIDO2 in your Secret Server.
2. Set up the user's credentials.
3. Distribute the FIDO2 device to the user.
4. User registers his or her device.

Prerequisites

- One FIDO2 device. We recommend the YubiKey series.
- A Secret Server Vault license or greater.
- Administer Users or User Owner permissions in SS.
- A Firefox or Chrome browser.

Enabling FIDO2 for a Single User

1. In Secret Server, click the **Admin** menu item. The Administration page appears:

Administration

- Active Directory
- Configuration
- Dependency Templates
- Diagnostics
- Discovery
- Distributed Engine
- DoubleLock
- Dual Controls
- Event Subscriptions
- Folders
- Groups
- IP Addresses
- Licenses
- Privilege Manager
- Privileged Behavior Analytics
- Remote Password Changing
- Roles
- Scripts
- SDK Client Management
- Search Indexer
- Secret Policy
- Secret Search Filters
- Secret Template Permissions
- Secret Templates
- Security Audit Log
- Session Monitoring
- SSH Command Menus
- SSH Proxy
- System Log
- Teams
- Themes
- Users
- Workflow Templates

Setup

- Setup Home
- Database
- Email
- Licenses
- Backup
- Internal Site Connector
- Server Nodes
- Upgrade Secret Server
- Add Privilege Manager Features
- Security Hardening Report

Third Party Integration

- Folder Synchronization

2. Click the **Users** button. The Users page appears:

Users

You are currently licensed for 101 user(s). You currently have 30 enabled user(s).

Search:

	User Name	Display Name	Email Address	Enabled	Domain	Two Factor
<input type="checkbox"/>	admin	Admin	admin@delinea.com	Yes	Local	< None >
<input type="checkbox"/>	will	Will	will@delinea.com	Yes	Local	< None >
<input type="checkbox"/>	john	John	john@delinea.com	Yes	Local	< None >
<input type="checkbox"/>	john.doe	John Doe	john.doe@delinea.com	Yes	delinea.com	< None >

3. Click the link in the **User Name** column for the user you want to configure. The View User page appears:

View User

User Name Will

Display Name Will

Email Address will@delinea.com

Domain Local

Two Factor < None >

Enabled Yes

Locked Out No

Application Account No

IP Address Restrictions
None

Restricted By Team No

4. Scroll down and click the **Edit** button. The Edit User page appears:

Edit User

User Name: bart.simpson

Display Name: Bart Simpson

Email Address: [Empty]

Domain: testparent.thycotic.com

Two Factor: <None >

Enabled:

Locked Out:

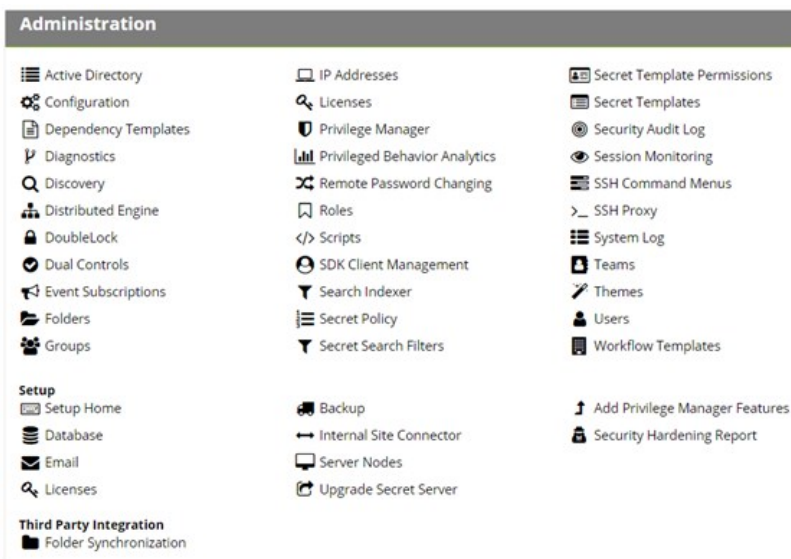
[Advanced](#)

5. Click the **Two Factor** list and select **FIDO2**.

6. Click the **Save** button.

Enabling FIDO2 for Multiple Users

1. In Secret Server, click the **Admin** menu item. The Administration page appears:



2. Click the **Users** button. The Users page appears:

Users

You are currently licensed for 101 user(s). You currently have 30 enabled user(s).

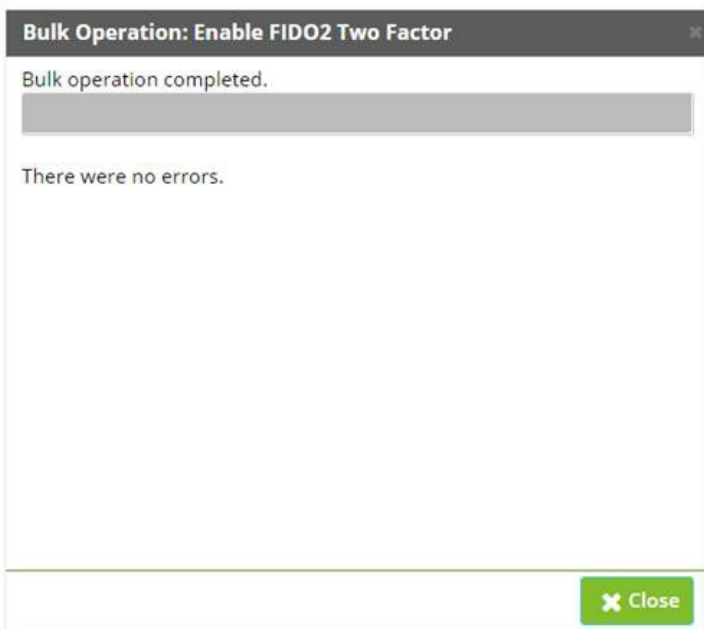
	User Name	Display Name	Email Address	Enabled	Domain	Two Factor
<input type="checkbox"/>	admin	Admin	admin@thycotic.com	Yes	Local	< None >
<input type="checkbox"/>	all	All	all@parent.thycotic.com	Yes	Local	< None >
<input type="checkbox"/>	test	Test	test@parent.thycotic.com	Yes	Local	< None >
<input type="checkbox"/>	test@parent.thycotic.com	Test User		Yes	testparent.thycotic.com	< None >

3. Click to select the unlabeled check box next to each user you wish to include.

4. Click the **Select Bulk Operation** list below the table and select **Enable FIDO2 Two Factor**. A warning popup page appears:



5. Click the **OK** button. A confirmation popup page appears:



6. Click the **Close** button.

Disabling FIDO2 for Users

Disabling FIDO2 for users, for both single and multiple, is almost the same as enabling them. There are two differences:

- For a single user, select **<None>** for the Two Factor list on the **Edit User** page.
- For multiple users, select **Disable FIDO2 Two Factor** in the **Select Bulk Operation** list on the **Users** page.

Note: Disabling FIDO2 2FA does **not** remove device registration information from Secret Server. If FIDO2 is re-enabled, the user can use the FIDO2 device without re-registering it.

Unregistering Users from FIDO2

Resetting FIDO2 serves to unregister existing users. There is no way to reverse it—users will have re-register a FIDO2 device, even the same one.

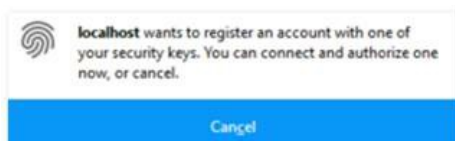
Resetting FIDO2 for both single and multiple users is very similar to enabling FIDO2 for multiple users. The only difference is you select **Reset FIDO2 Two Factor** in the **Select Bulk Operation** list on the **Users** page. That is right, for single users—you do a bulk operation.

Registering FIDO2 Devices (End User Operation)

1. After an admin registers the user in Secret Server the user is prompted upon his or her next log on. For example, in Chrome:



Or in Firefox:



Note: Legacy Microsoft Edge is not supported. Edge Chromium, version 79 or higher, is required for FIDO2 support.

2. The user inserts his or her FIDO2 device into a USB port on the computer.
3. The user activates it by touching the sensor on the device.
4. After successful registration, the user is **again** prompted with the same screen, which is authenticating the current session against the credentials that were just registered.
5. From then on, the user is prompted for his or her security after a successful username-password login. Once the key is authenticated, the Secret Server Dashboard appears.

Note: Only one FIDO2 device per user can be registered at any given time; however, the 2FA settings can be temporarily disabled or reset in the case of a lost or forgotten FIDO2 device.

Auditing and Security

- Upon registration, a user's FIDO2 Credential, the FIDO2 Public Key JSON string, and the FIDO2 Counter is stored in the User's audit log.
- Upon each successful FIDO2 authentication, the FIDO2 counter value is updated and noted in the User's audit log.

Troubleshooting and Issues

- If the user encounters an error or does not fulfill the authentication before the process times out, the user is redirected back to the username and password log on screen where the process can be reattempted.
- Authentication activities are logged in the user's audit log.
- System errors are logged in the Secret Server.log file in Secret Server's log directory.

RADIUS User Authentication

Secret Server allows the use of *Remote Authentication Dial-In User Service* (RADIUS) two-factor authentication on top of the normal authentication process for additional security needs. Secret Server acts as a RADIUS client that can communicate with any server implementing the RADIUS protocol.

Configuring RADIUS

Set up RADIUS on the **Login** tab of the **Configuration** page. This requires enabling RADIUS Integration, specifying the server address, the ports, and the RADIUS shared secret. The shared secret is a specific term for RADIUS clients and is not a reference to secrets in Secret Server.

You can customize the RADIUS "Login Explanation" to give users detailed instructions for entering their RADIUS information.

Once enabled, the **Test RADIUS Login** button appears on the **Login** tab for testing the communication with the RADIUS Server. If you have a failover RADIUS Server, you can specify it by clicking the **Enable RADIUS Failover** checkbox and entering the required information. If the primary RADIUS server cannot be accessed, the failover server is be used.

Enabling RADIUS for a User

After enabling RADIUS on your Secret Server, you must enable RADIUS two-factor authentication for each user on a per-user basis. On the **User Edit** page, type the **RADIUS User Name** for this user to match the RADIUS server. RADIUS can be enabled for new users by domain.

Enabling RADIUS Two-Factor Authentication

Secret Server allows the use of RADIUS two-factor authentication on top of the normal authentication process for additional security.

See the full [RADIUS Integration Guide](#) for additional information.

To configure RADIUS for the Secret Server instance:

1. Log on Secret Server with an account with "Administer Configuration" and "Administer RADIUS" permissions.
2. Navigate to **Administration menu > Configuration > Login**.
3. Click the **Edit** button.
4. Type the following:

- o **RADIUS Server IP** (IP address to your RADIUS Server)
- o **RADIUS Client Port** (default 1812)

Note: If your RADIUS server runs on the same machine as Secret Server, the client and server ports must be different.

- o **RADIUS Server Port** (default 1812 for RSA and 1812 for AuthAnvil).
- o **RADIUS Shared Secret**, which must match chosen RADIUS shared secret on your RADIUS Server. (Shared Secret is a RADIUS term and not related to any Secret Server secret.)
- o **RADIUS Login Explanation** (custom message or instruction). Defaults to "Please enter your RADIUS passcode."

5. Click the **Save** button.

To test RADIUS settings:

1. Click the **Test RADIUS Login** button. A popup appears.
2. Type the RADIUS username and password.
3. Click the **OK** button.
4. After enabling RADIUS on Secret Server, you must enable RADIUS two-factor authentication for each user:
 1. Sign into an account with "Administer Configuration" and "Administer RADIUS" permissions.
 2. Navigate to **Administration > Users**. The Users page appears.
 3. Select the desire user.
 4. Click the **Edit** button.
 5. Click to select the **RADIUS Two Factor Authentication** check box.
 6. Type the username in the **RADIUS Username** text box.

NOTE: Secret Server defaults this value to its username. If you wish to use this default name, it must match the username on the RADIUS server.

7. Review the settings and click **Save**.
8. Repeat these steps for each user that needs to use RADIUS.

TOTP

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server supports using any type of soft token or mobile application authentication using the *Time-Based One-Time Password* (TOTP) RFC6238 algorithm. TOTP's are typically generated and authenticated by a mobile application using an algorithm that incorporates the current time to ensure that each one-time password (OTP) is unique. TOTP applications include Authy, Google Authenticator, and Microsoft Authenticator.

Secret Server can also serve as an OTP generator, providing TOTP authentication for RPC and launchers. The soft token two-factor function in Secret Server is the "TOTP Authenticator" and you can use any application that uses the TOTP RFC6238 standard (details on the standard can be found at the [IETF Tools website](#)). An example of a TOTP application that works with Secret Server soft token two-factor authentication is Microsoft Authenticator.

Disabling TOTP for Users

To disable soft token two-factor authentication, follow almost the same process as enabling soft token two-factor authentication for a user, select **Disable TOTP Auth Two Factor** from the bulk operation drop-down menu instead of **Enable TOTP Auth Two Factor**.

Enabling TOTP for Secret Server Users

1. From the **Admin** menu, select **Users**.
2. Select the check box beside each user to enable two-factor authentication for.
3. From the **< Select Bulk Operation >** drop-down menu, select **Enable TOTP Auth Two Factor**.
4. Click **OK** in the dialog that appears, confirming the operation.
5. The user(s) are now required to complete the soft token setup with a mobile device the next time they log into Secret Server. See [Onboarding the Mobile App](#) for details on the account and mobile app setup that follow.

Enabling TOTP for Launchers

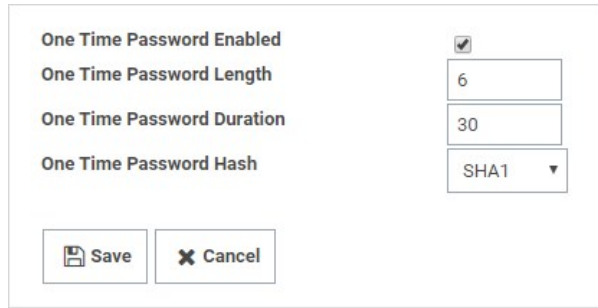
Most commonly, time-sensitive one-time passwords (TOTPs) are generated by a mobile application, such as Google Authenticator or Microsoft Authenticator. Additionally, Secret Server can be used as the TOTP generator for RPC or launchers (for web password secrets only at this time). Both the secret and the secret template require configuration for this use.

Secret Template Setup

To enable TOTP on a Secret Server template:

1. Go to **Admin > Secret Template**.
2. Select the desired template, and click the **Edit** button. The Secret Template Designer appears.
3. Navigate to the **Settings** section of the page, and click **Edit**.
4. Click to select the **One Time Password Enabled** check box. This enables the option with default settings:

Length: 6 Duration: 30 Hash: SHA1



The screenshot shows a configuration window for One Time Password settings. It includes a checked checkbox for 'One Time Password Enabled', input fields for 'One Time Password Length' (6) and 'One Time Password Duration' (30), and a dropdown menu for 'One Time Password Hash' (SHA1). At the bottom, there are 'Save' and 'Cancel' buttons.

Note: These are the values that most one-time password instances, such as Google and Microsoft Authenticator, use today. If you use these settings with another OTP provider and are unable to successfully use generated codes to authenticate, please review their documentation and adjust these settings as required.

5. Save the secret template. Any web password secret based upon this template can now use TOTP.

TOTP Secret Setup

Once a secret template is set up for TOTP, each secret based on that template also needs to be set up:

1. Click the **Secrets** menu item in the dashboard.
2. Open the desired secret.
3. Click the **Settings** tab:

Thycotic Web Password ☆

General Security Audit RPC Dependencies Sharing **Settings**

EMAIL NOTIFICATIONS - PERSONALIZED USER SETTINGS [Edit](#)

Send Email When Viewed	No
Send Email When Changed	No
Send Email When Heartbeat Fails	No

TIME-BASED ONE-TIME PASSWORD (TOTP)

Generate One-Time Passwords

Cancel Save

4. Click to select the **Generate One-Time Passwords** check box in the **TOTP** section. This exposes two text boxes:

Generate One-Time Passwords

TOTP Key * Show

TOTP Backup Codes Show

Cancel Save

5. Type the TOTP key in the **TOTP Key** text box. The TOTP Key is generated by the OTP-protected asset when you set up your account to use TOTP. Usually, you are prompted with a QR bar code that you can scan with a mobile device, or you can expose the key that the QR code represents. This text string is the value that is placed into the TOTP Key field.

Important: Treat the TOTP key and backup codes like you would any other password! If anyone obtains the key, it can be used to set up a valid TOTP generator for that account on any device, allowing that person to bypass the protection. Similarly, the backup codes allow users to temporarily bypass protection.

Note: If you have an account that has been TOTP protected and you did not save the TOTP key upon creation, you must

deactivate TOTP on that account and then reactivate it to retrieve the TOTP key to set up Secret Server.

6. Type the TOTP backup codes in the **TOTP Backup Codes** text box. The TOTP Backup Codes are often presented to a user while initially setting up an account for TOTP. These backup codes are single-use codes for use if a TOTP generator is not available or working. Again, these codes will be valid and allow the holder to get past the two-factor authorization to access an account, so protect them as you would a password!

Resetting TOTP for Secret Server Users

1. From the **Admin** menu, select **Users**.
2. Select the check box beside the user to reset two-factor authentication for.
3. Click select **Reset TOTP Auth Two FactorFrom** on the **< Select Bulk Operation >** drop-down menu.
4. Click **OK** in the dialog that appears, confirming the operation.
5. The user is now required to complete the soft token setup with a mobile device the next time they log into Secret Server. See **User Setup of Soft Token Two-Factor Authentication** for further details on the account and mobile app setup that follow.



Viewing a TOTP for a Web Secret

To view or copy the TOTP generated for an account:

1. Navigate to and open the desired secret.
2. Click the **General** tab:

Thycotic Web Password ☆


[General](#) [Security](#) [Audit](#) [RPC](#) [Dependencies](#) [Sharing](#) [Settings](#)

Secret Name *	Thycotic Web Password	Edit
Template	Web Password	Edit
URL *	http://www.thycotic.com	Edit
UserName *	myUserName	Edit
Password *	***** Show	Edit
One Time Password	 Generate One Time Password	
Notes		Edit
Launchers	 Web Password Filler	

[Show Advanced](#) [Edit all fields](#)

3. Click the **Generate One Time Password** link next to the **One Time Password** setting.
4. A dialog box appears with an OTP:

One Time Password for Thycotic Web Password

754 544 

Click the One Time Password to copy to clipboard

[Close](#)

1. Click the OTP to copy it to the clipboard.
2. Click the **Close** button.

Note: The "Generate One Time Password" link also appears on the preview pane when you click a secret on the All Secrets page.

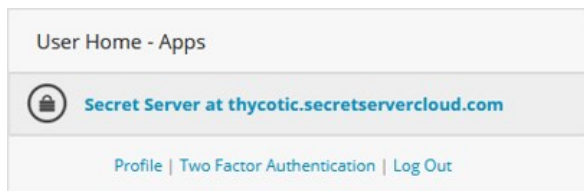
Enabling Two-Factor Authentication in Delinea One

When two-factor authentication is enabled, Delinea One presents a two-factor challenge to the user logging in. The Delinea One two-factor authentication supplements and does not replace any other two-factor authentication methods used by a client application such as Secret Server. Delinea One supports two-factor authentication using TOTP or SMS. You can have only one two-factor authentication method active at any time. We recommend using TOTP over SMS whenever possible for better security.

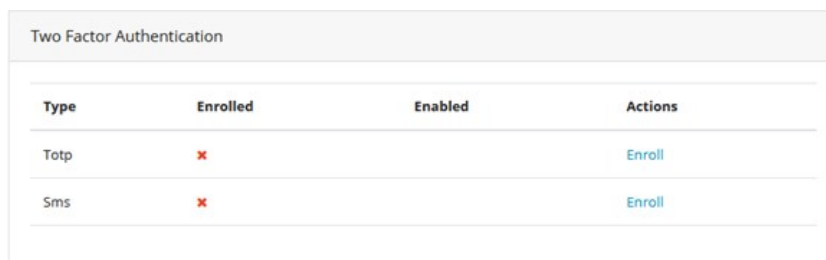
TOTP Two-Factor Authentication

To use TOTP two-factor authentication with Delinea One, you must first have a mobile device with an installed TOTP application such as Google Authenticator, Authy, or Microsoft Authenticator. When you have the app installed, follow the steps below.

1. Log into Delinea One and on the account homepage in the **User Home - Apps** dialog, click **Two-Factor Authentication**.

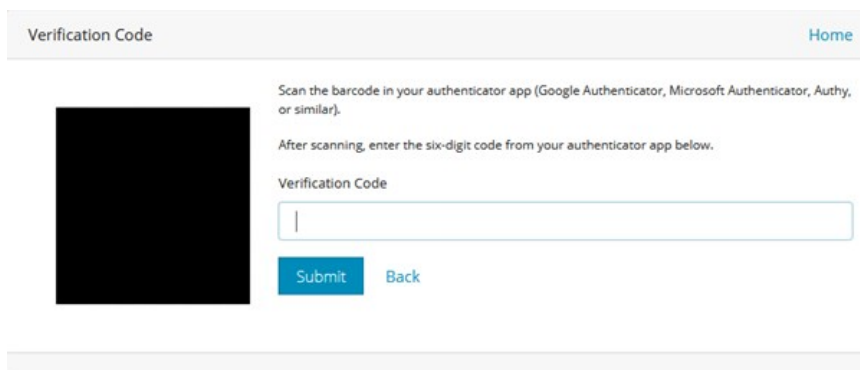


2. Choose **TOTP** and click **Enroll**.

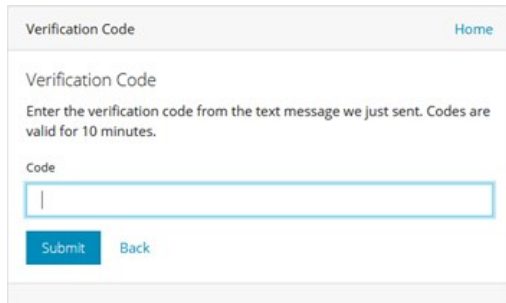


Type	Enrolled	Enabled	Actions
Totp	✘		Enroll
Sms	✘		Enroll

Delinea One displays a barcode (redacted in the example shown).



3. Using the TOTP app on your mobile device, scan the barcode. You will receive multiple six-digit codes.
4. In the Verification Code field, enter one of the six-digit codes (a new code is generated every 30 seconds).



Verification Code Home

Verification Code

Enter the verification code from the text message we just sent. Codes are valid for 10 minutes.

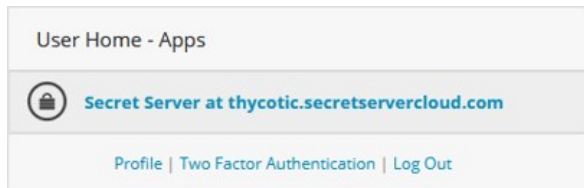
Code

When you have correctly entered and submitted a six-digit code, the setup of TOTP two-factor authentication is complete. From this point forward, each time you attempt to log in you will receive a text message on your mobile device with a code that you must enter to complete the login process.


SMS Two-Factor Authentication

To use SMS two-factor authentication with Delinea One, you must first provide and verify a mobile phone number. Delinea

1. To provide a mobile phone number, log into Delinea One, and on the account homepage in the **User Home - Apps** dialog, click **Profile**.

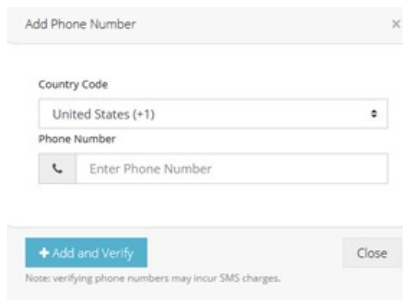


User Home - Apps

 **Secret Server at thycotic.secretservercloud.com**

[Profile](#) | [Two Factor Authentication](#) | [Log Out](#)

2. Click **Add Phone** and enter the country code and phone number of a mobile phone that accepts text messages.




Add Phone Number ×

Country Code

United States (+1) ▾

Phone Number

 Enter Phone Number

Note: verifying phone numbers may incur SMS charges.

3. Click **+Add** and **Verify**.

Delinea One sends a text message to the phone number, with a code.

4. Enter the code and click **Submit** in the Verification Code dialog.

Verification Code [Home](#)

To start using SMS two-factor authentication, click the button below to send a text message to your primary phone number: [REDACTED]

[Send Text Message](#)

When you receive the text message, enter the six-digit code below.


Verification Code

[Submit](#) [Back](#)

The phone number now appears as Verified on your profile page.

1. On the Delinea One account homepage in the **User Home - Apps** dialog, click **Two-Factor Authentication**.

User Home - Apps

 [Secret Server at thycotic.secretservercloud.com](#)

[Profile](#) | [Two Factor Authentication](#) | [Log Out](#)

1. Choose **SMS** and click **Enroll**.

Two Factor Authentication

Type	Enrolled	Enabled	Actions
Totp	✗		Enroll
Sms	✗		Enroll

Delinea One sends a text message to your phone with a six-digit code.

2. Enter the six-digit code into the box provided.

Verification Code [Home](#)

Verification Code

Enter the verification code from the text message we just sent. Codes are valid for 10 minutes.

Code

[Submit](#) [Back](#)

When you have correctly entered and submitted the six-digit code, the setup of SMS two-factor authentication is complete. From this point forward, each time you attempt to log in you will receive a text message on your mobile device with a code that you must enter to complete the login process.

Backup and Disaster Recovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server supports manual and scheduled database and IIS directory backups. The database access settings support SQL mirror and automatic failover. As an additional disaster recovery measure, administrators can export secrets to a CSV spreadsheet.

Secret Server can be configured to backup to a network share instead of a local folder on the server. For example, you may want to do this such as when the Secret Server database (SQL) is located on a different server than the web application server (IIS).

To back up:

1. Ensure the Secret Server IIS Application Pool is running as a service account if it is not already. See Secret Server [Running the IIS Application Pool As a Service Account](#).
2. Grant access to the network share (using Windows ACLs) to the account running the Secret Server IIS Application Pool (so that Secret Server can backup the application folder and zip it to the network share).
3. Grant access to the network share (using Windows ACLs) to the account running Microsoft SQL Server service. (so that Microsoft SQL Server can backup the Secret Server database to the network share). You can change the service account running Microsoft SQL Server by going to SQL Server Configuration Manager.
4. Go to **Admin > Backup**. This may require you to go to **Admin > All** and search for **Backup**.

Backup Configuration

i The AppPool running Secret Server must be configured to not shutdown. See the following KB Article.
Secret Server is currently running as "GAMMA\ss_iis_svc", you will need to grant Full Control to the backup folder specified for this user.

i To backup to a network share, see the following KB Article.

Enable Web Application Backup	Yes
Backup File Path	c:\backup
Enable Database Backup	Yes
Backup Database File Path	c:\backup
Database Backup SQL Timeout (Minutes)	30
Enable Copy-Only Database Backups	No
Keep Number of Backups	10
Notify Administrators on backup failure	No
Enable Scheduled Backup	Yes
Backup Start Time	5/15/2019 10:18 AM
Backup Every	1 days 0 hours 0 minutes
Next Scheduled Backup	5/28/2020 10:18 AM
Enable TMS Backup	No
TMS Installation Path	

← Back ✎ Edit 📄 View Audit ▶ Backup Now

5. Note that the two file paths are from two different perspectives—Backup File Path is from the ASP.NET application server and Backup Database Path is from the Microsoft SQL Server (these may be on the same box in your environment, or they might not be depending on how you have configured Secret Server).
6. Click the **Edit** button.
7. Type the Secret Server backup path, such as `\\server01\backup\secretserver\`, in the **Backup File Path** text box.
8. Type the database backup path in the **Backup Database Path** text box.
9. Click the **Save** button.

From the Backup Administration page, specify the correct directory paths for the IIS Secret Server file directory and the database backups to be stored. The backup path must be local to the server where the Secret Server database or file directory exists. The directories must also have the proper permissions to allow Secret Server to automatically store backups at those locations. The account that requires permission is displayed as an alert on the Backup page.

Overview

The following configuration options are available on the **Tools > Backup** page of Secret Server:

- **Backup Database File Path:** This folder must be accessible by the SQL server and stores the database.bak file. See [File Path Settings](#).
- **Backup File Path:** This directory must exist on the Web server and stores the zip file of the application directory. See [File Path Settings](#).
- **Database Backup SQL Timeout (Minutes):** Number of minutes that Secret Server waits for the database backup to complete successfully before timing out.
- **Enable Scheduled Backup:** Enables automatic backups on a set schedule.
- **Keep Number of Backups:** Number of previous backups to keep.
- **Notify Administrators on Backup Failure:** Users with the Administer Backup role permission are notified if the backup fails.
- **Days to Keep Operational Logs:** Sets the period to keep backup-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).

File Path Settings

There are two file path settings on the **Admin > Backup** page (`ConfigurationBackup.aspx`). The "Backup File Path" setting corresponds to the application backup. The "Backup Database Path" setting corresponds to the SQL server backup.

Generally, the "Backup File Path" setting can be set to a path local to the application server for backing up of application files. If Secret Server is running under an account that does not have permission to write to a local path, then a network share can be used. If the SQL server is located on the same server as the Web application server, the "Backup Database File Path" setting can be set to a local path.

If the SQL server is not located on the same server as the Web application server then a network share should be used. The account under which SQL server service is running either must have modify rights to that path or must be a member of a group with modify rights to that path. You must use UNC (Universal Naming Convention) notation to write to a network path. For example: `\\TESTVM0\c$\backupDirectory`.

If you get an error stating "Cannot open backup device... Operating system error 3," this is often due to an invalid path value.

Note: For Secret Server to delete old database backups, the backup database path must also be accessible by the Secret Server Application Pool account.

Cannot open backup device... Operating system error 3

This is often due to an invalid path value for the "Backup Database File Path" setting. For more information on the proper values for this setting, see [File Path Settings](#).

Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

This is often due to an overly-large database. The Secret Server database likely contains too many log entries. To clear these, within Secret Server, select System Log from the Administration menu. Click the "Clear" button below the data grid that contains the log entries. If the timeout occurs with the clear as well, an upgrade to the latest version should resolve this. If the timeout issue persists with the backup, additional SQL database clean-up may be necessary. Contact [Delinea Support](#) for instructions on shrinking the reserve database size.

The process cannot access the file... because it is being used by another process

The cause of this message is typically multiple backup threads running simultaneously with all attempting to write to the same file. To fix this, open IIS Manager and ensure the "Maximum Worker Processes" setting for Secret Server's application pool is set to 1. If it is not, set the value to 1 and then either recycle the application pool or perform an `iisreset`.

Unable to complete backup. The following exception occurred: System.Threading.ThreadAbortException: Thread was being aborted

If this error message appears in combination with the application backup files not completed or the size of the file is unusually small, the backup process may have been interrupted by anti-virus software. Disabling scanning of the backup folder should resolve the issue.

Also see: [Backing up Secret Server to a network share](#) (KBA)

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Overview

The Disaster Recovery (DR) feature is a tool for emergency access to critical systems in the event of an emergency, such as a network outage. DR generates updates at regular intervals from one Secret Server instance (the data source) to another, or many other, instances (the Replicas). A select set of Folders and the Secrets they contain will have any changes sent securely to the configured Replicas so that, in the event the Data Source becomes inaccessible, the Replicas can be quickly utilized for emergency access to vital systems

Setup

Role

Assign the "administer disaster recovery" role permission:

Note: Do this on both the data source and replica instances.

1. Go to **Admin > Roles**.
2. Select or create the role you want to configure this feature with.
3. Click the **Permissions** tab.
4. Click **Edit** and select **Administer Disaster Recovery** from the list of unassigned roles.
5. Click **Save**.

Data Source DR configuration

Configure the data source DR configuration:

1. Go to **Admin > See All**.
2. Search for **Disaster Recovery**.
3. Click **Disaster Recovery**.
4. Click the **Data Replicas** tab.
5. Click **Add Data Replica**.
6. Save the data source URL and key somewhere you can easily access. Click the button to copy the key to the Windows keyboard.

Replica DR configuration

1. Go to **Admin > See All**.
2. Search for **Disaster Recovery**.
3. Click the **Configuration** tab.
4. Click **Edit**.
5. Click to select the **Enable** checkbox.
6. Type the data source URL you saved before.
7. Search for the group that will be given owner permissions on the folders and secrets. This is a necessary because users and groups are *not* replicated, thus data source permissions are not mapped to the replica.
8. Type or paste the data source key you saved before.
9. Click **Save**.
10. At the top of the page, click **Test Connection**. This verifies everything is configured correctly.

Finishing up

1. Once configuration is complete, return to the **Disaster Recovery Configuration** section on the **Data Source** page to refresh the **Replicas** tab. You should see a block for the replica connection you just tested.
2. Expand the block and approve it.
3. We recommend expanding the block and clicking the Manage button for folders. This enables you to select which folders, subfolders, and secrets get replicated. This reduces the scope of the replicated data, enabling faster replication and a smaller duplicated data footprint.

Note: This will enable the regular interval synchronization

Replication

By design, the first time a replication from the data source to the replica occurs, all data is replicated. This is likely the slowest DR operates and can take up to 20 minutes to replicate a data source containing 100,000 secrets.

All subsequent replications are significantly smaller. Only information that has been changed since the previous replication is replicated again.

Replication Scope

The items that DR replicates:

- Folders: Configure this on the data source (under Admin > See All > Disaster Recovery > Data Replicas > Select a Replica > Manage) to limit what folders are replicated. By default, all folders are replicated.
- Launchers: All Launchers and associated mappings are replicated.
- Roles: All roles are replicated.
- Secret Templates: All secret templates are replicated.
- Secrets: All secrets within the replicated folders are also replicated.

Replica Mode

When you configure a Secret Server instance as a replica, it is automatically placed in replica mode. While in replica mode, the following features are disabled

- ConnectWise synchronization
- Discovery
- DoubleLock
- Heartbeat
- Pipeline and bulk operations on secrets
- Remote password changing
- Secret import
- Secret policy changes to secrets

Recommendations

We recommend placing a Secret Server replica into read-only mode. This further reduces the possibility of data changes to the instance.

Files uploaded to secrets can be backed up using the standard Secret Server backup function. Upon backup completion, they retain their encrypted status and are inside the application backup file (the .zip file).

Important: Geo replication requires a Secret Server Geo-Rep license and ongoing Delinea Professional Services to ensure that upgrades work correctly. Due to improvements in AlwaysOn technology, we recommend using it instead for most use cases.

Overview

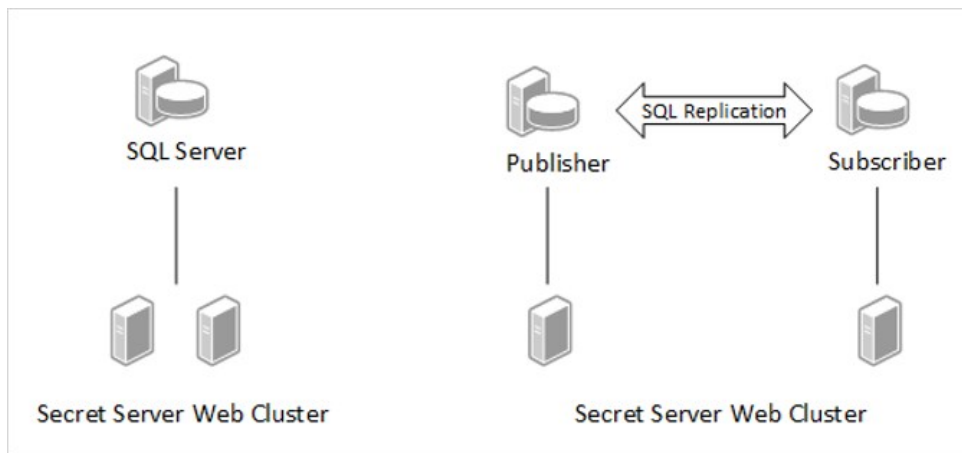
Secret Server geo replication is SQL Server replication using a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency across geographically distributed data centers.

Important: This topic is for information and planning only—**we strongly recommend contacting Delinea Professional Services before implementing any setup or strategy discussed here.**

Note: Secret Server uses merge replication. The only version validated is pull-based merge replication. Push based replication has not been validated.

SQL Server Replication

Figure: SQL Server Replication



Benefits of Replication

Enabling SQL Server replication allows a database and application to be hosted closer together and this allows for the mitigation of network latency and outages.

- Decrease application server to database network latency
- Resolve issues with unreliable network connectivity
- Allows for distribution of workload in a scale out fashion
- Works across large distances

In a typical web-clustered version of Secret Server, all application servers access a centralized database. In the event of a network outage, any users on affected application servers are not be able to use Secret Server until network access was restored. In addition, poorly performing networks can introduce latency that may decrease the responsiveness of Secret Server. This technology provides additional options when designing the network topology behind Secret Server that can help alleviate these issues.

High Availability

SQL Server replication is *not* an option for high availability, but it can be coupled with other technologies like SQL Server AlwaysOn Availability Groups to provide high availability. Any architecture should be reviewed and designed with your database group.

Architecture

The SQL Server Replication technologies do all the work of ensuring data consistency between each database, and Secret Server is designed to work well with this technology. When SQL Server replication is enabled on a specified database several system tables, views, stored procedures, and SQL Server jobs are added to the database schema. These tables store information about the data replication, and the procedures contain most of the code needed to perform the synchronization between databases.

Data Synchronization

This is the process through which SQL Server replication integrates changes from each database node. These changes include data changes as well as schema changes. Each subscribing database node has a synchronization interval that defines when it will synchronize any changes between the main publication node and itself. Secret Server has been tested using pull-based subscriptions where a scheduled job on each SQL Server subscriber runs at a specified interval to trigger the synchronization.

Data Conflicts

Enabling SQL Server replication introduces the possibility of data conflicts occurring in the Secret Server environment. This can happen when two people in different regions attempt to update the same set of data. Due to the disconnected nature of the technology, the system is unaware of this conflict until a data synchronization occurs. During synchronization, SQL Server attempts to resolve any conflicts based on a defined set of parameters. Secret Server provides a setup script to help define optimal parameters for each article (table, view, and stored procedure) in the Secret Server database.

SQL Server Replication Monitor and Conflict Viewer

SQL Server provides tools within SQL Server Management Studio that allow viewing of synchronization status, data conflicts, and publications. If a data conflict is not automatically resolved, use the Conflict Viewer tool to resolve the conflict and pick which set of data should be kept. It is possible that manual intervention could be required to resolve the conflict in the base tables on one of the nodes; otherwise, the conflict may still occur after the next synchronization.

For more information:

- [Replication Monitor](#)
- [Conflict Viewer and Interactive Resolver](#)

Tracking level

SQL Server replication tracks changes to each node by either row or column:

- Row-Level Tracking: Conflicts can occur when any change to any column is made on the same row even if it is not the same column.
- Column-Level Tracking: Conflicts only occur when the same columns are updated on a row.

For more information on row- and column-level tracking, see [Row-Level Security](#)

Conflict Resolvers

SQL Server replication uses resolvers to determine the outcome of data conflicts between two database nodes. For example, the same row or column was updated on different nodes. These can be as simple as the publisher database always wins, last change by date wins, and many others.

Switch Secret Server nodes over to subscriber databases. Any nodes that are configured to run background, engine, or session recording roles must remain on the publisher database.

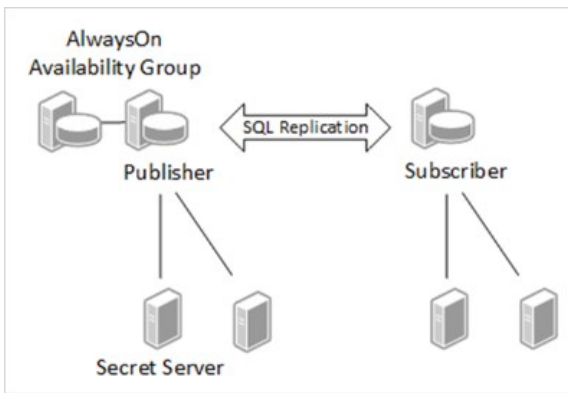
For more information see:

- [Advanced Merge Replication Conflict Detection and Resolution](#)
- [Specify Merge Replication Properties](#)

Secret Server and SQL Replication

There are many architectures for how SQL Replication can be setup. Determining the correct configuration requires proper planning with a good understanding of how SQL Replication works and the intended goals of using this technology. Here are a few examples.

Figure: Secret Server Web Cluster with SQL Server Replication

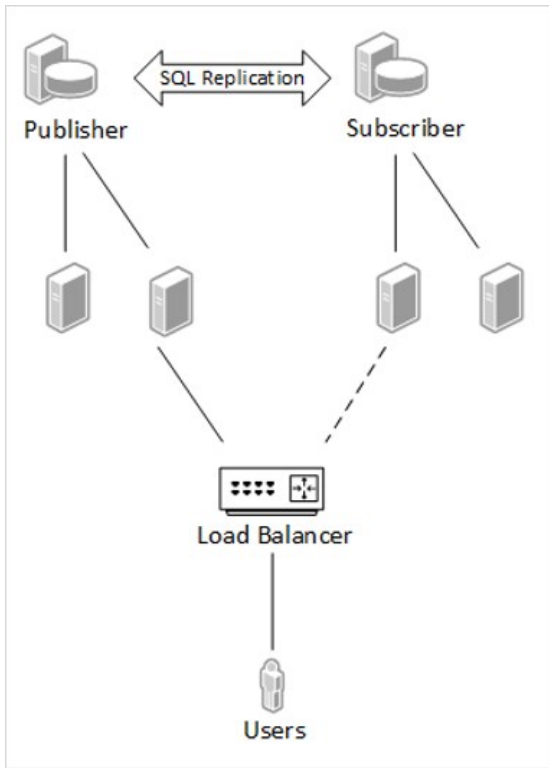


Some key points about designing the right architecture:

- Multiple web application servers can connect to the same database server.
- Secret Server allows you to configure various roles (background, engine, session recording) that perform various services, such as heartbeat and password changing. These roles can only function on a node that has a connection to the publisher database and will not run on other nodes even if configured to do so. If no suitable node is available that has roles enabled and is connected to the publisher database, then certain activities such as secret heartbeat and remote password changing will be offline until such a node becomes available.
- Secret Server Engines are an effective means to distribute workload to different networks or sites. Each engine must call back to a Web application node that connects to a publisher database.
- The diagram shows the publisher in an AlwaysOn availability group, but this is just an example. Depending on the needs of the organization, AlwaysOn could run on both the publisher and the subscribers or on neither. Many other high availability options could be leveraged alongside SQL Replication.

Using a Subscriber When the Publisher Is Offline

Figure: Using a Subscriber When the Publisher Is Offline



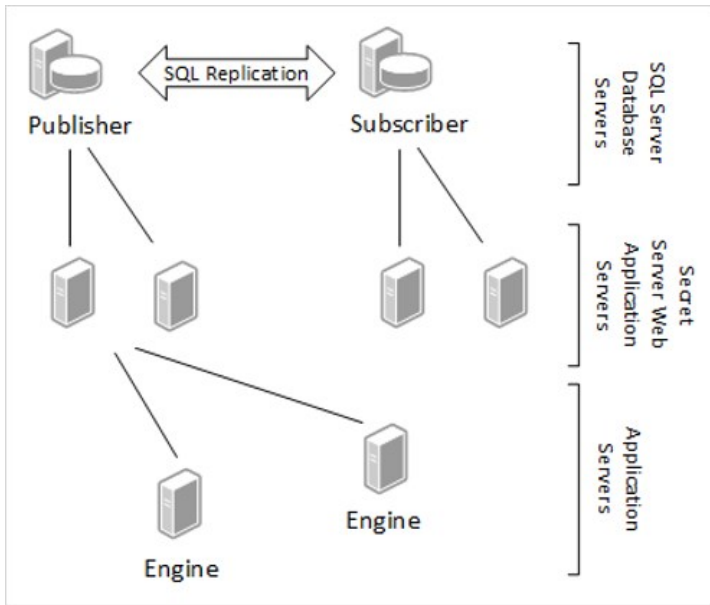
It is important to note that there is a delay in data synchronization between database nodes. This is based on how frequently SQL replication is configured to synchronize. One way to mitigate this data latency is to have every user access a node connected to the publisher. This means that every user would be accessing the same database.

In the event of a network outage, a load balancer could be configured to fail over to the subscriber nodes. The data on the subscriber would be as up to date as the last synchronization, and when network connectivity is restored all data activity will be synchronized again. Obviously, if the issues of network latency or disconnects occur too frequently, this may not be a workable solution.

Secret Server Distributed Engine

You can add Secret Server distributed engines to different network segments, but you must still be able to call back to a Secret Server Web application server that is connected to a publisher database node. This helps to ensure that heartbeats and password changes occur with the most up to date dataset.

Figure: SQL Replication and Secret Server Distributed Engines



Secret Server Replication Settings

Keep in mind that this is a disconnected technology, so there are required decisions when setting it up:

- How often should synchronization occur?
- Should any resolvers override the provided defaults?
- Are there any operational considerations based on the type of secret data?

Publications

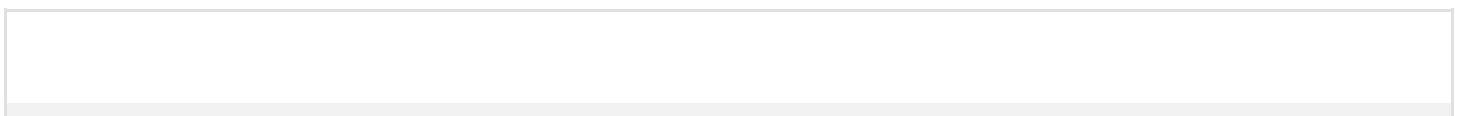
We recommend setting up two publications for Secret Server:

- **Events:** These are things that need to occur on a timely schedule, such as updating secrets. The events publication should be set to synchronize every minute. This means that if a secret were created in one region, it would not appear in another region for one minute. By default, this publication is called *SSPubEvents*.
- **Logs:** Information from each audit log is available on its server. We do not recommend setting log synchronization to less than hour. If this happens too frequently, the database could create deadlocks by constantly updating large sets of data. Once the logs merge, the events will appear in the order in which they occur and show on the server on which they took place. By default, this publication is called *SSPubLogs*.

Tracking Level and Resolvers

Using the default Secret Server implementation, most conflicts are resolved by taking the change made on the publication server as the winner (using the "publisher wins" resolver). It is assumed that the publication server is most likely the main server in an environment and therefore, most likely, the decider in a case of a conflict. Thus, we recommend doing functions, such as configuration changes or secret template definition changes, on the publication node. This is only an issue if two people update the same data on two different servers before a synchronization occurs.

There are some exceptions to the "publisher always wins" rule:



Secret Fields	Last Update Wins	If two people change a password on a secret to a new password, then the last person to make the change will win.
Secret Access Request Approval	Last Update Wins	If two people are in a group that is requested to approve access, then the last person to approve or disapprove access would set that approval.
File Attachment	Last Update Wins	
tbConfiguration	Column Level	Configuration changes are resolved by table column, not the default row-level where the entire row is resolved. This allows different configuration options to be set on different servers and not conflict with each other, but changing the same option would create a conflict.

Always remember that the conflict is only resolved when synchronization occurs. If there is a synchronization window of 24 hours, there could be very different data on different servers for that entire 24-hour period. The risk of data conflicts increases with larger synchronization windows. The default recommended synchronization is every minute for most items and every hour for system logs.

For a complete listing of recommended settings navigate to the **SQL Replication Administration** page in an installed version of Secret Server at **ADMIN > Nodes > SQL Server Replication > Show Articles**.

Conflict Auditing

When conflicts occur during synchronization, Secret Server writes an event to the system log on the server that it has detected these conflicts. Depending on the type of conflict, more-specific information may be written to the audit log for a specific data entity, such as a secret or folder. Notification of the conflict is written to the system, secret audit, folder audit, and other logs. To view complete details of the conflict, the SQL Conflict Viewer tool needs to be used to review the conflict by right clicking on the publication and choosing to View Conflicts.

As an example, if the same field changed on two different servers, resulting in a conflict, then the Secret Server audit log will indicate the field was changed twice and then denote that there was a data conflict.

Operational Latency

Give consideration to how your Secret Server data is setup and managed. This is most easily described by an example where a manager uses a Secret Server node in Europe and their employee works on a node in Australia. Due to network latency, the organization has changed the default event log synchronization interval from every minute to every five minutes. The employee requests access to a secret in Australia, but due to the synchronization window, the manager is not alerted until five minutes later because the manager is using the server in Europe. You can mitigate this issue by having operational groups work on data for their region or decreasing the synchronization window.

Options to manage data latency if data conflicts occur regularly:

- Distribution of duties: Only update secret templates on the main node.
- Distribution of data: Only edit secret data by region. For example, create a folder for Europe and one for Australia to segment the data. Only edit items in each folder when in the appropriate region.

Article Settings

When setting up the SQL Replication Publication certain articles need settings other than the default settings. The recommended settings can be found within Secret Server by accessing the SQL Server Replication page located at **Admin > Nodes > SQL Server Replication > Show Articles**. This page can also be used to generate a setup script for both the Publication and Subscribers that uses these default settings. A Distributor will need to be created before running these scripts. For more information, see [Configuring Distribution](#).

Compensate for Errors

When conflicts occur, an article that has the `compensate_for_errors` attribute set to true will automatically try to resolve the conflict. When false, a SQL Server administrator can use the SQL Server Conflict Viewer to review and resolve conflicts.

Identity Range

SQL Server replication manages table identities by assigning ranges to the publisher and subscribers. Certain tables (logging or auditing) require larger assigned identity ranges. New ranges are only assigned when a data synchronization occurs. For more information, see [Replicate Identity Columns](#).

Variations

How SQL Server replication is setup can vary greatly and there may be reasons to not use the standard setup. Consult with your database group for approaches that may work well in your environment. The architecture diagrams contained within this document are just high-level examples.

Installing and Configuring SQL Server Replication

Installation

There are a multitude of configuration options for SQL Server replication. At a high level, these are the steps to setup Secret Server in a SQL Server replicated environment:

1. Install Secret Server
2. Enable Secret Server Clustering. For more information, see the [Secret Server: Server Clustering Administration Guide](#).
3. Setup a SQL Server replication distributor:

Note: This can run on the same database as the publication database or on a separate one on another server.

1. This can be done from the SQL Server Management Studio by right clicking on the **Replication** node.
2. Review what settings are appropriate for the distributor setup with your on-premise database group or with a database consultant.
3. Download the SQL publication script.
 1. In Secret Server select **ADMIN > Nodes > SQL Server Replication > Get SQL Publication Script**.
 2. Review this script and update the variables according to your environment.

Note: Advanced users can use the article list on that same page to configure SQL replication differently than this script to suit your environment.

1. Run this script on the Secret Server Database. A DBA with administrative privileges is needed to run this. Please consult with your on-premise database group or review your configuration with a database consultant.
2. Create snapshots for each publication:
 1. Open SQL Server Management Studio
 2. Right click on each publication under **Replication > Local Publications**
 3. Select **View Snapshot Agent Status**.
 4. Click **Start**

3. Download the SQL subscriber script:

1. In Secret Server select **ADMIN > Nodes > SQL Server Replication > Get SQL Subscriber Script**.
2. Review this script and update the variables to match your environment
4. Create a new database on the database server you intend to be your subscriber. Ensure the script uses this database and machine name. Set up permissions for the user or network account that Secret Server uses to connect to this database.
5. Run the subscriber script on the publication database first and then on the subscriber database. If your variables are set properly, it will execute the appropriate part of the script.
6. Expand the **SQL Server Jobs** on the subscriber, and you should see two jobs named for each publication.
7. Right click these jobs to start them. After they complete, your subscriber database should have replicated the schema objects from the publication.
8. Switch Secret Server nodes over to subscriber databases. The primary node *must* remain on the Publisher database, as must any node that an engine calls back to, but all other nodes can be reconfigured to use subscriber databases. Which nodes to switch depends on your specific needs as described in the previous sections. To switch an existing node to a subscriber database, log into that node and go the DbConnectionReset.aspx page by entering that page name in the URL field of your browser ([http\[s\]://<your_secret_server_name>/DbConnectionReset.aspx](http[s]://<your_secret_server_name>/DbConnectionReset.aspx)). Step through the wizard, entering the name of the new server and database when prompted. After completing this step, recycle the node's application pool.

Troubleshooting the Installation

Replication Setup Scripts Fail

Make sure that the SQL Server replication feature is enabled before running the script. Check the error messages in the script output. Make sure you set all of the variables in the top of the script correctly (such as publisher server, database names, and subscriber server).

SQL Replication Job Fails

To see the error message, it is easiest to right click on the job and choose to view history. This error message can indicate the actual problem.

Removing SQL Server Replication

Certain operations such as upgrading Secret Server, adding or removing DoubleLocks from secrets, and enabling or disabling HSM cannot be performed while SQL Server replication is enabled. In order to perform these operations you must remove SQL Server replication. When you are done with the action that required you to remove replication you can install and configure it again by repeating the previous instructions.

You only need to remove replication from the Publisher and all Subscribers. The Replication Distributer does not need to be removed. These are the steps to remove SQL Server replication from your publisher and subscriber databases.

On Each Subscriber

1. Stop the websites of all nodes using the subscriber database.

Note: You can see what database each node is using from **Admin > Server Nodes**.

1. In SQL Server Management Studio, go to **Replication**, right-click **Local Subscriptions** and choose **Generate Scripts...**
2. In the **Generate SQL Script** dialog, click to select "Subscriptions in the following data sources" and select the Secret Server subscription databases.

3. Select "To drop or disable the components."
4. Click **Generate Script > Open in new query window**.
5. Click **Close** to close the dialog once the script is created.
6. The script contains sections to be run on both the subscriber and the publisher. Run the sections on the subscriber by uncommenting them and commenting those for the publisher.
7. Copy the script to a query window on the publisher server.
8. The script contains sections to be run on both the subscriber and the publisher. Run the sections on the publisher by uncommenting them and commenting those for the subscriber.
9. Perform any maintenance actions needed on the subscriber database and nodes.

On the Publisher

1. Stop the websites of all nodes using the publisher database.
2. On the publisher, go to **Replication**, right-click **Local Publications** and choose **Generate Scripts...**
3. In the **Generate SQL Script** dialog **check Publications in the following data sources** and select the Secret Server database.
4. Click to deselect the **Distributor Properties** check box.
5. Select **To drop or disable the components**.
6. Click **Generate Script > Open in new query window**.
7. Click **Close** to close the dialog once the script is created.
8. Run the script.
9. Perform any maintenance actions needed on the subscriber database and nodes.
10. After all maintenance tasks are done, restore replication as described in [Installing and Configuring SQL Server Replication](#).

Managing SQL Server Replication

Once replication is setup and working certain considerations should be given to managing it along with conflicts that can occur. The recommended settings for the publication have been tested to limit conflicts, but they can still occur. Here are some scenarios you might encounter:

Conflict automatically resolved	SQL Server was able to determine how to resolve the conflict. Secret Server will log and audit this conflict. To see the specific details for the conflict, use the SQL Server Conflict Viewer in SQL Server Management Studio.
Conflict unable to be automatically resolved	A user with SQL Server access needs to open the conflict in SQL Server Management Studio. Access these by right-clicking on the publication and choosing to view conflicts.
Some data stops replicating	A table could become blocked if there are conflicts. Other data may continue to synchronize. If data in one region or database node is different than another after a synchronization, there could be conflicts that need to be reviewed and resolved.

SQL Replication Synchronization Times	The status of each publication and subscribing server along with the last time of synchronization can be located in Secret Server by selecting ADMIN > Nodes > SQL Server Replication .
---------------------------------------	--

Web Server Nodes

The **Web Server Nodes** page now includes a column that lists the database server name and database name. This column also indicates whether the database is the publisher or a subscriber. To see the page:

1. Open Secret Server.
2. Click the **Admin** menu item and select **All**.
3. Click the **Server Nodes** button. The Server Nodes page appears (not shown).
4. If you click the **SQL Server Replication** button, you can see more information about your SQL Server replication. The page pulls from the replication data and shows the:
 - Publication name
 - SQL server
 - Database
 - Subscription type (push or pull)
 - Status of the last sync
 - Last time that subscription was synced
 - Status of the last sync
 - Date of the last sync

Note: The same information is available within SQL Server Management Studio, but this page gathers all of the subscription information together in one place.

As mentioned earlier, the **Get SQL Publication Script** and **Get SQL Subscriber Script** buttons will download replication script templates that you can use to set up replication for Secret Server. You can fill in the variables at the top of each script to match your environment and run them as-is or modify them further if you need to customize the default scripts.

The **Show Articles** button lists each article in Secret Server that should be included in replication along with the recommended settings for SQL Server replication. These are *recommended settings* and do not show the current state of replication on the publisher.

Secret Server Upgrade Scenario

New versions of Secret Server may issue schema changes including indexes, column changes, and views. In some cases, SQL merge replication will not automatically replicate these schema changes. For this reason, we recommend removing any publications and subscriptions targeting the Secret Server database, redirecting users to the web server at the primary site before performing any upgrade, and recreating the publication and subscriptions from new versions of the replication scripts.

In the following scenario, there are Secret Server web servers installed at a site in Australia, the U.S., and the U.K. The U.K. is the publisher node and Australia and U.S. nodes are the subscribers.

1. Redirect all users to the U.K. Secret Server URL.
2. Stop IIS at the Australia and U.S. sites.
3. Manually force a synchronization between the Australia subscriber and the U.K.:

1. Open SQL Server Management Studio and connect to the Australia subscriber database server.
2. Go to **Replication > Local Subscriptions**.
3. Right-click on one of the Secret Server subscriptions (if you used the scripts provided by Secret Server, they are called *SSPubLogs* and *SSPubEvents*) and select **View Synchronization Status**.
4. Click the **Start** button to force a synchronization.
4. Repeat sub-steps 3 and 4 for all Secret Server subscriptions.
5. Repeat sub-step 3 for the U.S. subscriber database.
6. Resolve any replication conflicts:
 1. Open SQL Server Management Studio and connect to the UK publisher database server.
 2. Go to **Replication > Local Publications**.
 3. Right-click on one of the Secret Server publications (if you used the scripts provided by Secret Server they are called *SSPubLogs* and *SSPubEvents*) and select **View Conflicts**.
 4. Examine and resolve any unresolved conflicts.
7. Generate the script to remove replication on the subscriber databases as specified above.
8. Run it to remove replication.
9. Generate the script to remove replication on the publisher database as specified above.
10. Run it to remove replication.
11. Restart IIS on the Secret Server web server in the U.K.
12. Run the Secret Server web upgrade wizard.
13. Copy the website application directory to the web servers in Australia and the U.S.
14. Use the scripts generated from the Secret Server UI to recreate replication on the publisher.
15. Push a snapshot to the Australia and U.S. subscriber databases.
16. Recreate replication at the subscribers.
17. Restart the jobs on the subscriber databases as described above.
18. Start IIS on the Australia and U.S. web servers.
19. Change redirection rules for Australia and U.S. users so they access the local Web server as normal.

Other Information about SQL Server Replication

- [Upgrading Secret Server with SQL Replication](#)
- [SQL Server Replication \(MS Books Online\)](#)
- [Snapshot Replication](#)

To back up your Secret Server installation:

Note: Your Secret Server instance may be running during this procedure.

1. Navigate to the directory where Secret Server is installed.
2. Copy the folder (holding the application) to your back up location.
3. Open your SQL Server Management Studio.
4. Right click the database your Secret Server is running on, and select **Tasks > Backup**.
5. Click the **Add** button. You will be prompted to enter a file path.
6. Make sure SQL Server has permissions for this location.
7. Copy the resulting database backup file to your backup location.

Note: You can also automate steps 2-4 using the command: `osql -S myserver\SQLEXPRESS -E - Q "BACKUP DATABASE SECRETSERVER TO DISK = 'c:\backup\ss.bak' .`

To restore your Secret Server from a backup:

Restoring the Application

1. Extract your backup zip file of the Secret Server application directory, or copy the files from your other backup location to the physical file path that your virtual directory is pointing to.
2. If you have configured encryption of your `encryption.config` using EFS or DPAPI, you will need to replace the file from the backup with the unencrypted one.
3. Check that FIPS mode is not enabled on the server to avoid an error during the process.

Restoring the SQL Server Database

Choose one of the following scenarios:

Scenario One: Database and Secret Server Are in the Same Location

1. Open SQL Server Management Studio and connect.
2. Right click **Databases** and click the **Restore Database** button.
3. In the **To database** text box, type the database name or select it from the drop down list.
4. Click to select the **Device** radio button.
5. Browse to your database backup file.
6. In the **Restore Database** window Options section, ensure the **Force Restore over Existing Database** check box is checked.
7. Click the **Ok** button.
8. If you get an error saying that Management Studio was unable to get exclusive access to the database:
 1. Right click on the Secret Server database and go to **Properties**.
 2. At the very bottom, change the **Restrict Access** property to "SINGLE_USER". This closes all other connections to the Secret Server database.
 3. Re-attempt the restore.
9. Disable **Force SSL** if there is no certificate installed on the server you are restoring to.
10. In SQL Server Management Studio, expand the databases and select the database for Secret Server.
11. Select **New Query** at on the menu bar to open a query pane.
12. Copy the following command: `UPDATE [dbo].[tbConfiguration] SET ForceHttps = 0` into the query pane
13. Click **Execute** on the menu bar.
14. After the query executes successfully, restart Internet Information Server (IIS) by running `iisreset` from the command line.

Note: If you are prompted for database credentials when accessing Secret Server and are unable to re-connect, you may need to remap the user.
15. Expand the **Security > Users** folder under the Secret Server database.

16. Remove the user that Secret Server will use to access the database.
17. Expand the **Security > Logins** folder under the SQL Server root.
18. Right click on the log on corresponding to Secret Server and select **User Mappings**.
19. Re-map the log on to the Secret Server database.
20. If necessary, activate your licenses by going to the **Licenses** page.

Scenario Two: The Database and Secret Server Are in Different Locations

1. Delete the `database.config` file from the Secret Server folder.
2. Restart Internet Information Server (IIS) by running `iisreset` from the command line.
3. Use your Web browser to navigate to the new instance of Secret Server. This redirects you to the Web installer because the `database.config` file is missing and it thinks you have not installed yet.
4. Open SQL Server Management Studio and connect.
5. Right click **Databases** and click the **Restore Database** button.
6. In the **To database** text box, type the database name.
7. Click to select the **Device** radio button.
8. Browse to your database backup file.
9. In the Restore Database window options make sure the Force Restore over Existing Database Check box is checked.
10. Click **Ok**.
11. If you get an error saying that Management Studio was unable to get exclusive access to the database:
 1. Right click on the Secret Server database and go to **Properties**.
 2. At the very bottom, change the **Restrict Access** property to "SINGLE_USER". This closes all other connections to the Secret Server database.
 3. Re-attempt the restore.
12. Disable **Force SSL** if there is no certificate installed on the server you are restoring to.
13. Copy the following command: `UPDATE [dbo].[tbConfiguration] SET ForceHttps = 0` into the query pane
14. Click **Execute** on the menu bar.
15. Navigate through the Web installer to Step 3.
16. Type the new database credentials (new server location, username, and password).
17. If you are unable to re-connect you may need to remap the user.

Note: If you are prompted for database credentials when accessing Secret Server and are unable to re-connect, you may need to remap the user.
18. Expand the **Security > Users** folder under the Secret Server database.

19. Remove the user that Secret Server will use to access the database.
20. Expand the **Security > Logins** folder under the SQL Server root.
21. Right click on the log on corresponding to Secret Server and select **User Mappings**.
22. Re-map the log on to the Secret Server database.
23. Once past Step 3, you are finished. Go to the `home.aspx` page (click the Secret Server logo). There is no need to go any further with the install because the `database.config` has been recreated with the new information.
24. If necessary, activate your licenses by going to the **Licenses** page.

There are numerous options to consider when backing up Secret Server. Backups can be scheduled to run on a specific time interval. To prevent the directory from growing too large, the number of backups to keep can be defined as well. Depending on size constraints or preferences of the DBA, the database backup can either truncate the transaction log or keep it intact. The additional schedule settings are available when "Enable Schedule Backup" is enabled, and the view page indicates the time and date of the next scheduled backup.

Secret Server can run with multiple front-end Web servers. For a critical instance, clustering offers a redundant system to limit potential down time from a single point of failure. Clustering also allows users to load balance for better performance. For instructions on enabling clustering in Secret Server, see [Setting up Clustering](#).

This topic describes the process of configuring Secret Server and SQL Server for a high-availability environment using Mirroring. The contents of this paper include:

- Configuring SQL Server 2016 for database mirroring with a failover partner and a witness
- The encryption used between the primary database and the mirror database
- Configuring Secret Server to use mirroring to achieve high availability

Note: This topic uses SQL Server 2016, but it is very similar to earlier versions.

Introduction

Three different SQL Server instances are required to implement this scenario:

- **Primary database:** The main application database
- **Mirror database:** Replicates all of the data on the primary database in a transactional manner
- **Witness database:** Monitors the health of the primary and mirror databases and initiates failover if necessary

In the setup described here, mirroring operates in synchronous mode, which means that a transaction does not commit on the primary database until it has committed on the mirror.

Note: See [Prerequisites, Restrictions, and Recommendations for Database Mirroring](#) for more on synchronous mirroring:

Procedures

Setting up Databases for Mirroring

To initiate database mirroring, the databases on the primary and secondary machines must have the same name. We recommend doing this before installation. To initially set up mirroring, in Microsoft SQL Server Management Studio, take a full backup of the database on the primary and then restore it onto the database on the secondary. When restoring the database, the "RESTORE WITH NORECOVERY" option must be selected.

SQL Server Configuration

The three SQL Server instances should all be running under the same domain account. It is possible to run under different accounts but the configuration is more complex and not supported by Delinea technical support. Each SQL Server instance should be configured to listen on TCP.

Configuring Mirroring

To configure mirroring:

1. In Microsoft SQL Server Management Studio, drill down to the primary database in the Object Explorer.
2. Right click the primary database and select **Properties**. The Database Properties window appears.
3. Select the **Mirror** page.
4. Click on the **Configure Security** button. The Configure Database Mirroring Security Wizard appears on the introduction page.
5. Click the **Next** button. The Include Witness Server page appears.
6. Click to select the **Yes** selection button.
7. Click the **Next** button. The Choose Server to Configure page appears.

8. Click to select all three interface check boxes (principal, mirror, and witness servers).
9. Click the **Next** button. The Principal Server Instance page appears.
10. Click the **Principal server instance** dropdown list to select the current (primary) server.
11. Type a port number for connecting to the other servers in the **Listener port** text box. The port must be open for TCP communication on the machine's firewall and on any network devices that restrict access to this machine.
12. Click to select the **Encrypt data sent through this endpoint** check box. This enables RC4 encryption on data sent through this endpoint.
13. Type *Mirroring* in the **Endpoint name** text box. The endpoint name is for referencing the endpoint later.
14. Click the **Next** button. The Mirror Server Instance page appears.
15. Repeat the exact same configuration you set for the primary server instance with only the server instance name different (choose the mirror instance).
16. Click the **Next** button. The Witness Server Instance page appears.
17. Repeat the exact same configuration you set for the primary server instance with only the server instance name different (choose the witness instance).
18. Click the **Next** button. The Service Accounts page appears.
19. Type the domain user that SQL Server runs under for each instance's Service Accounts text box. For example `mydomain\sql_svc`.
20. Click the **Finish >>** button. Logins are created for each account and are given CONNECT permission on each endpoint, if needed. The Complete the Wizard page appears.
21. Click the **Finish** button

Configuring Secret Server for Mirroring

Note: The credentials used to access the primary database must also be valid on the mirror database for failover to work.

1. Go to **Admin > See All**. The admin panel appears.
2. Type Database in the **Search** text box and select **Database**. The Database Configuration page appears:

Help

Secret Server supports Microsoft SQL Server versions 2012, 2014, 2016, 2017, and Express.

View [Collation Requirements](#). Need help installing or configuring SQL Server? View [Installing and Configuring SQL KB Article](#).

Database Configuration

SQL SERVER LOCATION

Server Name	QA-CUST-SQL-01
Database	SS_Playground

SQL AUTHENTICATION

- Windows Authentication using Application Identity (GAMMA\ss_iis_svc) - **Recommended**
(Advanced Setting. Windows Authentication requires a Service Account and advanced IIS setup. Detailed in [Windows Authentication KB](#).)
- SQL Server Authentication *(SQL Authentication requires Mixed Mode. See setup in [SQL Authentication KB](#).)*

[+] ADVANCED (NOT REQUIRED)

 Edit

 View Audit

3. Click the **Edit** button.
4. Click the **Advanced (Not Required)** link. A new section appears:

[-] ADVANCED (NOT REQUIRED)

SSL Encryption ? Enable

Trust Server Certificate Enable

Failover Partner ?
(Requires SQL Server Configuration change)

Multi-Subnet Failover Enable
(Enabling Multi-Subnet Failover for AlwaysOn Availability Groups requires SQL Server 2012 and higher with AlwaysOn enabled)

Connection Timeout (in seconds)

5. Click to select the **SSL Encryption** check box.
6. Type the mirror server name in the **Failover Partner** text box.
7. Click the **Save Database Connection Settings** button.

Testing Mirroring

This procedure is necessary to verify that failover will function correctly in the event that the primary server is unavailable or inoperable:

1. Open SQL Server Enterprise Manager.
2. Right click the primary database and select **Properties**.
3. Click the **Mirroring** tab.
4. Click the **Failover Now** button. This causes the database on primary to switch roles and become the mirror database. The mirror database becomes the primary. Clients using the application should be able to continue as before.

Note: One request may fail before Secret Server begins making requests to the new primary database.

Database SSL Configuration

Note: See [Enable encrypted connections to the Database Engine](#) for instruction on configuring SSL for SQL Server.

The certificate authority used for the SSL certificates must be trusted on all of the machines that are a part of Secret Server's installation. The SQL Server service account must be granted access to the certificate.

Procedure:

1. Open Microsoft Management Console by running `mmc` on the Windows command prompt.
2. Drill down to **Console Root > Certificates > Personal > Certificates** in the navigation tree.
3. Right click the certificate and select **All Tasks > Manage Private Keys**.

4. Grant the user account that SQL Server uses read permission.
5. Ensure SSL is enabled for both the primary and mirror database server. See [Configuring Secret Server for Mirroring](#). It is not necessary to configure SSL on the witness server.

Unlimited administration mode is a feature designed to allow an administrator access to all secrets and folders in their Secret Server instance without explicit permission. This can be used in the instance a company has an emergency where access to a secret is needed when no users who have permission are available. Alternately, it can be used when company policies require administrators to have access to all information in the system.

Note: An alert visible to all users displays at the top of the Secret View page when unlimited administration mode is enabled.

For a user to be an unlimited administrator they must be assigned a role with the Unlimited Administrator permission and Unlimited Administration Mode must be enabled in Configuration settings.

To navigate to the **Unlimited Administration** section, select **Configuration** from the **Administration** menu, and then click **Change Administration Mode**. We recommend administrators have specific permissions to folders and secrets and this mode is only used temporarily to assign the correct permissions.

Note: Changes to the administration mode are logged in an audit grid. The grid shows the user, time of the change, and any notes made by the user.

Best Practices

Overview

This document was written after helping many customers successfully deploy Secret Server in their organizations. It covers the issues that most customers tackle as they consider which data to store, who needs access, what permissions to apply, and how to organize all their sensitive data. This document is not meant to cover everything

Think of Secret Server as a platform for your organization to store all of its passwords and sensitive data. This means that it can be configured to work in many different ways depending on your industry, compliance requirements, and ultimate end goals. The trick is to know your objectives and then match the capabilities and best practices to your situation.

Terminology

Throughout this topic, certain terms are used to refer to specific features or concepts within Secret Server. Some of these terms corresponds to explicit roles defined within Secret Server that may be referenced, while others are broader terms that system administrators should be familiar with.

Administrator

Access to all the features within Secret Server can be granted to users by creating and assigning different roles. *Administrator* is one of the default roles that comes installed with Secret Server. By default, this role contains all role permissions, but it can be customized as well. In this guide, when it is used in the context of a Secret Server user, it is referring to the users who generally have most permissions and manage the system. Administrators have control over the global security and configuration settings.

Note: Administrators in Secret Server do **not** automatically have access to all data stored in the system—access to data is still controlled by explicit permissions on that data.

Basic User Role

The basic user role is a default role that comes installed with Secret Server. This role is a slimmed down version of the user role and primarily focuses on creating and modifying secrets, as well as limited "view" permissions. Users that have this role assigned to them also have their own personal folder.

Folder

A folder in Secret Server provides a hierarchical structure for organizing secrets. Some folders contain no secrets at all and may be used only to set permissions or policies on subfolders. Other folders may simply be a way to organize sub-folders that contain secrets. Folders are organized based on a "root" level folder structure, where "/" is the root level folder and any new folder created will be placed under that folder. Personal folders are unique and are created for each user, providing them the "personal folders" permission. Personal folders can contain sub-folders for the owner to organize their secrets.

Role Based Access Control (RBAC)

Secret Server role based access control (RBAC) is a mechanism that restricts system access to authorized users and defines what type of access a user has within the system. Often these roles correspond to features within the product and those features may give users greater privileges to make changes within the system. RBAC is a core Secret Server feature.

Secret

A secret is any sensitive piece of information (typically a password) that you would like to manage within Secret Server. Typical secrets

include (but are not limited to) privileged passwords on routers, servers, applications, and devices. Files can also be stored in secrets allowing for storage of private key files, SSL certificates, license keys, network documentation, or even a Microsoft Word or Excel document.

Site

A site is a logical work container that can tell Secret Server which distributed engines should manage work associated with specific tasks. Sites are critical to ensuring that Secret Server can manage remote network segments, alternate locations, or even DMZs. By default, Secret Server comes with the "local" site. That site is unique as it is the only site that can be configured for "web processing" or "engine processing." When the local site is configured for Web processing, the Web servers themselves act as distributed engines and are responsible for all engine work processing, in addition to the Web Server role specific work that they may be configured for. Any additional sites that are user created may only be configured for Engine processing. The "Local" site comes with two free engines under any licensing model that may be used. Any additional sites and engines must be licensed separately and will incur additional licensing costs.

User

This is the default role for new users that are added to Secret Server. By default, this role contains several permissions that enable new users to interact with Secret Server. Many of these permissions are centered around creating and modifying secrets, as well as several "view" permissions to access audit information. Additionally, access to advanced secret options, assigning secret policies, and a few other advanced permissions are assigned to this role. It also gives each user their own personal folder that is accessible only by each individual user added to the system. Besides the owner, only the "unlimited administrator" role can access these folders.

Know Your Edition

As you read through this guide, some features may be referenced that are only available in certain editions of Secret Server. To get an idea of what's available, you can reference the [main sales page](#) page online.

Installation

Before installing Secret Server, be sure to look at the [system requirements](#). The process for installing Secret Server is outlined in the [installation guides](#) matching the version of Windows Server you are using. If you have an active trial or have purchased Secret Server licenses, you can find your licenses by logging into your account at [Delinea.com](#).

Basic Configuration

Once Secret Server is installed, see the [End User Guide](#) to begin setting up Secret Server right away. This covers:

- Adding your licenses
- Basic security settings
- Configuring automatic backups
- Basic security settings
- Heartbeat
- Basic security settings
- Setting up access for local and AD users

Advanced Configuration

Secret Server's Advanced Configuration page is intentionally hidden from casual access. You have to enter a URL—the page is not accessible by clicking a link. The URL format is:

`https://<server>.<subdomain>.<domain>/<ss install>/ConfigurationAdvanced.aspx`

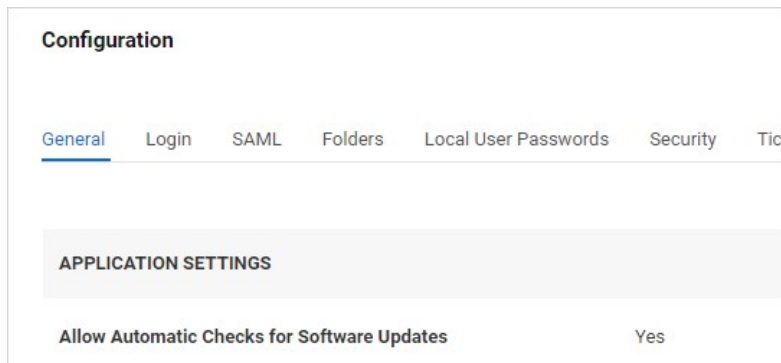
For example:

<https://qa-test.acme-east.acmewidgets.com/acmesecretserver/ConfigurationAdvanced.aspx>

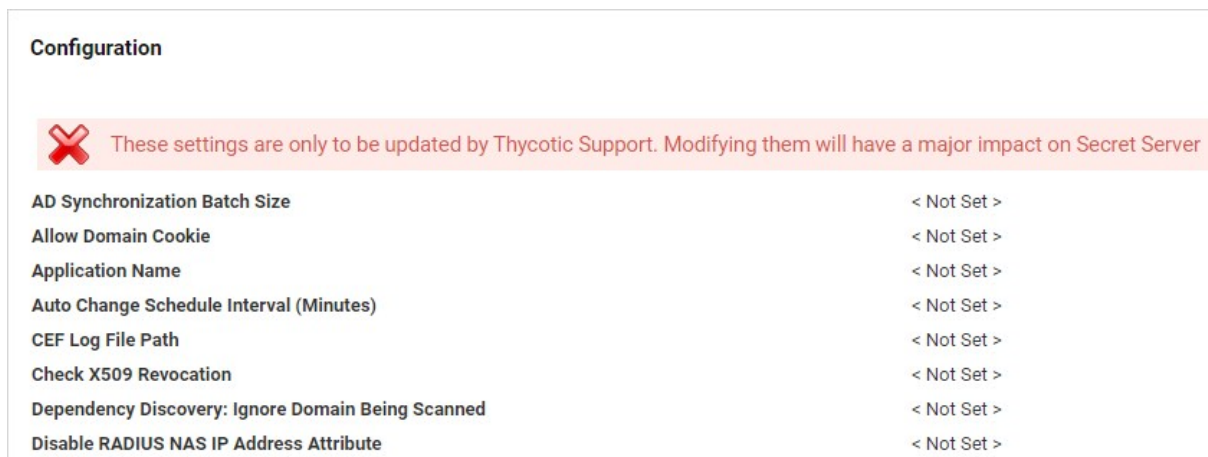
The easiest way to get to the page is:

1. Open your Secret Server instance.
2. Navigate to **Admin > Configuration**. The (regular) Configuration page appears:

Note: Administrators in Secret Server do **not** automatically have access to all data stored in the system—access to data is still controlled by explicit permissions on that data.



1. Look at the URL for the page. The file name is ConfigurationGeneral.aspx.
2. Change the name to ConfigurationAdvance.aspx, leaving the rest of the URL as is.
3. Press **<Enter>** the (advanced) Configuration page appears:



4. Note the warning at the top of the page. It is serious, but it is also not *completely* correct. There are a few settings that may be important to your initial deployment. **Do not change any settings not directly discussed here without contacting Delinea Customer Service first.**
5. The following settings might need adjustment:
 - **IP Address Header:** If you are using a load balancer and multiple Secret Server Web server nodes, it is important to set this to X-Forwarded-For. That way, user audits reflect individual user IP addresses and not your load balancer IP address.
 - **Secret Computer Matcher Once Per Discovery:** We mention this setting in the [Discovery Best Practices](#) topic, where we recommend setting it to `true` for for large environment discovery. Otherwise, the matcher runs every five hours, regardless of how

often discovery is configured to run.

You can view [reference architectures](#) for Secret Server. These reference architectures are, at minimum, refreshed every year and are created by our Professional Services Solutions Architect team. For this section, we provide some high-level architecture and design considerations that may help you design a more successful Secret Server or SSC installation.

Note: The following recommendations are primarily for Secret Server on-premises. For SSC customers, many of the recommendations are still relevant, even though you only have control over increasing distributed engines—the only Secret Server infrastructure you physically control when using SSC.

Consider some key questions about your SLA requirements for the application:

- What are the RPOs and RTOs for the application?
- Is high availability or disaster recovery required?
- Are you going to purchase Secret Server or SSC?

Answering these helps determine what initial infrastructure is needed for your environment. You can then look at the reference architectures to help select a variation of the reference architecture that works best for your requirements.

Note: Many customers take a posted variation and alter it to meet their own needs.

When the Professional Services team works with our customers, we gather both architectural and stakeholder requirements to come up with a design that is sized correctly to meet all business needs. If you are planning to design Secret Server's architecture yourself, we suggest planning additional infrastructure based on feature utilization needs in the following order:

Session Recording

This is the most process- and memory-intensive feature of the product if it is used heavily. We recommend reviewing [Session Recording Caveats and Recommendations](#) when planning to implement this feature, as it may require additional hardware or Web servers. Below are a few questions to ask yourself:

- Is the organization planning to use session recording and to what capacity?
- How many secrets may have session recording enabled?
- How many session recordings may occur concurrently?

We recommend only enabling session recording on secrets that absolutely need it—such as those with compliance or legal requirements. Otherwise, we recommend enabling session recording only on high value, high impact assets. This includes "global" admin accounts, domain administrator accounts, and other high-level privileged assets within your environment. This should minimize additional infrastructure just for session recording.

Discovery

This is another feature that can have a large impact on a Secret Server environment. A large enterprise discovering thousands of systems may require additional Web servers or distributed engines. Below are a few questions to ask yourself:

- How many systems do you intend to discover?
- How often should discovery run?
- How quickly does discovery need to complete?

We recommend using out-of-the-box discovery sources where possible. Since discovery cannot be scheduled to run at a specific time, consider enabling discovery for the first time during off-peak hours so it will run around the same time each day or week. If discovering a large number of systems, ensure you have ample Web servers and engines to handle the load. For example, increasing CPU count for each distributed engine can help distributed engines do more work in parallel. Please see [Discovery Best Practices](#) for details.

API Use Case

Employing multiple integrations with our product may impact a Secret Server environment. Below are a few questions to ask yourself:

- What integrations do you intend to use with Secret Server?
- What is the total number of API calls you anticipate per second, hour, or day?

We recommend that if you require several integrations with Secret Server where a high volume of API calls is anticipated, carefully consider how to configure your Web servers. You may want to have some Web servers dedicated to API use that have all Web roles explicitly disabled. You could place several such Secret Server Web servers in a load balancer configuration.

Remote Password Changes and Heartbeats

RPC and heartbeats may impact a Secret Server environment if used heavily. Below are a few questions to ask yourself:

- How many secrets RPC?
- How often should passwords be changed?
- How many RPC retries should be attempted?
- How often should we perform heartbeats?

We recommend carefully planning what types of secrets require different password changing schedules based on your company's information security policy. Generally, setting a large number of retry attempts for an RPC is not a good idea. The same goes for heartbeats. Match these settings to the business use case, such as 10 password retry attempts and having heartbeats occur once per day. These small refinements can greatly reduce the load on Secret Server. If you determine you need aggressive RPC and heartbeat schedules, consider having additional Web servers and distributed engines to handle the load.

Proxying

Heavy proxying can impact Secret Server infrastructure. Below are a few questions to ask yourself:

- How many systems are proxied?
- Are they SSH or RDP connections?
- What is the concurrent need?

We recommend proxy connections go through a distributed engine whenever possible. This offers a security advantage because ports, such as 3390 or 22, are not open inbound directly to your Web servers. You can review [SSH Proxy Configuration](#) to size proxy requests.

General On-Premise Considerations

The areas mentioned below are often where we spend the most time with customers who have spent professional services time performing architectural health checks. These are the main areas we typically improve:

- Ensure that you have a database maintenance plan in place for Secret Server. It should be implemented or reviewed by your organization's DBAs. Adjusting the data retention settings is not enough and does not substitute for having a maintenance plan.
- Ensure that RabbitMQ clustering configurations have work distribution policies in place. In most engagements, we use the [AutomaticSyncMode](#) policy.
- When designing multi-site, single-instance Secret Server implementations, be cautious when configuring Web servers and enabling roles on all Web server nodes when inter-location latency is high (50 ms or greater).
- When using the "local" site for Web processing when also using sites with distributed engine processing, consider using engine processing for the local site too. In most cases, when using both Web and engine processing, you are using both a built-in message broker (MemoryMQ) with (RabbitMQ).
- Consider having dedicated systems for Secret Server components, as proposed in our mid-range reference architectures. If using RabbitMQ, put it on a dedicated system that is not shared with the distributed engine service. Secret Server
- For environments that contain 75,000 to over 100,000 secrets where secret searches are noticeably slow, we strongly recommend

running `SecretSearchPerformance.sql` against your Secret Server database, which can be found in `C:\inetpub\wwwroot\SecretServer\Database\SqlServer\OptionalOptimizations`.

- For large environments where discovery, RPC, and heartbeats will be used simultaneously, carefully consider when to run discovery. Discovery can compete with RPC requests when both features are using the same site. For large environments, you may want to have a dedicated site and distributed engines for discovery and a separate dedicated site for secrets.

Security is a process—not a product. Take a look at the [Security Hardening Guide](#) to ensure your implementation of Secret Server has optimal security. The guide contains more in-depth recommendations for not only configuring the application in a secure manner but also hardening the server or servers Secret Server is hosted on. That guide complements the information provided here.

One of the most important areas for Secret Server hardening is protecting the `encryption.config` file that is created during installation. After the product is installed, this file exists in the main `\SecretServer\` directory. It is a very important file. This file (unencrypted), along with a backup of your Secret Server database, is all you need to get a Secret Server environment back up and running. Thus, it is imperative that you protect it. There are two ways to protect the `encryption.config` for on-premises Secret Server and two others for Secret Server Cloud.

Secret Server On-Premises

For an on-premise installation of Secret Server, we recommend protecting your `encryption.config` file with an HSM. When using an HSM, though, there are other things that you should be mindful of:

- Is the HSM highly available?
- Is the HSM capable of handling a high volume of access requests?
- What methods are available for retrieving the key from a backup if my HSM were to crash?

Important: If your HSM is down and you do not have backups, there is nothing we can do to help recover your data. Carefully consider the configuration of an HSM for protecting `encryption.config`.

A second, less secure, option for protecting the `encryption.config` file is to use DPAPI combined with EFS. DPAPI is a setting that is enabled on each Web server within your Secret Server cluster. EFS adds an additional password to the `encryption.config` file. It is worth noting that both protection mechanisms can be compromised if an attacker were to log on interactively to Secret Servers Web servers and become a local administrator. Give careful consideration to securing remote access to Secret Server when leveraging DPAPI and EFS.

We recommend storing an unencrypted copy of the `encryption.config` file for disaster recovery scenarios where the Secret Server Web server is irrecoverable. Make a backup of this file immediately after installation (before securing it with a HSM or DPAPI + EFS) and to store the file on one or more media devices such as a hardware encrypted USB drive. The device should then be placed in a secure location, such as a safe. Access to the device should go through a chain of custody process in the event of an emergency where the original file is needed.

Secret Server Cloud

If you are using Secret Server Cloud, there are two main methods for protecting your `encryption.config` file:

- Delinea owns your `encryption.config` file and is responsible for keeping it secure. We put internal mechanisms in place to ensure that Delinea does not have access to your data without your explicit permission. Delinea
- You configure a connection to AWS KMS to protect the `encryption.config` file. The master key is stored in AWS and under your complete control, inaccessible to Delinea staff

It is important to have a privileged account management (PAM) strategy that helps you determine which types of features to leverage for your various accounts and sensitive data you will be storing. Below are some suggested guidelines for creating a strategic plan. We recommend reading all sections of the guide for a comprehensive look at ways you can secure your Secret Server. However, these guidelines will also link to other parts of the guide so you can choose to jump to a specific section for more detail about a particular topic.

Identify Data at Risk

Consider all the types of sensitive data your team needs to be securely stored and managed. Where are the biggest risks and pain points in your current password management strategy? Data at risk also often includes more than just passwords.

To get started, think through these key accounts and principles:

- All shared privileged accounts: these are accounts that don't identify an individual (for example: administrator, root, enable, service accounts). All of these should have randomized passwords that are changed frequently.
- Do your users have individual privileged accounts? Maybe each user has—separate AD account for domain admin rights?
- Every password in your organization should be different.
- Do your users have individual privileged accounts? Maybe each user has—separate AD account for domain admin rights?
- What passwords could be needed in an emergency, outside of regular business hours, or when someone is on vacation?

Typical account passwords and sensitive data being stored in Secret Server:

- Active Directory domain administrator accounts
- Active Directory service accounts
- Application passwords (such as SAP and, custom apps)
- Cloud Administrative or Privileged Accounts
- Database accounts (such as MS SQL, Oracle, or MySQL)
- Network equipment passwords (such as router, switches, firewalls, phones, and appliances)
- Sensitive files (such as private key files, SSL certificates, and network documentation info)
- Software license keys, serial numbers, personnel data, and Wi-Fi passwords
- UNIX, Linux, Mac root, and local user accounts
- Website passwords (cloud services, DNS, Amazon AWS, vendors)
- Windows local administrator accounts

Who Accesses Secret Server?

After determining the data you will store in Secret Server, the next step is to decide [who](#) will use Secret Server to access and manage that data. A common approach is to begin by focusing on one group of users and the passwords they use on a regular basis, later expanding to other teams once a good strategy has been put in place. However, you may find it more beneficial to organize Secret Server for use by all of your users/teams at once so you can design an effective overall folder and policy structure that will work well across all teams.

What Privilege Levels Are Necessary?

Giving a user access to an account in Secret Server can entail different levels of privilege. Do you want a user to be able to edit the username, machine, or password of a secret, or only view the secret? Should they be able to share the secret with other users? Once you incorporate use of the Launcher into your users' workflow for authenticating to an application, do they really need to know a password, or can you mask it? The [Workflow Security](#) section can help you determine and implement key measures to ensure users have least privilege necessary.

What are your Password Requirements?

It's unlikely that all your accounts will have the same [password complexity requirements](#) and [rotation schedule](#). In fact, for best security, you should have some variation. You can create sets of password requirements to control password length, characters, and complexity, then apply those to various account types using [secret templates](#). Secret templates also allow you to set a default expiration period, which can translate to how often an account password will be changed automatically.

Evaluate your Existing Setup

While transitioning to using a new tool for managing your passwords, it is important to take into account how accounts are currently used in your environment. The following questions can help evaluate this:

- Do some of your users have their own, individual AD domain admin accounts, or are there only a few shared domain accounts?
- Do users use local administrator accounts or privileged domain accounts for admin access to systems?
- Are permissions to resources (such as servers and applications) controlled using AD group policy?

Define Your Core PAM Strategy

There are a few different strategies that typically work best in Secret Server. Other methods of password management may work but require a more significant amount of time and effort to configure and maintain. The most commonly-used strategies are defined below.

Individual Privileged Domain Accounts

In this scenario, IT team members have their own domain admin accounts that are tied to their identity. They use these accounts to gain elevated privileges to resources such as production servers. Permissions to the various resources they're permitted to access are controlled by AD.

To implement this in Secret Server, each account is stored as its own AD account secret. Only the user tied to that account is granted permissions to the secret. A security setting such as check out (one-time password) or hide launcher password is enabled so the user depends upon Secret Server to use the account. Therefore, all access to that account will be audited. When the IT admin needs elevated privilege to a box, they check out or view the secret and then use the launcher to access the machine.

A benefit of this strategy is that there is not conflict with multiple users trying to use the same account for access to one machine. This strategy provides great accountability—the security team knows the exact user accessing an account and the machine being accessed. The password is not shared among multiple users, and all privileged access is audited by Secret Server.

A pitfall of this strategy can be that there is more management of permissions required in AD. While machines could be access or deny listed to force users to use the Secret Server launcher, thus controlling machine access through Secret Server, this can be tedious.

It is more secure, less work, and simpler to organize permissions for access to domain resources in AD. This strategy works best for organizations that already use AD heavily to control permissions of individual privileged users to domain resources. Ongoing maintenance will rely on updating permissions to resources in AD and ensuring that all new individuals' privileged accounts are being added for management under Secret Server.

Shared Privileged Domain Accounts

You may choose to have your users use shared privileged accounts to access resources. This strategy involves creating a few service accounts that have permissions to OUs or groups of computers. In Secret Server, these accounts can be limited with the Launcher so they can only be used to Launch to certain computers. This means you can limit the number of domain accounts created and set permissions more broadly (such as at OU level). These passwords could be changed on a schedule or, where possible, used with Check Out to change the password after each use. Using this setup, accounts can be designated for team or function and can have varying Check Out intervals set to ensure that only one person at a time is using each account.

A benefit of this strategy is if individuals do not already have their own privileged domain accounts in AD, then giving them access to shared accounts means less setup in AD while still maintaining accountability for who uses which account, and which machine they access.

A pitfall of this strategy can be that if the team (or function-specific accounts) cover a broad number of machines that can be accessed, it may be a lot of work to set up launcher allow/deny lists to control access through Secret Server. However, if these permissions are set only through AD, it will be difficult to have the visibility into these limitations for an auditor.

Hybrid of Individual and Shared Accounts

Sometimes, your employees' roles may require longer, more specialized access. For those accounts, you can have individual privileged domain accounts, and for the other regular users you can use a few shared privileged domain accounts. All of these can be stored in Secret Server, but with different settings governing their usage. For example, the shared accounts would still have check out enabled, while the

individual privileged accounts will simply have permissions limited to an individual user, possibly with the password hidden using hide launcher password.

What Is the Highest Risk?

Implementing a comprehensive PAM policy should eventually cover all of your privileged/shared accounts, but this can take some time. When looking at where to start, it is important to consider the areas of risk—where are the areas that need more immediate attention:

- Is it local Windows admin accounts all sharing the same password?
- Pass-the-hash vulnerability?
- Protecting your network equipment passwords?
- Avoiding fines for not meeting compliance mandates?
- Password misuse and auditing employee access to accounts?

Choose a starting point that gives your organization the most value, and then branch out from there.

At minimum, the administrators who manage and use your organization's privileged passwords and data on a regular basis will need to access your Secret Server. Secret Server users can be defined in a few ways:

- Active Directory user accounts.
- Local Secret Server user accounts.
- User accounts from an Azure Active Directory tenant.
- User accounts from another LDAP source (Basic/Kerberos).
- User accounts from SAML integration (often AD accounts). If local accounts are provisioned via SAML, they must correspond and match local user accounts that are within Secret Server. SS also has the concept of groups, which can be local (you create them in SS), AD-synced (security groups from AD), oLDAP groups, or AzureAD groups. Groups are a powerful tool for assigning and maintaining permissions to secrets, and therefore should be given careful thought and planning. Below we review the two most common account and group strategies our customers use. These same concepts can apply for other directory service accounts and groups other than Active Directory.

Local Secret Server Accounts

Local users and groups have to be created and managed manually in Secret Server, as they are not integrated with AD. The first account you create in Secret Server is an example of a local account. Local groups can include local users and AD accounts, and can have a user established as the group owner that is permitted to add or remove users to or from the group.

Active Directory Accounts

AD accounts can be added for access to Secret Server either manually (one by one) or by AD security group. When adding users by security group, you choose which groups Secret Server will synchronize with AD to update which users' access to Secret Server is enabled or disabled. AD group synchronization happens on a regular, customizable interval to keep group membership changes that happen in AD up-to-date in Secret Server as well.

Local or Active Directory Accounts?

We recommend using one of these options:

- Only local users and groups (best security)
- Only AD users and groups (most convenient)
- A hybrid of AD users and local groups (balance of security and convenience)

You need to choose an option that provides the levels of security and convenience that are acceptable for your organization. Using the AD

accounts option is easy for user maintenance, but it limits the security of Secret Server to the level of security of your AD. This may be fine—just be sure to consider the question of domain admin access to AD in combination with Secret Server permissions.

Only Local Users and Groups

Creating local users and groups within Secret Server provides a lot of flexibility because you can tailor permission assignment by group to your exact needs. The major benefit of local users and groups is security: users and group membership can be controlled entirely by role-based access control (RBAC) within Secret Server. However, this approach requires more maintenance because creating or deleting users and managing group membership has to be controlled in Secret Server.

Only AD Users and Groups

If you are considering using AD users and groups for Secret Server access and permissions assignment, review your teams that need access to Secret Server. Compare them to the corresponding groups in your AD. If your AD groups map to ways you want to assign access to secrets, you can synchronize your AD groups with Secret Server and start assigning permissions to secrets (and levels of those permissions—View/Edit/Owner) by group. You can then effectively manage Secret Server access and secret permissions completely from AD by changing AD group membership.

Many customers choose this option because they can maintain control in AD and do not have to worry about any user or group maintenance within Secret Server. If you want to use this option but your AD groups don't match the way you want to assign secret permissions, you will need to create new AD groups to match this, or may want to consider the hybrid approach (below), using local groups instead.

This method, while more convenient, may require additional considerations:

- How are these AD groups being protected?
- Are there controls in place which require elevated or high privilege accounts to modify these AD groups?
- Are there alerts in place for when these groups are modified?
- Is the information security team closely monitoring these groups?

Hybrid of AD Users and Local Groups

A third option is to create local groups in Secret Server and add AD users to those groups for the purpose of organizing how permissions are assigned to secrets. Many customers who use this setup will create a single AD security group (for example, SecretServerUsers) to use to synchronize their AD users with Secret Server for log on. They then create additional local groups for their users to, which gives them permissions within Secret Server, such as to their teams folder. They may also be added to other Secret Server groups that provide them with other privileges within the environment.

This approach is more secure than using only AD groups and users, but if Active Directory were compromised, intruders may still be able to reset an account password and gain log in access to Secret Server. If secrets are stored in that user's personal folder, those secrets may be compromised which may lead to lateral movement elsewhere within the organization.

Business Users

A *business user* is a non-IT user, such as sales team members, office managers, data entry clerks, and marketing team members. They are allowed to access and use Secret Server for managing non-privileged accounts, including individual or team application accounts and credentials. Non-privileged accounts include email login, social media password, productivity software credentials, and more.

Business users are not permitted to manage privileged accounts, such as database server credentials, security appliance passwords, and cloud service root keys. Business users are also not permitted to administer Secret Server.

Business users can:

- Access secrets: They can create, update and delete their own secrets within Secret Server. For example, A user signing up for an online service can use the password generator to create a strong password, store that password in Secret Server, and use the Web Password

Filler to log in later.

- Request and approve access to secrets: Non-privileged secrets may need approval workflows. Business users can request access to these secrets and can act as approvers. For example, if access to an organization's social media accounts requires authorization from a member of the marketing team, a business user can request access to the secret, and another business user can approve access.
- Share secrets with other users: Business users can share non-privileged secrets with other users of Secret Server, whether or not they are business users.
- Access secrets using the mobile app: Business users can use the Delinea PAM application to access and manage their secrets.
- Use launchers: Business users accessing secured, but not privileged, systems can use Secret Server launchers. For example, a user in a secure network segment who can only access an application via RDP may use an RDP launcher, as long as the tasks they are performing do not require privileged access.
- View audits of their secrets: Auditing capabilities are included for business users, allowing them to see who has accessed their shared secrets.

Defining your authentication strategy ensures you have standardized authentication practices in line with your Secret Server RBAC scheme. Secret Server offers a wide variety of authentication options that can add flexibility and security to your end user's authentication process.

Strong Authentication

Protect the tool you are using to secure your privileged accounts by adding a second factor of authentication for users logging into Secret Server. Two-factor authentication can be added whether users are logging in with local or AD accounts. For more information about using two-factor authentication with Secret Server, see the [Security Hardening Guide](#).

SAML

If your organization is already using SAML for SSO across your organization, it might be a good option for Secret Server authentication too. SAML uses browser-based communication, between the service provider (Secret Server) and the identity provider (SSO providers) to broker authentication. For more information on configuring SAML with Secret Server, please see the [SAML configuration guides](#). The major benefits SAML provides are:

- A consistent MFA strategy across all applications in your environment.
- Simplified authentication communication: The browser handles the process. For SSC, if the authentication strategy is to authenticate against the domain, that communication must flow from SSC to the distribute engine and finally to the domain controller and back for authentication. SAML shortcuts this by having the browser communicate to the service provider and identity providers, reducing authentication latency.
- Easy to configure, manage, and add new users.
- Supports multiple MFA options based on conditional access. For example, a user may only need to verify with one factor for accessing less critical apps, but Secret Server uses two-factor authentication.

Directory Services

Secret Server provides a multitude of authentication options through directory services. It can sync users into the application from various LDAP sources. It is important to use an outside authentication source to automate user provisioning. The User Account Options setting in the directory services configuration provides these options:

- Users are Enabled By Default (Manual)
- Users are Disabled By Default (Manual)
- User Status Mirrors Active Directory (Automatic)

There are benefits to each strategy, but the last one is usually best. Using a hybrid group structure prevents new users from gaining permissions before they have been reviewed, and if they are disabled in active directory, they are automatically deprovisioned from the system. This provides automatic user management and easier offboarding strategies.

If you decide to use a manual strategy, we suggest using the automatic user management feature to disable users who have been inactive for a defined period of months. This can prevent long-inactive accounts from being compromised and keep your list of active users current.

Roles control which features of Secret Server a user is able to use, view, or administer. Existing roles can be customized, and new roles can be created as needed. Secret Server comes with several roles by default, including administrator, user, and read only. You should review the default roles and decide whether your organization needs further roles for various purposes such as third-party consultants or auditors.

Note: Users with the default administrator role (which contains all role permissions available) do **not** automatically have access to all data stored in your Secret Server. secrets are only visible to a user based on the explicit secret permissions assigned to them.

We strongly recommend pulling one or both role permissions pertaining to unlimited administration mode out of the default administrator role. Unlimited administrator mode is a "break-the-glass" feature that allows a user to view all secrets in Secret Server. By splitting the unlimited administration permissions into separate roles, it ensures no one user can both turn on the feature and operate as the unlimited administrator.

Commonly, operational employees are assigned to the "unlimited administrator" role and a CISO or senior manager that is responsible for Secret Server is assigned the "configure unlimited administrator" role.

Note: For more information about how unlimited administration mode works and how to effectively control the relevant role permissions, see the [Security Hardening Guide](#).

Other sensitive roles you may want to directly assign to individuals include:

- Administer Role Assignment
- Administer Role Permissions
- Bypass SAML Login
- Create Root Folders
- Delete Secret

For administrator-related roles, we typically recommend having these associated to your planned internal Secret Server subject matter experts. Usually customers have a dedicated group or groups associated with administrator only roles and functions. You can create administrator tiers depending on the size of your organization and the tasks you expect the administrators to perform. This can reduce administrative overhead and provide a path for employees to gain further experience with the product.

We recommend creating AD groups for these administrative roles to ensure that these groups are protected and can only be modified by higher-privilege accounts. Ensure proper monitoring and alerting is in place when creating groups that are intended for high-privilege role access within Secret Server.

Role Definition and Assignment

Once you have defined your roles, they will seldom need to be changed. Access to modify and assign roles should be tightly controlled.

Group Assignment

If roles are assigned to groups, then assignment of the groups will also need to be controlled. Often very sensitive role permissions, such as unlimited administrator, are assigned at the user level to limit the risk of granting group assignment permissions. Roles that are individually assigned should be routinely audited at least once a year to ensure users are only assigned the permissions needed for their job.

You have different sets of passwords that should only be viewed by particular administrators. You may also have passwords that should be

read-only to some administrators, editable by others, and not even visible to still others. All of these options are possible using the permissions within Secret Server.

Permissions can be allocated at the individual user level but it tends to be easier to manage over time if you allocate your permissions at the group level. You will need to decide if your existing AD groups could work for these permissions or if you need to create new AD groups or if you want to create and manage local groups in Secret Server.

For more information about what each level of permissions entails, see the [Secret Server Role Permissions List](#).

Using Folders to Control Access (Inherit Permission)

You can apply permissions (View/Edit/Owner) at the secret level. This allows you to apply very granular permissions on a single secret if needed. Managing permissions on each secret is powerful for situations where you need that flexibility, but it tends to be harder to manage over hundreds or thousands of secrets. Instead, you should consider using folders to control permissions for most secrets. This can be done by creating a folder structure that best represents your organization, teams or data being stored and then applying permissions on the folders, using inheritance across folders where appropriate. Secrets placed in a folder can then inherit the permissions of the folder.

Deciding on your Folder Structure

The folder structure creates a hierarchy for organization and permissions. This means that folders near the root level need to break out access in high level terms and then get more specific permissions (typically breaking inheritance) as you move down to the "leaf level" sub-folders.

For example:

- Customers
- Human Resources
- Information Technology
 - Development Services
 - Programmers
 - Technical Services
 - Database
 - Oracle
 - SQL Server
 - Systems
 - Network Infrastructure
 - Unix
 - Windows
- Vendors

The most typical configuration is to break out the folders based on the teams that need to use those folders with the most restrictive permissions at the deepest subfolders of the tree.

For instance, an Oracle DBA might have the following permissions on the above folders:

- Information Technology (view)
- Database (view)
- Oracle (view/edit/owner)
- SQL Server (view/edit)
- Technical Services (view)

Note: If the "require view permission on a specific folder for visibility" setting (Admin > Configuration > Folders) is enabled, a user

cannot see the full folder structure unless they have view permissions on all the parent folders of a folder. For example, a user with view on the Oracle folder in our example, would also need view on Database, Technical Services, and Information Technology to see the full folder path.

There are settings under **Admin > Configuration > Folders** to control whether inheritance on folders and secrets should be turned on and also whether users should always see all folders. There are many ways to configure this for your organization.

The most common approach is:

- Use inheritance.
- Do not allow users to see folders unless they explicitly have view permission by enabling the "require view permission on a specific folder for visibility" setting.
- Require all secrets to have a folder.

Note: Consider using our [User Teams](#) feature to align your groups within Secret Server to a team. This can help prevent users from sharing secrets with other individuals outside of their own team.

This approach allows different teams or even different departments within your organization to use the same Secret Server instance independently.

If a business need arises to break permission inheritance on a folder or secret, we recommend tracking or auditing those folders because manually applying permissions can increase your administrative overhead.

A *secret policy* is a set of security and remote password changing settings that are normally applied to a secret on the Security or Remote Password Changing tabs. The benefit of using a secret policy is not only that settings can be applied in bulk to secrets (that is, by folder), but that these settings can also be enforced, preventing users from changing them.

Secret policies should be established to apply settings to secrets that are key to the workflow your organization is working toward. For example, if your primary concern is more detailed auditing around service account usage and you also have a requirement that all service account passwords change every 60 days at 2 A.M. on the next Tuesday, you can create a policy that includes these settings and apply it to the folders that will contain all of your service accounts. Whenever new accounts are added to the folder, such as when they are imported via discovery, the settings will automatically be applied and enforced.

Secret policies can also be updated after they have been assigned to folders. Therefore, if your password policy changes and you need your service account passwords to change every 30 days, you can update the policy and it will immediately apply to all secrets the policy is assigned to.

As with permissions, secret policies can be inherited too. Be mindful of where you disable secret policy inheritance to ensure that exceptions to secret policies are tracked. Also, disabling secret policy inheritance may lead to increased administrative overhead.

This section discusses some key best practices around using Secret Server's Discovery feature to find and manage accounts in your environment. See [Further Resources](#) for a link to the comprehensive guide to configuring and using Discovery.

Discovery Workflow

While it may be tempting to immediately get started using discovery to get your accounts under control, there are a few things you can do ahead of time to make the enforcement of your organization's password policies more streamlined:

- Know which secret template you want to import accounts to. This can effect password changing and Launcher settings that are applied to your imported accounts.
- Have a folder structure established so you have folders appropriated for each type or category of discovered accounts.

- Apply a secret policy to the folders you import to.

Having these settings in place can save you the considerable amount of time it could take to have to re-organize all of your accounts and policies post-import.

Enterprise Deployment Considerations

We broadly recommend starting small and choosing specific objectives when working with discovery. If you are an organization that has 15 domains, for example, you may choose to first work with discovery within the domain you are most concerned about. Make the objectives even more specific where possible. An example first objective might be to configure discovery for finding all local administrator accounts on all your servers and creating discovery rules for ensuring that new servers have their password changed shortly after being built. Systems with internal elevated risk may also be a good place to start. Other examples are provided below.

Cloud Accounts

In more recent Secret Server versions, we support discovery of [Google Cloud Platform](#) service accounts, VM instances, and [AWS Instances](#).

Local Windows Accounts

How many local Windows accounts in your environment use the same password? Are they local admin accounts? Use discovery to quickly mitigate the risk of pass-the-hash attacks by finding all of your local Windows accounts and setting their passwords to unique, strong passwords managed by Secret Server. Where your admins previously had to remember one password to access all machines with local admin rights, they now have to remember zero passwords because they can use Secret Server to find the machine and launch an RDP session using the local admin account without ever knowing, copying, or typing the password.

Find Backdoor Accounts

Ensuring that users are not creating backdoor administrative accounts on Windows machines is very important as these can compromise general security as well as open the potential for a user to access a machine directly without being audited. By running discovery on a regular interval and having discovery rules alerting you when new accounts are found, you can ensure that users any new local Windows account being created are identified in addition to being either removed or brought into Secret Server.

Service Accounts

Many organizations do not know where their AD service accounts are being used across the network. By using discovery to scan your network, you can find all of the Windows services, application pools and scheduled tasks that are run by AD service accounts. Once these accounts are found and brought into Secret Server, having discovery run on a regular basis will find any new locations where the account is being used since they were added to Secret Server. With discovery rules, those additional dependencies can be automatically added to the existing secrets. We recommend making sure that the service account discovery has run before using Secret Server to change the service account password.

Unix Accounts

When scanning for Unix accounts, we recommend using SSH key validation, as discussed in the [Security Hardening Guide](#). This ensures that you are only connecting and trying to authenticate to UNIX servers that have a valid and trusted SSH key.

ESX/ESXi accounts

Local accounts on ESX/ESXi systems should not change once the server is set up and configured. We recommend creating discovery rules that monitor your ESX/ESXi servers and email the proper teams to inform them of any new account found. These accounts really should not be created, so it is important to monitor them and ensure that no one is creating them maliciously.

Often you will have situations in which you want users to have access to accounts, but only under certain circumstances, such as on a specific day or after the approval of a manager. Maybe your compliance requires that you have the ability to monitor an active RDP, or that you use a one-time password for certain accounts. This section examines best practices around workflow security settings in Secret Server as well as scenarios when these settings are commonly used.

Hide Launcher Password

Many times, giving an employee access to a resource through Secret Server does not require that he or she have access to the actual password for the account used. As long as the application a user needs can be started by the launcher, there is no reason the user needs to copy/paste or type the password. The hide launcher password setting implements the following:

- Users with access to the secret will see only asterisks (****) in the password field
- There will be no copy-to-clipboard, field history, or unmask icons next to the field

Note: Users with edit permissions to a secret with "hide launcher password" enabled can still view the password when editing the secret. To prevent all possible access to the password, limit users to view permission only.

This can be an important way to reduce exposure of your privileged account passwords. Hiding launcher passwords can be enabled for secrets under the Security tab of a secret or by applying a secret policy. You can also remove the ability for a user to see the password for any secret with a launcher by removing the "view launcher password" permission from their role.

Require Approval

The "requires approval for access" setting is typically employed in the following cases:

Simple approval workflows:

- When a user should be required to request access to a secret for a certain time period
- When an administrator would like to approve a user's access to a secret in advance for a time in the future (such as a maintenance period outside normal business hours)
- When a group of administrators would not like anyone to access a secret without the approval of another administrator

Advanced approval workflows:

- When requiring a multi-tier approval process that involves having more than one individual approve access to a secret
- When requiring multiple workflow steps, each with different reviewers and a varied number of required approvers
- When selecting "owners" as a review group

This setting can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy. When enabling "requires approval for access," remember that users will still need to have at least view permission to the secret to request access to it. Once access has been granted to the secret, they have whichever level of permission was assigned to them for the secret (view, edit or owner). The approvers of the secret are specified when enabling the setting, and these individuals will be able to modify the time that the requestor originally submitted their access request for or deny the request altogether.

Note: Please see our documentation on [Secret Workflows](#) for more advanced workflow use cases. For highly sensitive or privileged accounts, we do recommend implementing multi-tier approval processes where possible.

To require all approvers of a secret to also request access from another approver, be sure to enable the "owners and approvers also require approval" setting.

Require Comments

Requiring comments to be entered when viewing a secret can be an excellent way to ensure users are accessing a secret for legitimate reasons. You can even view the comments in the audit of the secret to historically track if a secret was accessed for the originally intended purpose. Managers can routinely review these comments and determine where employee training may be required.

A common example would be enabling require comments on a domain administrator account that is stored in Secret Server: A user may enter a comment that indicates he or she needs to use the domain administrator account to "perform adding a user to a group." In many cases, a domain administrator account should not be used for this purpose and often this work can be done with a lesser privileged account within the environment.

Requiring comments can also be combined with [Ticketing System Integration](#). This is a great way to align secret access with a valid ticket number and a comment. This can help with compliance and track usage of a secret tied to a specific task, which may provide more granular information as to why a secret is needed.

Check Out

There are times when users need to be able to access a password directly, but you still want to have control over how long they are able to use the account without the need to approve access each time. In this case, hiding the launcher password is not a possibility, but there is also concern about having the user know what the password is after they are done using it. Another concern is often the risk of the hash of the password being stored locally on remote devices after each use and potentially being vulnerable to a pass-the-hash attack.

"Check out" is a security setting that means:

- Only one user at a time has access to a secret
- A user can only access the secret for a predetermined check out interval, such as 30 minutes
- At the end of a check out interval (check in), or when a user manually checks in the account before the time is up, the secret is available for check out by another user
- When enabled, the password can also be changed automatically upon check in

Domain administrator accounts are a great example of a case in which using check out to change the password every time it is used can be extremely beneficial. This ensures that users are not copying the password to Notepad or writing it down for later use and also invalidates the hash that was stored on the remote machine after a remote desktop session.

Check out can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy.

Session Monitoring

For critical systems and highly privileged accounts, sometimes simply having an audit trail showing when someone viewed the account in Secret Server is not enough. Maybe the auditor also wants to be able to review what was done with the account on a remote session. For these critical secrets, it is recommended to enable session recording for the secret. When session recording is enabled, all launcher sessions can be recorded for later viewing by the auditor or manager in the event they need to investigate the actions performed during a remote session.

What constitutes a "critical system" is subjective. Departments may define this differently, so having them involved in those discussions can be helpful. Some of these systems may be explicitly selected based on risk, compliance, or from the auditing team.

One of the most important things is to not consider all accounts or secrets critical, enabling session recording for them all. That can cause performance issues within your environment—session recording is the single most intensive feature of Secret Server.

Before enabling session recording, you may want to evaluate your users' roles to determine who can monitor real-time sessions and view recordings. The permissions to look for are "administer session monitoring," "view session monitoring," and "view session recording."

Session monitoring can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy.

Secret templates in Secret Server define the types of data (secrets) that can be stored, and the settings for that data. You can store just about any type of sensitive data in Secret Server.

It is important to review the available templates and decide which ones should be available to your users as well as where you would like to

make changes to the default templates included.

Configuring Templates

You can customize existing templates or create new templates if necessary. Many templates are included by default that cover common account types. For example, the AD Account template contains the following settings:

- Domain, username, password, and notes fields
- 30-day expiration period, applying to the password field
- RDP launcher, requiring user input for computer to connect to
- Password changing and heartbeat enabled
- AD password changer, with default password requirements

These settings are typically sufficient for most organizations to use out-of-box. However, you may wish to enable other settings or change settings such as enforcement of a naming convention or more complex password requirements. In this case, you have the flexibility to either modify the existing template, copy the existing template to use as a base for a new template, or create a new template from scratch. The following sections cover some fundamental template settings available for you to customize.

One best practice we often recommend is simply leaving default templates the way they are and duplicating the templates you plan to use. Then customize the newly duplicated templates as needed. Name them something your employees will recognize and readily use.

File Attachments

Do not forget files. You can have fields on your secret template attaching files. This can be used for storing license key files, private keys, SSL certificates, even Microsoft Word or Excel documents that contain sensitive data.

Naming Patterns

Secret Server supports enforcement of naming patterns for secret names. Naming patterns allow you to maintain consistency for secret names and can help ease both browsing and grouping secrets by name. Naming patterns use regular expressions and allow you to enter a descriptive error message to describe your naming standard to users. The most common naming standard used is RESOURCE\ACCOUNT (for example, server0001\administrator). You can find this setting by clicking Edit from the template designer page.

Password History

Secret Server automatically keeps all history on all fields on a secret template. This means that all previous values for machine, username, password and any other fields will be kept. This helps ensure that previous passwords can be found if needed.

Password Requirements

Password Requirements determine the password compliance rules (for example, 16 characters, one uppercase, one lowercase, one symbol and one number). These can be customized and applied to passwords at the secret template level or per individual secret (under the Security tab). This controls the complexity of passwords generated by Secret Server. Password requirements can also be enforced when users try to edit or create new passwords, and can be viewed for password compliance in reports. This allows you to have different complexity rules for different types of passwords if needed (such as Oracle, SQL, Windows, and UNIX). You can choose to have Secret Server enforce the password requirements on add/edit by turning on validation on the secret template (click Edit from the template designer page).

We have added new password [validation criteria](#) to recent versions of Secret Server, which further helps create unique passwords. We also added [custom password-exclusion dictionaries](#) that can help personalize which words may never be used as part of a password that is generated.

Talk to your security management, auditors and industry experts to find out the best password complexity settings for your environment. Do not hesitate to stipulate complex passwords, such as 100-character random-generated passwords with symbols, alphanumeric uppercase and lowercase)—using a platform like Secret Server makes it easy to work with passwords so complexity and length do not matter (for

launchers, copy-to-clipboard, and auto change). In fact, very large passwords can add to security since administrators will be far less likely to remember them or write them down or want to type them.

Another thing to consider when creating password requirements is which character sets should be used. Some systems may not work well with certain characters. For example, underscores can be problematic in certain mainframe environments. You can create your own character sets (Admin > Secret Templates > Character Sets) for use in password requirements. These can then be used when passwords are generated by Secret Server.

Secret Expiration

Secret Server uses expiration to ensure that passwords are changed on a regular basis. Secrets can be set to expire on an interval such as 30 days (or other intervals as needed). Expiration is often combined with automatic password changing to control how often a password is changed (whenever it expires, Secret Server will queue the secret up for a password change).

You can also control which field is used for expiration. This does not have to be the password field—you could use expiration on a license key and set expiration to when the license is going to expire. When a secret expires, you can then update the expiration field (say license key) and it will no longer be expired. This is a generic way to ensure that a specific field on a secret is changed on a regular basis.

Session Launcher

The Launcher can be configured on the secret template to allow any tool to be launched using the secret such as Remote Desktop, PuTTY, Web launcher or a custom launcher you configure for a particular executable file, for example, MS SQL Management Studio, SSH clients, FTP tools and more. This can also be used with the "hide launcher password" setting to allow administrators to launch tools without revealing the password.

It is worth spending time in the beginning to get your secret templates the way you want them before users start adding data. Therefore, when a user goes to create a new secret it will be clear which secret template to use instead of selecting the wrong one and attempting to fit account information into an unsuitable template. You can use an option on the secret called "convert template" to later convert a secret to another template, but it is much simpler to plan before your organization begins adding data.

Basic Configuration

When creating new secret templates, make sure you configure Remote Password Changing, password requirements, secret expiration and the launcher. Ensure your secret template names are descriptive and use terms your users understand. For instance, if you have one template that expires and one that does not, make sure it is clear from the name. If your organization does not use the term AD account, change it to match the organization's language.

Deactivate Unused or Retired Templates

Secret Server comes with many secret templates preconfigured. You should decide which you want to use and then deactivate the others. You can also retire secret templates if your requirements change over time—secrets remain when a secret template is deactivated but no one will be able to create new secrets for that secret template. Secret Server SS uses soft deletes rather than hard deletes (data is marked as inactive rather than actually deleted), which is essential for auditing. Secrets and secret templates can be inactivated but not deleted.

Limit Secret Template Administrators

Changing secret templates should be limited to only a small subset of your Secret Server admins. Create a separate role that has the "administer secret templates" role permission and remove it from administrator if you have a lot of administrators. Once you have secret templates configured, it is unlikely they will need to be changed frequently so very few people should need access.

Override Settings at the Secret

Many of the settings at the secret template can also be overridden at a secret based on that template. For example, if you create a secret for your AD service accounts with a 30-day expiration but need 90 days for a specific AD service account, you can set it to 90 days for that one secret. This gives some flexibility for secrets that need to behave differently than other secrets using the same secret template.

Event subscriptions are a great way to send alerts based on various activities. One of the most common event subscriptions we recommend is alerts based on "unlimited administrator" mode being turned on. This can be aligned to alert your CISO or a manager within your information security team, as this should happen only rarely. Other useful event subscriptions to consider:

- Backup Configuration – Backup Failure
- Configuration – Edit
- Encryption – Key Management Disabled
- Role
- Role Permission
- Site – Engine Offline

These alerts can be sent to different people or can even be sent to users that do not have a Secret Server account. For some of the suggestions above, you may have some of these alerts go to a manager of the systems administration team rather than the information security team. Sending alerts to team members that are responsible for different portions of the application allows for flexibility in who may have to respond to these events.

There are many useful built-in reports. For example, a license audit report may be useful to an auditor, while the built-in reports for secret policies may be useful for information security. It is a good idea to meet with various teams to determine what reporting requirements they may have. Please review our [event subscription](#) and [reporting](#) documentation for more information.

The [Audit Data Retention](#) documentation may be helpful for very large environments where there is audit and log retention flexibility. For example, some of these tables within the database are sizeable, so if your environment exceeds 50,000+ secrets, it may be a good idea to make some adjustments. Similarly, you could have a smaller environment, say 25,000+ secrets, but if you are using all Secret Server features heavily, adjustment might be helpful.

Deleting the data within here should not substitute for a database maintenance plan and should be only considered complimentary to one. Lastly, individual audit tables, such as the "secret audit", cannot be managed independently outside of this configuration unless adjusting those tables directly within the database. Thus, we do not recommend doing unless under direct instruction of our support or professional services teams.

APIs and built-in extensibility features are one of the best ways to improve the automation and flexibility of Secret Server and address novel use cases. PAM Maturity relies on the consistent handling of credentials across your organization. Using the API is a great way to enforce consistency, without additional administrative overhead.

Features like event pipelines can help to organize secrets and users, as well as tweak some settings that are not enforced via policy. Additionally, you can discover and manage new device types using extensible discovery or custom password changers. This section covers some high-level best practices for these features and how best to employ them in your environment.

Running PowerShell with Secret Server

Secret Server's extensible features almost all use PowerShell from the Web servers or distributed engines to execute code. Create a service account for these tasks and store it in a secure location with the other Secret Server service accounts. Documentation for our [API and Scripting](#) components can be found in our Knowledge Base.

PowerShell Runspaces

Runspaces are instances of the PowerShell engine within a process. They define the context a PowerShell runs in and preserve session state using variables and functions that you define, including modules that you load.

For scripts executing on the local or default site, Secret Server generates a runspace to localhost on the Web server (Web site processing) or distributed engine (engine processing or SSC). For scripts running against a specified site, any engine associated to that site may run the script and accomplish the work. This means any dependent configurations need to be made across all Web servers or engines within a site.

This runspace is generated using a specified secret credential in all cases. The credentials used are critical in the operations of PowerShell in Secret Server and a least-permissioned approach should be used.

A domain account, that is, a member of the "remote management users" or "local admin" groups, on the engines or Web servers will have enough permissions to generate the runspace Secret Server needs to execute code. Further restrictions to deny interactive log on can be applied; however, it is important to know what account to use for a least-permissions approach.

You can simplify the assignment of the privileged account in individual locations in Secret Server by specifying a "site run as" secret. This is a default secret for PowerShell running that prevents you from having to manually associate the secret in each place it is used. We highly recommend to configuring this setting on each site under the site's settings.

CredSSP

CredSSP is an authentication mechanism that is designed to delegate credentials across multiple sessions in PowerShell. By default, the runspace uses CredSSP authentication. This is not ideal as there are some security concerns with CredSSP. It is generally better to disable CredSSP authentication under all used sites, and instead implicitly pass credentials into the scripts. CredSSP does have dependent configurations that are required before it can be used.

However, if CredSSP is required, be very explicit with your delegate computer list. Allowlisting this option with an asterisk (*) may seem easy, but it allows an attacker to scale horizontally across your network.

API

The API is a powerful tool for improving automation and flexibility of your Secret Server deployment. Automation tools can pull dynamic credentials from the vault or leverage the API to create new secrets as part of an automation pipeline.

If your organization does not have script experts to improve your PAM program's flexibility, Delinea Professional Services can help to scope and create custom integrations to address your use cases. Contact your account manager or customer success manager to engage professional services.

API Authentication

Carefully consider your authentication strategy for the API when automating workflows. Authentication strategies that do not require hard coding a password into a configuration or script file are preferred. Secret Server Provides a few mechanisms to accomplish this:

Software Development Kit

The Secret Server Software Development Kit (SDK) is a mechanism to authenticate machines to a Secret Server instance, without having to pass implicit credentials into the system for each set of calls. To ensure proper RBAC, the SDK is associated to a specific API user in Secret Server. On the local system, once the SDK is initialized, it leverages DPAPI to encrypt the config files, tying it to the user who initialized it. You can use multiple SDK instances on the same machine; however, each instance needs its own directory.

Integrated Windows Authentication

You can enable Integrated Windows Authentication (IWA) in IIS to use native Windows authentication for API user authentication. This allows

the domain to supply authentication between the Web server and Secret Server. This is a great option to remove credentials and automate authentication for automation processes in your environment.

Note: IWA is only available for Secret Server On-Premises.

Event Pipelines

[Event pipelines](#) are a powerful feature that configures tasks that run based on triggers and filters. If scripting is not an option for you, pipelines can be a script-free way to accomplish a lot of what the API provides. You can use event pipelines to build the core automation of a mature PAM deployment.

Developer Resources

This topic is a one-stop resource for Secret Server developers. It points to TDP topics, as well as legacy knowledgebase articles. See the main TDP [API and Scripting](#) section too.

- [Custom Reports Gallery](#)
- [Creating Custom Reports](#)
- [Using Dynamic Parameters in Reports](#)

- [Configure CredSSP for use with PowerShell](#)
- [Creating and Using PowerShell Scripts](#)
- [Creating and Using SSH Scripts](#)
- [Creating and Using SQL Scripts](#)
- [Password Changing Scripts](#)
- [Searching Secret Server - PowerShell](#) (KBA)
- [Using Secret Fields in Scripts](#)
- [Using Webservices with IWA via PowerShell](#)

- [REST API PowerShell Script Examples](#)
- [REST Web Services API Reference and Download](#)

- [Change SQL Service Account Passwords without Restarting SQL Service](#)
- [Dependency Token List](#)
- [Understanding Dependency Script Errors for SSH, Powershell and SQL](#)

- [Downloads for Secret Server Software Development Kit for DevOps](#)
- [Secret Server Software Development Kit for DevOps](#)
- [Delinea Community GitHub Repository](#)

- [SOAP Web Services API Guide](#) (PDF)
- [SOAP API PowerShell Script Examples](#)

Directory Services

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Directory services are components of network operating systems that map the names of network resources to their network addresses. Their shared information infrastructure locates, manages, and organizes network resources, which can include volumes, folders, files, users, groups, devices, and much more. Active Directory is Secret Server's native directory service.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server can integrate with Active Directory by allowing users to use their Active Directory credentials to log on Secret Server.

Note: Before synchronizing or creating users, you need to create a secret to be used as the "sync secret." This secret should contain Domain Admin credentials (or an account with appropriate permissions to search and view the attributes to all your organization's users and groups).

Active Directory Rights for Synchronization Account

Below is a listing of the Active Directory permissions required by the account used for synchronization. See [Configuring Active Directory](#) for more on selecting this account.

Recommended Permissions

Object Tab

This object and all descendant objects:

- List contents
- Read all properties

Minimum Required Permissions

Note: These all require ADSI Edit - Allow (Active Directory Service Interfaces Editor) permission.

Object Tab

This object and all descendant objects:

- List contents

Properties Tab

This object and all descendant objects:

- Read objectClass

Descendant User objects:

- Read Display Name
- Read Distinguished Name
- Read E-mail-Address
- Read objectGUID
- Read Logon Name
- Read Logon Name (pre-Windows 2000)

Descendant Group objects:

- Read displayName
- Read Distinguished Name
- Read Group name (pre-Windows 2000)
- Read groupAttributes
- Read memberOf
- Read Members
- Read objectGUID

Active Directory Credential Caching

Overview

Active Directory credential caching enables users to access Secret Server even when the domain controller is unavailable. When caching is enabled, Active Directory credentials are cached for 30 days in the on-premise editions, and for 90 days in Secret Server Cloud.

With credential caching enabled, whenever a domain user successfully logs into Secret Server, their domain password is hashed using PBKDF2, and stored in the Secret Server database along with the current time stamp.


If a domain user attempts to log in but Secret Server is unable to contact a domain controller, it falls back to the cached credentials to attempt to provide access. If the hash of the entered password matches the hash of the cached credentials and the time has not expired, the authentication will be successful.

AD Caching Configuration

AD credential caching is disabled by default, but an administrator can enable or disable it at any time using the steps below:

1. Click **Admin > Configuration** and click the **Login** tab.
2. Scroll to the bottom of the window and click the **Edit** button. The tab becomes editable:

Cache AD Credentials for when Engines are offline

 Clear Cached AD Credentials

3. To enable caching, click to select the **Cache AD Credentials for when Engines are offline** check box.
4. To disable caching, click the **Clear Cached AD Credentials** button.

Auditing

Audit logs are recorded in the system log whenever cached credentials are found to be expired or when a successful login attempt has been made using cached credentials.

ADFS Custom Rules

Overview

In Active Directory, when a user's sAMAccountName and userPrincipalName (UPN) differ, you must take some steps to accommodate those differences in Secret Server. For example, suppose a user's sAMAccountName is jsmith and the user's userPrincipalName is john.smith@somedomain.com. When Secret Server syncs with Active Directory, it obtains jsmith as the Secret Server login user name. However, with its standard ADFS rule passing in the UPN, Secret Server will receive john.smith@somedomain.com and it will not find the user.

To rectify this situation you must configure the SAML Username Attribute in Secret Server to be customvalue, and use three custom claim rules described below.

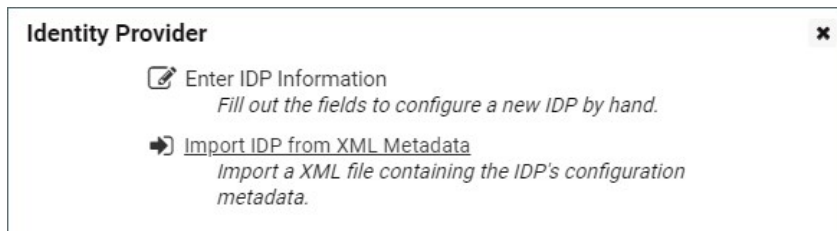
Change the SAML Username Attribute

To change the SAML Username Attribute in Secret Server, perform the following steps:

1. Click **Admin > Configuration**.
2. Click the **SAML** tab and scroll to the bottom of the window.
3. Click **Create New Identity Provider**.



4. In the Identity Provider dialog, click **Enter IDP Information**.



5. In the next Identity Provider dialog under **User Matching**, type customvalue in the box next to **Username Attribute** and click **OK**.

Identity Provider ✕

REQUIRED SETTINGS

Display Name ? *

Name ? *

Description ?

Active ?

Public Certificate ? *

Force Authentication ?

Single SignOn Service Binding ? ▼

Single SignOn Service URL ? *

USER MATCHING

Username Attribute ?

Domain Attribute ?

SINGLE LOGOUT

Enabled ?

Create Three Rules

To create the three rules you need, open the Active Directory application and follow these steps:

1. In the **Edit Claim Rules** window, select **Add Rule**.
2. Choose **Send Claims Using a Custom Rule** as the rule template.
3. Create each rule using the information below, in the order presented.

Note: If you copy code directly from the webpage for pasting, please ensure that you have copied everything you need, or correct the text after pasting it.

Rule 1: Query AD for UPN and sAMAccountname Attributes

```
c:[Type == http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname, Issuer == "AD AUTHORITY"]
=> add(store = "Active Directory", types = ("ssupn", "sswindowsaccountname"), query = ";userPrincipalName,sAMAccountName:{0}", param = c.Value);
```

Rule 2: Obtain the Domain from the UPN

```
c:[Type == "ssupn"]
=> add(Type = "ssnewupn", Value = RegExReplace(c.Value, "^(.*?)@", ""));
```

Rule 3: Combine the sAMAccountname with the Domain


```
c1:[Type == "ssnewupn"]  
&& c2:[Type == "sswindowsaccountname"]  
=> issue(Type = "customvalue", Value = c2.Value + "@" + c1.Value);
```

Configuration Parameters

Active Directory configuration can be enabled by a user with the Administer Active Directory role. To change these settings, select **Active Directory** from the **Administration** menu and then click **Edit**.

The configuration screen offers several options:

- **Enable Active Directory Integration:** Enable or disable the Active Directory Integration feature.
- **Enable Integrated Windows Authentication:** Enable or disable the Windows integrated authentication feature.
- **Enable Synchronization of Active Directory:** Enable or disable the automatic synchronization of the selected Synchronization Groups from Active Directory. If you have manually added users and will not use the Synchronization group, do not enable this setting or manual users can be locked out.
- **Synchronization Interval for Active Directory:** Set the interval that Secret Server synchronizes its users and groups with the Active Directory.
- User Account Options:
 - **Users are enabled by default (Manual):** Secret Server users are automatically be enabled when they are synced as new users from Active Directory. If they were disabled explicitly in Secret Server, they are not be automatically re-enabled. If creating a new user causes the user count to exceed your license limit, the user is created as disabled. Secret Server
 - **Users are disabled by default (Manual):** Secret Server users are automatically disabled when they are pulled in as new users from Active Directory. If they were enabled explicitly in SS, they are not automatically re-disabled.
 - **User status mirrors Active Directory (Automatic):** When a new user is pulled in from Active Directory, they are automatically enabled if active on the domain. The exception is when this causes you to exceed your license count. For existing users, they are automatically be disabled if they are removed from all synchronization groups, deleted in AD, or disabled in AD. They are automatically re-enabled when they are part of a synchronization group and are active in AD.

Configuring Active Directory

To allow users to log in with their Active Directory (AD) credentials, you can configure your AD domain settings in Secret Server and then add users either individually or by group.

Step 1: Enabling Active Directory Integration

1. Select **Admin > Directory Services**. The Directory Services page appears.
2. Click the **Configuration** tab.
3. If **Enable Directory Services** says **No**:
 1. Click the **Edit** link next to **Directory Services**.
 2. Click the **Enable Directory Services** check box.
 3. Click the **Save** button.

Step 2: Adding a Domain

1. Select **Admin > Directory Services**. The Domains tab of the Directory Services page appears.
2. Click the **Add Domain** button and select **Active Directory Domain**. The Active Directory popup appears.
3. Type the FQDN and friendly name in their text boxes.
4. If you wish to use Secure LDAP, enable the **Use LDAPS** checkbox.
5. If you have an existing sync secret, click the **No Secret Selected** link next to **Synchronization Secret** to select the secret containing the username and password for connection. Otherwise, click the **Create New Secret** link to create one.
6. Click the **Site** dropdown list to select the site for the AD location.
7. Click the **Multifactor Authentication** dropdown list to select the desired MFA, if any.
8. Click the **Validate & Save** button.

Now you are ready to add individual users or groups of users for access to Secret Server with AD credentials. See the relevant section below for instructions.

Step 3: Setting Up Synchronization Groups

Once you add a domain, add the applicable synchronization groups users from these groups are available for login, subject to synchronization settings. These groups can also be used to assign permissions, roles, and sharing. To add groups:

Note: If the specific group does not exist, one can be created by your Active Directory administrator. If you create domain users manually or want to convert local users to domain users, see the corresponding sections below before setting the synchronization group.

1. Click the **Domain Name** link on the **Domains** tab. The page for that domain appears.
2. Click the **Groups** tab.
3. Click the **Edit** link next to **Synchronized Groups**. The top Synchronized Group table and search box represent the available sync groups. The Select Groups list shows all the groups belonging to the domain (those with selected check boxes).
4. Type the name of the desired group in the **Search domain for groups** search text box or scroll down the list to find a group.
5. Click the group. The users belonging to the group appear below in a list to provide a preview to aid selecting groups.

Step 4: Adding or Removing Groups

1. Click the check box in the **Select Groups** list to add or remove groups from the domain.
2. If you want to remove an entire group instead of not selecting it (leaving it for later use), click the **Remove** link next to the group name in the **Synchronized Groups** table.
3. Click the **Save** button.

Note: Enabled users count towards your Secret Server user licensing.

Step 5: Enabling Active Directory Synchronization

1. Return to the **Directory Services** page.
2. Click the **Configuration** tab.
3. Click to select the **Enable User Synchronization** check box. Additional settings appear.
4. Choose how often you want Secret Server to sync with AD by configuring the **Synchronization Interval**. The default value is one day.
5. Click the **User Account Options** Dropdown list to select a default status for users. See below for a description of each option. We recommend selecting **Users are disabled by default (Manual)** for initial testing. The options are:
 - **Users are enabled by default (Manual)**: Secret Server users are automatically enabled when they are synced as new users from AD. If they were disabled explicitly in Secret Server, they are not automatically re-enabled. If creating a new user will cause the user count to exceed your license limit, the user created disabled. Secret Server
 - **Users are disabled by default (Manual)**: Secret Server users are automatically disabled when they are pulled in as new users from AD. If they were enabled explicitly in SS, they are not automatically re-disabled.
 - **User status mirrors Active Directory (Automatic)**: When new users are pulled in from AD, they are automatically enabled if active on the domain. The exception is when this will cause you to exceed your license count. For existing users, they are automatically be disabled if they are removed from all synchronization groups, deleted in AD, or disabled in AD. They are automatically re-enabled when they are part of a synchronization group and are active in AD. See [Understanding Active Directory Automatic User Management](#).
6. Change the **Days to Keep Operational Logs** text box to set the period to keep AD-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).
7. Click the **Save** button.

Step 6: Running Active Directory Synchronization

From the **Directory Services** page, click the **Sync Now** button to run a sync. As the sync progresses, you can click the **Refresh** button to monitor the logs until you see the message **Completed Domain synchronization for all domains**.

Converting Local Users to Domain Users

Local users can be converted to a domain user in a one-way irreversible process. This feature helps existing customers with extensive groups and permissions setup for a local user that they want to convert to an Active Directory user. The page can be accessed on the **Administration > Users** page by clicking the **Migrate to AD** button. For the conversion to work, the domain user must not exist within Secret Server. The username is changed to match the domain user throughout the system.

Creating Active Directory Users

Active Directory users can be created manually by a user that has the Administer Users role. You can do this by going to **Administration > Users**, then clicking the **Create New** button. See [Creating a User](#).

Enabling and Disabling Active Directory Users

If you selected a manual setting for **User Account Options**, you can now enable or disable your AD users' access to Secret Server:

1. Go to **Admin > Users**. The Users page appears.
2. To enable users:
 1. Click to select the **Show Inactive Users** check box.
 2. Click to select the check box next to the users to enable.
 3. Click The **Bulk Operation** dropdown list and select **Enable Users**.
3. To disable users, use the same process, selecting **Disable Users** from the **Bulk Operation** dropdown list.

Setting up SAML SSO for Active Directory

How to set up Single Sign-On (SSO) for users synced between an Active Directory domain server and a Secret Server user list.

Note: The interface and workflows for Active Directory Federation Services (ADFS) Server are subject to change. For more current workflow and interface references, please refer to the [Microsoft ADFS Server documentation](#).

ADFS Server

1. On your ADFS server, browse to your Secret Server instance and sign in.
2. Download the SecretServerSAMLMetadata.xml file from [YourSecretServerInstance.Name]/samlmetadata.
3. Open **Active Directory Federation Services Management**.
4. Under **Trust Relationships**, click **Add Relying Party Trust** to add your service provider information.
5. In the **Add Relying Party Trust** wizard, click **Start**.
6. Click **Import data about the relying party from a file**.
7. Browse to select the metadata XML file you downloaded earlier and click **Next**.
8. Enter a display name of your choice and click **Next**.
9. Decide whether to configure multi-factor authentication and click **Next**.
10. Choose **Issuance Authorization Rules** and click **Next**.
11. On the **Ready to Add Trust** page, make sure the box next to **Open the Edit Claim Rules dialog** is checked.
12. Click **Next** and then click **Close**. The **Edit Claim Rules for Secret Server** window should open automatically.
13. Click **Add Rule**.
14. Select **Send LDAP Attributes as Claims** as the Claim Rule Template and click **Next**.
15. Fill out the fields below as indicated:
 - o **Claim rule name:** Optional Name
 - o **Attribute Store:** Active Directory
 - o Add an **LDAP Attribute** of User-Principal-Name with an **Outgoing Claim Type** of Name ID.
16. Click **Finish**.
17. Click **Apply** and then click **OK**.
18. Run the following PowerShell command:

```
Set-ADFSRelyingPartyTrust -TargetName name of the relying party trust created in adfs -SamlResponseSignature "MessageAndAssertion"
```
19. Download the metadata for your ADFS IDP from [https://\[YOURSERVERNAME\]/FederationMetadata/2007-06/FederationMetadata.xml](https://[YOURSERVERNAME]/FederationMetadata/2007-06/FederationMetadata.xml).

Secret Server

1. In Secret Server, click **Admin > Configuration > SAML** tab.

- On the **SAML Configuration** tab, click **Create New Identity Provider**.

SAML Configuration


General Login **SAML** Folders Local User Passwords Security Ticket System

i SAML instructs Secret Server to trust a separate server as its Identity Provider. When SAML is enabled, the Identity Provider is responsible for asking the user for their username or password, but Secret Server will still ask the user for any configured 2-factor.

SAML GENERAL SETTINGS

SAML Enabled



Use Legacy SAML

 **Edit**


SAML SERVICE PROVIDER SETTINGS

Name SecretServerServiceProvider



Certificate

 **Edit**  **Download Service Provider Metadata (XML)**

IDENTITY PROVIDERS

 **Create New Identity Provider**

DISPLAY NAME	NAME	DESCRIPTION	CERTIFICATE THUMBPRINT	ACTIVE	ADVANCED

 **View Log**  **View Audit**

- Click **Import IDP from XML Metadata** and select the ADFS metadata you downloaded. If you don't see the file, you might need to change the metadata filetype to xml.

Identity Provider ✕

-  **Enter IDP Information**
Fill out the fields to configure a new IDP by hand.
-  **Import IDP from XML Metadata**
Import a XML file containing the IDP's configuration metadata.

Adding Users to ADFS

For users to be authenticated by the SSO workflow you are setting up, Secret Server usernames must match domain AD usernames. If you

manually add usernames to Secret Server or AD, you must inspect them carefully to ensure that they match. You can also use Secret Server Discovery to sync Secret Server usernames in bulk with AD usernames.

Once a username matches in both systems, the user can log into their desktop computer using their AD credentials and then browse to Secret Server without being prompted again for authentication.

Note: If you have accounts in which the sAMAccountName differs from the UPN name, you can create custom rules to accommodate the differences. See the Directory Services section of the Secret Server documentation.

Common Errors

If you encounter any of the errors below, check that the **RelyingPartyTrust Rule** on the ADFS server has both the message and assertion signed. By default, only the assertion is signed.

- "Attempt to login via SAML from identity provider had no signed responses or assertions"
- "Attempt to login via SAML with unsigned request"
- "Attempt to login via SAML with unsigned assertion"

If you encounter the error, "SAML Response signature message from IDP failed verification," it means that Secret Server cannot decrypt the assertion message from the IDP (ADFS) because the public certificate thumbprint is incorrect. To fix this issue, follow the steps below.

1. Download the ADFS certificate, upload it to Secret Server (**Admin > Configuration > SAML** tab) and edit the IDP configuration.
2. Check the token-decrypting in ADFS to verify the certificate.
3. Use the [Get-AdfsCertificate](#) cmdlet to retrieve the certificates listed below that ADFS uses, and check that they are appropriately identified as primary (**IsPrimary** is set to **True**):
 - A primary token-signing certificate is used to digitally sign outgoing claims.
 - A primary token-encrypting certificate is published in federation metadata for use by trusted claims providers.
 - Information card signing and service communications certificates are always primary.

Syncing and Authenticating AD Users via a Distributed Engine

Local Versus Distributed Engine Sites

Secret Server connects to the domain: from the Web server *or* routed through a distributed engine. If your Web server can reach your domain without issue, then using the local site option is recommended. When a user authenticates or AD synchronization is run, the connection to the domain is from the Web server. If your Web server cannot connect to the target domain, if it is a VM in a cloud environment for example, you can setup an engine on-premises and assign it to the domain. When a user authenticates, Secret Server routes the domain calls through the on-premises engine, eliminating the need for site to site connections or persistent VPNs. Review the Distributed Engine guide for steps on setting up sites and engines.

Note: The Active Directory secret is used to synchronize users and groups, it requires permission to search and view the attributes of the users and groups. If you plan on using discovery, the account also needs permissions to scan computers on the network for accounts.

To setup AD to sync from a DE:

1. Create a synced secret. Before synchronizing or creating users, create a secret for use as the sync secret. This secret should contain Domain Admin credentials (or an account with appropriate permissions for read access to all your organization's AD objects).
2. Specify the domain to authenticate against:
 1. Before synchronizing or creating users, you must first specify which domains Secret Server can authenticate against. Secret Server can synchronize with any number of domains.
 2. Go to **Admin > Active Directory**. The Active Directory Configuration page appears.
 3. Click the **Edit Domains** button.
 4. Click the **Create New** button. The Active Directory Domain page appears.
 5. Type the domain information that you want to authenticate to.
 6. Click the **Link a Secret** selection button.
 7. Click the **Sync Secret** list to select the AD secret you created earlier.

Note: If you do not have a secret setup yet, click the **Create New Secret** link to create your AD secret.

Note: The AD sync secret is used to synchronize users and groups. It requires permission to search and view the attributes of the users and groups. If you plan on using Secret Server discovery, the account will also need permissions to scan computers on the network for accounts.

8. Click the **Save and Validate** button.
3. Set up the synchronization groups:
 1. Once the domain has been added, go to **Admin > Active Directory**. The Active Directory Configuration page appears.
 2. Click the **Edit Synchronization** button. The Synchronization Edit page appears. The Available Groups represent all accessible groups on the specified AD domain. You can preview the user membership with the Group Preview control.
4. Select the desired group from the Available Groups that contains the AD accounts for users you would like to create in Secret Server.
5. Configure AD:

Note: See [Active Directory Configuration Parameters](#) for more information.

1. Go to **Admin > Active Directory**. The Active Directory Configuration page appears.
2. Click on the **Edit** button. The Edit Active Directory Configuration page appears.
3. Click to select the **Enable Active Directory Integration** check box.
4. Click to select the **Enable Synchronization of Active Directory** check box.
5. Click the **Save** button.
6. Turn on AD sync.

Understanding Active Directory Automatic User Management

Overview

When Active Directory (AD) Sync is run with the "User status mirrors Active Directory (Automatic)" option, it creates groups and users in Secret Server to mirror the organization's configured AD groups and users. A Secret Server user is created or enabled for every enabled AD user in the selected groups.

Thus, every enabled AD user in every synced group consumes a Secret Server license, whether or not they use Secret Server. As a result, an organization can end up paying for far more Secret Server licenses than they need.

AD Automatic User Management addresses this issue by automatically disabling the accounts of users who have not logged in to Secret Server in a specified number of months. This saves unnecessary licensing costs as inactive users do not count against the number of user licenses required by Secret Server.

You can configure the setting on the Edit Active Directory Configuration page. See [Configuring Active Directory](#). There is a checkbox to enable or disable the feature and a textbox to set the number of months before a user is auto-disabled. The default is three, but you can set it from one to 12.

Newly-added users remain enabled until the first synchronization after the configured number of months have passed. When a user whose account has been disabled by this feature attempts to log in they automatically have their account enabled, provided there are licenses available.

Examples

Example One

1. Maria joined the company today.
2. The next AD synchronization creates a Secret Server account for Maria.
3. Maria never logs in to Secret Server because she does not need it for her job.
4. Once the defined number of months have passed, the next AD synchronization disables Maria's Secret Server account.
5. The Secret Server license used by Maria's account becomes available for use.

Example Two

Note: This only pertains to users who have never logged into Secret Server and their account was disabled (never enabled). It does not apply to previously enabled users who have been disabled due to inactivity.

1. Joe gets added to Secret Server but never logs in.
2. The defined number of months later, Automatic User Management disables his account, freeing his license.
3. Joe gets promoted to a job that requires Secret Server.
4. Joe logs into Secret Server.
5. His account is automatically re-enabled, and he now takes up a license.
6. Joe gets demoted to his old job, which does not require Secret Server.
7. A defined number of months later, Automatic User Management disables his account, and the license is freed up once again.
8. Joe has no idea any of this has happened—the automated process is hidden from him.

Example Three

1. Rupert logs in to Secret Server several times per month.
2. The defined number of months for Automatic User Management to disable his account is never reached.
3. Rupert's account stays current and his license remains his. The entire process is invisible to Rupert.

Important: This integration requires .NET Framework version 4.8 or later.

Use these steps to integrate Secret Server with an Azure Active Directory tenant.

Configuration Parameters

Azure Active Directory (*Azure AD*) configuration can be enabled by a user with the Administer Active Directory role. To change these settings, navigate to **Admin | Directory Services**, click the **Domain Name** associated with your Azure AD directory, and then click **Edit**.

When creating a new directory, the required configuration screen settings have the following fields:

- **Domain Name:** A friendly display name for the Azure Directory.
- **Active:** Enable or disable the Azure Active Directory domain integration.
- **Tenant ID:** Globally unique identifier (GUID) value assigned to the Azure AD directory.
- **Client ID:** Globally unique identifier (GUID) value assigned to the Client Secret upon creation. Portal will also reference this as the *Application ID* or *App ID*.
- **Client Secret:** Unique, generated string for the Client Secret. *This value can only be retrieved upon creation.*

Optionally you can also configure the following:

- **Multifactor Authentication:** Drop-down selection for the desired MFA.

Setting Up Azure AD for SAML

Configuration Steps

1. Log into your portal.azure.com account.
2. Navigate to **Azure Active Directory**.
3. Navigate to **Enterprise Applications**.
4. Select **New Application**.
5. Select **Non-gallery application**.
6. Give your new IdP application a name and click **Add**.
7. Click **Single sign-on**.
8. In the dropdown, select **SAML-based Sign-on**.
9. If you haven't done so already, download the Secret Server metadata file named SecretServerSAMLMetadata.xml file from [YourSecretServerInstance.Name]/samlmetadata.
10. Click **Upload metadata file** and upload the Secret Server Metadata file you previously.
11. Click **Save**.
12. Scroll down and click **Metadata XML** to download the metadata for this application.
13. Go back to **Azure Active Directory** and click on **App registrations**.
14. Select your Azure Identity Provider (IdP) application.

If you don't see the application immediately, you might need to click **View all Applications**.
15. Click **Settings > Properties**, then enter the Logout URL field for your instance. The form for this URL will be: [https://\[YourSecretServerInstanceName\]/saml/SLOService.aspx](https://[YourSecretServerInstanceName]/saml/SLOService.aspx).
16. Click **Save**.
17. Navigate to your Secret Server instance and to the **Admin > Configuration > SAML** tab.
18. Click **Create New Identity Provider**.

SAML Configuration

General Login **SAML** Folders Local User Passwords Security Ticket System

SAML GENERAL SETTINGS

SAML Enabled

Use Legacy SAML

[Edit](#)

SAML SERVICE PROVIDER SETTINGS

Name SecretServerServiceProvider

Certificate

[Edit](#) [Download Service Provider Metadata \(XML\)](#)

IDENTITY PROVIDERS

[+ Create New Identity Provider](#)

DISPLAY NAME	NAME	DESCRIPTION	CERTIFICATE THUMBPRINT	ACTIVE	ADVANCED
View Log View Audit					

19. Click **Import IdP from XML Metadata**.

Identity Provider ✕

[Enter IDP Information](#)
Fill out the fields to configure a new IDP by hand.

[Import IDP from XML Metadata](#)
Import a XML file containing the IDP's configuration metadata.

20. Select the Azure AD metadata you saved/downloaded previously to import it.

If you don't see the file immediately, make sure the file type specified for your search is .xml.

Adding Users to Single Sign-On in Azure AD

For users to be authenticated by the SSO workflow you are setting up, Secret Server usernames must match Azure AD usernames. If you manually add usernames to Secret Server or Azure AD, you must inspect them carefully to ensure that they match. You can also use Secret Server Discovery to sync Secret Server usernames in bulk with Azure AD usernames.

1. Log into your portal.azure.com account.

Navigate to **Azure Active Directory > Enterprise Applications** and select your IdP from the list

2. Select **Users and groups** and **Add User**.
3. Click **Users and groups/None Selected**.
4. Search for the user you want to add to your SAML workflow. (Note that any users added must also exist in your Secret Server instance. Usernames must match between the systems).
5. Click **Select** at the bottom, then **Assign**.

This user should now be able to use the Single Sign-On workflow. To test this, log into Azure AD as the user, then browse to your Secret Server instance. The user should be logged into Secret Server automatically without being prompted again for login credentials.

Advanced Settings

The following Secret Server Identity Provider Advanced Settings can be configured in Azure AD.

1. Log in to portal.azure.com.
2. Navigate to **Azure Active Directory > Enterprise Applications**.
3. Select your IdP, then click **Single sign-on**.
4. Scroll down and check the box for **Show advanced certificate signing settings** checkbox.
5. Click the drop-down arrows to reveal options.

These advanced options correspond with advanced options in *Secret Server **Admin > Configuration > SAML** tab.

6. Click **Advanced Settings** next to your identity provider.

Require Signed SAML Response/

Require Signed Assertion/

Require Signed Assertion Or Signed SAML Response

Setting Up SAML SSO for Azure Active Directory

Azure AD Configuration

To set up SAML-based single sign-on for Secret Server in Azure Active Directory, see [Quickstart: Set up SAML-based SSO for an application in your Azure AD tenant](#) provided by Microsoft.

- You will need to download the SecretServerSAMLMetadata.xml file from [YourSecretServerInstance.Name]/samlmetadata.
- You will need to use https://[YourSecretServerInstanceName]/saml/SLOService.aspx for the logout URL.

Secret Server Configuration

After you have set up SAML SSO for Secret Server in Azure AD, open Secret Server and make the configurations specified below.

1. In Secret Server, click **Admin > Configuration > SAML** tab.
2. On the **SAML** tab, click **Create New Identity Provider**.

SAML Configuration

General Login **SAML** Folders Local User Passwords Security Ticket System

SAML GENERAL SETTINGS

SAML Enabled

Use Legacy SAML

[Edit](#)

SAML SERVICE PROVIDER SETTINGS

Name SecretServerServiceProvider

Certificate

[Edit](#) [Download Service Provider Metadata \(XML\)](#)

IDENTITY PROVIDERS

[+ Create New Identity Provider](#)

DISPLAY NAME	NAME	DESCRIPTION	CERTIFICATE THUMBPRINT	ACTIVE	ADVANCED
View Log View Audit					

3. Click **Import IDP from XML Metadata** and select the SecretServerSAMLMetadata.xml file you downloaded.

Identity Provider ✕

[Enter IDP Information](#)
Fill out the fields to configure a new IDP by hand.

[Import IDP from XML Metadata](#)
Import a XML file containing the IDP's configuration metadata.

Syncing Usernames in Azure AD and Secret Server

For users to be authenticated by the SSO workflow you are setting up, Secret Server usernames must match Azure AD usernames. If you manually add usernames to Secret Server or Azure AD, you must inspect them carefully to ensure that they match. You can also use Secret Server discovery to sync Secret Server usernames in bulk with Azure AD usernames.

Once a username matches in both systems, the user can log into their desktop computer using their Azure AD credentials and then browse to Secret Server without being prompted again for authentication.

Note: If you have accounts in which the sAMAccountName differs from the UPN name, you can create custom rules to accommodate the differences. See the Directory Services section of the Secret Server documentation.

Advanced Certificate Signing Settings

If you apply advanced certificate signing settings to the Secret Server IdP application in Azure AD, use the same settings found in Secret Server for the following:

- Require Signed SAML Response
- Require Signed Assertion
- Require Signed Assertion Or Signed SAML Response

You can find these settings in Secret Server by clicking **Admin > Configuration > SAML** tab and clicking **Advanced Settings** next to your identity provider.

Create Azure App Registration

The steps provided can be used to create the App Registration required for configuring Azure Active Directory integration.

Important: This integration requires .NET Framework version 4.8 or later.

Azure Portal Method

Create the Application Registration

1. Log on the Azure Portal
2. If needed, switch to the intended directory.
3. Navigate to the **Azure Active Directory** blade.
4. Click **App registrations** on the left pane in the **Manage** section.
5. Click the **New registration** button. The App Registrations page appears.
6. Type Thycotic Secret Server in the **Name** text box.
7. Click to select the **Accounts in the organizational directory only** selection button to choose single tenant.
8. In the **Redirect URI (optional)** section, click to select **Web** in the dropdown list.
9. Type `https://<Your Secret Server URL>/signin-oidc` in the text box to the right of the list.
10. Click the **Register** button. Once the app registration is created, the Azure portal opens the blade to this object.
11. In the blade for this app registration, take note of the **Application (client) ID** and **Directory (tenant) ID**. These will be needed for Secret Server configuration.

Add Client Secret to the Application Registration

1. Click **Certificates & secrets** on the left panel in the **Manage** section. The Certificates & Secrets page appears.
2. Go to the **Client Secrets** section.
3. Click the **New Client Secret** button. The Add a Client Secret section appears.
4. Type Secret Server in the **Description** text box.
5. Click to select your desired expiration in the **Expires** selection button.

Note: If the client secret is set to expire, Secret Server must be updated upon or before expiration for this integration to function correctly.

6. Click the **Add** button. The client secret appears in the Client Secrets section.
7. Record the text string in the Value column for that secret.

Add API Permissions to the Application Registration

1. Click **API Permissions** on the left panel in the **Manage** section. The API Permissions page appears.
2. If any default permissions appear in the unlabeled configured permissions table, click the ... button and select **Remove Permission**.

3. Click the **Add a Permission** button. The Request API Permissions page appears.
4. Click the **Microsoft Graph** panel button. A wizard begins.
5. Click **Application Permissions** when asked **What type of permissions does your application require?** The **Select Permissions** section appears.
6. In the search text box, type `Group`. A `GroupMember` section appears.
7. Click to expand the section.
8. Click to select the **GroupMember.Read.All** check box.
9. Repeat the process for the following application permissions:
 - o `Group.Read.All`
 - o `GroupMember.Read.All`
 - o `Member.Read.Hidden`
 - o `User.Read.All`

Do the same for the `User.Read` delegated permission.
10. Click the **Add Permissions** button. A prompt appears.
11. Click the **Yes** button to grant consent to all accounts in the directory. You will receive a notification for "grant consent," and a green check mark appears in the Status column on the Configure Permissions page.

Script Method

The script below is provided as-is, and future use may require adjustment if Microsoft changes the AzureAD PowerShell module.

At the time of writing, there is no command in the AzureAD module granting admin consent to the app. That step has to be performed via the Azure Portal.

[Download the script](#)

Configure Azure Active Directory Domain

The steps below are used for adding an Azure Active Directory configuration to Directory Services.

Add Azure Active Directory Domain

1. Navigate to **Admin | Directory Services**.
2. Click the **Add Domain** button.
3. Click the **Azure Active Directory Domain**.
4. Using the values saved from [Creating Azure App Registration](#), paste or type in:
 - o Friendly domain name
 - o Tenant ID
 - o Client ID
 - o Client Secret
5. Ensure the **Active** check box remains checked.
6. (Optional) Click the **Multifactor Authentication** dropdown list to select your desired MFA.
7. Click the **Validate & Save** button. Once validation completes, you will see the Friendly domain name listed.
8. Click the name of the new domain to open the configuration page.
9. Click the **Groups** tab.
10. Click the **Edit** link next to **Synchronized Groups**.
11. Scroll to or search for each desired group containing users you want to sync in the **Select Groups** table.
12. Ensure each group's check box is **checked**.
13. Click the **Save** button to save your changes. You will now see the selected groups in the Synchronized Groups table.
14. Click the **Directory Services** breadcrumb link at the top of the page to navigate back to the Directory Services page.
15. Click the **Sync Now** button to sync the directory groups.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

LDAP is an open, industry-standard application protocol for accessing and maintaining distributed directory information services over IP networks.

Secure LDAP

Overview

By default, Secret Server uses normal LDAP on port 389 to communicate with Active Directory. Although passwords are still transmitted using Kerberos or NTLM, user and group names are transmitted in clear text. In contrast, secure LDAP (LDAPS) requires that both port 389 and 636 are open.

If you want all information to be encrypted, then you can enable Secure LDAP (LDAPS) in Secret Server via the Advanced link on the Edit Domain page.

When LDAPS is used, Secret Server transmits and receives Active Directory data through port 636 (with port 389 open). A certificate on the domain controller is used to negotiate encryption, and no information is transmitted in clear text.

Note: If you want to use Integrated Windows Authentication and Secure LDAP, that is only supported in Windows Server 2008 R2 or greater.

Troubleshooting LDAPS Connection Issues

Common problems with LDAPS and Secret Server:

- When you turn on LDAPS you will get a "domain name is invalid" error.
- Users are suddenly unable to log on Secret Server.

Both issues are caused by LDAPS to Secret Server communication issues, usually one of the following:

- The certificate is expired (this is the client certificate, not the SSL on the Secret Server website).
- LDAPS is not enabled in your environment.
- A port is blocked that is denying successful communication between the server and AD.

To troubleshoot, use the [free LDP tool](#) to test LDAPS connections from the Secret Server Windows server to your AD server. If you are unable to establish a connection on port 636 (with 389 open too), then we recommend consulting with your AD or security team.

Note: Sometimes the Secret Server event viewer has information regarding invalid certificates.

Syncing with OpenLDAP Directory Service

Introduction

OpenLDAP is a free, open source version of the Lightweight Directory Access Protocol (LDAP) developed by the OpenLDAP Project. This topic describes syncing OpenLDAP to Secret Server.

Note: This feature is only supported by the new interface. The classic interface does not support OpenLDAP Directory Services.

Unsupported Use Cases

Anonymous User Authentication

We do not support anonymous user authentication:

When creating an OpenLDAP directory service, "Anonymous" is a supported authentication method. When this is chosen, Secret Server connects anonymously to the OpenLDAP directory service as configured during the synchronization process and creates any users found on the directory service.

When anonymous is selected, a secondary authentication option, "User Authentication," appears, which is the method used when the synchronized users attempt to authenticate to Secret Server. In short, user authentication cannot be anonymous because Secret Server does not allow anonymous access.

The valid options for user authentication when anonymous is selected for the synchronization process are "Basic," "Kerberos," or "No Authentication." "No Authentication" supports using an OpenLDAP directory service as a user directory while enabling alternative methods of authentication, such as SAML.

Duplicate User Attributes

We do not support configurations where using different attributes yield users with the same username, GUID, or user principal name (email address format—not necessarily an actual email address). These must all be unique to each user. If a duplicate exists, it may result in odd, unpredictable behavior from the application.

Procedure

1. Create a secret in Secret Server of type **OpenLDAP Account**. This sync secret is used to synchronize users and groups. It requires permission to search and view the attributes of the users and groups. If you plan on using Secret Server discovery, the account will also need permissions to scan computers on the network for accounts. Complete these parameters:
 - Domain. Example: ldap.omega.thycotic.com
 - Username. Example: cn=ldap,dc=omegaldap,dc=local
 - Password
2. Go to **Admin > Directory Services**. The Directory Services page appears:

Admin > Directory Services

Domains Configuration Logs Audit

Last Sync Finished: 11 minutes ago

Add Domain Sync Now

1 Item Include Inactive

DOMAIN NAME	FRIENDLY NAME ↑	DOMAIN TYPE	ACTIVE	LAST RUN RESULT
gamma.thycotic...	Gamma Domain	Active Directory	<input checked="" type="checkbox"/>	Group Membershi

3. Click the **Add Domain** dropdown list and select **OpenLDAP Domain**. The OpenLDAP popup appears:

Open Ldap

Fully Qualified Domain Name *

Friendly Name *

Active

Distinguished Name *

Authentication * Kerberos

Use LDAPS

Synchronization Secret * No Secret Selected [Create New Secret](#)

Site * Local

Multifactor Authentication < None >

Cancel Validate & Save

4. Type the domain's FQDN in the **Fully Qualified Domain Name** text box. For example: ldap.omega.thycotic.com.

5. Type any name you desire in the **Friendly Name** text box.

6. Ensure the **Active** check box is selected.

7. Type the distinguished name (node path) in the **Distinguished Name** text box. For example: dc=omegaldap,dc=local
8. Click the **Authentication** dropdown list to select either the **Basic** or **Anonymous** authentication method.
 - Basic authentication requires that valid credentials are assigned as the sync secret. Those credentials are used to authenticate to the OpenLDAP system on each sync.
 - Anonymous authentication does not require valid credentials and removes the Synchronization Secret section. Instead, it exposes a User Authentication field.

Note: The Kerberos authentication method probably works but has not been test by Delinea.
9. Basic authentication:
 1. Click the **No Secret Selected** link in the Synchronization Secret section. The Select Secret popup appears.
 2. Navigate to and select the secret you created earlier. The moment you click it, the popup disappears and the secret name appears in the Synchronization Secret section.
10. Anonymous authentication: Click the **User Authentication** list to select **Basic** or **No Authentication**. This sets which authentication method to use when users who are synced anonymously try to authenticate:
 - Basic authentication requires valid OpenLDAP account credentials.
 - No authentication is for when customers want users synced from OpenLDAP but use authentication through another service, such as SAML. We do *not* support anonymous authentication for security reasons.
11. Click to select the **Use LDAPS** check box if you intend to use secure LDAP.
12. Click the **Site** dropdown list to select your site.
13. Click the **Multifactor Authentication** dropdown list to select the desired authentication method.
14. Click the **Validate & Save** button. The information is validated. If there are any connectivity issues, an error message will appear stating what field is the likely cause. If the Active check box is not selected no validation occurs. If you chose anonymous authentication, no secret is needed and no credential validation occurs; however the distinguished name and FQDN are still used. Upon a successful save, a new box appears, prompting the user to select their initial synchronization groups. If groups appear in the search box that also indicates the connection was successful.

Discovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Discovery is the process where Secret Server scans an environment to find accounts and associated resources called *dependencies*. Once accounts are found, they can be used to create new secrets in Secret Server. Users with the "administer discovery" role permission can either manually import accounts or can create an automated process to do so. Using discovery does not stop users from manually creating their own secrets.

Some typical accounts that discovery can find include Windows local admin, Windows domain, and Unix non-daemon. Some typical dependencies discovery can scan for include scheduled tasks running as a domain user, application pools running as a domain user, and services running as a domain user.

Note: Account and dependency types not supported out-of-the-box in Secret Server can still be discovered by writing PowerShell scripts that you can run as custom scanners. See [Extensible Discovery](#).

We suggest reading (in order):

- [How Discovery Works](#)
- [Introduction to Discovery Sources, Scanners, and Templates](#)
- [Running and Interpreting Active Directory Discovery](#)

Quick Initial and Ongoing Importation of Network Credentials

By using discovery, your Secret Server offsets the burden of keeping track of computers and accounts on your network. This can be especially beneficial when getting started for discovering and importing accounts in bulk, as well as having Secret Server find accounts and create secrets whenever a new machine or account is provisioned.

Protection Against Backdoor Accounts

When Secret Server is configured to discover new accounts, it provides added protection by regularly running discovery on your network to identify those accounts. Secret Server adds the new accounts to its records and resets the accounts password to values that meet your security policy. Consequentially, if someone is setting up backdoor admin accounts on the network, they cannot use those accounts very long before they are imported into Secret Server and their passwords are changed with Remote Password Changing (RPC).

Active Directory Discovery

Secret Server AD discovery scans for AD machines, AD user accounts, local Windows accounts, and dependencies on an AD domain. First, Secret Server discovers machines from your domain. Next, Secret Server scans each machine for local Windows accounts and dependencies. By default, Secret Server scans for local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools. You can discover additional accounts and dependencies by creating PowerShell scanners. PowerShell scanners are an advanced topic described in the [Extensible Discovery](#) section.

ESX/ESXi Discovery

Secret Server provides a wizard to help configure ESX/ESXi discovery. You name the discovery Source, define the host ranges of the desired IP addresses, and choose a secret to use as credentials when scanning.

Note: Secret Server provides a "Generic Discovery—Only Credentials" secret type that stores a simple username and password pair for Unix or ESX/ESXi discovery. It is intended only for discovery and is incapable of RPC.

AWS Discovery

Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. Two types of secrets can be discovered and managed through Secret Server:

- AWS Access Key: Keys used for programmatic integration with AWS.
- AWS Console Account: User login accounts for AWS.

Google Cloud Platform Discovery

Secret Server can manage Google Cloud Platform (GCP) service accounts and VM instances. This feature allows users to run discovery to pull and manage VM Instances, as well as import and manage GCP service accounts.

Unix Discovery

Secret Server provides a wizard to help configure Unix discovery. You name the discovery Source, define the host ranges of the desired IP addresses, and choose a secret to use as credentials when scanning. The default command sets that Secret Server ships with discovers machines and accounts in most Unix environments.

By default, the "Find Non-Daemon Users (Basic Unix)" command set is used first. If a built-in account is discovered, you must modify the discovery source to use the "Find All Users (Basic Unix)" command set. You can create new command sets by clicking the Configure tab on the Discovery Sources page.

Extensible Discovery

You can customize discovery by changing parts of it to use PowerShell. The information a discovery scanner outputs is defined by its scanner template. For standard templates, the input and output information types are fixed. Extensible discovery allows you to customize or replace the unmanaged account, IP address and OU, account, and dependency discovery steps above. Extensible discovery does still have limitations on what information is passed between discovery scanners. For more information, see [Extensible Discovery](#).

Please see our [Discovery Best Practices Guide](#) to learn about optimizing discovery performance.

Automated Discovery

The following is a high-level overview of how the most common type of automated discovery works without customization. Discovery is organized into an ordered set of discovery scans that pass information based on input and output templates. This is all configured by default. You cannot alter the out-of-the-box discovery scanners, but you can copy them and then modify the copy.

Automated Discovery Terms

First, discovery has several terms that need defining:

Discovery Source

A named object that conducts discovery. There are five broad types: Active Directory, Amazon Web Services, Unix, VMware ESX\ESXi, and Google Cloud Platform.

Configuring discovery is defining the parameters of the discovery source, once the general type is chosen.

Discovery Scanner

A discovery component that collects information during a discovery. There are four general types, called *scan templates* (in their sequential running order): Find host ranges, Find machine, Find local accounts, and Find dependencies.

A discovery source consists of an ordered sequence of discovery scanners. Each scanner has a defined input and output. A discovery source can have more than one scanner of a given type.

Discovery Input Template

The defined input type for a discovery scanner. An instance of the template contains the data needed to conduct the scan. The input template is often, but not always, an output template of the preceding scanner in the sequence. Some examples include Active Directory domain, AWS discovery source, organizational unit, and Windows computer.

Discovery Output Template

The defined output type for a discovery scanner. An instance of the template contains the data produced by the scan. The output template is often, but not always, an input template of the next scanner in the chain. Other times, the output may be used by another non-adjacent scanner in the discovery source. Some examples include: Active Directory account, AWS access key, ESXi local account, host range, organizational unit, and Windows local account.

Discovery Rule

Discovery rules automatically create credential secrets or send emails when local accounts matching the rule criteria are discovered. Discovery rules are set in the discovery network view page because they are specific to portions of the discovered network and can be as granular as desired. Credential secrets can also be manually created as desired.

Example Automated Discovery Process

A typical automated discovery process for Active Directory domains, running on an interval, looks like this:

Note: The majority of current discovery processes are for AD discovery source type. The others types differ by input and output but follow a similar process.

Note: Even though automatic discoveries run on a set interval, you cannot schedule when those occur. The interval is from

whenever the discovery last ran.

1. Discovery matching runs. The discovery matcher creates a link between existing active secrets and any existing secrets in Secret Server based on their machine names, accounts and dependencies. The matcher is automatic. When matches are found, the corresponding existing discovery results appear as "managed" in the discovery network view with a link to the existing secret or dependency.
2. Discovery rules run and attempt to match any unmanaged discovery results to the rule's parameters. If a rule matches the results, discovery automatically imports the results using the settings in the discovery rule. Once finished, discovery begins.
3. The Find Host Ranges scanner (using the Windows Discovery base scanner) runs with an Active Directory Domain input template. The scanner determines which OUs are to be scanned and populates its Organizational Unit output template with a list of those OUs. The output template will be used by the following Find Machine scanner and also by the Find Local Accounts scanner, which does not require machine information.
4. The Find Machine scanner (using the Windows Discovery base scanner) examines OUs from its Organizational Unit input template via LDAP and creates a list of machines with which it populates its Windows Computer output template. This is the list of computers to run a dependency scan on. The Find Dependencies scanner uses this instance of the output template as its input template.
5. The Find Local Accounts scanner (using the File Load Discovery base scanner) examines OUs from its Organizational Unit input template via LDAP and creates a list of all AD admin accounts with which it populates its Active Directory Account output template. This is the list of discovered admin accounts.
6. The Find Dependencies scanner (using the Windows Discovery base scanner) examines a list of machines from its Windows computer input template using various technologies. For example, application pools use Microsoft Web Administration (WMA) or, failing that, Windows Management Instrumentation (WMI). Services use WMI, and scheduled tasks use Windows' task scheduler interfaces. The Find Dependencies scanner can return any number of output templates as desired. These include: Com+ Application, Computer Dependency (Basic), PS Dependency, Remote File, SQL Dependency (Basic), SSH Dependency (Basic), SSH Key Rotation Dependency, Windows Application Pool, Windows Scheduled Task, and Windows Service.

The discovered dependencies for local accounts are displayed at **Admin > Discovery > Discovery Network View > Local Accounts**. Returned accounts for AD users are displayed at **Admin > Discovery > Discovery Network View > Domain > Cloud Accounts**.

Note: Any dependencies that were discovered in prior discovery runs that are no longer present are removed from the discovery results, and their secret dependencies are deactivated.

Manual Discovery

You can also run discovery manually by going to **Admin > Discovery** and clicking the **Run Now** button and selecting **Discovery Scan**. We recommend waiting for any automatic discovery to idle before starting a manual discovery run. A discovery scan runs the first four of the automated steps above. When you click the "Run Now" button on the Scan Computers tab, the last two are run. These steps are the most time intensive steps because many machines may be scanned.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The sub-topics here discuss discovery for five platforms:

- [Active Directory](#)
- [Amazon Web Services](#)
- [Google Cloud Platform](#)
- [Unix](#)
- [VMware ESX/ESXi](#)

Active Directory Discovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server queries AD domains to obtain a list of Organizational Units (OUs) and Windows computers on the domain. These OUs and computers are recorded in the Secret Server database. Secret Server then attempts to connect to each computer and query for the following:

- **Domain Accounts:** AD user accounts
- **IIS Application Pools:** IIS application pools run by AD accounts
- **Local Accounts:** Local Windows accounts
- **Windows Services:** Windows services run by AD accounts
- **Scheduled Tasks:** Windows scheduled tasks run by AD accounts

Active Directory Local Account Discovery Methods

Remote Procedure Calls (RPC)

This is the method that is used for local account discovery for all versions of Secret Server prior to release 8.6.000000 and is the default for all upgrades and fresh installations. It uses the same technology as the Windows remote password changing in Secret Server and is the most dependable and proven of the options. It can, however, be slower in some environments when scanning computers over a WAN.

Windows Management Instrumentation (WMI)

This method uses the WMI technology to query the Windows computer. In some environments, this method can be faster than the Remote Procedure Call. It does, however, require having the proper permissions and network configuration setup correctly for WMI to run.

Attempt WMI First, and Fallover to RPC if Needed

This option attempts to use the WMI method first, and if that fails to work correctly, it attempts the RPC method for local account discovery. This option is potentially slower because it has the possibility of performing two separate scans for each computer.

Creating an Active Directory Discovery Source

Discovery sources define a set of discovery operations. You must create one based on the built-in types prior to running discovery. To do so for AD:

Note: Adding a new domain as a discovery source also adds it as a synchronization source and vice versa.

Note: If you add a domain as an AD synchronization source within Secret Server but discovery was not initially enabled, the domain is listed as an inactive discovery source. To see such a domain, on the Discovery Sources page, click to select the Show Inactive and Disabled check box.

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

The screenshot shows the 'Admin > Discovery' page. At the top, there are navigation tabs: 'Discovery Sources' (selected), 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there are three buttons: 'Discovery Network View', 'Create Discovery Source' (with a dropdown arrow), and 'Run Discovery Now' (with a dropdown arrow). The page displays two summary cards: 'Discovery' and 'Computer Scan', both showing 'Last Started: 2 months, 8 days ago' and 'Next Run: soon'. Below these cards, there is a search bar and a toggle for 'Include Inactive'. The main content is a table with the following data:

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 ...
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 ...
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Note the list of existing discovery sources.

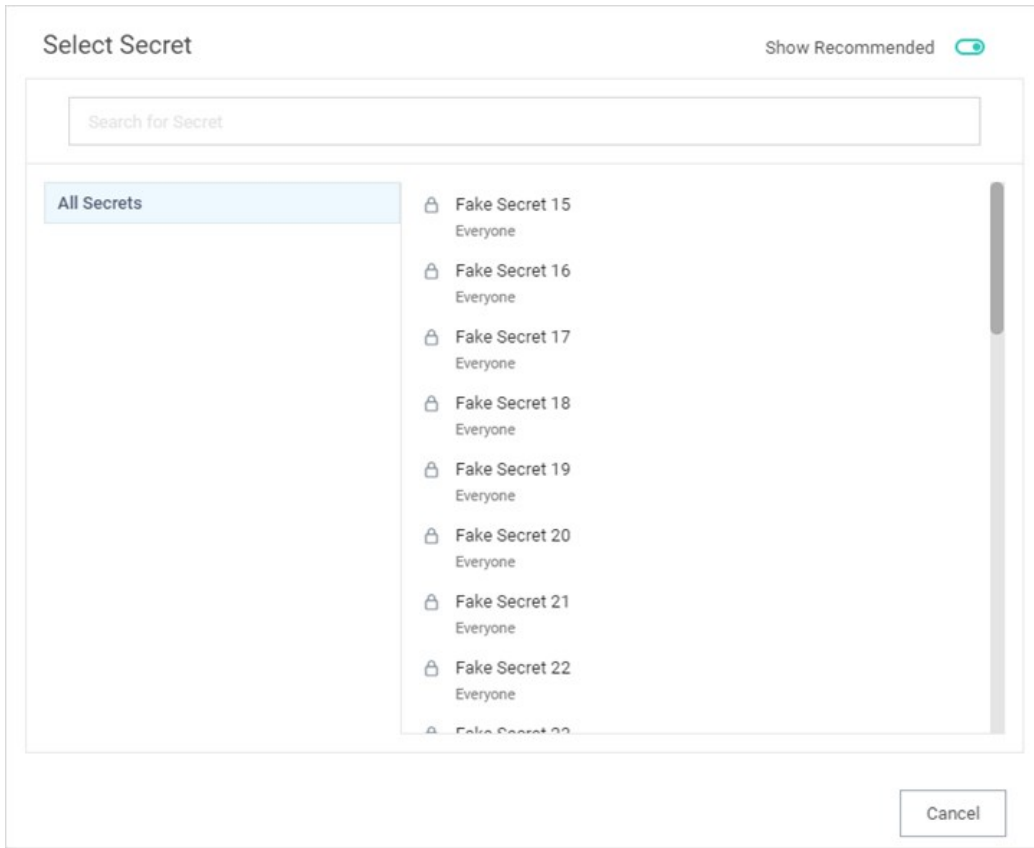
Note: If you upgraded from an earlier Secret Server version and have created an AD domain within Secret Server, a corresponding discovery source is displayed on this page. If discovery was not enabled on that domain, the discovery source Active column is not checked for that discovery source.

3. Click the **Create Discovery Source** button and select **Active Directory** to choose that discovery source type. A Discovery Source page appears for that type:

Discovery Source

Discovery Source Name *	<input type="text"/>
Fully Qualified Domain Name *	<input type="text"/>
Friendly Name *	<input type="text"/>
Active *	<input checked="" type="checkbox"/>
Discovery Secret *	No Secret Selected Create New Secret
Discovery Site *	Local <input type="text"/>
Discover Specific OU *	<input type="checkbox"/>
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended) <input type="text"/>
Use LDAPS *	<input type="checkbox"/>

4. Type the parameters for the discovery source name, FQDN, and friendly name. The parameters with asterisks are required.
5. Ensure the **Active** check box is selected. This activates this discovery Source for scanning. Active discovery sources are scanned at the defined discovery interval defined. If you have multiple discovery sources, the discovery source with the most un-scanned computers is scanned first.
6. Next, you select a secret that is used as the credentials for discovery scanning and AD synchronization. These credentials must have the proper rights to scan the remote machines. Click the **No Secret Selected** link. The Select Secret popup page appears:



1. **Either** search for and click the secret you want to use for the account credentials during the scan. The popup page closes. The name of the secret you chose replaces the No Secret Selected link.

Or create a new secret for the credentials:

1. Click the **Create New** Secret link. The Create New Secret page appears:

Create New Secret

No Folder Selected [Change](#)

Choose a Secret Template

- Combination Lock
- Contact
- Copy of CreditCard
- Copy of Unix Account SSH
- Credit Card
- DevOps Secrets Vault Client Credentials
- Generic Discovery Credentials**
- Generic ODBC (DataSource)
- Google IAM Service Account Key
- Healthcare
- HP iLO Account (SSH)
- IBM iSeries Mainframe
- MySql Account

2. Click the **Generic Discovery Credentials** secret template. Another Create New Secret page appears:

Create New Secret

Secret Template Generic Discovery Credentials [Change](#)

Folder [No Folder Selected](#)

Secret Name *

Username *

Password Show Generate

Notes

Generate SSH Key

Private Key [Change](#)

Cancel Create Secret

3. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.
4. Click the **Create Secret** button.
2. Click the **Discovery Site** dropdown list to select the desired site for the discovery source. If distributed engines are setup, the list shows all active sites. If no distributed engines are setup, the list defaults to local, and you cannot change it.
3. Click the **Discover Specific OU** check box to limit your discovery to an OU. See [Enabling Specific OU Domain Discovery](#) to define the scanned OU. When you select this option, a Domain Scope tab appears on the Discovery Source page for the created AD discovery source.
4. Leave the **Machine Resolution Type** dropdown list set to **Use Machine and Fully Qualified Name** unless you have a specific reason to change it.
5. Click the **Create** button. Secret Server attempts to access the domain with your specified credentials to ensure the configuration is correct. Thus, Secret Server must have access to the domain provided, and the account credentials must work.

Setting Permissions for Active Directory Scans

Local Windows Accounts

The scanning account needs the "Access This Computer From the Network" permission (and possibly one more) on the endpoint:

1. Open the local group policy editor (gpedit.msc).
2. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
3. Double-click the **Access this computer from the network** policy. The properties for the policy appears.
4. Ensure the scanning account is one of the listed users. If not, click the **Add User or Group** button to add it.
5. Look at the following list of operating systems and updates to determine if any of them match your system:
 - o Windows 10, version 1607 and later
 - o Windows 10, version 1511 with [KB 4103198](#) installed
 - o Windows 10, version 1507 with [KB 4012606](#) installed
 - o Windows 8.1 with [KB 4102219](#) installed
 - o Windows 7 with [KB 4012218](#) installed
 - o Windows Server 2019
 - o Windows Server 2016
 - o Windows Server 2012 R2 with [KB 4012219](#) installed
 - o Windows Server 2012 with [KB 4012220](#) installed
 - o Windows Server 2008 R2 with [KB 4012218](#) installed

Note: For more information on this security issue, see [Network access: Restrict clients allowed to make remote calls to SAM](#).

Windows Services, Scheduled Tasks, App Pools, and COM+ Applications

Note: There are special considerations for discovering service accounts running COM+ Applications, please contact Delinea for more information.

To scan for service accounts, the account entered must be a domain account that is in the Administrators group on the target machines. Follow the instructions below in either case to ensure your account has the privileges to run a successful scan:

1. Open the group policy editor for your domain policy.
2. Go to **Computer Configuration > Preferences > Control Panel Settings**.
3. Right-click **Local Users and groups** and select **New > Local Group**.
4. Leave the **Action** dropdown list set to **Update**.
5. Click to select **Administrators (Built-in)** in the **Group Members** dropdown list.
6. Click the **Add...** button.
7. Search for the account you will use for discovery scanning.
8. Click the **OK** button to save your changes. The next time the group policy updates across your environment, the discovery account will be part of the local administrators group.
9. For strong security, configure the group policy to limit the logon privileges of that account:

1. Open the group policy editor
2. For your domain policy, go to **Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
3. Add your discovery account to the **Deny log on locally** policy.
4. Add your discover account to the **Deny log on through Remote Desktop Services** policy.
5. (Optional) Ensure the account is not part of the remote desktop users group.

Running and Interpreting Active Directory Discovery

This topic discusses how to configure, run, and interpret discovery scans on Active Directory systems. After the initial configuration, normally the discovery source is set to active, which runs the follow-on tasks automatically. You can also manually run either a discovery (locate computers on the domain) or computer (inspect the individual computers) scan.

Step One: Discovery Configuration

Running a discovery on an AD system is easy, assuming everything was configured correctly. To that end, follow these instructions first:

- [Setting Permissions for Active Directory Scans](#)
- [Creating Active Directory Discovery Sources](#)
- [Enabling Specific OU Domain Discovery](#) (optional)

Step Two: Discovery Scan

When you complete the configuration and there is at least one active discovery source and discovery is enabled (the Active check box is selected), you can run a discovery scan manually or wait for an automatic one to start. A typical scan:

1. Runs discovery matching: The discovery matcher creates a link between existing active secrets and any existing secrets in Secret Server based on their machine names, accounts and dependencies. The matcher is automatic. When matches are found, the corresponding existing discovery results appear as "managed" in the discovery network view with a link to the existing secret or dependency.
2. Runs discovery rules: Secret Server attempts to match any unmanaged discovery results to the rule's parameters. If a rule matches the results, discovery automatically imports the results using the settings in the discovery rule. Once finished, discovery begins.
3. Runs the find host ranges scanner: The scanner (using the Windows discovery base scanner) runs with an Active Directory domain input template. The scanner determines which OUs are to be scanned and populates its organizational unit output template with a list of those OUs. The output template will be used by the following find machine scanner and also by the find local accounts scanner, which does not require machine information.
4. Runs the find machine scanner: The scanner (using the Windows Discovery base scanner) examines OUs from its organizational unit input template via LDAP and creates a list of machines with which it populates its Windows computer output template. This is the list of computers to run a dependency scan on. The find dependencies scanner uses this instance of the output template as its input template.

Note: You can see logs of this process by going to the Discovery Logs tab on the Discovery page.

To run a manual discovery scan, on the **Admin** menu, click the **Run Discovery Now** button and select **Run Discovery Scan**.

Step Three: Computer Scan

Once the computers in the desired AD domain or OU are discovered, a computer scan runs AD queries on each machine found during the discovery scan to attempt to collect the information the discovery source was configured to collect, which can include local accounts, Windows services, scheduled tasks, and IIS application pools. Specifically, the scan:

1. Runs the find local accounts scanner: Using the file load discovery base scanner, Secret Server examines OUs from its organizational unit input template via LDAP and creates a list of all AD admin accounts with which it populates its Active Directory account output template. This is the list of discovered admin accounts.
2. Runs the find dependencies scanner: Using the Windows discovery base scanner, Secret Server examines a list of machines from its Windows computer input template using various technologies. For example, application pools use Microsoft Web Administration (WMA) or, failing that, Windows Management Instrumentation (WMI). Services use WMI, and scheduled tasks use Windows' task

scheduler interfaces. The find dependencies scanner can return any number of output templates as desired. These include: com+ application, computer dependency (basic), PS dependency, remote file, SQL dependency (basic), SSH dependency (basic), SSH key rotation dependency, Windows application pool, Windows scheduled task, and Windows service.

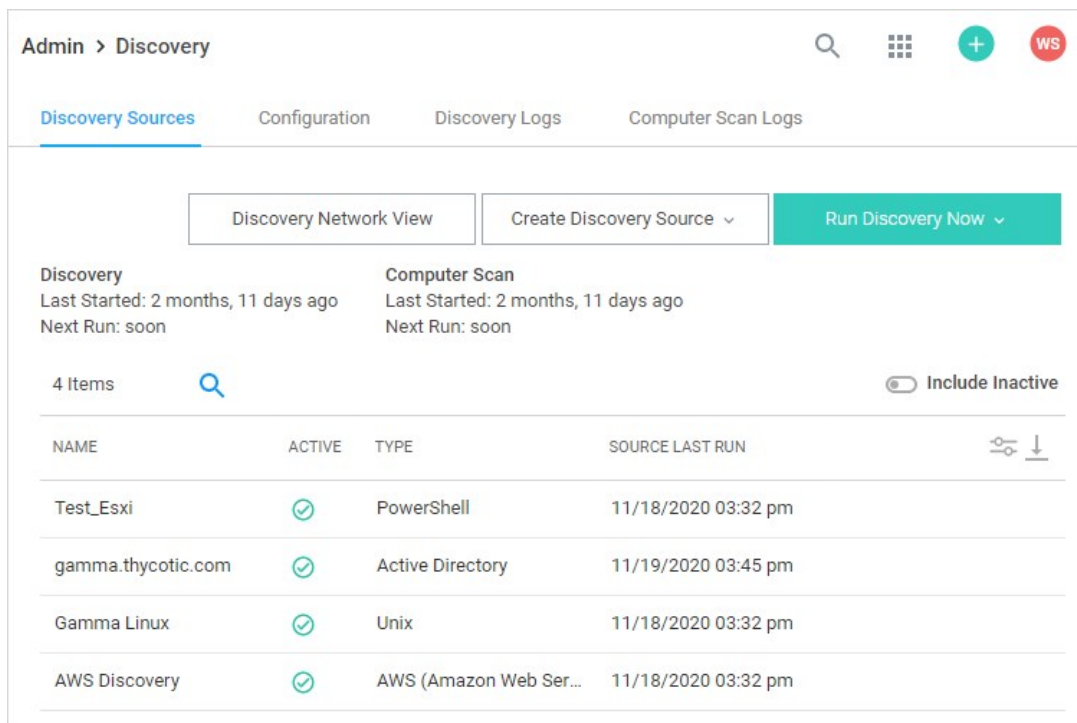
Note: You can see logs of this process by going to the Computer Scan Logs tab on the Discovery page.

To run a manual computer scan, on the **Admin** menu, click the **Run Discovery Now** button and select **Run Computer Scan**.

Step Four: Viewing Discovery Results

Browsing Discovery Results

1. Go to **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:



The screenshot shows the 'Admin > Discovery' page. At the top, there are navigation tabs: 'Discovery Sources' (selected), 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there are three buttons: 'Discovery Network View', 'Create Discovery Source', and 'Run Discovery Now'. The page is divided into two sections: 'Discovery' and 'Computer Scan'. Both sections show 'Last Started: 2 months, 11 days ago' and 'Next Run: soon'. Below these sections, there is a search bar with '4 Items' and a toggle for 'Include Inactive'. A table lists the discovery sources:

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	✓	PowerShell	11/18/2020 03:32 pm
gamma.thycotic.com	✓	Active Directory	11/19/2020 03:45 pm
Gamma Linux	✓	Unix	11/18/2020 03:32 pm
AWS Discovery	✓	AWS (Amazon Web Ser...	11/18/2020 03:32 pm

2. Click the **Discovery Network View** button. The Discovery Network View Page appears:

Discovery Network View

Explain

The Discovery Network View shows the computer accounts that have been found by Discovery.

The Domain Tree on the left displays the OUs available in the domain. Clicking on an OU displays the computers in that OU in the search grid. Select the Computer accounts to be imported into Secret Server and click the Import button to have Secret Server automatically create Secrets for those accounts.

The screenshot shows the 'Discovery Network View' interface. At the top, there are tabs for 'Local Accounts', 'Service Accounts', and 'Domain \ Cloud Accounts'. The 'Local Accounts' tab is active, showing a search box and a tree view of the domain structure. The tree view shows 'gamma.thycotic.com' expanded, with sub-items 'Gamma Linux', 'AWS Discovery', and 'Test_Esxi'. To the right, there is an 'Advanced' search box and a table with columns: 'COMPUTER', 'ACCOUNT', 'SCAN TEMPLA'OS', 'CONTAINER', 'SECRET', and 'LAST SC.STATUS'. Below the table are buttons for 'Import', 'Create Rule', and 'View Rules'. The table is currently empty, displaying 'Page 1 of 0' and 'No records to view'. A 'Back' button is located at the bottom left.

The Discovery Network View page shows any discovered computer accounts. The domain tree on the left displays the domains as folders with OUs for that domain presented as folder contents. Clicking on a folder and then on an OU displays the computers in that OU in the table on the right. For example:

This screenshot shows the same interface as the previous one, but with the 'Domain \ Cloud Accounts' tab selected. The domain tree on the left shows '10.60.12.1/24' expanded, with a sub-item '10.60.19.1/24'. The table on the right now contains several rows of data. The first four rows show computer accounts with IP addresses and failed status. The last three rows show user accounts with usernames and 'Unman...' status.

COMPUTER	ACCOUNT	SCAN TEMPLA'OS	CONTAINER	SECRET	LAST SC.STATUS
10.60...			10.60.12.1/:		11/18/ Failed t...
10.60...			10.60.12.1/:		11/18/ Failed t...
10.60...			10.60.12.1/:		11/18/ Failed t...
10.60...			10.60.12.1/:		11/18/ Failed t...
10.60...	root	SSH Local A Linux	10.60.12.1/:		11/18/ Unman...
10.60...	developer	SSH Local A Linux	10.60.12.1/:		11/18/ Unman...
10.60...		SSH Local A Linux	10.60.12.1/:		11/18/ Unman...

Note: For large numbers of domains you can type the domain name in the unlabeled search box over the domain folder tree and press <Enter> to narrow what domains are presented to you.

The discovery page has tabs for local account, service accounts, and domain or cloud accounts. All are very similar and draw from the same network tree on the left.

Searching Discovery Results

To search for a specific discovery source or OU, type the source or OU name in the search bar displayed at left. If results are found, click the result shown below the search field to highlight it. Now, only machines from that source or OU will be displayed at right.

To search for a specific computer name, account, or service name, type the search term in the search field on the right. Matching results are filtered below the search field.

To use advanced search settings, click the **Advanced** link beside the search field. The Search By and Account Status dropdown list boxes and Create Rule button appear:

Select an option in the **Search By** menu to narrow the search results to match an account, computer, operating system, or rule.

Note: "Rule" only appears in the list box if discovery rules exist for local accounts. When you select it, another menu appears for selecting a rule. For more information about creating and searching with rules, see [Discovery Rules](#).

Click the **Account Status** dropdown list to select accounts managed or unmanaged by Secret Server.

Click to select the **Include Child Organization Units** check box to match search results within child OU's of the OU highlighted in the folder tree.

Understanding Discovery Results

The table below describes the contents of each column:

Table: Discovery Results

Column	Description	Account Type (Local, Service)
Account	Username of discovered account.	Both
Computer	Computer name of the machine scanned.	Both
Last Connected	Last date a user logged into the machine.	Both
Last Scanned	Last date that the machine was scanned by discovery.	Both
Org Unit	Organizational Unit the machine is joined to. This information is obtained from AD during the first part of the discovery process.	Local
Secret	If a secret name appears here, a credential secret already exists for the account listed in the account column. Otherwise, this column is blank.	Both
Service Name	Name of a discovered dependency.	Service
Status	Indicates that an account is managed by Secret Server, connectivity issues, or no accounts detected. For more information about error messages, see Discovery Error Messages .	Both
Type	Discovered dependency type icon. See the following table.	Service

[Unexpected Link Text](#) Service account dependency types identified in the **Type** column:

Table: Service Account Dependency Types

Type	Icon	Service Name
Application Pool		IIS application pool name
Scheduled Task		Scheduled task name
Windows Service		Service name

[Unexpected Link Text](#)

Note To correctly identify and import IIS application pools for IIS 7 or higher, Secret Server requires a trust relationship between the scanned domain and domain that the Secret Server Web server is joined to.

AWS Account Discovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: Discovery must be enabled in Secret Server to discover AWS accounts.

Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. Two types of secrets can be discovered and managed through SS:

- AWS Access Key: Keys used for programmatic integration with AWS.
- AWS Console Account: User login accounts for AWS.

AWS Instance Discovery

Secret Server can now scan for instance resources in AWS. You can add this ability in the scanner settings section or through the wizard.

1. Create and AWS discovery source. See [Enabling AWS Discovery](#).
2. Navigate to **Admin > Discovery**:

Admin > Discovery

Discovery Sources | Configuration | Discovery Logs | Computer Scan Logs

Discovery
Last Started: 4 hours, 22 minutes ago
Next Run: 19 hours, 37 minutes

Computer Scan
Last Started: 4 hours, 50 minutes ago
Next Run: 19 hours, 9 minutes

Discovery Network View | Create Discovery Source | Run Discovery Now

4 Items Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web Services)	7/17/2020 10:50 am
Gamma Linux	<input checked="" type="checkbox"/>	Unix	7/17/2020 10:46 am
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	7/17/2020 10:31 am
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	7/17/2020 10:55 am

3. Click the **Create Discovery Source** dropdown list and select **AWS (Amazon Web Services)**. The AWS Discovery Source wizard Overview page appears:

AWS Discovery Source

[Overview](#)

Getting Started

Amazon Web Services (AWS) Discovery allows you to use Secret Server to scan for IAM Credentials, Regions, and Instances.

The Wizard will help you get AWS Discovery configured in 5 simple steps:

1. Name the Discovery Source.
2. Choose the Site used for Discovery scanning.
3. Select the IAM Credential Scanner.
4. Provide the Region Scanner information.
5. Choose a Secret to use as credentials for Discovery scanning.

4. Click the **Next** button. The Discovery Source Name page appears:

AWS Discovery Source

Overview > Discovery Source Name

Discovery Source Name
Define the Name of the new Discovery Source

i Choose a name that will help quickly identify this Discovery source in the future.

← Previous ✕ Cancel → Next

5. Type the name of the AWS discovery source in the **Discovery Source Name** text box.
6. Click the **Next** button. The Site page appears:

AWS Discovery Source

Overview > Discovery Source Name > Site

Add Site
Select the Site to be used for this Discovery Source

Local ▾

i The list contains all active Sites regardless of whether they have an active Engine.

← Previous ✕ Cancel → Next

7. Click the **Add Site** list box to select the site.
8. Click the **Next** button. AWS Service Account Scanner page appears:

AWS Discovery Source

Overview > Discovery Source Name > Site > IAM Credential Scanner

IAM Credential Scanner
Select which IAM users to scan.

FIND ACCOUNTS

AWS User Account Scanner

AWS Access Key Scanner

9. Click the check boxes for the scanners you desire.

10. Click the **Next** button.

AWS Discovery Source

Overview > Discovery Source Name > Site > IAM Credential Scanner > AWS Region Scanner

AWS Region Scanner
Enter a comma-delimited list of regions that will be scanned for availability zones. Example: us-east-1,us-west-1

SCAN AWS INSTANCES

Scan AWS Instances

SCAN AWS REGIONS

AWS Region Scanner

FIND MACHINES

AWS Windows Machine Scanner

AWS Machine (Non-Windows) Scanner

i Regions must be listed in a comma delimited list in order for instances to be discovered.

11. Click to select the **Scan AWS Instances** check box.
12. Type the regions you wish to scan for instances. The regions must be listed in a comma-delimited list for instances to be discovered.

Note: See [Regions, Availability Zones, and Local Zones](#) for more information on AWS regions.

13. Click to select the check boxes for the scanners you desire:
 - **AWS Windows Machine Scanner:** This is a machine scanner that scans each region and pulls all of the AWS Windows OS VM instances.
 - **AWS Machine (Non-Windows) Scanner:** This is a machine scanner that scans each region and pulls all of the AWS Non-Windows OS VM instances.
14. Click the **Next** button. The Credential Secrets page appears:

AWS Discovery Source

Overview > Discovery Source Name > Site > IAM Credential Scanner > AWS Region Scanner > Credential Secrets


Credential Secrets

Select the IAM Access Key Secret to be used as Authentication Credentials for Discovery.

[Add Secret](#)

If a suitable Secret is not already stored in Secret Server, create it below.

[Create New Secret](#)

 Only AWS IAM Access Key may be used. IAM Console passwords accounts may not be used to query the API.

[← Previous](#) [✕ Cancel](#) [🚩 Finish](#)

15. Click the **Add Secret** link. The Select a Secret popup appears:

Select a Secret

< All Folders >

- Personal Folders
- Everyone
- RPC/Heartbeat

SECRET	FOLDER	TEMPLATE
<input type="checkbox"/> [blurred]	Launcher	Unix Account (SSH)
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account
<input type="checkbox"/> [blurred]	Everyone	Windows Account

16. Navigate the folder tree and select the secret you created earlier. As soon as you select the check box, the popup disappears and the secret appears under the Add Secret link.

17. Click the **Finish** button.

Enabling AWS Discovery

1. For Secret Server to communicate with AWS, users with sufficient privileges need to create an access key for their account in AWS Identity and Access Management (IAM). The account used to do this requires the following permissions to discover users and access keys:

- iam:ListUsers
- iam:GetLoginProfile
- iam:ListAccessKeys

Note: These permissions are limited to the resources the user is allowed to access.

2. Once this access key is created, use the access key and secret key to create a secret in Secret Server using the Amazon IAM key template.
3. Create a new AWS discovery source and use the Amazon IAM key as the credentials secret for that discovery source.

Note: AWS only allows programmatic integration through access keys. This type of secret is required for discovery to work. Discovery must be enabled in Secret Server for this feature to work.

Password Management in AWS

Secret Server can manage password and access keys for AWS IAM accounts.

Amazon IAM Keys

Password changing, privileged password changing, and running heartbeats are available for Amazon IAM key secrets. When an Amazon IAM key has its password changed through Secret Server, the new secret key is generated automatically and is not set by user input.

During password changing, you can disable or remove old keys through settings available in the advanced configuration:

- `<add key="ShouldDeletePreviousKey" value="true" />`
- `<add key="ShouldInactivatePreviousKey" value="true" />`

Important: Altering advanced settings can significantly impact the performance and behavior of Secret Server, so there is no direct link anywhere in Secret Server to the Advanced Settings page. If you need to change any advanced setting (as mentioned in this guide), please contact Delinea Technical Support.

Amazon IAM Console Password

Password changing, and privileged password changing are available for Amazon IAM console password secrets. Due to AWS IAM's restrictions on programmatic integration, this secret type cannot use Secret Server heartbeat.

In addition, an Amazon IAM key secret must be associated with an Amazon IAM console password secret for password changing to occur. To associate the two:

1. Create the Amazon IAM console password secret, and an Amazon IAM Key secret for an account that has the permissions to change the console user's password. This can be the console account's own access keys, if the user has permission.
2. Navigate to the RPC tab of the Amazon IAM Console Password.
3. Under **Change Password Using Privileged Account** select **Edit** and choose the IAM key secret created in the previous step. RPC should now be possible on the console password secret.

Permissions Required for Secret Key Changes

Note: These permissions are at the most granular level. You can implement broader methods through wildcard resource restrictions, permission policies, or groups.

Privileged Permissions: (those the AWS account needs to change another users' access keys):

- `iam:DeleteAccessKey ON RESOURCE arn:aws:iam::<account>:user/<otherUserName>`
- `iam:UpdateAccessKey ON RESOURCE arn:aws:iam::<account>:user/<otherUserName>`
- `iam:CreateAccessKey ON RESOURCE arn:aws:iam::<account>:user/<otherUserName>`
- `iam:ListAccessKeys ON RESOURCE arn:aws:iam::<account>:user/<otherUserName>`

Basic Permissions (those the AWS account needs to change its own access keys):

- `iam:DeleteAccessKey ON RESOURCE arn:aws:iam::<account>:user/${aws:username}`
- `iam:UpdateAccessKey ON RESOURCE arn:aws:iam::<account>:user/${aws:username}`
- `iam:CreateAccessKey ON RESOURCE arn:aws:iam::<account>:user/${aws:username}`

- iam:ListAccessKeys ON RESOURCE arn:aws:iam::<account>:user/\${aws:username}

Permissions Required for Changing the Amazon IAM Console Password

Note: These permissions are at the most granular level. You can implement broader methods through wildcard resource restrictions, permission policies, or groups.

The permissions are:

- Privileged Permission: iam:UpdateLoginProfile ON resource arn:aws:iam::account>:user/<otherUserName>
- Basic Permission: iam:ChangePassword ON RESOURCE arn:aws:iam::<account>:user/\${aws:username}

Viewing AWS Discovery Source Scanners

To view these scanners:

1. Go to **Admin > Discovery**.

The screenshot shows the 'Admin > Discovery' page. At the top, there are tabs for 'Discovery Sources', 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there is a 'Discovery' status indicator (Running, Just Now) and a 'Computer Scan' status indicator (Last Started: 29 minutes ago, Next Run: 23 hours, 30 minutes). There are also buttons for 'Discovery Network View', 'Create Discovery Source', and 'Run Discovery Now'. Below this, there is a search bar and a toggle for 'Include Inactive'. The main content is a table with the following data:

NAME	ACTIVE	TYPE	SOURCE LAST RUN
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web Services)	7/29/2020 02:00 pm
[Redacted]	<input checked="" type="checkbox"/>	Unix	7/29/2020 02:00 pm
[Redacted]	<input checked="" type="checkbox"/>	Active Directory	7/30/2020 12:10 pm
[Redacted]	<input checked="" type="checkbox"/>	PowerShell	7/29/2020 02:00 pm

2. Click the discovery source name link in the table. The Discovery Source page for it appears:

The screenshot shows the 'Discovery Source' page for 'AWS (Amazon Web Services)'. At the top, there are tabs for 'Discovery Source' and 'Audit', and a 'Scanner Settings' button. Below the tabs, there is a description: 'Scan and discover resources in Amazon Web Services' with a 'KB Link'. The settings are as follows:

Discovery Source Name *	AWS Discovery
Active *	Yes
Discovery Site *	Local
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended)

3. Click the **Scanner Settings** button in the top right of the page. The Discovery Source Scanner Settings page appears, which lists the scanners.

Discovery Source Scanner Settings

FIND HOST RANGES

+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
AWS Region Scanner	AWS Discovery Source	AWS Region	
AWS Path Scanner	AWS Discovery Source	AWS Path	

Scanners: 2

FIND MACHINES

+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
AWS Windows Machine Scanner	AWS Region	Windows Computer	
AWS Machine (Non-Windows) Scanner	AWS Region	Computer	

Scanners: 2

FIND ACCOUNTS

+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
AWS User Account Scanner	AWS Path	AWS User Account	
AWS Access Key Scanner	AWS Path	AWS Access Key	

Scanners: 2

FIND DEPENDENCIES

+ Add New Dependency Scanner

Add a dependency scanner once you have a machine scanner added to the discovery source.

4. Click pencil edit icon for the machine listing. The settings for that scanner appears:

Settings - AWS Windows Machine Scanner x

SECRET CREDENTIALS

1. AWS Discovery cred
[Add Secret](#)
[Add Secret Search Filter](#) ⓘ Create Secret Search Filter

ADVANCED SETTINGS

Platform Include Filter ⓘ

Platform Exclude Filter ⓘ

Custom Additional Filters ⓘ

Instance Name Preference ⓘ

✓ OK ✕ Cancel

5. Complete the following settings:

- **Platform Include Filter:** Comma separated list for platforms to include in the scan. Example: windows.
- **Platform Exclude Filter:** Comma separated list for platform to exclude from the scan. Example: windows,
- **Custom Additional Filters:** Additional filters to scan. Example: tag:Purpose=store,database;
- **Instance Name Preference:** If found on the instance, this is used for the Computer Name. Consider how the machine will be accessed with the selection. If selection is not found, it defaults to PrivateDnsName.

6. Click the **OK** button.

Google Cloud Platform Discovery

Overview

Secret Server can manage Google Cloud Platform (GCP) service accounts and VM instances. This feature allows users to run discovery to pull and manage VM Instances, as well as import and manage GCP service accounts.

Configuration

Task 1: Creating GCP Service Accounts

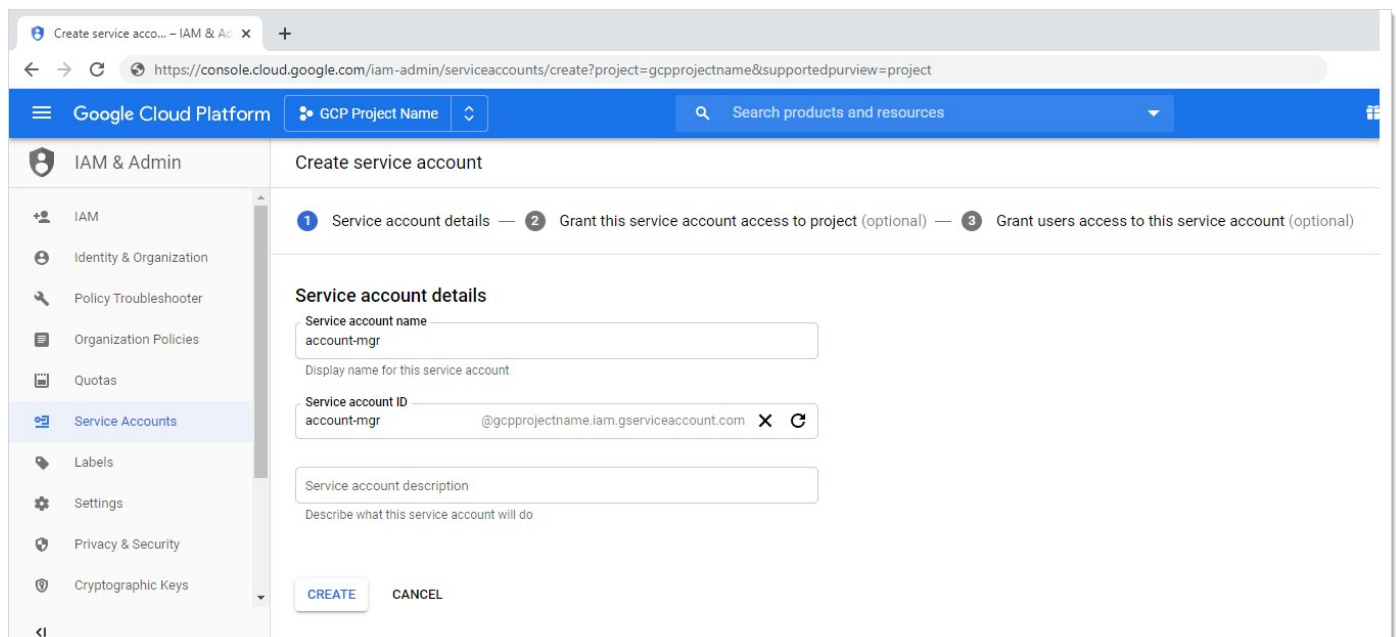
These are special accounts created in GCP to make authorized API calls for Compute Engine and other GCP applications.

Note: See [GCP Service Accounts](#) for more information.

Secret Server uses the GCP service account to make authorized API calls to GCP to pull projects, zones, instances, service accounts and service account keys.

To create the service account:

1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **Service Accounts**. A list of service accounts appears.
2. Click the **+ Create Service Account** button. The "Service account details" page of the Create Service Account wizard appears:

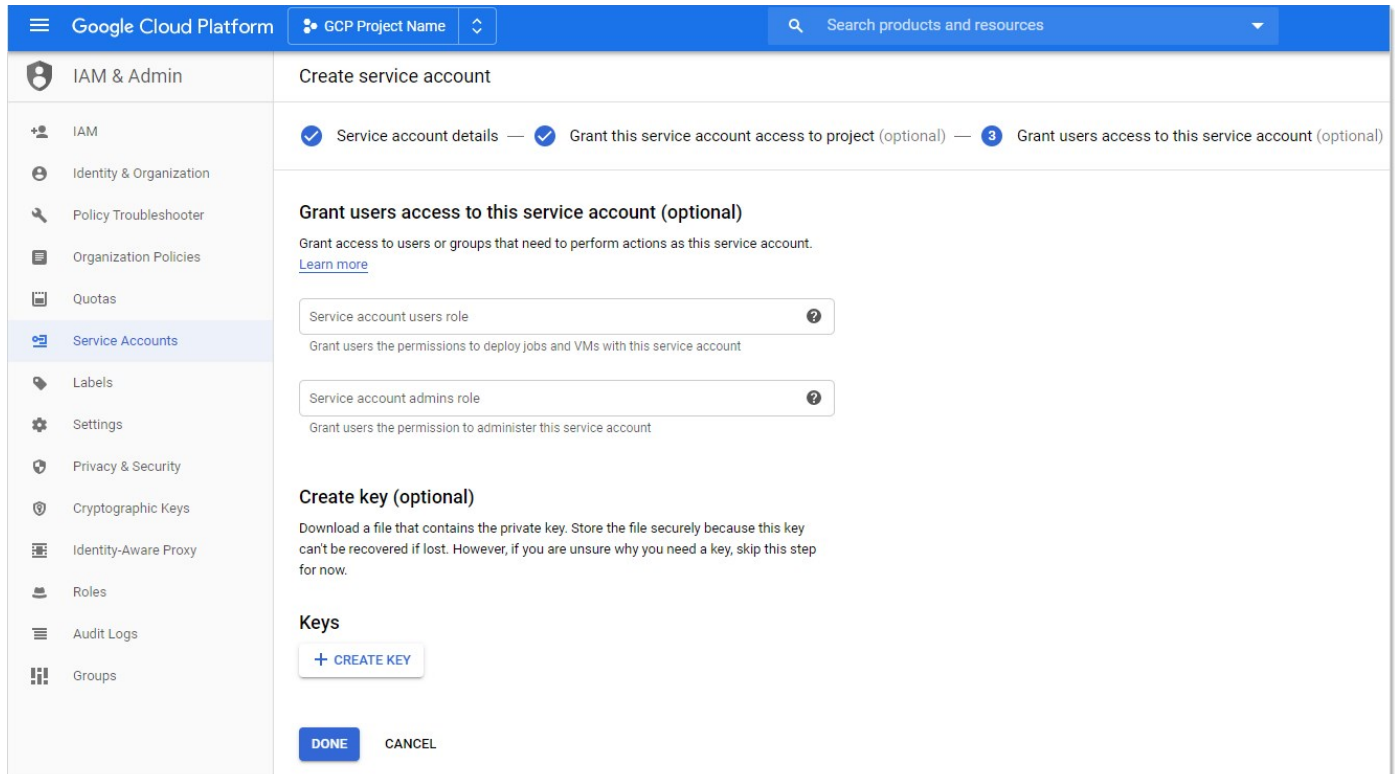


The screenshot shows the Google Cloud Platform console interface for creating a service account. The left sidebar is open to 'IAM & Admin' > 'Service Accounts'. The main content area is titled 'Create service account' and shows a three-step wizard: 1. Service account details, 2. Grant this service account access to project (optional), and 3. Grant users access to this service account (optional). The first step is active. It contains the following fields: 'Service account name' with the value 'account-mgr', 'Service account ID' with the value 'account-mgr@gcpprojectname.iam.gserviceaccount.com', and 'Service account description' which is empty. At the bottom of the form are 'CREATE' and 'CANCEL' buttons.

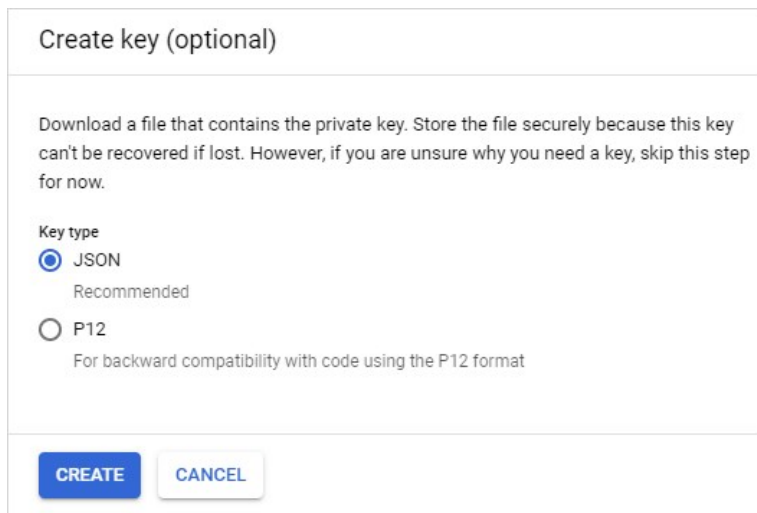
3. Type the service account name in the **Service Account Name** text box.
4. Start to type the service account ID name and select the service account in the **Service Account Name** text/list box.
5. Click the **Create** button. The "Grant this service account access to project (optional)" page appears:

The screenshot shows the Google Cloud Platform IAM & Admin console. The left sidebar contains the navigation menu with 'Service Accounts' selected. The main content area is titled 'Create service account' and shows a progress indicator with three steps: 1. Service account details (checked), 2. Grant this service account access to project (optional) (active), and 3. Grant. The 'Service account permissions (optional)' section is expanded, showing a list of roles. The first role is 'Service Account Key Admin' with the description 'Create and manage (and rotate) service account keys.' The second role is 'API Keys Admin' with the description 'Ability to create, delete, update, get and list API keys for a project.' There are 'Add condition' links and trash icons for each role. At the bottom, there are 'CONTINUE' and 'CANCEL' buttons.

6. Click the **Role** list box and select **Service Account Key Admin**.
 7. Click the **+ Add Role** button to add another role.
 8. Click the new **Role** list box and select **API Keys Admin roles**.
 9. Click the **Continue** button. The "Grant users access to this service account (optional)" page appears:
-

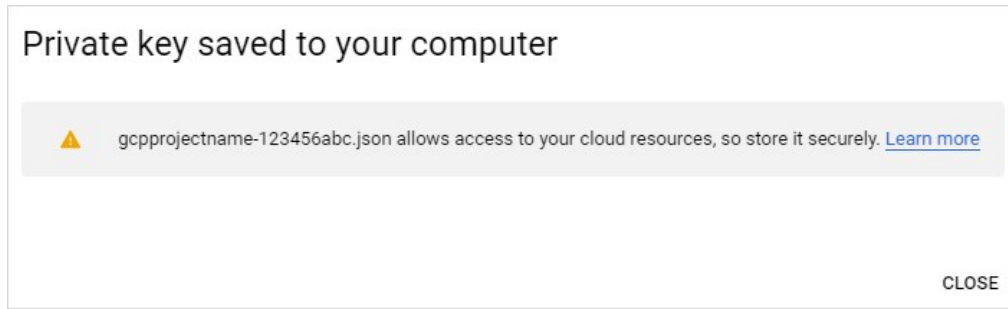


10. Click the **+ Create Key** button in the **Keys** section. The "Create key (optional)" popup appears:



11. Click to select the **JSON** selection button.

12. Click the **Create** button. This creates and downloads a JSON private key file. A confirmation popup appears:



13. Click the **Close** button in the bottom right. The service account is created, and its JSON private key is on your computer.

Note: Note where you downloaded the file. You will need it later in this instruction.

Note: For more information on this process, see [Creating and managing service accounts](#) on the GCP website.

Task 2: Setting GCP Permissions

GCP permissions are IAM permissions from the IAM & Admin section of GCP. Without the proper permissions, GCP discovery, RPC, and heartbeat may not function properly.

For the service accounts to have access to a project, you must add the service account IAM permissions in each Project. If you did not add the permissions when you created the service account, you need to add the IAM permissions in the project they were created in as well.

Discovery

To run discovery in Secret Server, the GCP service account needs the "project viewer" read only permission, which can list projects, zones, service accounts, and instances.

To add the permission In GCP:

1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **IAM**. The "Permissions for project..." page appears.
2. Click the **Add** button. The "Add member to..." page appears.
3. Type the service account email address in the **Members** text box.
4. Click the **Roles** dropdown list to select **Project > Viewer** (you can also type it).
5. Click the **Add** button. The new member appears in the table on the "Permissions for project..." page.

RPC/Heartbeat

To run RPC/Heartbeat in Secret Server, the service account needs the "service account key admin" permission, which can create, delete, and rotate service account keys.

To add the permission In GCP:

1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **IAM**. The "Permissions for project..." page appears.
2. Click the **Add** button. The "Add member to..." page appears.
3. Type the service account email address in the **Members** text box.
4. Click the **Roles** dropdown list to select **Service Account Key Admin** (you can also type it).

5. Click the **Add** button. The new member appears in the table on the "Permissions for project..." page.

Task 3: Creating a GCP IAM Service-Account Secret

Secret Server now has a build in GCP IAM Service Account Key template.

Note: To create a Secret using GCP IAM service account key template, you must have the service account's JSON private key file from GCP (created earlier).

Create a new secret (see [Creating Secrets](#) for details):

1. Click the **+** on the **Secrets** item on the main menu. The "Create New Secret" page appears:

Create New Secret

Please select a folder to synchronize. [Change](#)

Choose a Secret Template

Search for template name

- Active Directory Account
- Amazon IAM Console Password
- Amazon IAM Key
- Bank Account
- Cisco Account (SSH)
- Cisco Account (Telnet)
- Cisco Enable Secret (SSH)
- Cisco Enable Secret (Telnet)
- Cisco VPN Connection
- Combination Lock
- Contact
- Copy of CreditCard
- Credit Card
- DevOps Secret Vault Client Credentials
- Generic Discovery Credentials
- Generic ODBC (DataSource)
- Google IAM Service Account Key**
- Healthcare
- HP iLO Account (SSH)
- IRM iSeries Mainframe

Cancel Create Secret

2. Select **Google IAM Service Account Key** as the template. Another "Create New Secret" page, tailored to GCP, appears:

Create New Secret

Secret Template	Google IAM Service Account Key Change
Folder	No Folder Selected
Secret Name *	<input type="text"/>
Email *	<input type="text"/>
Private Key Id *	<input type="text"/>
JSON Private Key *	Change
Notes	<div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div>
Site	<input type="text" value="Local"/>
Auto Change Enabled	<input type="checkbox"/>

3. Click to select a folder for the new secret.
4. Type the secret's name in the **Secret Name** text box.
5. Type the service account email address (use client_email from the JSON private key file) in the **Email** text box.
6. Type the private key ID (use private_key_id from the JSON private key file) in the **Private Key ID** text box.
7. Click the **Change** button to upload the JSON private key file you created earlier.
8. Click the **Create Secret** button.

Task 4: Creating an RPC/Heartbeat Password Changer

Secret Server can check if a service Account key is valid and can rotate the Service Account key. This should work the same as any other

RPC or Heartbeat. **Note:** RPC and Heartbeat must be enabled

RPC/Heartbeat can be tested from the Password Changers page

1. In Secret Server, go to **Admin > Remote Password Changing:**

Remote Password Changing Configuration

Enable Remote Password Changing	Yes
Enable Password Changing on Check In	No
Enable Heartbeat	Yes

[Advanced \(not required\)](#)

Days to Keep Operational Logs	30
-------------------------------	----

[← Back](#) [✎ Edit](#) [✎ Configure Password Changers](#)

[⚙️ Configure Dependency Changers](#) [🏠 Distributed Engine Configuration](#)

[☰ View Audit](#)

2. Click the Configure Password Changers button. The Password Changers Configuration page appears:

Password Changers Configuration

PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (Telnet)	SSH Local Account	Yes
Cisco Enable Secret Custom (SSH)	SSH Local Account	Yes
Cisco Enable Secret Custom (Telnet)	SSH Local Account	Yes
ESX/ESXi (API)	ESXi Local Account	Yes
F5 BIG-IP Root Account (SSH)	SSH Local Account	Yes
Generic Discovery-Only Credentials	< None >	Yes
Generic ODBC (DataSource)	SQL Local Account	Yes
Google IAM Service Account Key	GCP Service Account	Yes
HP iLO Account Custom (SSH)	SSH Local Account	Yes

- Click the **Google IAM Service Account Key** link. The "Google IAM Service Account Key" page appears:

Google IAM Service Account Key

Verify Password Changed Commands [Test Action](#)

i This process is done through internal commands. The commands cannot be edited.

Password Change Commands [Test Action](#)

i This process is done through internal commands. The commands cannot be edited.

Password Change By Admin Credentials Commands [Test Action](#)

i This process is done through internal commands. The commands cannot be edited.

[Back](#) [Configure Scan Template](#) [View Audit](#)

4. Test the heartbeat: Click the **Test Action** button in the **Verify Password Changed Commands** section. The Test Action popup appears:

Test Action

Please enter the information to test the action. Note this test may take a while to complete since it will run against the actual host.

FIELDS

EMAIL: account-manager@gcpp

JSONPRIVATEKEY: {"type":

PRIVATEKEYID: 123456abcde

Site: Local

OK Cancel

5. Ensure that the **JSONPRIVATEKEY** text box is populated. The others are optional.
6. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:

Verify Password Changed Commands Test Action

Password Verify was Successful
(No console output).

7. Test RPC: Click the **Test Action** button in the **Password Change Commands** section. The Test Action popup appears:

Test Action

Please enter the information to test the action. Note this test may take a while to complete since it will run against the actual host.

Warning: This will change the password on the target account if successful.

FIELDS

EMAIL: account-manager@gcpp

JSONPRIVATEKEY: {"type":

PRIVATEKEYID: 123456abcde

Site: Local

OK Cancel

8. Ensure that the **JSONPRIVATEKEY** and **Email** text boxes are populated. The others are optional.
9. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:

Password Change Commands Test Action

Password Successfully Changed
(No console output)

10. Test RPC with admin credentials: Click the **Test Action** button in the **Password Change By Admin Credentials Commands** section. The Test Action popup appears:

Test Action ✕

i Please enter the information to test the action. Note this test may take a while to complete since it will run against the actual host.

⚠ Warning: This will change the password on the target account if successful.

FIELDS

EMAIL	<input type="text" value="service-account@gcppro"/>
JSONPRIVATEKEY	<input type="text"/>
PRIVATEKEYID	<input type="text" value="123456abcde"/>
Admin EMAIL	<input type="text" value="account-manager@gcpp"/>
Admin JSONPRIVATEKEY	<input "="" type="text" value='{"type":'/>
Admin PRIVATEKEYID	<input type="text" value="123456abcdeadmin"/>
Site	<input type="text" value="Local"/>

11. Ensure that all text boxes are populated except **JSONPRIVATEKEY**, **Admin Email**, and **Admin PRIVATEKEYID**, which are optional.
12. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:

Password Change By Admin Credentials Commands

Password Successfully Changed using Admin Credentials

(No console output).

Task 5: Creating a GCP Discovery Source

Secret Server now has a built-in GCP discovery source wizard that creates the scanners to pull the projects, zones, service accounts. To create a GCP discovery source:

1. In Secret Server, go to **Admin > Discovery**:

Admin > Discovery



Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery
 Last Started: 4 hours, 22 minutes ago
 Next Run: 19 hours, 37 minutes

Computer Scan
 Last Started: 4 hours, 50 minutes ago
 Next Run: 19 hours, 9 minutes

Discovery Network View Create Discovery Source ▾ Run Discovery Now ▾

4 Items 🔍 Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN	 
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web Services)	7/17/2020 10:50 am	
Gamma Linux	<input checked="" type="checkbox"/>	Unix	7/17/2020 10:46 am	
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	7/17/2020 10:31 am	
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	7/17/2020 10:55 am	

- Click the **Create Discovery Source** dropdown list and select **GCP (Google Platform)**. The GCP Discovery Source wizard Overview page appears:

GCP Discovery Source

[Overview](#)

Getting Started

Google Cloud Platform (GCP) Discovery allows you to use Secret Server to scan for Projects, Zones, Service Accounts, and Instances.

The Wizard will help you get GCP Discovery configured in 5 simple steps:

1. Name the Discovery Source.
2. Choose the Site used for Discovery scanning.
3. Select the Service Account Scanner.
4. Select the Instance Scanner.
5. Choose a Secret to use as credentials for Discovery scanning.

- Click the **Next** button. The Discovery Source Name page appears:

GCP Discovery Source

Overview > Discovery Source Name

Discovery Source Name
Define the Name of the new Discovery Source

i Choose a name that will help quickly identify this Discovery source in the future.

← Previous ✕ Cancel → Next

4. Type the name of the GCP discovery source in the **Discovery Source Name** text box.
5. Click the **Next** button. The Site page appears:

GCP Discovery Source

Overview > Discovery Source Name > Site

Add Site
Select the Site to be used for this Discovery Source

Local ▾

i The list contains all active Sites regardless of whether they have an active Engine.

← Previous ✕ Cancel → Next

6. Click the **Add Site** list box to select the site.
 7. Click the **Next** button. GCP Service Account Scanner page appears:
-

GCP Discovery Source

Overview > Discovery Source Name > Site > GCP Service Account Scanner

GCP Service Account Scanner
Finds Service Accounts defined in GCP

FIND ACCOUNTS

GCP Service Account Scanner

← Previous ✕ Cancel → Next

8. Click the **Next** button.

GCP Discovery Source

Overview > Discovery Source Name > Site > GCP Service Account Scanner > GCP Instance Scanner

GCP Instance Scanner
Finds machines hosted in GCP.

SCAN GCP INSTANCES

Scan GCP Instances

FIND MACHINES

GCP Windows Instance Scanner
GCP (Non-Windows) Instance Scanner

← Previous ✕ Cancel → Next

9. Click to select the **Scan GCP Instances** check box.

10. Click the check boxes for the scanners you desire. Currently, there are four discovery scanners for the GCP discovery source.

Note: In the future, we may add an Instance Local Account and a Service Account Dependency scanner.

- **GCP Project Scanner:** This is a host range scanner that scans the GCP and pulls all of the projects that the provided GCP service account secret has access to.
- **GCP Windows Instance Scanner:** This is a machine scanner that scans each project and pulls all of the GCP Windows OS VM instances.

- **GCP (Non-Windows) Instance Scanner:** This is a machine scanner that scans each project and pulls all of the GCP Non-Windows OS VM instances.
- **GCP Service Account Scanner:** This is an account scanner that scans each project and pull all of the GCP Service accounts.

11. Click the **Next** button. The Credential Secrets page appears:

GCP Discovery Source

Overview > Discovery Source Name > Site > GCP Service Account Scanner > GCP Instance Scanner > Credential Secrets

Credential Secrets

Select the GCP IAM Service Account Key Secret to be used as Authentication Credentials for Discovery.

[Add Secret](#)

If a suitable Secret is not already stored in Secret Server, create it below.

[Create New Secret](#)

i Only GCP IAM Service Account Key may be used.

← Previous
✕ Cancel
🏠 Finish

12. Click the **Add Secret** link. The Select a Secret popup appears:

Select a Secret

< All Folders >

- + Personal Folders
- + Everyone
- + RPC/Heartbeat

	SECRET	FOLDER	TEMPLATE
<input type="checkbox"/>	[blurred]	Launcher	Unix Account (SSH)
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account
<input type="checkbox"/>	[blurred]	Everyone	Windows Account

13. Navigate the folder tree and select the secret you created earlier. As soon as you select the check box, the popup disappears and the

secret appears under the Add Secret link.

14. Click the **Finish** button.

Viewing Discovery Scanners for the GCP Discovery Source

To view these scanners:

1. In Secret Server, go to **Admin > Discovery**:
2. Go to **Admin > Discovery**.
3. Click the discovery source name link in the table. The Discovery Source page for it appears.
4. Click the **Scanner Settings** button in the top right of the page. The Discovery Source Scanner Settings page appears, which lists the scanners.

Instance Custom Filter

This option is only available for the instance scanners. The Custom Filter Setting can be used to include or exclude instances using a filter expression on the name, label, or any other field allowed by GCP. The filter must:

- Be a string, number, or Boolean value
- Use these comparison operators: =, !=, >, or <
- Use parentheses () around each filter
- Combine different filters using AND or OR (all caps). For example: (name="instanceName") AND (labels.key="value")

Note: See [Method: instances.aggregatedList](#) for more on filtering instances.

Other useful filters:

Status:

status="StatusValue"

StatusValue Can be Running Or Terminated

Zone:

zone=https://www.googleapis.com/compute/v1/projects/{ProjectName}/zones/{ZoneName}

Note: Unfortunately, at this time of this topic, Google has an [open issue](#) of the tag filter not working.

Importing Service Accounts

From the Discovery Network View, Secret Server can import Service Account keys and automatically take over the account. This import process will create a new Secret for the Service Account key, delete the associated key, create a new key, and save the json private key file with the Secret, so this can be easily managed by Secret Server.

To Import a Service Account

1. Go to **Admin > Discovery**.
2. Click the **Discovery Network View** button. The Discovery Network View page appears.
3. Select the **Domain\Cloud Account** tab

- Click to select the Service Account (s) to import in the unlabeled Domain/Cloud tree on the left.
- Click the **Import** button. The importation wizard begins:

Bulk Operation: Import Accounts ✕

The account(s) will be imported as 1 new Secret(s).

[Secret](#) > [Key](#) > [Import Key](#) > [Key Rotation](#) >

Scan Template GCP Service Account

Secret Type

Folder [GCP\Imports](#)

Secret Name *

Site

[← Previous](#) [Next →](#)

- For secrets:
 - Click the **Secret Type** dropdown list and select **Google IAM Service Account Key**.
 - Click the link after **Folder** to select a folder.
 - Type a name in the **Secret Name** text box (It auto fills \$EMAIL).
 - Click the Site dropdown list to select a site.
- Click the **Next** button. The Key page appears:

Bulk Operation: Import Accounts ✕

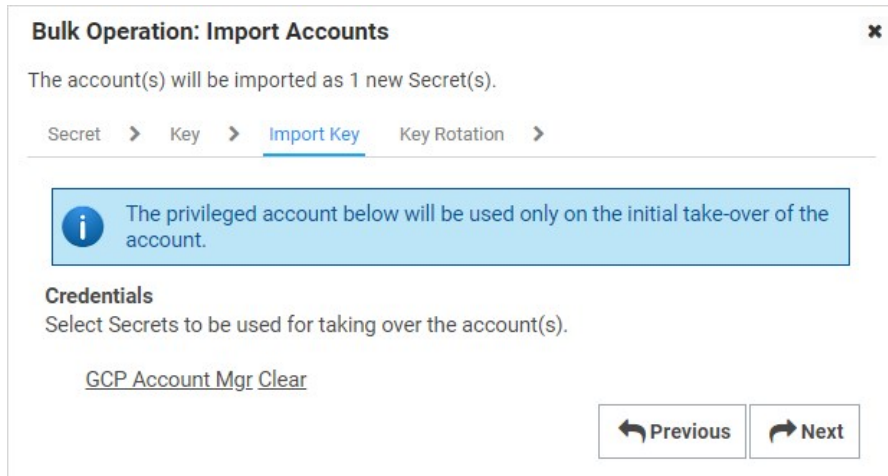
The account(s) will be imported as 1 new Secret(s).

[Secret](#) > [Key](#) > [Import Key](#) > [Key Rotation](#) >

i This will take over the selected account(s) and create a new key for each.

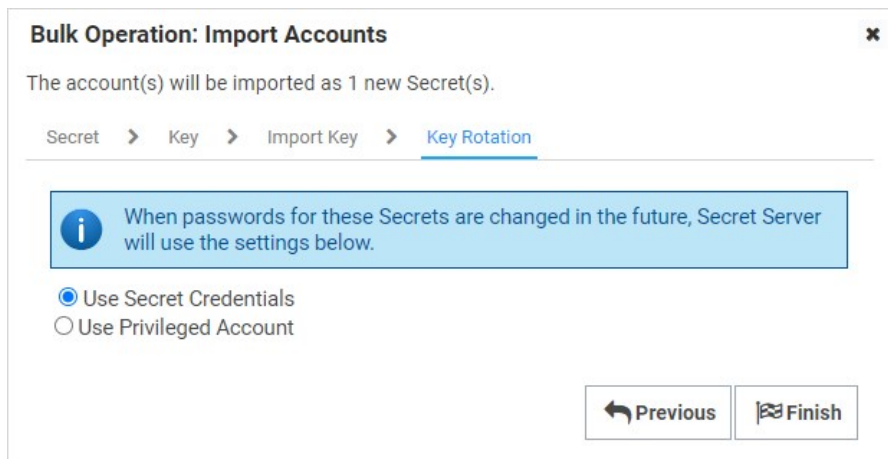
[← Previous](#) [Next →](#)

- When importing GCP service account keys, the only option is take over the account. Meaning, Secret Server triggers a remote password change on import to rotate the imported key and obtain a new JSON private key file. With the JSON private key file, Secret Server can then manage the GCP service account.
- Click the **Next** button. The Import Key page appears:



10. Click the link to select a secret to use for the initial take over of the account.

11. Click the **Next** button. The Key Rotation page appears:



12. For key rotation, click one of two selection button options to choose a secret for future key rotations. Either option would need the permissions mentioned above. When the password for the chosen secret are changed in the future, Secret Server will use one of these two options:

- **Use Secret Credentials:** Use the imported service account to rotate itself, and it has permissions to rotate keys.
- **Use Privileged Account:** Use another service account that has permissions to rotate keys

13. Click the **Finish** button.

GCP APIs

Overview

To make API calls to GCP, you need to enable the following APIs to use GCP discovery in Secret Server. More information can be found on the [GCP Getting Started](#) page. The APIs are:

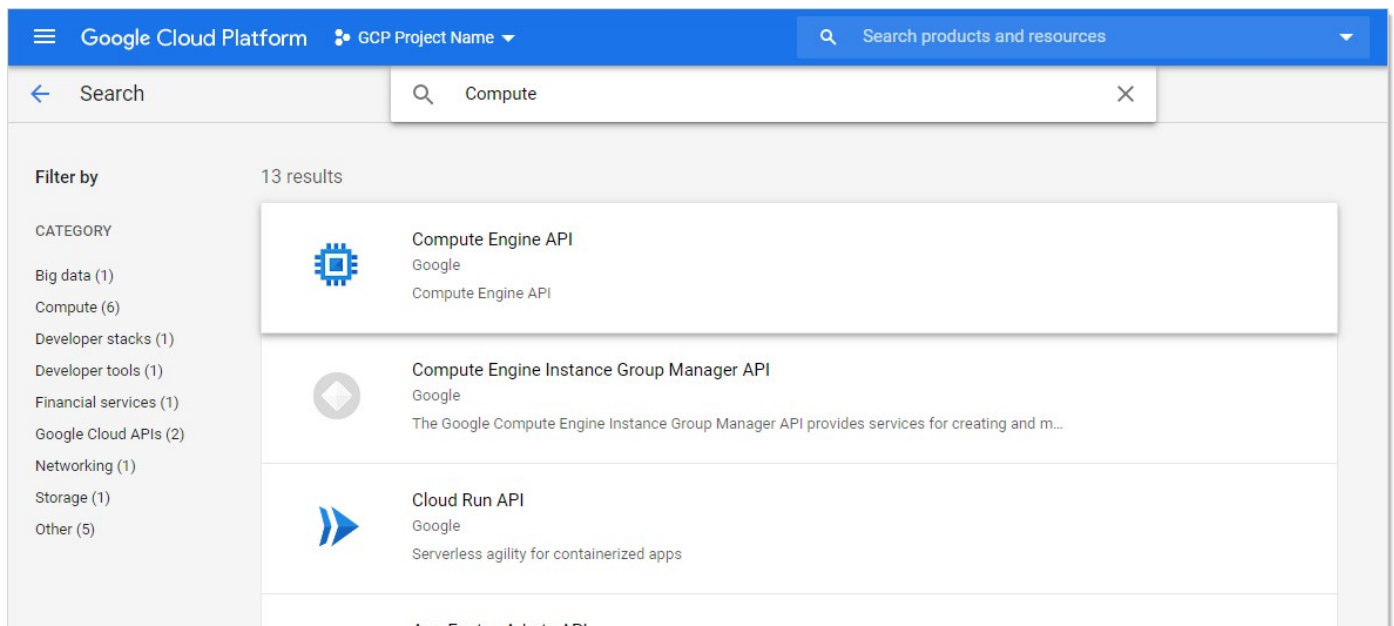
- **Cloud Resource Manager API:** Used for managing GCP resource containers, such as Projects.

- **Compute Engine API:** Used for managing GCP instances (virtual machines).
- **Identity and Access Management (IAM) API:** Used for managing identity and access control for GCP resources, such as service accounts.

Enabling GCP APIs

In GCP:

1. In GCP, click the **APIs & Services** menu item and select **Library**. The Library page appears.
2. Type the name of the API in the Search text box and press **<Enter>**. Matching APIs appear:



3. Click the button for the desired API. That API's page appears:

The screenshot shows the Google Cloud Platform console interface. At the top, there is a blue navigation bar with a hamburger menu icon, the text 'Google Cloud Platform', a dropdown menu for 'GCP Project Name', and a search bar with the placeholder text 'Search products and resources'. Below the navigation bar is a white header area with a back arrow icon on the left. The main content area features the 'Compute Engine API' logo (a blue square with a grid pattern) and the text 'Compute Engine API' and 'Google'. There are two buttons: a blue 'ENABLE' button and a white 'TRY THIS API' button with an external link icon. Below these buttons is a grey button that says 'Click to enable this API'. Underneath the buttons are three tabs: 'OVERVIEW' (which is selected and underlined), 'DOCUMENTATION', and 'SUPPORT'. The 'Overview' section contains the following text: 'Creates and runs virtual machines on Google Cloud Platform.' Below this is the 'About Google' section, which states: 'Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.' To the right of the 'About Google' section is the 'Additional details' section, which lists: 'Type: [APIs & services](#)', 'Last updated: 12/9/19', 'Category: [Compute](#), [Networking](#)', and 'Service name: compute.googleapis.com'.

4. Click the **Enable** button.

Errors and Solutions

Create Keys Failed: Access Denied

Error

Create Keys Failed: AccessDenied, Google.Apis.Requests.RequestError Permission iam.serviceAccountKeys.create is required to perform this operation on service account projects/-/serviceAccounts/discovery-me@gcpprojectname.iam.gserviceaccount.com. [403] Errors [Message[Permission iam.serviceAccountKeys.create is required to perform this operation on service account projects/-/serviceAccounts/discovery-me@gcpprojectname.iam.gserviceaccount.com.] Location[-] Reason[forbidden] Domain[global]]

Likely Cause

The service account used to rotate the key does not have necessary permission to perform this task.

Solution

1. Go to the GCP console.
2. Select **IAM > Permissions**.
3. Select the service account.

4. Add the **Service Account Key Admin** permission.
5. Once the service account has permission:
 1. In Secret Server, select the secret to rotate.
 2. Stop the current rotation.
 3. Try the operation again.

Create Keys Failed: Maximum Number of Keys on Account Reached

Error

Create Keys Failed: ArgumentError, Google.Apis.Requests.RequestError Maximum number of keys on account reached. [429] Errors [Message[Maximum number of keys on account reached.] Location[-] Reason[rateLimitExceeded] Domain[global]]

Likely Cause

The rotated service account has reached the maximum number of keys allowed. GCP maximum is 10 keys.

Solution

1. Go to the GCP console.
2. Select **IAM > Permissions**.
3. Remove the unused keys.
4. Once the service account has less than 10 keys, in Secret Server:
 1. In SS, select the secret to rotate.
 2. Stop the current rotation.
 3. Try the operation again.

Discovery Consumer: Syncing OUs Failed

Error

DiscoveryConsumer: Synchronizing Organizational Units failed for [Our Google Cloud]! Error: An issue was encountered during the scan. Google.Apis.Requests.RequestError Access Not Configured. Compute Engine API has not been used in project 123456 before or it is disabled. Enable it by visiting <https://console.developers.google.com/apis/api/compute.googleapis.com/overview?project=123456> then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry. [403] Errors [Message[Access Not Configured. Compute Engine API has not been used in project 123456 before or it is disabled. Enable it by visiting <https://console.developers.google.com/apis/api/compute.googleapis.com/overview?project=123456> then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.] Location[-] Reason[accessNotConfigured] Domain[usageLimits]] , -2146233088

Likely Cause

The discovery service account used for has access to a GCP project that has not been set up or is disabled.

Solution

1. Go to GCP console.
2. Go to **Compute Engine > VM Instances**.
3. Set up the compute engine

Note: This requires billing information.

Discovery Consumer: Syncing Machines Failed

Error

DiscoveryConsumer: Synchronizing Machines failed for [GCP Discovery Source]! Error: An issue was encountered during the scan. Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [Message[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.] Location[-] Reason[invalid] Domain[global]] , -2146233088 Exception Caught: Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [Message[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.] Location[-] Reason[invalid] Domain[global]] Attempting GCP scan for Instances Parameters are valid. Checking for permissions to list Projects.. Has permissions to list Projects.. Starting scan..

Likely Cause

The instance scanner custom filter is not valid.

Solution

1. In Secret Server, go to the GCP discovery source.
2. Edit the instance scanner.
3. Update the "custom filter" setting.

Note: See [Method: instances.aggregatedList](#) for more on filtering instances.

Discovery Consumer: Machine Scan Completed but Computers Failed Authentication

Error

DiscoveryConsumer: Synchronizing Machines failed for [GCP Discovery Source]! Error: An issue was encountered during the scan. Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [Message[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.] Location[-] Reason[invalid] Domain[global]] , -2146233088 Exception Caught: Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [Message[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.] Location[-] Reason[invalid] Domain[global]] Attempting GCP scan for Instances Parameters are valid. Checking for permissions to list Projects.. Has permissions to list Projects.. Starting scan..

Likely Cause

The instance scanner custom filter is not valid.

Solution

1. In Secret Server, go to the GCP discovery source.
2. Edit the instance scanner.
3. Update the "custom filter" setting.

Note: See [Method: instances.aggregatedList](#) for more on filtering instances.

Invalid Grant: Account Not Found

Error

An issue was encountered during the scan. Error:"invalid_grant", Description:"Invalid grant: account not found", Uri:"", -2146233088

Likely Cause

The service account does not exist in GCP. There may be a typo or it was deleted.

Solution

1. Go to GCP console.
2. Create a service account to use. See [Task 1: Creating GCP Service Accounts](#).

Request Error: Caller Does Not Have Permission

Error

An issue was encountered during the scan. Google.Apis.Requests.RequestError The caller does not have permission [403] Errors [Message[The caller does not have permission] Location[-] Reason[forbidden] Domain[global]], -2146233088

Likely Cause

The service account does not have permissions in IAM.

Solution

1. Go to GCP console.
2. Select IAM.
3. Click the **Service Account** menu item to create a service account with the desired permissions. See [Task 1: Creating GCP Service Accounts](#) and [Task 2: Setting GCP Permissions \(#Task-2:-Setting-GCP Permissions\)](#).

Unix Account Discovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Unix account discovery follows these steps:

1. During configuration, Secret Server is given a list of IP address ranges and ports on the network to scan for. See [Creating a Unix Discovery Source](#)
2. Within that range, discovery searches for computers listening on the specified ports (default is 22). The ports and other parameters are configurable via the scanners belonging to the discover source. See [Discovery Sources, Scanners, and Templates](#)
3. Secret Server then attempts to use DNS to resolve the found IPs to discover the associated computer name.
4. Secret Server saves all the collected information to its database.
5. Secret Server then attempts to connect to each computer using the provided credentials and query for a list of user accounts on the target system.

Creating a Unix Discovery Source

Discovery sources define a set of discovery operations. You must create one based on the built-in types prior to running discovery. To do so for Unix:

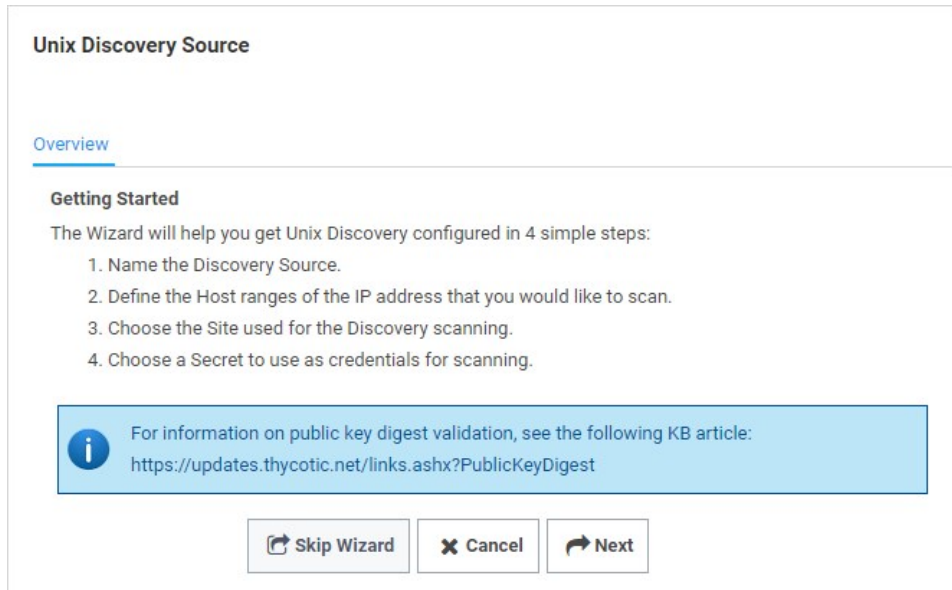
Creating the Discovery Source

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

The screenshot shows the 'Admin > Discovery' page. At the top, there are navigation tabs: 'Discovery Sources' (selected), 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there are three buttons: 'Discovery Network View', 'Create Discovery Source' (with a dropdown arrow), and 'Run Discovery Now' (with a dropdown arrow). Below these buttons, there are two summary cards: 'Discovery' and 'Computer Scan', both showing 'Last Started: 2 months, 8 days ago' and 'Next Run: soon'. Below the summary cards, there is a search bar with '4 Items' and a search icon, and a toggle for 'Include Inactive'. Below the search bar is a table with the following data:

NAME	ACTIVE	TYPE	SOURCE LAST RUN	
Test_Esxi	✓	PowerShell	11/18/2020 03:32 ...	
gamma.thycotic.com	✓	Active Directory	11/19/2020 03:45 ...	
Gamma Linux	✓	Unix	11/18/2020 03:32 ...	
AWS Discovery	✓	AWS (Amazon Web...	11/18/2020 03:32 ...	

2. Note the list of existing discovery sources.
3. Click the **Create Discovery Source** button and select **Unix** to choose that discovery source type. A Discovery Source page appears for that type:



4. The page briefly summarizes what a Unix discovery Source is. The Unix setup does not allow you to skip the creation wizard. A Unix discovery source created with the wizard has the command set, ports, maximum TCP connections, and parse format settings assigned default values. You can change these by editing the discovery source scanners after finishing the wizard.
5. Click the **Next** button to continue. The Discovery Source Name wizard page appears:



6. Type an identifying, human-readable name in the **Discovery Source Name** text box.
7. Click the **Next** button. The Scan Range page of the wizard appears:

Unix Discovery Source

Overview > Discovery Source Name > Scan Range

IP Scan Range

The IP Address Range to scan for Unix computers.

Examples:

192.168.40.1/24

192.168.40.123-249

192.168.40.1-192.168.60.255

i Advanced Settings that allow customization of how Unix Discovery works are available after the Wizard is complete.

← Previous
✕ Cancel
↩ Next

8. Type the desired range in the **IP Scan Range** text box using one of the listed formats. Multiple entries should each be on their own line. Host name entries are also allowed. The more precisely you specify the ranges you wish to discover Unix machines on, the faster the discovery scan will run.
9. Click the **Next** button. The Add Site wizard page appears:

Unix Discovery Source

Overview > Discovery Source Name > Scan Range > Site

Add Site

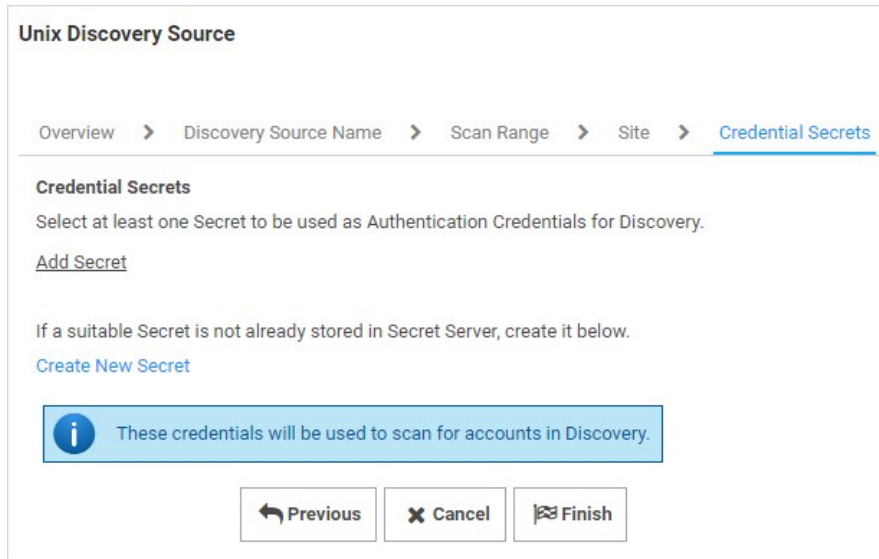
Select the Site to be used for this Discovery Source

Local ▼

i The list contains all active Sites regardless of whether they have an active Engine.

← Previous
✕ Cancel
↩ Next

10. Click the **Add Site** dropdown list to select the desired site for the discovery source. If distributed engines are setup, the list shows all active sites. If no distributed Engines are setup, the list defaults to local, and you cannot change it.
11. Click the **Next** button. The Credential Secrets wizard page appears:



12. **Either** click the **Add Secret** link to search for and click the secret you want to use for the account credentials during the scan. The popup page closes, and the selected secret appears:



Or create a new secret for the credentials:

1. Click the **Create New Secret** link. The New (secret) popup page appears:

New

i This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

General

Secret Template	Generic Discovery Credentials ▼
Secret Name *	
Username *	
Password 🔒	<input type="password"/> <input type="button" value="* Generate"/>
Notes	
Private Key	<input type="button" value="Choose File"/> No file chosen <input type="checkbox"/> Generate New SSH Key
Private Key Passphrase 🔒	<input type="password"/> <input type="button" value="* Generate"/>
Folder	<input type="button" value="Folder icon"/> \Personal Folders\Will Sprunk Clear
Inherit Secret Policy	<input checked="" type="checkbox"/>
Secret Policy	< No Policy >
Site	Local ▼

2. Click the **Secret Template** dropdown list and select **Generic Discovery Credentials** secret template.
3. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.
4. Click the **Save and Add New** button. The popup page disappears.
13. Click the **Add Secret** link to add any additional secret credentials. When using multiple credentials, discovery goes through the list of secrets attempting each credential until it either has a successful authentication or has run out of provided accounts. This loop is done for each computer.
14. Click the **Finish** button to complete the wizard. You are returned to the Discovery Sources page where you see the new Unix discovery source.
15. Proceed to the next section to further customize the discovery source scanners.

Editing the Unix Discovery Source Scanners

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

The screenshot shows the 'Admin > Discovery' page. At the top, there are navigation tabs: 'Discovery Sources' (selected), 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there are three buttons: 'Discovery Network View', 'Create Discovery Source', and 'Run Discovery Now'. Underneath, there are two summary cards: 'Discovery' and 'Computer Scan', both showing 'Last Started: 2 months, 8 days ago' and 'Next Run: soon'. A table below shows a list of 4 items with columns for NAME, ACTIVE, TYPE, and SOURCE LAST RUN. The 'Include Inactive' toggle is turned off.

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 ...
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 ...
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web...	11/18/2020 03:32 ...

2. In the list of existing discovery sources, click the name of the one you want to edit. The Unix discovery source's page appears, in this case, Gamma Linux.

The screenshot shows the 'Admin > Discovery > Gamma Linux' page. At the top, there are navigation tabs: 'Discovery Source' (selected) and 'Audit'. A 'Scanner Settings' button is visible. Below the tabs, there is a section for 'Unix' with an 'Edit' link. The page contains a detailed description of the Unix discovery source and a configuration table.

<p>The default command sets that Secret Server ships with will work to discover machines and accounts in most Unix environments.</p> <p>By default, the Find Non-Daemon Users (Basic Unix) command set will be used. If the built-in account should be discovered, the Discovery Source must be updated to use the Find All Users (Basic Unix) command set. New command sets can be created by clicking Configure Command Sets when on the Discovery Sources list page KB Link</p>	Discovery Source Name *	Gamma Linux
	Active *	Yes
	Discovery Site *	Local
	Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended)

3. Click the **Scanner Settings** button. The Discovery Source Scanner Settings page appears:

Discovery Source Scanner Settings

FIND HOST RANGES

+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Manual Host Range	Discovery Source	Host Range	

Scanners: 1

FIND MACHINES

+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Unix Machine	Host Range	Computer	

Scanners: 1

FIND ACCOUNTS

+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Unix Non-Daemon User	Computer	SSH Local Account	

Scanners: 1

FIND DEPENDENCIES

+ Add New Dependency Scanner

Add a dependency scanner once you have a machine scanner added to the discovery source.

4. A summary of the scanner, inputs, and outputs is as follows:

Table: Unix Scanner Summary

Scanner	Input Template	Output Template				
Manual Host Range (Find Host Ranges)	Discovery Source	Host Range				Manual Host Range
Unix Machine (Find Machines)	Host Range	Computer				Unix User (Find Accounts)
Unix User (Find Accounts)	Computer	SSH Local Account				SSH Public Key Scanner (Find Accounts)
SSH Public Key Scanner (Find Accounts)	Computer	SSH Public Key				None (Find Dependencies)
None (Find Dependencies)	None	None				Unexpected Link Text Notice that there is no dependency scanner defined, and it was not an option in the discovery source wizard used to create this scanner set, so if you want to discover dependencies, you <i>must</i> manually edit the scanner set.

Note: If no dependency scanners are available with an input template matching an output scan template from the previous step that has not already been used by another scanner in this step, you cannot add a dependency scanner. The output template must be unique for each scanner but the input template may be shared.

- Manual Host Range is the first scanner of the discovery source, and it is located in the Find Host Ranges section. The input template for that scanner is Discovery Source. This means the initial information comes from info you entered into the discovery source when you created it. Similarly, the output template is Host Range as you would expect.
- Click the pencil icon. The Settings - Manual Host Range page appears:

Settings - Manual Host Range [x]

SECRET CREDENTIALS

[Add Secret](#)
[Add Secret Search Filter](#) ⓘ Create Secret Search Filter

ADVANCED SETTINGS

Lines

10.60.12.1/24 ⓘ
10.60.19.1/24

✓ OK ✕ Cancel

- Note the following:
 - The host IP ranges and credential secret could have been filled in when creating the discovery source. In this particular case, no credential secret was linked to the discovery source.
 - The Unix discovery source finds all machines and local accounts on a set of manually defined host ranges for Unix machines accessible with SSH.
 - The **Lines** text box may input multiple IP address ranges but not overlapping IP address ranges on the same discovery source. There should be one IP address range per line in the input text box.
- The next scanner, Unix Machine, is the consumer of the Computer output template, has the following configurations available:

9. Note the following:

- Each machine is scanned using SSH and the settings defined in the scanner. To obtain more information from the machine scan, use the default custom commands and authentication and the scanner can return the OS of the Unix machine.
- The **Secrets Credentials** may be generic discovery Credentials or a Unix Account (SSH) secret. You can add multiple accounts when editing the scanner. The secret should contain a host name instead of an IP address to minimize potential problems with Secret Server or the machines associated with that account.
- The **Command Set** contains customizable Unix command sets sent over an SSH connection that are used to gather information from them machine when it is scanned. The command set is defined on the discovery scanner. To change a command set, you must create a new machine scanner.

Note: Click the Configure Command Sets button on the Discovery Network View page to view a list of all of the custom command sets that are available for discovery. You can select any existing command set to edit, or you can create a new one. When you create a new command set, you must give it a name and save it before you are able to enter commands.

- Commands are only run on machines when authentication is enabled and a credential secret is added to the Find Machine settings.
- The default command set is Find Machine (Basic Unix). This command set returns the OS.
- The **Ports** text box contains a comma-delimited list of port values (1-65535). SSH generally uses port 22.
- The **Max TCP Connections** text box limits the concurrent threads used for scanning your network.
- The **Attempt Authentication** check box must be selected to run commands on the machines being scanned. The credentials supplied by the secret will be used to access each machine during the scan. If the credentials are correct, the custom commands are run to extract the OS information from the machine.

10. The next scanner, Unix Non-Daemon User, has the following configurations available:

Settings - Unix Non-Daemon User

SECRET CREDENTIALS

1. gamma root
[Add Secret](#)
[Add Secret Search Filter](#) Create Secret Search Filter

COMMAND SET

Find Non-Daemon Users (Basic Unix)

ADVANCED SETTINGS

Port: 22

User Regex Format: ^\s*([^\s:]+):([^\s:]*):([^\s:]*):{

Parse Format: Username, Password, U:

Newline Separator Character: New Line (\n)

OK Cancel

11. Note the following:

- The **Secret Credentials** secret is the same one used for the Unix Machine scanner, but it is possible to use a different one.
- As earlier, the **Ports** text box is a comma-separated list of port values (1-65535). SSH generally uses port 22. The default port used when attempting to scan a machine for users. This may be overridden by a specific port found during machine scanning.
- The **User Regex Format** text box contains a regular expression that finds the lines of text received during the scan that are valid for user parsing. The matched groups in the regular expression should correspond to the comma-separated items in the parse format.
- The **Parse Format** text box defines the order of values retrieved during a scan. If the parse names match the fields defined in the imported secret, the values will be populated from the data collected on the scan.
- The **Newline Separator Character** text box defines the character that divides the lines in the output received during a scan.

12. The next scanner, SSH Public Key Scanner, has the following configurations available:

Settings - SSH Public Key Scanner ✕

SECRET CREDENTIALS

1. gamma root

[Add Secret](#)

[Add Secret Search Filter](#) Create Secret Search Filter

COMMAND SET

Find SSH Public Keys (Basic Unix)

ADVANCED SETTINGS

Port	22	
User Regex Format	^###\s*([\s]+)##([\s]+)	
Parse Format	Directory, Username, File	
Newline Separator Character	New Line (\n)	

✔ OK
✕ Cancel

13. Note the following:

- The Secret Credentials secret is the same one used for the Unix Machine scanner, but it is possible to use a different one. To discover user SSH public keys, the secret user should have sudo or su permissions.
- As earlier, the Ports text box is a comma-separated list of port values (1-65535). SSH generally uses port 22. The default port used when attempting to scan a machine for users. This may be overridden by a specific port found during machine scanning.
- The User Regex Format text box contains a regular expression that finds the lines of text received during the scan that are valid for SSH public key parsing. The matched groups in the regular expression should correspond to the comma-separated items in the parse format.
- The Parse Format text box defines the order of values retrieved during a scan. If the parse names match the fields defined in the imported secret, the values will be populated from the data collected on the scan.
- The Newline Separator Character text box defines the character that divides the lines in the output received during a scan.

Discovering SSH Public Keys

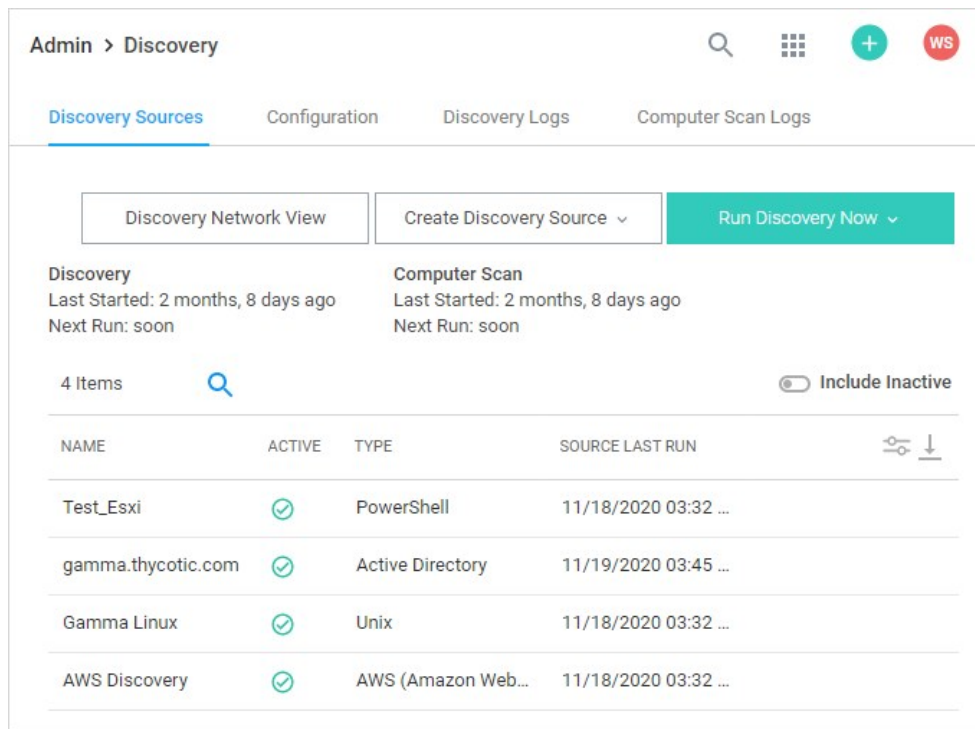
Secret Server can scan for SSH public keys on Unix machines. You can add this ability in the scanner settings section of Unix Account Discovery.

Note: This instruction assumes you already created a Unix account discovery source. See [Creating a Unix Discovery Source](#).

Task 1: Viewing Discovery Scanners for the Unix Discovery Source

To view the scanners:

1. In Secret Server, click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:



The screenshot shows the 'Admin > Discovery' page in Secret Server. The 'Discovery Sources' tab is selected. The page displays two summary cards: 'Discovery' and 'Computer Scan', both showing 'Last Started: 2 months, 8 days ago' and 'Next Run: soon'. Below these is a table of discovery sources with 4 items. The table has columns for NAME, ACTIVE, TYPE, and SOURCE LAST RUN. The 'Include Inactive' toggle is turned off. The table lists four sources: Test_Esxi (PowerShell), gamma.thycotic.com (Active Directory), Gamma Linux (Unix), and AWS Discovery (AWS (Amazon Web...)).

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	✓	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	✓	Active Directory	11/19/2020 03:45 ...
Gamma Linux	✓	Unix	11/18/2020 03:32 ...
AWS Discovery	✓	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Click the discovery source name link in the table for your new Unix discovery source. The Discovery Source page for it appears:

Discovery Source Audit **Scanner Settings**

Unix [Edit](#)

The default command sets will work to discover machines and accounts in most Unix environments.

By default, the Find Non-Daemon Users (Basic Unix) command set will be used. If the built-in account should be discovered, the Discovery Source must be updated to use the Find All Users (Basic Unix) command set. New command sets can be created by clicking Configure Command Sets when on the Discovery Sources list page
[KB Link](#)



Discovery Source Name *	IAMBugTest
Active *	Yes
Discovery Site *	Local
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended)

3. Click the **Scanner Settings** button in the top right of the page. The Discovery Source Scanner Settings page appears, which lists the scanners:

Discovery Source Scanner Settings

FIND HOST RANGES



+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Manual Host Range	Discovery Source	Host Range	 

Scanners: 1

FIND MACHINES







+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Unix Machine	Host Range	Computer	 

Scanners: 1

FIND ACCOUNTS


+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Unix Non-Daemon User	Computer	SSH Local Account	  
SSH Public Key Scanner	Computer	SSH Public Key	  

Scanners: 2

FIND DEPENDENCIES

+ Add New Dependency Scanner

 Add a dependency scanner once you have a machine scanner added to the discovery source.

Task 2: Adding the SSH Public Key Scanner for the Unix Discovery Source

1. In the **Find Accounts** section, click **Add New Account Scanner**. The Available Scanners page appears:

Available Scanners ✕				
SELECT	NAME	BASE SCANNER	INPUT TEMPLATE	OUTPUT TEMPLATE
<input type="checkbox"/>	ESXi Local Account	ESX Discovery	Computer	ESXi Local Account
<input type="checkbox"/>	File Load Local Account	File Load Discovery	Computer	Account (Basic)
<input type="checkbox"/>	SSH Public Key Scanner	SSH Discovery	Computer	SSH Public Key

2. Click the **SSH Public Key Scanner**.
3. Click the **Add Secret** link and choose secret(s) that have Unix sudo or su permissions for the host range selected in the discovery source. These permissions are necessary to navigate each user's home directory on a machine in search of SSH public key entries in the user's <user home directory>/.ssh/authorized_keys file.

Task 3: Importing SSH Public Keys

From the Discovery Network View, Secret Server can import SSH public keys and potentially take over the account. The import process creates a new secret for the SSH public key in one of two ways:

- Including a provided matching SSH private key and passphrase.
- Taking over the key by creating a new key and saving the private key file and passphrase with the secret. This can be easily managed by Secret Server.

To Import an SSH public key or keys:

1. Go to **Admin > Discovery**:

Admin > Discovery

Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery Network View Create Discovery Source Run Discovery Now

Discovery
Last Started: 2 months, 8 days ago
Next Run: soon

Computer Scan
Last Started: 2 months, 8 days ago
Next Run: soon

4 Items Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 ...
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 ...
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Click the **Discovery Network View** button. The Discovery Network View page appears.

Discovery Network View

Explain

Local Accounts Public Keys Service Accounts Domain \ Cloud Accounts

Unix Discovery Source
192.168.60.0-192.168.60.255

Advanced

Search By: All

Account Status: All

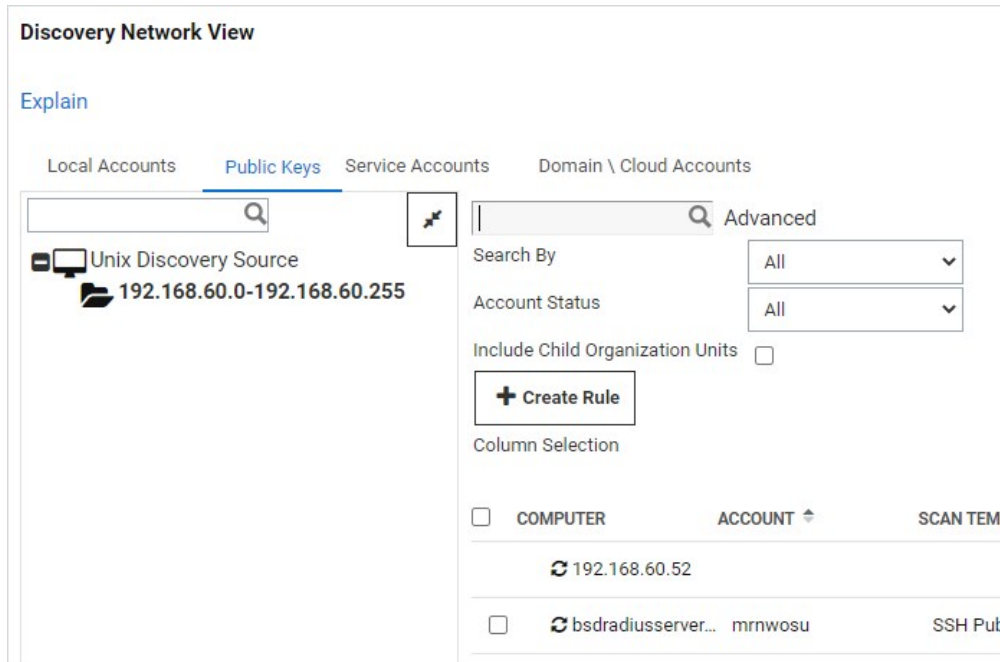
Include Child Organization Units:

+ Create Rule

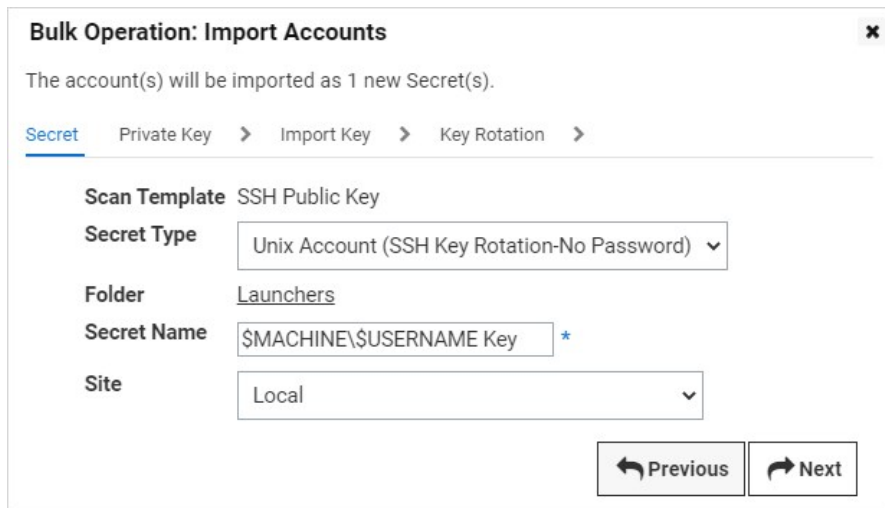
Column Selection

<input type="checkbox"/>	COMPUTER	ACCOUNT	SCAN TEMPLATE	OS	CONTAINER	SECRET
<input checked="" type="checkbox"/>	192.168.60.52			Linux	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	root	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	toor	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	uucp	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	

3. Click the **Public Keys** tab.



4. Click to select the public keys to import.
5. Click the **Import** button. The importation wizard begins:



6. Click the **Secret Type** dropdown list and select either **Unix Account (SSH Key Rotation - No Password)** or **Unix Account (Privileged Account SSH Key Rotation - No Password)**.
7. Click the **Folder** link to select a folder.
8. Type a name in the **Secret Name** text box. (It auto fills \$MACHINE\USERNAME Key).
9. Click the **Site** dropdown list to select a site.
10. Click the **Next** button. The Private Key page appears:

Bulk Operation: Import Accounts ✕

The account(s) will be imported as 1 new Secret(s).

Secret > Private Key > Import Key > Key Rotation >

I have the matching private key.
 I want to change the public SSH key on the Account.

i This will save the provided private key for the selected account(s).

← Previous
Next →

11. Click to select a selection button:

- Choose **I have the matching private key** if you want to upload a private key and passphrase known to match the public key discovered. There is a test button on the next page to verify the match.
- Choose **I want to change the public SSH key on the Account** if you wish to take over the discovered public key.

12. Click the **Next** button.

- If you chose **I have the matching private key**, the Import Key wizard page appears:

Bulk Operation: Import Accounts ✕

The account(s) will be imported as 1 new Secret(s).

Secret > Private Key > Import Key > Key Rotation >

Current private key Choose File

Current passphrase 🔒

Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC3/DJbP8Eae
gK0JPDGmtDy9zg332XGwYj2hhEnJ0Q22aaLcHdVs5sE
ZV0qixwwumlxWM/b3gYmsY0j7qwbs1BbRa06wuoi72y
V/7htq3Ekg69+WpDXjnQa63NvOSuGH+Kbg3UHqSZuH1
XntiW8pvDKa4WRcOzQ5W7laAaF9+b+T2kPRqA1B1dKg
g30ry+y4wgxtOwoAWFi8YRx+GEFe5uKrGsuasFQWT1ww
R7zE6WXredmdqkx0suYvCf0JOJ29+DlIjnyppkgU4CyxR
dBp6vezliDh5c+7HNMr/otSy6DbQCdUEY14oxBlIFyMwst
Pw6vKjkHeka1HNwF+rQxLiIV
```

Test SSH key pair match with discovered public key

← Previous
Next →

The public key chosen to import will be displayed, along with a **Choose File** button and **Current Passphrase** field. Test the key match with the public key by clicking the **Test** button below.

Note: You cannot import this public key if the key pair is not a match. If the private key or passphrase is unknown, use

takeover to import this public key.

Note: You may import multiple public keys this way if they are all identical by selecting multiple checkboxes in the Admin Network view page.

Click the **Next** button. You may add a unix sudo or su secret here for future password changing.

- If you chose the **I want to change the public SSH key on the Account** selection button, the selected secrets are all taken over and each given a random new SSH key.

13. Click the **Next** button. The Initial Takeover page appears:

Bulk Operation: Import Accounts

The account(s) will be imported as 1 new Secret(s).

Secret > Private Key > Initial Takeover > Key Rotation >

i The commands for the selected password type below will be run only on the initial take-over of the account.

Credentials
Select Secrets to be used for taking over the account(s).
ubuntu\toor Remove

[Add Secret](#)

[← Previous](#) [Next →](#)

14. Click the **Add Secret** link to choose a Unix sudo or su secret to take over the public key on the account. This removes the public key from the user's authorized keys file and adds a new random SSH key.

15. Click the **Next** button, the **Key Rotation** page appears:

Bulk Operation: Import Accounts

The account(s) will be imported as 1 new Secret(s).

Secret > Private Key > Import Key > Key Rotation

i When passwords for these Secrets are changed in the future, Secret Server will use the settings below.

Secret Type Unix Account (SSH Key Rotation-No Password)

Credentials
Select Secrets to be used for future password changing.
[Add Secret](#)

[← Previous](#) [Finish](#)

16. Click the **Add Secret** link to choose a Unix sudo or su secret for future key rotations.

17. Click the **Finish** button to complete the dialog and import the selected secrets.

Task 4: Creating SSH Public Key Import Rules

Discovery rules automatically create secrets and send emails when local accounts or public keys match the rule. To create a rule to import discovered SSH public keys:

1. In Secret Server, click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

Admin > Discovery

Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery Network View Create Discovery Source Run Discovery Now

Discovery
Last Started: 2 months, 8 days ago
Next Run: soon

Computer Scan
Last Started: 2 months, 8 days ago
Next Run: soon

4 Items Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 ...
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 ...
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Click the **Discovery Network View** button. The Discovery Network View page appears:

Discovery Network View

Explain

Local Accounts Public Keys Service Accounts Domain \ Cloud Accounts

Unix Discovery Source
192.168.60.0-192.168.60.255

Advanced

Search By: All

Account Status: All

Include Child Organization Units:

+ Create Rule

Column Selection

<input type="checkbox"/>	COMPUTER	ACCOUNT	SCAN TEMPLATE	OS	CONTAINER	SECRET
<input checked="" type="checkbox"/>	192.168.60.52			Linux	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	root	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	toor	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	
<input type="checkbox"/>	bsdradiusserver.te...	uucp	SSH Local Account	FreeBSD	192.168.60.0-192.168.60.	

3. Click the **Public Keys** tab:

Discovery Network View

Explain

Local Accounts **Public Keys** Service Accounts Domain \ Cloud Accounts

Unix Discovery Source
192.168.60.0-192.168.60.255

Advanced

Search By: All

Account Status: All

Include Child Organization Units:

+ Create Rule

Column Selection

<input type="checkbox"/>	COMPUTER	ACCOUNT	SCAN TEMP
<input checked="" type="checkbox"/>	192.168.60.52		
<input type="checkbox"/>	bsdradiusserver...	mrnwosu	SSH Pub

4. Click the **Create Rule** button. The New Rule wizard begins:

New Rule [Close]

Rule [Progress Indicators]

Discovery rules will automatically create Secrets or send emails when local accounts or public keys that match the rule criteria are discovered.

Name All containing [til9Hw] in *

Description All containing [til9Hw] in Unix Discovery Source *

Active

[Previous] [Next]

5. Type a rule name in the **Name** text box.
6. Type a description for the rule in the **Description** text box.
7. Ensure the **Active** check box is selected.
8. Click the **Next** button. The **Source** page of the wizard appears:

New Rule [Close]

[Progress Indicators] **Source** [Progress Indicators]

Discovery Source Unix Discovery Source Clear

Scan Template SSH Public Key [v]

Computer Name Contains [Text Box]

OR [v] **Account Name Contains** [Text Box]

AND Operating System Name Contains [Text Box]

AND Public Key Contains tilHw [Text Box]

[Previous] [Next]

9. Select or type in the filter criteria as desired. Accounts or public keys matches are found during the next discovery run and are imported as secrets.
10. Click the **Next** button. The **Secret** page of the wizard appears:

New Rule

Secret

Create Secrets

Secret Type Unix Account (SSH Key Rotation-No Password) ▾

Folder [Launchers](#)

Secret Name \$MACHINE\\$USERNAME *

New Secret Permissions New Secrets copy permissions from folder ▾

Site Local ▾

Previous Next

11. Click the **Secret Type** dropdown list and select either **Unix Account (SSH Key Rotation - No Password)** or **Unix Account (Privileged Account SSH Key Rotation - No Password)**.
12. Click the **Folder** link to select a folder.
13. Type a name in the **Secret Name** text box (It auto fills \$MACHINE\$USERNAME Key).
14. Click the **New Secret Permissions** dropdown list to choose how permissions are propagated for the new secret.
15. Click the **Site** dropdown list to select a site.
16. Click the **Next** button. The **Private Key** page of the wizard appears:

New Rule

Private Key

I have the matching private key.

I want to change the public SSH key on the Account.

i This will save the provided private key for the selected account(s).

Previous Next

17. Click to select a selection button:
 - Choose **I have the matching private key** if you want to upload a private key and passphrase known to match the public key discovered.
 - Choose **I want to change the public SSH key on the Account** if you wish to take over the discovered public key.
18. Click the **Next** button.
 - If you chose **I have the matching private key**, the Import Key wizard page appears:

New Rule

◀ ▶ ⏪ ⏩ Import Key ⏪ ⏩ ▶ ▶

Current private key *

Current passphrase

Public Key Contains `tilHw`

1. Click the **Choose File** button to select a key file. Type the passphrase in the **Current passphrase** text box.

Note: Public keys matching the filter will not be imported if the private key and passphrase provided are not a valid match. If you do not know the private key or passphrase is unknown, use takeover to import these public keys.

2. Click the **Next** button.
 3. Click the **Add Secret** link to add an optional Unix sudo or su secret for future password changing.
 4. Click the **Next** button. The Key Rotation page of the wizard appears (see below).
- o If you chose the **I want to change the public SSH key on the Account** selection button, an initial takeover page appears:

New Rule

◀ ▶ ⏪ ⏩ Initial Takeover ⏪ ⏩ ▶ ▶

i The commands for the selected password type below will be run only on the initial take-over of the account.

Credentials
Select Secrets to be used for taking over the account(s).
ubuntu\toor Remove

[Add Secret](#)

1. Click the **Add Secret** link to add a Unix sudo or su secret to take over the public key on a discovered account. This will remove the public key from the user's authorized keys file and adds a new random SSH key.
2. Click the **Next** button. The Key Rotation page of the wizard appears.

The screenshot shows the 'New Rule' wizard at the 'Key Rotation' step. At the top, there is a breadcrumb trail with six steps, the fifth of which is 'Key Rotation'. Below the breadcrumb is a blue information box with an 'i' icon and the text: 'When passwords for these Secrets are changed in the future, Secret Server will use the settings below.' Underneath, the 'Secret Type' is set to 'Unix Account (SSH Key Rotation - No Password)'. The 'Credentials' section includes the text 'Select Secrets to be used for future password changing.' followed by 'ubuntu\toor Remove' and an 'Add Secret' link. At the bottom right, there are 'Previous' and 'Next' buttons.

19. Click the **Add Secret** link to add an optional Unix sudo or su secret for future key rotation.

20. Click the **Next** button. The **Alerts** page of the wizard appears:

The screenshot shows the 'New Rule' wizard at the 'Alerts' step. The breadcrumb trail at the top has six steps, with the sixth step, 'Alerts', highlighted. The main content area has three sections: 'ACCOUNTS FOUND' with a 'Send Email Alert for Accounts Found' checkbox; 'LIMIT DISCOVERY TAKE-OVER IMPORT' with a blue information box stating 'If the number of accounts that will be taken over exceeds the max threshold, the import is cancelled and the subscribed users below are notified by email.' and a 'Take-Over Threshold' text box containing '1000'; and 'SUBSCRIBED USERS' with radio buttons for 'Notify Discovery Administrator(s)' (selected) and 'Notify Subscribed User(s)'. At the bottom right, there are 'Previous' and 'Finish' buttons.

21. Click to select the **Send Email Alert for Accounts Found** check box if you want an email alert if public keys found during discovery matched the rule and were successfully imported.

22. Type a number in the **Take-Over Threshold** text box if you want to limit the number of accounts imported.

23. Click to select the **Subscribed Users** selection button to chose who gets notified.
24. Click the **Finish** button to save the rule. The rule is evaluated the next time discovery runs.
25. Return to the Discovery Network View page.
26. Click **View Rules** to see a list of rule for your discovery source.

VMware ESX/ESXi Account Discovery

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

During configuration, Secret Server is given a list of IP addresses or computer names that correspond to ESX or ESXi servers. Secret Server then connects to each server using the provided credentials to query for a list of user accounts on the target system.

Creating an ESX/ESXi Discovery Source

Discovery sources define a set of discovery operations. You must create one based on the built-in types prior to running discovery. To do so for ESX/ESXi:

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

The screenshot shows the 'Admin > Discovery' page with the 'Discovery Sources' tab selected. At the top, there are buttons for 'Discovery Network View', 'Create Discovery Source', and 'Run Discovery Now'. Below these are two summary cards: 'Discovery' and 'Computer Scan', both showing 'Last Started: 2 months, 8 days ago' and 'Next Run: soon'. A table below lists 4 items with columns for NAME, ACTIVE, TYPE, and SOURCE LAST RUN. The 'Include Inactive' toggle is turned off.

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	✓	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	✓	Active Directory	11/19/2020 03:45 ...
Gamma Linux	✓	Unix	11/18/2020 03:32 ...
AWS Discovery	✓	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Note the list of existing discovery sources.
3. Click the **Create Discovery Source** button and select **VMware ESX/ESXi** to choose that discovery source type. A Discovery Source page appears for that type:

The screenshot shows the 'ESX Discovery Source' configuration page. Under the 'Overview' tab, the 'Getting Started' section provides a 4-step wizard for configuration:

1. Name the Discovery Source.
2. Define the IP addresses of the ESX/ESXi Servers.
3. Choose the Site used for Discovery scanning.
4. Choose a Secret to use as credentials for scanning.

At the bottom of the page, there are three buttons: 'Skip Wizard', 'Cancel', and 'Next'.

4. The page briefly summarizes what an ESX/ESXi discovery Source is. The ESX/ESXi setup does not allow you to skip the creation wizard.

A Unix discovery source created with the wizard has the setting assigned a default value. You can change these by editing the discovery source after finishing the wizard.

5. Click the **Next** button to continue. The Discovery Source Name wizard page appears:

The screenshot shows the 'ESX Discovery Source' wizard at the 'Discovery Source Name' step. The breadcrumb trail is 'Overview > Discovery Source Name'. The main heading is 'Discovery Source Name' with the instruction 'Define the Name of the new Discovery Source'. There is an empty text input field. A blue information box contains the text: 'Choose a name that will help quickly identify this Discovery source in the future.' At the bottom are three buttons: 'Previous', 'Cancel', and 'Next'.

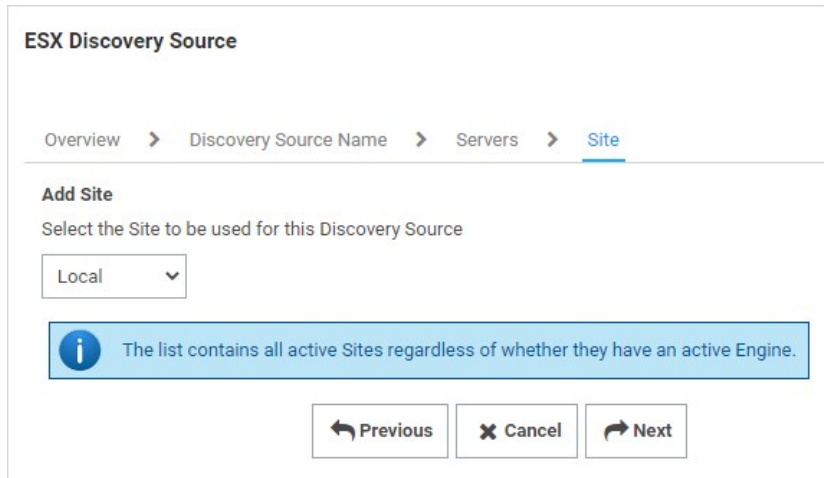
6. Type an identifying, human-readable name in the **Discovery Source Name** text box.

7. Click the **Next** button. The Servers page of the wizard appears:

The screenshot shows the 'ESX Discovery Source' wizard at the 'Servers' step. The breadcrumb trail is 'Overview > Discovery Source Name > Servers'. The main heading is 'ESX/ESXi Server IP Address/DNS Name' with the instruction 'Enter the IP Addresses or DNS Names of the ESX/ESXi Servers to scan for accounts. Enter one per line.' Below this is an example list: '192.168.40.124', '192.41.50.205', and 'ESXSVR01'. There is a large text area for input. A blue information box contains the text: 'Advanced Settings that allow customization of how ESX Discovery works are available after the Wizard is complete.' At the bottom are three buttons: 'Previous', 'Cancel', and 'Next'.

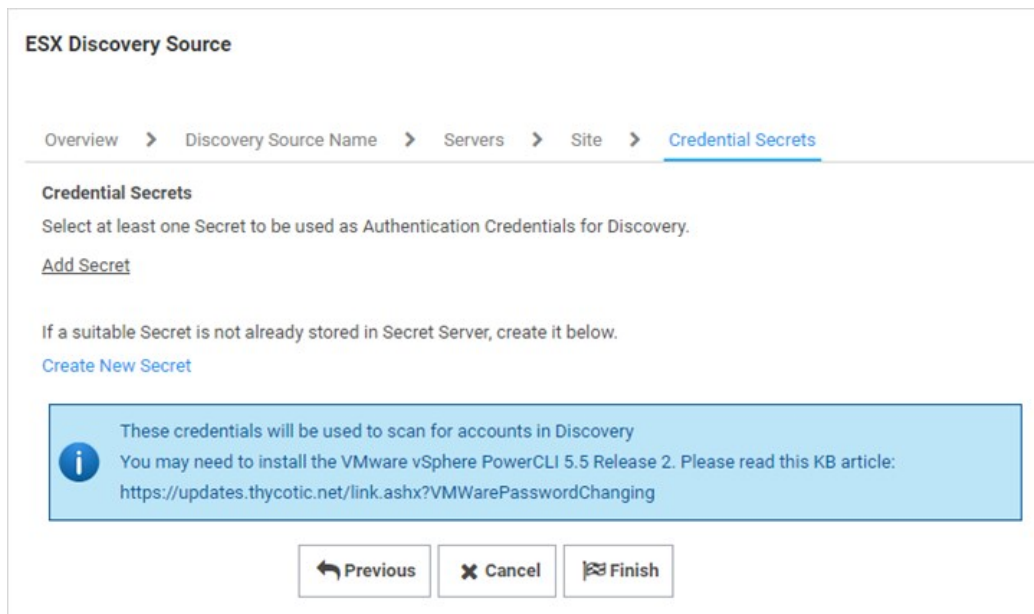
8. Type IP address or DNS name in the **ESX/ESXi Server IP Address/DNS Name** text box. Multiple entries should each be on their own line.

9. Click the **Next** button. The Add Site wizard page appears:

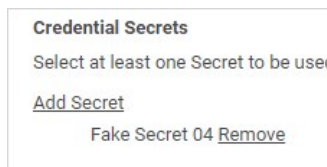


10. Click the **Add Site** dropdown list to select the desired site for the discovery source. If distributed engines are setup, the list shows all active sites. If no distributed Engines are setup, the list defaults to local, and you cannot change it.

11. Click the **Next** button. The Credential Secrets wizard page appears:



12. **Either** click the **Add Secret** link to search for and click the secret you want to use for the account credentials during the scan. The popup page closes, and the selected secret appears:



Or create a new secret for the credentials:

1. Click the **Create New Secret** link. The New (secret) popup page appears:

2. Click the **Secret Template** dropdown list and select **Generic Discovery Credentials** secret template.
 3. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.
 4. Click the **Save and Add New** button. The popup page disappears.
13. Click the **Add Secret** link to add any additional secret credentials. When using multiple credentials, discovery goes through the list of secrets attempting each credential until it either has a successful authentication or has run out of provided accounts. This loop is done for each computer.
 14. Click the **Finish** button to complete the wizard. You are returned to the Discovery Sources page where you see the new ESX/ESXi discovery source.

VMware ESX/ESXi Account Discovery and RPC Configuration

Note: Please see the [Discovery Topic](#) for a comprehensive guide to configuring and using discovery.

Overview

The ESX/ESXi (API) password changer verifies (using heartbeat) and changes VMware ESX/ESXi passwords via the vSphere API. Password changing and discovery for Secret Server 10.6 and later requires PowerCLI 6.5.1 or higher.

Either must be installed on the servers running discovery—your local Secret Server machine or machines running distributed engine. Earlier versions of the password changer are now deprecated.

Important: VMware PowerCLI 11.5 does not work due to VMware.Binding.WsTrust.dll file missing from the directory.

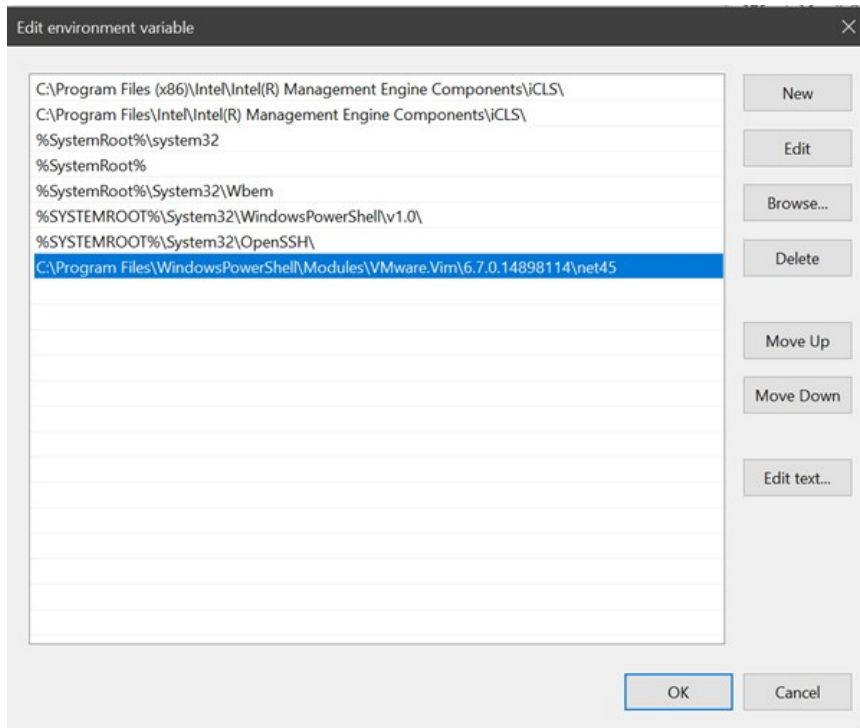
Details

Secret Server searches the machine's Windows path PATH for the VMWare SDK, therefore installing the correct version of it is all that is needed. On the machine you install VMware PowerCLI, update the Windows "Path" environment variable to include the folder where the file VMware.Vim.dll is located.

Note: After installing the VMware PowerCLI, the default installation path is: C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\[version]\net45. The PowerCLI installation path **must be** in the system PATH variable.

To edit your PATH:

1. Add C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\[version]\net45 to the PATH using the system panel (sysdm.cpl).
2. From the **System Properties** dialog, select **Advanced** tab
3. Click **Environment Variables...**
4. Under the **System Variables** section, highlight **Path** then **Edit**. The Edit Environment Variable dialog box appears:



5. Click the **New** button
6. Type C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\{version}\net45, similar to the example above:
7. Click the **OK** button when done.

Download Locations

Download supported versions of PowerCLI from VMware:

[VMware PowerCLI 11.4.0](#)

Troubleshooting and Issues

- The error "The VMware VIM API is not installed or is the wrong version" indicates that PowerCLI needs to be installed.
- We recommend not using an outdated SDK with an updated version of VMWare.
- Secret Server's VMWare password changer rejects self-signed SSL certificates. Make sure your VMWare servers have valid SSL certificates (see below for settings).
- The error "Exception: The remote certificate is invalid according to the validation procedure" indicates that vCenter server root certificates needs to be installed. More info [here](#).
- For Secret Server installed editions, you may need to restart the Secret Server website after installing PowerCLI. Do this by recycling the Secret Server application pool or performing an IIS reset.
- For distributed engines, the distributed engine service may need to be restarted after PowerCLI is installed.

ESXi Certificate Settings

Note: VMware recommends not including a CRL/CDP in certificate templates. To that end, we recommend adding the X509RevocationMode.NoCheck option to the ESXi.CertificateChainPolicyOptions setting.

Delinea added a configuration option for Secret Server to allow ESXi TLS connections to ignore self-signed certificates, allow certificates from specific issuers (even if issuer is not in trusted certificate lists), or completely skip certificate validation when using ESXi password changer, heartbeat, or discovery.

Important: For security reasons, we do **not** encourage customers to use self-signed certificates. Therefore, the new configuration settings listed below are not accessible through the UI. If you need to alter the default ESXi certificate validation settings, **submit a case through Delinea's Support Portal** for assistance.

New advanced configuration settings include:

- ESXi.IgnoreSelfSignedCerts: If true, ignores any self-signed certs (subject = issuer) from ESXi hosts during heartbeat, RPC, and discovery.
- ESXi.CertIssuersToIgnore: Semi-colon delimited list of issuer names (in format shown on certificate---such as "O=Issuer Name"). Ignores partial chain errors due to certificate being issued by any issuer in this list when that issuer is not in the trusted root or intermediate CAs lists on the server.
- ESXi.IgnoreAllCertErrors: If true, certificate validation will not be performed. All certificate errors will be ignored.
- ESXi.CertificateChainPolicyOptions: Identical to TLS Audit option, but specifically for ESXi. Allows setting X509 options to be applied to certificate validation. This is a comma-delimited list of options. See TLS Auditing or the Details section for more information.
- ESXi.ClientCertificateIds: identical to TLS Audit option, but specifically for ESXi. If ESXi host requires the client to present a valid certificate, this is a semi-colon delimited list of client certificates on the server to try to present.
- ESXi.AuditTlsErrorsDebug: Identical to TLS Audit option, but specifically for ESXi. If set to true and Secret Server (or DE) auditing is set to DEBUG, detailed debug messages about the certificate chain will be written to the log file.
- ESXi.IgnoreDefaultHostCert: Sets all the TLS configuration options necessary to not fail due to a default ESXi host certificate and its issuer not being in the trusted certificates lists. This is a combination of setting the issuer to ignore and not performing a revocation check. Setting this to true should be the first change to make when attempting to resolve heartbeat, RPC, or discovery issues to ESXi hosts when using PowerCLI versions later than 5.5.

Note: Issues with self-signed certificates previously implemented by customers were caused by a security update to the VMware vSphere PowerCLI in versions after 5.5 that no longer permits the use of self-signed certificates.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The sub-topics of this contain information that is common to discovery on all platforms.

Account Permissions for Discovery

Unix

Local Accounts

The scanning account needs to be able to connect over SSH and read the contents of `/etc/passwd`. If discovery needs to take over an account then the scanning account will also need the ability to run `sudo passwd <username>`

SSH Public Keys

The scanning account needs to ability to login and execute `sudo` without a password prompt.

Note: Please see [Discovering SSH Public Keys](#) for more information.

ESXi

The scanning account needs "Shell Access" and the "Query VRM Policy" permission.

Local Windows Accounts

The scanning account needs the "Access This Computer From the Network" permission (and possibly one more) on the endpoint:

1. Open the local group policy editor (`gpedit.msc`).
2. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
3. Double-click the **Access this computer from the network** policy. The properties for the policy appears.
4. Ensure the scanning account is one of the listed users. If not, click the **Add User or Group** button to add it.

Important: Modifying this policy may overwrite or remove access to the device for remote processes. This policy is not usually configured by default, so any existing inherited permissions could be overwritten.

5. Look at the following list of operating systems and updates to determine if any of them match your system:

- o Windows 10, version 1607 and later
- o Windows 10, version 1511 with [KB 4103198](#) installed
- o Windows 10, version 1507 with [KB 4012606](#) installed
- o Windows 8.1 with [KB 4102219](#) installed
- o Windows 7 with [KB 4012218](#) installed
- o Windows Server 2019
- o Windows Server 2016
- o Windows Server 2012 R2 with [KB 4012219](#) installed
- o Windows Server 2012 with [KB 4012220](#) installed
- o Windows Server 2008 R2 with [KB 4012218](#) installed

Note: For more information on this security issue, see [Network access: Restrict clients allowed to make remote calls to SAM](#).

6. If you found a match, do the following too:

1. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options**.
2. Double-click the **Network access: Restrict clients allowed to make remote calls to SAM** policy. The policy properties appear.

3. Click the **Edit Security** button to select an account for the Security descriptor text box. The Security Setting for Remote Access to SAM dialog box appears.
4. Ensure the scanning account is present (if not add it).
5. Click the account in the **Group or user names** list. The permissions for that account appear.
6. Ensure the **Allow** check box next to the **Remote Access** permission is selected.
7. Click the **OK** button.

Note: The discovery account must be part of the local admin's group to be able to pull back any local accounts.

Windows Services, Scheduled Tasks, App Pools, and COM+ Applications

Note: There are special considerations for discovering service accounts running COM+ Applications, please see [COM+ Dependency Scanner](#).

To scan for service accounts, the account entered must be a domain account that is in the Administrators group on the target machines. Follow the instructions below in either case to ensure your account has the appropriate privileges to run a successful scan:

1. Open the group policy editor for your domain policy.
2. Go to **Computer Configuration > Preferences > Control Panel Settings**.
3. Right-click **Local Users and groups** and select **New > Local Group**.
4. Leave the **Action** dropdown list set to **Update**.
5. Click to select **Administrators (Built-in)** in the **Group Members** dropdown list.
6. Click the **Add...** button.
7. Search for the account you will use for discovery scanning.
8. Click the **OK** button to save your changes. The next time the group policy updates across your environment, the discovery account will be part of the local administrators group.
9. For strong security, configure the group policy to limit the logon privileges of that account:
 1. Open the group policy editor
 2. For your domain policy, go to **Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
 3. Add your discovery account to the **Deny log on locally** policy.
 4. Add your discover account to the **Deny log on through Remote Desktop Services** policy.
 5. (Optional) Ensure the account is not part of the remote desktop users group.

Discovery Best Practices

Overview

This document covers the most common settings to tune to make discovery more efficient. Environmental factors contribute to some these settings.

Global Settings

The settings below might make discovery more efficient, regardless an organization's size.

Enabling Port Scanning

Introduction

Port scanning is a scan that can be conducted before the regular discovery scan to potentially reduce discovery time—if specified ports are unavailable on a given machine, the standard discovery scan will eventually timeout (the default is five minutes). Port scanning eliminates that timing out process, which saves time.

Figure: Edit Discovery Scanner for Windows Local Accounts

Setting	Value	Help Icon
Get Additional Local Account Info	<input type="checkbox"/>	?
Local Account Discovery Method	Remote Procedure Call (RPC)	?
Scanner Timeout (minutes)	5	?
Port Scan Enable	<input type="checkbox"/>	?
Port Scan Timeout	30	?
Port Scan List	135,445	?
Exclude By Name List (semi-colon)		?

Port scanning for discovery has three configurations or controls:

- Port Scan Enable: Whether to port scan at all. Defaults to unchecked.
- Port Scan Timeout: How long (in seconds) the port scan will try before giving up. Defaults to 30.
- Port Scan List: A comma-delimited list of ports to scan. These depend on the configuration of the systems you will scan. Defaults to NetBIOS (135) and Active Directory services (445).

Examples of scanners that have a port-scanning timeout option for Active Directory include:

- Windows local accounts
- Active Directory user accounts
- All dependency scanners

Accessing Port Scanning

Simply go to **Admin > Discovery Configuration > Edit Discovery Sources (button) > Configure Discovery Scanners (button) > Accounts (tab)**, and then click the pencil icon for the desired scanner. If the configurations are on that page, that scanner supports port scanning. See the previous figure.

Additional Reasons to Consider Discovery Port Scanning

Lowering the Discovery Scanner Timeout May Cause Issues

If you lower the regular discovery scanner timeout, without port scanning enabled, you may kill a running scan. In addition, non-Active-Directory discovery scanners, such as a custom PowerShell scanner, that are slow or prone to hanging may also be disrupted or even crash if the regular discovery scanner timeout is set too low. As a best practice, we recommend enabling port scanning and not lowering the regular scanner timeout, which defaults to five minutes, unless Delinea Support asks you to. Do not lower the port scanning timeout below 15 seconds.

Secrets with Multiple Dependencies May Create Especially Long Timeouts

Without discovery port scanning enabled, discovery scanners rely on the standard timeout, which defaults to five minutes. If a secret has multiple dependencies, the system may have a chain of discovery timeouts to process, one at a time. With the default five-minute timeout on all the systems, timing out can take a long time, especially if you have a lot of machines turned off or unavailable. Discovery port scanning greatly reduces that.

To calculate the maximum timeout for discovery use this formula (with all systems using the same timeout value and each secret having the same number of dependencies):

$$(\text{number of secrets}) \times (\text{number of dependencies}) \times (\text{timeout value}) = (\text{maximum minutes for discovery scans})$$

For example, using the default five-minute timeout value for 35 secrets, each with three dependencies:

$$35 \times 3 \times 5 = 525$$

Thus, 8.75 hours (525 ÷ 60) of timeout are possible and enabling discovery port scanning becomes a really good idea, especially if you have a lot of machines down at any given time.

Note: We can ignore clustered objects as part of a discovery scan, but we cannot ignore disabled computer objects, so Secret Server tries to scan each object that exists within AD. If you have a centralized area for disabled computer objects, consider configuring discovery to be OU specific and excluding your disabled computers OU to make discovery more efficient.

When to Run Discovery

Currently, you cannot set when discovery runs via a control or setting. You can, however, approximately set when it runs by disabling and enabling it at the desired time. It runs daily around the same time as when it was first enabled and then again according to whatever the [discovery scan offset hours](#) interval was set to. If you are running discovery once per day, we suggest:

- Choosing a start time outside your normal business hours, such as midnight.
- First running several ad-hoc discoveries when your network traffic normally drops at the end of the day. Record how long each discovery process takes. Remember, this can vary greatly if a lot of machines are down, which is why we suggest conducting more than one discovery.

Note: It might be fun to run one test with discovery port scanning disabled, just to see the difference.

- Using the average time the test runs took, calculate when to start discovery at a time when no anticipated portion of the discovery period is during your high-traffic times. We suggest having an end buffer as long as possible to account for variability, so if your average discovery time is fairly long, it might be best to start discovery soon after your network traffic drops off for the evening. This is especially true if your machine pool is growing.

For example, if your tested average discovery time was four hours and your network traffic is busy between 0600 and 1800, you should run discovery between 1800 and 0200, the closer to 1800 the better.

Discovery Settings

Figure: Discovery Settings Page

Discovery Sources	Configuration	Discovery Logs	Computer Scan Logs
-----------------------------------	-------------------------------	--------------------------------	------------------------------------

Discovery [Edit](#)

Discovery is used to scan for machines, local accounts and dependencies on Active Directory, Unix systems, and VMware ESX servers, AWS, and GCP. Discovery is easy to set up and provides a great range of customizations for specific network requirements.

[Learn More](#)

Enable Discovery	Yes
Discovery Interval Days	0
Discovery Interval Hours	1
Ignore Cluster Node Objects	No
Discovery Scan Offset Hours	0
Days to Keep Operational Logs	30
Deactivate Dependency Not Found Threshold	1

The settings are:

- Discovery interval for days and hours: How often you want the regular discover scan to occur.
- Ignore Cluster Node Objects: Tells Secret Server to not run discovery on machines identified as "msclustervirtualserver." Do not change this setting.
- See [Discovery Scan Offset Hours](#) for a discussion of the last setting.
- Deactivate Dependency Not Found Threshold: If set to 0, this setting means we will never disable dependencies if they are not found during a scan. If set to a higher number, it means the number of times that we will miss the dependency before deleting the dependency.

Note: There is another "Discovery Batch Size" setting on the Advance Settings page, which is usually only available to Delinea Customer Support. This setting, too, is legacy, and should not be set.

Environment-Specific Considerations

Discovery Scan Offset Hours

This section discusses a setting that allows you to quickly discover changes without greatly increasing traffic.

Figure: Discovery Settings Page in View Mode

DISCOVERY SETTINGS	
Enable Discovery	Yes
Synchronization Interval for Discovery	1 day 0 hours
Ignore Cluster Node Objects	No
Engine AD Discovery Batch Size	1
Discovery Scan Offset Hours	0

The "discovery scan offset hours" (DSOH) setting is for customers that need to detect new (to the network) systems quickly without excessive network traffic during business hours. For example, you might need this feature if you have lots of server testing (systems are up and down) or laptops (systems are connected or not). The trick is doing this while minimizing the networking load.

We accomplish this with discovery scan offsets. With these, you have multiple synchronization scans per day, rather than just one, where Secret Server attempts to scan each and every system, but first Secret Server looks up each system to see if that system is flagged for scanning. The process goes like this:

1. Initially, Secret Server scans each discovered system and resets its DSOH timer, which is set to the number of hours defined by the DSOH setting value. Secret Server has a separate timer for each scanned system.
2. Once set, each timer starts counting down. Until that timer runs out, Secret Server ignores the scanned system if it runs a discovery scan.
3. When the timer is finished, the system is again flagged for scanning.
4. The next time Secret Server does a discovery scan, it sees the flag is present and scans the system.

The period the "scan me" flag is down (the period the timer is running) is defined by the DSOH setting. Thus, DSOH essentially tells Secret Server how long before scanning that discovered system again.

For example, if you have a discovery scan offset of 12 hours and a discovery interval of four hours:

1. Start: The first time discovery runs, it scans every object because each one's timer is zeroed out, which makes it flagged for scanning. After scanning, each object's timer starts to count down, which makes it unflagged for scanning.

2. At four-hours: The next time discovery runs , it ignores the objects that were scanned the first time (because their timer was set to 12 hours), but it does process any newly discovered objects.
3. At eight-hours: In four more hours the same happens—only new objects are processed.
4. At 12 Hours: In four more hours, the scan runs again. This time, the 12-hour scan offset has expired, and all the timers of the original objects are zeroed out. The process begins anew—discovery scans every object because its timer is zeroed out, which makes it flagged for scanning. After scanning, each object's timer starts to count down, which makes it unflagged for scanning.

Advanced Settings

Note: These settings reside in the ConfigurationAdvanced.aspx file, which you should not edit unless Delinea Support asks you to.

Run Secret Computer Matcher Once per Discovery

Figure: Secret Computer Matcher Once per Day

Remote Password Changing Heartbeat Interval (Seconds)	< Not Set >
Remote Password Changing: Check for DNS Mismatch	< Not Set >
Secret Computer Matcher Dependency Password Type	< Not Set >
Secret Computer Matcher Once Per Discovery	< Not Set >
Session Callback Interval (Seconds)	< Not Set >
Should Save Files to Database	< Not Set >

During the discovery process, secrets are matched with their machine. For smaller customers, this likely has little performance impact. For very large customers, the performance impact is noteworthy. We recommend that large businesses enable this option to decrease matcher resource use.

By default, the secret computer matcher runs once every five hours (this is non-configurable). This means the matcher runs four times per day, and only one of those times could coincide with discovery running at four-hour intervals. The other three will not run in tandem with discovery and thus will increase network traffic. If you enable this setting, the matcher will instead run after each discovery completes. If discovery only runs once, the matcher only runs once too. This more efficient because discovery can take hours to run, and having the matcher run several times during that period wastes processing.

Limit the Network Traffic Caused by Nested Organizational Units

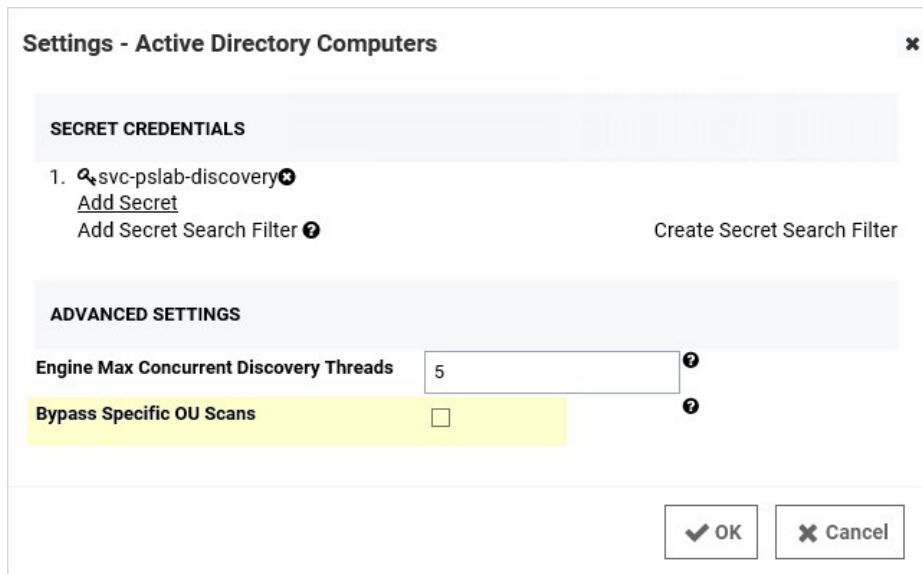
Figure: Discovery: Bypass "Scan Specific OUs"

Dependency Discovery: Ignore Domain Being Scanned	< Not Set >
Disable RADIUS NAS IP Address Attribute	< Not Set >
Disable SysLog Connection Caching	< Not Set >
Discovery: Bypass "Scan Specific Ous"	< Not Set >
Discovery Batch Size	< Not Set >
ESXi: Enable TLS Debugging and Connection Tracking	< Not Set >
ESXi: Certificate chain policy options	< Not Set >

If you configure discovery for Active Directory to scan by separate OUs and not by the entire domain, nested OUs can overwhelm your message bus. This occurs because each OU generates its own message unless you enable this setting. So if your enterprise has a complex tree of nested OUs, as many large businesses do, you could experience this issue. Smaller enterprises with single or a small number of nested OUs can ignore it. If you change the configuration in the Advance Configuration page file, it will affect all discovery source settings

(some scanners have a similar configuration that only affects them). Alternatively, for more flexibility, you can configure this individually at the scanner level by checking the Bypass Specific OU Scan check box on the Settings - Active Directory tab for the scanner:

Figure: Tuning Active Directory Settings



Engines and Engine Workers

The number of distributed engines and engine workers within your environment can affect how fast discovery completes. Increasing CPU counts on your existing engines may help them to complete a diverse set of tasks more efficiently but might not have much effect on discovery processing time. If an engine is doing discovery, only a subset of consumers run and they will run into a prefetch count limit (30 messages per engine). Thus, increasing the number of engines and engine workers might decrease total discovery time by increasing that prefetch limit.

Discovery Error Messages

The following are common error messages received when performing discovery and their possible causes:

User credentials cannot be used for local connections

This error typically occurs when attempting to run discovery on the server that Secret Server is running on, due to WMI restrictions.

No AD Account Services

No services run by Active Directory accounts have been found on the machine.

Computer is inaccessible or does not exist

Port 135 is blocked.

The target computer could not be reached

The machine is not connected to the network.

Access Denied

The account used to sync the domain with Secret Server does not have domain admin or local admin privileges for the machine it is attempting to scan for accounts.

Bad parameters - Script Error: Cannot bind argument to parameter 'Message' because it is null.

There is a mismatch between parameters referenced by the script and the arguments passed in. Check the script arguments on the scanner or dependency changer against the script.

Discovery on Non-Domain Joined or Unix Targets

Overview

When running discovery on non-domain joined targets or Unix targets, there are two methods of finding local administrator credentials to authenticate to the target, and the methods can be used together.

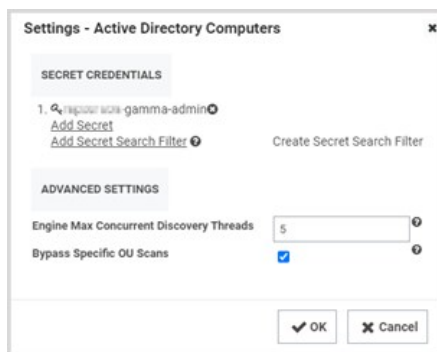
- Specify a secret with an expected default password, recommended for performing an initial scan if you have a known password or key for a privileged account.
- Specify a secret search filter, recommended when you cannot use a default password because each machine's account password is unique. A Secret Search Filter dynamically searches for a secret with a name or folder location that corresponds to the target scanned. If a matching secret is found, Secret Server will authenticate to the target using the administrator credentials in the secret.

Note: To use a secret search filter, the administrator account names must exist as secrets in Secret Server and they must follow a regular naming pattern.

Note: The discovery secret search filter is available in Secret Server 10.0.000006 and newer.

Setting Credentials on a Discovery Scanner

1. In a discovery source, click the Edit icon on a scanner.

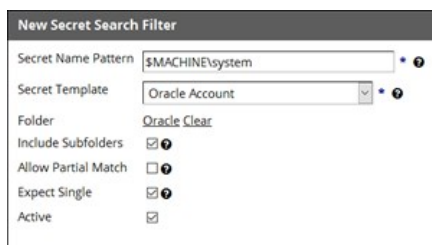


2. In the Settings window, under **Secret Credentials**, choose among the following options:
 - Click **Add Secret** to specify a default credential.
 - Click **Add Secret Search Filter** to specify an existing secret search filter.
 - Click **Create Secret Search Filter** to create a new secret search filter.

Secret Server will try the secrets and secret search filters in sequence until it finds a match.

Creating a Secret Search Filter

1. In the **Secret Credentials** section, click **Create Secret Search Filter**.



2. In the New Secret Search Filter window, specify a secret name pattern and settings as described below:

- **Secret Name Pattern:** Specifies the pattern that Secret Server will search for. The search is dynamic based on the target. In the example depicted in the screen shot, if scanning a machine named `appserver01`, Secret Server will also search for a secret named `appserver01\system`.
- **Secret Template:** Specifies the template that returned secrets should be based on. In the example depicted in the screen shot, only secrets based on the Oracle Account template will be returned.
- **Folder:** Specifies the folder to search within.
- **Include Subfolders:** Specifies that the search should include not only the specified folder, but the subfolders as well.
- **Allow Partial Match:** Specifies that secret names will be returned if they partially match the pattern. By default the secret name must be an exact match to the secret name pattern.
- **Expect Single:** Specifies that only one result should be returned. If more than one is returned, Secret Server will log an error to the discovery log.

3. Click **OK** and go back to your discovery source.

4. Click **Add Secret Search Filter** and select the filter you just created.

Now, when scanning a machine, Secret Server will try a default credential, and then it will try the secret returned by the search filter.

Introduction to Discovery Sources, Scanners, and Templates

Discovery Source

A *discovery source* is a named collective, ordered system that conducts discovery. There are five broad types: Active Directory, Amazon Web Services, Unix, VMware ESX\ESXi, and Google Cloud Platform.

Configuring discovery is defining the parameters of the discovery source, once the general type is chosen.

Each discovery source is a configurable definition of how to scan for computer assets in a given environment. A subcomponent of discovery source, called a scanner, details how to perform those scans.

Discovery Scanner

A *discovery scanner* is a component of a discovery source that that collects information during a discovery. There are four general types of scanners, called *scan templates* (in their sequential running order):

- Find host ranges
- Find machines
- Find (local) accounts
- Find dependencies

Note: They are called *scan templates* because when you create an instance of a discovery source, it includes scanners based on a standardized set of scanners specific to the platform the discovery source is designed for and the type of performed scan. That is, when you create a discovery source, you are instantiating a set of scanner objects copied from a set of static templates. You cannot modify the templates, but you can modify the scanners based on them.

Figure: Example Scanner Template for Active Directory

Discovery Source Scanner Settings



Specific organizational units are enabled for this Discovery Source. [Click here](#) to manage specific OUs for this Discovery Source.

FIND HOST RANGES

+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Organizational Units	Active Directory Domain	Organizational Unit	

Scanners: 1

FIND MACHINES

+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Computers	Organizational Unit	Windows Computer	

Scanners: 1

FIND ACCOUNTS

+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Windows Local Accounts	Windows Computer	Windows Local Account	

Scanners: 1

FIND DEPENDENCIES

+ Add New Dependency Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Windows Service	Windows Computer	Windows Service	

Scanners: 1

Back

Thus, a discovery source consists of an ordered sequence of discovery scanners, along with some data specific to the whole discovery source. Each scanner has a defined input and output, which are also based on object templates. A discovery source can have more than one scanner of a given type.

Discovery Input Template

The defined input type for a discovery scanner. An instance of the template contains the data needed to conduct the scan. The input template is often, but not always, an output template of the preceding scanner in the sequence. Some examples include Active Directory domain, AWS discovery source, organizational unit, and Windows computer.

Discovery Output Template

The defined output type for a discovery scanner. An instance of the template contains the data produced by the scan. The output template is often, but not always, an input template of the next scanner in the chain. Other times, the output may be used by another non-adjacent scanner in the discovery source.

You can also have more than one scanner of the same type in the same discovery source. For example, you could have both the Windows Local Accounts and Active Directory User Accounts scanner active in the Find Accounts section. Click the + icon next to the scanner section to see what other scanners are available there.

Some examples include: Active Directory account, AWS access key, ESXi local account, host range, organizational unit, and Windows local account.

Example

The following figure shows the data flow through the discovery source as the scanners receive and output data via input and output templates. The dataflow is as follows:

1. The original input, the domain, comes from the discovery source and was manually inputted.
2. The Active Directory Organizational Units (of type Find Host Ranges) scanner receives the domain via the Active Directory Domain input template.
3. The scan discovers the OUs of the inputted domain and returns those OUs via the Organizational Unit output template.
4. The Active Directory Computers (of type Find Machines) scanner receives the OUs from the Organizational Unit output template.
5. The scanner discovers the Windows computers belonging to the inputted OUs and returns those Windows computers via the Windows Computer output template.
6. The Windows Local Accounts (of type Find Accounts) scanner receives the Windows computers from the Windows Computer output template.
7. The scanner discovers local accounts belonging to the inputted Windows computers and returns those local accounts via the Windows Local Account output template to Secret Server *and* to the Windows Service scanner.
8. The Windows Service (of type Find Dependencies) scanner receives the Windows computers from the Windows Computer output template. This is the same input received in the last step by SS.
9. The scanner discovers Windows services belonging to the inputted Windows computers and returns those Windows services via the Windows Service output template to SS.

Figure: Discovery Scanner Dataflow

Discovery Source Scanner Settings



Specific organizational units are enabled for this Discovery Source. [Click here](#) to manage specific OUs for this Discovery Source.

FIND HOST RANGES

Manual Entry

+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Organizational Units	Active Directory Domain	Organizational Unit	

Scanners: 1

FIND MACHINES

+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Computers	Organizational Unit	Windows Computer	

Scanners: 1

FIND ACCOUNTS

+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Windows Local Accounts	Windows Computer	Windows Local Account	

Scanners: 1

FIND DEPENDENCIES

+ Add New Dependency Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Windows Service	Windows Computer	Windows Service	

Scanners: 1

Back

Editing and Adding Discovery Scanners

Many of the discovery scanners can be edited after instantiation by the discovery source. Sometimes the editable data is the same as was originally inputted for the discovery, and sometimes it is something else altogether. For example, with the above example, when you click the pencil icon in the Options column of the scanner for the Active Directory computers scanner, you see:

Settings - Active Directory Computers

SECRET CREDENTIALS

1. [Add Secret](#)
[Add Secret Search Filter](#) ? Create Secret Search Filter

ADVANCED SETTINGS

Engine Max Concurrent Discovery Threads ?

Bypass Specific OU Scans ?

The Secret Credentials section is the same as the credential secret defined when creating the discovery source. The Advanced Settings section contains settings that were not initially configurable in the discovery scanner.

In another example, when you edit the Windows Local Account scanner, you see this:

Settings - Windows Local Accounts

SECRET CREDENTIALS

[Add Secret](#)
[Add Secret Search Filter](#) ? Create Secret Search Filter

ADVANCED SETTINGS

Get Additional Local Account Info ?

Local Account Discovery Method ?

Scanner Timeout (minutes) ?

Port Scan Enable ?

Port Scan Timeout ?

Port Scan List ?

Exclude By Name List (semicolon) ?

In this case, you see many default configuration settings that were not originally settable in the discovery source. For example, you can use a different credential secret or change the ports scanned by the scanner. As mentioned above, you can also add entire scanners too—that is, more than one scanner of the same type.

Enabling Specific OU Domain Discovery

1. On the **Admin** menu click **Discovery**. The Discovery Sources tab appears:

Admin > Discovery

Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery Network View Create Discovery Source Run Discovery Now

Discovery
Last Started: 2 months, 10 days ago
Next Run: soon

Computer Scan
Last Started: 2 months, 10 days ago
Next Run: soon

4 Items Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 pm
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 pm
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 pm
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web Ser...	11/18/2020 03:32 pm

2. Click the link for the discovery source you want to configure. The Discovery Source tab for that discovery source appears:

Admin > Discovery > gamma.thycotic.com

Discovery Source Audit [Scanner Settings](#)

Active Directory [Edit](#)

Active Directory Discovery allows Secret Server to scan for Active Directory (AD) machines, Active Directory user accounts, local Windows accounts and dependencies on an AD domain. Secret Server will first discover machines from your domain; next, each machine is scanned for local Windows accounts and dependencies. By default, you can have Secret Server scan for local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools.

You can find additional accounts and dependencies by creating PowerShell scanners. PowerShell scanners are an advanced topic described in the Extensible Discovery section [KB Link](#)

Discovery Source Name *	gamma.thycotic.com
Fully Qualified Domain Name *	gamma.thycotic.com
Friendly Name *	Gamma Domain
Active *	Yes
Discovery Secret *	Gamma Domain\alwayson
Discovery Site *	Local
Discover Specific OU *	No
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended)
Use LDAPS *	No

3. Click the Edit link next to the discovery source name. The page becomes editable:

Admin > Discovery > gamma.thycotic.com

Discovery Source Audit **Scanner Settings**

Active Directory

Active Directory Discovery allows Secret Server to scan for Active Directory (AD) machines, Active Directory user accounts, local Windows accounts and dependencies on an AD domain. Secret Server will first discover machines from your domain; next, each machine is scanned for local Windows accounts and dependencies. By default, you can have Secret Server scan for local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools.

You can find additional accounts and dependencies by creating PowerShell scanners. PowerShell scanners are an advanced topic described in the Extensible Discovery section [KB Link](#)

Discovery Source Name * gamma.thycotic.com

Fully Qualified Domain Name * gamma.thycotic.com

Friendly Name * Gamma Domain

Active *

Discovery Secret * Gamma Domain\alwayson [Create New Secret](#)
[Clear](#)

Discovery Site * Local

Discover Specific OU *

Machine Resolution Type * Use Machine and Fully Qualified Name (Recommended)

Use LDAPS *

Cancel **Save**

4. Click to select the **Discover Specific OU** check box.
5. Click the Save button. The Domain Scope tab appears.
6. Click the **Domain Scope** tab:

Admin > Discovery > gamma.thycotic.com

Discovery Source **Domain Scope** Audit [Scanner Settings](#)

Domain Scope [Edit](#)

Allows discovery to target specific OUs within Active Directory. Add all of the OUs to include and then OUs that exist within any already added can be found in the search results and be selected to be excluded.

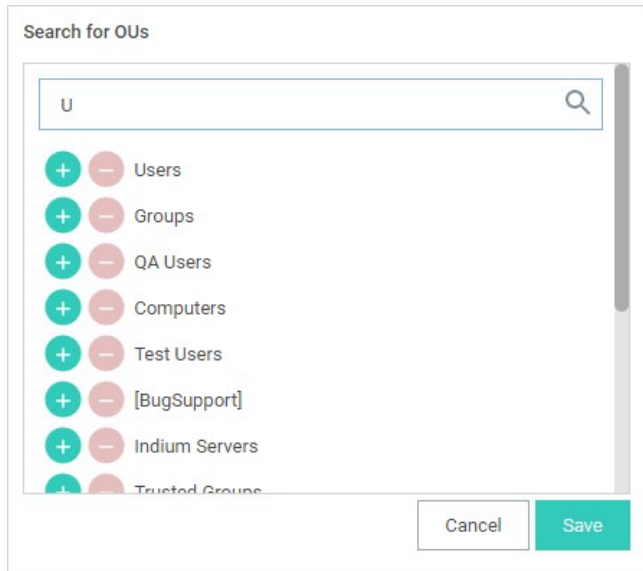
Selected OUs

No filters have been selected

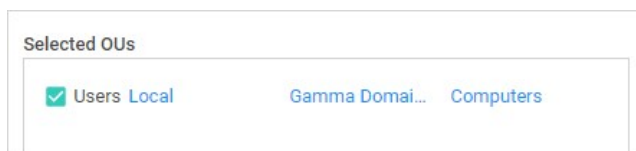
7. Click the **Edit** link next to **Domain Scope**. The page becomes editable:

The screenshot shows the 'Admin > Discovery > gamma.thycotic.com' interface. The 'Domain Scope' tab is active, with 'Discovery Source' and 'Audit' also visible. A 'Scanner Settings' button is in the top right. The 'Domain Scope' section includes a descriptive text: 'Allows discovery to target specific OUs within Active Directory. Add all of the OUs to include and then OUs that exist within any already added can be found in the search results and be selected to be excluded.' Below this is a 'Selected OUs' box containing the text 'No filters have been selected'. A 'Search for OUs' section features a search input field with a magnifying glass icon. At the bottom right are 'Cancel' and 'Save' buttons.

8. Type the name of the desired OU in the Search for OUs text box. Matching OUs appear:



9. Click the **+** in the green icon to add the OU. To remove an OU, click the red icon. The selected OU appears:



10. Click the site link to edit the sites (distributed engines) included.
11. Click the credential secret link to edit the secret used.
12. Click the scan target link to edit what you want scanned in the OU.
13. Repeat the previous steps for any additional OUs.
14. Click the **Save** button.

Note: The ports required for Discovery are documented in [Secret Server Ports](#).

Creating Discovery Rules

Discovery *account rules* automatically create secrets or send emails when local accounts that match the rule criteria are discovered.

Discovery *dependency rules* automatically add discovered dependencies to *existing* secrets when rule criteria are met—no secrets are created.

Creating Local Account Rules

Discovery account rules are search queries against the accounts found by discovery (and visible in the discovery network view). When these rules are created and run, accounts that match rules can be automatically imported as secrets. When matches are found, email notifications can also be sent out. The rule order determines the rule application order. Drag rules to reorder them. Rules can specify a combination of the domain or OU, the computer name and the account name.

To create a rule:

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

The screenshot shows the 'Admin > Discovery' interface. At the top, there are navigation tabs: 'Discovery Sources' (selected), 'Configuration', 'Discovery Logs', and 'Computer Scan Logs'. Below the tabs, there are three main buttons: 'Discovery Network View', 'Create Discovery Source', and 'Run Discovery Now'. Below these buttons, there are two summary cards: 'Discovery' (Last Started: 7 hours, 57 minutes ago; Next Run: 16 hours, 2 minutes) and 'Computer Scan' (Last Started: 16 hours, 14 minutes ago; Next Run: 7 hours, 45 minutes). Below the summary cards, there is a search bar with '4 Items' and a toggle for 'Include Inactive'. Below the search bar is a table with the following data:

NAME	ACTI...	TYPE	SOURCE LAST RUN
GCP Discovery	✓	GCP (Google Cloud Plat...	1/13/2022 07:43 AM
gamma.thycotic.com	✓	Active Directory	1/13/2022 07:43 AM
ESXi Discovery	✓	VMware ESX/ESXi	1/13/2022 07:43 AM
delta.thycotic.com	✓	Active Directory	1/13/2022 07:43 AM

2. Click the **Configuration** tab:

Admin > Discovery 🔍 🗃️ + WS

Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery [Edit](#)

Discovery is used to scan for machines, local accounts and dependencies on Active Directory, Unix systems, and VMware ESX servers, AWS, and GCP. Discovery is easy to set up and provides a great range of customizations for specific network requirements.
[Learn More](#) [🔗](#)

[Import Rules](#) [Discovery Configuration Options](#) ▾

Enable Discovery	Yes
Discovery Interval Days	1
Discovery Interval Hours	0
Ignore Cluster Node Objects	No
Discovery Scan Offset Hours	0
Days to Keep Operational Logs	30
Deactivate Dependency Not Found Threshold	2

3. Click the **Import Rules** button. The Discovery Rules page appears:

Discovery Rules

[Account Rules](#) [Dependency Rules](#)

[Explain](#)

Discovery rules are search queries against the accounts found by Discovery (and visible in the Discovery Network View). When these rules are created and run, accounts that match rules can be automatically imported as Secrets. When matches are found, email notifications can also be sent out. The order of rules determines the order in which the rules are applied. Drag rules to reorder them. Rules can specify a combination of the Domain or OU, the Computer Name and the Account Name.

+ Create Rule All Show Inactive

1) Import All

Description	This will pull it all	✎ 🗑
Search Terms	Find Accounts On Discovery Source: <i>gamma.thycotic.com</i> where the Computer Name contains: <i>yyy</i>	
When Match Found	Create Secrets	
Active	Yes	
Scan Template	Windows Local Account	
Secret Template	Windows Account	
Folder	Everyone	
Secret Name	\$MACHINE\%USERNAME	
Take-Over Threshold	1000	

[← Back](#)

Notice that a "import everything found" rule already exists.

Note: The rule order determines the order in which the rules are applied. Drag rules to reorder them.

4. Click the **Create Rule** button. The Rule page of the New Rule wizard appears:

New Rule

Rule

Discovery rules will automatically create Secrets or send emails when local accounts that match the rule criteria are discovered.

Name

Description

Active

Previous Next

5. Type the name for the new rule in the **Name** text box.
6. Type a description in the **Description** text box. At a minimum, leave the suggested log on account name as is.
7. Ensure the **Active** check box is selected.
8. Click the **Next** button. The Source page of the wizard appears:

New Rule

Source

Discovery Source None Selected*

Scan Template (Select Source)*

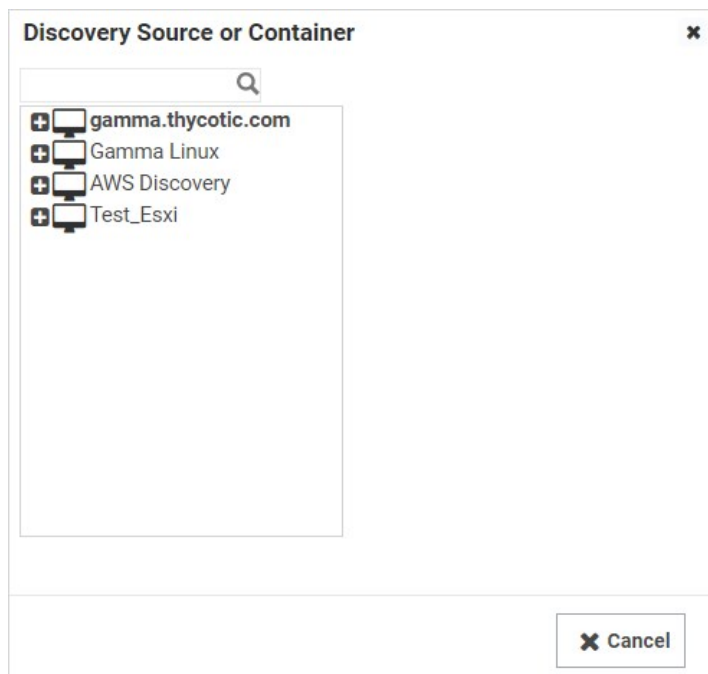
Computer Name Contains

AND Account Name Contains

AND Operating System Name Contains

Previous Next

9. Click the **Discovery Source** link to select a discovery source or container (folder). The Discovery Source or Container popup appears:



When you click a domain or subfolder with no children, the popup automatically disappears, and the information you chose appears on the Source tab:



10. If you want the rule to apply to children of what you chose, ensure the **Include Children** check box is selected.
11. Click the **Scan Template** dropdown list to select an output scan template. For a standard discovery configuration, without scripted scanners, there should only be one option here. If you added multiple local account scanners, then you can select one of their output scan templates. This limits the rule to the output results of scanners with the listed output template.
12. (Optional) Filter when the rule applies:

Note: Using a discovery rule as a search filter only applies to accounts that are found on computers in the OUs included in the discovery scan. To change those settings, modify the AD source to include more OUs or the entire domain.

1. (Optional) Type any computer name substring to filter the rule in the **Computer Name Contains** text box.
2. (Optional) **Either** if you want to add any of the following parameters to the computer name portion of the rule (one must apply), click to select the unlabeled AND/OR dropdown list and select **OR**. **Or** if you want to mandate using any of the following parameters in addition to the computer name portion of the rule (all must apply), select **AND**.
3. (Optional) Type any account name substring to filter the rule in the **Account Name Contains** text box.
4. (Optional) If you chose to use it, type any OS name substring to filter the rule in the **Operating System Name Contains** text box.

Note: The AND/OR dropdown can radically change your results, so carefully think it through. The OS name is ANDed by default—it cannot be ORed.

- Click the **Next** button. The Secret tab appears:

New Rule

Secret

Create Secrets

Secret Type: < Select > *

Folder: No Folder Selected *

Secret Name: \$MACHINE\,\$USERNAME *

New Secret Permissions: Secrets inherit permissions from folder

Site: Local

Previous Next

This is where you add creating secrets as accounts are discovered to the rule.

Note: Your previous choice of scan template (on the Source tab) alters the follow-on parameters on this tab.

- Click the **Secret Type** dropdown list to select the secret template the new secret will originate from.
- Click the **Folder** link to select a folder for the new secret to belong to.
- Click the **New Secret Permissions** dropdown list to select whether you want secrets to copy (standalone) or inherit (change with the folder) the permissions from the folder.
- Type the naming convention for the new secret in the **Secret Name** text box. You may use [dependency tokens](#) for the name. We automatically suggest a naming convention based on the hostname and username.
- Click the **Site** dropdown list to select the Secret Server local installation or a distributed engine to run the rule from.
- Click the **Next** button. The Password tab appears:

New Rule

Password

I know the current password.

I want to change the password on the Account.

(Remote Password Changing must be enabled to take over accounts)

Previous Next

- Click to select **I know the current password** selection button if you do not want Secret Server to change the account password when the secret is created. If you want Secret Server to change it, choose the other option.
- If you chose to change the password, additional selection buttons appear:

I know the current password.
 I want to change the password on the Account.
(Remote Password Changing must be enabled to take over accounts)

i Choosing this option will change the password for the selected account(s) on the remote machine(s).


I want to choose the password for all created Secrets.
 I want a new random password for each created Secret.
All selected Secrets will be given the same password.

Note: Remote password changing must be enabled to change the password.

22. **Either** click the **I want to choose...** selection button if you want all the new secrets to have the same password, which you can later change. **Or** Click the **I want a new random password...** selection button to have Secret Server create a strong password for the secret.
23. Click the **Next** button. The Import Password tab appears:

New Rule ✕

➡ > ➡ > ➡ > ➡ > Import Password ➡ > ➡ >


Current password  !

⬅ Previous Next ➡


For the random password choice, you see:

Edit Rule ✕

➡ > ➡ > ➡ > ➡ > Import Password ➡ > ➡ >

New password  *

i The privileged account below will be used only on the initial take-over of the account.

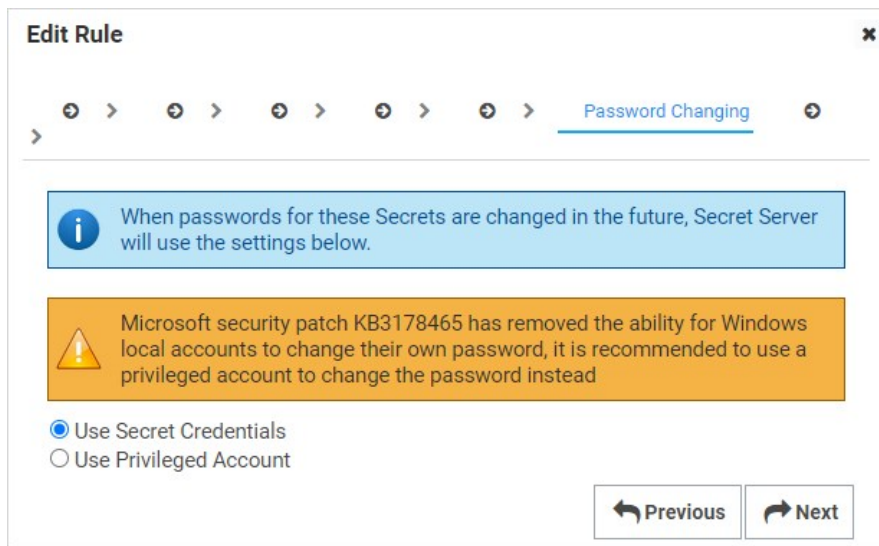
Password Type 

Credentials
Select Secrets to be used for taking over the account(s).

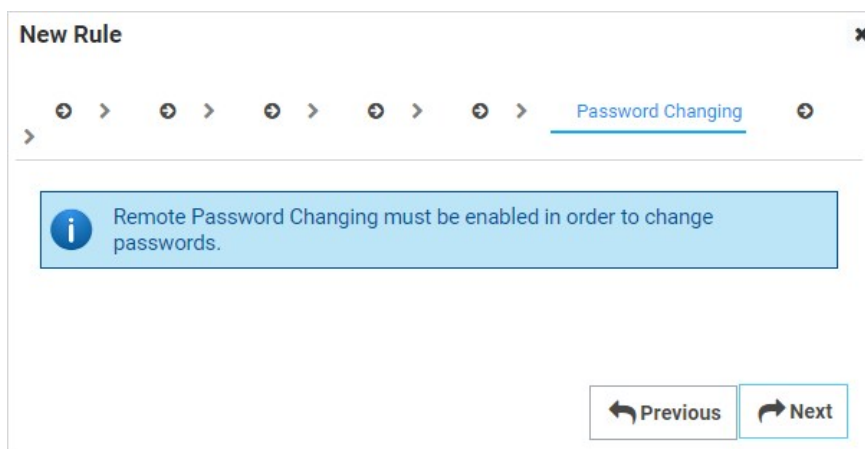
Clear

⬅ Previous Next ➡

24. If you chose a random password:
 1. Type the new password for the account used to take over the accounts for the password change in the **New Password** text box. This is *not* the password for the created secret.
 2. Click the **Password Type** dropdown list to select a password template.
 3. Click the folder link to select the existing secrets to use for taking over the accounts.
 4. For Unix Rules, select the password type command set for taking over the account. You can hold your cursor over the eye icon to see the commands to be used to change the password.
25. Type the password to use in the **Current Password** text box.
26. Click the **Next** button. The Password Changing tab appears:



If you do not have RPC enabled, you will see this instead:



27. Click to select the password changing selection button to choose whether you want to access the accounts with a secret credential or a privileged account. If you choose the latter, you will be prompted to select a secret for that account.
28. Click the **Next** button. The Alerts tab appears:

New Rule [Close]

← > ← > ← > ← > ← > ← > Alerts

ACCOUNTS FOUND

Send Email Alert for Accounts Found

← Previous Finish

29. Click to select the **Send Email Alert for Accounts Found** check box. Additional controls appear:

New Rule [Close]

← > ← > ← > ← > ← > ← > Alerts

ACCOUNTS FOUND

Send Email Alert for Accounts Found

SUBSCRIBED USERS

Notify Discovery Administrator(s)
 Notify Subscribed User(s)

← Previous Finish

30. Click to select the **Subscribed Users** selection button to choose who receives an email alert. If you select **Notify Subscribed User(s)** a text box appears for you to add email addresses. Other wise Secret Server discovery admins receive one.

31. Click the **Finish** button. The rule is created.

Creating Dependency Rules

Dependency rules automatically add dependencies (Windows services, schedule tasks, application pools) to existing secrets. You can receive email notifications of linkages by adding an event subscription in the Event Subscriptions page. Rules can specify a combination of the domain or OU.

Note: The rule order determines the order in which the rules are applied. Drag rules to reorder them.

Note: You must have a discovery scanner and dependency template configured to apply a dependency rule.

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

Admin > Discovery 🔍 ☰ + WS

[Discovery Sources](#) [Configuration](#) [Discovery Logs](#) [Computer Scan Logs](#)

Discovery Network View Create Discovery Source ▾ Run Discovery Now ▾

Discovery
Last Started: 7 hours, 57 minutes ago
Next Run: 16 hours, 2 minutes

Computer Scan
Last Started: 16 hours, 14 minutes ago
Next Run: 7 hours, 45 minutes

4 Items 🔍 ☑ Include Inactive

NAME	ACTI...	TYPE	SOURCE LAST RUN	⚙️ ↓
GCP Discovery	☑	GCP (Google Cloud Plat...	1/13/2022 07:43 AM	
gamma.thycotic.com	☑	Active Directory	1/13/2022 07:43 AM	
ESXi Discovery	☑	VMware ESX/ESXi	1/13/2022 07:43 AM	
delta.thycotic.com	☑	Active Directory	1/13/2022 07:43 AM	

2. Click the **Configuration** tab:

Admin > Discovery 🔍 ☰ + WS

[Discovery Sources](#) [Configuration](#) [Discovery Logs](#) [Computer Scan Logs](#)

Import Rules Discovery Configuration Options ▾

Discovery [Edit](#)

Discovery is used to scan for machines, local accounts and dependencies on Active Directory, Unix systems, and VMware ESX servers, AWS, and GCP. Discovery is easy to set up and provides a great range of customizations for specific network requirements. [Learn More](#) [🔗](#)

Enable Discovery	Yes
Discovery Interval Days	1
Discovery Interval Hours	0
Ignore Cluster Node Objects	No
Discovery Scan Offset Hours	0
Days to Keep Operational Logs	30
Deactivate Dependency Not Found Threshold	2

3. Click the **Import Rules** button. The Discovery Rules page appears:



Discovery Rules

[Account Rules](#) [Dependency Rules](#)

[Explain](#)

Create Rule Show Inactive

1) Import All

Description	This will pull it all	 
Search Terms	Find Accounts On Discovery Source: <i>gamma.thycotic.com</i> where the Computer Name contains: *and the Account Name contains: *and the Operating System Name contains: *	
When Match Found	Create Secrets	
Active	Yes	
Scan Template	Windows Local Account	
Secret Template	Windows Account	
Folder	Everyone	
Secret Name	\$MACHINE\%USERNAME	
Take-Over Threshold	1000	

[Back](#)

4. Click the **Dependency Rules** tab:

Discovery Rules

[Account Rules](#) [Dependency Rules](#)

[Explain](#)

Create Rule Show Inactive

[Back](#)

5. Click the **Create Rule** button. The New Rule page appears:

New Rule ✕

Dependency rules will automatically add dependency to existing Secrets. Note: No Secrets will be created.

Name *

Discovery Source None Selected*

Scan Template (Select Source) *

Dependency Template (Select Source) *

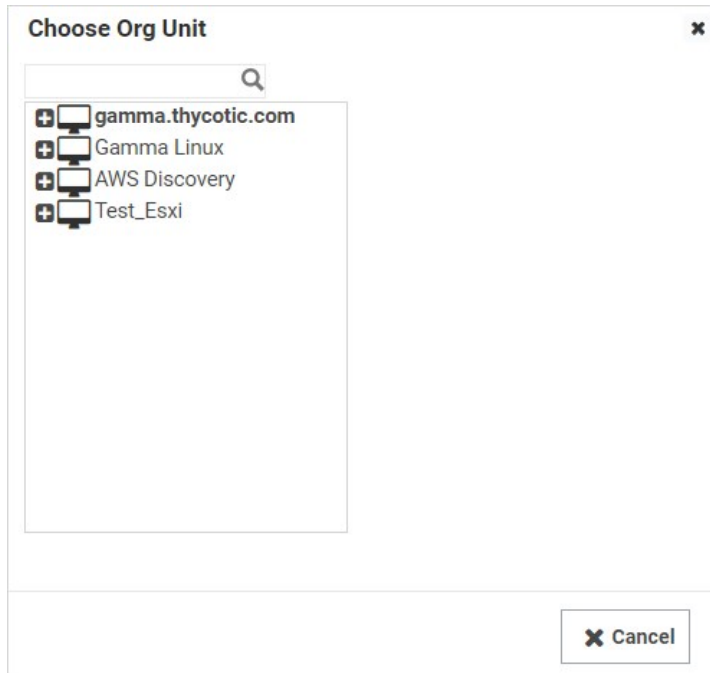
Site: ▾

Settings

Privileged Account [No Secret Selected](#)

Windows Services: Restart on Change

6. Click the **Discovery Source** link to select a discovery source or container (folder). The Choose Org Unit popup appears:



When you click a domain or subfolder with no children, the popup automatically disappears.

7. Click the **Scan Template** dropdown list to select an output template.
8. Click the **Dependency Template** dropdown list to select a dependency template.
9. Click the **Site** dropdown list to select the Secret Server local installation or a distributed engine to run the rule from.
10. Click the **Privileged Account** link to choose a secret for the scanning account.
11. Click to select the **Windows Services: Restart on Change** check box if you want the services restarted after discovery.
12. Click the **OK** button.

Discovery and Sites—Where Does Secret Server Run Discovery Scans?

Like many operations in Secret Server, you can configure discovery to run locally on IIS machines running Secret Server using website processing or by running through a distributed engine. Distributed engines are agents that you can deploy to remotely process work. They are useful for scenarios where performance is an issue or the work must take place in a remote network where the ports required by discovery are not available. You can configure discovery to use a single site location per discovery source or on a per-OU basis for AD.

Overview

Note: It is important to have an understanding of built-in discovery scanning before attempting to create your own custom scanner. Please see [Discovery Sources, Scanners, and Templates](#) and [How Discovery Works](#).

Extensible discovery lets you extend the already powerful scanning abilities of Secret Server by creating custom scanners that run PowerShell. You can use either built-in or custom scanners and templates at each step of the discovery process in extensible discovery.

If the built-in discovery sources, scanners, or input and output template, cannot you meet your needs, you can use PowerShell scripts to perform any part of discovery. Doing so requires that you define your own input and output templates and scanners and then add them to a new or existing discovery source.

When to Use Extensible Discovery

Creating a discovery source using scripted scanners can be a lot of work to set up, so when should you consider it? If you only need to discover local administrator accounts with standard dependencies (Windows services, application pools, and scheduled tasks), our built-in scanners will do the job, and extensible discovery is not necessary. However, your network probably contains other items you want to discover and bring under managed control. Here are some examples:

- Discover configuration files containing passwords and automatically add them as dependencies.
- Scan computers not joined to the domain.
- Create "dependencies" that run a SQL, SSH, or PowerShell script when a secret's password changes to log events to an external source, such as an external auditing system or an external monitoring system).
- Record information not currently imported by local account discovery or custom fields in a secret template.
- Discover SQL Server logins as "local accounts" and import them as SQL Server account secrets.

Note: To run PowerShell scanners against machines for local account and dependency discovery, you may need to configure WinRM and CredSSP. See [Configuring WinRM for PowerShell](#) and [Configuring CredSSP for WinRM with PowerShell](#)

Extensible Discovery Tutorial

Note: Extensible Discovery can be a long process for new users. It has many hands-on steps. As such, we believe a step-by-step tutorial is the best way to learn it.

Discovery scanners can run custom PowerShell scripts, as well as our built-in scanners for Active Directory, UNIX, and VMWare ESXi. You can use one or more built-in or custom scanners at each step of the discovery process: host range discovery, machine discovery, local account discovery, and dependency discovery.

Roughly, this tutorial has four tasks:

- Understanding the process
- Setting up scripts
- Creating scan templates
- Setting up discovery scanners and sources. You define dependency templates to change items manually added to secrets and added through discovery rules.

The tutorial shows you how to take full advantage of extensible discovery by creating an Active Directory discovery source that replaces each of the built-in scanners with a PowerShell script.

For simplicity, we will only create a Windows service dependency scanner in the dependencies step, but you can add additional built-in or custom PowerShell scanners to scan for application pools, scheduled tasks, or any other item that requires an action triggered when a secret's password is changed.

Task One: Understanding the Process

For most of this tutorial we will use the "Extensible Discovery Configuration" page as our launch page into each of the features that needs to be configured. Setting up Extensible Discovery requires making changes on several pages. The "Extensible Discovery Configuration" page has buttons linking to each of these pages as well as short explanations of what you need to do on them. The high-level process is:

1. Create the scripts for each discovery step.
2. If necessary, create scan templates to define the information to return from the objects discovered at each step.
Note: You want to avoid altering scan templates unless you absolutely need to. First, ensure the regular scanners cannot do the job. Once you change a template, you cannot use out-of-the-box scanners and must maintain your own PowerShell scripts for the local account and dependency scanners.
3. Create discovery scanners for each step to define which script to use for scanning, which scan template represents the objects used as the source of data for the script, and which scan template represents the object returned from the script.
4. Create a discovery source that is configured to use these discovery scanners in lieu of the default scanners.
5. Create a dependency changer for the type of dependency we want to manage.
6. Create a dependency template for the changer.
7. Manually add a dependency to a secret using the dependency changer.
8. Create a local account rule to import discovered accounts as secrets.
9. Create a dependency rule to import discovered dependencies as secrets.

Task Two: Creating the Scripts for Each Discovery Step


First, we add the scripts that we will use as our scanners to Secret Server:

Note: To use extensible discovery, you must use a Secret Server edition that supports scripts or has an "Advanced Scripting" add-on license.





















1. Go to **Admin > Scripts**. The PowerShell tab of the Scripts page appears:

Scripts

PowerShell SQL SSH

 Script testing will execute from the Web Server you are connected to but scripts may run on other nodes outside of testing

+ Create New [Information on PowerShell Scripts](#)

NAME	DESCRIPTION	CATEGORY	ACTIVE	USAGE COUNT	
Create OU Placeholder	Discovery	Discovery Scanner	Yes	2	   
Find Users in AD	Users	Discovery Scanner	Yes	0	   
Find Host Ranges	HostRange	Discovery Scanner	Yes	0	   
Find Machines	Machine	Discovery Scanner	Yes	0	   
Find Local Accounts	local Acc	Discovery Scanner	Yes	0	   

2. Click the **Create New** button. The New PowerShell Script popup page appears:

New PowerShell Script

Name

Description

Category

Script

1

3. For each of the scripts listed below:
 1. Use the name below for the **Name** text box.
 2. Copy and paste each script into a separate, new script.
 3. Use the **Discovery Scanner** category for each script.
 4. Click the **OK** button.
 5. Repeat the process for each script below.

Script Name: Host Range Scanner

```
$passwordArg = $args[2]
$username = $args[1]
$domain = $args[0]
write-debug "$domain $username $passwordArg"
$distinguisheddomain = "DC=" + ($domain.Split('.') -join ",DC=");
$spassword = ConvertTo-SecureString "$passwordArg" -AsPlainText -Force #Secure PW
$cred = New-Object System.Management.Automation.PSCredential ("$domain$username", $spassword) #Set credentials for PSCredential logon
$ous = Get-ADOrganizationalUnit -filter 'Name -like ""' -Server $domain -Credential $cred | select-object -property Name, ObjectGUID, @{Name = 'DistinguishedName'; Expression = {$_DistinguishedName.Replace("$distinguisheddomain", "")}}, ObjectClass
return $ous
# Script Args: $[1]$Domain $[1]$username $[1]$Password
```

Script Name: Machine Scanner

```
$Ou = $args[0];
$domain = $args[1];
$distinguisheddomain = "DC=" + ($domain.Split('.') -join ",DC=");
if ($distinguisheddomain.ToLower() -eq $Ou.ToLower()) {
    $searchbase = $distinguisheddomain
} else {
    $searchbase = "$Ou,$distinguisheddomain"
}
$FoundComputers = @()
$ComputersinOU = Get-ADComputer -Filter 'Name -like ""' -Server $domain -SearchBase $searchbase -properties *
foreach ($comp in $ComputersinOU) {
    $object = New-Object -TypeName PSObject
    $object | Add-Member -MemberType NoteProperty -Name ComputerName -Value $comp.Name
    $object | Add-Member -MemberType NoteProperty -Name DNSHostName -Value $comp.DNSHostName
    $object | Add-Member -MemberType NoteProperty -Name ADGUID -Value $comp.ObjectGuid
    $object | Add-Member -MemberType NoteProperty -Name OperatingSystem -Value $comp.OperatingSystem
    $object | Add-Member -MemberType NoteProperty -Name DistinguishedName -Value $comp.DistinguishedName.Replace("$distinguisheddomain", "")
    $FoundComputers += $object
}
}
```

```
return $FoundComputers
```

```
# args: $target ${1}$domain
```

Script Name: Local Account Scanner

```
$ComputerName = $args[0]
```

```
$username = $args[1]
```

```
$domain = $args[2]
```

```
$password = $args[3]
```

```
$objComputer = New-Object System.DirectoryServices.DirectoryEntry("WinNT://$ComputerName", "$domain$username", $password)
```

```
$children = $objComputer.Children | select-object SchemaClassName, Path, Name, Properties, userflags, SIDType, Disabled
```

```
$results = @()
```

```
foreach ($child in $children){
```

```
    Write-Debug $child
```

```
    if ($child.SchemaClassName -eq "User"){
```

```
        write-debug "adding to results"
```

```
        $object = New-Object -TypeName PSObject;
```

```
        $object | Add-Member -MemberType NoteProperty -Name Username -Value ${child.Name}(http://child.name/)[0];
```

```
        $object | Add-Member -MemberType NoteProperty -Name Resource -Value $ComputerName;
```

```
        $object | Add-Member -MemberType NoteProperty -Name Disabled -Value $child.Disabled;
```

```
        $results += $object;
```

```
    }
```

```
}
```

```
return $results
```

```
# Arguments $target ${1}$username ${1}$Domain ${1}$Password
```

Script Name: Windows Service Dependency Scanner

```
$ComputerName = $args[0]
```

```
$accounts = Get-WMIObject Win32_Service -ComputerName $computername | Where-Object{($_.StartName -like "*" -or $_.StartName -like "*@*") -and $_.StartName -notlike "NT *"} 
```

```
if ($accounts) {
```

```
    $dependencyaccounts = @()
```

```
    foreach($dependency in $accounts)
```

```
    {
```

```
        $object = New-Object -TypeName PSObject;
```

```
        $object | Add-Member -MemberType NoteProperty -Name ServiceName -Value $dependency.DisplayName;
```

```
        $object | Add-Member -MemberType NoteProperty -Name Enabled -Value $dependency.Started;
```

```
        if ($dependency.startname.contains('@'))
```

```
        {
```

```
            $accountinfo = $dependency.startname.split('@')
```

```
$username = $accountinfo[0]
$domain = $accountinfo[1]
}
else
{
$accountinfo = $dependency.startname.split('\')
$username = $accountinfo[1]
$domain = $accountinfo[0]
}
$object | Add-Member -MemberType NoteProperty -Name Username -Value $username;
$object | Add-Member -MemberType NoteProperty -Name Machine -Value $ComputerName;
$object | Add-Member -MemberType NoteProperty -Name Domain -Value $domain;
$object | Add-Member -MemberType NoteProperty -Name DependencyType -Value 'Powershell Script';
$object | Add-Member -MemberType NoteProperty -Name AccountStatus -Value 'Expired';
$dependencyaccounts += $object
$object = $null
}
return $dependencyaccounts;
}
throw "Error - no service accounts found"
return $null
# args: $target
```

Task Three: Creating Scan Templates

The second task is to create scan templates for each object to be discovered. Scan templates define the types of objects that can be retrieved by discovery scanners. The goal of discovery scanning is to retrieve the following:

- Accounts that can be imported and managed as secrets
- Entities (dependencies) requiring knowledge of password changes to managed secrets.

The process of finding these usually involves the following scans prior to scanning for accounts and dependencies:

- Host ranges where machines containing accounts can be found. For Active Directory, this usually involves scanning a domain for organization units or defining which organization units in a domain to check. For Unix and ESXi, this usually involved defining one or more lists of IP address ranges to scan.
- Machines to be scanned for accounts.

Each of these items—host ranges, machines, accounts, and dependencies—is defined by a scan template. The scan template specifies:

- At which step of the scanning process the item is created (scan type)
- The scan template from which the current template gets its required fields (Parent scan template)
- A list of fields that the item contains

The Fields section describes the list of properties that will be returned by the built-in scanner or script for the item. At a minimum, you need to define one field for each of the fields on the parent scan template. For items that are returned from a script, you can define additional

fields that the script will return on the object. These are mapped by name to the corresponding field on the scan template. These additional fields can then be used by later scanners.

Secret Server defines scan templates for all of its built-in scanners. Whenever possible, you should use these as input and output sources for your scripted scanners. Create your own scan templates if you need to capture additional information as data for your scripts or if you need to use specific input and output templates on the discovery scanners to drive multiple discovery workflows on a single discovery source. For this tutorial, we create new scan templates for the output of each of our scripts in this tutorial.

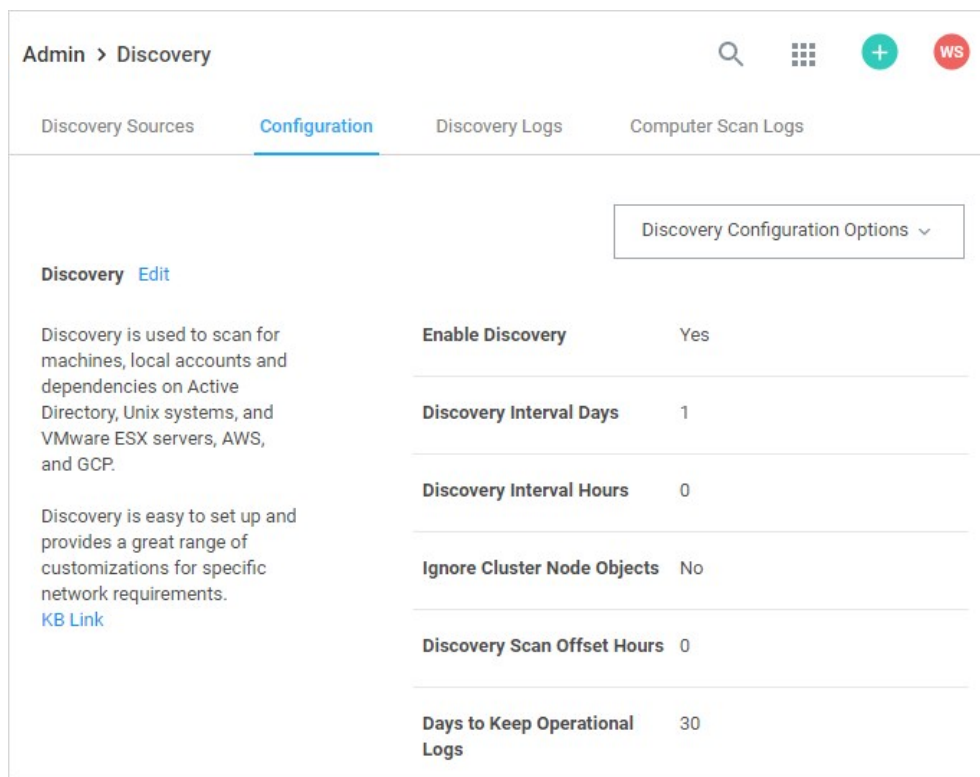
Host Range

The first scan template is the one that stores the results from our Host Range Scanner script. The script outputs an object with the following properties:

- Name
- ObjectGUID
- DistinguishedName

Thus, our scan template must have fields to store the values of these three properties. The process is as follows:

1. Go to **Admin > Discovery**.
2. Click the **Configuration** tab:



3. Click the **Discovery Configuration Options** button and select **Extensible Discovery**. The Extensible Discovery Configuration page appears:

Extensible Discovery Configuration

EXTENSIBLE DISCOVERY OVERVIEW

Extensible Discovery allows custom PowerShell Scripts to programmatically discover items in a network and bring them under management. Anything that supports PowerShell interaction can be discovered and then managed by Secret Server. Before attempting to use Extensible Discovery it is advisable that you thoroughly understand Secrets, Dependencies, and Secret Server's built in Discovery process. Below are the steps needed to successfully utilize Extensible Discovery. Additional information may be found [here](#).


Scripts

Scripts allow you to define PowerShell scripts that find objects on your network, that you link into Secrets via Scan Templates, Discovery Scanners, and Discovery Sources.

 [Edit Scripts](#)

Scan Templates

Scan Templates define the objects you are trying to find on your network. On the Scan Template, you select what type of object you are trying to find – Dependencies, Local Accounts, Machines, Host Ranges – the base parent for the Object, and the properties you want to retrieve from the object. Account-based Scan Templates can be mapped to a Password Changer for matching existing Secrets.

 [Configure Scan Templates](#)

Discovery Scanners

Discovery Scanners define the details of what is to be discovered by linking input and output Scan Templates to scripts or predefined scanners.

 [Configure Discovery Scanners](#)

4. Click the **Configure Scan Templates** button. If you are working from a new installation of Secret Server, you should see the following:

Scan Templates

Host Ranges Machines Accounts Dependencies

+ Create New Scan Template

NAME	ACTIVE
AWS Path	Yes
AWS Region	Yes
GCP Project	Yes
Host Range	Yes
Organizational Unit	Yes
PS Organizational Unit	Yes

Scan Templates: 6

Show Inactive

Back

5. On the **Host Ranges** tab, click the **Create New Scan Template** button. The Scan Template Designer page appears:

Scan Template Designer

Name *

Scan Type ▾

Parent Scan Template ▾ ?

Active

FIELDS

FIELD NAME	PARENT FIELD ?	INCLUDE IN MATCH ?
<input type="text" value="Ex: OU, Host Range"/> *	HostRange	<input checked="" type="checkbox"/>

Save Cancel

6. Type PS Organizational Unit in the **Name** text box.

7. Leave the **Scan Type** dropdown list set to **Find Host Ranges**.

8. Leave the **Parent Scan Template** dropdown list set to **Host Range**.

9. In the **Fields** section, use the blue **+** button to add a field for each of our script output object's properties:

| **Field Name** | **Parent Field** | | -- | -- | | DistinguishedName | <None> | | Name | HostRange | | ObjectGUID | <None> | [Unexpected Link Text](#)

10. When done, click the **Save** button.

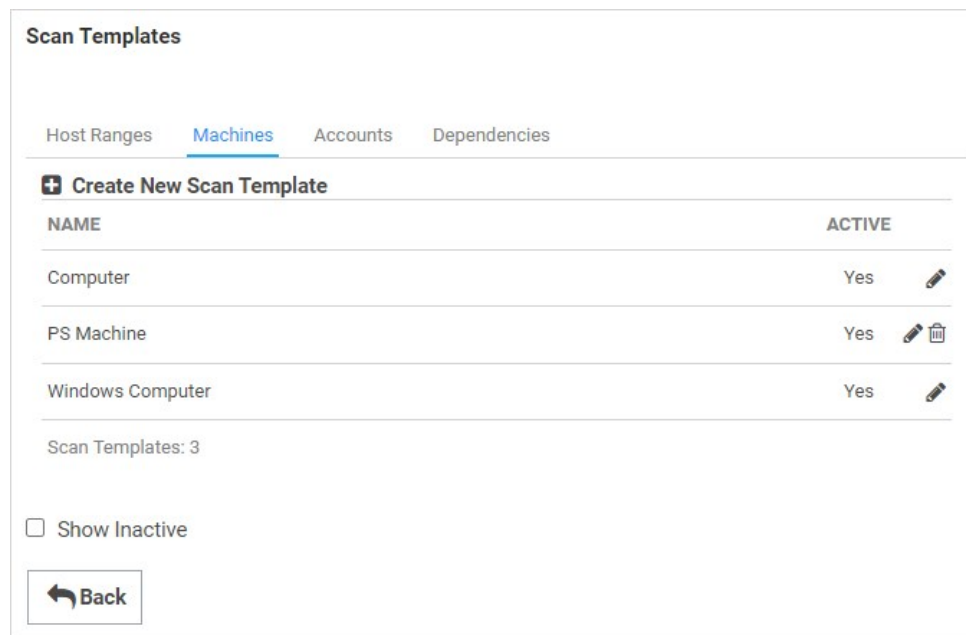
Machines

Next, create the scan template to contain the output from our Machine Scanner script. This script takes the name of an OU retrieved from our previous step, scans that OU for computers, and returns a list of custom objects containing some properties of each computer. In this tutorial we are capturing these properties:

- ADGUID
- ComputerName
- DistinguishedName
- DNSHostName
- OperatingSystem

Note: If your later scanners need more information about the computer, you can easily modify this script to return additional properties.

1. Click the **Machines** tab:



The screenshot shows the 'Scan Templates' interface with the 'Machines' tab selected. At the top, there are tabs for 'Host Ranges', 'Machines', 'Accounts', and 'Dependencies'. Below the tabs is a '+ Create New Scan Template' button. A table lists existing templates:

NAME	ACTIVE
Computer	Yes
PS Machine	Yes
Windows Computer	Yes

Below the table, it says 'Scan Templates: 3'. There is a checkbox for 'Show Inactive' which is currently unchecked. At the bottom left, there is a 'Back' button with a left-pointing arrow.

2. Click the **Create New Scan Template** button. The Scan Template Designer page appears:

Scan Template Designer

Name *

Scan Type Find Machine ▼

Parent Scan Template Computer ▼ ⓘ

Active

FIELDS

FIELD NAME	PARENT FIELD ⓘ	INCLUDE IN MATCH ⓘ
<input type="text" value="Machine"/> *	Machine	<input checked="" type="checkbox"/>
<input type="text" value="OperatingSystem"/> *	OperatingSystem ▼	<input type="checkbox"/>

3. Type PS Machine in the **Name** text box.
4. Leave the **Scan Type** dropdown list set to **Find Machine**.
5. Leave the **Parent Scan Template** dropdown list set to **Computer**.
6. In the **Fields** section, click the blue + to add a field for each of our script output object's properties:

```

| Field Name | Parent Field | | ----- | ----- | | ADGUID | <None> | | ComputerName | Machine | |
DistinguishedName | <None> | | DNSHostName | <None> | | OperatingSystem | OperatingSystem | Unexpected Link Text

```

7. Click the **Save** button.

Local Accounts

Our Local Account Scanner script takes the computer name of a computer retrieved from the previous step, scan that computer for local accounts, and return a list of custom objects containing the following properties from each account:

- Disabled
- Name
- Resource












The setup of these fields on the Local Account scan template is a bit different than the other templates that we have created so far. The parent template for local accounts is "Account" and it has three fields: Username, Password, and Resource. Our script is not able to return the password on the account so the objects returned do not have that as a property. We need to map this parent field to a field on our template, but it is only used internally.

1. Click the **Accounts** tab:

Scan Templates


Host Ranges Machines Accounts Dependencies

+ Create New Scan Template

NAME	ACTIVE	
Account (Basic)	Yes	
Active Directory Account	Yes	
AWS Access Key	Yes	
AWS User Account	Yes	
ESXi Local Account	Yes	
GCP Service Account	Yes	
PS Account	Yes	 
SQL Local Account	Yes	
SSH Local Account	Yes	
Windows Local Account	Yes	

Scan Templates: 10

Show Inactive

 Back

2. Click the **Create New Scan Template** button. The Scan Template Designer page appears:

Scan Template Designer

Name *

Scan Type Find Local Accounts ▾

Parent Scan Template Account (Basic) ▾ ⓘ

Active

FIELDS

FIELD NAME	PARENT FIELD ⓘ	INCLUDE IN MATCH ⓘ	
<input type="text" value="Ex: Machine Host, Domain"/> *	Resource	<input checked="" type="checkbox"/>	
<input type="text" value="Username"/> *	Username	<input checked="" type="checkbox"/>	
<input type="text" value="Password"/> *	<input type="text" value="Password"/> ▾	<input type="checkbox"/>	<input type="button" value="🗑️"/> <input type="button" value="➕"/>

3. Type PS Account in the **Name** text box.
4. Leave the **Scan Type** dropdown list set to **Find Local Accounts**.
5. Leave the **Parent Scan Template** dropdown list set to **Account (Basic)**.
6. In the **Fields** section, click the blue + button to add a field for each of our script output object's properties:

| Field Name | Parent Field | | -- | -- | | Disabled | < None > | | Name | Username | | Password | Password | | Resource | Resource | | [Unexpected Link Text](#)

7. Click the **Save** button.

Dependencies Scan Template

The final scan template we are going to set up is one to find Windows Service dependencies. Our script will return a list of all Windows Services on a computer along with account information for that service. The properties returned by the script for each service are:

- AccountStatus
- DependencyType
- Domain
- Enabled
- Machine
- ServiceName
- Username












Thus, our setup for this scan template will be:

1. Click the **Dependencies** tab:

Scan Templates


Host Ranges Machines Accounts Dependencies

+ Create New Scan Template

NAME	ACTIVE
COM+ Application	Yes 
Computer Dependency (Basic)	Yes 
PS Dependency	Yes  
Remote File	Yes 
SQL Dependency (Basic)	Yes 
SSH Dependency (Basic)	Yes 
SSH Key Rotation Dependency	Yes 
Windows Application Pool	Yes 
Windows Scheduled Task	Yes 
Windows Service	Yes 

Scan Templates: 10

Show Inactive

 Back

2. Click the **Create New Scan Template** button. The Scan Template Designer page appears:

Scan Template Designer

Name *

Scan Type Find Dependencies ▾

Parent Scan Template Computer Dependency (Basic) ▾ ?

Account Scan Template None ▾ * ?

Active

FIELDS

FIELD NAME	PARENT FIELD ?	INCLUDE IN MATCH ?
<input type="text" value="Machine"/> *	Machine	<input checked="" type="checkbox"/>
<input type="text" value="Ex: Service Name, Scheduled Task, Dependency Name"/> *	ServiceName	<input checked="" type="checkbox"/>
<input type="text" value="Username"/> *	Username	<input checked="" type="checkbox"/>
<input type="text"/> *	< None > ▾	<input type="checkbox"/>

3. Type PS Dependency in the **Name** text box.
4. Leave the **Scan Type** dropdown list set to **Find Dependencies**.
5. Leave the **Parent Scan Template** dropdown list set to **Computer Dependencies (Basic)**.
6. Set the **Account Scan Template** dropdown list set to **PS Account**.
7. In the **Fields** section, click the blue **+** button to add a field for each of our script output object's properties:

Field Name | **Parent Field** | -- | -- | AccountStatus | None | DependencyType | None | Domain | Domain | Enabled | <None > | Machine | Machine | ServiceName | ServiceName | Username | Username | [Unexpected Link Text](#)

8. Click the **Save** button.

Task Four: Setting up Discovery Scanners and Sources

Discovery Scanners

Now that you have created the scan templates that our scripted discovery source will need, you can create the discovery scanners.

When creating a new scanner you specify:

- Which step the scanner runs on
- What type of base scanner to use (for example, Manual Input, Windows Discovery, or PowerShell Discovery)

- Which scan provides the input for the scan
- Which scan template represents the output of the scan.
- When using a PowerShell base scanner, you also select what script to run and any arguments to pass to the script.















To get started:

1. Return to the **Extensible Discovery** page.
2. Click the **Configure Discovery Scanners** button. The Discovery Scanners page appears:

Discovery Scanners

Host Ranges Machines Accounts Dependencies

+ Create New Scanner

NAME	BASE SCANNER	INPUT TEMPLATE	OUTPUT TEMPLATE	ACTIVE	OPTIONS
Active Directory Organizational Units	Windows Discovery	Active Directory Domain	Organizational Unit	Yes	 
AWS Path Scanner	AWS Discovery	AWS Discovery Source	AWS Path	Yes	 
AWS Region Scanner	AWS Discovery	AWS Discovery Source	AWS Region	Yes	 
AWS Specific Availability Zones Scanner	AWS Discovery	AWS Discovery Source	AWS Region	Yes	 
GCP Project Scanner	GCP Discovery	GCP Discovery Source	GCP Project	Yes	 
Manual Host Range	Manual Input Discovery	Discovery Source	Host Range	Yes	 
PS Host Ranges	PowerShell Discovery	Active Directory Domain	PS Organizational Unit	Yes	 

Items: 7

Show Inactive

[← Back](#)

3. The page is similar to the scan templates page, with a tab for each type of scanner and a list of configured scanners within each tab. Secret Server comes with discovery scanners for each built-in scanner. We will add a new PowerShell scanner of each type, using the scripts and scan templates we set up in the previous sections.

Host Ranges

1. If necessary, click the **Host Ranges** tab.
2. Click the **Create New Scanner** button. The New Discovery Scanner popup page appears:

New Discovery Scanner [X]

SETTINGS

Name [] *

Description [] *

Active

Discovery Type Find Host Ranges ▾

Base Scanner Manual Input Discovery ▾

Input Template Active Directory Domain ▾

Output Template AWS Path ▾

✓ OK [X] Cancel

3. Type PS Host Ranges in the **Name** text box.
4. Type a description in the **Description** text box.
5. Leave the **Discovery Type** dropdown list set to **Find Host Ranges**. This is the scanner type we are creating. This where in the discovery process the scanner runs. Discovery scanning always proceeds Host Ranges > Machines > Local Accounts > Dependencies. The scan templates, discovery scanners, and discovery source pages all organize their contents in the same order.

Note: Although the discovery scanning process proceeds in that order, it is important to realize the output of each step *may* not be the input of the next step. Machines take host ranges as their input, and local accounts take machines as their input, but dependencies do not take local accounts as their input. Like local accounts, dependencies are on machines, so they also take machines as their input.

6. Click to select **PowerShell Discovery** in the **Base Scanner** dropdown list. The popup expands to show more controls:

New Discovery Scanner ✕

SETTINGS

Name *

Description *

Active

Discovery Type

Base Scanner

Input Template

Output Template

Script

Script Arguments

For any scripted scanner, choose "PowerShell Discovery" as the "Base Scanner." That tells the discovery process this scanner is running a script. Other options are available here based on the discovery type, such as manual entry, Windows discovery, or SSH discovery. If you do not need to run a script for a specific step of discovery but need to use a custom scan template for the input, output, or both to create a specific workflow, you can choose an option other than "PowerShell Discovery" here.

7. Click to select **Active Directory Domain** in the **Input Template** dropdown list. The input and output templates are where you define the information flow through the discovery process. Each scanner uses the output of a previous step as its input. Each scanner returns a list of results as its output. The input template defines what to use as the input data for this scanner. The output template defines what is returned from the scan and used elsewhere. To see what scanner (if any) consumes the output of a given scanner, look for the one that has the same input template as the original scanner's output.

Note: You can have multiple scanners in each step with the same input template, but each scanner has to have a unique output template. When a scanner runs, it compares the results of the current scan with the results of the previous scan that was stored in the database. It updates any existing records, adds new records for new items, and removes any records that do not match items found during the current scan. Thus, if there were more than one scanner with the same output template, the second scanner would overwrite the results of the first scan, making it pointless. This is why each output template must be unique.

8. Click to select **PS Organizational Unit** in the **Output Template** dropdown list. This is the Host Range template we created in the previous section. Generally, each output template feeds a single scanner at the next level, but you can have multiple scanners using the same input template with each using the results to find different things. For example, you could have two local account scanners defined that both use the input from the previous find machines step—one for finding Windows local accounts and the other for finding AD accounts that have rights on the computer. In turn, each scanner returns its results to its own output scan template—one creating Windows account secrets and the other creating Active Directory account secrets.
9. Click to select **Find Host Ranges** in the **Script** dropdown list. The script runs for each object matching the input template, using the arguments in the next step, finally returning an object defined by the output template.

10. Type the following in the **Script Arguments** text box, separating each with a space: `${1}$Domain ${1}$username ${1}$Password`. Script arguments can be a combination of literal values and tokens. When the script runs, these tokens are replaced with values from the input object and any privileged accounts associated with the scanner. Privileged accounts are assigned to scanners when the scanners are added to a discovery source. The table below lists the tokens that can be used as script arguments.
11. Ensure the **Active** check box is selected.
12. Click the **OK** button to save the scanner.

Table: Script Tokens

Token	Description
<code> \$target</code>	A generic placeholder for the input object. This is not used when scanning for host ranges because there is no previous scanner input source. For machine scanners, <code> \$target</code> refers to either the OU (for Active Directory discovery sources) or the IP address (for Unix and ESXi discovery sources) from the host range input. For local account and dependency scanners, <code> \$target</code> is the name of the scanned computer.
<code> \${x}\$Username</code>	The username of the nth privileged account associated with the scanner ("x" represents n). Each scanner can have one or more privileged accounts associated with it. Thus, if you need to use the username of the first privileged account in your script, you would pass in <code> \${1}\$Username</code> . The second would be <code> \${2}\$Username</code> and so forth. You can have as many privileged accounts as necessary.
<code> \${x}\$Password</code>	Similar to <code> \${x}\$Username</code> , this is the password of the nth privileged account associated with the scanner.
<code> \${x}\$Domain</code>	Similar to <code> \${x}\$Username</code> , this is the fully-qualified domain name of the nth privileged account associated with the scanner.

Machines

Once you set up one discovery scanner, the rest should be straight-forward:

1. Click the **Machines** tab:

Discovery Scanners

Host Ranges Machines Accounts Dependencies

+ Create New Scanner

NAME	BASE SCANNER	INPUT TEMPLATE	OUTPUT TEMPLATE	ACTIVE	OPTIONS
Active Directory Computers	Windows Discovery	Organizational Unit	Windows Computer	Yes	
AWS Machine (Non-Windows) Scanner	AWS Discovery	AWS Region	Computer	Yes	
AWS Windows Machine Scanner	AWS Discovery	AWS Region	Windows Computer	Yes	
GCP (Non-Windows) Instance Scanner	GCP Discovery	GCP Project	Computer	Yes	
GCP Windows Instance Scanner	GCP Discovery	GCP Project	Windows Computer	Yes	
Manual Machine List	Manual Input Discovery	Host Range	Computer	Yes	
PS Machines	PowerShell Discovery	PS Organizational Unit	PS Machine	Yes	
Unix Machine	SSH Discovery	Host Range	Computer	Yes	

Items: 8

Show Inactive

Back

- Click the **Create New Scanner** button. The New Discovery Scanner popup page appears.
- Type PS Machines Ranges in the **Name** text box.
- Type a description in the **Description** text box.
- Leave the **Discovery Type** dropdown list set to **Find Machine**.
- Click to select **PowerShell Discovery** in the **Base Scanner** dropdown list. The popup expands to show more controls.
- Click to select **PS Organizational Unit** in the **Input Template** dropdown list. This is the same as the output template from the last scanner.
- Click to select **PS Machine** in the **Output Template** dropdown list. This is the Host Range template we created in the previous section.
- Click to select **Machine Scanner** in the **Script** dropdown list. The script runs for each object matching the input template, using the arguments in the next step, finally returning an object defined by the output template.
- Type the following in the **Script Arguments** text box, separating each with a space: \$target \$[1]\$domain.
- Ensure the **Active** check box is selected.
- Click the **OK** button to save the scanner.

Local Accounts





















Repeat the process for the local accounts scanner:

1. Click the **Accounts** tab:

Discovery Scanners

Host Ranges Machines Accounts Dependencies

+ Create New Scanner

NAME	BASE SCANNER	INPUT TEMPLATE	OUTPUT TEMPLATE	ACTIVE	OPTIONS
Windows Local Accounts	Windows Discovery	Windows Computer	Windows Local Account	Yes	 
Active Directory User Accounts	Windows Discovery	Organizational Unit	Active Directory Account	Yes	 
AWS Access Key Scanner	AWS Discovery	AWS Path	AWS Access Key	Yes	 
AWS User Account Scanner	AWS Discovery	AWS Path	AWS User Account	Yes	 
ESXi Local Account	ESX Discovery	Computer	ESXi Local Account	Yes	 
File Load Local Account	File Load Discovery	Computer	Account (Basic)	Yes	 
GCP Service Account Scanner	GCP Discovery	GCP Project	GCP Service Account	Yes	 
PS Accounts	PowerShell Discovery	PS Machine	PS Account	Yes	 
Unix Non-Daemon User	SSH Discovery	Computer	SSH Local Account	Yes	 
Unix User	SSH Discovery	Computer	SSH Local Account	Yes	 

Items: 10

Show Inactive

[← Back](#)

2. Click the **Create New Scanner** button. The New Discovery Scanner popup page appears.
3. Type PS Accounts in the **Name** text box.
4. Type a description in the **Description** text box.
5. Leave the **Discovery Type** dropdown list set to **Find Local Accounts**.
6. Click to select **PowerShell Discovery** in the **Base Scanner** dropdown list. The popup expands to show more controls.
7. Click to select **PS Machine** in the **Input Template** dropdown list. This is the same as the output template from the last scanner.
8. Click to select **PS Account** in the **Output Template** dropdown list. This is the Host Range template we created in the previous section.

9. Click to select **Local Account Scanner** in the **Script** dropdown list. The script runs for each object matching the input template, using the arguments in the next step, finally returning an object defined by the output template.
10. Type the following in the **Script Arguments** text box, separating each with a space: \$target \${1}\$username \${1}\$Domain \${1}\$Password.
11. Ensure the **Active** check box is selected.
12. Click the **OK** button to save the scanner.

Dependencies

And repeat the process for the dependencies scanner:

1. Click the **Dependencies** tab:

The screenshot shows the 'Discovery Scanners' interface with the 'Dependencies' tab selected. At the top, there are tabs for 'Host Ranges', 'Machines', 'Accounts', and 'Dependencies'. Below the tabs is a '+ Create New Scanner' button. A table lists existing scanners with columns for NAME, BASE SCANNER, INPUT TEMPLATE, OUTPUT TEMPLATE, ACTIVE, and OPTIONS. The table contains five rows of scanner configurations. Below the table, it says 'Items: 5' and there is a 'Show Inactive' checkbox. At the bottom left, there is a 'Back' button with a left-pointing arrow.

NAME	BASE SCANNER	INPUT TEMPLATE	OUTPUT TEMPLATE	ACTIVE	OPTIONS
Application Pool	Windows Discovery	Windows Computer	Windows Application Pool	Yes	
COM+ Application	Windows Discovery	Windows Computer	COM+ Application	Yes	
PS Windows Services	PowerShell Discovery	PS Machine	PS Dependency	Yes	
Scheduled Task	Windows Discovery	Windows Computer	Windows Scheduled Task	Yes	
Windows Service	Windows Discovery	Windows Computer	Windows Service	Yes	

2. Click the **Create New Scanner** button. The New Discovery Scanner popup page appears.
3. Type PS Windows Services in the **Name** text box.
4. Type a description in the **Description** text box.
5. Leave the **Discovery Type** dropdown list set to **Find Dependencies**.
6. Click to select **PowerShell Discovery** in the **Base Scanner** dropdown list. The popup expands to show more controls.
7. Click to select **PS Machine** in the **Input Template** dropdown list. This is the same as the output template from the PS Machines scanner.
8. Click to select **PS Dependency** in the **Output Template** dropdown list. This is the template we created in the previous section.

9. Click to select **Windows Service Scanner** in the **Script** dropdown list. The script runs for each object matching the input template, using the arguments in the next step, finally returning an object defined by the output template.
10. Type the following in the **Script Arguments** text box: \$target.
11. Ensure the **Active** check box is selected.
12. Click the **OK** button to save the scanner.

Discovery Sources

The final step is to create a discovery source and assign the discovery scanners we just created to it:

1. Click **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears:

Admin > Discovery

Discovery Sources Configuration Discovery Logs Computer Scan Logs

Discovery Network View Create Discovery Source Run Discovery Now

Discovery
Last Started: 2 months, 8 days ago
Next Run: soon

Computer Scan
Last Started: 2 months, 8 days ago
Next Run: soon

4 Items Include Inactive

NAME	ACTIVE	TYPE	SOURCE LAST RUN
Test_Esxi	<input checked="" type="checkbox"/>	PowerShell	11/18/2020 03:32 ...
gamma.thycotic.com	<input checked="" type="checkbox"/>	Active Directory	11/19/2020 03:45 ...
Gamma Linux	<input checked="" type="checkbox"/>	Unix	11/18/2020 03:32 ...
AWS Discovery	<input checked="" type="checkbox"/>	AWS (Amazon Web...	11/18/2020 03:32 ...

2. Note the list of existing discovery sources.

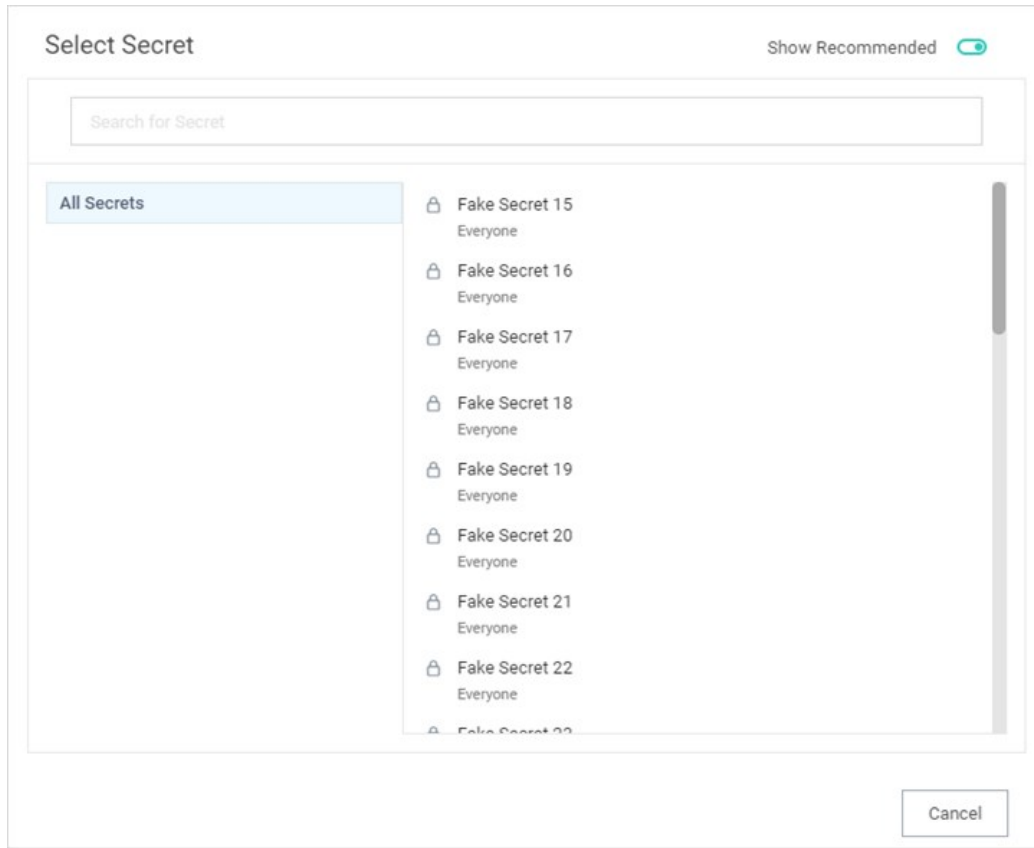
Note: If you upgraded from an earlier Secret Server version and have created an AD domain within Secret Server, a corresponding discovery source is displayed on this page. If discovery was not enabled on that domain, the discovery source Active column is not checked for that discovery source.

3. Click the **Create Discovery Source** button and select **Active Directory** to choose that discovery source type. A Discovery Source page appears for that type:

Discovery Source

Discovery Source Name *	<input type="text"/>
Fully Qualified Domain Name *	<input type="text"/>
Friendly Name *	<input type="text"/>
Active *	<input checked="" type="checkbox"/>
Discovery Secret *	No Secret Selected Create New Secret
Discovery Site *	Local <input type="text"/>
Discover Specific OU *	<input type="checkbox"/>
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended) <input type="text"/>
Use LDAPS *	<input type="checkbox"/>

4. Type the parameters for the discovery source name, FQDN, and friendly (human readable) name. The parameters with asterisks are required.
5. Ensure the **Active** check box is selected. This activates this discovery Source for scanning. Active discovery sources are scanned at the defined discovery interval defined. If you have multiple discovery sources, the discovery source with the most un-scanned computers is scanned first.
6. Next, you select a secret that is used as the credentials for discovery scanning and AD synchronization. These credentials must have the proper rights to scan the remote machines. Click the **No Secret Selected** link. The Select Secret popup page appears:



7. **Either** search for and click the secret you want to use for the account credentials during the scan. The popup page closes. The name of the secret you chose replaces the No Secret Selected link.

Or create a new secret for the credentials:

1. Click the **Create New** Secret link. The Create New Secret page appears:

Create New Secret

No Folder Selected [Change](#)

Choose a Secret Template

- Combination Lock
- Contact
- Copy of CreditCard
- Copy of Unix Account SSH
- Credit Card
- DevOps Secrets Vault Client Credentials
- Generic Discovery Credentials**
- Generic ODBC (DataSource)
- Google IAM Service Account Key
- Healthcare
- HP iLO Account (SSH)
- IBM iSeries Mainframe
- MySql Account

2. Click the **Generic Discovery Credentials** secret template. Another Create New Secret page appears:

Create New Secret

Secret Template Generic Discovery Credentials [Change](#)

Folder [No Folder Selected](#)

Secret Name *

Username *

Password [Show](#) [Generate](#)

Notes

Generate SSH Key

Private Key [Change](#)

Cancel
Create Secret

3. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.
4. Click the **Create Secret** button.
8. Click the **Discovery Site** dropdown list to select the desired site for the discovery source. If distributed engines are setup, the list shows all active sites. If no distributed engines are setup, the list defaults to local, and you cannot change it.
9. Click the **Discover Specific OU** check box to limit your discovery to an OU. See **Enabling Specific OU Domain Discovery** ADD LINK to define the scanned OU. When you select this option, a Domain Scope tab appears on the Discovery Source page for the created AD discovery source.
10. Leave the **Machine Resolution Type** dropdown list set to **Use Machine and Fully Qualified Name** unless you have a specific reason to change it.
11. Click to select the **Use LDAPS** check box to use secure LDAP for the discovery.
12. Click the **Create** button. Secret Server attempts to access the domain with your specified credentials to ensure the configuration is correct. Thus, Secret Server must have access to the domain provided, and the account credentials must work. The new discovery source is created. It appears on the Discovery Source tab of the Discovery page.
13. Click the link for the newly created discovery source. The Discovery Source tab for the source appears:

Admin > Discovery > gamma.thycotic.com

Discovery Source Domain Scope Audit **Scanner Settings**

Active Directory [Edit](#)


Active Directory Discovery allows Secret Server to scan for Active Directory (AD) machines, Active Directory user accounts, local Windows accounts and dependencies on an AD domain. Secret Server will first discover machines from your domain; next, each machine is scanned for local Windows accounts and dependencies. By default, you can have Secret Server scan for local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools.

You can find additional accounts and dependencies by creating PowerShell scanners. PowerShell scanners are an advanced topic described in the Extensible Discovery section [KB Link](#)

Discovery Source Name *	gamma.thycotic.com
Fully Qualified Domain Name *	gamma.thycotic.com
Friendly Name *	Gamma Domain
Active *	Yes
Discovery Secret *	Gamma Domain\alwayson
Discovery Site *	Local
Discover Specific OU *	Yes
Machine Resolution Type *	Use Machine and Fully Qualified Name (Recommended)
Use LDAPS *	No



14. Click the Scanner Settings button. The Discovery Source Scanner Settings page appears:

Discovery Source Scanner Settings

 Specific organizational units are enabled for this Discovery Source. Click [here](#) to manage specific OUs for this Discovery Source.

FIND HOST RANGES



+ Add New Host Range Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Organizational Units	Active Directory Domain	Organizational Unit	 

Scanners: 1

FIND MACHINES





+ Add New Machine Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Active Directory Computers	Organizational Unit	Windows Computer	 

Scanners: 1

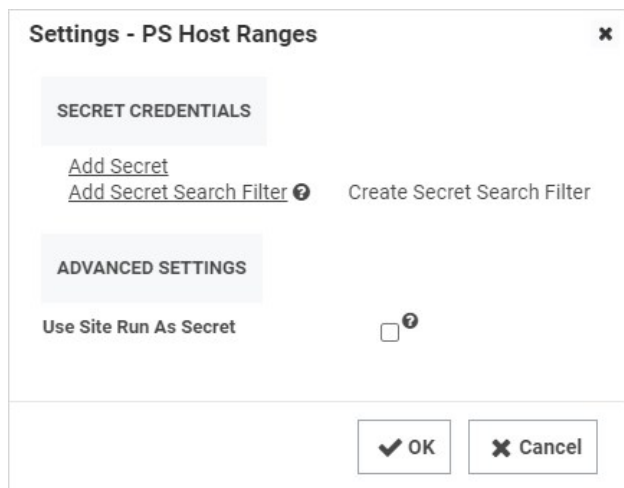
FIND ACCOUNTS

+ Add New Account Scanner

NAME	INPUT TEMPLATE	OUTPUT TEMPLATE	OPTIONS
Windows Local Accounts	Windows Computer	Windows Local Account	   

Scanners: 1

- Note that a discovery source with default scanner options is already created. For this tutorial, we are not using any of those scanners.
- Click the trashcan icon next to each scanner.
- In the **Find Host Ranges** section, click the **Add New Host Range Scanner** button. The Available Scanners popup appears.
- Click the **+** icon next to the PS Host Range scanner to select it. A settings popup for the scanner appears:



19. Click the **Add Secret** link. The Select a Secret popup appears.
20. Select a secret that has permissions to scan the domain, such as the account you linked to the domain when adding the discovery source.

Note: The "Advanced Settings" section allows you to configure options necessary for running a PowerShell script. For more information see [Configuring CredSSP for WinRM with PowerShell](#).
21. Click the **OK** button to save your settings.
22. Now that you have defined the host range scanner, repeat the process for the machine scanner, **PS Machines**, with the appropriate secret. Repeat any advanced settings from the last scanner. Once you have added a machine scanner, you can add a local account scanner.
23. Repeat the process for the local account scanner, **PS Accounts**, with the appropriate secret. Repeat any advanced settings from the last scanner.
24. Finally, repeat the process for the dependency scanner, **PS Window Services**, with the appropriate secret. Repeat any advanced settings from the last scanner.
25. Your scripted discovery source is now complete. You can go to the main discovery page to run discovery followed by a computer scan. When both are done, you should see identical results in your discovery network view to what you would get if you ran discovery with our built-in scanners.

Alerts, Auditing, Events and Logs

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server records specific events and optionally sends you alerts when they happen.

In This Section

- [Overview](#)
- [Data Retention Policies](#)
- [Permissions](#)
- [Procedures](#)
 - [Viewing the Status and History of Audit-Data Retention Policies](#)
 - [Editing Audit Data Policies](#)
 - [Running an Old Audit-Data Purge Right Now](#)

Overview

Secret Server can automatically delete older audit and audit-like information (both are called "audit data" here). By default, Secret Server does not delete any audit data.

Important: Do not configure automatic record deletion for compliance or other important data.

If enabled, old data deletion occurs automatically at 0200 EST every Sunday. Data deletion can be run immediately by clicking the "Run Now" button. The maximum record age for each audit-data retention policy is configurable to any value greater than or equal to 30 days.

Data Retention Policies

The audit data retention offers two data retention policies:

- Personally Identifiable Information (PII): Tables containing identifiable user or organization data.
- Database Size Management: Tables that are prone to grow large, which may affect Secret Server performance.

Each policy has a title and description, which are displayed to users, as well as a defined set of Secret Server audit tables it manages. There is some overlap between the two policies' table sets as some tables fall under both PII and size management.

When an audit-data retention policy runs, all records in each table for that policy that are older than the set maximum record age in days are deleted from the database. This also includes all dependent records in other tables that would otherwise prevent deletion.

Permissions

Access to the audit-data retention management pages in Secret Server is limited to users with the roles "View Data Retention" and "Administer Data Retention." As the names imply, only the latter role can manage audit data retention, such as editing and running now.

Note: The "Unlimited Admin" role does not include audit data retention management at this time.

By default, these two audit-data retention roles are not assigned to users. An admin must first assign the roles to users requiring access.

Procedures

Viewing the Status and History of Audit-Data Retention Policies

1. Go to **Admin > Data Retention Management:**

Admin > Data Retention Management

Data Retention Audit

Configure automatic permanent deletion of older audit information. By default Secret Server does not delete any audit information. Do not configure deletion of records that you need for compliance or other purposes.

The deletion of old data occurs automatically at 2 AM EST every Sunday and can be run immediately by clicking Run Now below.

Personally Identifiable Information (PII)

Enabled	Yes	Run Now	Edit
Max Record Age	365 Days		Edit
Last Start Time			
Last Complete Time			

Personally identifiable information is information such as email addresses and names that can be used to identify an individual.

This list details which data is managed by this policy.

- Event Subscription Audit
- Dual Control Audit
- Group Audit
- Secret Audit
- Folder Audit
- Secret Policy Audit
- Workflow Template Audit
- Event Audit
- User Audit
- Admin Log

The Personally Identifiable Information (PII) policy is displayed on the Data Retention tab. If you scroll down, you will see the Database Size policy:

Database Size Management

Enabled	No	Edit
Max Record Age		
Last Start Time		
Last Complete Time		

These tables may grow very large over time, which can impact performance.

This list details which data is managed by this policy.

- Group Audit
- SDK Client Audit
- Secret Audit
- Event Audit
- User Audit
- Secret Log
- Secret Item Transition History
- Secret History
- User Secret Event

2. Notice that each policy lists:

- The enabled status (editable)

- The maximum age audits are allowed to remain (editable)
- The last time the policy ran
- The last time the policy finished running
- All the audit data tables that the policy covers

3. To view a list of previous "runs," click the **Audit** tab. You can also hover the mouse pointer over individual records to view details:

Admin > Data Retention Management

Data Retention **Audit**

9 Audits

DATE RECORDED	NAME	USER	ACTION	NOTES
11/12/2019 3:29 pm	Personally Identifia...	ThycoticSystem	Truncate Records	Removed 65 total re...
11/12/2019 3:29 pm	Personally Identifia...	ThycoticSystem	Truncat	Removed 65 total records
11/12/2019 3:29 pm	Personally Identifia...	Jonathan Cogley	Truncat	Removed 1 records from [Event Subscription Audit] Removed 0 records from [Dual Control Audit]
11/12/2019 3:28 pm	Personally Identifia...	Jonathan Cogley	Edit	Removed 0 records from [Group Audit] Removed 17 records from [Secret Audit]
11/12/2019 3:05 pm	Personally Identifia...	Jonathan Cogley	Edit	Removed 5 records from [Folder Audit] Removed 3 records from [Secret Policy Audit]
11/12/2019 3:05 pm	Personally Identifia...	Jonathan Cogley	Edit	Removed 0 records from [Workflow Template Audit] Removed 6 records from [Event Audit]
11/2/2019 2:27 pm	Personally Identifia...	ThycoticSystem	Truncat	Removed 8 records from [User Audit] Removed 13 records from [Admin Log]
11/2/2019 2:27 pm	Personally Identifia...	ThycoticSystem	Truncat	Removed 0 records from [Access Request] Removed 0 records from [Access Response]
11/2/2019 2:27 pm	Personally Identifia...	Jonathan Cogley	Truncate Records	Removed 12 records from [Secret Access Request] Process Requested

Editing Audit Data Policies

1. Go to **Admin > Data Retention Management:**

Admin > Data Retention Management

Data Retention Audit

Configure automatic permanent deletion of older audit information. By default Secret Server does not delete any audit information. Do not configure deletion of records that you need for compliance or other purposes.

The deletion of old data occurs automatically at 2 AM EST every Sunday and can be run immediately by clicking Run Now below.

Personally Identifiable Information (PII)

Enabled	Yes	Run Now	Edit
Max Record Age	365 Days		Edit
Last Start Time			
Last Complete Time			

Personally identifiable information is information such as email addresses and names that can be used to identify an individual.

This list details which data is managed by this policy.

- Event Subscription Audit
- Dual Control Audit
- Group Audit
- Secret Audit
- Folder Audit
- Secret Policy Audit
- Workflow Template Audit
- Event Audit
- User Audit
- Admin Log

2. Click the **Edit** link on the **Enabled** row on the policy that you wish to edit. A popup appears (not shown).
3. Click to select the **Enabled** check box.
4. Click the **Save** button. The policy becomes enabled.
5. Click the **Edit** link on the **Max Record Age** row on the policy that you wish to edit. A popup appears (not shown).
6. Type the number of days you want to retain the data (at least 30) in the **Max Record Age** text box.
7. Click the **Save** button. The maximum record age changes.

Running an Old Audit-Data Purge Right Now

1. Go to **Admin > Data Retention Management:**

Admin > Data Retention Management

Data Retention Audit

Configure automatic permanent deletion of older audit information. By default Secret Server does not delete any audit information. Do not configure deletion of records that you need for compliance or other purposes.

The deletion of old data occurs automatically at 2 AM EST every Sunday and can be run immediately by clicking Run Now below.

Personally Identifiable Information (PII)

Enabled	Yes	Run Now	Edit
Max Record Age	365 Days		Edit
Last Start Time			
Last Complete Time			

Personally identifiable information is information such as email addresses and names that can be used to identify an individual.

This list details which data is managed by this policy.

- Event Subscription Audit
- Dual Control Audit
- Group Audit
- Secret Audit
- Folder Audit
- Secret Policy Audit
- Workflow Template Audit
- Event Audit
- User Audit
- Admin Log

2. Click the **Run Now** link on the **Enabled** row on the policy that you wish to edit. A popup appears (not shown).
3. Click the **Run Now** button. The popup disappears and the policy is run now.

Note: If a policy is currently running and you click the Run Now button. It will not work, and a popup will tell you so. There is a built-in five-minute wait after a policy finishes before you can run it again.

4. The **Last Start Time** row changes to the current time, and a progress indicator appears.
5. When the run is complete, the **Last Complete Time** row changes to the current time.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Overview

You can enable the marking or obfuscation (hiding) of Personally Identifiable Information (PII) in audit exports. This allows for data exportation for review by third parties without including any PII. Marking PII prepares exports for external, separate cleanup, and obfuscation automatically hides or removes it during exportation.

PII includes many internal stored attributes, such as IP addresses, usernames, and email addresses. Metadata fields can be flagged on creation as potentially containing PII, which aids applying the same filtering to user-configured metadata fields. PII can potentially appear in:

- System logs
- Thread logs (RPC, heartbeat, discovery, and others)
- Event logs
- Log4net files

Note: This feature currently only applies to audit tables available in the interface.

Procedure

1. In the Admin Side Panel, search for and click **Application**. The Application Configuration page appears.

Configuration

Search configuration section names, labels, and actual configuration values.

2. Click the **Edit** button.
3. Scroll down and click to select the **Obfuscate Personally Identifiable Information** check box. An Obfuscation Level dropdown list appears:

Obfuscate Personally Identifiable Information

Obfuscation Level

4. Click the Obfuscation Level dropdown list to select one of the following:
 - o **Mark PII and Obfuscate in Audit Exports:** Permanently hide or remove all PII in the exported audit report. The following PII data is hashed (replaced with unique, scrambled text that provides no PII): user ID or name, IP address, and folder path. Any other PII data is replaced with [PII REMOVED]. The hash value allows audit viewers to see that multiple entries belong to a single user

without having any specifics on who that user is.

- o **Mark PII:** Delimit all PII in easy-to-search-for markers. This allows other applications to process the PII after the export.

5. Marked PII is surrounded with three square brackets on each side, as seen below:

The screenshot shows the 'User One' profile page in the Delinea Admin console. The 'Audit' tab is selected, displaying a table with one audit record. The record shows an 'EDIT' action performed by 'Admin' on '3/23/2022 4:12 PM' from IP '127.0.0.1'. The notes field contains obfuscated PII: 'DisplayName: [[[User 1]]] to [[[User One]]]; EmailAddress: [[[user1@thycotic.com]]] to [[[userOne@thycotic.com]]]'. Buttons for 'Group Assignment Audit', 'IP Address Audit', and 'Role Assignment Audit' are visible above the table.

Obfuscated exports appear as seen in MS Excel below. Note the removed data [PII REMOVED] and the unique hash values for the user and display names:

Date Recorded	User	Action	IP Address	Notes
2022-03-23T20:12:10.483Z	1788267786	EDIT	1638446219	DisplayName: 665120776 to 1077913347; EmailAddress: [PII REMOVED] to [PII REMOVED];

This topic discusses enabling DEBUG mode for distributed engine logs for troubleshooting.

Overview

You can expand Secret Server logging capability to locate additional information regarding an error or to help with troubleshooting an issue.

Procedure

How to enable DEBUG logging mode:

1. Log in as an administrator on the distributed engine server.
2. Locate the `Thycotic.DistributedEngine.Service.exe.config` in the `C:\Program Files\Thycotic Software Ltd\Distributed Engine` directory.
3. Open the file in a text editor.
4. Run a find (**<Control>+ <F>**) command.
5. Type in `log4net` and press **<Enter>** to locate that section, which is usually at the top.
6. Locate the two lines that contain "INFO"
7. Replace each INFO with DEBUG.
8. Restart the Delinea Distributed Engine service to apply the log configuration change.
9. After DEBUG mode is enabled in the system log, you can reproduce the issue, investigate the error, or send the updated logs in with your support case.

Verbose Mode

On occasion, you may be instructed to enable the VERBOSE mode to capture details for troubleshooting. Do this by using the same procedure and replacing "INFO" or "DEBUG" with "VERBOSE" instead.

Important: Enabling VERBOSE mode will create very detailed log information with large numbers of log files that can accumulate and quickly consume available resources on the machine. Therefore, you should only enable it during the troubleshooting process and immediately turned off afterward in order to prevent performance issues.

This topic discusses enabling DEBUG mode for application system logs for troubleshooting.

Overview

You can expand Secret Server logging capability to locate additional information regarding an error or to help with troubleshooting an issue.

Note: To enable DEBUG logging for distributed engine log files, see [Enabling Debug Mode in Distributed Engine Log Files](#).

Procedure

How to enable DEBUG logging mode:

1. Log in as an administrator on the application server.
2. Locate the `web-log4net.config` file. This file can be found in the web application's root directory. If you cannot locate your web application directory, see [How to: Find the Web Application Root](#). Secret Server is typically located in the `*C:\inetpub\wwwroot\` directory; however, this is configurable so the location may be different in your environment.
3. Open the file in a text editor.
4. Run a find (**<Control>+ <F>**) command.
5. Type in `log4net` and press **<Enter>** to locate that section, which is usually at the top.
6. Locate the commented out `<level value="DEBUG" />` line.
7. Uncomment the line by removing the `<!--` and `-->`.
8. Locate the `<level value="INFO" />` line in the same section.
9. Comment out the entire line by adding a `<!--` and `-->` around it.
10. Restart IIS to apply the log configuration change.

Note: This restarts all websites hosted under IIS.

11. After DEBUG mode is enabled in the system log, you can reproduce the issue, investigate the error, or send the updated logs in with your support case.

Verbose Mode

On occasion, you may be instructed to enable the VERBOSE mode to capture details for troubleshooting. Do this by using the same procedure and replacing "INFO" or "DEBUG" with "VERBOSE" instead.

Important: Enabling VERBOSE mode will create very detailed log information with large numbers of log files that can accumulate and quickly consume available resources on the machine. Therefore, you should only enable it during the troubleshooting process and immediately turned off afterward in order to prevent performance issues.

Overview

Event pipelines (EPs) are a named group of triggers, filters, and tasks to manage events and responses to them. Event pipelines themselves can be grouped into EP policies. The Secret Server EP system is essentially a flexible instruction set builder and manager for controlling events and responses.

Event Pipeline Components

Definitions

Event Pipeline

An EP is a single named group of triggers, filters, and tasks. The same EP can be in multiple EP policies. Changing an EP affects all EP policies that EP is a part of. EPs do nothing if not assigned to an EP policy. There are two types of event pipelines—secret and user. To run an EP, include it in an active EP policy of that same type (secret or user) with set EP policy targets, such as user groups or folders.

Event Pipeline Policy

An *EP policy* is a named group of EPs that are run at the same time (in sequential order). Similar to EPs, there are two types of EP policies: *secret* or *user*. Secret EP policies target secret policies or folders and can only contain secret EPs. User EP policies target users in Groups and can only contain user EPs. EP policies must have an assigned EP policy target to work. Similarly, an EP policy with no assigned EPs does nothing.

Event Pipeline Filter

EP Filters are parameters that limit when an EP task runs. All Filters have settings and can be added to an EP multiple times. The filters are:

Secret Policy Filters

The current secret policy filters:

- Custom Variable
- Day of Week
- Event Time
- Event User: Group
- Event User: Has Two Factor
- Event User: Role
- Event User: Role Permission
- Event User: Team
- Event User: User Domain
- Event User: User Last Login
- Event User: User Setting
- IP Address
- Group
- Policy on a Secret
- Role
- Role Permission
- Secret Access Role Permission
- Secret Field
- Secret has Field

- Secret has RPC enabled
- Secret Name
- Secret Setting
- Secret Template
- Site
- Target User: Two Factor Type
- Two Factor Type

User Policy Filters

The current user policy filters:

- Custom Variable
- Day of Week
- Event Time
- Event User: Group
- Event User: Has Two Factor
- Event User: Role
- Event User: Role Permission
- Event User: Team
- Event User: User Domain
- Event User: User Last Login
- Event User: User Setting
- IP Address
- Multi-Group
- Target User: Group
- Target User: Has Two Factor
- Target User: Multi-Group
- Target User: Role
- Target User: Role Permission
- Target User: Team
- Target User: Two Factor Type
- Target User: User Domain
- Target User: User Setting
- Two Factor Type

Some filters prompt you for additional information when you select them.

Event Pipeline Policy Target

EP policy *targets* are Secret Server folders, secret policies, or user groups that are the *subject* an EP policy is applied to. For secret EP types, the secrets inside the folders or secrets under the secret policies trigger the EPs in an EP policy. As targets, folders are not recursive—only the secrets directly in the folder can trigger an EP. For user EP types, only users in the selected groups can trigger an EP.

Note: EP policy targets are *not* the receivers of task action. Those receivers are usually components of Secret Server. The term *target* is instead used for the *subject* of an EP policy—the policy targets the secret in the policy or folder to trigger the EPs to process.

Note: Event users are different than target users: The event user triggers the event. The target user is the recipient of the event.

Event Pipeline Task

Important: Tasks are powerful and can potentially do a lot of damage, so we highly recommend testing EPs in a safe

environment before using them on production secrets.

EP *tasks* are actions that are triggered in an EP, assuming any filtering conditions are met. Tasks can edit secrets, move secrets, change permissions, send notifications, and more.

Tasks run in order of their appearance on the Task tab of the Event Pipeline details page. To change the task running order, hover the mouse pointer over the one you want to move, and use the anchor on the left of its card to drag the task to the order you want it to run. If a task fails, the follow-on tasks will not run.

Note: EP targets are *not* the receivers of task action. Those receivers are usually components of Secret Server. The term *target* is instead used for the *subject* of an EP policy—the policy targets the secret in the policy or folder to trigger the EPs to process.

Note: To reference the additional secrets in the script's Args field for the update secret with a script task or run script, use `${ADD:1}` before the token. For example: `${ADD:1}$USERNAME` to reference additional secret one and `${ADD:2}$USERNAME` to reference additional secret two.)

Secret Tasks

The secret tasks are:

- Add Custom Audit
- Add Share
- Assign Secret Policy
- Assigning Site to Secret
- Change Password Remotely
- Change Secrets to not require a comment when viewed
- Change Secrets to not require Check Out
- Change Secrets to require Check Out
- Change Secrets to require Comment on View
- Change to Inherit Permissions
- Delete
- Disable Auto Change
- Disable Heartbeat
- Edit Share
- Enable Heartbeat
- Expire Secrets
- Fail with a message
- Hide Launcher Password
- Move to Folder
- Post Slack Message (WebHook)
- Retry with new random password
- Run Heartbeat
- Run Script
- Schedule Pipeline
- Secret: Add Custom Audit
- Secret: Add Share
- Secret: Assign Secret Policy
- Secret: Assigning Site to Secret
- Secret: Change Password Remotely
- Secret: Change Secrets to not require a comment when viewed
- Secret: Change Secrets to not require Check Out
- Secret: Change Secrets to require Check Out
- Secret: Change Secrets to require Comment on View

- Secret: Change to Inherit Permissions
- Secret: Delete
- Secret: Disable Auto Change on Secret
- Secret: Disable Heartbeat
- Secret: Edit Share
- Secret: Enable Auto Change on Secret
- Secret: Enable Heartbeat
- Secret: Expire Secrets
- Secret: Fail with a message
- Secret: Move to Folder
- Secret: Retry with new random password
- Secret: Run Heartbeat
- Secret: Send Email to Owners
- Secret: Set Privileged Account
- Secret: Stop RPC
- Secret: Undelete
- Secret: Update Secret by field
- Secret: Update Secret Name
- Secret: Update Secret with a script
- Secret: Viewing Password Does Not Require Edit
- Secret: Viewing Password Requires Edit
- Send Email to Event User
- Send Email to Group
- Send Email to List
- Send Email to Owners
- Set Custom Variable
- Set Privileged Account
- Stop RPC
- Undelete
- Unhide Launcher Password
- Update Secret by field
- Update Secret Name
- Update Secret with a script
- Update Secrets to automatically change the password

User Tasks

The user tasks are:

- Post Slack Message (WebHook)
- Run Script
- Schedule Pipeline
- Send Email to Event User
- Send Email to Group
- Send Email to List
- Set Custom Variable
- Target User: Add User to Group
- Target User: Add User to Team
- Target User: Disable Duo Two Factor
- Target User: Disable Email Two Factor
- Target User: Disable FIDO2 Two Factor
- Target User: Disable RADIUS Two Factor

- Target User: Disable TOTP Auth Two Factor
- Target User: Disable Users
- Target User: Enable Duo Two Factor
- Target User: Enable Email Two Factor
- Target User: Enable FIDO2 Two Factor
- Target User: Enable RADIUS Two Factor
- Target User: Enable TOTP Auth Two Factor
- Target User: Enable Users
- Target User: Force Logout
- Target User: Lock User
- Target User: Remove User from Group
- Target User: Remove User from Team
- Target User: Reset FIDO2 Two Factor
- Target User: Reset TOTP Auth Two Factor
- Target User: Send Email to Target User
- Target User: Unlock User

Event User

An event user is the user making the action. For example: Admin updated user Jane's email. Admin is the event user.

Event Variable

An event variable is a place holder for a piece of information that will manifest when the event occurs, for example the user initiating the event (`$ByUser`) or whether or not the applicable secret is active (`$secret.active`).

Target User

A target user is the affected user. Example: Admin updated user Jane's email. Jane is the target user.

Triggers

EP *triggers* are events in Secret Server that cause the EP to begin processing. All triggers have no settings and can only be added to an EP once.

Note: Pipelines are triggered following an action in most cases—pre-checkout being the exception. Therefore, you cannot interrupt an action during an event.

The triggers are:

Secret Triggers

- Access Approved
- Access Denied
- Cache View
- Check In
- Check Out
- Copy
- Create
- Custom Audit
- Custom Password Requirement Added To Field
- Custom Password Requirement Removed From Field
- Delete

- Dependency Added
- Dependency Deleted
- Dependency Failure
- Edit
- Expired Today
- Expires in 1 Day
- Expires in 15 Days
- Expires in 3 Days
- Expires in 30 Days
- Expires in 45 Days
- Expires in 60 Days
- Expires in 7 Days
- Export
- File Save
- Heartbeat Failure
- Heartbeat Success
- Hook Create
- Hook Delete
- Hook Edit
- Hook Failure
- Hook Success
- Launch
- Password Change
- Password Change Failed
- Password Change Maximum Attempts Reached
- Password Displayed
- Pre-Check In

Note: When using the Pre-Check In trigger, we recommend applying a group filter too. That trigger is a blocking call prior to secret check in that runs a script or causes the check in to fail with a warning. A problem arises when Secret Server does the same check-in process for the system "user" in the background at the end of the checkout interval. When the Pre-Check In trigger causes the check in to fail with a warning, the Secret Server background process continues to attempt check in forever, causing Secret Server to disable the pipeline. Applying a group filter ensures the trigger does not apply to the system user.

- Pre-Check Out
- Secret Policy Change
- Session Recording View
- Undelete
- View
- Viewed Secret Edit
- Web Password Fill

User Triggers

- Added to Group
- Challenge Applied
- Challenge Cleared
- Disable
- Enable
- Lockout
- Login
- Login Failure
- Logout

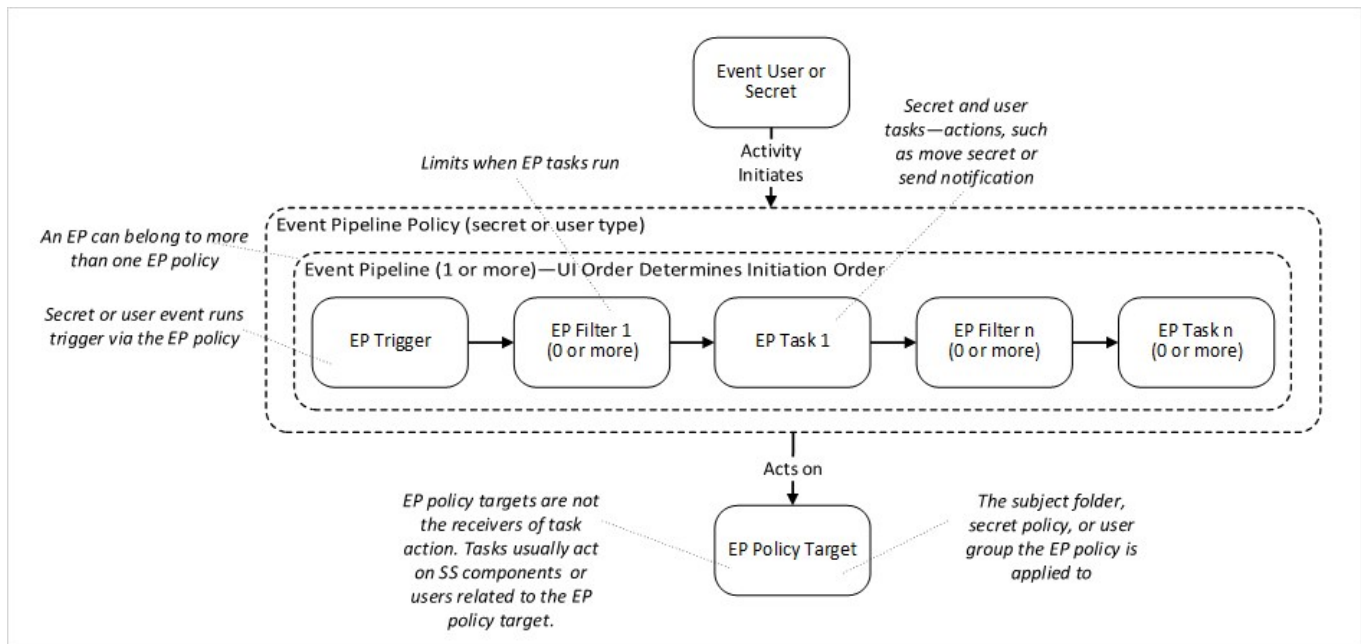
- Owners Modified
- Remove Personally Identifiable Information
- Removed From Group
- Two Factor Changed
- Two Factor Reset Failure
- Two Factor Reset Success
- User: Create
- User: Edit
- User: Password Change

Component Relationships

The following diagram shows how the components in the Definitions section relate.

Note: Please refer to the Definition section when viewing this diagram.

Figure: Component Relationships



Event Variables

Event variables are used in EP filters or tasks. They are:

Secret Field Tokens

These can be any secret field name in the tbSecretField table that is not a Password (IsPassword=0) or File (IsFile=0) type. For example, for an Active Directory Account (SecretTypeID=6001), these tokens are available: \$Username, \$Domain, or \$Notes.

Event Setting Tokens

Table: Event Setting Tokens with Filter Values

- \$metadata.secret.section.fieldname

- \$metadata.folder.section.fieldname
- \$metadata.user.section.fieldname
- \$metadata.secretfolderuser.section.fieldname

\$ByUser	Username that initiated the event	Text
\$ByUserDisplayName	Display name of user that initiated event	Text
\$ContainerName	Folder name for the event	Text
\$EventAction	Action that occurred on the event entity type. See list of triggers.	Text
\$EventDetails	Event notes. For heartbeats and RPC, this contains the status and any error message.	Text
\$EventUserKnownAs	Username for user that caused the event. If a domain account exists, then this appears as domain\username.	Text
\$ItemId	Secret ID for the event	Text
\$ItemNameForDisplay	Event secret name	Text

[Unexpected Link Text](#)

Secret Setting Tokens

Table: Secret Setting Tokens with Filter Values

\$Secret.Active	Active	Boolean
\$Secret.AutoChangeOnExpiration	Auto change on expiration	Boolean
\$Secret.ChangePasswordNow	Change password now	Boolean
\$Secret.CheckOutChangePassword	Checkout change password	Boolean
\$Secret.CheckOutEnabled	Checkout enabled	Boolean
\$Secret.EnableInheritPermissions	Enable inherit permissions	Boolean
\$Secret.EnableInheritSecretPolicy	Enable inherit secret policy	Boolean

\$Secret.Expired	Expired	Boolean
\$Secret.HideLauncherPassword	Hide launcher password	Boolean
\$Secret.IsDoubleLock	Double lock	Boolean
\$Secret.IsSessionRecordingEnabled	Session recording enabled	Boolean
\$Secret.IsSSHProxyEnabled	SSH proxy enabled	Boolean
\$Secret.LastHeartBeatStatus	Status of last heartbeat	AccessDenied; AccountLockedOut; ArgumentError; Disabled; DnsMismatch; Failed; IncompatibleHost; Pending; Processing; Success; UnableToConnect; UnableToValidateServerPublicKey; UnknownError
\$Secret.PasswordChangeFailed	Password change failed	Boolean
\$Secret.PasswordChangeOutOfSync	Password change out of sync	Boolean
\$Secret.PasswordChangeStatus	Password change status	None; Pending; Processing
\$Secret.PasswordComplianceCode	Password compliance code	Pending; Pass; Fail
\$Secret.RequireApprovalForAccess	Require approval for access	Boolean
\$Secret.RequireApprovalForAccessForEditors	Require approval for access for editors	Boolean
\$Secret.RequireApprovalForAccessForOwnersAndApprovers	Require approval for access for owners and approvers	Boolean
\$Secret.RequireViewComment	Require view comment	Boolean

\$Secret.RestrictSshCommands	Restrict SSH commands	Boolean
\$Secret.RPCAttemptCount	RPC attempt count	Boolean
\$Secret.SecretId	Secret ID	Text
\$Secret.SecretPolicyId	Secret policy ID	Text
\$Secret.SecretTemplateName	Secret template name	Text

[Unexpected Link Text](#)

Additional Tokens

Secret

- \$SecretName
- \$SecretId

Folder

- \$FolderId
- \$FolderName
- \$FolderPath

Event User

- \$EventUserDomain
- \$EventUserKnownAs
- \$EventUserName
- \$EventUserLastLogin
- \$EventUserId

Metadata

- \$metadata.folder.section.fieldname (secret event pipelines)
- \$metadata.secret.section.fieldname (secret event pipelines)
- \$metadata.secretfolderuser.section.fieldname (secret event pipelines)
- \$metadata.user.section.fieldname (secret and user event pipelines)
- \$metadata.user.section.fieldname (user event pipelines)

Target User

- \$TargetUser.DisplayName
- \$TargetUser.IsApplicationAccount
- \$TargetUser.IsSystemUser
- \$TargetUser.UserEmail
- \$TargetUser.UserEnabled
- \$TargetUser.UserName
- \$TargetUserDomain
- \$TargetUserId
- \$TargetUserKnownAs
- \$TargetUserLastLogin
- \$TargetUserName

Custom Task Variables

These are variables created with the EP task. There are two types, global and item, both of which are referenced in the same way.

Note: You must set a custom variable before using it. Thus, you cannot set a variable and use it in the same pipeline. One way around this is to create two pipelines in a policy—the first pipeline sets the variable and the second one uses it. Another way is to first set the variable in Secret Server.

Global Variable

- \$GlobalVariable.CustomVariableName
- This custom task variable is global, so there should only be one per variable name.

Item Variable

- \$ItemVariable.CustomVariableName
- This variable is per SecretId (secret pipeline) or UserId (user pipeline).

Note: The first time an EP task is invoked, an item variable is not translated, but subsequent invocations have the variable. Global variables are immediately available.

Permissions

There are three permissions:

- **Administer Pipelines:** Allows the user to create, edit, and remove EPs and EP policies.
- **Assign Pipelines:** Allows the user to assign an EP policy to secret policies, or folders.
- **View Pipelines:** Allows the user to view EP policies and policy activities.

Procedures

Event Pipelines

Activating or Deactivating Event Pipelines

To control if an EP is available to all EP policies, you can toggle the EP's active status:

1. Go to the **Event Pipelines** page.
2. Click the **Pipelines** tab.

3. Locate the card for the EP you want to activate or deactivate.
4. Click the **Active/Inactive** toggle button. A confirmation popup appears.
5. Click the **OK** button. The EP's status is changed for all EP policies it belongs to.

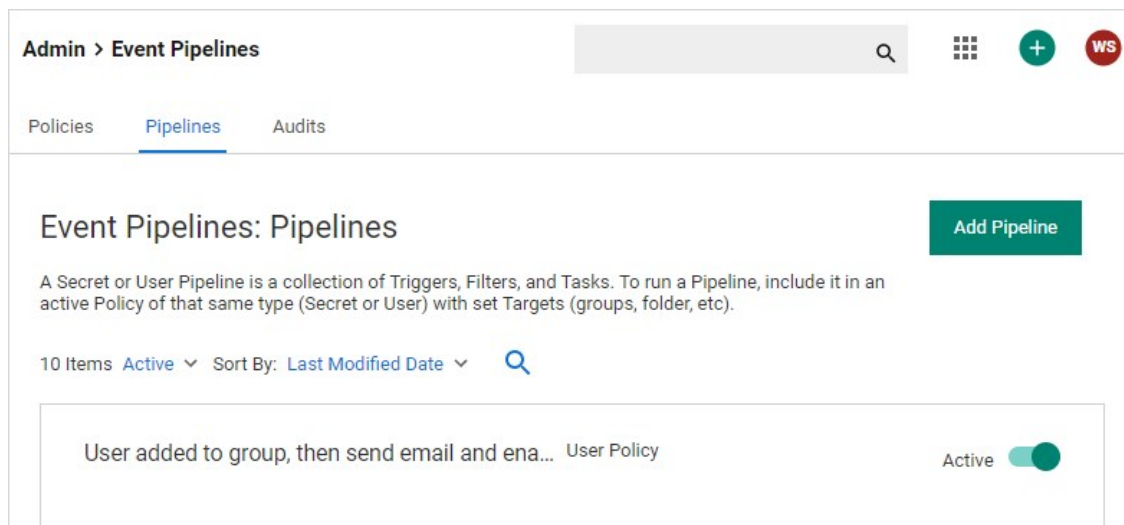
Creating New Event Pipelines

Note: You can create EPs from the Event Pipelines list (shown below) or an EP policy's details view. With the former method, you will have to add the EP to an EP policy separately. With the latter method, the EP is automatically added to the EP policy you are viewing. You can later manually add additional EPs to the policy as desired.

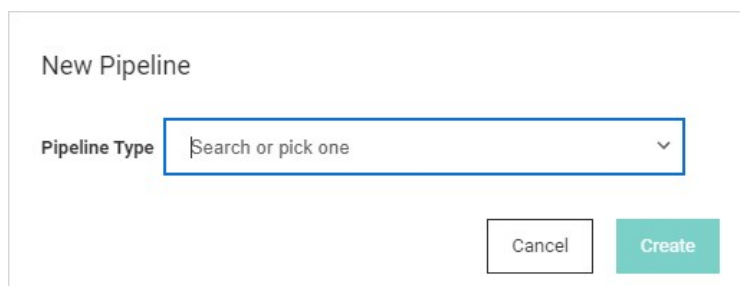
To create a new EP:

Step One: Create EP

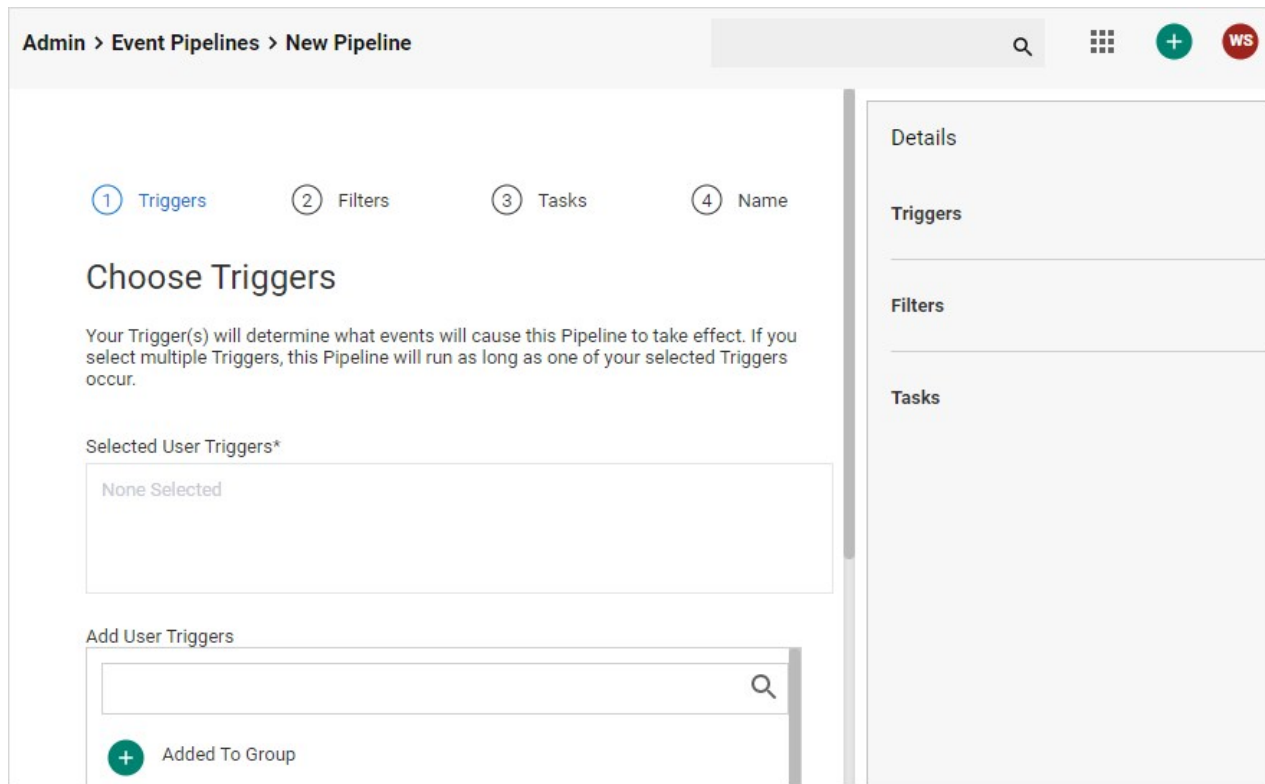
1. Go to the **Event Pipeline** page.
2. If necessary, click the **Pipelines** tab. The Event Pipeline Pipelines page appears:



3. Click the **Add Pipeline** button. The New Pipeline popup page appears:



4. Click the **Pipeline Type** dropdown list to select the EP type: **Secret** or **User**. For this instruction, we chose User.
5. Click the **Create** button. The New Pipeline wizard appears on the Choose Triggers page:



Step Two: Add Triggers

Note: When using the pre-check-in trigger, we recommend applying a group filter too. That trigger is a blocking call prior to secret check in that runs a script or causes the check in to fail with a warning. A problem arises when Secret Server does the same check-in process for the system "user" in the background at the end of the checkout interval. When the Pre-Check In trigger causes the check in to fail with a warning, the Secret Server background process continues to attempt check in forever, causing Secret Server to disable the pipeline. Applying a group filter ensures the trigger does not apply to the system user.

1. In the **Add Triggers** section, click the **+** button next to the triggers you desire. You can also search for a trigger by typing in the search text box. The selected triggers appear in the Selected Triggers list. Consider the following when selecting triggers:
 - o Currently triggers are centralized around events that are linked to a secret.
 - o You can add multiple triggers.
 - o You can limit when the EP runs by adding filters.
 - o Multiple triggers are logically ORed (not XORed) together. Each trigger is considered individually, and only one needs to apply for the EP to run—if concurrent triggers do not apply, it does not matter. If multiple triggers do apply, the EP will only run once per EP policy.

The added trigger appears in the Selected User Triggers box and the Details Triggers section:

Your trigger(s) will determine what events will cause this Pipeline to take effect. If you select multiple Triggers, this Pipeline will run as long as one of your selected Triggers occur.

Selected User Triggers*

User: Create [Remove](#)

Details

Triggers

User: Create

Filters

2. Click the **Next** button. The Choose Filters page of the wizard appears:

① Triggers ② **Filters** ③ Tasks ④ Name

Choose Filters

Your Filter(s) will limit when your Pipeline takes effect. All selected filters will apply to this Pipeline every time it runs.

Selected User Filters

No filters have been selected

Add User Filters

+

Custom Variable

Step Three: Add Filters

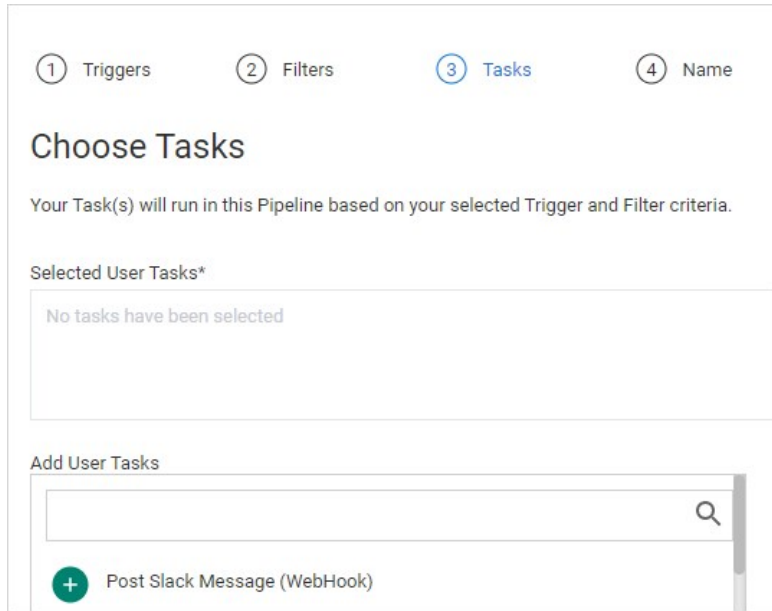
1. Use the exact same method to add filters to the EP. All filters present a popup page for you to provide additional information when you click on them. Consider the following when selecting filters:
 - Whereas triggers focused on secrets, filters can also access secret and user information.
 - Because the same filter can differ by its settings, you can add the same filter multiple times to an EP.
 - Filters are logically ANDed together—all filters apply at once and all matter.
 - Metadata (MetaData) filters are available for secrets, folders, or user (secret/folder/user hierarchy). Metadata filters can filter for group names.

The selected filters appear in the Selected User Filters section:

Selected User Filters

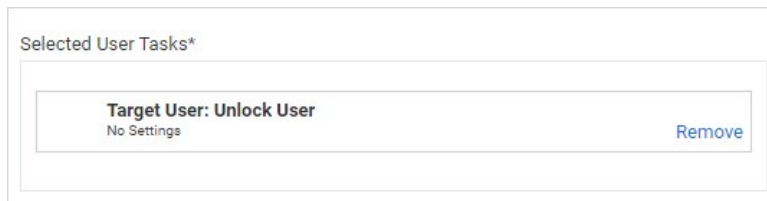
Day of Week
Day of Week: Monday; Time Zone: ; [Remove](#) [Edit](#)

2. Click the **Next** button. The Choose Tasks page of the wizard appears:



Step Four: Choose Tasks

1. Use the exact same method to add tasks to the EP. Many tasks present a popup page for you to provide additional information when you click on them. The selected user tasks appear:



2. Set the task order if you selected more than one. Tasks run in order of their appearance in the **Task** tab of the **Event Pipeline** page. To change the task running order, hover the mouse pointer over the one you want to move, and use the anchor on the left of its card to drag the task to the order you wish it to run. If a task fails, then the following tasks will not run.

Warning: Tasks are very powerful and thus can be dangerous. You can alter Secret Server in dramatic, sometimes irreversible ways. We strongly recommend testing EPs in a safe sandbox environment before applying them to production Secret Server servers.

3. Click the **Next** button. The Name Pipeline page of the wizard appears:

1 Triggers 2 Filters 3 Tasks 4 Name

Name Pipeline

Give the Event Pipeline a recognizable name, and a helpful description.

Pipeline Name

Pipeline Description

Cancel Save

4. Type the EP's name in the **Pipeline** text box.
5. Type a description of the EP in the **Pipeline Description** text box.
6. Click the **Save** button.

Editing Existing Event Pipelines

To create an EP:

1. Go to the **Event Pipeline** page.
2. If necessary, click the **Pipelines** tab. The Event Pipeline Pipelines page appears.
3. Click the title of the card representing the EP you want to edit. The EP wizard appears.
4. See [Creating New Event Pipelines](#) for instructions on using the wizard.

Viewing Event Pipelines

Because EPs are not directly tied to a single EP policy, they can be viewed through an EP policy or directly from the EP list. The EP list is a tab on the main Event Pipeline Policy page directly after navigating from the Admin page. After selecting an EP policy, its associated EPs are displayed in cards.

Event Pipeline Policies

Activating or Deactivating Event Pipeline Policies

To control if an EP policy is available, you can toggle its active status:

1. Go to the **Event Pipelines** page.

2. If necessary, click the **Policies** tab.
3. Locate the card for the EP policy you want to activate or deactivate.
4. Click the **Active/Inactive** toggle button. A confirmation popup appears.
5. Click the **OK** button. The EP policy's status is changed.

Note: The EPs belonging to the EP policy remain available to other EP policies.

Adding an Existing Event Pipeline

Note: Adding an existing pipeline enables that pipeline to be used in other policies. Only pipelines of the same type (secret or user) can be added.

Note: This does not create a copy of the existing pipeline, it creates a link. Thus, any changes to the pipeline will affect the other policies that use it.

1. Go to the **Event Pipelines** page.
2. If necessary, click the **Policies** tab.
3. Select the EP policy you want to add a pipeline to.
4. Click the **Add Pipeline** button.
5. Click the **Add Existing Pipeline** dropdown list and select the pipeline (only pipelines of the same type will show).
6. Click the **Create** button.

Assigning Folders and Secret Policies to Event Policy Targets

Folders

1. Go to the **Event Pipeline** page.
2. If necessary, click the **Policies** tab. The Event Pipeline Policies page appears.
3. Click the title of the EP policy on its card on the **Event Pipeline Policies** page. The page for that EP policy appears.
4. Click the **No Folder Selected** link in the **Targets** section. A destination page appears.
5. Click to select the check boxes for the desired target folders in the tree. Click the tiny arrow next to the check box to expand the tree. Remember, selecting a folder does *not* automatically select its subfolders.
6. Click the **Save** button.

Secret Policies

1. Click **Admin > Secret Policies**. The Secret Policy page appears.
2. Click the desired secret policy's name in the list. The Secret Policy page for that policy appears.
3. Click the **Edit** button. The list becomes editable.
4. Click the **Event Pipeline Policy** dropdown list in the **Security Setting** section and select **Enforced**.
5. Click the **Save** button. All secrets under that secret policy are now affected by the EP policy.

Creating, Importing, and Duplicating Event Pipeline Policies

Note: Newly added EP policies are deactivated by default.

1. Click **Admin > See All**.
2. Click the **Action** button and select **Event Pipeline Policy**.
3. If you plan to duplicate an existing EP policy, click the card for that policy in the **Event Pipeline Policies** list.
4. Click the **Add Policy** button, and you will be presented with the following options:
 - **Create New Policy:** Click the selection button, and type a name in the **Policy Name** text box, and optionally type a description in the **Policy Description** text box.
 - **Import Policy:** Import an exported EP policy in JSON format. This can be a policy exported from a separate Secret Server instance. Click the selection button, and paste the JSON payload in the **Add Policy** text box, click the **Create** button.
 - **Duplicate Selected Policy:** Copy an existing EP policy. Click the selection button, and then click the **Create** button. The new EP appears in the Event Pipeline Policies list.

Monitoring Event Pipeline Policies

There are two ways to monitor your EP policy:

- **Audit:** Shows changes to EP policies, targets, and EPs. Click the **Audits** tab on the **Event Pipeline Policies** page.
- **Activity:** Shows the actions each EP policy or single EP took each time it is triggered. This includes failures, skips, and successes. Click the card for the desired EP policy, and then click the **View Policy Activity** button on the right. Alternatively, you can click the title on the card. When the page for the EP policy appears, click the **Activity** tab.

Ordering Event Pipelines in Event Pipeline Policies

Event Pipelines run in order they appear in the EP policy. Since EPs can be in multiple EP policies, the order is unique to each policy. To change the EP order in the EP policy:

1. Go to the **Event Pipeline Policies** page.
2. Click the name on the card for the EP policy you want to edit. The policy's page appears on the Details tab.
3. Hover the mouse pointer over the EP you want to reorder. An anchor appears on the left of the card.
4. Drag that anchor to the desired position.

Note: If an error occurs in a policy's EP, then the following EP still runs.

Removing Event Pipelines from Event Pipeline Policies

To remove an EP from an EP:

1. Go to the **Event Pipeline Policies** page.
2. Click the name on the card for the EP policy you want to edit. The policy's page appears on the Details tab.
3. Click on an EP in the details of an EP policy. A panel appears on the right of the page.
4. Click the **Remove Pipeline** button.

Note: The button removes the EP from the EP policy, but it does not remove it from Secret Server. Other EP policies using the EP

still have access to it.

Advanced Settings and Troubleshooting

Configuring Advanced Settings

There are a few new advanced settings you can use with EP policies:

- **Event Pipeline Activity Log entries removed after (days):** The EP activity log entries stay in the log for this many days. Default value: 90.
- **Event Pipelines: Allow Confidential Secret Fields to be used in Scripts:** Allows confidential secret fields to be used in EP script, such as \$password. Default value: False.
- **Event Pipelines Infinite Loop Time (Minutes):** If an EP executes the number of times specified in the infinite loop threshold during the Infinite Loop Time period, it is marked as an infinite loop. Default Value: 5 (on premises), 20 (cloud).
- **Event Pipelines Infinite Loop Threshold:** Number of times that an EP can execute within the infinite loop time on an individual item before it is considered to be an infinite loop. Default Value: 5.
- **Event Pipelines Log Skipped Policies:** If true, the the pipeline activity log will log filtered policies runs. Default value: False.
- **Event Pipelines Maximum Script Run Time (Minutes):** Scripts ran by EP tasks are stopped after this many minutes. Default Value: 5 minutes.
- **Heartbeat: Include UnableToConnect as Heartbeat Failure Event:** Adds the ability to trigger EPs on heartbeat UnableToConnect status. When toggled to true, this setting allows the user to include UnableToConnect as part of the heartbeat failure EPs. It defaults to false.

Infinite Loops

It is possible for EPs to trigger each other over and over in an endless loop. For example:

1. Editing a secret triggers one EP to run a heartbeat on the secret.
2. The heartbeat triggers another EP to edit the secret.
3. Editing the secret triggers the original EP to run another heartbeat, restarting the cycle, creating an infinite feedback loop.

Fortunately, Secret Server detects these loops and automatically deactivates the involved EPs. So, if you have EPs that seem to be deactivating themselves, look for circular logic paths involving the EPs.

Note: By default, pipelines are configured to consider any event that executes five tasks within five minutes from the same trigger as an infinite loop. For example, "secret edit" is selected as a pipeline trigger, and "remote password change" is selected as the task. After the first edit is made on a secret, an RPC is triggered. Every time the RPC completes, a new edit is triggered, which, in turn, triggers another RPC. If this happens five times within five minutes, then an infinite loop is declared. If the RPC is slow, taking more than five minutes for five password changes to occur, then an infinite loop is **not** declared. In this case, use the "configuration advanced" page to change "event pipelines infinite loop time (minutes)" to a longer time.

Note: Please click the table of contents on the left to see the sub-pages to this one. Click the table of contents on the right to see headings on this page.

Event subscriptions trigger notifications of defined events within the system. These notifications are sent to the inbox and from there can be sent externally via email or Slack.

Note: The notifications are an alert of specific events and not for archived reporting.

Go to **Admin > Event Subscriptions** to view the Event Subscriptions page:

The screenshot shows the 'Admin > Event Subscriptions' page. At the top, there is a search icon, a grid icon, a green plus icon, and a red 'WS' icon. Below the header, there are two tabs: 'Subscriptions' (active) and 'Audit'. The main content area shows '3 Items' and a toggle for 'Include Disabled'. A green button labeled 'Create Event Subscription' is on the right. Below this is a table with the following data:

NAME	ACTIVE
App Settings Exported	Yes
Event Stuff	Yes
Secret Creation	Yes

Click the name of one of the subscriptions to see its page:

Event Subscriptions [Edit](#)

Meant to trigger notifications of defined events within the system. These notifications will be sent to the inbox and from there sent externally via email or slack. These notifications are an alert of specific events and not intended to be used for archived reporting.

Events [Edit](#)

Any of these events will trigger a notification.

Subscribers [Edit](#)

Users explicitly defined or as a member of a subscribed group will received a notification in their inbox. Communication to external emails or other channels is defined by inbox notification rules.

Associated Rules

These rules target this specific event subscription. They can be updated or reviewed to change the actions, email template, or other communication preferences. Please note that notifications could target conditions and not the specific event subscription and those rules may not appear in this list.

Name Secret Creation

Active Yes

Inbox Expiration 7

ENTITY	ACTION	CONDIT...	TARGET	
Secret	Create	This f...		

NAME		

[Secret Creation](#)

The Event Subscription Page includes:

- **Event Subscriptions:** The name of the event subscription, whether it is active, and how long notifications from last in the inbox before expiration.
- **Events:** .
- **Send Email with High Priority:** Sends the email for this subscription with high priority set.
- **Subscribed Events:** List of the events that are contained in this subscription.
- **Subscribed Users:** List of the Secret Server users and groups subscribed to this event.
- **Subscription Name:** Name for the subscription.

Creating Event Subscriptions

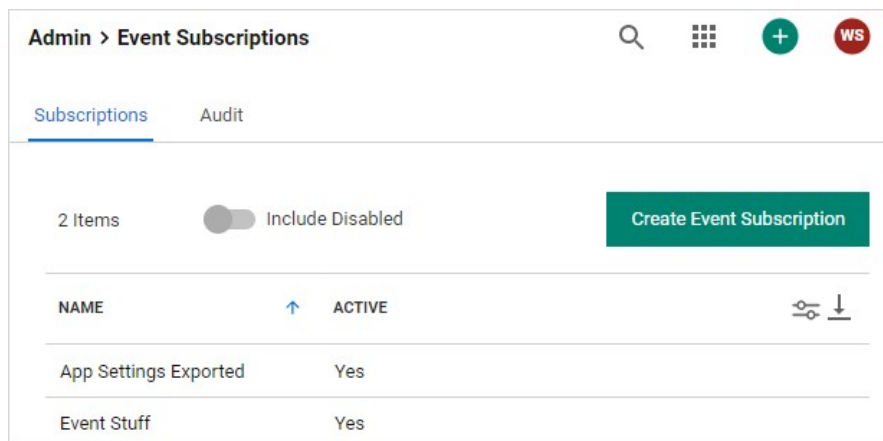
Event subscriptions trigger notifications of defined events within the system. These notifications are sent to the inbox, which may send them externally via email or Slack, depending on your configuration.

Note: These notifications are an alert of specific events and not intended to be used for archived reporting.

To add an event subscription:

Task 1: Creating an Event Subscription

1. Navigate to **Administration > Event Subscriptions**:



2. Click the **Create Event Subscription** button. The Create Event Subscription page appears:

Create Event Subscription

Event Subscription Name

Send Email

Send Slack

Events

Search or pick one ▾

Search or pick one ▾

Add Event

Add conditions to this event subscription to indicate which events should trigger a notification.

Cancel
Create Event Subscription

3. In the **Subscription Name** text box, enter a name for this new event subscription.
4. Click to select the **Send Email** check box if you want to send an email via an inbox notification.
5. Click to select the **Send Slack** check box if you want to send a Slack message via an inbox notification.

Task 2: Adding Events

Create the events that trigger notifications:

1. Click the **Events** dropdown list to select an event object to trigger a notification.
2. Click the second **Events** dropdown list to select the event for the chosen object. For some events, one or more follow-on dropdown lists may appear. For example, if you chose Secret and then Create, another dropdown list would appear for you to select whether you want all secrets, those in the selected folder or those in the selected folder and its subfolders. Similarly, if you chose those in a folder, a link appears for you to choose the folder.
3. Click the **Add Event** button to add the event. An event table appears:

ENTITY	ACTION	CONDITIONS	TARGET	⊞
Secret	Create	In this folder		Remove

- Add more events as desired.
- Click the **Create Event Subscription** button. The event subscription's page appears:

Admin > Event Subscriptions > Secret Creation 🔍 🏠 + WS

Event Subscriptions [Edit](#)

Meant to trigger notifications of defined events within the system. These notifications will be sent to the inbox and from there sent externally via email or slack. These notifications are an alert of specific events and not intended to be used for archived reporting.

Name

Secret Creation

Active


Yes

Inbox Expiration

7

Events [Edit](#)

Any of these events will trigger a notification.

ENTITY	ACTION	CONDITIONS	TARGET	⚙️
Secret	Create	This folder Only		

- If you want to change the status from active to inactive or adjust the number of days before an event notification expires, click the **Edit** link next to **Event Subscriptions** and make changes.

Task 3: Adding Subscribers

Subscribers are users that are explicitly defined or are a subscribed group member. They receive a notification in their inbox when this event subscription is triggered.

Note: Communication to external emails or other channels is defined by inbox notification rules.

- Scroll down the the **Subscribers** section:

Subscribers [Edit](#)

Users explicitly defined or as a member of a subscribed group will receive a notification in their inbox. Communication to external emails or other channels is defined by inbox notification rules.

2. Click the **Edit** link. Three buttons appear.
3. Click the **Add** button. The Users & Groups popup page appears:

The screenshot shows a 'Users & Groups' dialog box. At the top is a search bar with the placeholder text 'Search...'. Below the search bar is a dropdown menu currently set to 'All' with a sub-menu 'Users & Groups' visible. The main area contains a list of system groups, each with an unchecked checkbox and a name followed by its domain in parentheses. The groups listed are: Access Control Assistance Operators (gamma.thycotic.com), Account Operators (gamma.thycotic.com), Administrators (gamma.thycotic.com), Administrators (testparent.thycotic.com), Allowed RODC Password Replication Group (gamma.thycotic.com), Auditors, Backup Operators (gamma.thycotic.com), Cert Publishers (gamma.thycotic.com), and Certificate Service DCOM Access (gamma.thycotic.com). At the bottom right of the dialog are two buttons: a 'Cancel' button and a green 'Add' button.

4. Type the name of the user or group you want to add in the **Search** text box.
5. Click to select the check box next to the user or group that remains.
6. Repeat the process for additional users or groups.
7. Click the **Add** button. The new subscriber appears in a table in the section.

Task 4: Associating Inbox Rules

Inbox rules filter a specified event subscription (or other alerts)—in this case, the one you just created. When you create an event subscription, an inbox rule based on the event subscription system rule is automatically created. It can be updated or reviewed to change the actions, email template, or other communication preferences.

The event subscription subscribers defines who *potentially* receives the event alert. The associated inbox rules filters which events are shared with those subscribers via Slack or email messages. Inbox rules search for specified text strings in specified locations in the incoming notification.

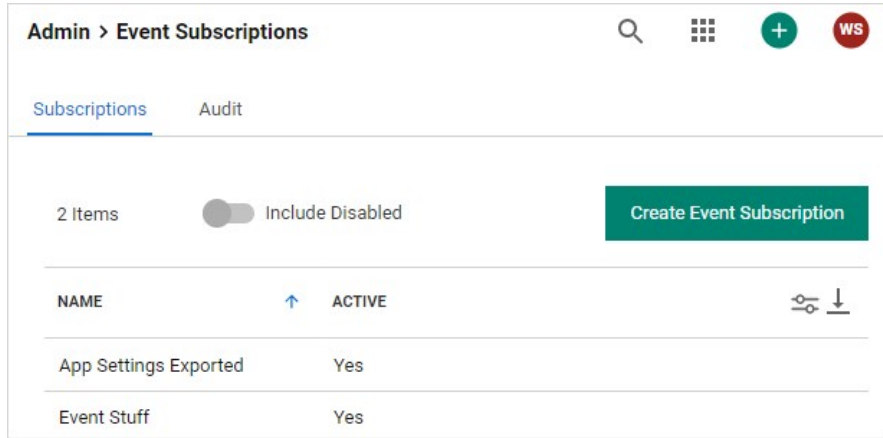
Note: Notifications can be triggered by defined conditions and not a specific event subscription. Those rules may not appear in this list.

1. Scroll down to the **Associated Rules** section. Note that a link to your new event subscription inbox rule already appears at the bottom.
2. If you want to edit the rule, click the link for the inbox rule. See [Using Inbox Rules](#).

Deleting an Event Subscription

To delete an event subscription:

1. Navigate to **Administration > Event Subscriptions**:



2. Click the subscription name link.
3. Click **Delete** on the following page.

Editing an Event Subscription

To edit an event subscription:

1. Navigate to **Administration > Event Subscriptions**, click the subscription name, and then **Edit**.
2. To remove a subscribed user, group, or event, click the button next to the entry in the appropriate list.
3. To add entries to either list, see [Creating Event Subscriptions](#).
4. Click **Save** to save all changes.

Event List

The following events are available:

Table: Folder Events

Create	All, In this Folder
Delete	All, For this Folder, In this Folder
Edit Permissions	All, For this Folder, In this Folder
Secret Policy Change	All, For this Folder, In this Folder

Table: Secret Events

Access Approved	All, For this Secret, In this Folder
Access Denied	All, For this Secret, In this Folder
Cache View	All, For this Secret, In this Folder
Check In	All, For this Secret, In this Folder
Check Out	All, For this Secret, In this Folder
Copy	All, For this Secret, In this Folder
Create	All, In this Folder
Custom Audit	All, For this Secret, In this Folder
Custom Password Requirement Added to Field	All, For this Secret, In this Folder
Custom Password Requirement Removed from Field	All, For this Secret, In this Folder
Delete	All, For this Secret, In this Folder
Dependency Added	All, For this Secret, In this Folder
Dependency Deleted	All, For this Secret, In this Folder
Dependency Failure	All, For this Secret, In this Folder
Edit	All, For this Secret, In this Folder
Expired Today	All, For this Secret, In this Folder

Expires in 1 Day	All, For this Secret, In this Folder
Expires in 3 Days	All, For this Secret, In this Folder
Expires in 7 Days	All, For this Secret, In this Folder
Expires in 15 Days	All, For this Secret, In this Folder
Expires in 30 Days	All, For this Secret, In this Folder
Expires in 45 Days	All, For this Secret, In this Folder
Expires in 60 Days	All, For this Secret, In this Folder
Export	All, For this Secret, In this Folder
File Save	All, For this Secret, In this Folder
Heartbeat Failure	All, For this Secret, In this Folder
Heartbeat Success	All, For this Secret, In this Folder
Hook Create	All, For this Secret, In this Folder
Hook Delete	All, For this Secret, In this Folder
Hook Edit	All, For this Secret, In this Folder
Hook Failure	All, For this Secret, In this Folder
Hook Success	All, For this Secret, In this Folder
Launch	All, For this Secret, In this Folder
Password Change Maximum Attempts Reached	All, For this Secret, In this Folder
Password Copied to Clipboard	All, For this Secret, In this Folder
Password Displayed	All, For this Secret, In this Folder
Password Change	All, For this Secret, In this Folder
Secret Policy Change	All, For this Secret, In this Folder
Session Recording View	All, For this Secret, In this Folder
Undelete	All, For this Secret, In this Folder
View	All, For this Secret, In this Folder

View Secret Edit	All, For this Secret, In this Folder
Web Password Fill	All, For this Secret, In this Folder

Table: User Events

Added to Group	All, For this User, For this Group
Challenge Applied	All, For this User
Challenge Cleared	All, For this User
Create	-
Disabled	All, For this User
Edit	All, For this User
Enable	All, For this User
Lockout	All, For this User
Login	All, For this User
Login Failure	All, For this User
Logout	All, For this User
Owners Modified	All, For this User
Password Change	All, For this User
Removed From Group	All, For this User, For this Group
Two Factor Changed	All, For this User

Table: Other Events

Automatic Export	Download, Edit, Export, Run Export
Configuration	Edit
Dual Controls	Create, Delete, Edit

Encryption	HSM Disable, HSM Enable, Rotate Secret Keys, Rotate Secret Keys Cancel Requested, Rotate Secret Keys Failure, Rotate Secret Keys Success
Engine	Engine Activated, Create, Deactivate, Delete, Offline, Online
Export Secrets	Exported
Group	Owner Modified
Import Secrets	Imported
IP Address Range	Create, Delete, Group Assigned, Group Unassigned, Edit, User Assigned, User Unassigned
Licenses	Expires in 30 Days
Role	Assigned User or Group, Create, Edit, Role Disabled, Role Enabled, Unassigned User or Group
Role Permission	Added to Role, Removed From Role
Script - PowerShell	Create, Deactivate, Edit, Reactivate, View
Script - SQL	Create, Deactivate, Edit, Reactivate, View
Script - SSH	Create, Deactivate, Edit, Reactivate, View
Secret Policy	Create, Edit
Secret Template	Create, Create Secret Access Changed, Edit, Field Encrypted, Field Exposed, Owners Modified, Copy
Privileged Behavior Analytics Configuration	Edit
Site	Engine Added, Domain Assigned to Site, Create, Disable, Edit, Enable, Engine Downloaded, Engine Offline, Engine Online, Domain Removed from Site, Engine Removed
Site Connector	Create, Credential View, Disable, Edit, Enable
Unlimited Administrator	Disable, Enable
User Audit	Expire Now

Overview

When the database becomes inaccessible, Secret Server will try to log errors to the Windows event log. By default, network service and standard service accounts will not have permissions to the event log. Permissions must be added to specific event log registry keys.

Required Registry Permissions

The follow permissions are required for the identity configured on the Secret Server application pool in IIS:

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog

Applies to key and subkeys

- Read permissions:
 - Query Value
 - Enumerate Subkeys
 - Notify
 - Read Control
- Set Value permission
- Create Subkey permission

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > Security

Applies to key and subkeys

Read permissions:

- Query Value
- Enumerate Subkeys
- Notify
- Read Control

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > State

Applies to key and subkeys. Only applies to Windows Server 2019.

Read permissions:

- Query Value
- Enumerate Subkeys
- Notify
- Read Control

Applying Windows Event Log Permissions

1. Determine the account that is running Secret Server:
 1. Log on Secret Server.
 2. Go to **Admin > Diagnostics**.

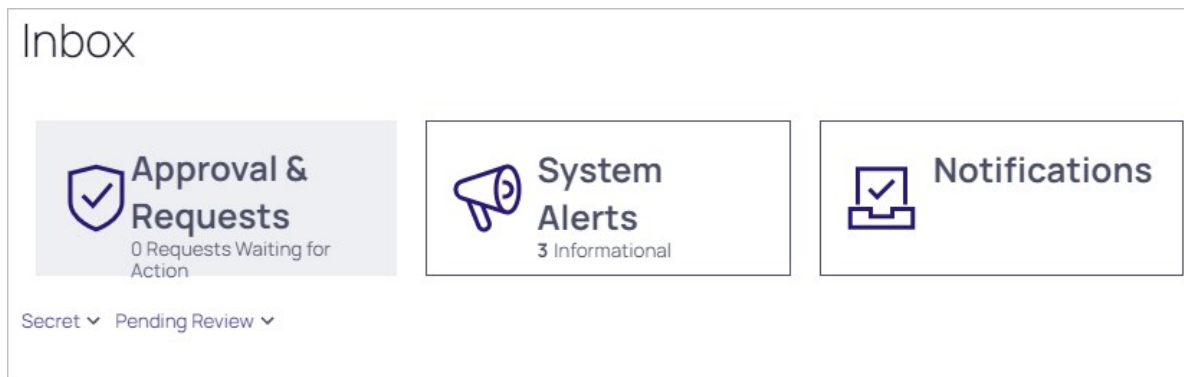
3. Look for any of the **Thread Identity** labels. These contain the identity of Secret Server (often NT AUTHORITY\NETWORK SERVICE or IIS APPPOOL\SecretServer or the service account set up for IWA. See [Running the IIS Application Pool As a Service Account](#).

Note: You can also determine the identity by logging in and navigating to <http://yoursecretserverurl/Installer.aspx>. The first step of this page will tell you the application pool identity.

2. Open the Windows registry editor on the machine running Secret Server (regedit at the command prompt or Window search text box).
3. On the left, navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog**.
4. Right click the **EventLog** folder in your registry editor and select **Permissions**. A permissions dialog box appears.
5. Click the **Advanced** button.
6. On the **Permissions** tab, Click the **Add** button. A Permission Entry dialog appears.
7. Click the **Select a principal** link. The Select User, Computer... dialog box appears.
8. Find the account running Secret Server, such as Thycotic_Service (svc_thycotic@test.com).
9. Click the **OK** button. The dialog box closes.
10. In the **Basic Permissions** section of the **Permission Entry** dialog, click to select the **Read** check box.
11. Click the **Show advanced permissions** link. The pane switches.
12. Click to select the **Set Value** and **Create Subkey** check boxes in the **Advanced Permissions** section.
13. Click **OK** buttons on the remaining dialogs to apply the permissions. You are returned to the main registry editor window.
14. Navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > Security**, right-click and select "**Permissions...**"
15. Right click **Security** folder and select **Permissions**. A permissions dialog box appears.
16. Click the **Add** button.
17. Find the account running Secret Server.
18. Click the **OK** button.
19. Click to select the **Read** check box in the Allow column.
20. Click the **OK** button to apply the permission.
21. If you are running Windows Server 2019, use the same procedure to add Read permission to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > State**.

The *Inbox* page shows notifications such as event subscription alerts, access requests and approvals, and other configuration alerts in a single interface. In addition to viewing notifications in the inbox, you can configure the inbox to forward them via email or Slack, subject to numerous configurable criteria. You can also customize the format of the email messages. You access the inbox by clicking the **Inbox** button on the main menu. See the subordinate topics for details.

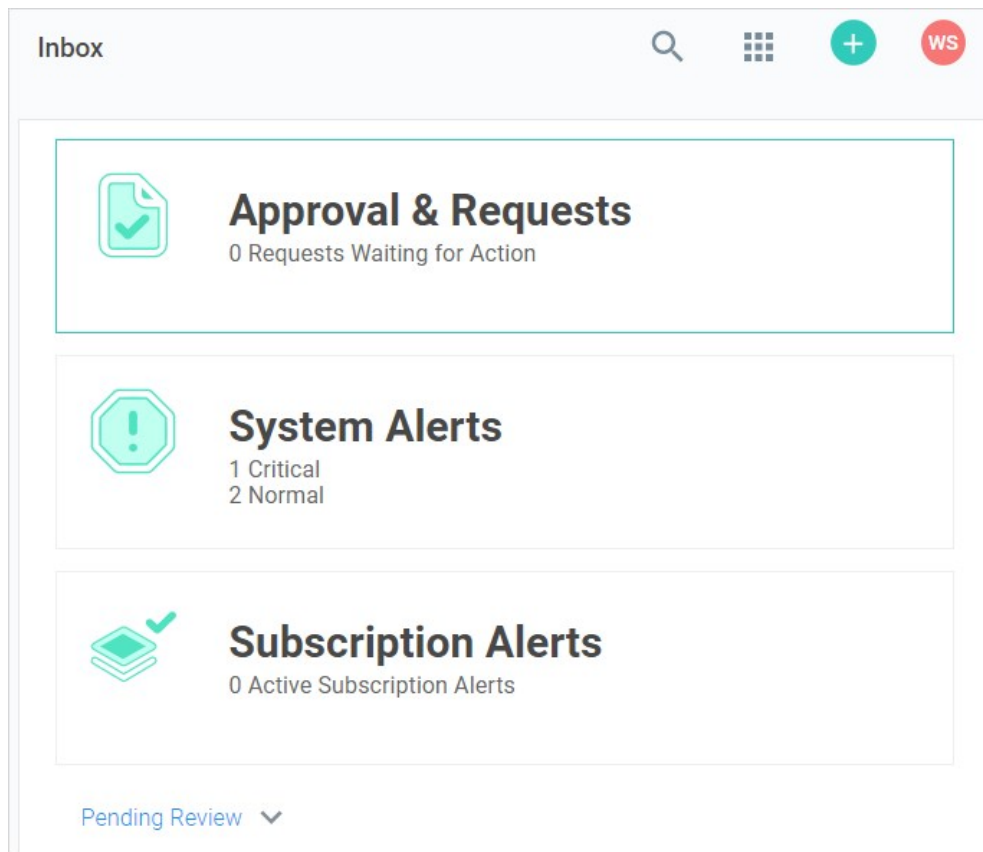
Note: The Inbox was also called the *Alert Notification Center* in earlier Secret Server versions.



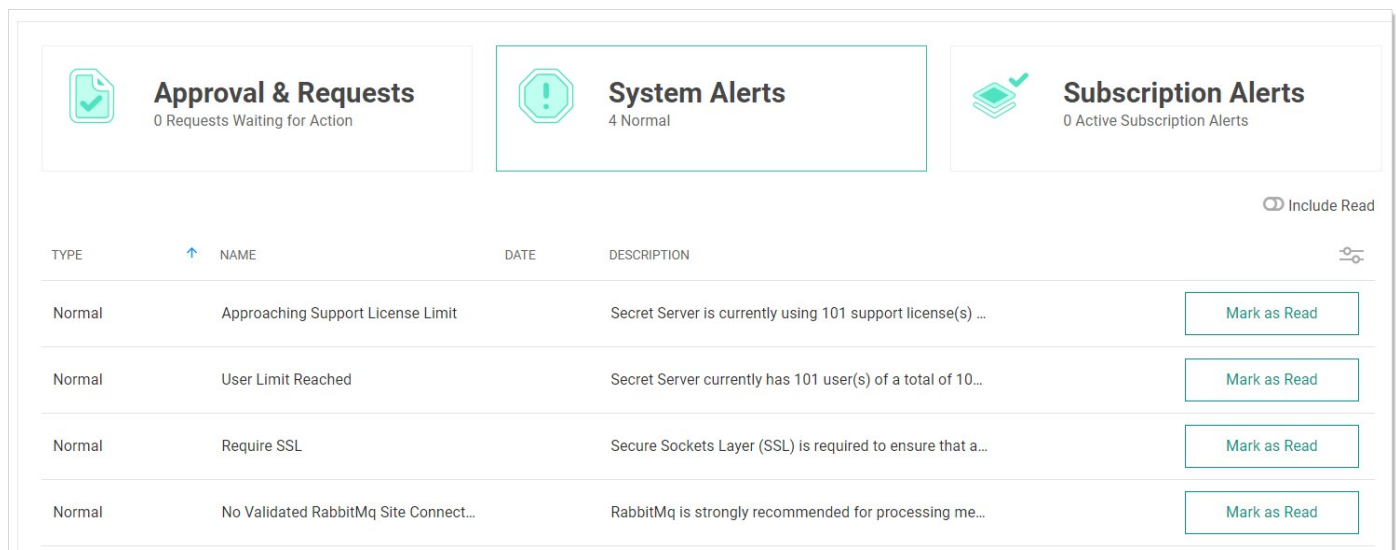
Event subscriptions disappear from the notification center after you view them. System alerts and access requests stay active until resolved.

Marking Alerts as Viewed

1. Access the alert notification center by clicking the **Inbox** button on the main menu. The Inbox appears:



2. Click the **System Alerts** button. The System Alerts page appears:



3. Click the **Mark as Read** button for the each alert you no longer wish to view. The alert disappears, but you can still see it if you click the **Include Read** toggle button.

Using Inbox Rules

Overview

An inbox rule (notification rule) triggers on notifications and sends either an email or a Slack message to a specified group of users. First, we discuss an inbox rule's components. Second, we show how to create an inbox rule from scratch and based on an existing notification.

Note: There are some emails types that Secret Server sends that do not go through the inbox. For example, the test email button on the email configuration page sends a plain-text email directly. There are also some diagnostic emails that are directly sent. For example, discovery can directly send a detailed log which is essentially a text dump. Inbox rules are primarily for non-admin end-user communications and event subscriptions.

Note: You can still send legacy emails (earlier email notifications that did not go through the inbox) if desired. Go to **Admin > Configuration > Email Tab > Enable Legacy Email** to set this up.

Inbox Rule Components

Inbox rules have the following components:

Message (Notification) Types

This is the notification message (alert) types that the rule responds to. These include:

- Dependency Failure
- Event Subscription
- Inbox Test Message
- Password Reset
- Secret Access Approved
- Secret Access Cancel Request
- Secret Access Deny Request
- Secret Access Request
- Secret Changed
- Secret Heartbeat Failed
- Secret View
- Workflow Access Approval Request
- Workflow Access Request Expired
- Workflow Access Request Incomplete
- Workflow Access Request Next Step
- Workflow Access Step Approved

Rule Conditions

Rule conditions are filters that define who receives the email or Slack message when a notification arrives. Conditions are matched with text string matching: equals, not equals, or regex. If no condition exists, the rule triggers for every message of the defined types. Condition types include:

- **ActionType:** Specific entities via text matching of the action's value or display value. Actions types vary by inbox message type. For example, "EXPORTED."

Note: In the case of an event subscription notification, these are the same as the event subscription events. Other notification types have different action types.

- **Container:** Specific containers via text matching of the container name. Containers include folders (secret containers) and roles (quasi user containers).

- **Details:** Specific text string in the details section. The Details type serves as a summary of the message for display in the inbox. Sometimes it contains information that is from other condition types. For example: "App Settings Exported - SECRETSERVERSETTINGS - EXPORTED."
- **EntityType:** Specific entities via text matching of the entity name or value. Entities are what is having the action done to it, for example, "SECRETSERVERSETTINGS." These are the same as the event subscription entities.
- **EventDetails:** Specific text string in the Event Details section. For example: "Application Settings," "Launcher Settings," and "Protocol Handler Settings."
- **ItemName:** The source item. Specific items via text matching of the item's value or display value. The "item" varies by message. For an event subscription secret action it contains the secret name. For an event subscription folder action, it contains the folder name.
- **SubscriptionName:** The source event subscription. Specific subscriptions via text matching of the subscription's value or display value.
- **User:** The user that created the rule. Specific users via text matching of the username's value or display value.
- **Rule Subscribers:** What users receive the action result (an email or Slack message).
- **Rule Actions:** What actions the rule performs:
 - Send to an email address using a specific HTML template, which the user defines.
 - Send to Slack using a specific markdown template, which the user defines.

Predefined System Rules

Secret Server ships with several predefined system rules in inbox templates. These rules can only be disabled or enabled. However, you can copy a system template to your own custom template and edit that. This allows us to upgrade system rules without interfering with your customizations. The predefined system rules are:

- Dependency Failure
- Inbox Test Message
- Password Reset
- Secret Access Deny Request
- Secret Access Request
- Secret Changed
- Secret Heartbeat Failed
- Workflow Access Approval Request

Example Rule Diagram

Note: This example diagram is specific to event subscription notifications. There are many other types that differ slightly, especially in content.

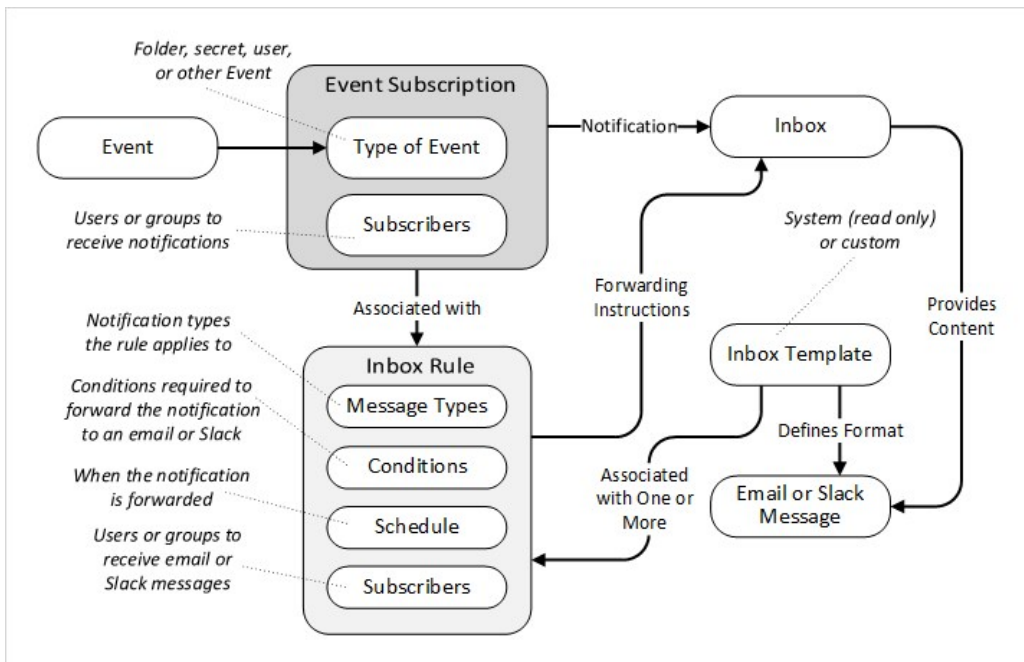
In the following diagram:

1. A secret triggers an event matching an event subscription.
2. Secret Server notifies the users and groups on the event subscription subscribers list. The notifications appear in their inboxes.
3. The inbox rule associated with the event subscription evaluates the conditions for forwarding the notification via email or Slack.
4. The inbox rule checks its schedule to determine when to forward the message.
5. The inbox rule checks its subscribers list to determine who to forward the message to.

Note: Remember, arriving notifications may have already been filtered by whose inbox gets the notification in the first place. Thus, the inbox rule could be set to sent to everybody but only those who receive the notification in their inbox will receive the email or Slack message.

- The inbox rule references the message type's inbox template to format and populate the message's variables.
- When the scheduled time arrives, which can be immediately, Secret Server sends the messages to the subscribers.

Figure: Event Subscription Using an Inbox Rule to Forward a Notification via the Inbox



Procedures

Creating a Rule from Scratch

Let us say we want to be notified when anybody tries to edit the permissions on the "No-Go Secrets" folder. This is an event, so the notification type will be an event subscription.

Task 1: Create the Inbox Rule

- Go to **Admin > See All**.
- Click the **Notification Rules and Templates** link. The Notification Settings page appears:

Inbox > Notification Settings

Rules Templates Resources

Send Test Notification

18 Items Include Inactive Create Rule

RULE NAME	ACTIVE	DIGEST	SYSTEM	USAGE (L...	NOTIFICATION TYPES
App Settings Exported	Yes	No	No	0	Event Subscription
Dependency Failure	Yes	No	Yes	0	Dependency Failure
Event Subscription	Yes	No	Yes	13	Event Subscription

3. Click the **Create Rule** Button. The Create Rule popup appears:

Create Rule

Rule Name *

Message Types *

- Dependency Failure
- Event Subscription
- Inbox Test Message
- Password Reset
- Secret Access Cancel Request
- Secret Access Deny Request
- Secret Access Request
- Secret Changed
- Secret Heartbeat Failed
- Workflow Access Approval Request
- Workflow Access Request Next Step
- Secret Access Approved
- Secret View
- Workflow Access Request Expired
- Workflow Access Request Incomplete
- Workflow Step Approved





Active

Action Email

Cancel Create Rule

4. Type the new rule's name in the **Rule Name** text box.
5. Click to select the **Message Types** check box for the message types you wish to apply the new rule to.
6. Ensure the **Active** check box is selected if you want to use the rule right away.
7. Click to select the **Action** selection button for the type of notification.
8. If necessary, scroll down to the bottom of the popup.
9. Click the **Template** dropdown list to select the desired inbox template to associate the rule with.
10. Click the **Create Rule** button. The configuration page for the new rule appears:

Inbox > Notification Settings > Rules > My Notification Rule

General
Subscribers
Log

Copy Rule

Rule Details [Edit](#)

An inbox rule is a set of conditions that trigger based on inbox message data. The result of the rule allows the message to be delivered based on defined rule conditions either immediately or on a schedule. This rule will apply to any inbox message type selected.

Rule Name *	My Notification Rule
Active	Yes
Message Types *	Event Subscription

Conditions [Edit](#)

These conditions must all be met for this rule to run.

This rule currently does not have any conditions defined. This means it will trigger for every message type defined above.

Schedule [Edit](#)

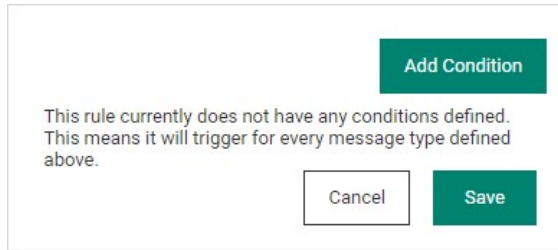
Define when this rule is evaluated. Immediate rules will run for a single message as soon as it is delivered to the inbox. Scheduled messages can be defined as a digest of all messages that meet the rule conditions during the time frame defined by the schedule or can still be single detailed messages.

Immediate	Yes
------------------	-----

Task 2: Add Rule Conditions

1. Click the **Edit** link for the **Conditions** section to add conditions.

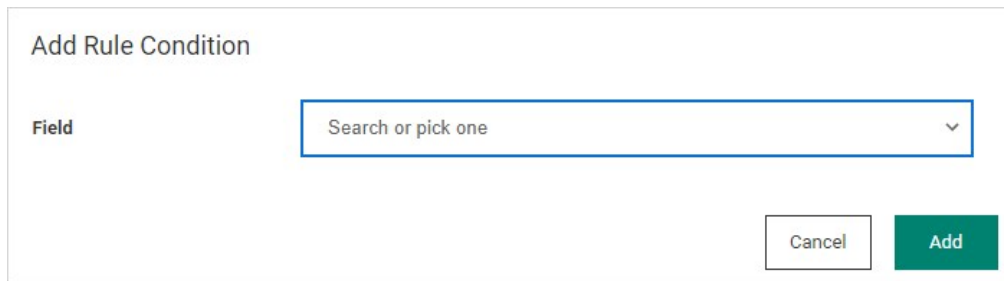
Note: If you do not add any conditions, the rule will apply to all messages of the types you chose. New buttons appear:



This rule currently does not have any conditions defined. This means it will trigger for every message type defined above.

Cancel Save

2. Click the **Add Condition** button. The Add Rule Condition popup appears:

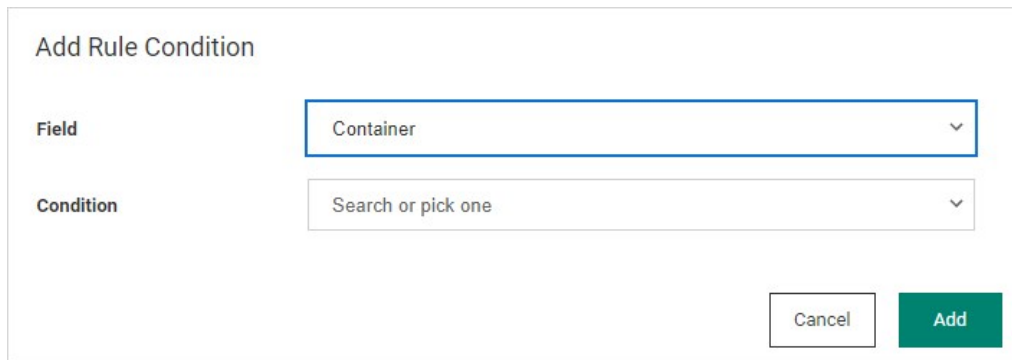


Add Rule Condition

Field Search or pick one

Cancel Add

3. Click the **Field** dropdown list to select the type of rule condition. For this instruction, we chose Container (a folder). A Condition dropdown list appears:



Add Rule Condition

Field Container

Condition Search or pick one

Cancel Add

4. Click the **Condition** dropdown list to select how the value (added next) is compared to the message. Choose "Value RegEx" if you want to create a regular expression to further refine the condition. We chose Value Equals. The Value text box appears:

Add Rule Condition

Field

Condition

Value

5. Type the string you want to test for in the **Value** text box.
6. Click the **Add** button. The new condition appears:

Container Equals No-Go Secrets [Edit](#) [Delete](#)

It tells us the rule is triggered if there is an associated event.

7. Click the **Save** button.
8. Click the **Edit** link in the **Schedule** section. The section becomes editable:

Schedule

Define when this rule is evaluated.
 Immediate rules will run for a single message as soon as it is delivered to the inbox. Scheduled messages can be defined as a digest of all messages that meet the rule conditions during the time frame defined by the schedule or can still be single detailed messages.

Immediate

Note: Any schedule choice other than *Immediate* produces a digest (summary) of notifications. Users receiving a digest can click on individual entries to see the notification. The following instructions show how to set up a digest.

9. If you want to send one email per notification, ensure the Immediate check box is selected, and skip to Task 4.

Task 3: Set up an Email Digest

1. If you do not want the notifications sent immediately, click to deselect the Immediate check box. A time setting section appears:

2. Click the **Timezone** dropdown list to select a time zone for the scheduled notification.
3. Click the clock icon to select a time to add. A time setting table appears:

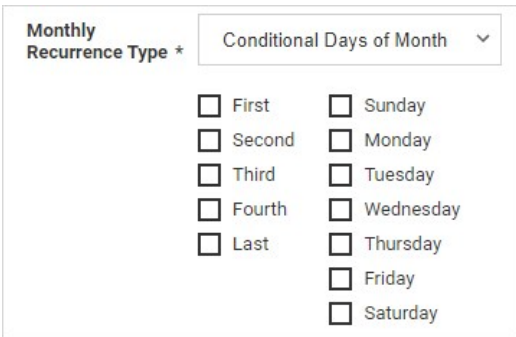
12	00	AM
	30	PM
<input type="button" value="OK"/>		

4. Hover the mouse over the one of the columns for a scrollbar appears. Click or drag to select the hour, half hour, or AM/PM.

Note: You also can simply type the time prior to clicking the clock icon in the format hh:mm AM/PM. If you choose minutes other than 00 or 30, it will be converted to the nearest 00 or 30 when you input it.

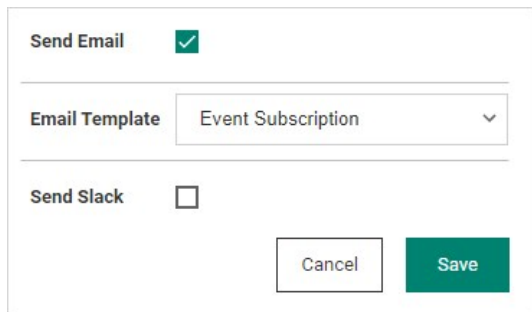
5. Click the **OK** button. The time appears in the text box.
6. Click the **Add Another Time** link to commit the time.
7. Add more times as desired.
8. Click the **Recurr** dropdown list to choose the frequency. The Every section will change, depending on what you chose:
 - o Daily: Type the number of days that pass between notifications in the **Days** text box.
 - o Weekly: Type the number of weeks that pass between notifications in the **Weeks** text box. Click to select the days of the week check box to select which days to send the notifications.

- Monthly: Click the **Monthly Recurrence Type** dropdown list to select either specific or conditional days of the month. The former provides a calendar to choose which days. The latter provides check boxes for selecting relative days where you choose the day of the week and the position in the month, for example, Last and Friday.



The screenshot shows a dropdown menu titled "Monthly Recurrence Type *". The selected option is "Conditional Days of Month". Below the dropdown, there are two columns of checkboxes. The first column contains "First", "Second", "Third", "Fourth", and "Last". The second column contains "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", and "Saturday". All checkboxes are currently unchecked.

- Click the **Starting** calendar icon to select the date when you want to start the notification schedule.
- Click the **Save** button to commit your choices.
- Scroll down to the **Actions** section.
- Click the **Edit** link next to **Actions**. The section becomes editable:

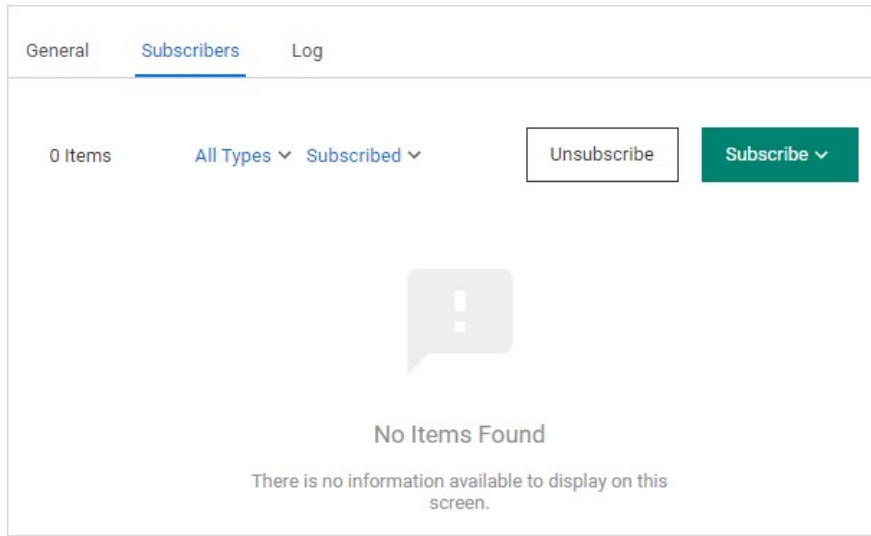


The screenshot shows a configuration form for the "Actions" section. It has three main sections: "Send Email" with a checked checkbox, "Email Template" with a dropdown menu showing "Event Subscription", and "Send Slack" with an unchecked checkbox. At the bottom, there are two buttons: "Cancel" and "Save".

- Click to select either the **Send Email** or **Send Slack** check box (or both).
- If you chose to send email, click the **Email Template** dropdown list to choose the email format (inbox template). There are several standard ones, and you can customize your own. For this instruction, we choose **Standard Email** if did not create a digest and **Standard Email Digest** if we did. See [Using Inbox Templates](#) for details.
- Click the **Save** button

Task 4: Add Subscribers to the Email or Slack Message

- Click the **Subscribers** tab:



2. Click the Subscribe button and select **Users, Groups, or External Emails**. Users provides a list of users for you to select from. Groups provides a list of groups for selection. External Emails provides a text box to enter a specific email to somebody without a Secret Server account. For this instruction, we added one of each:

Note: External emails get sent without regard to whether somebody has a notification in their inbox because external "users" do not have an inbox.

The screenshot shows the 'Subscribers' page with 3 items. At the top, it says '3 Items' followed by two dropdown menus: 'All Types' and 'Subscribed'. There are two buttons: 'Unsubscribe' and 'Subscribe'. Below the buttons is a table with the following data:

NAME	↑	TYPE	STATUS	⚙️ ↓
Backup Operators		Group	Subscribed	Remove
mister_jones@email...		External Email	Subscribed	Remove
[Name]		User	Subscribed	Remove

3. You can return to this page at a later date to edit this list. You can also unsubscribe users that are members of a subscribed group without actually removing them from the list for ease of subscribing them later.

Creating an Inbox Rule from a Notification

If an inbox notification is what you want to forward to an email or Slack message from here on out, we provide a shortcut feature that allows you to quickly build an inbox rule from the notification. To create the inbox rule:

1. Click a notification in the inbox. A details popup appears. For instance:

Event Subscription

[Details](#) Notification Rules

Created Date	7/6/2021 04:51 PM
SubscriptionName	App Settings Exported
EntityType	SECRETSERVERSETTINGS
ActionType	EXPORTED
User	James.Johnson@delinea.com
EventDetails	Application Settings, Launcher Settings, Protocol Handler Settings (Install-Time), Permission Options, User Interface, User Experience, Advanced Settings, Folder Settings, Local User Passwords, Email, Security, Ticket System, Session Recording, Login, SAML, Licenses
Details	App Settings Exported - SECRETSERVERSETTINGS - EXPORTED

[Quick Create Rule](#) [Close](#)

2. If you desire an inbox rule to react to that sort of message, click the **Quick Create Rule** button. A very similar, editable page appears:

Event Subscription

Details
Notification Rules

Rule Name *	<input style="width: 95%;" type="text"/>	
Active *	<input checked="" type="checkbox"/>	
Created Date	7/6/2021 04:51 PM	
SubscriptionName	App Settings Exported	<input type="checkbox"/>
EntityType	SECRETSERVERSETTINGS	<input type="checkbox"/>
ActionType	EXPORTED	<input type="checkbox"/>
User	[Redacted]	<input type="checkbox"/>
EventDetails	Application Settings, Launcher Settings, Protocol Handler Settings (Install-Time), Permission Options, User Interface, User Experience, Advanced Settings, Folder Settings, Local User Passwords, Email, Security, Ticket System, Session Recording, Login, SAML, Licenses	<input type="checkbox"/>
Details	App Settings Exported - SECRETSERVERSETTINGS - EXPORTED	<input type="checkbox"/>

Email
 Slack

3. Type the new rule's name in the **Rule Name** text box.
4. Click to select the check box for each rule component you want to include.
5. Click to select the **Action** selection button for the type of notification.
6. If necessary, scroll down to the bottom of the popup.
7. Click the **Template** dropdown list to select the desired inbox template to associate the rule with.
8. Click the **Add Rule** button. The configuration page for the new rule appears:

Inbox > Notification Settings > Rules > Exported Settings Inbox Rule

General Subscribers Log


[Copy Rule](#)

Rule Details [Edit](#)

An inbox rule is a set of conditions that trigger based on inbox message data. The result of the rule allows the message to be delivered based on defined rule conditions either immediately or on a schedule. This rule will apply to any inbox message type selected.

Conditions [Edit](#)

These conditions must all be met for this rule to run.

Rule Name *	Exported Settings Inbox Rule
Active	Yes
Message Types *	Event Subscription
SubscriptionName	Equals App Settings Exported 2
EntityType	Equals SECRETSERVERSETTINGS 10024
ActionType	Equals EXPORTED
User	Equals 
EventDetails	Equals Application Settings, Launcher Settings, Protocol Handler Settings (Install-Time), Permission Options, User Interface, User Experience, Advanced Settings, Folder Settings, Local User Passwords, Email, Security, Ticket System, Session Recording, Login, SAML, Licenses
Details	Equals App Settings Exported - SECRETSERVERSETTINGS - EXPORTED

9. Edit the rule as desired. See [Creating a Rule from Scratch](#).

Using Inbox Templates

Overview

First, let us open a system inbox template to look at:

1. Go to **Admin > All**.
2. If necessary, click the view link to switch to **Alphabetized View**.
3. Click the **Notification Rules and Templates** link. The Notification Settings page appears:

Inbox > Notification Settings

Rules Templates Resources Send Test Notification

14 Items Include Inactive Create Rule


RULE NAME	ACTIVE	DIGEST	SYSTEM	USAGE (L...	NOTIFICATION TYPES
App Settings Exported	Yes	No	No	0	Event Subscription
Dependency Failure	Yes	No	Yes	0	Dependency Failure
Event Subscription	Yes	No	Yes	6	Event Subscription
Event Subscription: Eve...	Yes	No	No	6	Event Subscription
Inbox Test Message	Yes	No	Yes	0	Inbox Test Message
Password Reset	Yes	No	Yes	0	Password Reset
Secret Access Approved	Yes	No	Yes	0	Secret Access Approved
Secret Access Cancel ...	Yes	No	Yes	0	Secret Access Cancel Request
Secret Access Deny Re...	Yes	No	Yes	0	Secret Access Deny Request
Secret Access Request	Yes	No	Yes	0	Secret Access Request
Secret Changed	Yes	No	Yes	0	Secret Changed
Secret Creation	Yes	No	No	0	Event Subscription, Secret Changed
Secret Heartbeat Failed	Yes	No	Yes	0	Secret Heartbeat Failed
Workflow Access Appr...	Yes	No	Yes	0	Workflow Access Approval Request,

4. Click the **Templates** tab:

Inbox > Notification Settings 🔍 ☰ + WS

Rules Templates Resources Send Test Notification

8 Items All Types ▾ Create Template

TEMPLATE NAME ↑	SYSTEM	TYPE	RULES LEVERAGED	
Event Subscription	Yes	Email	App Settings Expor...	
Password Reset	Yes	Email	Password Reset	
Secret Access Appr...	Yes	Email	Secret Access App...	
Secret Access Req...	Yes	Email	Secret Access Req...	
Standard Email	Yes	Email	Dependency Failur...	
Standard Email Dig...	Yes	Email		
Standard Slack	Yes	Slack		
Workflow Access A...	Yes	Email	Workflow Access A...	

- Note that most, if not all, of the inbox templates are system templates. These are **read-only** templates that you can clone to create custom templates. That is, system templates are templates for your templates. Most of the templates are email templates, and one is a Slack template.
- Click the Template Name for the Event Subscription template. The template's page appears:

Inbox > Notification Settings > Templates > Event Subscription

System templates cannot be modified.

[Copy Template](#)

Template Details

Templates are used when an inbox rule sends a message.

Template Name *	Event Subscription
System	Yes
Template Type *	Email

Template Body

Define the message text and layout including languages and merging message data.

Language	English (English) ▼
Subject *	\$Details
Body *	<pre> <!DOCTYPE html> <html lang="en"> <head> <title>Message Type</title> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" /> </head> <body> <table border="1"> <tr> <td> <div style="font-family: Roboto, Helvetica, Arial; font-size: 16px;> ... </div> </td> </tr> </table> </pre>

7. Note that each template has:

- A details section that contains the name, a system template flag, and a template type (email or Slack).
- A body section that defines the subject, language used, and the canned text for the message. The message contains variables that are drawn from the alert or event. The body is read only in system templates and is editable in custom templates cloned (copied) from system templates.
- Zero or more associated inbox rules. These are the inbox rule types that use this inbox template (message type). Rules define filters for the alerts or events (what characteristics trigger the rule) and who gets externally notified via email or Slack (the subscribed users or groups). The following table lists the system templates and their associated inbox rules that use them.
- Zero or more resources. These are items, such as images, that go along with any email based on the template.

Table: System Inbox Rules by Inbox Template

Event Subscription	Email	App Settings Exported	Event Subscription

Password Reset	Email	Password Reset
Secret Access Approved	Email	Secret Access Approved
Secret Access Request	Email	Secret Access Request
Standard Email	Email	Dependency Failure Inbox Test Message Secret Access Cancel Request Secret Access Deny Request Secret Changed Secret Heartbeat Failed
Standard Email Digest	Email	
Standard Slack	Slack	
Workflow Access Approval Request	Email	Workflow Access Approval Request

8. Note the Body section is HTML for emails and Slack template text. For the Event Subscription template it looks like this:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Message Type</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
  <style>
    body, td {
      font-family: Roboto, Helvetica, Arial;
      font-size: 16px;
    }
    table {border-collapse: separate;}
    a, a:link, a:visited {text-decoration: none; color: #1071D4;}
    a:hover {text-decoration: underline;}
    h2, h2 a, h2 a:visited, h3, h3 a, h3 a:visited, h4, h5, h6, t_cht {color: #000 !important;}
    .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td {line-height: 100%;}
    .ExternalClass {width: 100%;}
    h1 { color: #121212; font-family: Roboto, Helvetica, Arial; font-style: normal; font-weight: bold; font-size: 32px; }
  </style>
</head>
<body style="background-color: #F7F7F7;">

  <table width="100%" border="0" cellspacing="0" cellpadding="0"><tr><td align="center">

<table cellspacing="0" cellpadding="0" border="0" width="100%" style="max-width: 640px; ">
  <tr>
<td style="background-color: #121212; width: 80%; height: 48px; color: #ffffff; padding-left: 32px;">
  $SystemLogo
  </td>
</tr>
<tr>
<td style="background-color: #ffffff">

  <table cellspacing="16" width="100%">
    <tr>
      <td width="24"> &nbsp; </td>
      <td align="center" style="padding-top: 42px; color: #121212; font-family: Roboto, Helvetica, Arial; font-style: normal; font-weight: bold; font-size: 32px; text-align: center;">
        $InboxMessageType - $SubscriptionName
      </td>
      <td width="24"> &nbsp; </td>
    </tr>
  </table>
  </td>
  </tr>
  </table>

```

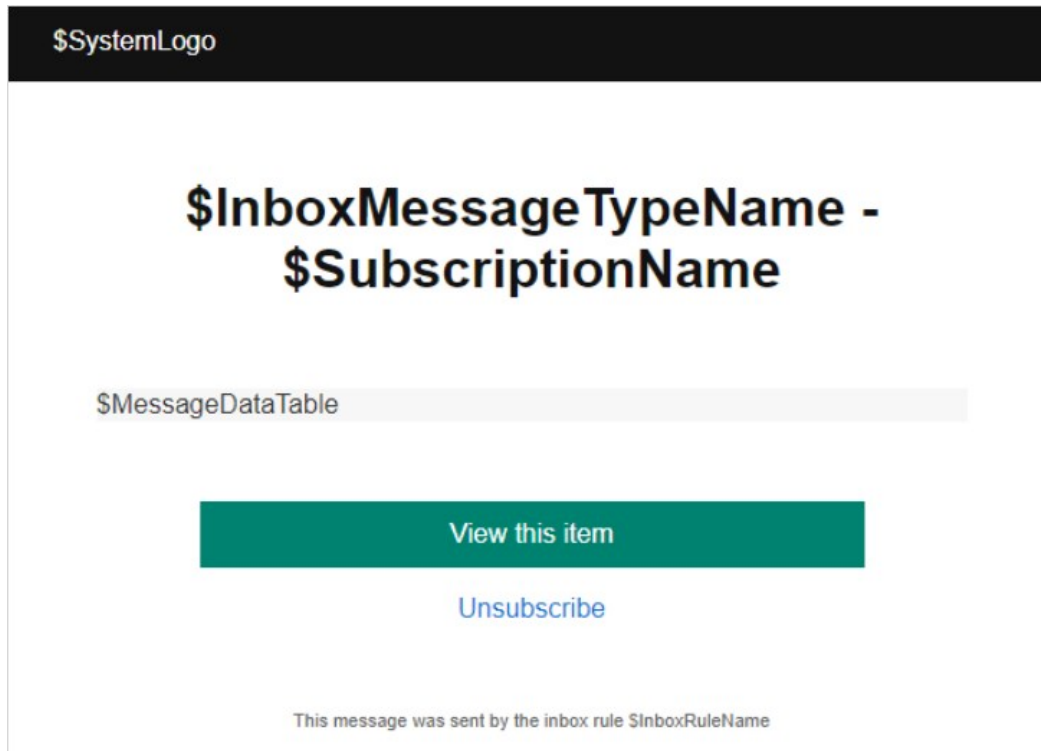
```

    < / t r >
    < t r >
    < t d > & n b s p ; < / t d >
<td style="font-family: Roboto, Helvetica, Arial; font-weight: normal; font-size: 16px; color: #323232;">
    & n b s p ;
    < / t d >
    < t d > & n b s p ; < / t d >
    < / t r >
    < t r >
    < t d > & n b s p ; < / t d >
<td style="background-color: #F7F7F7; font-family: Roboto, Helvetica, Arial; font-weight: normal; font-size: 16px; color: #323232; padding: 0px">
    $ M e s s a g e D a t a T a b l e
    < / t d >
    < t d > & n b s p ; < / t d >
    < / t r >
    < t r >
    < t d > & n b s p ; < / t d >
<td style="font-family: Roboto, Helvetica, Arial; font-weight: normal; font-size: 16px; color: #323232; padding: 16px" align="center">
    < p >
    <a href="$ApplicationUrl/app/#/inbox/view/notifications?messageId=$MessageId" style="text-decoration: none; display: inline-block;
background-color: #008270; width: 400px; height: 40px; line-height: 40px; color: #ffffff; text-align: center">
    V i e w   t h i s   i t e m
    < / a >
    < / p >
    < p >
    <a href="$ApplicationUrl/app/#/inbox/view/notifications?messageId=$MessageId&unsubscribe=true">
    U n s u b s c r i b e
    < / a >
    < / p >
    < / t d >
    < t d > & n b s p ; < / t d >
    < / t r >
    < t r >
    < t d > & n b s p ; < / t d >
<td style="font-family: Roboto, Helvetica, Arial; font-weight: normal; font-size: 11px; color: #646464" align="center">
    This message was sent by the inbox rule $InboxRuleName
    < / t d >
    < t d > & n b s p ; < / t d >
    < / t r >
< / t a b l e >

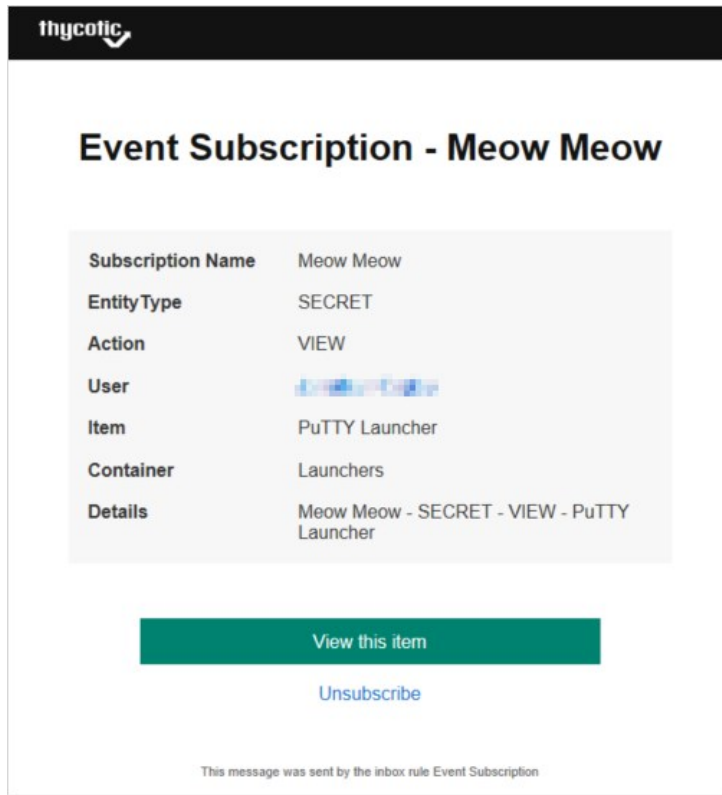
< / t d >
</table>
</td></tr></table>
</body>
</html>

```

9. Rendered, the body looks like this:



Note the variables starting with \$ that are in the message. These are replaced by Secret Server when it sends the message. For example:



10. The variables here include:

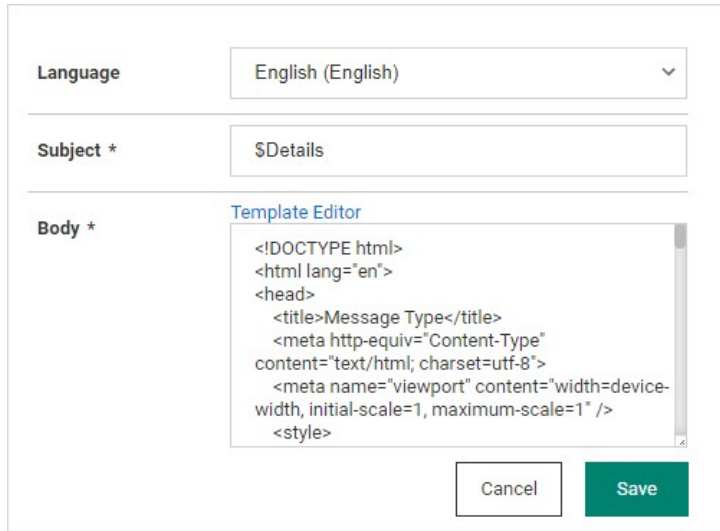
- \$InboxMessageType was replaced by the inbox template type.
- \$InboxRuleName was replaced by the inbox rule that sent the message. In this case, it is the same name as the inbox template type—Event Subscription.
- \$MessageDataTable was replaced by an entire table that summarized the message.
- \$SubscriptionName was replaced by the event subscription name.
- \$SystemLogo was replaced by the image resource containing the Delinea logo.

Note: For a complete list of variables for the template, go to the template editor (see below).

11. Nearly the entire template HTML is customizable once you make a customized clone of the system template. To clone the template click the **Copy Template** button at the top. The Copy Template popup appears:

The "Copy Template" popup dialog has a title bar "Copy Template". Below the title is a label "New Template Name *" followed by an empty text input field. At the bottom right of the dialog are two buttons: "Cancel" and "Copy Template".

12. Type the name of the new template in the **New Template Name** text box.
13. Click the **Copy Template** button. The template page reappears, but this time it is editable and named differently.
14. Click the **Edit** link next to **Template Body**. The section becomes editable:



The screenshot shows a form with three main sections:

- Language:** A dropdown menu currently set to "English (English)".
- Subject *:** A text input field containing "\$Details".
- Body *:** A text area containing HTML code. Above the text area is a link labeled "Template Editor".

At the bottom of the form are two buttons: "Cancel" and "Save".

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Message Type</title>
  <meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1, maximum-scale=1" />
<style>
```

15. You can directly edit the HTML, but if you intend to add variables for Secret Server to fill in, click the **Template Editor** link. The Inbox Template Editor popup appears:

Inbox Template Editor

Search or pick one ▾

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Message Type</title>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
7 <style>
8   body, td {
9     font-family: Roboto, Helvetica, Arial;
10    font-size: 16px;
11  }
12  table {border-collapse: separate;}
13  a, a:link, a:visited {text-decoration: none; color: #1071D4;}
14  a:hover {text-decoration: underline;}
15  h2,h2 a,h2 a:visited,h3,h3 a,h3 a:visited,h4,h5,h6,.t_cht {color:#000 !important;}
16  .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td {line-height: 100%;}
17  .ExternalClass {width: 100%;}
18  h1 { color: #121212; font-family: Roboto, Helvetica, Arial;font-style: normal;font-weight: bold;font-size: 24px;}
19 </style>
20 </head>
21 <body style="background-color: #F7F7F7;">
22
23 <table width="100%" border="0" cellspacing="0" cellpadding="0"><tr><td align="center">
24
25 <table cellspacing="0" cellpadding="0" border="0" width="100%" style="max-width: 640px; ">
26 <tr>
27 <td style="background-color: #121212;width: 80%; height: 48px; color: #ffffff; padding-left: 32px;">
28

```

Cancel Apply

16. In addition to directly editing the HTML, you can insert variables by clicking the **Search or pick one** dropdown list:

Search or pick one ▾

MESSAGE

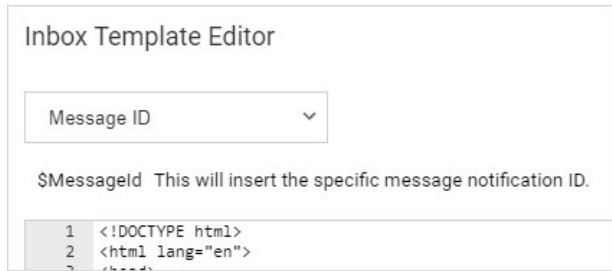
- Data Fields
- Message Data Table
- Message Type
- Message ID

GLOBAL

- Application URL

17. You have three categories of variables: message, global, and digest.

18. Click the desired variable. The dropdown changes to show your choice:



19. The variable appears immediately below the dropdown list, as well as a description of the variable.
20. Copy or type the variable in the desired location in the HTML.
21. Edit and insert more variables as desired.
22. Click the **Apply** button. The popup disappears.
23. Click the **Save** button.

SQL Server maintains a history of all operations in a transaction log. If this transaction log becomes full, you may receive one or more of the following errors:

System.ArgumentException: Cannot add two background tasks with the same name.

Thycotic.Data.DataAccessorException: The transaction log for database '{database}' is full. To find out why space in the log cannot be reused, see the log_reuse_wait_desc column in sys.databases

By default, a transaction log can grow to an unrestricted size, but some may become full in the following circumstances:

- The drive where the transaction log file is kept is out of disk space.
- The transaction log file hits its growth limit.

Potential Solutions

- Back up the log.
- Free up disk space so that the log can grow automatically.
- Move the log file to a disk drive with sufficient space.
- Increase the size of the log file.
- Add a log file on a different disk.
- Complete or kill a long-running transaction.
- Switch to simple recovery mode and truncate the log.

For more detailed information on transaction logs in SQL, see [Understanding and Managing Transaction Logs](#).

In addition to the user audit and individual secret audit, the reporting feature provides a series of activity, user, and secret reports. See [Built-in Reports](#) for the most up-to-date list of reports included.

Note: Users can also create their own, custom reports. See [Creating and Editing Reports](#).

The audit log for a secret can be accessed by clicking the **View Audit** button on the **Secret View** page or navigating from the User Audit report. The log shows the date, the username, the action, and any other details about the event. Secret auditing provides a detailed view of each change or view on a secret.

Note: Audit logs are visible to anyone with the "list" permission. Thus, anybody with that permission can view permission changes, users whose permissions were changed, secret dependency information, and the machine.

Secret audits are taken for the following user actions:

- Adding, updating and removing secret dependencies
- Check out
- Editing permissions
- Forced expiration
- Hide launcher password changes
- Set for check-in
- Update
- View

For certain audit items, action notes are added providing additional details. For example, if permissions are edited, an audit record is generated detailing which users or groups gained or lost permissions. Detailed audit records add accountability to sensitive secrets where auditors or administrators need to know exactly what was modified.

Below the audit records is a **Display Password Change Log** check box. Clicking to select this check box displays logs for Heartbeat and Remote Password Changing amongst the audit items

Secret Server Logs

SS log

The Secret Server system log is a top-level IIS log that reports when roles start and stop and any activity occurring on the site, as well as any legacy monitors.

Location: C:\inetpub\wwwroot\SecretServer\log

See [Enabling Debug Mode in System Log Files](#) for more on using this log.

SS-BSSR log

The *background scheduler server role log* is responsible for jobs that fire upon a trigger. Currently, we have some monitors that schedule work from the website but will transition to trigger jobs in the scheduler.

Location: C:\inetpub\wwwroot\SecretServer\log

SS-BWSR log

The *background worker server role log* is responsible for logging work triggered by the background scheduler and legacy monitors. Work includes heartbeat, password changing, discovery, event pipelines, and others.

Location: C:\inetpub\wwwroot\SecretServer\log

SS-EWSR log

The *engine worker server role log* is responsible for processing all responses from distributed engines, such as discovery and heartbeat.

Location: C:\inetpub\wwwroot\SecretServer\log

SS-MMSR log

The *MemoryMq server role log* records internal site connector activity when RabbitMq is not installed or used.

Location: C:\inetpub\wwwroot\SecretServer\log

SS-SRWSR log

The *session recording worker server role log* is responsible for processing session recordings from Secret Server.

Location: C:\inetpub\wwwroot\SecretServer\log

Protocol Handler Log

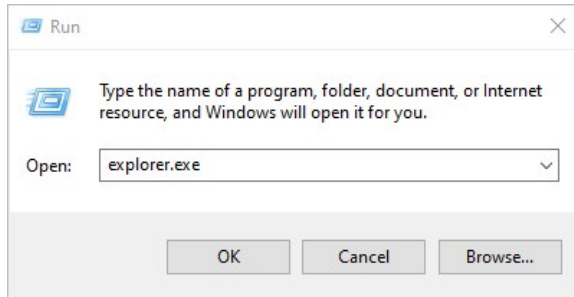
SS-RDPWin log

The Remote Desktop Protocol (RDP) for Windows log records protocol handler activity. It is located at C:\Users\Administrator\AppData\Roaming\Thycotic\log

Note: Despite the name, the log covers more than just RDP.

To access the log:

1. Type Run in the Windows start menu search text box to launch the Run Command application. The Run popup appears:



2. Type %AppData%\Thycotic\log into the **Open** text box.
3. Click the **OK** button. The folder containing the file appears.

Enabling Debug Logging for the Protocol Handler Log

This section explains how to enable verbose debug logging for protocol handler.

1. Navigate to C:\Program Files\Thycotic Software Ltd\Secret Server Protocol Handler.
2. Open the log4net-rdp.config configuration file in a text editor as an administrator:

```
<?xml version="1.0" encoding="utf-8"?>
<log4net>
  <root>
    <!-- Troubleshooting: detailed log information (large number of log files can accumulate) -->
    <!--<level value="DEBUG" />-->

    <!-- Troubleshooting: very detailed log information (large number of log files can accumulate) -->
    <!--<level value="VERBOSE" />-->

    <!-- Logging: off -->
    <!--<level value="OFF" />-->

    <level value="INFO" />
    <appender-ref ref="Thycotic.LogFileAppender" />
  </root>

  <appender name="Thycotic.LogFileAppender" type="log4net.Appender.RollingFileAppender">
    <!--<file value="C:\LogFiles\Thycotic\SS-RDPWin.log" />-->
    <file value="%{AppData}\Thycotic\log\SS-RDPWin.log" />
    <rollingStyle value="Size" />
    <maxSizeRollBackups value="34" />
    <maximumFileSize value="10MB" />
    <lockingModel type="log4net.Appender.FileAppender+MinimalLock" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%utcdate [CID:%property{Correlation}] [C:%property{Context}] [TID:%thread] %-5level %logger - %message%newline" />
    </layout>
  </appender>
</log4net>
```

3. Comment out <!--<level value="INFO" />.
4. Remove the comment out of <level value="DEBUG" />.
5. Re-create the original issue a couple of times with DEBUG enabled.
6. Navigate to ~\AppData\Roaming\Thycotic\log" on the machine you are launching protocol handler from. Copy and save the ss-RDPWin log file.
7. Return log4net-rdp.config to its original state by removing the comment out of <!--<level value="INFO" /> and commenting out <level value="DEBUG" />

Distributed Engine Log

SSDE log

The *Secret Server distributed engine log* is responsible for recording distributed engine activity.

Location: C:\Program Files\Thycotic Software Ltd\Distributed Engine\log

To configure the DE log:

1. Log in as an administrator on the distributed engine server.
2. Locate the `Thycotic.DistributedEngine.Service.exe.config` in the C:\Program Files\Thycotic Software Ltd\Distributed Engine directory.
3. Open the file in a text editor to change it as desired.

Enabling Debug Mode in Distributed Engine Log Files

Please see [Enabling Debug Mode in Distributed Engine Log Files](#).

Enabling Discovery Logging

To enable discovery logging, change the `Thycotic.DistributedEngine.Service.exe.config` configuration file as follows:

```
<logger name="Thycotic.Discovery">  
  <level value="VERBOSE" />  
</logger>
```

RabbitMQ Log

Access the RabbitMQ logs at C:\RabbitMq\log.

Overview

Secret Server can send a copy of important log messages to an external syslog server for added security using the following protocols:

Note: Common Event Format (CEF) is an industry-standard format on top of syslog messages that ensures event interoperability between different platforms.

Table: Syslog Transportation Protocols

UDP	No	Least reliable. User Datagram Protocol (UDP) traffic is fire-and-forget with no assurance messages are delivered and no error checking.
TCP	No	More reliable. Transmission Control Protocol (TCP) ensures messages arrive in order, missing messages are resent, and has built in error checking.
Secure TCP	Yes	Establishes a secure connection — Transport Layer Security (TLS) 1.1 or 1.2 only. Syslog Server's certificate is validated by Windows to ensure it is trusted and not revoked. Can be used with or without client certificates (configured in Configuration > Security tab > TLS Auditing > Advanced).

[Unexpected Link Text](#)

Due to the sensitive nature of Secret Server logs, we strongly recommend using Secure TCP.

Configuring a Secure TCP Syslog/CEF External Audit Server in Secret Server

Compatible Audit Servers

- syslog-ng
- Any Audit server that accepts TLS encrypted messages using the BSD syslog protocol

Configuring an External Audit Server

1. Navigate to **Admin > Configuration**.
2. Click the **General** tab.
3. Click the **Edit** button at the bottom of the page.
4. Go to the **Application Settings** section.
5. Click to select the **Enable Syslog/CEF Logging** check box. A syslog/CEF section appears:

Syslog/CEF Logging Advanced Settings Information

Enable Syslog/CEF Logging

Syslog/CEF Server

Syslog/CEF Port

Syslog/CEF Protocol ▼

Syslog/CEF Time Zone ▼

Syslog/CEF Site ▼

Write Syslogs As Windows Events

Note: syslog/CEF may require an additional license key. To install licenses, navigate to **Admin > Licenses > Install New License**. Once installed, the license requires activation. Contact your Delinea Sales Representative with any questions.

6. Type IP address or name for the IIS server hosting the syslog/CEF server in the **Syslog/CEF Server** text box.
7. Type the port number where the logging information will be passed (6514 is the default port for secure TCP syslog) in the **Syslog/CEF Port** text box.

Note: Secret Server requires outbound access to this server and port so communication can pass freely.

8. Click the **Syslog/CEF Protocol** dropdown list and select **Secure TCP**. Secure TCP means either TLS v1.2 or v1.1 because other versions of SSL, such as SSL v3 and TLS v1.0, have known weaknesses.
9. Click to select **Syslog/CEF Time Zone** list box to **UTC Time** or **Server Time**, depending on your preference.
10. Click the **Save** button.

Caching Syslog Audits

If the connection between the external syslog server and Secret Server breaks once secure syslog logging is enabled in Secret Server, syslog failure notification messages is cached in the Secret Server database and re-sent at regular intervals until the connection between the syslog server and Secret Server is reestablished.

Configure Auditing for TLS Connections

To track problems with TLS connections (including whenever the connection fails), enable the TLS certificate chain policy and error auditing in Secret Server:

1. Navigate to **Admin > Configuration**.
2. Click the **Security** tab.
3. Click the **Edit** button at the bottom of the page.

4. Scroll to the **TLS Auditing** section.
5. Ensure the **Apply TLS Certificate Chain Policy and Error Auditing** check box is enabled. If not, you cannot use client certificates.

Note: If secure TCP is used for the syslog/CEF protocol and there are one or more client certificate thumbprints entered, Secret Server checks the local computer's Web hosting and personal certificate store and uses the first one it finds.

Adding Client Certificate Thumbprints

1. Navigate to **Admin > Configuration**.
2. Click the **Security** tab.
3. Click the **Edit** button at the bottom of the page.
4. Scroll to the **TLS Auditing** section.
5. Click the **Advances (not required)** link. A client certificate thumbprint section appears:

Advanced (not required)

Secret Server's IIS AppPool must be granted permission to use the Client Certificate, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). Example usage:
winhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "Certificate Subject" -a "HOSTNAME\IIS APPPOOL\SecretServer"
[Download WinHttpCertCfg - Official WinHttpCertCfg documentation](#)

Client Certificate Thumbprint(s) ⓘ

6. Copy and paste a list of SHA1 SSL certificate thumbprints into the **Client Certificate Thumbprints(s)** text box. Separate each thumbprint (40 characters each) with a semicolon. Up to ten are allowed.

Note: Secret Server's IIS application pool must be granted permission to use the client certificates, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). See [Compatibility Notes for Client Certificates](#).

Determining the Status of a Remote Audit Server

To view the logs for any TLS-Connection related errors, perform the following:

1. Open the **Microsoft SQL Server Management Studio**.
2. Navigate to your SecretServer database at **<DB Machine Name > > Databases > SecretServer**).
3. Set up a new query.
4. Type and enter `select from tbSecurityAuditLog` to view the events from the TLS audit.

Note: For more detailed troubleshooting reporting, reference the logs on the Secret Server Web server at `C:\inetpub\wwwroot\SecretServer\log`. View the `SS.log`, `SS-BSSR.log` (background scheduler), and `SS-BSWR.log` (background worker) for any errors.

Compatibility Notes for Client Certificates

IIS Application Pool Certificate Permissions

Secret Server's IIS application pool must be granted permission to use the client certificates, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe).

For example: `winhttpcertcfg.exe -g -c LOCAL_MACHINEMY -s "Certificate Subject" -a "HOSTNAME\IIS APPPOOL\SecretServer"`

You can download the tool at:

[Windows HTTP Services Certificate Configuration Tool \(WinHttpCertCfg.exe\)](#)

You can view the documentation at:

[WinHttpCertCfg.exe, a Certificate Configuration Tool](#)

Otherwise, if Secret Server is configured to use a client certificate, and IIS does not have permission, errors like this may appear in the logs:

TLS Error Detected (Authentication Error connecting to IP:PORT) - The credentials supplied to the package were not recognized.

If you are using a client certificate, and a syslog-ng logging server, the following message may occasionally appear in the main syslog-NG log file:

SSL error while reading stream; tls_error='SSL routines:ssl_get_prev_session:session id context uninitialized'

On the Secret Server side, this appears:

TLS Error Detected (Authentication Error connecting to IP:PORT) - Authentication failed because the remote party has closed the transport stream.

This is caused by Windows trying to cache secure connections when client certificates are used, but because syslog-ng has not configured the OpenSSL "session id context", OpenSSL displays this error when it tries to resume a previous session.

Secret Server automatically reconnects and resends any missed messages, so the errors should not have an impact. However, you can disable Windows' secure connection caching by adding the [ClientCacheTime](#) setting set to 0 in the Registry and then rebooting. This did not cause any significant performance impact in internal testing.

Note: If changing back to a previous syslog IP address and port, you will receive a closed connection TLS error on the first attempted syslog connection after making the change. A subsequent call will succeed as the first failure will clear the cached connection on Windows. This is due to the issue with syslog-ng.

Note: If syslog-ng configures their OpenSSL session id context, this error message correction is no longer needed.

AlienVault

It is common for people to incorrectly use the client certificate thumbprints feature when setting up secure AlienVault for syslog. This can cause Secret Server to try to connect to LDAPS with the AlienVault certificate, which can break LDAPS. Users should not use the Secret Server client certificates thumbprint for specifying one certificate for syslog and another for LDAP. The certificate list is intended for each Secret Server or DE to have its own, unique certificate.

Overview

Secret Server Web nodes and distributed engines log levels are remotely configurable and collectable. This feature is especially useful for large systems with many nodes and engines.

Configuration for the web nodes is found on the Server Nodes configuration page, alongside role settings. Configuration for distributed engines is found in the Distributed Engine configuration page. Log levels include: All, Debug, Info, Warn, Error, Off, and Not Set (the default). Previously, manual configuration file changes were required. "Not Set" relies on the configuration files for the logging level, which was the previous default behavior.

The diagnostic feature for collecting logs is improved and now gathers logs from all nodes and engines.

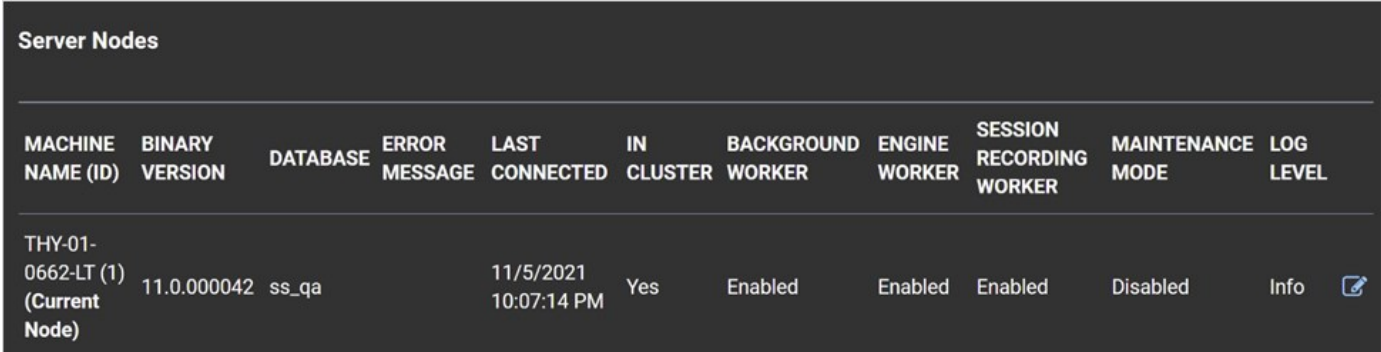
This feature is at Admin > Server Nodes and Admin >


Setting the Logging Level

Secret Server Web Servers

For Web servers, you can set the log Level for web server nodes from the Server Nodes page:

1. Go to **Admin > See All**. A list of admin settings appears.
2. Click the **Server Nodes** link. The Server Nodes page appears:



MACHINE NAME (ID)	BINARY VERSION	DATABASE	ERROR MESSAGE	LAST CONNECTED	IN CLUSTER	BACKGROUND WORKER	ENGINE WORKER	SESSION RECORDING WORKER	MAINTENANCE MODE	LOG LEVEL
THY-01-0662-LT (1) (Current Node)	11.0.000042	ss_qa		11/5/2021 10:07:14 PM	Yes	Enabled	Enabled	Enabled	Disabled	Info 

3. Click the edit icon next to the server you want to change. The settings become editable.
4. Click the **Log Level** dropdown list to select the desired level.
5. Click the save icon on the far right to save the change.

Distributed Engines

Similarly, for distribution engines:

1. Go to **Admin > Distributed Engine**. The Distributed Engine page appears:

Admin > Distributed Engine

Sites and Engines | Site Connectors | Configuration | Logs | Audit

Add Site | Add Engine

Show Inactive

Pending Engines Engines: 3 3 Engines Pending

3 Items All Connection Statuses All Activation Statuses

FRIENDLY NAME ↑	CONNECTION STATUS	ACTIVATION STATUS	LAST CONNECTED
THY-01-0128-LT	Offline	Inactive	28 days, 23 hours ago
THY-01-0128-LT	Offline	Inactive	28 days, 23 hours ago
THY-01-0128-LT	Offline	Active	21 days, 20 hours ago

2. Hover the mouse pointer over the desired engine. Three dots appear on the far right.
3. Click the dots icon and select **Show Settings**. The Engine Settings popup page appears:

Engine Settings

Settings to allow separate engine functionality within a Site to a set of Engines; There is no requirement to make changes and it is recommended to not change settings unless necessary

Directory Services	<input checked="" type="checkbox"/>
Advanced Auditing	<input checked="" type="checkbox"/>
Local Account Discovery	<input checked="" type="checkbox"/>
Password Changing	<input checked="" type="checkbox"/>
Secret Workflow	<input checked="" type="checkbox"/>
Service Account Management	<input checked="" type="checkbox"/>
SSH Proxy	<input checked="" type="checkbox"/>
SSH Terminal	<input checked="" type="checkbox"/>
Log Level	< Not Set > <input type="button" value="v"/>

4. Click the **Log Level** dropdown list to select the desired level.

5. Click the **OK** button to save the change.

To view a user audit report:

1. From the **Reports** page, click the **User Audit** tab.
2. From the dialog on the tab, select a user and a date range to view.
3. Click **Search History** to view the user's audit trail.

The audit search displays results for all the secrets the selected user has viewed or edited during the selected time period. The administrator has the option of expiring all the viewed secrets, to notify users to change sensitive information, or to force password changing (if the RPC is configured).

To get a full view of the actions taken on a secret, select that secret from the results list. The secret audit displays the specific user actions for a secret.

The System Log is used to communicate the different events that are occurring while Secret Server is executing. It can be helpful in troubleshooting unexpected behavior. The system log can be enabled by clicking **Edit** and checking the **Enable System Log** check box on the **Administration > System Log** page.

System log parameters include:

- **Maximum Log Length:** This is the maximum number of rows to keep in the system log table in the SQL database. When it reaches that amount, it is reduced by 25%.
- **Notify Administrators when System Log is Shrunk:** This setting is used to send an email to all system log administrators when the system log has been truncated. A system log administrator is any user in a role with the Administer System Log permission included.

To clear the system log of all its records, click **Clear**.

To view the events that have been triggered in a subscription, navigate to **Administration > Event Subscriptions** and click **View Audit**. In the Event Subscription Activity list, the most recent events to have been triggered are on top of the list. To select a specific time frame, click the ... buttons and select start and end dates at the top of the page. Click **Update Report** to return the corresponding log entries.

Note: It may take a few seconds for the events to make it into the log.

Folders

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret folders allow you to create containers of secrets based on your needs. They help organize your customers, computers, regions, and branch offices, to name a few. Folders can be nested within other folders to create sub-categories for each set of classifications. Secrets can be assigned to these folders and sub-folders. Folders allow you to customize permissions at the folder level, and all secrets within can inherit the folder's permissions. Setting permissions at the folder level ensures future secrets placed in that folder have the same permissions, simplifying management across users and groups.

Note: You can "favorite" a folder in the main menu by right clicking it.

If the new folder is a subfolder, you can have it use the sharing settings of its parent folder by enabling the "inherit permissions from parent" setting for the folder.

Folders can apply one of the following permissions to users or groups in the folder's Permissions table:

- **View:** Allows the user to see the folder and secrets in that folder that are inheriting permissions from their folder.
- **Edit:** Allows the user to create new folders in that folder, which forces the "Inherit Permissions from Parent" permission on the new folder, move secrets into that folder, and add new secrets into that folder.
- **Add Secret:** Allows the user to add a secret in that folder. Does not grant access to the added secret.
- **Owner:** Allows the user to create new folders in that folder without forcing inheritance, move the folder, delete the folder, rename the folder, and change the permissions and inheritance settings on the folder.

Depending on your configuration, these settings could affect the permissions of subfolders and secrets contained in this folder. Folders are not visible to users that do not have at least View permission. This allows users to create and manage their own folders without making them visible to all users.

Personal Folders

In Secret Server, a *personal folder* is a folder that one (and only one) individual has owner access to. No user can modify sharing permissions on these folders. A user cannot add subfolders to their personal folder. The purpose of this folder is to allow a user to securely store work-related secrets that other users do not require access to. Note that when in break-the-glass mode, an unlimited admin can access a user's personal folder in order to recover secrets if needed.

Required Role Permissions for Managing Folders

Folder management is subject to these role permissions:

- The Administer Folders role permission allows a user to create new folders and manage folders, but specific folder permissions still apply.
- Any user with the Administer Folders role permission can create new folders; however, to create folders at the root level, the user also needs the Create Root Folders permission. They also can add new folders to any folders where they have Edit or Owner permission on that folder.
- They must have Owner permission to delete a folder.
- Users can also move folders where they have Owner permission on the source folder and Edit or Owner permission on the target folder (where they are moving it). The folder automatically inherits Permissions from its parent when it is moved, which is the same as when secrets are moved.

To setup this feature, navigate to **Administration > Folder Synchronization**. To edit the settings, you must have a role assignment with Administer ConnectWise Integration permissions.

Enabling folder synchronization requires specifying the synchronization interval in days, hours, and minutes. The "Folder to Synchronize" is the parent folder where you create the folder structure. There are two methods of Folder Synchronization, through the ConnectWise API or through a database view.

Synchronizing with the ConnectWise API

The ConnectWise API is the recommended way to sync folders from ConnectWise. To sync:

1. Select ConnectWise API from the Folder Synchronization Method list.
2. Enter your ConnectWise site name.
3. Select a ConnectWise Integrator Secret for API Access.

Folder Synchronization Configuration Edit

[Explain](#)

Folder Synchronization Method	ConnectWise API
Synchronization Interval for Folder	Days: 0 Hours: 0 Minutes: 30
Folder to Synchronize	\Clients Clear
Site URL	* staging.connectwisedev.com
Company ID	* acmeinc
Integrator Credentials	ConnectWise (secretserver001) Create New Secret
Folder Structure	\$TYPE\STATUS

[Save](#) [Cancel](#) [Test Connection](#)

Note: The Integrator account must have access to the Company API in ConnectWise and access to all records

ConnectWise

Recent Chat with Support

Setup Tables > Integrator Login List > Integrator Login

Integrator Login

Setup Logs

Updated: 7/12/2016 2:06:55 PM by Admin1

Username: secretserver001

Password: ●●●●●●●●

Access Level: All records

Enable Available API(s)

Service Ticket API

Service Board: [dropdown]

Callback URL: [text box]

Use legacy callback format

Time Entry API

Member: [text box]

Callback URL: [text box]

Use legacy callback format

Managed Services API

Enable automatic childing of Configurations (more info)

Allow for Configurations to be childed by the Integrator

Contact API

Callback URL: [text box]

Use legacy callback format

Company API

Callback URL: [text box]

Use legacy callback format

Folder structure defines how folders are named under the client's folder. By default, \$TYPE\$STATUS creates sub-folders based on the customer type in ConnectWise, then further sorted by the active status in ConnectWise. For example, the active prospect "Acme Inc" in ConnectWise would get the following folder created: Clients\Prospects\Active\Acme Inc

The supported folder structure tokens are:

- **\$COMPANYINITIAL**: First letter of company name. Use to organize companies into subfolders of A, B, C, and the like.
- **\$STATUS**: Company status, such as active, inactive, or not-approved.
- **\$TYPE**: Company type, such as competition, customer, partner, prospect, suspect, or vendor.

When configured, save and scroll down to the bottom and click **Synchronize Now** to run the synchronization

Note: See the [How to create a custom view for ConnectWise synchronization](#) article for more advanced technical information on

setting up the SQL View.

Synchronizing with a Database (Advanced)

The database synchronization method queries an on-premises database for a custom view and parse company information out of it.

Enter the SQL Server location, SQL database name, and the credential information for accessing the reference database, for example, to your ConnectWise instance. The SQL view defaults to a standard ConnectWise customer layout but can be customized to meet the desired folder Layout.

Folder Synchronization Configuration Edit

[Explain](#)


Folder Synchronization Method Database (Advanced) ▾

Synchronization Interval for Folder

Days

Hours

Minutes

Folder to Synchronize  [No Selected Folder](#)

SQL Server Location !

SQL Database Name !

SQL Username !

SQL Password !

SQL View

ConnectWise

Custom View

[Advanced \(not required\)](#)

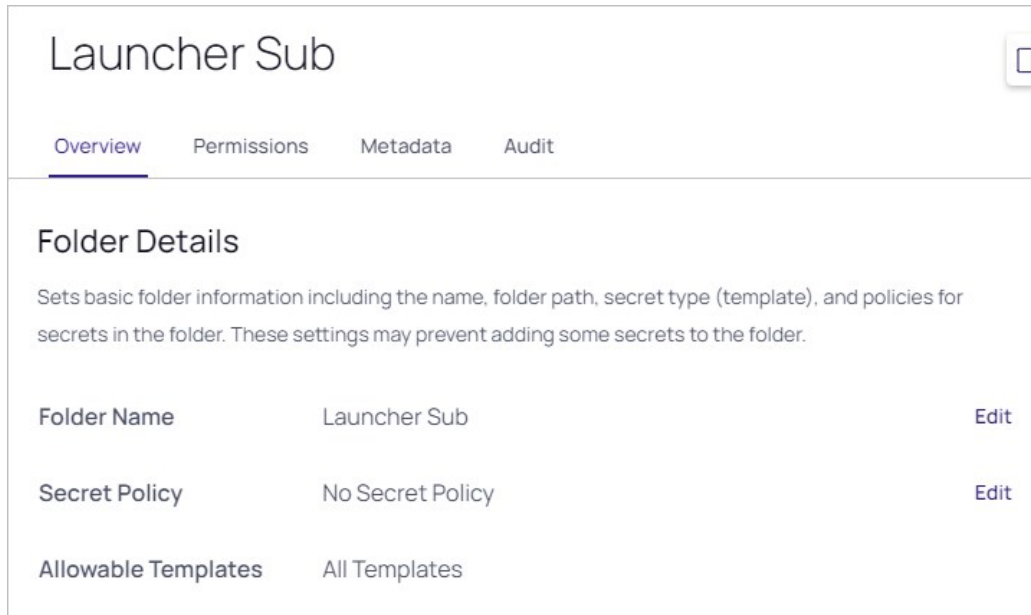
Days to Keep Operational Logs

"Days to Keep Operational Logs" sets the period to keep folder-synchronization-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Assigning Secret Policies to Folders

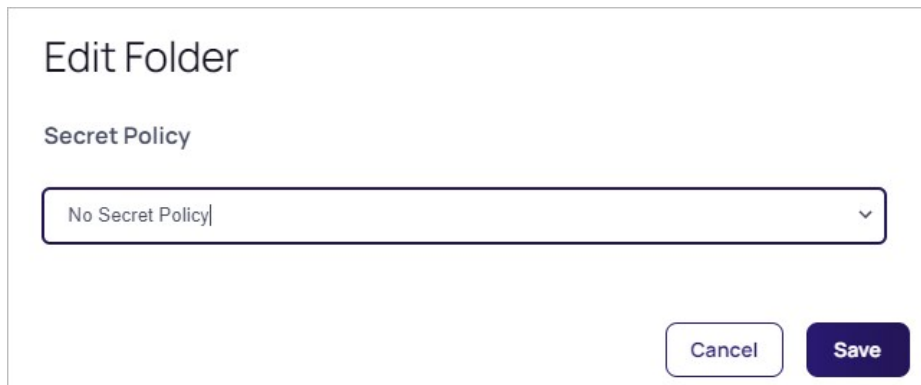
1. Navigate to the folder containing the secret you want to assign a policy to.
2. Right-click the folder and select Edit Folder. The Folder Details page appears:



The screenshot shows the 'Folder Details' page for a folder named 'Launcher Sub'. The page has a header with the folder name and a navigation bar with tabs for 'Overview', 'Permissions', 'Metadata', and 'Audit'. Below the navigation bar, there is a section titled 'Folder Details' with a description: 'Sets basic folder information including the name, folder path, secret type (template), and policies for secrets in the folder. These settings may prevent adding some secrets to the folder.' Below this description is a table with three rows, each representing a folder property and its current value, with an 'Edit' link to the right of each value.

Property	Value	Action
Folder Name	Launcher Sub	Edit
Secret Policy	No Secret Policy	Edit
Allowable Templates	All Templates	

3. Click the **Secret Policy Edit** link. The Edit Folder popup appears:



The screenshot shows the 'Edit Folder' popup. It has a title 'Edit Folder' and a section titled 'Secret Policy'. Below the title is a dropdown menu with the text 'No Secret Policy' and a downward arrow. At the bottom of the popup are two buttons: 'Cancel' and 'Save'.

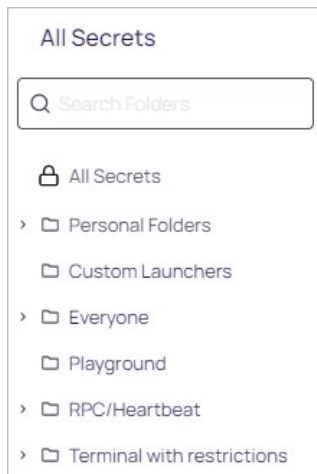
4. Click to select the desired policy in the **Secret Policy** dropdown list.
5. Click the **Save** button.

Creating Folders

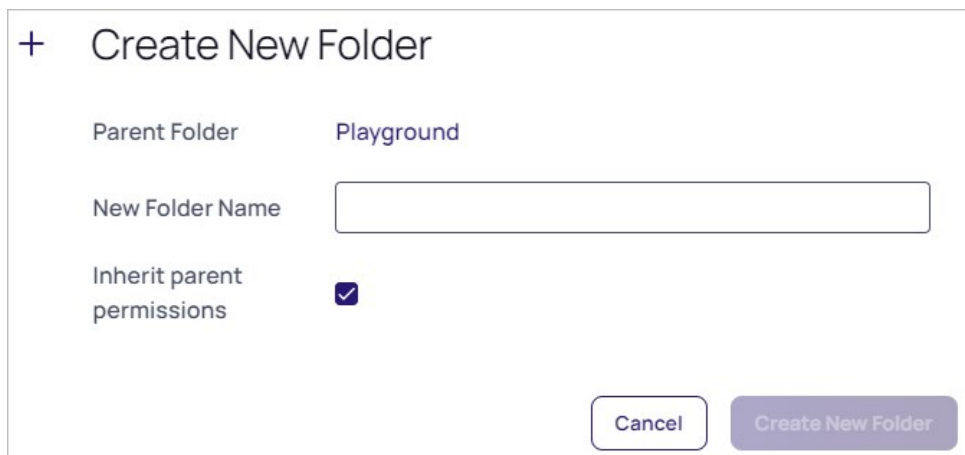
To create a folder:

Note: To create folders, you must have a role with the "administer folder" permission. You also must have edit or owner permission for the parent folder.

1. Click >> on the **Secrets** menu item to view the Secret Folder-Tree Panel.



2. Drill down and click the folder you wish to add a subfolder to. We chose the Playground folder.
3. Right click the folder and select **Add Subfolder**. The Create New Folder popup appears:

A screenshot of the 'Create New Folder' popup dialog. The title is '+ Create New Folder'. It has two rows of labels and values: 'Parent Folder' with the value 'Playground', and 'New Folder Name' with an empty text input box. Below these is a label 'Inherit parent permissions' with a checked checkbox. At the bottom right, there are two buttons: 'Cancel' and 'Create New Folder'.

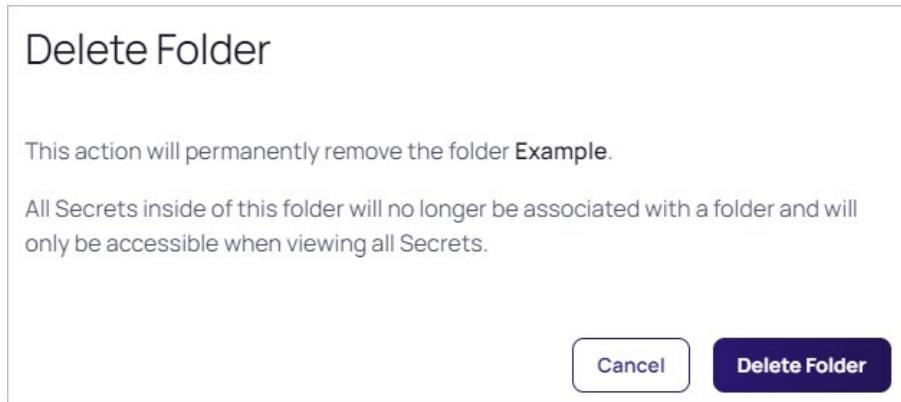
4. Type the new folder's name in the **New Folder Name** text box.
5. If desired, click to toggle the **Inherit Parent Permissions** check box.
6. Proceed to [Editing Folder Permissions](#) to customize permissions for the new folder.

Deleting Folders

To delete a folder:

Note: To delete folders, you must have a role with the "administer folder" permission. You also must have edit or owner permission for the parent folder.

1. Navigate to the folder in the folder tree on the main menu.
2. Right click the folder and select **Delete Folder**. The Delete Folder pop-up page appears:



3. Click the **Delete Folder** button.

Editing Folder Permissions

To edit folder permissions:

Note: To edit folder permissions, you must have a role with the "administer folder" permission. You also must have edit or owner permission for the parent folder.

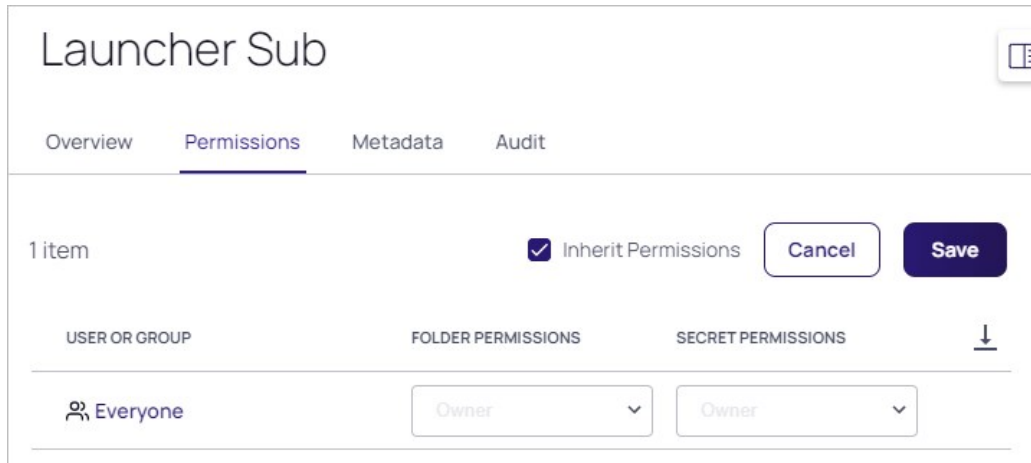
1. Navigate to the folder to edit.
2. Right-click the folder and select Edit Folder. The Folder Details page appears:

Folder Details		
Folder Name	Launcher Sub	Edit
Secret Policy	No Secret Policy	Edit
Allowable Templates	All Templates	

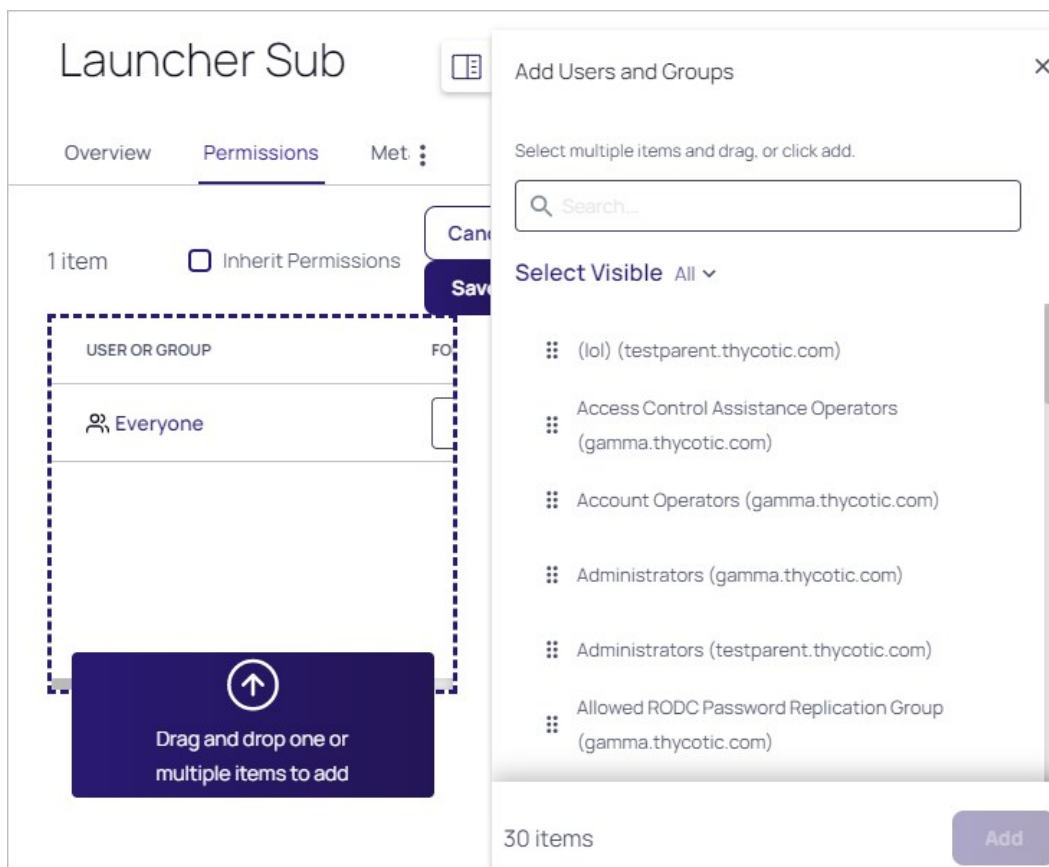
3. Click the Permissions tab:

USER OR GROUP	FOLDER PERMISSIONS	SECRET PERMISSIONS	
Everyone	Owner	Owner	↓

4. Click the **Edit** button. The Folder Permissions section becomes editable. It is currently set to the default, which is Inherit Permissions, so the Inherit Permissions check box is selected and the selected groups are not editable:



5. Click to deselect the **Inherit Permissions** checkbox. The permissions section becomes editable and the Add Users and Groups popup appears:



6. Drag and drop any desired users or groups from the popup into the table. Ctrl+click to select multiple items before dragging and dropping. You can also select items and click the **Add** button instead.

7. Click the **X** button on the popup to close it. The added users and groups appear:

Launcher Sub

Overview
Permissions
Metadata
Audit
⋮

4 items
 Inherit Permissions
Cancel
Save

USER OR GROUP	FOLDER PERMISSIONS	SECRET PERMISSIONS	↓
Everyone	Owner ▼	Owner ▼	
Administrators	View ▼	View ▼	
Account Operators	View ▼	View ▼	
Auditors	View ▼	View ▼	

Note: This image is for demonstration. Normally, adding users or groups to a folder that already has the Everyone group is redundant.

8. To delete any user or group, hover the mouse pointer over its row and click the trashcan icon.
9. For each user or group, click the **Folder Permission** dropdown list for the desired user of group to select the desired maximum permission available to them for the folder: **View** (folder), **Add Secret** (to folder), **Edit** (folder), or **Owner** (of folder).
10. Similarly, click the **Secret Permissions** dropdown list for the desired user of group to select the desired maximum permission available to them for secrets in the folder: List (secrets in folder), View (secrets in folder), Edit (secrets in folder), or Owner (of secrets in folder).
11. Click the **Save** button.

Enabling Personal Folders

To use personal folders, you must first enable them:

1. Click **Admin > Configuration**.
2. Click the **Folders** tab:

Configuration

General Login SAML **Folders** Local User Passwords Security Ticket System Email Session Recording HSM

Require View Permission on Specific Folder for Visibility	Yes
Enable Personal Folders	Yes
Personal Folder Name	Personal Folders
Should the warning message be displayed to the user when creating Secrets?	Yes
Warning message text	This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.
Personal Folder Display Name	Display Name

[← Back](#) [✎ Edit](#)

3. Click the **Edit** button:

Configuration

General
Login
SAML
Folders
Local User Passwords
Security
Ticket System
Email
Session Recording
HSM

Require View Permission on Specific Folder for Visibility

Enable Personal Folders

Personal Folder Name

Should the warning message be displayed to the user when creating Secrets?

Warning message text

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Personal Folder Display Name

Save

Cancel

4. Click to select the **Enable Personal Folders** check box.
5. (Optional) Type a new folder name in the **Personal Folder name** text box to customize the root-level folder that contains all personal folders.
6. (Optional) If you want to display a warning message to users when placing secrets in their personal folders:
 1. Click to select the **Should the warning message be displayed to the user when creating Secrets?** check box.
 2. (Optional) Edit the **Warning message text** box.
7. (Optional) Click the **Personal Folder Display Name** dropdown list to choose whether to display the personal folder display name or the username and domain.
8. Click the **Save** button. A personal folder for each user is now created in a root-level folder with the personal folder name specified.

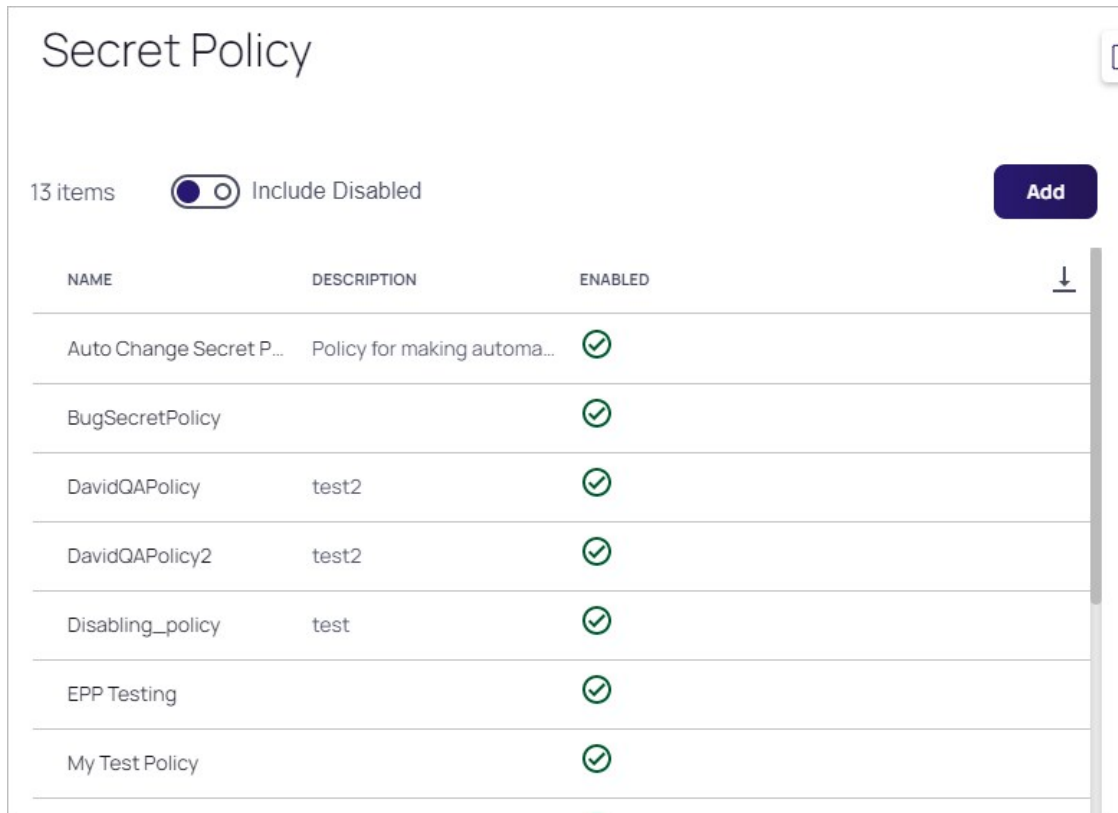
Note: When personal folders are enabled, a user requires the Personal Folders role permission in their role to be able to view and use their own personal folder.

Modifying Folders with Secret Policies

You can configure secret policies to apply RPC and security settings to an entire folder of secrets.

To create a new secret policy:

1. Click **Admin > Secret Policy**. A Secret Policy page appears:



13 items Include Disabled **Add**

NAME	DESCRIPTION	ENABLED	
Auto Change Secret P...	Policy for making automa...	✓	↓
BugSecretPolicy		✓	
DavidQAPolicy	test2	✓	
DavidQAPolicy2	test2	✓	
Disabling_policy	test	✓	
EPP Testing		✓	
My Test Policy		✓	

2. Click the **Add** button. The (new) Secret Policy page appears:

+ Secret Policy

Name *

Description

Enabled

3. Type a name for the new secret policy in the **Name** text box.
4. (Optional) Type a description in the **Description** text box.
5. Ensure the **Enabled** check box is selected.

Note: To deactivate a policy that you no longer want, edit the policy and deselect the **Enabled** check box. For information about applying a secret policy to a folder, see [Editing Folder Permissions](#).

Moving Folders

To move a folder:

1. Ensure that you have edit permission for both the source and destination folders.
2. Expand the folder tree to make both folders visible.
3. Drag the source folder and drop it on the destination folder. The Move Folder page appears:

Move Folder

Please press the 'Confirm Move Folder' button to move the folder **Example 2** to **Example**.

Select a different folder

Inherit Policy

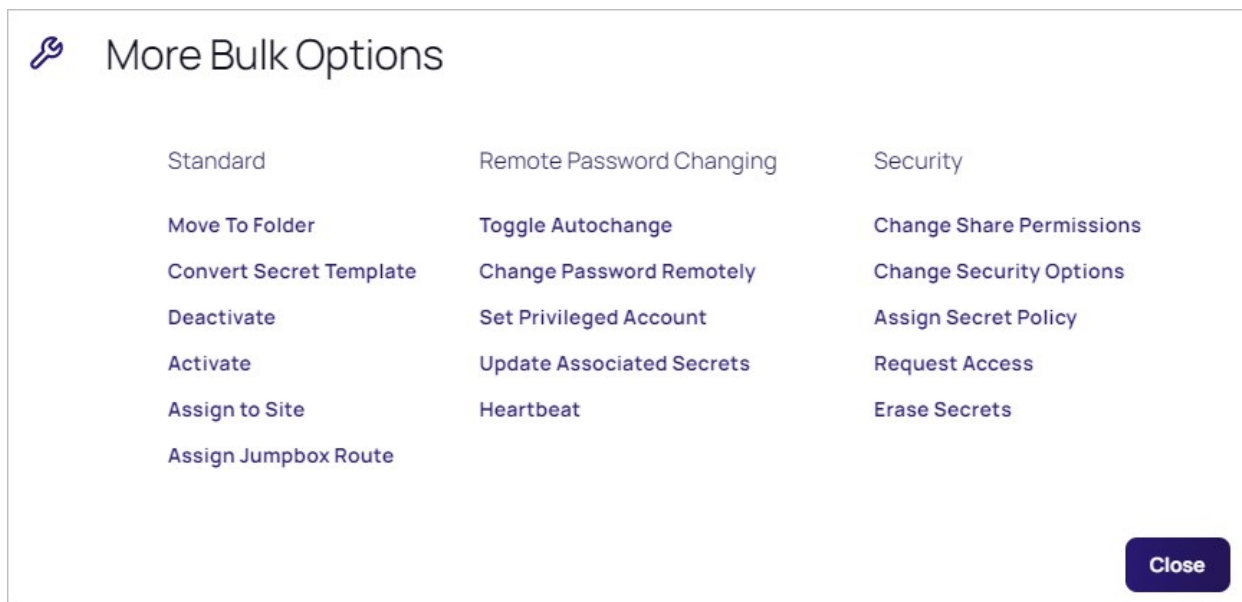
Inherit Permissions

4. Click the **Inherit Policy** dropdown list to select how you want policy inheritance to work for the folder in the new location.
5. Click the **Inherit Permissions** dropdown list to select how you want permission inheritance to work for the folder in the new location.
6. Click the **Confirm Move Folder** button.

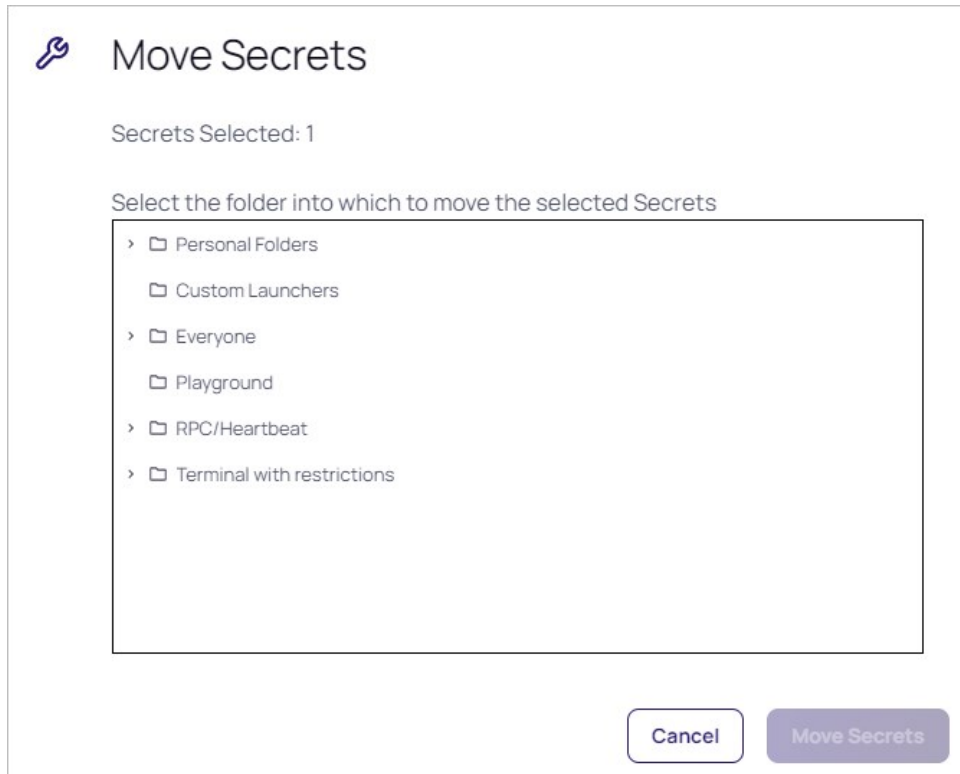
Note: You can also right click the source folder and select **Move** to see an alternative method of moving folders where you are prompted to select the destination folder.

Moving Secrets Between Folders

1. Consider the following before moving a secret between folders:
 - To add or move a secret to a folder, you must have edit permission on that folder (either direct or through inheritance).
 - To move a secret from a folder, you must have edit permission on that secret. If the secret has the "Inherit Permissions from folder" setting enabled, then you must have owner permission to move that secret to a new folder.
 - When a secret is moved to a folder, it automatically gets the "Inherit Permissions from folder" setting even if it had specific permissions before the move.
2. Navigate to the folder containing the secret or secrets you want to move.
3. Click to select the check box on the left for all the secret you want to move. The Bulk Actions button appears at the bottom of the page.
4. Click the **Bulk Actions** button. A popup appears:



5. Click the **Move to Folder** link. The Move Secrets pop-up page appears:



6. Navigate to and select the target folder for the secret or secrets.
7. Click the **Move Secrets** button. The Bulk Progress popup appears.
8. The secret moves to the selected folder.

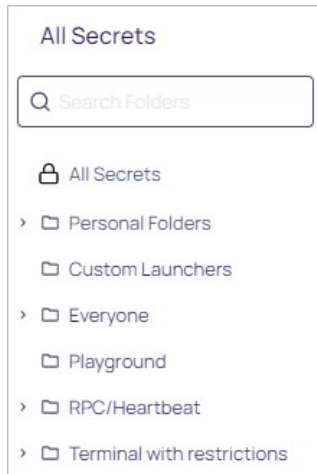
Pinning Folders

You can pin a folder, limiting the folder tree to showing the contents of that pinned folder. Applying filters to the table in one pinned folder will not affect others, and settings are remembered upon returning to that pinned folder.

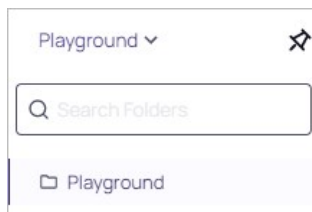
Pinning Folders

To pin a folder:

1. Click > > on the **Secrets** menu item to view the Secret Folder-Tree Panel.



2. Drill down and click the folder you wish to pin. We chose the Playground folder.
3. Right click the folder and select **Pin**. The chosen folder appears alone on the panel with a thumbtack icon next to it:



4. Now, when click the folder dropdown list at the top you have a choice of pinned folders to choose from. All Secrets is permanently pinned.



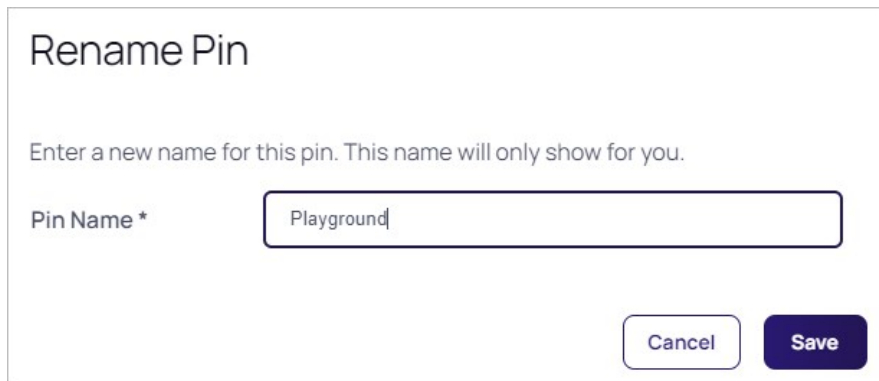
Unpinning Folders

1. Click > > on the **Secrets** menu item to view the Secret Folder-Tree Panel.
2. Select the folder in the folders dropdown list. The panel shows only that pinned folder.
3. Click the thumbtack icon and select **Unpin**.

Custom Naming Pinned Folders

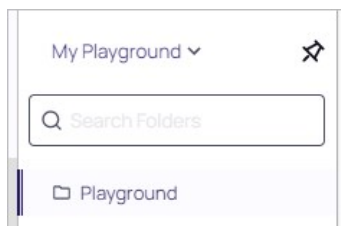
You do not have to use the default name of the top folder as the pin's name. To change it:

1. Click >> on the **Secrets** menu item to view the Secret Folder-Tree Panel.
2. Select the folder in the folders dropdown list. The panel shows only that pinned folder.
3. Click the thumbtack icon and select **Rename**. A Rename Pin popup appears:



The image shows a 'Rename Pin' dialog box. At the top, it says 'Rename Pin'. Below that, it says 'Enter a new name for this pin. This name will only show for you.' There is a text input field labeled 'Pin Name *' with the text 'Playground' inside. At the bottom right, there are two buttons: 'Cancel' and 'Save'.

4. Type the new name in the **Pin Name** text box.
5. Click the **Save** button. The Folder dropdown list now displays a different name than the top folder in the pin (we chose My Playground):



The image shows a folder dropdown list. At the top, it says 'My Playground' with a dropdown arrow and a thumbtack icon. Below that, there is a search box labeled 'Search Folders'. At the bottom, there is a folder icon and the text 'Playground'.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The secrets menu item in the new UI has a collapsible secret folder-tree panel that occupies the full height of the page. This replaces the legacy folder tree at the bottom of the left navigation bar and delivers a significantly improved folder browsing experience. To view the panel, click the >> icon on the **Secrets** menu item.

Mobile Computing

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This section addresses issues related to Secret Server's interaction with mobile devices, such as smart phones.

Overview

The **Maximum Time for Offline Access on Mobile Devices** setting in Secret Server determines how long to cache secret data on the mobile device. Once the device is not in contact with the server for longer than the specified amount of time, the device removes its cache of the stored secrets. The only way to view secrets on the device once the cache is cleared is to connect to Secret Server again so that the secrets can be re-downloaded and cached.

Procedure

To set the maximum time:

1. In Secret Server dashboard, click **Admin > Configuration**. The Edit Configuration page appears:

The screenshot shows the 'Configuration' page in the Secret Server dashboard. The 'General' tab is selected. The 'APPLICATION SETTINGS' section is visible, containing several configuration items. The 'Maximum Time for Offline Access on Mobile Devices' is set to 30 days.

APPLICATION SETTINGS	
Allow Automatic Checks for Software Updates	Yes
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	Yes View Metric Data
View Webservices	
Enable Webservices	Yes
Maximum Time for Offline Access on Mobile Devices	30 days
Session Timeout for Webservices	20 minutes
Enable Refresh Tokens for Web Services	No
Prevent Application from Sleeping When Idle	Yes

2. On the **General** tab, click the **Edit** button at the bottom of the page.
3. Click to select the **Enable Webservices** check box in the **Application Settings** section:

[View Webservices](#)

Enable Webservices

[Maximum Time Offline Explanation](#)

Maximum Time for Offline Access on Mobile Devices

Days

Hours

Session Timeout for Webservices

Unlimited

Days

Hours

Minutes

Enable Refresh Tokens for Web Services

4. Type your preferred interval in the **Days** and **Hours** text boxes in the **Maximum Time for Offline Access on Mobile Devices** section.

Note: Setting the Maximum Time Offline to less than an hour prevents the device from caching as the cache window is too small.

Note: Because caching all secrets creates an audit record in the database for each secret, we recommend not setting the window too short so that users constantly need to cache all secrets.

5. Click the **Save** button at the bottom of the page.

Example

An example of a cache window:

If Maximum Offline Time is set to 7 days, an iPhone user can cache secrets. If the iPhone has connectivity every hour the iPhone is used, it will check in with the server. Each time the iPhone checks in the 7 days, the cache window is extended. Thus, if the user uses the app once every 7 days, the app cache will remain. If the user does not have connectivity (such as in Airplane Mode) or does not turn on the app for longer than 7 days, then the next time the app is used the cache will be cleared because the maximum allowed time offline has been surpassed.

Secret Server Networking

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: As of SQL Server 2008 R2, Microsoft no longer recommends changing the password of the SQL service via the services console. According to Microsoft Documentation for SQL Server 2008 R2 and above: *Changing a SQL Server service by using the Windows Service Control Manager (services.msc) application does not always change all of the necessary settings and might prevent the service from functioning properly. However, in a clustered environment, after changing the password on the active node by using SQL Server Configuration Manager, you must change the password on the passive node by using the Service Control Manager.*

Requirements

- a PowerShell script that you will create
- a SQL dependency changer that you will create
- a secret that contains the AD credentials of the SQL instances
- a secret that contains the AD administrative credentials for PowerShell to run with
- SQL services imported via discovery

Create a PowerShell Script in Secret Server

Note: You must have administrative permission to add the script in Secret Server.

1. In Secret Server, go to **Admin > Scripts > PowerShell > Create New**.
2. In the **New PowerShell Script** dialog, fill in the following values:
 - **Name:** SQL Service Password Changer
 - **Description:** Changes SQL service account's password without disrupting the Services



New PowerShell Script

Name: SQL Service Password C *

Description: Changes SQL service account's password without disrupting the Services *

Category: Dependency

```
1 $TargetComputer = $args[0]
2 Write-Debug $TargetComputer
3 $domain= $args[1]
4 Write-Debug $domain
5 $UName= $args[2]
6 Write-Debug $uname
7 $PWord= $args[3]
8 Write-Debug $PWord
9 $$SQLService=$args[4]
10 Write-Debug $$SQLService
11 $$SvcAcctUsr=$args[5]
12 Write-Debug $$SvcAcctUsr
13 $$SvcAcctPWD=$args[6]
14 Write-Debug $$SvcAcctPWD
15
16 $Spassword = ConvertTo-SecureString $Pword -AsPlainText -Force #Secure PW
17 $creds = New-Object System.Management.Automation.PSCredential ("$domain\$UName", $Sp
18
19 $ScriptBlock= {
20 param($$SQLService,$TargetComputer,$domain,$username,$pword)
21 "SQL Service: $TargetComputer $domain $username $pword" | Out-Null
22
```

OK Cancel

3. In the **Script** field, paste in the script below:

```

$TargetComputer = $args[0]
Write-Debug $TargetComputer
$domain= $args[1]
Write-Debug $domain
$UName= $args[2]
Write-Debug $uname
$PWord= $args[3]
Write-Debug $PWord
$SQLService=$args[4]
Write-Debug $SQLService
$SvcAcctUsr=$args[5]
Write-Debug $SvcAcctUsr
$SvcAcctPWD=$args[6]
Write-Debug $SvcAcctPWD

$Spassword = ConvertTo-SecureString $Pword -AsPlainText -Force #Secure PW
$creds = New-Object System.Management.Automation.PSCredential ("$domain\$UName", $Spassword) #Set credentials for PSCredential logon

$ScriptBlock= {
param($SQLService,$TargetComputer,$domain,$username,$pword)
"$SQLService,$TargetComputer,$domain,$username,$pword" | out-file test.txt -Append
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SqlWmiManagement") | out-null
$SMOWmiserver = New-Object ('Microsoft.SqlServer.Management.Smo.Wmi.ManagedComputer') # $TargetComputer
try
{
#Specify the "Name" (from the query above) of the one service whose Service Account you want to change.
$domainuser="$domain$username"
$ChangeService=$SMOWmiserver.Services | where {$_.displayname -eq $SQLService -or $_.name -eq $SQLService } #Make sure this is what you want changed!
#$ChangeService | out-file c:\temp\test.txt -append # Remove this line for production for debugging and development only. Requires you create a temp directory on the
targetmachine.
$ChangeService.ChangePassword("$pword", "$pword")
$ChangeService.Alter()
}
catch
{
$ErrorMessage = $_.Exception.Message
$FailedItem = $_.Exception.ItemName
throw "Error $ErrorMessage : $FailedItem while setting $sqlservice on $targetComputer with $domainuser"
}
}

```

4. Create secure credentials to access the SQL box using a modified invoke-command. Add \$domain, \$SvcAcctUsr, and \$SvcAcctPWD for the domain, username, and password, respectively. For example:

```

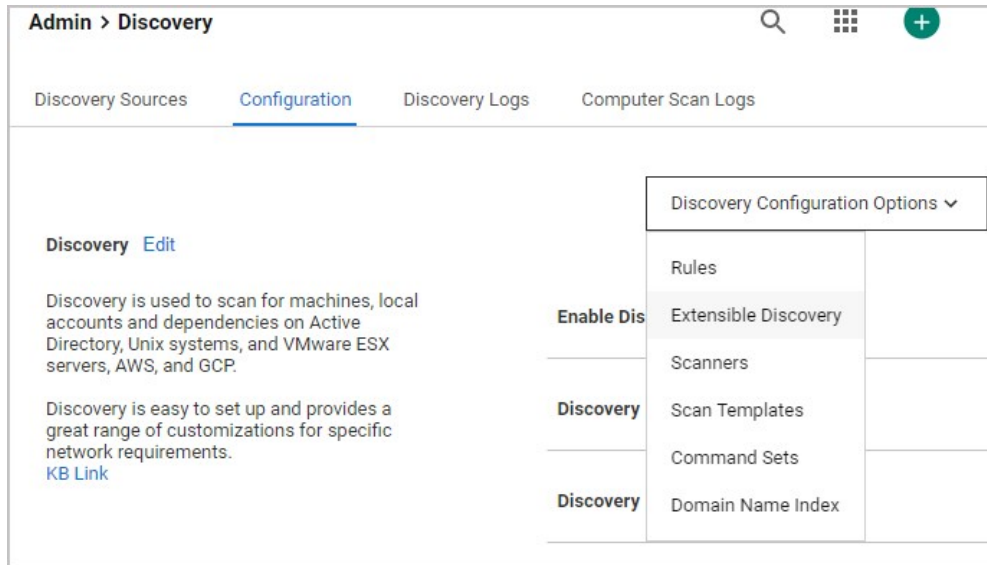
Invoke-Command
-Authentication Default
-ComputerName $TargetComputer
-ScriptBlock $ScriptBlock
-ArgumentList $SQLService,$TargetComputer,$domain,$SvcAcctUsr,$SvcAcctPWD
-Credential $creds

```

Note: You might want to test the Script in PowerShell ISE. If the script succeeds, the SQL password has changed. You can test out on a SQLEXPRESS.

Create a New Dependency Changer

1. Got to **Admin > Discovery**.
2. Click the **Configuration** tab.



3. Click **Discovery Configuration Options**.
4. From the drop-down menu, click **Extensible Discovery**.
5. Click **Configure Dependency Changers**.
6. Click **Create New Dependency Changer**.
7. On the **Basic** tab, enter the following:
 - **Type:** PowerShell Script
 - **Scan Template:** Windows Service
 - **Name:** SQL Service Dependency Changer
 - **Description:** SQL Service Dependency Changer

New Dependency Changer ✕

Basic Scripts

Explain

Type ▼
Powershell Script

Scan Template ▼ ?
Windows Service

Name *
SQL Service Dependency Changer

Description

Port

Wait (s) ?
0

Enabled

Create Template ?

8. On the **Scripts** tab, enter the following:

- **Script:** (Select the script you just created).
- **Arguments:** \$MACHINE \$DOMAIN \${1}\$USERNAME \${1}\$PASSWORD \$SERVICENAME \$USERNAME \$PASSWORD

New Dependency Changer ✕

Basic **Scripts**

Explain

Use advanced scripts ?

▼ Change Script SQL Service Password Changer ✓

Script ▼ * 👁
SQL Service Password Changer

Arguments

```
$MACHINE $DOMAIN ${1}$USERNAME
${1}$PASSWORD $SERVICENAME
$USERNAME $PASSWORD
```

9. Click **Save**.

Discovery and Importing to the Right Template

When discovery finds your SQL services and you start importing them, you'll be able to choose the template you just created as well as your privileged account, which will be running PowerShell. Once the import is finished, associate a Secret on the **Remote Password Changing** tab, which is the same as the privileged account used during the import process. Your SQL dependencies will use the SQL PowerShell changer you created, and the passwords will be changed *without* restarting SQL.

Note: This is for Secret Server version 7.1 and later.

Once Secret Server is installed, it may be necessary to change the connection string that Secret Server uses to connect to its database. You must be authenticated to access Secret Server and have the Administer Configuration role permission.

1. Click **Admin > See All**.
2. Type **Database** in the search text box and select **Database** in the dropdown list. The Database Configuration page appears:

Help

Secret Server supports Microsoft SQL Server versions 2012, 2014, 2016, 2017, and Express.

View [Collation Requirements](#). Need help installing or configuring SQL Server? View [Installing and Configuring SQL KB Article](#).

Database Configuration

SQL SERVER LOCATION

Server Name	QA-CUST-SQL-01
Database	SS_Playground

SQL AUTHENTICATION

Windows Authentication using Application Identity (GAMMA\ss_iis_svc) - **Recommended**
(Advanced Setting. Windows Authentication requires a Service Account and advanced IIS setup. Detailed in [Windows Authentication KB](#).)

SQL Server Authentication *(SQL Authentication requires Mixed Mode. See setup in [SQL Authentication KB](#).)*

[+] ADVANCED (NOT REQUIRED)

Edit View Audit

3. Click the **Edit** button. The page enters edit mode:

Help

Secret Server supports Microsoft SQL Server versions 2012, 2014, 2016, 2017, and Express.

View [Collation Requirements](#). *Need help installing or configuring SQL Server? View [Installing and Configuring SQL KB Article](#).*

Database Configuration

This page modifies the Secret Server database connection settings, which are stored in C:\inetpub\wwwroot\Playground\database.config. This file can be backed up to revert or simply return to this page to reset the connection again. If you need to modify TMS database settings navigate to /setup/database/connectdatabase in the TMS web site.

SQL SERVER LOCATION

Server Name *

For example:
localhost
(local)
MYDBSERVER
localhost\SQLEXPRESS

Database *

SQL AUTHENTICATION

Windows Authentication using Application Identity (GAMMA\ss_iis_svc) - **Recommended**
(Advanced Setting. Windows Authentication requires a Service Account and advanced IIS setup. Detailed in [Windows Authentication KB](#).)

SQL Server Authentication *(SQL Authentication requires Mixed Mode. See setup in [SQL Authentication KB](#).)*

[+] ADVANCED (NOT REQUIRED)

4. Edit the parameters as desired.
5. Click the **Save Database Connection Settings** button. A confirmation message appears. Secret Server recycles its application pool (needed to clear the connection string cache), and then returns you to the Secret Server dashboard.

To query Secret Server status without authentication for basic latency check, follow the steps below.

1. In a web browser, go to <https://yoursecretserverurl/healthcheck.aspx>
2. Compare the information displayed in your browser to the information below:

```
{"healthy":true,"now":"2019-04-08T12:59:06.0455458-04:00","utcNow":"2019-04-08T16:59:06.0455458Z"}
```

- If your information is similar, your Secret Server should be operational.
- If your information displays other text such as **timed out** or **service unavailable**, there may be issues with the web site where the application is installed.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: Distributed engine configurations for Secret Server On-Premises and Secret Server Cloud are not 100% equivalent. To view the differences, please refer to our [reference architectures](#) for Secret Server. They are, at minimum, refreshed every year and are created by our Professional Services Solutions Architect team.

Overview

Out of the box, Secret Server performs all functions from the Web server it is installed on; however, specific features can be routed through a distributed engine for enhanced performance. For example, synchronize and authenticate AD users can be done in Secret Server via your local site or from a distributed engine (DE).

You can install a DE in a remote site and allow it to operate many functions. Communication with Secret Server Cloud also requires the distributed engine to be installed.

Architecture and Workflow

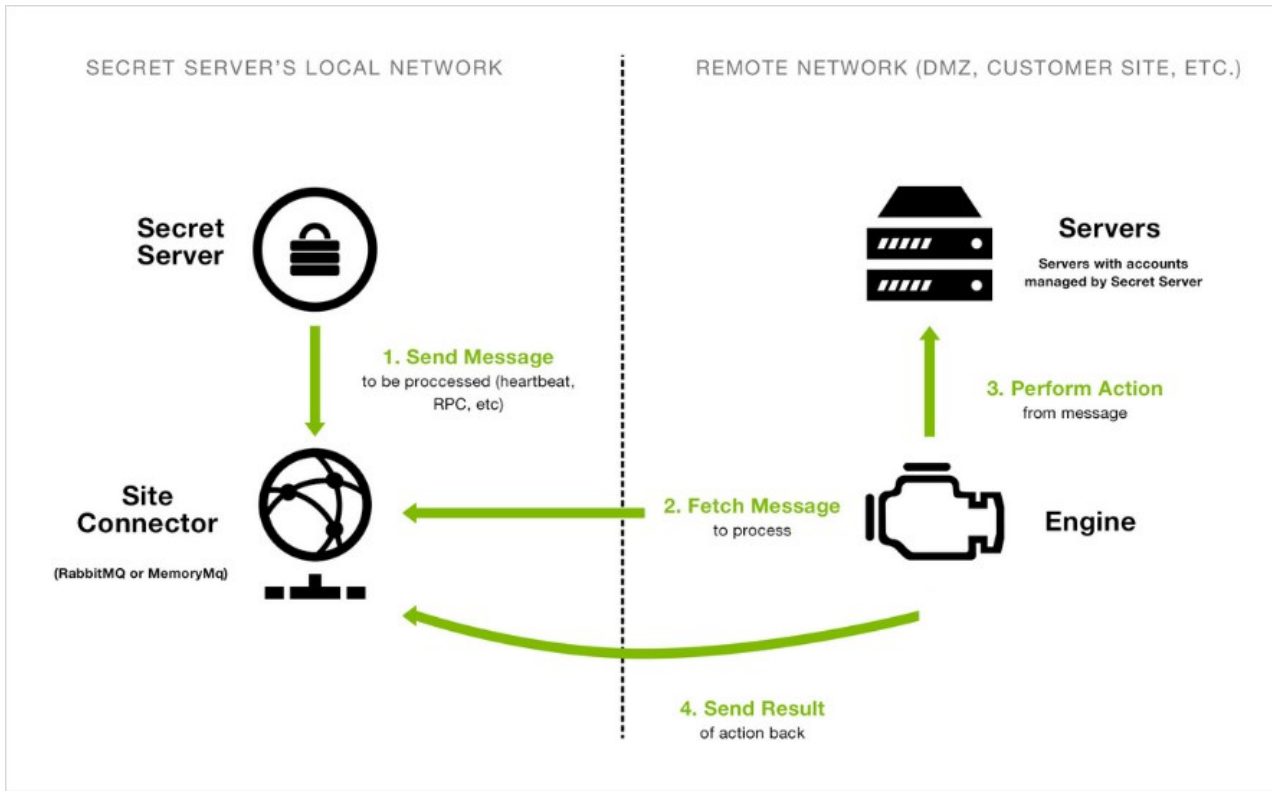
Main Components

DEs support heartbeat, Remote Password Changing (RPC), and discovery. A DE is composed of site connectors, sites, and engines:

- An **engine** is a Windows service that does the actual work, such as password changing, heartbeat, Discovery, and more. Each engine belongs to a site.
- A **site** can be thought of as a bucket of work items for a particular network area. Each engine is assigned to a single site, but each site can include multiple engines, significantly increasing throughput.
- A **site connector** is a Windows service that holds the work items for a number of sites. The site connector can be either [RabbitMQ](#) or MemoryMQ (a built-in service developed by Delinea). Each site can only be assigned to a single site connector, but you can have multiple site connectors running on separate machines, each storing work items for multiple sites. Those sites, in turn, distribute the work items among multiple engines. The ability to add new Site Connectors, Sites, and Engines as needed makes Distributed Engine a highly-scalable solution.

Note: For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative for customers who do not need many engines or sites.

Figure: Distributed Engine Components



Note: The above diagram is a simplified, conceptual one, not a network diagram. It does not show a callback port from the DE to Secret Server. DEs require either an HTTPS or TCP port to communicate with Secret Server for initial activation, updates, and continuous periodic check of site and site connector settings.

Ports

DEs have two configurable ports: one for connecting to the site connector, and one for the engine to retrieve configuration information from Secret Server at regular intervals. The callback port from an engine to Secret Server can be configured to contact the website directly over HTTP, HTTPS, or TCP. HTTP and HTTPS connections use the existing IIS port bindings. All connections are outbound—no inbound connections are made from Secret Server or the site connectors to the remote networks.

Note: If using Secret Server Cloud, port 9354 must also be opened for outbound messages.

Default ports:

- RabbitMQ: 5672 (non-SSL), 5671 (SSL)
- MemoryMQ: 8672 (non-SSL), 8671 (SSL)
- Secret Server: existing IP address bindings or custom port over TCP. We reserve one port for legacy upgrades, usually port 9999.
- Secret Server Cloud:
 - 443 (Web sockets—the default)
 - 5671 and 5672 (AMQP)

Note: These ports are used for outbound traffic for engines to communicate with SSC instances. They are set by the "Azure ServiceBus Transport Type" global engine setting.

Security

Distributed engines have multiple security layers:

- Engines must be approved within Secret Server before they will be given access to a site.
- Work items are encrypted with a site-specific symmetric key prior to sending them to the site connector.
- Communication to the site connector supports SSL and TLS.
- Direct communication from engine to Secret Server uses a public-private key exchange.
- The engine configuration file is DPAPI encrypted.

For more information about DE security, see the [Distributed Engine Security Guide](#).

Engine Workflow

When an engine Windows service starts, the following steps occur:

1. The service contacts Secret Server directly using the engine callback port.
2. The service receives configuration information for the site connector to connect to and what site to process work items for.
3. The service connects to the site connector and registers with the site for work item processing.
4. The service fetches a work item from the site.
5. The service processes the work item.
6. The service gives the site the result of the processing, such as heartbeat success or discovery results.
7. The service fetches another work item, and the process continues.

Configuring Distributed Engines

Below is a summary of the steps required to configure DEs:

1. Enable the DE and specify the engine callback settings.
2. Configure and Install the site connector.
 - If you plan to use RabbitMQ (recommended), please see [Installing RabbitMQ](#). You can find general information on using RabbitMQ Helper to install RabbitMQ can be found in [Delinea's GitHub Repository](#)
 - If you plan to use MemoryMQ, create the site connector record within Secret Server then click the **Download Site Connector Installer** button to get the MSI. Run the MSI on the desired host.
3. Setup sites.
4. Install engines.
5. Assign secrets to sites. Secrets can be assigned to a site through their Remote Password Changing tab or via a bulk operation on the Secret Server dashboard. Once assigned to a site, all heartbeat or password changing operations take place through that site.
6. Assign discovery sources to sites. To run discovery through a site, edit the discovery source and assign the site. Once assigned, all discovery operations for that discovery source take place through that site.

FAQ

What happens if Secret Server sends work items to the site connector, but no engines are running to consume them?

Work items continue to build up in the site connector until a limit is reached. Heartbeat work items have a Time To Live (TTL) of 5 minutes, Password Changing work items have a TTL of 20 minutes. Expired work items are thrown away and will not be processed. Once a heartbeat or password changing work item is sent to the site connector, Secret Server will not send the same work item to the queue until 5 minutes after the TTL is up (10 and 25 minutes for heartbeat and password changing, respectively). This prevents multiple pending heartbeat or password changing work items for the same secret at the same time.

How many Sites can a Site Connector hold?

MemoryMQ supports up to 100. RabbitMQ supports up to 200.

Can I cluster Site Connectors?

RabbitMQ supports clustering, MemoryMQ does not.

Can I use both RabbitMQ and MemoryMQ?

Yes. You can have as many site connectors, of either type, installed as needed. Note that while you can have both RabbitMQ and MemoryMQ installed on a single machine, you cannot have two RabbitMQ instances or two MemoryMQ instances on the same machine.

Can I convert a site connector from MemoryMQ to RabbitMQ or vice versa?

Yes. You can install the new site connector, swap the sites over to the new service, and then decommission the old site connector.

Configuration and Sizing

Requirements

Windows Server 2012

Starting in Secret Server version 8.9.000000, DEs require that one of following two server features be installed when the Secret Server website is running on a Windows Server 2012. This depends on which protocol is selected in the engine's callback settings. If HTTPS is selected, the HTTP activation is required. If TCP is selected, then TCP activation is required. This accomplished by going to one of the following in Windows Server 2012:

- **.NET Framework 4.5 Features > WCF Services > HTTP Activation**
- **.NET Framework 4.5 Features > WCF Services > TCP Activation**

If the feature is not installed, there will be an error message in the DE logs: (405) Method Not Allowed. ---> System.Net.WebException: The remote server returned an error: (405) Method Not Allowed.

Distributed Engine Offline and Online Events

As of version 10.7.000059, Delinea updated the definition of distributed engines' offline status to be the configured heartbeat interval times three. For instance, if your heartbeat interval is configured at 5 minutes, the engine will report offline if Secret Server and the engine do not successfully communicate within a 15-minute time period. Engine online and offline states were also added to subscription actions to allow notification to admins when engine states change.

Internal Site Connector

Starting in Secret Server 10.2 it became possible to change how Secret Server processes messages by navigating to:

<Your Secret Server URL>/AdminBackboneBusConfigurationView.aspx

These messages are generated and placed on the internal site connector, or backbone bus, every time a background operation is triggered whether by a schedule or on-demand.

The internal site connector receives and processes messages as a result of numerous actions:

- Bulk Operations
- Generate Password
- Secret Import (CSV and XML)
- Run Heartbeat Now
- Run Heartbeat (Scheduled)
- Run Password Change Now
- Run Password Change (Scheduled)
- Run Discovery Now
- Run Discovery (Scheduled)
- Run AD Sync Now
- Run AD Sync (Scheduled)
- Elements of Session Recording

The internal site connector, using the internal hosted bus, is adequate for bulk operations, heartbeat, discovery, and the like, but some Secret Server features, such as a clustered Web server node configuration or session recording, require a scalable messaging solution to boost processing performance. Our choice is [RabbitMQ](#), which is an intermediary messaging broker that can handle large-scale message processing.

The following is a typical internal hosted bus operation (for a bulk operation):

1. A Secret Server user triggers the a bulk operation.
2. A message is formed and sent over a TCP connection to the internal hosted bus.
3. Secret Server (on the same machine) receives the message.
4. Secret Server (on the same machine) processes the message.

We continually improve the internal hosted bus but still recommend RabbitMQ for a scalable performance boost. See [Installing RabbitMQ](#) for more information.

RabbitMQ Durable Exchanges

Overview

As of Secret Server 10.7.59, the Secret Server MessageQueue Client attempts to create RabbitMQ durable exchanges, logging the activity. A durable exchange is normally automatically re-created if RabbitMQ restarts for any reason. Any legacy non-durable exchanges disappear when RabbitMQ goes down and can only be manually recreated.

If the MessageQueue client detects that creating a durable exchange failed, it will log an error and attempt to create a non-durable one.

Important: Any existing non-durable exchanges, from previous versions of Secret Server, will also cause durable exchange creation to fail. See [Manually Creating Durable RabbitMQ Exchanges](#).

Non-durable RabbitMQ exchanges for Secret Server would look similar to this, whether created by an earlier Secret Server version or by a durable-version-creation failure:

The screenshot shows the 'Exchanges' tab in a management console. It displays a table with 10 exchanges. The 'Features' column for all exchanges is empty, indicating they are non-durable. The table includes columns for Name, Type, Features, Message rate in, Message rate out, and a +/- toggle.

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct				
amq.direct	direct				
amq.fanout	fanout				
amq.headers	headers				
amq.match	headers				
amq.rabbitmq.trace	topic				
amq.topic	topic				
thycotic-sr-agent-response	topic		0.00/s	0.00/s	
thycotic-ss	topic		0.00/s	0.00/s	
thycotic-ss-engine-response	topic				

Note the absence of a 'D' in the Features column, meaning that exchange is not durable. Durable exchanges, created by the current Secret Server version (10.7.59+), look like this:

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			
thycotic-sessionrec	topic	D	0.00/s	0.00/s	
thycotic-sr-agent-response	topic				
thycotic-ss	topic	D	0.20/s	0.20/s	
thycotic-ss-engine-response	topic	D			

Earlier versions of Secret Server (before 10.7.59) created non-durable RabbitMQ exchanges during a Secret Server server or IIS restart. If the environment is clustered, the same is true of every node in that cluster. The current durable exchanges persist during any IIS restart, eliminating the need to restart Secret Server or recreate the exchanges.

However, any existing non-durable exchanges prevent the creation of the newer durable ones. To remedy that, you need to restart all of the RabbitMQ servers in the cluster at the same time or manually delete the non-durable exchanges.

Manually Creating Durable RabbitMQ Exchanges

To enjoy the benefits of the durable exchanges, you must first eliminate any legacy non-durable exchanges from your RabbitMQ server or servers. There are two ways to do this:

- Restart the RabbitMQ server or all of the RabbitMQ servers in the cluster at the same time. You can also stop the RabbitMQ service in `services.msc`.

Note: Customers usually reset or turn off all servers via third party tools, but some prefer to shut off the service via `services.msc` because of their system configuration.

- Delete the exchanges manually:
 - Click to select each Secret Server non-durable exchange, including distributed engine ones.
 - Scroll to the bottom of the window.
 - Click the **Delete** button.
 - Restart all of the Secret Server instances and distributed engines to recreate the exchanges and connections.

Creating Durable RabbitMQ Exchanges with a PowerShell Script

Using the Script

```
powershell.exe -file exchangedurability.ps1 -username "guest" -password "guest" -computerName "localhost" -port "15672"
```

The user has access to the RabbitMQ admin interface. The computername and port is where the admin interface is located.

The script:

1. Removes all of the exchanges that are not durable and any that are not the `thycotic-sr*` ones for legacy ASRAs.
2. Kills all of the connections. This forces the distributed engines and Secret Server to reconnect to the durable exchanges.

Script

```
param([string] $computerName = "",
      [string] $userName = "",
      [string] $password = "",
      [string] $port = ""
)

$defaultComputerName = if ($computerName -eq "") { "localhost" } else { $computerName }
$defaultVirtualhost = "/"
$defaultUserName = if ($userName -eq "") { "guest" } else { $userName }
$defaultPassword = if ($password -eq "") { "guest" } else { $password }
$defaultPort = if ($port -eq "") { "15672" } else { $port }
$defaultHttp = "http" #Use https if ssl

$defaultCredentials = New-Object System.Management.Automation.PSCredential ($defaultUserName, $(ConvertTo-SecureString $defaultPassword -AsPlainText -Force))

#LICENSE FOR LINKS - All the RabbitMQ PowerShell calls are based on this:
#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/LICENSE
#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetConnection.ps1

function Get-RabbitMQConnection
{
    [CmdletBinding(DefaultParameterSetName='defaultLogin', SupportsShouldProcess=$true, ConfirmImpact='None')]
    Param
    (
        # Name of RabbitMQ Connection.
        [parameter(ValueFromPipeline=$true, ValueFromPipelineByPropertyName=$true)]
        [Alias("Connection", "ConnectionName")]
        [string[]]$Name = "",

        # Name of the computer hosting RabbitMQ server. Default value is localhost.
        [parameter(ValueFromPipelineByPropertyName=$true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,

        # UserName to use when logging to RabbitMq server.
        [Parameter(Mandatory=$true, ParameterSetName='login')]
        [string]$UserName,

        # Password to use when logging to RabbitMq server.
        [Parameter(Mandatory=$true, ParameterSetName='login')]
        [string]$Password,

        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory=$true, ParameterSetName='cred')]
        [PSCredential]$Credentials
    )

    Begin
    {
        $Credentials = NormaliseCredentials
    }
    Process
    {
        if ($?scmdlet.ShouldProcess("server $ComputerName", "Get connection(s): $(NamesToString $Name '(all)')"))
        {
            $result = GetItemsFromRabbitMQApi -ComputerName $ComputerName $Credentials "connections"

            $result = ApplyFilter $result 'name' $Name

            $result | Add-Member -NotePropertyName "ComputerName" -NotePropertyValue $ComputerName

            SendItemsToOutput $result "RabbitMQ.Connection"
        }
    }
}
```

```

    }
    End
    {
    }
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOnUri.ps1
if (-not $UnEscapeDotsAndSlashes) { Set-Variable -Scope Script -name UnEscapeDotsAndSlashes -value 0x2000000 }
function GetUriParserFlags {

    $getSyntax = [System.UriParser].GetMethod("GetSyntax", 40)
    $flags = [System.UriParser].GetField("m_Flags", 36)

    $parser = $getSyntax.Invoke($null, "http")
    return $flags.GetValue($parser)
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOnUri.ps1
function SetUriParserFlags([int]$newValue) {
    $getSyntax = [System.UriParser].GetMethod("GetSyntax", 40)
    $flags = [System.UriParser].GetField("m_Flags", 36)

    $parser = $getSyntax.Invoke($null, "http")
    $flags.SetValue($parser, $newValue)
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOnUri.ps1
function PreventUnEscapeDotsAndSlashesOnUri {
    if (-not $uriUnEscapesDotsAndSlashes) { return }

    Write-Verbose "Switching off UnEscapesDotsAndSlashes flag on UriParser."

    $newValue = $defaultUriParserFlagsValue -bxor $UnEscapeDotsAndSlashes

    SetUriParserFlags $newValue
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOnUri.ps1
function RestoreUriParserFlags {
    if (-not $uriUnEscapesDotsAndSlashes) { return }

    Write-Verbose "Restoring UriParser flags - switching on UnEscapesDotsAndSlashes flag."

    try {
        SetUriParserFlags $defaultUriParserFlagsValue
    }
    catch [System.Exception] {
        Write-Error "Failed to restore UriParser flags. This may cause your scripts to behave unexpectedly. You can find more at get-help about_UnEscapingDotsAndSlashes."
        throw
    }
}

if (-not $defaultUriParserFlagsValue) { Set-Variable -Scope Script -name defaultUriParserFlagsValue -value (GetUriParserFlags) }
if (-not $uriUnEscapesDotsAndSlashes) { Set-Variable -Scope Script -name uriUnEscapesDotsAndSlashes -value (($defaultUriParserFlagsValue -band $UnEscapeDotsAndSlashes) -eq $UnEscapeDotsAndSlashes) }

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/Invoke_RestMethodProxy.ps1
function Invoke-RestMethod {
    [CmdletBinding(HelpUri = 'http://go.microsoft.com/fwlink/?LinkID=217034')]
    param(
        [Microsoft.PowerShell.Commands.WebRequestMethod]
        ${Method},

        [Parameter(Mandatory = $true, Position = 0)]
        [ValidateNotNullOrEmpty()]
        [uri]
        ${Uri},

        [Microsoft.PowerShell.Commands.WebRequestSession]
        ${WebSession},

        [Alias('SV')]
        [string]
        ${SessionVariable},
    )
}

```



```
[pscredential]
${Credential},

[switch]
${UseDefaultCredentials},

[ValidateNotNullOrEmpty()]
[string]
${CertificateThumbprint},

[ValidateNotNull()]
[System.Security.Cryptography.X509Certificates.X509Certificate]
${Certificate},

[string]
${UserAgent},

[switch]
${DisableKeepAlive},

[int]
${TimeoutSec},

[System.Collections.IDictionary]
${Headers},

[ValidateRange(0, 2147483647)]
[int]
${MaximumRedirection},

[uri]
${Proxy},

[pscredential]
${ProxyCredential},

[switch]
${ProxyUseDefaultCredentials},

[Parameter(ValueFromPipeline = $true)]
[System.Object]
${Body},

[string]
${ContentType},

[ValidateSet('chunked', 'compress', 'deflate', 'gzip', 'identity')]
[string]
${TransferEncoding},

[string]
${InFile},

[string]
${OutFile},

[switch]
${PassThru},

[switch]
${AllowEscapedDotsAndSlashes})

begin {
  try {
    $outBuffer = $null
    if ($PSBoundParameters.TryGetValue('OutBuffer', [ref]$outBuffer)) {
      $PSBoundParameters['OutBuffer'] = 1
    }

    $wrappedCmd = $ExecutionContext.InvokeCommand.GetCommand('Microsoft.PowerShell.Utility\Invoke-RestMethod',
[System.Management.Automation.CommandTypes]::Cmdlet)

    # check whether need to disable UnEscapingDotsAndSlashes on UriParser
    $requiresDisableUnEscapingDotsAndSlashes = ($AllowEscapedDotsAndSlashes -and $Uri.OriginalString -match '%2f')

    # remove additional proxy parameter to prevent original function from failing
```

```
if ($PSBoundParameters['AllowEscapedDotsAndSlashes']) { $null = $PSBoundParameters.Remove('AllowEscapedDotsAndSlashes') }

$scriptCmd = { & $wrappedCmd @PSBoundParameters }
$steppablePipeline = $scriptCmd.GetSteppablePipeline($myInvocation.CommandOrigin)
$steppablePipeline.Begin($PSCmdlet)
}
catch {
    throw
}
}

process {
    try {
        # Disable UnEscapingDotsAndSlashes on UriParser when necessary
        if ($requiresDisableUnEscapingDotsAndSlashes) {
            PreventUnEscapeDotsAndSlashesOnUri
        }

        $steppablePipeline.Process($_)
    }
    finally {
        # Restore UnEscapingDotsAndSlashes on UriParser when necessary
        if ($requiresDisableUnEscapingDotsAndSlashes) {
            RestoreUriParserFlags
        }
    }
}

end {
    try {
        $steppablePipeline.End()
    }
    catch {
        throw
    }
}
}
<#
.ForwardHelpTargetName Invoke-RestMethod
.ForwardHelpCategory Cmdlet
#>

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetRabbitMQCredentials.ps1
function GetRabbitMQCredentials {
    Param
    (
        [parameter(Mandatory = $true)]
        [string]$userName,

        [parameter(Mandatory = $true)]
        [string]$password
    )

    $secpasswd = ConvertTo-SecureString $password -AsPlainText -Force
    return New-Object System.Management.Automation.PSCredential ($userName, $secpasswd)
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/NamesToString.ps1
function NamesToString {
    Param
    (
        [string[]]$name,
        [string]$saltText = ""
    )

    if (-not $name) { return $saltText }

    return $name -join ';'
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/ApplyFilter.ps1
function ApplyFilter {
    Param (
        [parameter()]
        [PSCollection]$items,
    )
}
```

```

    [parameter(Mandatory = $true)]
    [string]$prop,

    [string[]]$name
)

if (-not $name) { return $items }

# apply property filter
$filter = @()
foreach ($n in $name) { $filter += '$_' + $prop + '-like "' + $n + '"' }

$sb = [scriptblock]::create($filter -join ' -or ')
return $items | ? $sb
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/NormaliseCredentials.ps1
function NormaliseCredentials() {
    switch ($PsCmdlet.ParameterSetName) {
        "defaultLogin" { return GetRabbitMqCredentials $defaultUserName $defaultPassword }
        "login" { return GetRabbitMqCredentials $UserName $Password }
        "cred" { return $Credentials }
    }
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/SendItemsToOutput.ps1
function SendItemsToOutput {
    Param
    (
        [parameter()]
        [PSObject[]]$items,

        [parameter(Mandatory = $true)]
        [string[]]$typeName
    )

    foreach ($i in $items) {
        $i.PSObject.TypeNames.Insert(0, $typeName)
        Write-Output $i
    }
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetItemsFromRabbitMQApi.ps1
function GetItemsFromRabbitMQApi {
    [CmdletBinding(DefaultParameterSetName = 'login')]
    Param
    (
        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 0)]
        [string]$cn,

        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 1)]
        [string]$userName,

        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 2)]
        [string]$password,

        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 3)]
        [string]$fn,

        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 0)]
        [string]$computerName,

        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 1)]
        [PSCredential]$cred,

        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 2)]
        [string]$function
    )

    Add-Type -AssemblyName System.Web
    #Add-Type -AssemblyName System.Net

    if ($PsCmdlet.ParameterSetName -eq "login") {

```

```

    $computerName = $cn
    $cred = GetRabbitMqCredentials $userName $password
    $function = $fn
}
Write-Output $computerName
$url = $defaultHttp + "://" + ([System.Web.HttpUtility]::UrlEncode($computerName)):$defaultPort/api/$function
Write-Verbose "Invoking REST API: $url"

return Invoke-RestMethod $url -Credential $cred -DisableKeepAlive -AllowEscapedDotsAndSlashes
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetExchange.ps1
function Get-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess = $true, ConfirmImpact = 'None')]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true)]
        [Alias("ex", "Exchange", "ExchangeName")]
        [string[]]$Name = "",

        # Name of RabbitMQ Virtual Host.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("vh")]
        [string]$VirtualHost = "",

        # Name of the computer hosting RabbitMQ server. Default value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,

        # Username to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$UserName,

        # Password to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$Password,

        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'cred')]
        [PSCredential]$Credentials
    )

    Begin {
        $Credentials = NormaliseCredentials
    }
    Process {
        if ($?pscmdlet.ShouldProcess("server $ComputerName", "Get exchange(s): $(NamesToString $Name '(all)')") {
            $exchanges = GetItemsFromRabbitMQApi -ComputerName $ComputerName $Credentials "exchanges"

            $result = ApplyFilter $exchanges 'vhost' $VirtualHost
            $result = ApplyFilter $result 'name' $Name

            $result | Add-Member -NotePropertyName "ComputerName" -NotePropertyValue $ComputerName

            SendItemsToOutput $result "RabbitMQ.Exchange"
        }
    }
    End {
    }
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/RemoveExchange.ps1
function Remove-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess = $true, ConfirmImpact = "High")]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(Mandatory = $true, ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("Exchange", "ExchangeName")]
        [string[]]$Name,

        # Name of RabbitMQ Virtual Host.
        [parameter(ValueFromPipelineByPropertyName = $true)]

```

```

[Alias("vh", "vhost")]
[string]$VirtualHost = $defaultVirtualHost,

# Name of the computer hosting RabbitMQ server. Defalut value is localhost.
[parameter(ValueFromPipelineByPropertyName = $true)]
[Alias("HostName", "hn", "cn")]
[string]$ComputerName = $defaultComputerName,

# UserName to use when logging to RabbitMq server.
[Parameter(Mandatory = $true, ParameterSetName = 'login')]
[string]$UserName,

# Password to use when logging to RabbitMq server.
[Parameter(Mandatory = $true, ParameterSetName = 'login')]
[string]$Password,

# Credentials to use when logging to RabbitMQ server.
[Parameter(Mandatory = $true, ParameterSetName = 'cred')]
[PSCredential]$Credentials
)

Begin {
    $Credentials = NormaliseCredentials
    $cnt = 0
}
Process {
    if ($pscmdlet.ShouldProcess("server: $ComputerName, vhost: $VirtualHost", "Remove exchange(s): $(NamesToString $Name '(all)')") {
        foreach ($n in $Name) {
            $url = $defaultHttp +
                "://$([System.Web.HttpUtility::UrlEncode($ComputerName)):$defaultPort/api/exchanges/$([System.Web.HttpUtility::UrlEncode($VirtualHost))/$([System.Web.HttpUtility::UrlEncode($n)])"
            Write-Output $url
            $result = Invoke-RestMethod $url -Credential $Credentials -AllowEscapedDotsAndSlashes -DisableKeepAlive -ErrorAction Continue -Method Delete

            Write-Verbose "Deleted Exchange $n on server $ComputerName, Virtual Host $VirtualHost"
            $cnt++
        }
    }
}
End {
    if ($cnt -gt 1) { Write-Verbose "Deleted $cnt Exchange(s)." }
}

#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/AddExchange.ps1
function Add-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess = $true, ConfirmImpact = "Medium")]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(Mandatory = $true, ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("Exchange", "ExchangeName")]
        [string[]]$Name,

        # Type of the Exchange to create.
        [parameter(Mandatory = $true, ValueFromPipelineByPropertyName = $true)]
        [ValidateSet("topic", "fanout", "direct", "headers")]
        [string]$Type,

        # Determines whether the exchange should be Durable.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$Durable,

        # Determines whether the exchange will be deleted once all queues have finished using it.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$AutoDelete,

        # Determines whether the exchange should be Internal.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$Internal,

        # Allows to set alternate exchange to which all messages which cannot be routed will be send.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("alt")]
        [string]$AlternateExchange,
    )
}

```

```

# Name of RabbitMQ Virtual Host.
[parameter(ValueFromPipelineByPropertyName = $true)]
[Alias("vh", "vhost")]
[string]$VirtualHost = $defaultVirtualhost,

# Name of the computer hosting RabbitMQ server. Defalut value is localhost.
[parameter(ValueFromPipelineByPropertyName = $true)]
[Alias("HostName", "hn", "cn")]
[string]$ComputerName = $defaultComputerName,

# UserName to use when logging to RabbitMq server.
[Parameter(Mandatory = $true, ParameterSetName = 'login')]
[string]$UserName,

# Password to use when logging to RabbitMq server.
[Parameter(Mandatory = $true, ParameterSetName = 'login')]
[string]$Password,

# Credentials to use when logging to RabbitMQ server.
[Parameter(Mandatory = $true, ParameterSetName = 'cred')]
[PSCredential]$Credentials
)

Begin {
    $Credentials = NormaliseCredentials
}
Process {
    if ($pscmdlet.ShouldProcess("server: $ComputerName, vhost: $VirtualHost", "Add exchange(s): $(NamesToString $Name '(all)')") {

        $body = @{
            type = "$Type"
        }

        if ($Durable) { $body.Add("durable", $true) }
        if ($AutoDelete) { $body.Add("auto_delete", $true) }
        if ($Internal) { $body.Add("internal", $true) }
        if ($AlternateExchange) { $body.Add("arguments", @{"alternate-exchange" = $AlternateExchange }) }

        $bodyJson = $body | ConvertTo-Json

        foreach ($n in $Name) {
            $url =
$defaultHttp+":"/{([System.Web.HttpUtility]::UrlEncode($ComputerName)):$defaultPort/api/exchanges/{([System.Web.HttpUtility]::UrlEncode($VirtualHost))/{([System.Web.H
ttpUtility]::UrlEncode($n))"
            Write-Verbose "Invoking REST API: $url"

            $result = Invoke-RestMethod $url -Credential $Credentials -AllowEscapedDotsAndSlashes -DisableKeepAlive -ErrorAction Continue -Method Put -ContentType
"application/json" -Body $bodyJson

            Write-Verbose "Created Exchange $n on server $ComputerName, Virtual Host $VirtualHost"
            $snt++
        }
    }
}
End {
    if ($snt -gt 1) { Write-Verbose "Created $snt Exchange(s)."}
}
}

#Modified to allow + in url.
#https://github.com/mariuszwojcik/RabbitMQTools/blob/master/RemoveQueue.ps1
function Remove-RabbitMQConnection {
    Param
    (
        # Name of RabbitMQ connection.
        [parameter(Mandatory = $true, ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("ConnectionName")]
        [string] $Name = "",

        # Name of the computer hosting RabbitMQ server. Defalut value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,

        # Credentials to use when logging to RabbitMQ server.

```

```

[Parameter(Mandatory = $false)]
[PSCredential]$Credentials = $defaultCredentials
)

$url = $defaultHttp + "://" + ([System.Web.HttpUtility]::UrlEncode($ComputerName)) + ":" + $defaultPort + "/api/connections/" + ([System.Web.HttpUtility]::UrlEncode($Name))
$url = $url.Replace("+", "%20")
Write-Output $url
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("X-Reason", "Removing To Create Durable Exchanges")
$result = Invoke-RestMethod $url -Credential $Credentials -Headers $headers -DisableKeepAlive:$InvokeRestMethodKeepAlive -ErrorAction Continue -Method Delete
Write-Output "$url closed."

Write-Verbose "Closed connection $n to server $ComputerName"
}

function MakeExistingExchangesDurable() {
    Param(
        [string] $HostName = $defaultComputerName,
        [string] $UserName = $defaultUserName,
        [string] $Password = $defaultPassword,
        [string] $VirtualHost = "/",
        [bool] $IgnoreConfirms = $false
    )

    $exchanges = Get-RabbitMQExchange
    $nondurableExchanges = New-Object System.Collections.ArrayList
    Foreach ($exchange in $exchanges) {
        if ($exchange.name -and -not ($exchange.durable) -and -not $exchange.name.Contains("Delinea-sr")) {
            $nondurableExchanges.Add($exchange) > $null
        }
    }
    if ($nondurableExchanges.Count -eq 0) {
        Write-Output "All the exchanges are durable."
        return
    }

    Write-Output "`r`nFound these exchanges as not durable:"
    Write-Output $nondurableExchanges | ForEach-Object { '{0}' -f $_.Name }

    $confirmation = ""
    if ($IgnoreConfirms -eq $false) {
        $confirmation = Read-Host "Are you Sure You Want To Proceed [y/n]"
    }
    if ($confirmation -eq 'y' -or $IgnoreConfirms -eq $true) {
        try {
            Foreach ($nondurableExchange in $nondurableExchanges) {
                Remove-RabbitMQExchange -Name $nondurableExchange.Name -VirtualHost $nondurableExchange.vhost -Confirm:$(-not $IgnoreConfirms)
                Add-RabbitMQExchange -Name $nondurableExchange.Name -Durable:$true -Type $nondurableExchange.type -AutoDelete:$nondurableExchange.auto_delete -
                Internal:$nondurableExchange.Internal -VirtualHost $nondurableExchange.vhost -Confirm:$(-not $IgnoreConfirms)
            }
            $connections = Get-RabbitMQConnection
            Foreach ($connection in $connections) {
                if ($connection.Name) {
                    {
                        Remove-RabbitMQConnection $connection.Name
                    }
                }
            }
        }
        catch {
            throw $_
        }
        Write-Output "Exchanges are now durable."
    }
    else {
        Write-Output "Not going to make the exchanges durable."
    }
}

MakeExistingExchangesDurable -IgnoreConfirms $true

```

RabbitMQ Naming Conventions for Queues

Introduction

This document addresses RabbitMQ naming conventions for its queues. These queues are useful for application troubleshooting and proactive application monitoring.

Secret Server is an asynchronous message-based system where operational instructions and data are passed back and forth between various components running in Web nodes or distributed Engines. A GUI interaction to perform an action, such as heartbeat, remote password change, or bulk operations publishes a message and then returns control back to the user. RabbitMQ is the message bus or broker that facilitates the message traffic.

Note: All Secret Server messages are encrypted on the bus, so you cannot peek into the message contents during transit.

Note: Messages have a lifetime, and consumers discard expired messages. Therefore, any accumulation or backup of messages in any queue is abnormal and indicative of an application problem.

Secret Server Roles

Secret Server divides its functionality by named and unnamed roles, and only named roles are configurable in a Web node via the GUI.

Table: Secret Server Roles Related to Message Queues

Background Worker	Named	Background work initiated by a task, schedule or UI interaction. Final action of the work might be done in the current Web node, another Web node or sent to a distributed Engine to complete.
Engine	Unnamed	Processes work related to but not limited to: Active Directory synchronization, discovery, heartbeat, and remote password change.
Engine Worker	Named	Processes the response sent back from an engine.
Session Recording Worker	Named	Background work dedicated to session recording processing.
UI	Unnamed	IIS/ASP.NET processing, inbound access controlled by a load balancer.
API	Unnamed	IIS/ASP.NET processing, inbound access controlled by a load balancer.

[Unexpected Link Text](#)

Queue Names

A queue name is divided into three major sections with a colon (:) delimiter between each section:

Section1:Section2:Section3

Section1

Section1 represents a RabbitMQ exchange name. There are three predetermined exchange names, two legacy predetermined exchange names, and then a variable number of exchanges determined by the number of Secret Server sites.

Table: RabbitMQ Exchange Names

Background Worker	Delinea-Secret Server	Predetermined	Web Node	
Engine	Site Name	Variable	Web Node or Distributed Engine	The out-of-the-box local site can be configured for either a Web node or a distributed engine. Any other site name is processed by a distributed engine.
Engine Worker	Delinea-Secret Server-engine-response	Predetermined	Web Node	
Session Recording Worker	thycotic-sessionrec	Predetermined	Web Node	
Session Recording Worker	thycotic-sr	Predetermined-Legacy	Web Node	Legacy: background work dedicated to session recording processing.
Session Recording Worker	thycotic-sr-agent-response	Predetermined-Legacy	Web Node	Legacy: processes data sent by an advanced session recoding agent.

[Unexpected Link Text](#)

Variable site exchanges: If the Secret Server site is called local, then local: will also be the exchange name. If the Site is called Mars, then Mars: will be the exchange name.

Section2

Section2 typically has a name which represents a functional area in Secret Server code that is a consumer of the message.

Section3

Section3 represents the message name.

Secret Server Roles and Queues

This section of the message associates roles with queues and breaks the down by product functionality. Functionality can span multiple roles, for example, discovery is done by background worker, engine and engine worker roles while event pipelines is only done by a background worker role.

Background Worker Role Queues

List of queues for background worker's functional areas:

Active Directory Synchronization

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.ActiveDirectorySynchronization.SynchronizationConsumer:Thycotic.ihawu.Business.Messages.Areas.ActiveDirectorySynchronization.Request.RunNowSynchronizationMessage-Secret Server:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ActiveDirectorySynchronization.SynchronizationConsumer:Thycotic.ihawu.Business.Messages.Areas.ActiveDirectorySynchronization.Request.SynchronizationMessage

Bulk Operation

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.BulkOperation.BulkOperationConsumer:Thycotic.ihawu.Business.Messages.Areas.BulkOperation.Request.BulkOperationMessage

ConnectWise Integration

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.ConnectWise.ConnectWiseConsumer:Thycotic.ihawu.Business.Messages.Areas.ConnectWise.Request.ConnectWiseMessage-Secret Server:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ConnectWise.ConnectWiseConsumer:Thycotic.ihawu.Business.Messages.Areas.ConnectWise.Request.RunNowConnectWiseMessage

Discovery

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.Discovery.ComputerScanConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.ComputerScanMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.ComputerScanConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowComputerScanMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.DiscoveryMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowDiscoveryMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryRuleApplierConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunDiscoveryRuleApplierMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.SecretComputerMatcherConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowSecretComputerMatcherMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.SecretComputerMatcherConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.SecretComputerMatcherMessage

Duo Integration

- thycotic-Secret Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Duo.DuoAuthConsumer:Thycotic.Messages.ihawu.Areas.Duo.DuoRequestMessage

Email Processing

- thycotic-Secret

Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Email.SendEmailConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.SystemSendEmailMessage

- thycotic-

ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Email.VerifySendEmailConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.VerifySendEmailRequest

Event Pipelines

- thycotic-Secret

Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.EventPipelineActivityConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.EventPipelineActivityEventMessage

- thycotic-

ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelinePolicyProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePoliciesProcessBlockingMessage

- thycotic-

ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelinePolicyProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePoliciesProcessMessage

- thycotic-

ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelineProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelineProcessBlockingMessageWithPolicies

- thycotic-

ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelineProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelineProcessMessageWithPolicies

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelinePolicyProcessEventConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePolicyProcessEventBlockingMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelinePolicyProcessEventConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePolicyProcessEventMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelineProcessScheduledEventConsumer:Thycotic.ihawu.Business.Messages.Areas.EventPipelines.ProcessPipelineScheduledEventMessage

Heartbeat and Remote Password Change

- thycotic-Secret

Server:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.CheckinExpiredCheckedoutSecretConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.CheckinExpiredCheckedoutSecretMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretLocalPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ExpiredSecretLocalPasswordChangeMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretLocalPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowExpiredSecretLocalPasswordChangeMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ExpiredSecretPasswordChangeMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowExpiredSecretPasswordChangeMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ProcessHeartbeatMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowProcessHeartbeatMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessLocalHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ProcessLocalHeartbeatMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessLocalHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowProcessLocalHeartbeatMessage

Import

- thycotic-Secret Server:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportConsumer:Thycotic.ihawu.Business.Messages.Import.SecretImportBulkMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportFileConsumer:Thycotic.ihawu.Business.Messages.Import.SecretImportFileMessage

Management: Backup, and Cleanup

- thycotic-Secret

Server:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackgroundWorkerTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.BackgroundWorkerTaskMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackupConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.BackupMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackupConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.RunNowBackupMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.GenerateSLMConsumer:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.GenerateSLMMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.ArchiveRecordedSessionsMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.DeleteRecordedSessionsMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.RunNowDeleteRecordedSessionsMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.TruncateRecords.TruncateRecordsConsumer:Thycotic.ihawu.Business.Messages.Areas.TruncateRecords.TruncateRecordsForAllConfigurationsMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.TruncateRecords.TruncateRecordsConsumer:Thycotic.ihawu.Business.Messages.Areas.TruncateRecords.TruncateRecordsForConfigurationMessage

- thycotic-

ss:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Distributed Engine Management

- thycotic-Secret

Server:Thycotic.ihawu.Business.Logic.Areas.DistributedEngine.EngineStatusUpdateConsumer:Thycotic.ihawu.Business.Messages.Areas.DistributedEngine.Request.EngineStatusUpdateMessage

- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.DistributedEngine.TruncateEngineLogConsumer:Thycotic.ihawu.Business.Messages.Areas.DistributedEngine.Request.TruncateEngineLogMessage

Password Generation

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.PasswordGeneration.GeneratePasswordConsumer:Thycotic.ihawu.Business.Messages.Areas.PasswordGeneration.Request.GeneratePasswordMessage

Reports

- thycotic-Secret Server:Thycotic.ihawu.Business.Logic.Areas.Report.EmailReportConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.EmailReportMessage-Secret Server:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Report.ScheduledReportConsumer:Thycotic.ihawu.Business.Messages.Areas.Reports.Request.ProcessReportsMessage

Run Now

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessDashboardJsonValidationConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessDashboardJsonValidationMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ProcessFieldEncryptionChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretKeyRotationConsumer:Thycotic.ihawu.Business.Logic.Areas.SecretKeyRotation.Messages.RunNowProcessSecretKeyRotationMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ProcessSecretPolicyChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowToggleHsmConsumer:Thycotic.ihawu.Business.Logic.Areas.SecretKeyRotation.Messages.RunNowToggleHsmMessage

Scheduled Tasks

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.DatabaseCleanupConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.DatabaseCleanupMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.EventQueueMonitorConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.EventQueueMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.ExpiringLicenseTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ExpiringLicenseTaskMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.ExpiringSecretTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ExpiringSecretTaskMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.PasswordRequirementConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.Password

ordRequirementMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.SqlReplicationConflictConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.SqlReplicationConflictMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.TruncateDatabaseCacheConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.TruncateDatabaseCacheMessage

Search

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.ProxySessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.ProxySessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowProxySessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RdpSessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RdpSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowRdpSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SecretItemHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowSecretItemHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SecretItemHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SecretItemHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SessionDataHashReIndexRequest

SSH Terminal

- thycotic-Secret
Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.SSHTerminal.TerminalCommandBackgroundConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.TerminalCommandMessage

Delinea Privilege Behavior Analytics Integration

- thycotic-Secret

Server:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaAppendMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAAppendMetadataSinkMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaCreateMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SACreateMetadataSinkMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaDirectiveConsumer:Thycotic.ihawu.Business.Messages.Areas.PBA.Request.DirectiveProcessMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaEventConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaEventUploadConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventUploadMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaMetadataUploadConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAMetadataUploadMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveAddConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveSendMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveCheckConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveCheckMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.HealthCheckConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHealthCheckMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.HeartbeatConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHeartbeatMessage

Delinea Privilege Manager Integration

- thycotic-Secret
Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsDatabaseUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsDatabaseUpdatedMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsEmailSettingsUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsEmailSettingsUpdatedMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsLicenseUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsLicenseUpdatedMessage

Delinea Telemetry

- thycotic-Secret
Server:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Telemetry.TelemetryConsumer:Thycotic.Messages.ihawu.Areas.Telemetry.Request.SendAnonymousTelemetryMessage

Delinea One Identify Provider Integration

- thycotic-Secret
Server:Thycotic.ihawu.Business.Logic.Areas.ThycoticOne.ThycoticOneSyncUserConsumer:Thycotic.ihawu.Business.Messages.Areas.ThycoticOne.Request.ThycoticOneScheduledSyncMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ThycoticOne.ThycoticOneSyncUserConsumer:Thycotic.ihawu.Business.Messages.Areas.ThycoticOne.Request.ThycoticOneSyncUserMessage

Engine Role Queues

List of queues for engines' functional areas.

Note: In the example listed below, the Secret Server site name is called "Gamma-Engines".

Active Directory Synchronization

- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ADSyncRequestConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ADSyncMessage
- Gamma-

Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.AllUsersByDomainQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.AllUsersByDomainQueryMessage

- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.GroupsAndMembersQueryMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.GroupsByDomainQueryMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.UsersByGroupsQueryMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ResolveDomainNameConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ResolveDomainDistinguishedNameMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ResolveDomainNameConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ResolveFullyQualifiedDomainNameMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.Authentication.AuthenticateByAdConsumer:Thycotic.Messages.DE.Engine.Areas.Authenticate.Request.AuthenticateByAdMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.AdSync.Areas.General.DomainCredentialTestConsumer:Thycotic.Messages.DE.Engine.Areas.General.Request.DomainCredentialTestMessage

Discovery

- Gamma-
Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.HostRangeConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanHostRangeMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.HostRangeConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.SpecificOutScanHostRangeMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.LocalAccountConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanLocalAccountMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.MachineConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanMachineMessage

Heartbeat, Remote Password Change, and Dependency

- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.BlockingChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.BlockingPasswordChangeMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.BlockingPrivilegeChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.BlockingPrivilegedPasswordChangeMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.Heartbeat.SecretHeartbeatConsumer:Thycotic.Messages.DE.Engine.Areas.Heartbeat.Request.SecretHeartbeatMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretBasicChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request

t.SecretBasicPasswordChangeMessage

- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretPrivilegeChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretPrivilegedPasswordChangeMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretRunDependenciesConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretChangeDependencyMessage Gamma-
Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.Verification.VerifyPasswordConsumer:Thycotic.Messages.DE.Engine.Areas.Verify.Request.VerifyPasswordMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.ServiceAccountManagement.Areas.Dependency.DependencyConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanDependencyMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.ServiceAccountManagement.Areas.Dependency.SecretTestDependencyConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretTestDependencyMessage

Management

- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Connectivity.PingConsumer:Thycotic.Messages.DE.Engine.Areas.Connectivity.Request.PingMessage
- Gamma-
Engines:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Proxy

- Gamma-
Engines:Thycotic.DE.Feature.SS.RdpProxy.AssignProxiedRdpSessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.AssignProxiedRdpSessionMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.SshProxy.Areas.Proxy.AssignProxiedSessionConsumer:Thycotic.Messages.DE.Engine.Areas.SSHProxy.Request.AssignProxiedSessionMessage

Scripting

- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.PowerShellScriptMessage
- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.SqlScriptMessage
- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.SshScriptMessage

Syslog Integration

- Gamma-Engines:Thycotic.DE.Feature.SS.AdvancedAuditing.Areas.SIEM.SysLogConsumer:Thycotic.Messages.DE.Engine.Areas.SIEM.Request.SysLogMessage

Delinea Privilege Behavior Analytics Integration

- Gamma-Engines:Thycotic.DE.Feature.SS.Pba.Areas.Event.PbaEventConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.Pba.Areas.Metadata.PbaAppendMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAAppendMetadataSinkMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.Pba.Areas.Metadata.PbaCreateMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SACreateMetadataSinkMessage
- Gamma-
Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveAddConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveSendMessage

- Gamma-
Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveCheckConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveCheckMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.HealthCheckConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHealthCheckMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.HeartbeatConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHeartbeatMessage

Ticketing System Integration

- Gamma-
Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingAddCommentConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingAddCommentBasicRequest
- Gamma-
Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingAddCommentConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingAddCommentMessage
- Gamma-
Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingGetStatusConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingGetStatusMessage

Engine Worker Role Queues

List of queues for engine worker's functional areas:

Active Directory Synchronization

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.ActiveDirectory.ActiveDirectorySynchronizationConsumer:Thycotic.Messages.DE.Server.Areas.ActiveDirectory.Request.ADSyncMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.ActiveDirectory.AllUsersByDomainQueryConsumer:Thycotic.Messages.DE.Server.Areas.ActiveDirectory.Request.AllUsersByDomainQueryMessage

Discovery

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanDependencyConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanDependencyMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanHostRangeResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanHostRangeMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanLocalAccountConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanLocalAccountMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanMachineResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanMachineMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.SpecificOuScanHostRangeResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.SpecificOuScanHostRangeMessage

RDP Proxy, SSH Proxy, and SSH Terminal

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.RDPProxy.AppendKeystrokeDataConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.AppendKe

ystrokeDataMessage

- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.AppendSessionDataConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.AppendSessionDataMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.CloseSecretSessionConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.EndSessionDataMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.EndRdpProxySessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.EndRdpProxySessionMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.GetStatusUpdatesRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.GetStatusUpdatesMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateRDPProxiedSessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.InitiateProxiedRdpSessionMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateSSHProxiedSessionConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.InitiateProxiedSessionMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateSshSessionDataCaptureSinkConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.InitiateProxiedSessionDataCaptureSinkMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.RdpProxySessionStatusesConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.GetRdpProxySessionStatusesMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.UpdateSessionsRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.UpdateSessionsMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.UpdateUserPasswordRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.UpdateUserPasswordMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHTerminal.TerminalCommandEngineConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.TerminalCommandMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.UserSession.CloseUserSessionConsumer:Thycotic.Messages.DE.Server.Areas.UserSession.CloseUserSessionMessage

Syslog Integration

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.SEIM.SysLogResultResponseConsumer:Thycotic.Messages.DE.Server.Areas.SEIM.Request.SysLogResultMessage

Heartbeat, Remote Password Change, and Dependency

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Dependency.DependencyChangeConsumer:Thycotic.Messages.DE.Server.Areas.Dependency.Request.DependencyChangeMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Heartbeat.SecretHeartbeatConsumer:Thycotic.Messages.DE.Server.Areas.Heartbeat.Request.SecretHeartbeatMessage

- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.PasswordChanging.RemotePasswordChangeResponseStoreConsumer:Thycotic.Messages.DE.Server.Areas.PasswordChanging.Request.RemotePasswordChangeMessage

Delinea Privilege Behavior Analytics Integration

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.PasswordChanging.PbaDisableConsumer:Thycotic.Messages.DE.Server.Areas.PBA.PbaDisableMessage

Distributed Engine Management

- thycotic-Secret Server-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Connectivity.PingConsumer:Thycotic.Messages.DE.Server.Areas.Connectivity.Request.PingMessage
- thycotic-ss-engine-
response:Thycotic.ihawu.EngineWorker.Logic.Areas.Maintenance.LogConsumer:Thycotic.Messages.DE.Server.Areas.Maintenance.Request.EngineLogMessage
- thycotic-ss-engine-
response:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Session Recording Worker

List of queues for session recording worker's functional areas:

Post Recording

- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostMetadataConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessUploadedMetadataMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.RecordedSessionChunkMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessBusStreamedSessionMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessUploadedSessionMessage

Video Conversion

- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ConvertAllVideosMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ConvertVideoMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.DeleteOldCompletedImagesMessage
- thycotic-
sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.RunNowConvertVideoMessage
- thycotic-

sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.RunNowSetStatusForTimedOutSessionsMessage

- thycotic-

sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.SetStatusForTimedOutSessionsMessage

Post Recording (Legacy)

- thycotic-sr-agent-

response:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostMetadataConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.PostMetadataMessage

- thycotic-sr-agent-

response:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.PostRecordedSessionMessage

Management

- thycotic-

sessionrec:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Security

Please see [Distributed Engine Hardening](#).

You can change how Secret Server processes messages by navigating to **Admin > See All** and selecting **Networking**.

Messages are generated and placed on the internal site connector, or backbone bus, every time a background operation is triggered whether by a schedule or on-demand.

The internal site connector receives and processes messages as a result of numerous actions:

- Bulk Operations
- Generate Password
- Secret Import (CSV and XML)
- Run Heartbeat Now
- Run Heartbeat (Scheduled)
- Run Password Change Now
- Run Password Change (Scheduled)
- Run Discovery Now
- Run Discovery (Scheduled)
- Run AD Sync Now
- Run AD Sync (Scheduled)
- Elements of Session Recording

The internal site connector, using the internal hosted bus, is adequate for bulk operations, heartbeat, discovery, and the like, but some Secret Server features, such as a clustered Web server node configuration or session recording, require a scalable messaging solution to boost processing performance. Our choice is [RabbitMQ](#), which is an intermediary messaging broker that can handle large-scale message processing.

The following is a typical internal hosted bus operation (for a bulk operation):

1. A Secret Server user triggers the a bulk operation.
2. A message is formed and sent over a TCP connection to the internal hosted bus.
3. Secret Server (on the same machine) receives the message.
4. Secret Server (on the same machine) processes the message.

We continually improve the internal hosted bus but still recommend RabbitMQ for a scalable performance boost. See [Installing RabbitMQ](#) for more information.

Overview

This article lists ports typically used in Secret Server. Please note the following:

- The RPC Dynamic Port ranges are a range of ports utilized by Microsoft's Remote Procedure Call (RPC) functionality. This port range varies by operating system. For Windows Server 2008 or greater, this port range is 49152 to 65535 and this entire port range must be open for RPC technology to work. The RPC range is needed to perform Remote Password Changing since Secret Server will need to connect to the computer using DCOM protocol.
- The range can vary separately for Exchange servers. For more information about changing the RPC port range, see the related Microsoft's Knowledge Base article on how to configure RPC dynamic port allocation to work with firewalls.
- To see your ipv4 dynamic range on a given machine, type `netsh int ipv4 show dynamicport top` in the command line.
- To specify a specific port on your environment that Secret Server will communicate to, see the related article on enabling WMI ports on Windows client machines

Port Listing

Table: Active Directory Sync Ports

Kerberos	TCP/88, UDP/88
LDAP	TCP/389, UDP/389
LDAPS	TCP/636, UDP/636
SMB/Microsoft-DS	TCP445, UDP/445

[Unexpected Link Text](#)

Note: For LDAPS to work the LDAP port (389) must also be open.

Table: Discovery Ports

RPC Dynamic Port Range	TCP/49152-65535, UDP/49152-65535
SMB/Microsoft-DS	TCP/445, UDP/445
RPC Endpoint Mapper	TCP/135
SSH	TCP/22

[Unexpected Link Text](#)

Table: Remote Password Changing Ports

--	--

RPC Dynamic Port Range	TCP/49152-65535, UDP/49152-65535
SSH	TCP/22
Telnet	TCP/23
Microsoft SQL	TCP/1433, UDP/1434
SMB/Microsoft-DS	TCP/445, UDP/445
LDAP	TCP/389, UDP/389
LDAPS	TCP/636, UDP/636
Sybase	TCP/2638, TCP/5000
Oracle Listener	TCP/1521
Kerberos Password Change	TCP/464, UDP/464
Windows Privileged Account (WinNT ADSI Service Provider)	TCP/139

[Unexpected Link Text](#)

Table: Web Server Incoming Ports

HTTP	TCP/80
HTTPS	TCP/443

[Unexpected Link Text](#)

Table: Database Server Incoming Ports

SQL Connection	TCP/1433, UDP/1434
----------------	--------------------

[Unexpected Link Text](#)

Table: Email Ports

SMTP	TCP/25
------	--------

[Unexpected Link Text](#)

Table: RADIUS Server Ports

RADIUS Authentication	UDP/1812
-----------------------	----------

[Unexpected Link Text](#)

Table: Syslog Ports

Syslog	TCP/514, UDP/514
--------	------------------

[Unexpected Link Text](#)

Table: Internal Site Connector Ports

RabbitMQ	TCP/5672 (non-SSL), TCP/5671 (SSL)
MemoryMQ	TCP/8672 (non-SSL), TCP/8671 (SSL)

[Unexpected Link Text](#)

Table: RabbitMQ Clustering Ports

EPMD	TCP/4369
Inter-node Communication	TCP/25672
Inter-node Communication	TCP/44002

[Unexpected Link Text](#)

Related Articles and Resources

- [How to configure RPC dynamic port allocation to work with firewalls](#) (KBA)

Note: RDP Proxy requires .NET 4.7.2 or later.

Overview

The RDP Proxying feature allows RDP connections, established using a launcher, to be routed through Secret Server. You can set it up one of two ways:

- **Recommended method:** The launcher connects to the newer RDP proxy with temporary credentials, and the RDP proxy connects to the remote server using the protected credentials from the secret. This method is preferred because it prevents the secret credentials from reaching the client machine. For this method, you simply configure the RDP proxy.
- **Alternative method:** The launcher uses an SSH proxy to tunnel a local RDP connection to a remote server. This method does not protect the credential from reaching the client machine. For this method you configure the SSH proxy and enable SSH tunneling.

Note: We provide the alternate method to support legacy installations and troubleshooting (it can potentially be more stable when the RDP proxy does not work).

These two approaches to RDP proxying are not compatible—you may use one or the other but not both. We performance tested both methods. Either can support 100 concurrent connections.

Recommended Method

How It Works

1. The user clicks the RDP launcher in Secret Server.
2. The launcher executes on the client's machine.
3. The launcher establishes a connection to the RDP Proxy using credentials generated for the session. These credentials are short lived and can only be used once.
4. Once the launcher has successfully authenticated with the RDP proxy, the RDP proxy looks up the credentials and target hostname to connect to.

Note: The secret credentials *do not* get served to the client machine in this flow, which improves credential security.

5. The RDP proxy connects to the desired remote host with the secret credentials.
6. The RDP session is established.
7. RDP traffic is sent back and forth over the RDP proxy, session keystrokes are monitored if session recording is enabled.

Configuration

1. Navigate to the **Admin > Proxying** page.
2. Click the **RDP Proxy** tab.

Admin > Proxying

SSH Proxy **RDP Proxy** Endpoints Proxy Audit

The RDP proxy is currently running on 0 site(s) and 0 engine(s). Please see the Endpoints tab for more details.

RDP Proxy Settings

- Secret Server can proxy Remote Desktop connections through a Distributed Engine to a Windows end point.
- You can configure your SSH server to only accept connections from the proxy, thus forcing all connections through Secret Server.
- With Session Recording enabled, you will be able to record keystrokes sent during RDP proxy sessions
- To enable RDP proxying, install at least one Distributed Engine and specify the Public Host and Bind IP address.
- Secrets with launchers which support proxying can have it enabled or disabled on an individual basis from their security tab. This setting can also be set via Secret Policy.

Enable RDP Proxy	Yes	Edit
RDP Proxy Port	3390	Edit
Validate Remote Certificates	Yes	Edit
Allow AD Site Selection	Yes	Edit
Proxy New Secrets By Default	Yes	Edit
Days to Keep Operational Logs	30	Edit
RDP Server Certificate	thyp2.pfx	Edit Generate Self-Signed

- If necessary, enable the RDP proxy.
- Click the **Endpoints** tab to ensure that your server nodes, sites, and engines have RDP Proxy enabled.
- Proxied RDP secrets now launch into the RDP proxy using short-lived credentials, protecting the secret credentials from the client machine.

Configuration Settings

The RDP proxy configuration settings for the recommended method:

- Enable RDP Proxy:** This setting determines whether or not the RDP proxy is enabled
- RDP Proxy Port:** This setting is the port that the RDP proxy runs on (defaulting to 3390). You usually cannot set this to 3389 as that port is already occupied by default by the Windows operating system.
- Validate Remote Certificates:** Delinea recommends that you operate in an environment where RDP server certificates are created by a controlled CA and are trusted by machines in the domain. If that is not possible, you can disable remote certificate validation to allow connection to machines that do not serve trusted certificates.
- Allow AD site selection:** This setting allows you to select any configured sites when using the RDP launcher on an Active Directory secret. This allows a secret credential to access machines that may exist in different network boundaries.
- Proxy New Secrets By Default:** This setting determines if SSH and RDP secrets are created with "Proxy Enabled" set by default. This setting is shared with the SSH proxy configuration.
- Days To Keep Operational Logs:** This setting determines how long, in days, the operational logs for the RDP proxy are kept.
- RDP Server Certificate:** This setting is the certificate that is served to the clients who connect to the RDP proxy. You can generate a certificate for a given DNS name, or you can upload your own.

Alternative Method

Note: This approach is not recommended as it exposes the secret credentials to the client machine.

How It Works

- The user clicks the RDP launcher in Secret Server.

2. The launcher executes on the client's machine.
3. The launcher establishes a connection to the SSH proxy to begin port forwarding.
4. The launcher authenticates with the SSH Proxy.
5. The launcher opens a socket.
6. The launcher listens for a connection on an available ephemeral port (the forwarding port) on the client's machine.
7. RDP launches on the client machine using the secret credentials and connects locally to the forwarding port.
8. All RDP traffic for this session is routed through the SSH tunnel to Secret Server, then forwarded to the target machine.
9. The RDP session is established.

Configuration

1. Navigate to the **Admin > Proxying** page.

The screenshot shows the 'Admin > Proxying' page. At the top, there is a navigation bar with 'SSH Proxy', 'RDP Proxy', 'Endpoints', and 'Proxy Audit' tabs. Below the navigation bar, a message states: 'The SSH proxy is currently running on 0 site(s) and 0 engine(s). Please see the Endpoints tab for more details.' The main content area is titled 'SSH Proxy Settings' and contains a table of configuration options.

SSH Proxy Settings	Value	Action
Secret Server can proxy connections through a Distributed Engine to an SSH server end point.	Enable SSH Proxy: Yes	Edit
Remote Desktop Sessions can also be proxied if enable SSH tunneling is set.	SSH Proxy Port: 22	Edit
You can configure your SSH server to only accept connections from the proxy, thus forcing all connections through Secret Server.	Enable SSH Tunneling: No	Edit
All proxied traffic can be recorded for security and auditing purposes.	Proxy New Secrets By Default: Yes	Edit
To enable SSH proxying, install at least one Distributed Engine and specify the Public Host and Bind IP address.	Enable SSH Proxy Inactivity Timeout: No	Edit
Secrets with launchers which support proxying can have it enabled or disabled on an individual basis from their security tab. This setting can also be set via Secret Policy.	SSH Proxy Banner: Welcome to the Secret Server SSH Proxy	Edit
	SSH Proxy Host Fingerprint: SHA1 - 8e:80:65:1fd8:a7:84:c1:33:f2:80:7f:dc:84:48:d0:41:29:06:c7 MD5 - 99:0b:df:99:15:e4:a1:7f:7c:70:ec:12:87:16:8e:fb	Edit Generate
	Days to Keep Operational Logs: 50	Edit

2. Enable the **Enable SSH Tunneling** option.
3. Click the **Endpoints** tab to ensure that your server nodes, sites, and engines are properly configured.
4. Proxied RDP secrets now launch into the SSH proxy using local port forwarding.

Known Issues

"Could not load file or assembly..." Error

Error appears in Secret Server.log or DE.log. Install the most recent version of the .NET Framework to correct it.

RDP Proxy Does Not Work with FIPS Validation

RDP proxy does not work on machines the have the FIPS validation security policy active. No fix is currently available.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This document is a guide to Delinea's Secret Server clusters for administrators and advanced users. Secret Server can run with multiple front-end Web servers. For a critical instance, clustering offers a redundant system to limit potential down time from a single point of failure. Clustering also allows users to load balance for better performance.

Overview

Clustering and Background Thread Changes in 10.7.

There are two major architectural changes in Secret Server 10.7:

Note: The first change is obvious in the Secret Server user interface, and the second is hidden but very important to those supporting Secret Server.

- **Primary Node:** We eliminated "primary nodes." Previously, some important background operations, such as password changing and heartbeat, would only run from the primary node. Now they run from all nodes. Given that, there is no longer a "Make Primary" button, and the ValidPrimaryNode setting no longer applies.
- **Background Operations:** There are no longer background threads for scheduled operations. Instead, operations are scheduled by Quartz.

Clustering Overview

With Secret Server clustering, you can easily scale Secret Server for redundancy and performance. Basic Secret Server clustering is simple—you install Secret Server and then copy the installation to another machine. Secret Server clustering has four core concepts or components:

Nodes

Each machine with Secret Server installed on it, pointing to the same database, is a *node*. All nodes respond to Web requests and thus are Web servers.

Backbone Bus

The backbone bus internally handles all communication between the roles. In a clustered environment, the backbone bus should always be an installed RabbitMq messaging queue. This allows every node in the cluster to help with the workload. If the backbone bus is set to "internal," then each node is using its own internal backbone bus.

Engine Response Bus

The engine response bus facilitates communication from Secret Server to distributed engines and back.

Worker Roles

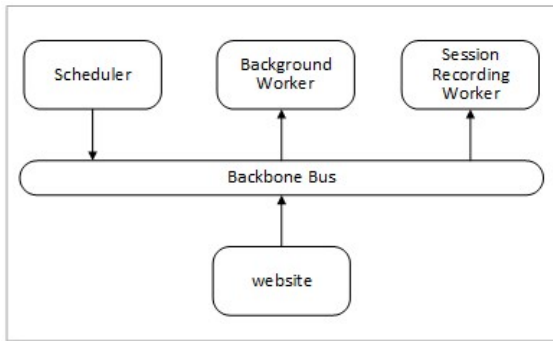
Each node can optionally run one or more worker roles: background Worker, engine worker, and session recording worker. Though they may run on the same machine, the roles do not directly communicate with each other.

Each node that is set to run the background worker role automatically runs the scheduler role as well. The scheduler role is responsible for running the vast majority of Secret Server background operations. It uses Quartz to run "trigger jobs" that send a message on the backbone bus for each scheduled operation. One or more background worker roles then processes those messages.

Note: See the article [Troubleshooting Quartz Trigger Jobs](#) for more information about Quartz.

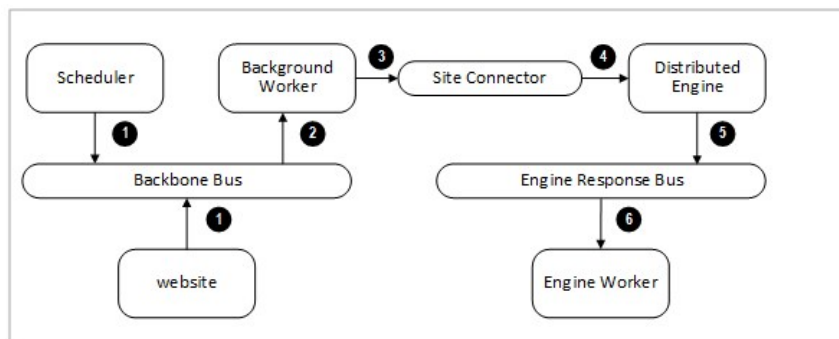
Component Communication

Figure: Secret Server Internal Cluster-Component Communication



Messages are placed on the backbone bus by the Scheduler role and the website. Messages are retrieved from the backbone bus.

Figure: Secret Server Distributed Engine Communication





1. Manual or scheduled operation.
2. Background worker processes a message.
3. Outbound messages (password changes, heartbeats, and others) are placed on the site connector.
4. Distributed engine performs the operation.
5. Engine worker processes the response.

Server Node Configurations

The work an individual node handles depends entirely on which boxes are checked on the Server Nodes page (in edit mode):

Server Nodes

MACHINE NAME (ID)	BINARY VERSION	DATABASE	ERROR MESSAGE	LAST CONNECTED	IN CLUSTER	BACKGROUND WORKER	ENGINE WORKER	SESSION RECORDING WORKER	MAINTENANCE MODE	
QA-CUST-01 (1) (Current Node)	10.7.000000	SS_Playground		8/13/2019 5:35:08 PM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 

Enable Clustering No

- **In Cluster** is a toggle that turns a server node on or off. If enabled this node can process Web requests, and (if configured) will run the background, engine, and session recording roles. If disabled, the node is just a backup—it cannot run any roles, and trying to access the website on the node will redirect to the server nodes page.
- **Background Worker** is a toggle for all background operations, such as password changing, heartbeat, and discovery. When it is set to false, only the bulk operations, password generation, email, and secret import operations run on the node. See the list of background operations below.
- The **Background Worker**, **Engine Worker**, and **Session Recording Worker** check boxes enable the corresponding roles for that node.
- **Engine Worker** enables or disables the engine worker role, which processes responses from distributed engines.
- **Session Recording Worker** enables or disables the session recording role, which encodes session videos.
- **Maintenance Mode** enables or disables a read-only mode where the node cannot change secrets or related data.

Scheduled Background Operations

The current scheduled background operations operations in Secret Server are:

- ActiveDirectorySynchronizationMonitor
- BackgroundWorkerTaskTriggerJob
- BackupMonitor
- Bulk Operations When triggered by user
- CheckOutMonitor
- ComputerScanMonitor
- ConnectWiseMonitor
- DatabaseCleanupTriggerJob
- DiscoveryMonitor
- EventQueueMonitor
- ExpiredSecretPasswordChangeTriggerJob
- ExpiringLicenseTaskTriggerJob
- ExpiringSecretTaskTriggerJob
- HeartbeatMonitor
- Local Heartbeat Trigger Job
- Local Password Change Trigger Job
- NodeClusteringMonitor
- NodeTaskTriggerJob
- PasswordRequirementTriggerJob
- PbaDirectiveTriggerJob
- PbaMetadataUploadTriggerJob
- PrimaryNodeTaskMonitor
- Process Field Encryption Changes Task

- ProcessDashboardJsonValidationTask
- ProcessSecretPolicyChangesMessage
- ScheduledReportMonitor
- SecretComputerMatcherMonitor
- SecretItemHashMonitor
- SqlReplicationConflictMonitor
- TelemetryTriggerJob
- ThycoticOneSyncUserTriggerJob
- TruncateDatabaseCacheTriggerJob
- TruncateEngineLogTriggerJob
- VideoConversionTriggerJob

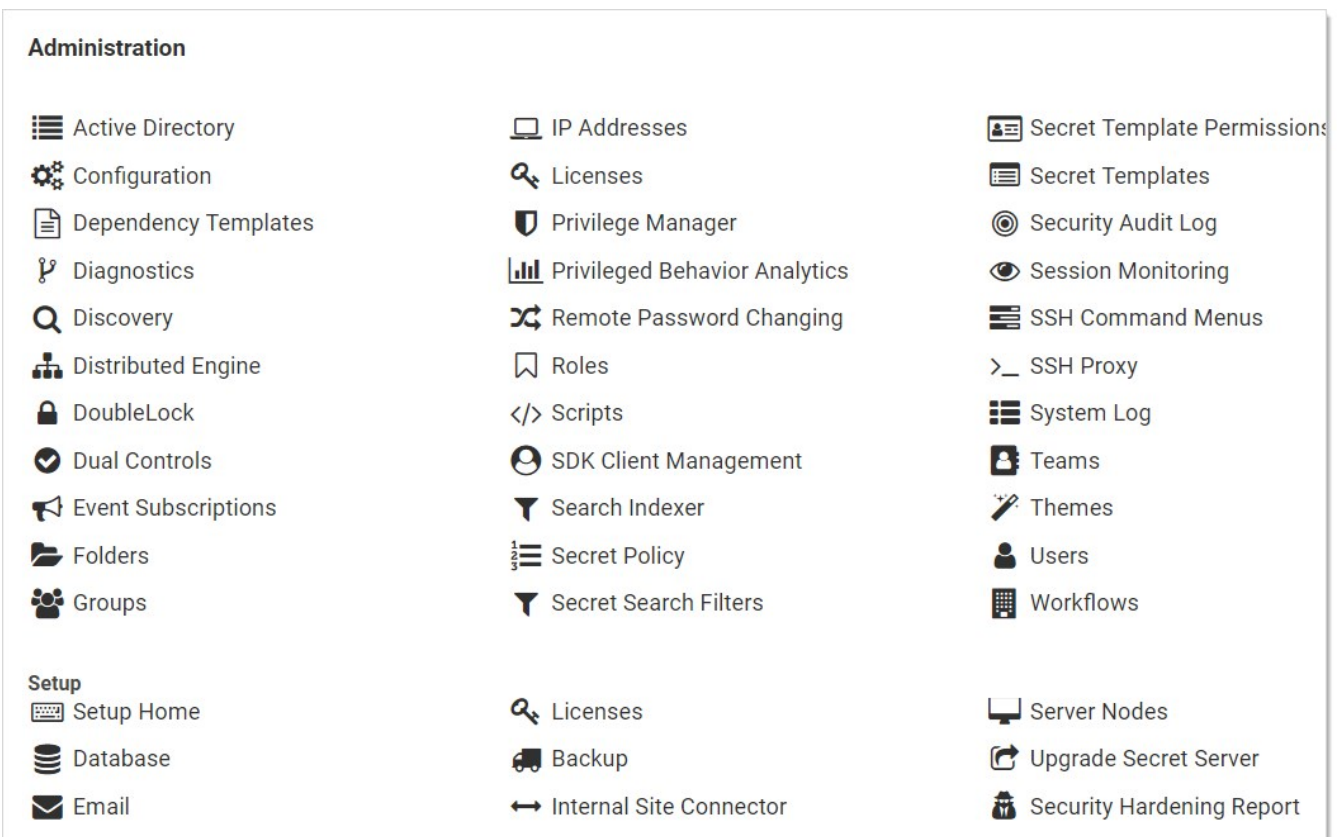
To see the current state of these jobs, such as the last time they ran and how long until they run again, go to **Admin > Diagnostics**.

Procedures

Markdig.Syntax.Inlines.EmphasisInLine

Note: Clustering requires a Secret Server Premium add-on or Enterprise Plus edition license.

1. Have Secret Server upgraded or installed and running on a server.
2. Enable clustering on the node:
 1. In Secret Server, click **Admin > See All**. The Administration page appears:



2. Click the **Server Nodes** button in the **Setup** section. The Server Nodes page appears:

Server Nodes

MACHINE NAME (ID)	BINARY VERSION	DATABASE	ERROR MESSAGE	LAST CONNECTED	IN CLUSTER	BACKGROUND WORKER	ENGINE WORKER	SESSION RECORDING WORKER	MAINTENANCE MODE
QA-CUST-01 (1) (Current Node)	10.7.000000	SS_Playground		8/13/2019 8:05:19 PM	Yes	Enabled	Enabled	Enabled	Disabled

Enable Clustering No

← Back
✓ Enable Clustering
SQL Server Replication

[KB Article: Server Nodes, Clustering and Worker Roles](#)

3. Click the **Enable Clustering** button.

3. Copy the entire Secret Server application folder (typically `c:\inetpub\wwwroot\SecretServer`) from the existing node to the secondary node.

4. Follow the steps in the Installation Guide for setting up the application pool and virtual directory in IIS.

Note: If you use DPAPI encryption for your `encryption.config` file, you need to transfer the non-DPAPI-encrypted version of the file to the secondary node. You can turn on DPAPI encryption from that server node locally after Secret Server is running. This setting can be found at **ADMIN > Configuration** on the **Security** tab.

5. If running Secret Server 8.9.300000 or later, ensure that both servers are using the same date and time.

6. Once the secondary server is running, navigate to its Secret Server using a Web browser.

7. Reset the database connection, following the instructions at [Changing SQL Server Connection Parameters](#).

8. Activate licenses for the new node. You can do this on either server once the database connection is established on the secondary node.

9. Configure your load balancer for the two sites to have "sticky sessions" to prevent a user from bouncing between server on each request.

10. Configure the worker roles for the cluster:

- Each server node can optionally run the background worker, engine worker, and session recording worker roles.
- At least one instance of **each** type of those roles must be active in the cluster for the clustered Secret Server application to function.
- You may run more than one instance of each role as desired to improve the performance of the clustered Secret Server application.

Note: For more information on what the various roles do, please see the [Worker Roles](#) section.

Upgrading Secret Server in a Clustered Environment

Overview

Secret Server has a built-in Web installer. That installer is a series of pages inside Secret Server for downloading and updating Secret Server. Secret Server is accessible by users for most of the upgrade process. You can stop outside access to the site if you want to prevent users from making changes during the upgrade. Preventing user access will make restoring the database and site backups simpler if you decide to roll back the upgrade immediately afterward.

Warning: Before upgrading, **backup your Secret Server folder and database**. See [Upgrading Secret Server - Single Instance and Web Clustering](#) for important steps for ensuring your data is backed up.

Important: Upgrading to Secret Server version 8.9.000000+ requires Windows Server 2008 R2 or greater.

Important: If upgrading to Secret Server version 8.5.000000+, there are changes in the required .NET Framework that may require additional steps in the upgrade process. For more information, see [Secret Server Moving to .NET Framework 4.5.1](#).

Important: Upgrading to Secret Server 10.0.000000 and above requires configuring integrated pipeline mode on the Secret Server application pool. Please see [Configuring IIS for Installing or Upgrading to Secret Server 10](#) for details.

Important: If using Integrated Windows authentication you will also need to update IIS authentication settings as detailed in [Setting Up Integrated Windows Authentication in Secret Server 10.0+](#). If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000+.

Note: You do **not** need to download the Secret Server installer to perform an upgrade.

Procedure

1. Before you start:
 - Ensure that you have account credentials information and access for the server hosting Secret Server and the SQL Server instance hosting your Secret Server database.
 - Have a recent backup of the application files and database available.
 - Stop the application pools on all of the servers except the one that you have chosen to upgrade.
2. Choose one Secret Server server to upgrade
3. Perform a backup of that server.
4. Stop the Web servers of all other nodes.
5. Perform the upgrade using the same procedure as a single instance.

Note: If applicable, see [Upgrading Secret Server without Outbound Access](#).
6. Once Secret Server is upgraded and working, copy the Web application folder (without the database.config or encryption.config files) to all other servers.

Warning: Never overwrite or delete the encryption.config file on a Secret Server server.

Note: Both encryption.config and database.config are automatically propagated to the new servers from the original. If you need to copy those files because of database configuration changes and are using DPAPI, disable DPAPI encryption in Secret Server by going to **Admin > Configuration** on the **Security tab**, and clicking **Decrypt Key to not use DPAPI** before copying those files to secondary servers.

Note: EFS encryption is tied to the user account running the Secret Server application pool, so it is not machine-specific. Thus, it is not necessary to copy EFS encrypted files between Secret Server instances, but it is allowed.

7. If Delinea management server (TMS) is installed and clustered, copy the TMS directory to the secondary servers as well. The TMS directory is included by default for new installs of Secret Server 10.2+. TMS is used by advanced session recording and Privilege Manager. If the TMS folder and site does not exist in IIS, then no additional actions are needed.
8. Start the secondary servers to confirm they still work.

Upgrading Database Mirroring

1. If there is more than one Web server running Secret Server, ensure all instances are pointing to their primary database.
2. Select one server to perform the upgrade on, stop all other web servers.
3. Perform the upgrade on the single instance.
4. Once upgraded and working, copy the Web application folder to all other Web servers.
5. Start all other Web servers and confirm they work
6. Ensure all instances are properly activated
7. Ensure that the primary database changes have been replicated to the mirror database.
8. If one of the servers was pointing originally to the secondary database, adjust it to point there again.

Upgrading Disaster Recovery Installations

1. Perform the upgrade on the production instance.
2. Backup the production instance.
3. Copy the database backup to the remote DR instance and restore the database.
4. Once the database is upgraded and working, copy the web application folder (but not the database.config or encryption.config files) to the remote DR instance, overwriting the existing files.
5. Restart IIS or recycle the application pool running Secret Server on the remote DR instance.
6. Confirm that the remote DR instance is working correctly.

Load Balancing Secret Server Clusters

In a clustered Secret Server environment set up behind a load balancer, the accessible outside URL may be something other than the server name.

Important: Customers often configure the service monitor on the load balancer to hit `login.aspx`. This can cause performance issues and memory leaks. We strongly suggest hitting `healthcheck.aspx` instead.

Custom URL Configuration

In Secret Server 8.5 and later, the Custom URL setting can be configured to ensure that links and resources are resolved correctly and are not based upon the server name. The Custom URL sets a definitive URL for Secret Server. Without it, features in Secret Server that need to build a link back to the server must construct the link using the host value on the request, which is susceptible to manipulation.

1. Navigate to **Admin > Configuration**.
2. On the **General** tab, click the **Edit** button.

3. Go to the **Application Settings** section.
4. Click to select the **Custom URL** check box.
5. Type the desired URL in the **Secret Server Custom URL** text box.

SSL Recommendations

For the best security, we recommend placing the SSL certificate on each of the Web servers. This ensures the traffic leaving the server is encrypted by SSL. Optionally, the load balancer would need the certificates as well for adding the client's IP address.

If the connection between the load balancer and the server is isolated in a security zone, you could leave HTTP between the load balancer and the server and have the SSL on the load balancer.

Configuring Client's IP Address (X-Forwarded-For)

Routing traffic through a load balancer will cause Secret Server to audit the IP address of the load balancer instead of the end user. To avoid this:

First, configure the load balancer to pass along the client's IP address in the header.

Then add the `appSettings` key `IpAddressHeader` with the value of the name of the header field containing the client's IP address. This setting must be added to **all** Secret Server Web servers. Depending on your Secret Server version, do this in one of two ways:

For Secret Server prior to 10.5.000000:

In the `web-appSetting.config` file in your Secret Server directory, add the following key:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
  <add key="IpAddressHeader" value="X-Forwarded-For" />
</appSettings>
```

For Secret Server 10.5.000000 and later:

1. Go to `https://<SecretServerAddress>/ConfigurationAdvanced.aspx`.
2. Scroll to the bottom and click **Edit**.
3. Locate the **IP Address Header** text box, type `X-Forwarded-For`.
4. Click the **Save** button.

Note: The SSL certificate needs to exist on the load balancer and the Web server to ensure it has access to add the client IP address header.

Clustering Errors

The following error may arise when setting up or operating Secret Server clustering:

- Server dates do not match: If the Web server dates do not match, the audit records could be bad. The fix is to set the servers to the same time.

Note: This only applies to Secret Server before version 8.9.300000.

- Secret Server version does not match: If some of the cluster nodes have been upgraded and others have not, their versions will conflict, producing this error. Nodes which have the wrong (older) version will not function correctly. To fix this issue, ensure that all the nodes in your cluster are upgraded. For nodes that are having this issue, you can copy the application folder (minus the `database.config` file)

to replace the server files with the correct version.

HTTP/2 is supported in IIS 10. HTTP/2 is handled within IIS, so this is primarily a Microsoft question in regards to compatibility. Please see [HTTP/2 on IIS](#). At the end of this article, it clarifies when HTTP/2 is not supported

Secret Server does support Windows Integrated Authentication where a user's windows session is passed through for authentication to Secret Server. That is, there is no log on page for Secret Server. The majority of our customers are (and the default configuration for Secret Server is) using forms-based authentication with a log on page. Only the latter is HTTP/2 compliant.

HTTP/2 is only compatible with HTTPS protocol. Secret Server can also be configured to operate only on HTTPS (Admin > Configuration > Security > Force HTTPS/SSL), which we strongly recommend.

HTTP Strict Transport Security (HSTS) is an additional security layer for HTTPS that ensures anybody accessing a given Web site or entity is forced to use HTTPS and not HTTP *prior* to making any HTTP requests, eliminating man-in-the-middle attacks. HSTS is an IETF Internet Standards Track protocol as specified in RFC 6797.

When the Force HTTPS/SSL option is enabled in Secret Server, the **Enable HSTS** check box is displayed. After the option is turned on, you can click the **Advanced** link to specify the **Maximum Age** in seconds, which is how long the policy is in affect before re-evaluating. The default value is 31536000 seconds or one year. We recommend that you set the value as high as possible, up to a year if the site should never be accessed without SSL. Even after HSTS is disabled, your browser automatically redirects to over SSL for the duration of the configured maximum age.

Note: We recommend using the IISReset command-line utility or restarting IIS in IIS manager after enabling the setting for the setting to take effect.

This feature is available in Secret Server version 8.6.000009 and higher and Password Reset Server version 4.0.000000 and higher.

Note: To see which browsers support HSTS, please refer to the [Strict Transport Security](#) page on the Can I Use website.

This topic details SSH cipher suite encryption, key exchange, and MAC algorithms.

Important: Enable FIPS in Secret Server to ensure all algorithms are FIPS-certified.

Note: SecureBlackbox enables all available SSH encryption, key exchange, and MAC algorithms by default.

This information applies to the following as of Secret Server 11.2.X (June 2022).

- SSH Server: Used by SSH proxy
- SSH Client: Used by SSH proxy, RPC, heartbeat, discovery, and script runners.
- Local port forwarding: Used by SSH proxy Jumpbox routes)

Secret Server with FIPS Enabled

Default Encryption Algorithms,

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

aes256-gcm@openssh.com	SSH_EA_AES256_GCM_OPENSSSH	2147483646
aes128-gcm@openssh.com	SSH_EA_AES128_GCM_OPENSSSH	2147483645
aes256-gcm	SSH_EA_AES256_GCM	2147483644
aes128-gcm	SSH_EA_AES128_GCM	2147483643
aes256-ctr	SSH_EA_AES256_CTR	2147483642
aes192-ctr	SSH_EA_AES192_CTR	2147483641
aes128-ctr	SSH_EA_AES128_CTR	2147483640
aes256-cbc	SSH_EA_AES256	2147483639
aes128-cbc	SSH_EA_AES128	2147483638
3des-cbc	SSH_EA_3DES	2147483637

Default Key Exchange Algorithms,

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

curve25519-sha256@libssh.org	SSH_KEX_CURVE25519	2147483646
------------------------------	--------------------	------------

diffie-hellman-group-exchange-sha256	SSH_KEX_DH_GROUP_EXCHANGE256	2147483645
diffie-hellman-group14-sha1	SSH_KEX_DH_GROUP_14	2147483644
diffie-hellman-group-exchange-sha1	SSH_KEX_DH_GROUP_EXCHANGE	2147483643
diffie-hellman-group14-sha256	SSH_KEX_DH_GROUP_14_SHA256	2147483642
ecdh-sha2-nistp521	SSH_KEX_ECDH_NIST_P521	2147483641
ecdh-sha2-nistp384	SSH_KEX_ECDH_NIST_P384	2147483640
ecdh-sha2-nistp256	SSH_KEX_ECDH_NIST_P256	2147483639

Default MAC Algorithms,

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

chacha20-poly1305@openssh.com	SSH_MA_POLY1305	2147483646
aes256-gcm	SSH_MA_AES256_GCM	2147483645
aes128-gcm	SSH_MA_AES128_GCM	2147483644
hmac-sha2-512	SSH_MA_HMAC_SHA2_512	2147483643
hmac-sha2-256	SSH_MA_HMAC_SHA2_256	2147483642
hmac-sha256@ssh.com	SSH_MA_HMAC_SHA256	2147483641
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	2147483640
hmac-sha1	SSH_MA_HMAC_SHA1	2147483639

Secret Server with FIPS Disabled

Default Encryption Algorithms, Non-FIPS

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

aes256-gcm@openssh.com	SSH_EA_AES256_GCM_OPENSSSH	2147483646
aes128-gcm@openssh.com	SSH_EA_AES128_GCM_OPENSSSH	2147483645

aes256-gcm	SSH_EA_AES256_GCM	2147483644
aes128-gcm	SSH_EA_AES128_GCM	2147483643
aes256-ctr	SSH_EA_AES256_CTR	2147483642
aes192-ctr	SSH_EA_AES192_CTR	2147483641
aes128-ctr	SSH_EA_AES128_CTR	2147483640
aes256-cbc	SSH_EA_AES256	2147483639
aes192-cbc	SSH_EA_AES192	2147483638
aes128-cbc	SSH_EA_AES128	2147483637
3des-cbc	SSH_EA_3DES	2147483636
twofish256-cbc	SSH_EA_TWOFISH256	36
twofish192-cbc	SSH_EA_TWOFISH192	35
twofish128-cbc	SSH_EA_TWOFISH128	34
serpent256-cbc	SSH_EA_SERPENT256	33
serpent192-cbc	SSH_EA_SERPENT192	32
serpent128-cbc	SSH_EA_SERPENT128	31
blowfish-cbc	SSH_EA_BLOWFISH	30
twofish128-ctr	SSH_EA_TWOFISH128_CTR	29
twofish192-ctr	SSH_EA_TWOFISH192_CTR	28
twofish256-ctr	SSH_EA_TWOFISH256_CTR	27
serpent128-ctr	SSH_EA_SERPENT128_CTR	26
serpent192-ctr	SSH_EA_SERPENT192_CTR	25
serpent256-ctr	SSH_EA_SERPENT256_CTR	24
blowfish-ctr	SSH_EA_BLOWFISH_CTR	23
idea-ctr	SSH_EA_IDEA_CTR	22
cast128-ctr	SSH_EA_CAST128_CTR	21

arcfour128	SSH_EA_ARCFOUR128	20
arcfour256	SSH_EA_ARCFOUR256	19
cast128-cbc	SSH_EA_CAST128	18
3des-cbc	SSH_EA_3DES	17
3des-ctr	SSH_EA_3DES_CTR	16
chacha20-poly1305	SSH_EA_CHACHA20	15
arcfour	SSH_EA_ARCFOUR	14
idea-cbc	SSH_EA_IDEA	13
chacha20-poly1305@openssh.com	SSH_EA_CHACHA20_OPENSSSH	12
des-cbc	SSH_EA_DES	11
none	SSH_EA_NONE	10

Default Key Exchange Algorithms, Non-FIPS

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

curve25519-sha256@libssh.org	SSH_KEX_CURVE25519	2147483646
diffie-hellman-group-exchange-sha256	SSH_KEX_DH_GROUP_EXCHANGE256	2147483645
diffie-hellman-group14-sha1	SSH_KEX_DH_GROUP_14	2147483644
diffie-hellman-group1-sha1	SSH_KEX_DH_GROUP	2147483643
diffie-hellman-group-exchange-sha1	SSH_KEX_DH_GROUP_EXCHANGE	2147483642
diffie-hellman-group14-sha256	SSH_KEX_DH_GROUP_14_SHA256	2147483641
ecdh-sha2-nistp521	SSH_KEX_ECDH_NIST_P521	2147483640
ecdh-sha2-nistp384	SSH_KEX_ECDH_NIST_P384	2147483639
ecdh-sha2-nistp256	SSH_KEX_ECDH_NIST_P256	2147483638

Default MAC Algorithms, Non-FIPS

Note: The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

chacha20-poly1305@openssh.com	SSH_MA_POLY1305	2147483646
aes256-gcm	SSH_MA_AES256_GCM	2147483645
aes128-gcm	SSH_MA_AES128_GCM	2147483644
hmac-sha2-512	SSH_MA_HMAC_SHA2_512	2147483643
hmac-sha2-256	SSH_MA_HMAC_SHA2_256	2147483642
hmac-sha256@ssh.com	SSH_MA_HMAC_SHA256	2147483641
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	2147483640
hmac-sha1	SSH_MA_HMAC_SHA1	2147483639
umac-128@openssh.com	SSH_MA_UMAC128	2147483638
umac-96@openssh.com	SSH_MA_UMAC96	2147483637
umac-64@openssh.com	SSH_MA_UMAC64	2147483636
umac-32@openssh.com	SSH_MA_UMAC32	2147483635
hmac-sha2-512-etm@openssh.com	SSH_MA_HMAC_SHA2_512_ETM	28
hmac-sha2-256-etm@openssh.com	SSH_MA_HMAC_SHA2_256_ETM	27
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	24
hmac-ripemd160	SSH_MA_HMAC_RIPEMD160	23
hmac-ripemd	SSH_MA_HMAC_RIPEMD	22
hmac-ripemd160@openssh.com	SSH_MA_HMAC_RIPEMD_OPENSSSH	21
hmac-sha1-96	SSH_MA_HMAC_SHA1_96	15
hmac-md5	SSH_MA_HMAC_MD5	13
hmac-md5-96	SSH_MA_HMAC_MD5_96	12
none	SSH_MA_NONE	10

Note: Please click the table of contents on the left to see the sub-pages to this one. Click the table of contents on the right to see headings on this page.

SSH Blocked Command Lists

Important: This feature is part of the early release of Secret Server 10.11. The general release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version.

Overview

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that you can apply to secrets with SSH launchers over SSH proxy. With privilege management, you can grant users access to a machine to block specific commands that a user may run as root or any other privileged account.

Note: To use command restrictions, Secret Server must have SSH Proxy and Enable Block Listing enabled.

With SSH blocked command lists, you can define disallowed commands when connecting as a privileged account. The blocked command list is defined by a series of regular expressions.

Upon launching a secret with an assigned SSH command blocklist, each command sent to the target is evaluated for a match on the blocklist. If the command is found to match a list entry, that command is blocked from execution. Blocked output is shown to the user at the terminal.

Requirements

System requirements:

- Secret Server 10.11 or later
- Secret Server Platinum Edition license or Secret Server Professional and Unix SUPM license
- SSH proxy must be enabled

Creating SSH Blocked Command Lists

The format for specifying a blocked command follows a regular expression syntax that is typical to most scripting languages. The blocked commands are surrounded with `\b` word boundary anchors. For example, to block the `sudo` command, you use:

```
\bsudo\b
```

This expression blocks the execution of any command with the `sudo` string in it (as a separate word), for example, these are blocked:

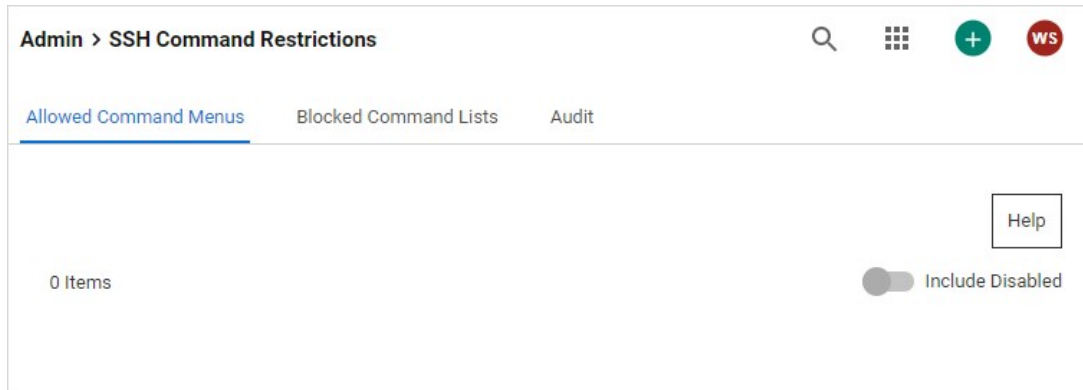
- `sudo`
- `sudo root`
- `sudo ls /usr/local/protected`
- `sudo shutdown -r +15 "quick reboot"`

And these are not:

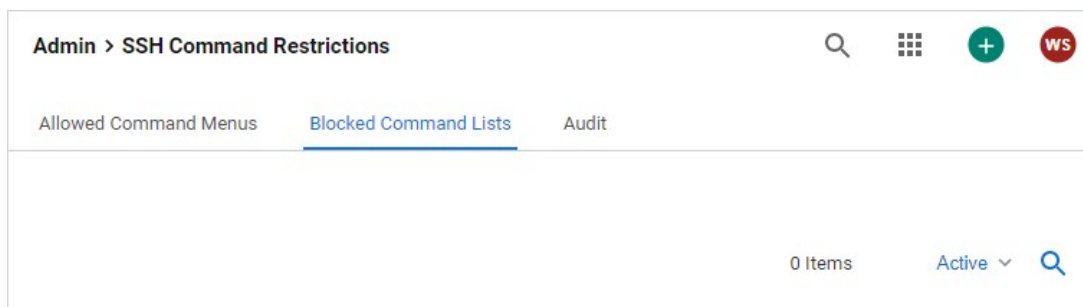
- `cat sudoku`
- `echo "sudo"`

To create a list of blocked commands:

1. Go to **Admin > See All**.
2. Hover the mouse pointer over the **Actions** menu item and select **SSH Command Restrictions**. The SSH Command Restrictions page appears:



3. Click the **Blocked Command List** tab:



4. Click the **Create Blocked Command List** button. The New Blocked Command List page appears:

New Blocked Command List

Name *

Enabled *

Description

Blocked Commands *

Prevent superuser commands

Prevents superuser commands

Selected Commands (2)

Blocks su commands : \bsu\b	Remove
Blocks sudo commands : \bsudo\b	Remove

Add Commands to Blocked Command List

- +
Blocks copy commands : \wcp\b
- +
Blocks move commands : \bmv\b
- +
Blocks reboot commands : \breboot\b
- +
Blocks remove commands : \brm\b
- +
Blocks shred commands : \bshred\b
- +
Blocks wget commands : \bwget\b

Cancel

Create Blocked Command List

5. Type the list name in the **Name** text box.
6. Ensure the **Enabled** check box is selected.
7. **Either** select a predefined regex from the **Add Commands to Blocked Command List** dropdown list. **Or** add a custom regex of your own:
 1. Scroll to the bottom of the dropdown list.
 2. Click the **Create New Command** link. The Create Command and Add to Blocked Command List popup appears:

Create Command and add to Blocked Command List

Command Name *

Command Pattern(regex) *

1	\bls\b
---	--------

3. Type the name for the command in the **Command Name** text box.
4. Type or paste the regex in the **Command Pattern** text box.
5. Click the **Create and Add** button. The command is added to the dropdown list.
8. Click the **Create Blocked Command List** button.

Applying SSH Command Blocked Lists in Secret Settings

To enable privilege management for an account:

1. Navigate to a PuTTY secret's **Settings** tab.
2. Go to the **SSH Launcher** section.
3. Click the **Edit** link.
4. Click the **Connect Using** dropdown list and select **Credentials on Another Secret**.
5. Click the **No Secret Selected** link to choose a secret containing your log on credentials, which the launcher uses when logging on the SSH service.
6. Enable command restrictions:
 1. Click the **Security** tab.
 2. Go to the **Other Security** section.
 3. Click the **Edit** link to set **Enable Proxy** to Yes.
 4. Click the **Edit** link for the **Enable SSH Command Restrictions**. The Edit SSH Command Restrictions popup appears:

Edit SSH Command Restrictions

Restrict SSH Commands

Allowed Command Menus
 Blocked Command Lists

Owner Permission Unrestricted

Edit Permission Admin Command Blocklist

View Permission User Command Blocklist

Cancel
Save

5. Ensure the Restrict SSH Commands check box is selected.
6. Click to select the Blocked Command Lists selection button.
7. Click the **Owner**, **Edit**, and **View Permission** dropdown lists to map the blocked command lists to users via those permissions. You can also leave them as unrestricted.
8. Click the **Save** button.

SSH Command Restrictions via a Secret Policy

You can apply SSH command restrictions to a secret policy for ease of management. You can apply secret policies to secret folders or directly to a secret itself. To apply command restrictions, set a policy as follows:

Table: Secret Policy Security Settings for SSH Command Restrictions

Security Settings	Enable Proxy	Enforced	Checked
Security Settings	Enable SSH Command Restrictions	Enforced	Checked
Security Settings	SSH Command Restriction Type	Enforced	Blocked List
Security Settings	SSH Command Blocklist for Secret Owners	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Editors	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Viewers	Enforced or Not Set	Desired Block Command List or Not Set

SSH Command Menus

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that can be applied to secrets with SSH Launchers over SSH Proxy. Privilege management gives the ability to grant users access to a machine with specific command restrictions to define the available commands that a user may run as root or another privileged account.

Note: To use command restrictions, Secret Server must have SSH Proxy enabled.

With command menus, you can configure predefined commands that users or groups will be able to access when connecting as a privileged account. A command menu is a list of command names mapped to system commands. The format for specifying a command is to separate a name and command with an equals symbol. For example:

```
restart_apache = /usr/sbin/service apache restart
```

You may also use parameters in commands so users can execute more complex commands. For example:

```
move_file = /bin/mv $src $dst
```

You can specify environmental variables by escaping dollar signs in commands. For example:

```
go_home = cd $$HOME
```

Command restrictions currently do not support complex commands, such as multiple commands on one line, piping, or output redirection. To support these functions, you may add a script to the system that has these capabilities and point map the command to that script. We highly recommend that generated scripts have proper user permissions and that the absolute path is used. The absolute path ensures that the correct script is being executed.

Commands may not be named as numbers or one of the following predefined commands:

```
..
up
-help
?
-more
logout
exit
```

To enable privilege management for an account, navigate to a PuTTY Secret's launcher tab and specify the "Connect As" secret that you wish to connect as. When launching this secret, the launcher uses it as credentials to log into the SSH service and then log into the credentials specified on the secret.

To enable command restrictions, navigate to a PuTTY secret's security tab and specify "Enable Proxy" and "Enable SSH Command Restrictions." This gives you the ability to map users and groups to command menus. When the unrestricted command menu is specified for a user, the user is launched into a normal shell environment without command restrictions. Likewise, if the "Allow Owners Unrestricted SSH Commands" option is enabled, the owners of the secret are also launched into a normal shell environment without command restrictions.

When specifying command menus on a secret, at least one command menu must be selected unless "Allow Owners Unrestricted SSH Commands" is enabled.

You can apply command restrictions to a secret policy for ease of management.

An *SSH jumpbox route*, is a series of regular Linux servers, accessible from the Internet, that is a gateway to other Linux machines on a private network using the SSH protocol. This topic and its subtopics address using jumpbox routes.

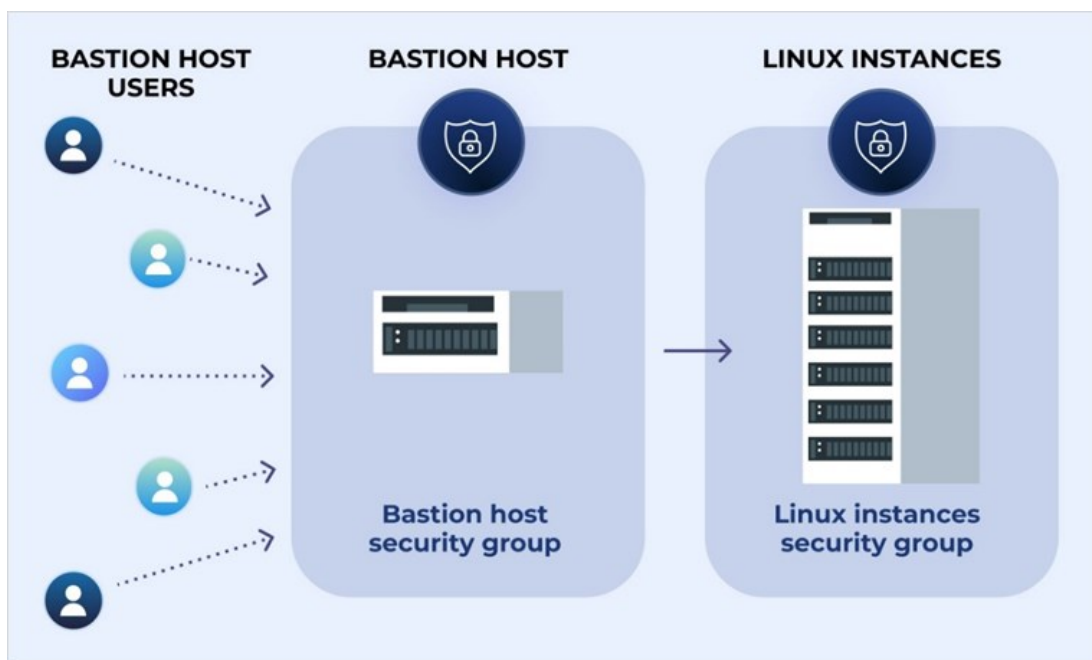
Note: SSH jumpboxes are also called *bastion hosts*, *jump hosts*, or *jump box servers*. *Bastion* is a military term meaning a *projecting part of a fortification*. Bastion hosts are hardened and monitored servers that reside outside of an organization's security zone, usually exposed to the internet. All jumpboxes are bastion hosts, but all bastion hosts are not necessarily jumpboxes.

Because SSH jumpboxes usually reside on the Internet, they run a minimum of services to reduce their attack vulnerability. Similarly, limiting the Internet access to your infrastructure to one hardened gateway server also reduces risk. In addition, a dedicated SSH access point makes it easier to have an aggregated audit log of all SSH connections.

With early SSH, users had to SSH into a jump host and then type `ssh` again to manually jump to a destination host. Today, this is done automatically using the built-in SSH `-J` ProxyJump option.

Secret Server can now create a chain of jumpbox secret connections to reach an otherwise inaccessible Linux instance. This sequence is called a *jumpbox route* and can contain up to 20 jumpbox levels (hops between instances).

Figure: SSH Jumpbox Route Setup



Best Practices for Jumpbox Routes

Note: See [SSH Jumpbox Routes](#) for an introduction.

Overview

This topic discusses configuring an SSH jumpbox route using OpenSSH.

We strongly recommend a dedicated SSH jumpbox without any other publicly accessible software on the machine. Occasionally, multiple jumpboxes (a jumpbox route) may be required to access a target machine that is behind several layers of network security.

Assumptions

We want:

- The jumpbox to forward SSH connections to our internal hosts. Users must authenticate to the SSH jumpbox and the internal target host. Users need valid credentials for both hosts.
- A single shared user for everyone with no interactive terminal sessions allowed.
- Users to connect to internal target hosts using Secret Server's jumpbox route feature, which uses the jumpboxes as jump route "levels" to ultimately launch a target secret that has this jumpbox route selected.

Best Practices

We recommend not allowing users to log into a jumpbox directly because they might introduce security issues by:

- Inadvertently updating the jumpbox configuration
- Using the jumpbox machine for unrelated tasks.
- Making copies of cryptographic keys used to access destination servers.

We recommend changing the default TCP port on the SSH jumpbox from 22 to something else.

Example Setup

The following example uses these conventions:

- The organization domain is example.com.
- The DNS name of the jumpbox is proxy.example.com, which is the only machine accessible from the Internet.

Task 1: Launching a New Linux Instance

- Stand up a Linux instance on your cloud provider. We use Ubuntu 20.04 LTS because it is simple, well-supported, and includes the recently released OpenSSH 8.2.
- Set up a firewall or security group policy to restrict connections to the jumpbox to port 22 (SSH).
- Allow connections only from IPs you trust.

Task 2: Configuring OpenSSH

- Read and apply [Mozilla's OpenSSH Security Guidelines](#).

Note: The guidelines only cover up to OpenSSH 6.7. Most are still relevant to OpenSSH 8.2.

- OpenSSH is bundled by default with most Linux distributions. It is almost certain your Linux machine already has it installed. If the server is accessible via proxy.example.com, clients can access other servers behind the same NAT boundary via the -J command line flag. For instance: `$ ssh -J proxy.example.com 10.2.2.1`

In the example above, the jumpbox is used to access another host on an AWS VPC with an address of 10.2.2.1.

Task 3: Configuring Public Key Authentication and Disabling Root Logins

Configuring the sshd_config File

In your new jumpbox's `/etc/ssh/sshd_config` file, consider adding the following SSHD config parameters:

```
# Supported HostKey algorithms by order of preference.
HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_rsa_key

# Password based logins are disabled - only public key based logins are allowed.
AuthenticationMethods publickey

# LogLevel VERBOSE logs user's key fingerprint on login. Needed to have a clear audit track of which key was using to log in.
LogLevel VERBOSE

PermitRootLogin no

# Log sftp level file access (read/write/etc.) that would not be easily logged otherwise.
Subsystem sftp /usr/lib/ssh/sftp-server -f AUTHPRIV -l INFO
```

The setup above works only when the public SSH keys are properly distributed, not only between clients and the jumpbox but also between the clients and the destination servers. With these settings, password authentication is disabled.

Setting Key Exchange Algorithms, Ciphers, and MACs

Consider which algorithms and key types you want to support. Mozilla recommends the following key types (more restrictive than the OpenSSH defaults):

In `/etc/ssh/sshd_config`, add:

```
KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com
```

Task 4: Disabling Forwarding

Next, harden the server configuration by disabling interactive SSH sessions on the jumpbox for regular users, leaving it turned on for the administrators. To do this, update the sshd configuration `/etc/ssh/sshd_config` with the following:

```
# Do not let SSH clients do anything except be forwarded to the destination:
AllowAgentForwarding no
AllowStreamLocalForwarding no
X11Forwarding no
```

If you only have a few users, consider creating a separate user account on the jumpbox dedicated to "jumping users." For example, call it "jumpuser" and update the configuration `/etc/ssh/sshd_config`:

```
Match User jumpuser
PermitTTY no
X11Forwarding no
PermitTunnel no
GatewayPorts no
```


Task 5: Changing the Default SSH Port

If you do not want to restrict access by IP address in your security group rules, consider changing your default SSH port. This will deter many basic bots.

In `/etc/ssh/sshd_config`, add Port 37271.

Task 6: Testing Configuration and Restarting SSHD Service

You can test your configuration with `sshd -t`, then restart the SSHD server. Make sure you can still ssh into the machine before you continue!

Task 7: Send Your Users' SSH Keys to the Jumpbox

Add the jump user's public keys to the `/home/jumpuser/.ssh/authorized_keys` file. These keys can be stored in Secret Server and used during the Jumpbox Route launch session.

Using Bastion Hosts with Secret Server and Jumpbox Routes

Please see [Creating and Testing Secrets for Jumpbox Routes](#).

References

Johnson, Brian. "How to Create a Bastion Host | Part 1 of a Step-by-step Tutorial." *Strongdm*, 25 March, 2021, <https://www.strongdm.com/blog/bastion-hosts-with-audit-logging-part-one>, Accessed 12 October, 2021

Kontsevov, Ev. "How to Set Up an SSH Jump Server." *Teleport*, 30 March, 2021, <https://goteleport.com/blog/ssh-jump-server/>, Accessed 12 October, 2021

Tashian, Carl. "DIY SSH Bastion Host." *Smallstep*, 08 July 2020, <https://smallstep.com/blog/diy-ssh-bastion-host/>, Accessed 12 October, 2021

Creating and Testing Secrets for Jumpbox Routes

Overview

This topic shows how to create secrets used for jump server routes by way of an extended example.

Note: Please see [Best Practices for Jumpbox Routes](#) for creating hardened jump servers for use in jump server routes.

We use following scenario:

A target Ubuntu host is only accessible via an SSH proxyjump directive. The sequence is Secret Server to bastion01.thycotic.com to bastion02.thycotic.com to remote-hostname.thycotic.com, the target Ubuntu host.

As suggested in the **Bastion Design Best Practices** guide:

- Both bastion hosts 01 and 02 have changed their default SSH ports for all incoming connections to 2222 and 3333, respectively.
- Both use a single jump user, betty on bastion01 and wilma on bastion02.
- Both users have SSH public key authentication enabled and have an `/home/<username>/.ssh/authorized_keys` file storing all the public keys.

You can connect to the target server without Secret Server from any SSH client using the following SSH command, with the built-in SSH ProxyJump directive `-J` and defined aliases:

```
$ ssh -J <jumpbox1>, <jumpbox2> <remote-host-target>
```

...with local `~/.ssh/config` aliases configured this way:

```
### First Jumpbox
Host jumpbox1
  HostName bastion01.thycotic.com
  Port 2222
  User betty
  IdentityFile /home/betty/.ssh/jumpbox1_ssh_rsa

### Second Jumpbox
Host jumpbox2
  HostName bastion02.thycotic.com
  Port 3333
```

To replicate this jump scenario using Secret Server, we need to create three new secrets and a new jumpbox route, and then assign that route to the target secret. Once that is done, launching the remote-host-target secret is as simple as clicking the PuTTY launcher icon—the jumpbox route is automatically set up for you.

Task 1: Creating Jumpbox and Target Host Secrets

Note: All three connections require SSH public key authentication.

1. Create a new secret based on the Unix Account (SSH Key Rotation) template with these particulars:
 - Secret name: bastion01
 - Machine: bastion01.thycotic.com
 - Username: betty
2. Upload or generate the private/public key pair. Set a passphrase if necessary.
3. If the key pair is generated, save the public key on the secret to the user's `/home/betty/.ssh/authorized_keys` file on the jumpbox.
4. Create a second new secret also based on the Unix Account (SSH Key Rotation) template with these particulars:
 - Secret name: bastion02

- Machine: bastion02.thycotic.com
 - Username: wilma
5. Upload or generate the private/public key pair. Set a passphrase if necessary.
 6. If the key pair is generated, save the public key on the secret to the user's `/home/wilma/.ssh/authorized_keys` file on the jumpbox.
 7. Create a third new secret also based on the Unix Account (SSH Key Rotation) template with these particulars:
 - Secret name: remote-host-target
 - Machine: remote-hostname.thycotic.com
 - Username: remote_user
 8. Upload or generate the private/public key pair. Set a passphrase if necessary.
 9. If the key pair is generated, save the public key on the secret to the user's `/home/remote_user/.ssh/authorized_keys` file on the target host.

Task 2: Creating a Jumpbox Route with Secret Levels

Note: See [Creating and Editing Jump Server Routes](#) for more on creating jumpbox routes.

1. Create a new jumpbox route named flintstone.
2. For level 1, select the bastion01 secret and choose 2222 as the port.
3. For level 2, Select the bastion02 secret and choose 3333 as the port.

Task 3: Assigning a Jumpbox Route on the Target Secret

1. Open the remote-host-target secret.
2. Ensure SSH proxying is enabled (Security > Enable SSH Proxy) on the secret.
3. Navigate to the secret's **Settings** tab.
4. Scroll down to edit the **Jumpbox Route** section.
5. Select the **Flintstone Route** in the drop-down.
6. Save your changes.

Task 4: Test Launching the Target Secret with PuTTY Launcher over SSH Proxy

1. Open the remote target secret.
2. Click the **General** tab.
3. Click the **PuTTY Launcher** link to begin a new SSH proxy session with a jump server route. The launcher connects to the target via the jump server route:

```

toor@dev-ssh-client01:~
Using username "30005f3e-6ca2-441e-aac5-35be56f9d9ef".
Keyboard-interactive authentication prompts from server:
End of keyboard-interactive prompts from server
Pre-authentication banner message from server:
| Welcome to Thycotic SSH Proxy!
End of banner message from server

Connecting to target through jumpbox route...
Successfully connected to jumpbox level #1. Forwarding to local port: 17392
Successfully connected to jumpbox level #2. Forwarding to local port: 17365
Jumpbox route complete. Connecting to target [toor@10.61.10.200:22]...

Last login: Tue Oct 12 14:50:44 2021 from 10.60.23.141
[toor@dev-ssh-client01 ~]$ █

```

Task 5: Test Launching the Target Secret with SSH Terminal

1. Note the secret ID of your remote target secret.
2. Using an SSH client, log in to SSH Terminal. See [SSH Terminal Administration](#) for details.
3. Enter `cat <secret id>` to view the remote target secret's details:

```

THY-01-0325-LT.testparent.thycotic.com - PuTTY
launch - begin SSH Proxy session using credentials on specified Secret
man - an interface to the on-line reference manuals
search - search for Secrets by keyword
exit - exits this terminal session
clear - clears the screen
[j@c.com@127.0.0.1 ~] $ cat 79
Secret Name:    remote-host-target
Secret ID:     79
Secret Type:   Unix Account (SSH Key Rotation) [ID:6029]
Folder:       \Launchers\Jumpbox Route (19)
SSH Key:      *****
Machine:      remote-hostname.thycotic.com
Username:     remote_user
Password:     *****
Private Key:   *** Not Valid For Display ***
Private Key Passphrase: *****
Public Key:   *** Not Valid For Display ***
SECURITY
  Restrict SSH Commands:  disabled
  Session Recording Enabled: False
-----
LAUNCH INSTRUCTIONS
Can be launched with Terminal command:

launch 79

-----
Jumpbox Route: Flintstone Route (ID: f89f35b2-0934-4b44-898c-80802d630efc)
SSH Key Secret: on the Secret (if available)
Connect as Secret: Credentials on the Secret
[j@c.com@127.0.0.1 ~] $ █

```

4. Enter `launch <secret id>` to launch the remote target secret with a jumpbox route:

```
[j@c.com@127.0.0.1 ~] $ launch 74
Secret Server Launch: Secret ID 74 found. Attempting launch...

Connecting to target through jumpbox route...
Successfully connected to jumpbox level #1. Forwarding to local port: 16634
Successfully connected to jumpbox level #2. Forwarding to local port: 18676
Jumpbox route complete. Connecting to target [toor@10.61.10.200:22]...

Proxy shell tunnel to target opened. Attempting to connect...
Last login: Tue Oct 12 13:00:29 2021 from 10.60.23.141
[toor@dev-ssh-client01 ~]$
```

5. You should now be connected to remote-hostname.thycotic.com as remote_user.
6. Enter exit to return to the terminal.

Creating and Editing SSH Jumpbox Routes

Note: See [SSH Jumpbox Routes](#) for an introduction.

Introduction

You can add jumpbox routes to Unix secrets when SSH proxying is enabled. A jumpbox route assigned to a target secret specifies a series of Unix servers called *jumpboxes*, *jumpbox servers*, or *bastion hosts*. The proxied connection is forwarded through the jumpbox route before reaching the server specified in the target secret.

Prerequisites

Using jumpbox routes requires these prerequisites:

- Only use Unix servers and secrets: At this time, jumpbox routes are only supported by Unix secrets. The option to add a jumpbox route does not appear on any target secret that is not based on the Unix secret template. Similarly, a jumpbox route that attempts to connect to jumpbox servers that do not use a Unix secret template will also fail.
- Enable SSH proxy on Secret Server: For more information on how to enable proxying on Secret Server, see [SSH Proxy Configuration](#).
- Enable the SSH Proxy setting for the target secret using a jumpbox route: To assign a jumpbox route to a target secret, the target secret itself must have SSH proxying enabled. The enable SSH Proxy check box appears in the secret's Security tab if proxying is enabled on Secret Server itself.

Permissions

Using jumpbox routes requires these permissions:

- Adding a jumpbox route to a target secret: A user must have owner permissions on a secret to assign, change, or remove that secret's jump server route. Additionally, users are only able to pick from a list of routes where they have at least list permission on the first jump route server.
- Editing Jumpbox Routes: Users must have the "Administer Jumpbox Route" permission to create, edit, or deactivate jump server routes. Users with the "View Jumpbox Route" permission can view the details of all jump server routes in the Admin Jumpbox Route page, but they cannot make any changes.

Creating, Editing, and Deactivating Jumpbox Routes

1. Go to **Admin > See All**.
 2. Search for and select **Jumpbox Routes**. The Jumpbox Routes page appears:
-

Custom URL needs to be defined as it is used to build links in email. ×

Admin > Jumpbox Routes Q + JC

Configure Jumpbox Routes in order to select which machines and networks are used as intermediaries between the user and the target secret. Permissions to assign to Secret are based off having List permission on the Level 1 Secret in the Route.
[Learn More](#)

2 Items Active Q Create Jumpbox Route

NAME	↑ DESCRIPTION	ACTIVE	⌵ ↓
Router Network	Contains router machines for providing VPN access	Yes	
Secure DC Route	Contains server machines off the main lan network	Yes	

This page lists all existing jumpbox routes.

3. Select an existing route to edit or create a new route by clicking the **Create Jumpbox Route** button.
4. If you chose to create a new route, a popup appears. Type the name and description and click the **Create Jumpbox Route** button.
5. In either case, the Jumpbox Routes page appears for the new or edited route:

Custom URL needs to be defined as it is used to build links in email.

Admin > Jumpbox Routes > Router Network

General Audit

Jumpbox Route Status [Edit](#)

Disable a route to prevent it from being used. **Active** Yes

Jumpbox Route Details [Edit](#)

Configure Jumpbox Routes in order to select which machines and networks are used as intermediaries between the user and the target secret. Permissions to assign to Secret are based off having List permission on the Level 1 Secret in the Route.

Route Name * Router Network

Route Description Contains router machines for providing VPN access

Jumpbox Route Levels

Each level represents a bastion host used in the jumpbox route. Ports must be specified at each level. Permissions of the Jumpbox Route are based on the Level 1 Secret. Users need List on the Level 1 Secret to assign to the Secret on the Secret Settings tab.

2 Items [Add Level](#)

ORDER	↑	PORT	SECRET	⌵
1		22	Custom Lau...	
2		22	runscripts@...	

6. Note that routes can only be deactivated, not deleted, in the **Jumpbox Route Status** section.
7. You can view or edit the the route's jump servers in the **Jumpbox Route** Levels section. Each row in the grid represents a *level*, each of which has these attributes:
 - Secret: Each level contains one secret that represents a jump server.
 - Port: The port used on the jump server for the route. Choose an available port on which the server listening for SSH connections. Typically, this is port 22, but we recommend changing it to deter basic script bots, see [Best Practices for Jumpbox Routes](#) for more information.

Note: This value overrides any port specified on the secret.
 - Order: The sequence of jumpboxes forwarding the proxied connection, starting with 1. You can hover over any level to present a handle for dragging it to reorder the list.
8. Click the **Add Level** hyperlink to add a jumpbox to the route. An Add Level popup appears.

9. Select or type your desired port in the **Port** list box.
10. Click the **No Secret Selected** hyperlink to choose the secret for the new level (server).
11. Click the **Save** button.
12. Hover over any level to present a handle (on the left) for dragging the level to reorder the list.
13. Hover over any level to present an ellipsis icon (on the right).
14. Click the icon and choose **Edit** or **Delete** to change or remove the level. You can change the port or related secret. Deletion permanently removes the level from the jumpbox route. It is not deactivated as with the route itself.

Assigning a Jumpbox Route to a Secret

1. Select or create a Unix secret for the route. Ensure that proxying and all other prerequisites described above are met.
2. Navigate to the secret's **Settings** tab and go to the **Jumpbox Route** section.

CentOS Test Server ☆

General Security Audit Remote Password Changing Dependencies Sharing Settings Metadata

Email Notifications [Edit](#)

Choose your personalized email settings for this specific Secret.

Send Email When Viewed	No
Send Email When Changed	No
Send Email When Heartbeat Fails	No

SSH Launcher [Edit](#)

Configure your settings when launching SSH for this Secret.

Connect Using	Credentials on this Secret
----------------------	----------------------------

Jumpbox Routes [Edit](#)

Choose a Jumpbox Route for this Secret.

Jumpbox Route	No Jumpbox Route
----------------------	------------------

3. Click the **Edit** hyperlink. The section becomes editable:

The screenshot shows a dialog box titled "Jumpbox Routes". Inside the dialog, there is a label "Choose a Jumpbox Route for this Secret." followed by a "Jumpbox Route" label and a dropdown menu. The dropdown menu currently displays "No Jumpbox Route" with a downward arrow. Below the dropdown are two buttons: "Cancel" and "Save".

4. Click to select the desired jumpbox route from the drop-down. Only active jump server routes to which you have access are visible. You can unassign the jumpbox route by selecting **No Jumpbox Route** in the dropdown list.

Note: If the jumpbox route becomes deactivated, "Router Network (inactive)" appears in the Jumpbox Route dropdown list. You should select a new one.

Note: You can also assign jumpbox routes through a secret policy. See [Creating Secret Policies](#).

Global Jumpbox Settings

The **Admin > Proxying** page has these jumpbox settings that impact your jumpbox routes:

- **Tunnel Keep Alive:** This setting is to stop intermediaries (such as firewalls or proxies) from deeming your connections inactive and timing them out and closing them. The default is 50 seconds (just shy of a common timeout setting). Adjust this to be just short of your network's timeout setting. Type 0 to disable the setting.
- **Available Port Range:** This sets the port range for the SSH proxy endpoint for jumpbox forwarding. Type a range separated by a hyphen, such as 10000-15000. The ports are used for local port forwarding for jumps. You cannot use the ports for other processes, so we recommend using a high range (ports above 10000).

The Secret Server proxy routes SSH and RDP sessions and helps protect the endpoint credentials. There are two configuration options for proxying:

- Proxy through the Secret Server Web application
- Proxy through a distributed engine

Note: To learn more about RDP Proxying, please see [RDP Proxy Configuration](#).

Enabling Proxy

1. Go to **Admin > Proxying**.
2. Enable **SSH Proxying**.
3. Generate a new key.
4. To enable proxying on Web nodes, edit the row in the **Endpoints** tab to set the **Public Host** and **Bind IP Address**. For a standard server, these can be the same, but if the public IP of the server is not set on the server (such as a load balancer or an EC2 instance with an elastic IP), they will be different.
5. To enable proxying for a specific site and all engines within that site, edit the row in the **Sites** section and enable proxying and set the **SSH Port**.
6. The engines for the sites are listed in the **Engines** section. The **Hostname/IP Address** is the public host or IP the launcher connects to and the **SSH Bind Address** is the IP on the server that the SSH proxy is listen on. Again, these will typically be the same, but may be different if the resolvable IP or host of the engine machine is different than the IP on the network adapter on the machine.
7. Enable proxying on a secret with a PuTTY launcher. The launcher now connects to the assigned site, which is set on the **General** tab. If the site has proxying enabled, it will go through the engines available in the site, otherwise it will use the Secret Server Web application proxy.

Web Application Proxy Performance

Minimum Hardware

- Intel 3.7 GHz Quad Core
- 16 GB of RAM
- 100 MB/s plus network capability

Session Activity

We tested sessions with standard usage, such as opening and modifying files and navigating the file system on Linux. On Windows, the activity was opening MMC snap-ins, editing files, and copying files through the RDP session. If you have constant large file transfers across multiple concurrent sessions or otherwise transferring large amounts of data (such as streaming a video through an RDP session), the maximum concurrent sessions will be significantly reduced.

Table: Concurrent Proxy Sessions

SSH	300
-----	-----

RDP	100
-----	-----

Proxy Connections

Connections from the user to the proxy are over SSH, and you can configure the port. The user's machine will connect to either an engine SSH proxy or the Secret Server Web application SSH proxy.

Figure: Default Secret Server Web Application Proxy (example)

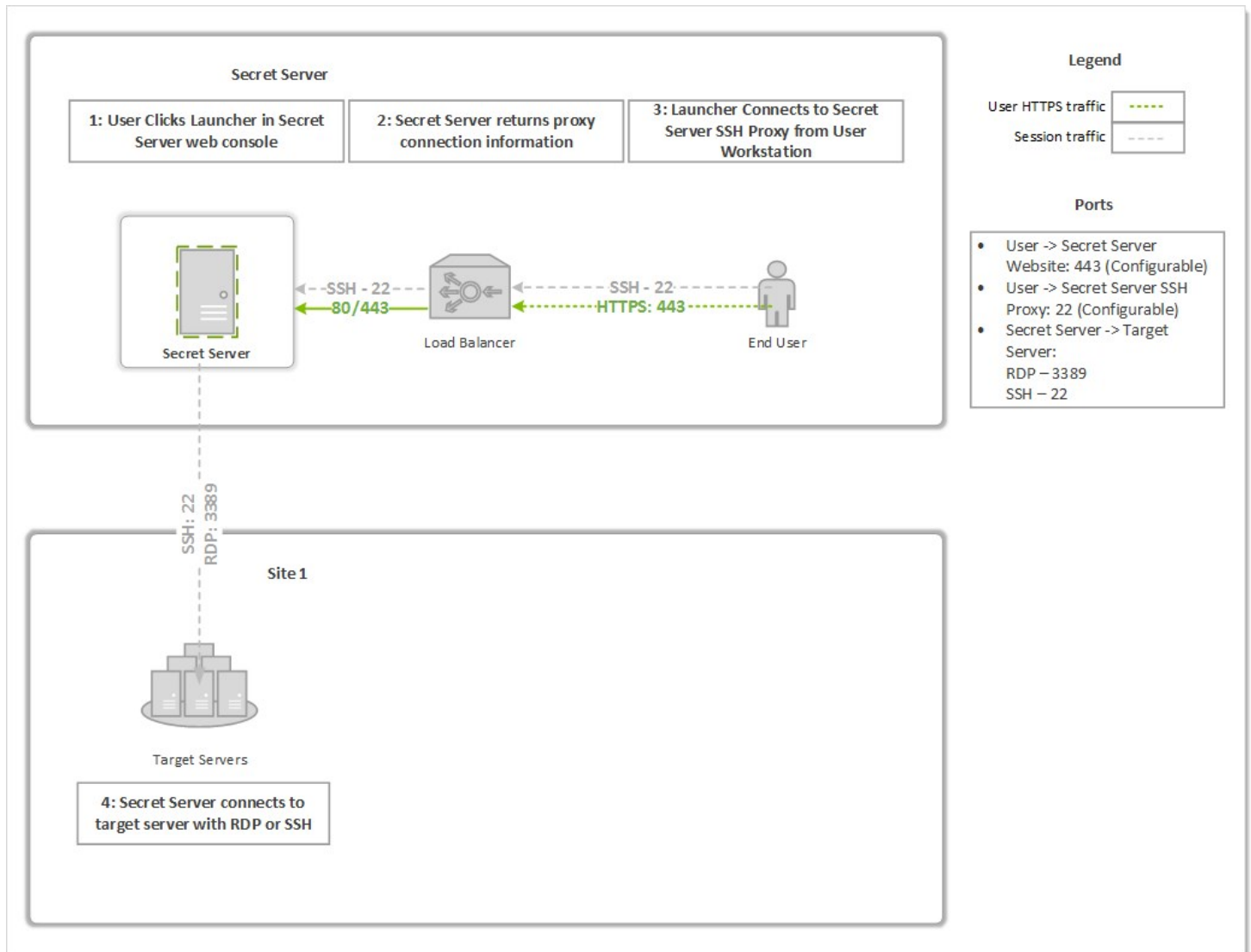
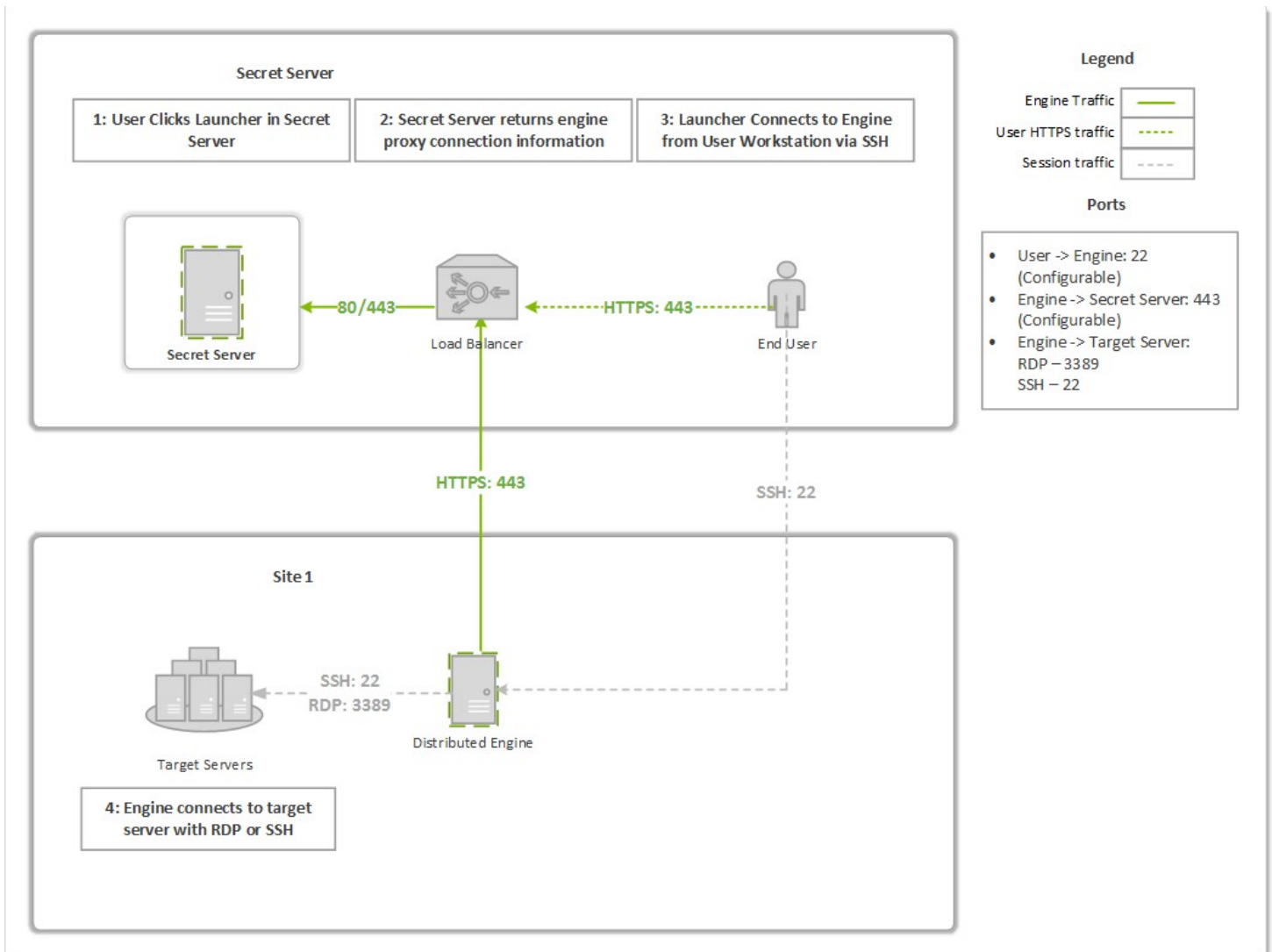


Figure: Proxy through a Distributed Engine (example)



SSH Proxy with Multiple Nodes

If you are using clustering with Secret Server, you can pick exactly which of your nodes act as a SSH proxy by going to the **Admin > Proxying** page and scrolling down to the **Nodes** section. For each node you wish to be a proxy, configure the **SSH Public Host** (must be an IP address, not a DNS name) and the **SSH Bind IP Address** (use 0.0.0.0 to easily bind to all IPv4 Ps on a server). There is no need to configure all nodes if you do not want them all to be proxies.

As soon as the IPs are saved for each node, the node should start listening on the SSH proxy port. You can verify that with netstat. If you do not see the node listening on your chosen port, perform an IIS reset and hit its Secret Server website. It should be listening once Secret Server starts up again. For example:

```
C:\Users\Administrator>netstat -ano | find ":22"
TCP 0.0.0.0:22 0.0.0.0:0 LISTENING 3600
```

Now, when a user connects to the Secret Server Web page, if the node they are hitting is setup to be a SSH proxy, they will connect to that node's SSH public host IP. If the node they are connected to is not setup to be a SSH proxy, then users will round robin between the other nodes that are SSH proxies and connect to their SSH public host IP.

Introduction

This document discusses using an SSH terminal with Delinea Secret Server.

Feature Summary

- Connect using SSH to a terminal hostname and port to log in to terminal and run commands
- Display custom terminal banner after successful connection
- Display available commands on successful login (display again with `man` command)
- Log in to the terminal as a Secret Server user (SSH Proxy must be enabled)
- Can set an inactivity timeout. Can be set to *disabled* or with a two-minute minimum.
- Start a terminal connection and launch in a single line. For example:

```
ssh <user>@<ss_ip> -t launch <secret_id>
```
- Use two-factor authentication (2FA) for access (optional)
- Use the SSH terminal interface to Secret Server for viewing and launching secrets
- Use these commands:
 - `Man` command to display detailed command description
 - `Search` command to display matching secrets
 - `Cat` command to display secret details of with specified secret ID
 - `Launch` command to begin a Proxy launch session with specified secret ID
- Use up and down keystrokes for command history
- Supports custom SSH command menus and session recording logging

Requirements

System Requirements

- Secret Server 10.7.000000 Secret Server
- Secret Server **Professional** or **Platinum** Edition license

Recommended

[RabbitMq Site Connector](#)

Secret Server Permission Requirements

Admin:

- Administer Configuration
- Administer Proxying Configuration
- View Configuration
- View Proxying Configuration

User: View Secret

Configuring SSH Terminal

Enabling SSH Terminal on Secret Server

1. Prerequisites:

- Must meet Admin permission requirements (see [Secret Server Permission Requirements](#))
- Secret Server **Professional** or **Platinum** Edition license

2. Navigate to **Secret Server > Admin > Proxying**.

SSH Proxy Configuration


[Explain](#)

SSH PROXY SETTINGS

Enable Proxy	Yes
Enable SSH Tunneling	Yes
Proxy New Secrets By Default	Yes
SSH Banner	Welcome to Secret Server
SSH Proxy Host Fingerprint	SHA1 - 50:2d:99:d9:f3:2a:b8:9d:68:b4:9e:a5:2b:a2:9a:18:2f:b8:bf:61 MD5 - 04:9e:8b:44:f1:ed:5b:fd:e1:18:79:9c:9c:fb:66:41
Enable Inactivity Timeout	No

SSH TERMINAL SETTINGS

Enable Terminal	No
-----------------	----

 Edit

3. Click the **Edit** button.

4. Type your SSH proxy configuration settings (see [Configuring SSH Proxies for Launchers](#)):

1. Enable **SSH Proxy** (required to use SSH terminal).
2. (optional) Enable **Proxy New Secrets by Default**.

Note: To launch a secret via the terminal, the secret must have proxy enabled. Only SSH-based credentials can be launched in the terminal.

1. Click to enable **SSH Terminal**.
2. (optional) Customize the **Terminal banner** for your environment.
3. (optional) Click to enable **Terminal Inactivity Timeout** (in seconds).
4. The resulting settings should look something like this:

SSH PROXY SETTINGS

Enable Proxy	Yes
Enable SSH Tunneling	No
Proxy New Secrets By Default	Yes
SSH Proxy Port	22
SSH Banner	Welcome to Secret Server SSH Proxy
SSH Proxy Host Fingerprint	SHA1 - 13:a8:b4:93:b9:17:11:02:18:5f:7c:5c:56:e9:3f:c7:c1:1d:9a:8e MD5 - d5:f8:7f:85:a0:db:d8:6d:b1:44:3c:20:10:bc:d7:66
Enable Inactivity Timeout	No

SSH TERMINAL SETTINGS

Enable Terminal	Yes
SSH Terminal Banner	===== WELCOME TO THYCOTIC TERMINAL (ThyTTY) =====

Edit

1. Specify the IP address for nodes (and engines) that will run SSH proxy:

1. Navigate to **Admin > Proxying > Nodes**.
2. Set the **SSH Public Host**. This is the public hostname or IP that the client launcher connects to. In most cases, this is the same as the SSH bind address; however, there are cases where the public IP or host differs from the private IP that Secret Server should bind to, such as NAT or Amazon EC2 instances.
3. Set the **SSH Bind IP Address**. This defaults to (0.0.0.0). The IP Address of the network adapter that the Secret Server SSH listener should bind to. This should not be localhost or 127.0.0.1. If you are not sure which bind IP address to use, you may use 0.0.0.0, which binds to all IPv4 interfaces on the machine.

Enabling Terminal on Secret Server Distributed Engine

SSH terminal can also run on each proxy-enabled distributed engine (DE) site.

Note: To launch secrets on non-local sites, users **must** connect to an SSH terminal over an engine on this site.

1. Go to **Admin > Proxying > Sites**.
2. Click to select **Proxy Enabled**.
3. Type an **SSH Port**.
4. Go to **Admin > SSH Proxy > Engines**.
5. Type the **Hostname** and **IP Address** (description above).
6. Type the **SSH Bind Address** (description above).

Logging into the SSH Terminal

1. From any SSH terminal, connect to hostname or IP address and port, as specified in the SSH Proxy Configuration page. Use the DE

hostname or IP if connecting to an engine. Examples:

```
ssh 127.0.0.1 -p 22
```

```
ssh user54@127.0.0.1 -p 22
```

2. If not provided in the SSH connect command, enter your Secret Server username and password at the **Login as:** prompt.
3. If successful, you will see the terminal banner displayed, along with a list of available commands.

Increasing Maximum Concurrent Logins for Users

Logging in to SSH terminal counts against the number of concurrent Secret Server sessions a user is allowed. For example, if **Maximum concurrent logins per user** is set to "1" and the user john.smith is logged into the Secret Server Web user interface, then john.smith logs into SSH terminal, his first Web session will end, and he will have to log in again to use the Web user interface.

To increase the maximum concurrent logins per user:

1. Go to **Admin > Configuration**. The Configuration page appears.
2. Click the **Login** tab.
3. Click the **Edit** button at the bottom of the page. The page becomes editable.
4. Click the **Maximum concurrent logins per user** dropdown list and select the desired number.
5. Click the **Save** button.

SSH Terminal Login with Two Factor Authentication

SSH terminal is considered a Web service and can be used with two factor authentication (2FA). To enable 2FA for terminal:

1. Follow the steps under [Two-Factor Authentication](#) to set up 2FA.
2. Go to **Admin > Configuration > Login > Require Two Factor for these Login Types** and select one of these:
 - o **Website and Web Service Login**
 - o **Web Service Log on Only**
3. Enable 2FA on the Secret Server user by going to **Admin > Users > Select a user > Edit > Two Factor** and select the 2FA option.

Note: FIDO2 authentication is not supported in this version of SSH terminal.

4. From any SSH terminal, connect to hostname or IP address and port, as specified in the SSH Proxy Configuration page. Use the distributed engine hostname or IP if connecting to an engine. Examples:

```
ssh 127.0.0.1 -p 22
```

```
ssh username@127.0.0.1 -p 22
```

5. If not provided in the SSH connect command, enter your Secret Server username and password at the **Login as:** prompt.
6. You will be prompted for a PIN or custom challenge message by your 2FA provider. Example:

```
login as: duouser <Enter>
```

Using keyboard-interactive authentication:

```
duouser@127.0.0.1's password: uewori#$$%tdtd <Enter>
```

Using keyboard-interactive authentication:

Pin Code: 3787 <Enter>

7. If successful, you will see the terminal banner displayed, along with a list of available commands.

Escaping Special Characters

When manipulating secrets containing special characters, such as single quotes and double quotes, you must escape those characters in the command.

Example: To search for an item with a space in the name, put the name in single or double quotes:

```
search "My Secret" Or search 'My Secret'
```

Example: To search for an item with a single quote embedded in the name, there are two options:

- Encase the term in double quotes:

```
search "Bob's Secret"
```

- Escape the single quote with a backslash:

```
search 'Bob\'s Secret'
```

Example: Similarly, to search for an item with a double quote embedded in the name, there are two options:

- Encase the term in single quotes:

```
search "'Weird' Secret"
```

- Escape the internal double quotes with a backslash:

```
search "\"Weird\" Secret"
```

Terminal Commands

man

Syntax

```
man [command name]
```

Description

Displays command help for specific or all commands. *Man* is short for *manual*.

Examples

```
man
```

Short help for all commands.

```
man cat
```

A detailed description of the `cat` command.

search

Syntax

```
search [-st] <search_text> [-f <folder_id>] [-fav] [-r] [-sf <search_field>] [-skip <skip_results>] [-s] [-t <secret_template_id>] [-take <max_results>]
```

Description

Returns a list of Secret Server secrets by keyword, which you can filter using several command-line switches.

Parameters

-st <search_text>

Required. Text to search for. -st is optional. Returns 25 results by default. Use -take to change from the default.

-f <folder_id>

ID of the secret folder to limit the search to.

-fav

Only search "favorite" secrets.

-r

Ignore restricted secrets in the search. Restricted secrets are included by default.

-s

Ignore subfolders in the search. Subfolders are included by default.

-sf <search_field>

ID of the secret field to limit the search to. Potential fields, which vary by secret template, can include the following examples:

- Address1
- Address2
- Address3
- Blog
- CardType
- City
- Combination
- Contact Number
- Country
- Email Address
- ExpirationDate
- Fax
- First Name
- FullName
- Home Phone
- Last Name
- Machine
- Mobile Phone
- Notes
- Number
- Password
- Pin
- PinCode
- Server

- SSN
- State
- Username
- Website
- Work Phone
- Zip

Note: These fields match those on the REST API endpoint.

`-skip <skip_results>`

Skip this number of initial results. Useful for processing "pages" of results.

`-t <secret_template_id>`

Only search secrets based on the template with this template ID.

`-take <max_results>`

Take a total of only this number of results. Useful for processing "pages" of results. Defaults to 25 results.

Examples

```
search -st admin
```

Find a list of secrets matching "admin." Returns 25 results (the default).

```
search admin
```

Same search using alternate syntax. `-st` is not required.

```
search -st jones -fav
```

Find a list of "favorite" secrets matching "jones" in any field Returns 25 results (the default).

```
search admin -take 50
```

Outputs a list of secrets matching "admin", up to 50 results.

```
search Zardoz -take 50 - skip 50 -sf "Secret Name"
```

Find a list of secrets with "Zardoz" in the "Secret Name" field. Return 50 results, starting with the 51st secret found.

```
search admin -skip 25 -r
```

Find a list of secrets matching "admin" in any field. Return 25 results, which is the default. Skip the first 25 results. Ignore restricted secrets.

cat

Syntax

```
cat [-sl-idl-secret-id] <secret_id> [-cl-comment <comment_or_access_request>] [-tl-ticket <ticket_number>] [-ticketsystemid <ticket_system_id>]
```

Description

- Displays information on a secret. The available information depends on the secret's template. *cat* is short for *concatenate*.
- Catches access errors, such as "comment required" or "requires approval", and displays them on the terminal
- Audits "view" comments.

- Provides launch connection command instructions. Shows the correct launch parameter and a connection string (if the terminal connection and the site on the secret do not match).

Note: If a required access element is not provided in the command, the terminal will respond with an error that should indicate what is missing.

Parameters

`[-sl-idl-secret-id] <secret_id>`

Required. The secret ID. Three optional switches.

`[-cl-comment <comment_or_access_request>]`

The text for the comment or access request.

`[-tl-ticket <ticket_number>]`

The ticket number for the request.

`[-ticketsystemid <ticket_system_id>]`

The unique ticket system ID.

Examples

```
cat 24
```

Display the contents of the secret with the ID 24. Only works after access is approved.

```
cat -id 24
```

Alternate syntax. Display the contents of the secret with the ID 24.

```
cat -id 25 -comment "Viewing this secret"
```

Add a "view" comment to, and then display the contents of the secret with the ID 25.

```
cat -id 26 -comment "Requesting view access to install software" -ticket 123 -ticketsystemid 2
```

Add an "access request" comment to the secret with the ID 26. Assign the request the ticket number 123 and the ticket system ID of 2 to that request.

Note: The most common secret restrictions are "requires view comment" or "requires access request." The `-comment` parameter takes care of both of these because the underlying API call (`SecretAccessCreateArgs`) is agnostic.

launch

Syntax

```
launch [-sl-idl-secret-id] <secret_id> [-ml-Machine <machine_name>] [-cl-comment <view_comment_or_approval_request_reason>] [-tl-ticket <ticket_number>] [-ticketsystemid <ticket_system_id>]
```

Description

- Creates a proxy connection to the machine
- Secret must have proxy enabled
- Supports launch from secrets with private keys
- Audits launches

Parameters

`[-sl-idl-secret-id] <secret_id>`

Required. The secret ID. Three optional switches.

`[-cl-comment <comment_or_access_request>]`

The text for the comment or approval request.

`[-ml-Machine <machine_name>]`

Machine name for the launch. This may be required if a customized secret template does not contain a machine field or a launcher requires a machine entry on launch.

`[-tl-ticket <ticket_number>]`

The ticket number for the request.

`[-ticketsystemid <ticket_system_id>]`

The unique ticket system ID.

Examples

`launch 24`

Begins the SSH proxy session with the secret with the ID 24 and the specified credentials and machine. Only works after access is approved.

`launch -id 24`

Alternate syntax. Begins the SSH proxy session with the secret with the ID 24 and the specified credentials and machine. Only works after access is approved.

`launch -id 25 -comment "Launching this secret"`

Submits a "view" comment to the secret with ID 25. Begins the SSH proxy session with secret credentials and machine.

`launch -id 26 -machine XYZ -comment "Requesting view to launch temporary sudo account for the XYZ machine"`

Submits an "access request" comment to the secret with ID 26 on the machine XYZ with the ticket number 123 and ticket system ID 2.

Launching a Secret with the SSH Terminal

Launching a Secret on a Local Site

1. To launch, the secret must be:
 - Enabled for proxy (**Secret Server > Secret > Security > Enable Proxy**)
 - Shared with the terminal user
2. Log in to the terminal with Secret Server user credentials:

```
login as: sshuser
Keyboard-interactive authentication prompts from server:
| sshuser@127.0.0.1's password:
| End of keyboard-interactive prompts from server
| Pre-authentication banner message from server:
| ===== WELCOME TO THYCOTIC TERMINAL (ThyTTY) =====
| End of banner message from server

Available Commands
-----
cat - concatenate Secrets and print on the standard output
launch - begin SSH Proxy session using credentials on specified Secret
man - an interface to the on-line reference manuals
search - search for Secrets by keyword
exit - exits this terminal session
[sshuser@127.0.0.1 ~] $
```

3. If the secret ID is unknown, search for the desired secret with the search command:

```
[sshuser@127.0.0.1 ~] $ search ubuntu
Folder Id      Template      Secret Name    Proxy Enabled
-----
Everyone Owns 68          Unix Account (SSH)  ubuntu\steph    True
Showing 1 - 1 of 1 results
[sshuser@127.0.0.1 ~] $
```

4. To view secret detail, get the secret ID from search results, and run

cat <secret_id>

```
[sshuser@127.0.0.1 ~] $ cat 68
Secret Name:  ubuntu\steph
Secret ID:    68
Secret Type:  Unix Account (SSH)
Folder:      \Everyone Owns
Launch Enabled: True
Machine:     [REDACTED].thycotic.com
Username:    steph
Password:    *****

Can be launched with command:

launch 68
[sshuser@127.0.0.1 ~] $
```

5. To launch the secret, enter the launch command as specified in the last line of secret details:

launch <secret_id>

```
[sshuser@127.0.0.1 ~] $ launch 68
Secret Server Launch: Secret ID 68 found. Attempting launch...
Connected to Target! Attempting Authentication...
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Jul 24 13:07:53 EDT 2019

System load:  0.0                Processes:            101
Usage of /:   40.2% of 14.58GB    Users logged in:     0
Memory usage: 22%                IP address for eth0: 192.168.60.252
Swap usage:   0%                IP address for docker0: 172.17.0.1

Graph this data and manage this system at:
https://landscape.canonical.com/

82 packages can be updated.
63 updates are security updates.

Last login: Wed Jul 24 12:59:59 2019 from 192.168.68.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

steph@ubuntu:~$
```

6. To exit the launch session and return to the terminal, type `exit`.

```
Last login: Wed Jul 24 12:59:59 2019 from 192.168.68.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

steph@ubuntu:~$ exit
logout
Socket was shutdown.
[sshuser@127.0.0.1 ~] $
```

7. To exit the terminal session, type `exit` again.

Launching a Secret on a Distributed Engine Site

1. To launch, the secret must be:
 - Enabled for proxy (**Secret Server > Secret > Security > Enable Proxy**)
 - Shared with the terminal user
2. Log in to the terminal with Secret Server user credentials:


```

login as: sshuser
Keyboard-interactive authentication prompts from server:
| sshuser@127.0.0.1's password:
| End of keyboard-interactive prompts from server
| Pre-authentication banner message from server:
| ===== WELCOME TO THYCOTIC TERMINAL (ThyTTY) =====
| End of banner message from server

Available Commands
-----
cat - concatenate Secrets and print on the standard output
launch - begin SSH Proxy session using credentials on specified Secret
man - an interface to the on-line reference manuals
search - search for Secrets by keyword
exit - exits this terminal session
[sshuser@127.0.0.1 ~] $

```

3. If secret ID is unknown, search for the desired secret with the search command:

```

[sshuser@127.0.0.1 ~] $ search ubuntu
Folder Id      Template      Secret Name    Proxy Enabled
-----
Everyone Owns 68      Unix Account (SSH)  ubuntu\steph   True
Showing 1 - 1 of 1 results
[sshuser@127.0.0.1 ~] $

```

4. To view secret detail, get the secret ID from search results, and run

cat <secret_id>

```

[sshuser@127.0.0.1 ~] $ search "engine site"
Folder Id      Template      Secret Name    Proxy Enabled
-----
Everyone Owns 69      Unix Account (SSH)  ubuntu\steph (Engine Site)  True
Showing 1 - 1 of 1 results
[sshuser@127.0.0.1 ~] $ cat 69
Secret Name:    ubuntu\steph (Engine Site)
Secret ID:      69
Secret Type:    Unix Account (SSH)
Folder:         \Everyone Owns
Launch Enabled: True
Site:           ihawu-mmq (ID: 9)

Active Proxy Engines:

Engine Host/Port: [REDACTED].thycotic.com:23

Machine:        [REDACTED].thycotic.com
Username:       steph
Password:       *****
Different Site: Unable to launch Secret '69' with current Terminal connection.
Exit and connect to Terminal host with the command below to launch.
-----
LAUNCH INSTRUCTIONS:
To launch this Secret on a different Site, SSH to the following Terminal host/port:

SSH Host: [REDACTED].thycotic.com
SSH Port: 23

Command:

ssh sshuser@[REDACTED].thycotic.com -p 23 -t launch 69

[sshuser@127.0.0.1 ~] $ █

```

5. Note that the connection is not made, and instructions are displayed for logging into another distributed engine terminal to launch the

secret.

- Note the suggested parameters in the **Launch Instructions**.
- Type `exit` and press **<Enter>** to disconnect from the current session:

```
Last login: Wed Jul 24 12:59:59 2019 from 192.168.68.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

steph@ubuntu:~$ exit
logout
Socket was shutdown.
[sshuser@127.0.0.1 ~] $ █
```

- Open a new SSH session suggested parameters:
`ssh <secret_server_username>@<engine_hostname_or_ip> -p <Port> -t launch <secret_id>`
- Enter the password to log in, and the secret should immediately launch.

Launching a Secret upon Terminal Connection

- To launch, the secret must be:
 - Enabled for proxy (**Secret Server > Secret > Security > Enable Proxy**)
 - Shared with the terminal user
- If the secret ID and connection string is known, you can log in and immediately launch the secret with the following command:
`ssh <secret_server_username>@<hostname_or_ip> -p <Port> -t launch <secret_id>`
- If you do not know that connection string, log into terminal and run:
`cat <secret_id>`
- Look at the **Launch Instructions** at the end of secret details, and note the parameters.

SSH Terminal Launching with a Custom SSH Command Allowlist

Secret Server terminal can launch secrets with custom SSH Command restrictions. For detailed instructions on SSH command menus, please see [Managing Superuser Privilege](#).

Note: Custom SSH command menus require either the Secret Server Platinum or Unix SUPM add on license.

- Go to **Admin > See All**. The Administration page appears.
- Click the **SSH Command Menus** link. The SSH Command Menus page appears.
- Click the **Create New** button.
- Type a name, description and the SSH commands:

New Command Menu ✕

Name *

Description

SSH Commands

1	view_shadow = cat /etc/shadow
2	view_secure_log = cat /var/log/secure
3	start_apache = /usr/sbin/service apache start
4	stop_apache = /usr/sbin/service apache stop

Once one or more command menus have been created, access can be controlled to individual Unix SSH secrets.

5. On the **Security** tab of a secret that can use a proxied SSH session, proxy must be enabled, as well as command menu restrictions. If **Allow Owners Unrestricted SSH Commands** is enabled, any user who is an owner of the secret has unrestricted use of the launched session. That is, that user is able to type in commands as in a normal SSH session. Additionally, other groups can be assigned the unrestricted role as well.
6. In the following example, the "admin" group is unrestricted, and everyone who is not in that admin group is restricted to only being able to run the allowlisted commands that are specified in the user command menu created above.

Terminal > Shared with Local User > ubuntu\ [redacted] ☆

General **Security** Audit Dependencies Sharing Settings

APPROVAL [Edit](#)

Require Approval	No	
-------------------------	----	--

OTHER SECURITY

Require Comment Users will be prompted for comment and ticket number when accessing a Secret.	No	Edit
Enable DoubleLock	No	
Enable Proxy	Yes	Edit
Hide Launcher Password	No	Edit
Restrict SSH Commands	Owners Unrestricted 2 items	Edit

When you click the Edit link:

Edit SSH Command Restriction

Restrict SSH Commands

Allow Owners Unrestricted SSH Commands

Unrestricted (1) ▼

Add User / Group: All ▼ Search for groups or use

admin	Remove
-------	------------------------

And click the dropdown list to select Allowlisted Commands:

Edit SSH Command Restriction

Restrict SSH Commands

Allow Owners Unrestricted SSH Commands

Whitelisted Commands (1) ▼

Add User / Group: All ▼ Search for groups or use Q

Everyone	Remove
----------	------------------------

7. A user who is subject to SSH command restrictions is presented with a screen similar to the following when launching this secret from Secret Server terminal:

```
Using username "729ddaef-38d0-48e0-b9dc-d4911e76d0c1".

1. User Command Menu

    ?. Show Command Menus
    exit. Exit session
Last login: Thu Mar 17 12:38:46 2016 from 192.168.60.153
[runscripts@centostestserver ~]$ █
```

The user simply enters the number of the command menu to see available commands or types "?" to display the options again:

```
1. User Command Menu

    ?. Show Command Menus
    exit. Exit session
Last login: Thu Mar 17 12:38:46 2016 from 192.168.60.153
[runscripts@centostestserver ~]$ 1

1. view_shadow = cat /etc/shadow
2. view_secure_log = cat /var/log/secure
3. start_apache = /usr/sbin/service apache start
4. stop_apache = /usr/sbin/service apache stop

    up. Return to Command Menu selection. You may also type ..
    ?. Show Commands
    exit. Exit session
[runscripts@centostestserver ~]$ █
```

Only the commands listed can be run by this user. The user can either enter the number of the command to be run, or the name of the command, which is the word to the left of the equal = sign. Other options are available (as shown) to navigate through the available command menus, display help, or exit the session.

SSH Terminal Launching with Session Recording

Secret Server SSH terminal launches also support session recording for session client or server data. When a user launches a secret with

session recording enabled through SSH terminal, session data is available in the Secret Audit tab as session data.

Note: Session recording requires either Secret Server Platinum or the session recording add-on license.

Note: See the Session Recording section for more information.

To enable session recording:

1. Go to **Admin > Configuration**. The Configuration page appears.
2. Click the **Session Recording** tab.
3. Click the **Edit** button at the bottom of the page. The page becomes editable.
4. Ensure the **Enable Session Recording** check box is selected.
5. Modify other settings as desired.
6. Click the **Save** button.

To enable session recording on a secret:

1. Open a secret.
2. Click the **Security** tab.
3. Click the **Edit** link to the right of the **Session Recording Enabled** setting. The Edit Security popup page appears.
4. Click to select the **Session Recording Enabled** check box.
5. Click the **Save** button.

To view session data following a terminal secret launch:

1. Open a secret.
2. Click the **Audit** tab.
3. Find the **LAUNCH** action in the table.
4. Click the **View SSH Session Log** link.

SSH Key Pairs for Terminal

Overview

SSH key pairs allow users to authenticate to Secret Server terminal without using a password. The user generates a key pair in Secret Server, at which time the private key can be downloaded by the user locally in the format they require. The key pair generation process is the only time the private key will be provided to the user. If this private key is lost, the user must log back into Secret Server and generate a new public/private key pair.

Limitations

- Currently users can only authenticate to Secret Server using SSH keys by using Secret Server's SSH terminal.
- Only PuTTY and OpenSSH keys can be generated.

Enabling Users to use SSH Key Pairs to Authenticate

There are three requirements for enabling Public SSH Keys:

- SSH Proxy is enabled in Secret Server.
 - SSH Terminal is enabled in SS.
 - SSH key integration is enabled in SS's Configuration > Login settings. To do so:
 1. **Unix Authentication Method**: choose **Public Key only**, **Password or Public Key** or **Password and Public Key** to enable SSH key pair authentication.
1. Once done, the admin can also set an optional expiration time frame for the public SSH keys, which applies to all users.

Once these 3 requirements have been met, users can use the main navigation to create SSH key pairs.

Creating SSH Key Pairs

An SSH key pair consists of a private key and a public key. Only the public key is stored in the user's settings—the private key downloaded during generation is **not** saved inside Secret Server and should only be available to the user, to remain secure.

During terminal login, if the user provides a private key for authentication, Secret Server validates the provided private key against the user's available (and enabled) saved public keys. If a key pair match is found, the authentication succeeds (or the next required authentication step, for example a password prompt, is shown).

For security reasons, only users can create their own SSH key pairs. However, Secret Server Administrators can deactivate any user's public SSH keys as follows:

1. Navigate to the **Public SSH Keys** page using the main navigation at the top right of the page.
2. Click the **Create SSH Key** button above the grid, then fill out the form in the popup page.
3. Click the **Create SSH Key** button in the popup. After a moment you will be able to save the private key.

Administering Public SSH Keys

1. Navigate to the User by going to **Admin > Users**.
2. Locate the user in the dropdown list and select it.
3. On the **General** tab click the **Administer Public SSH Keys** button. You can now deactivate the user's public SSH keys.

Using SSH Keys for Authentication (PuTTY Example)

1. In PuTTY, fill in the **Session** view to match your SSH proxy connection settings in Secret Server.
2. In the **SSH > Auth** section of PuTTY, add the private key file that was saved when generating the key in Secret Server.
3. You will be prompted to enter your passphrase for the key if one was set.
4. You will be prompted to enter your password if **Unix Authentication Method** also requires a password.

Remote Password Changing

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Remote Password Changing (RPC) allows secrets to automatically update a corresponding remote account. You can set secrets for automatic expiration, followed by automatic strong password generation and a remote password update to keep the subject accounts synchronized with Secret Server.

RPC allows Secret Server to rotate passwords to meet domain password policy requirements. In most cases, RPC is configured with the secret "auto change" setting set to true. This causes the secret to rotate the password as soon as it expires. The "auto change schedule" setting changes the password on a set schedule, rather than when it expires. This provides the ability to change passwords when network activity is lower. You have a choice of changing the password as soon as the schedule interval arrives or only after the secret expires *and* the interval arrives. It is important to choose a large enough interval to complete all your password changes, otherwise any excess changes will have to wait for the next interval. Because the smallest interval is one day, this is only relevant if you have thousands of changes. If Secret Server fails to change a remote password, an alert states there are secrets out of sync.

You can pair secrets with Secret Server checkout, which is Delinea's one-time password functionality (not the same as [TOTP](#)). This allows you to rotate the password on a particular expiration schedule and limit the password to a single user for a set time period, after which it is changed. This is for situations where you need the password to change after every use, such as vendors who need temporary access to a server or system. For additional security on sensitive systems, approval workflow or session recording can be paired with checkout to add layers of authentication to gain access to the secret and track how that secret is used.

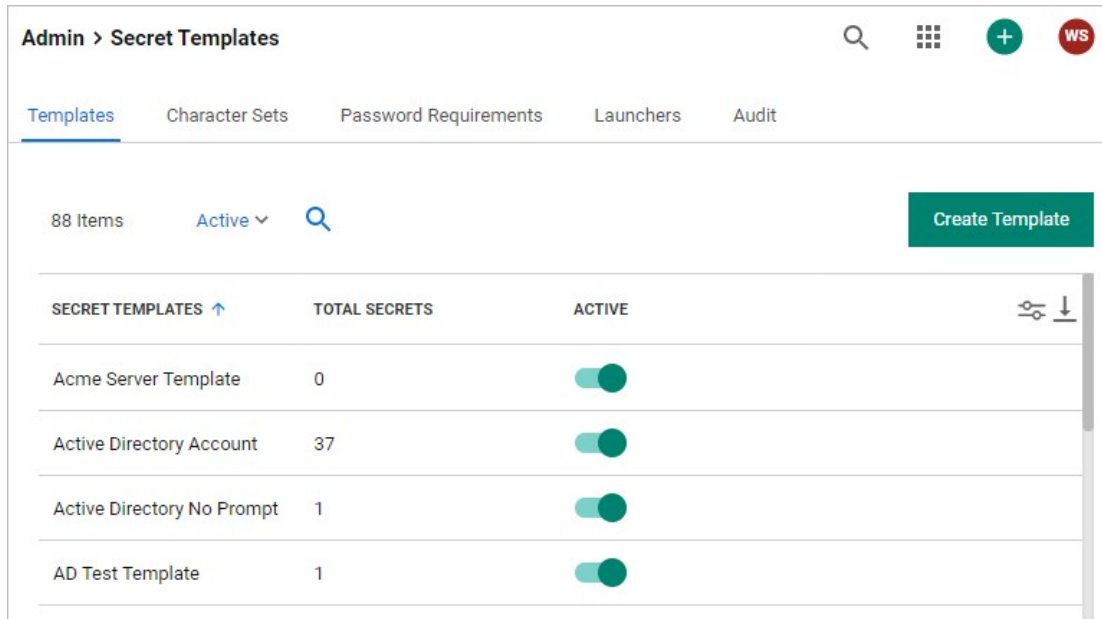
Regardless of the timing of password change, you may want to rotate dependent resources (dependencies) right after you rotate the password on a secret. For example, a Windows domain account could be a service account that starts many windows services. In the event that you rotate that password, you would need to also rotate the password for this account on the services which start using that account. If you do not, starting those services will fail the next time they are restarted, which could cause other components to fail too. You can create dependencies on a secret for scheduled tasks, application pools, or services (with or without using PowerShell to undertake tasks at rotation time).

We have a large number of out-of-the-box RPC changers, which are expandable by writing your own SSH, SQL or PowerShell scripts to do RPC, which can get information from the secret. See [Configuring Secret Dependencies for RPC](#) and the [Password Changer List](#).

Note: You can configure [event pipelines](#) and [notifications](#) to track whether an RPC has failed. Heartbeats allow you to check whether a password is incorrect and the machine is online.

To assign any type of password changer to a Secret template, use the procedure below.

1. From the **Admin** menu, select **Secret Templates**. The Secret Templates page appears:



The screenshot shows the 'Admin > Secret Templates' page. At the top, there is a navigation bar with 'Admin > Secret Templates', a search icon, a grid icon, a green plus icon, and a red 'WS' icon. Below the navigation bar are tabs for 'Templates', 'Character Sets', 'Password Requirements', 'Launchers', and 'Audit'. The 'Templates' tab is selected. Below the tabs, there is a header section with '88 Items', a dropdown menu set to 'Active', a search icon, and a green 'Create Template' button. The main content is a table with the following columns: 'SECRET TEMPLATES' (with an upward arrow), 'TOTAL SECRETS', and 'ACTIVE' (with a toggle switch icon). The table contains four rows of data:

SECRET TEMPLATES ↑	TOTAL SECRETS	ACTIVE
Acme Server Template	0	<input checked="" type="checkbox"/>
Active Directory Account	37	<input checked="" type="checkbox"/>
Active Directory No Prompt	1	<input checked="" type="checkbox"/>
AD Test Template	1	<input checked="" type="checkbox"/>

2. Click the name of the desired template in the list. The page for that template appears:

Admin > Secret Templates > Acme Server Template

General Fields Mapping Permissions Audit

Export Duplicate

Template Status [Edit](#)

Indicates how many Secrets use this template and whether or not the template is active.

[Secret Templates](#)

Template Usage 0

Active Yes

Template Settings [Edit](#)

Configure settings for this template.

Secret Template Name * Acme Server Template

Name Pattern None

Description None

All History No

Secret Name History Length * 0

Validate Password Requirements On Create No

Validate Password Requirements On Edit No

3. Click the **Mapping** tab:

Admin > Secret Templates > Acme Server Template

General Fields **Mapping** Permissions Audit

Export Duplicate

Scan Templates **Add Mapping**

Password Changing [Edit](#)

Enables heartbeat, remote password changing, and configures / maps the password changer type and fields.

Enable RPC No

Enable Heartbeat No

4. Click the **Edit** link next to the **Password Changing** section. The section becomes editable:

Scan Templates Add Mapping

Password Changing

Enables heartbeat, remote password changing, and configures / maps the password changer type and fields.

Enable RPC

Enable Heartbeat

Cancel Save

5. Click to select the **Enable RPC** check box. An editable section appears:

Note: This enables the feature, but auto RPC is set through policies or secret settings.

Enable RPC

RPC Max Attempts

RPC Interval Days

RPC Interval Hours

RPC Interval Minutes

Enable Heartbeat

Password Type to use

Cancel Save

6. Click the **Password Type to Use** dropdown list to select the desired type. For this instruction, we chose Active Directory Account. New controls appear for that specific type:

Enable RPC	<input checked="" type="checkbox"/>
RPC Max Attempts	<input type="text" value="10000"/>
RPC Interval Days	<input type="text" value="0"/>
RPC Interval Hours	<input type="text" value="1"/>
RPC Interval Minutes	<input type="text" value="0"/>
Enable Heartbeat	<input type="checkbox"/>
Password Type to use	<input type="text" value="Active Directory Account"/>
Default Privileged Account	No Secret Selected
Domain	<input type="text" value="Search or pick one"/>
Password	<input type="text" value="Search or pick one"/>
User Name	<input type="text" value="Search or pick one"/>
Domain Controller (DC)	<input type="text" value="<blank>"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

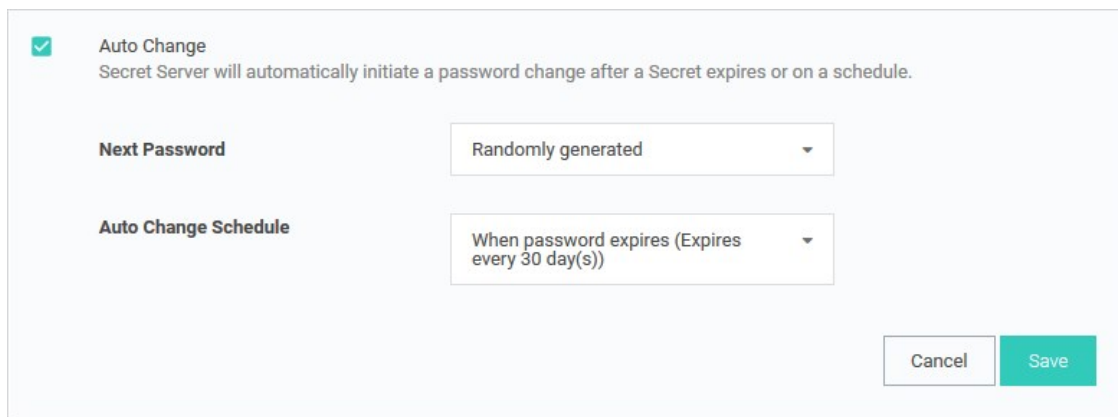
7. Click to select or type text for each item to configure your desired setup. For more on configuring your choice, refer to the applicable configuration in the [RPC](#) section for details. For example, for Active Directory, go to [Minimum Permissions for Active Directory Remote Password Changing](#).
8. When finished, click the **Save** button.

The Remote Password Changing tab contains the settings for configuring RPC on an individual secret. Enabling RPC *auto change* on a secret allows Secret Server to remotely change the password when it expires. The user must have owner permission on the secret to enable auto change.

Note: If the password change fails, Secret Server flags the secret as out of sync and continue to retry until it is successful. If the secret cannot be corrected or brought In sync, manually disabling auto change stops the secret from being retried.

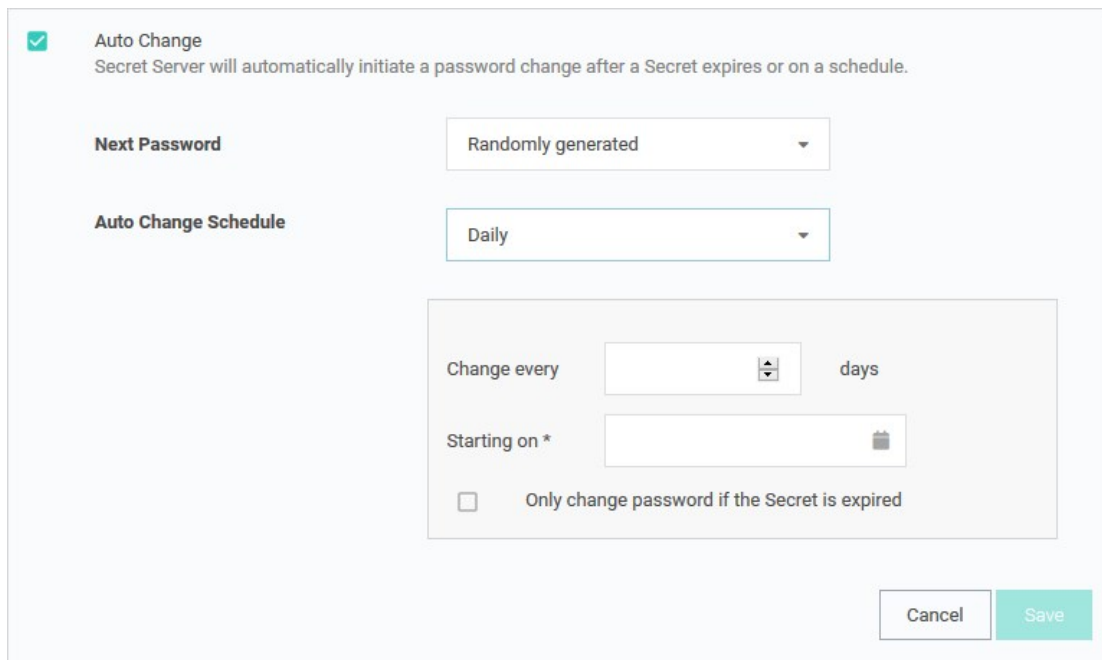
Auto Change Schedule

The Auto Change Schedule button is visible on the secret View RPC tab when RPC and autochange is enabled on a secret.



The screenshot shows a dialog box with a checked checkbox for "Auto Change". Below it is a description: "Secret Server will automatically initiate a password change after a Secret expires or on a schedule." There are two dropdown menus: "Next Password" set to "Randomly generated" and "Auto Change Schedule" set to "When password expires (Expires every 30 day(s))". At the bottom right are "Cancel" and "Save" buttons.

The Auto Change Schedule section, which appears when you set the Auto Change Schedule list box to other than "When password expires," allows you to specify an interval (daily, weekly, or monthly), start date, start time, and time frame (interval count) for when the password can be changed:



This screenshot shows the same dialog box as above, but with the "Auto Change Schedule" dropdown set to "Daily". This has expanded the settings to include a "Change every" field with a spinner set to 1, followed by "days". Below that is a "Starting on *" field with a calendar icon. At the bottom of this section is a checkbox labeled "Only change password if the Secret is expired" which is currently unchecked. The "Cancel" and "Save" buttons are at the bottom right.

This setting is useful for having the RPC occur during off-hours in order to prevent disruptions. By default, this setting is "When password expires," which allows the secret to be changed immediately upon expiration.

Note: There is a check box in the auto change schedule settings labeled "Only change password if the secret is expired." When it is enabled, auto change will not change the password until after the secret expires. The auto change occurs on the first scheduled time after the secret expires. If the box is unchecked, auto change occurs on the defined schedule, whether or not the secret has expired.

Note: While the password change is waiting for this next scheduled time, the RPC Log (visible by navigating to **Configuration > Remote Password Changing**) reports the secret could not be changed because of a time schedule. The secret also remains expired until this auto change schedule is reached, even if the secret was forced to expire.

Understanding Expiration, Auto Change and Auto Change Schedules

Definition

What is the difference between expiration, auto change and auto change schedules?

- **Expiration:** sets whether or not a secret in Secret Server is marked as expired and the period Secret Server counts down before marking the secret as expired.
- **Auto Change:** sets Secret Server to automatically initiate a password change after a secret expires.
- **Auto Change Schedule:** sets the day and time to initiate the password change after the secret has expired. This cannot be configured without also enabling Auto Change.

Examples

Some examples to illustrate this:

Scenario One: Expiration with Auto Change and No Auto Change Schedule

- A Secret has an expiration period of 30 days, and auto change is enabled. No auto change Schedule has been set.
- At the end of the 30-day expiration period, the secret will expire.
- Immediately after the secret expires, it will be queued for a password change.
- Once the password has been changed, the secret is no longer marked as expired and expiration is reset to count down again from 30 days.

Scenario Two: Expiration with Weekly Auto Change

- A secret has an expiration period of 30 days, auto change is enabled, and an auto change schedule is configured for Weekly, recurring once a week on on Tuesday, changing at 0300.
- At the end of the 30-day expiration period, the secret will expire.
- Immediately after the secret expires, Secret Server will comply with the auto change schedule to determine when a password change occurs.
- The secret is queued for a password change as soon as it becomes 0300 on a Tuesday.
- Once the password is changed, the secret is no longer marked as expired. Expiration is reset to count down again from 30 days.

Scenario Three: Expiration with No Auto Change

- A Secret has an expiration period of 30 days, and auto change is not enabled.
- At the end of the 30-day expiration period, the secret expires.
- The secret remains expired until the field it applies to (usually the password field) is updated on the secret. This happens by manually updating the field or by using the "Change Password Remotely" button on the Remote Password Changing tab of the secret.
- Once the password is changed, the secret is no longer be marked expired, and expiration is reset to count down again from 30 days.

Important Considerations and Best Practices

- If you want to rely strictly on expiration for password changing, enable auto change but set the schedule to none. Leave "Only change password when Secret is expired" checked.
- If you want to set an auto change schedule to run daily at a specific time, the change will only happen at maximum once per day at that given time. If a change happens already within that same day for the same secret, you cannot adjust the auto change schedule to run later within the same day and then have a password change occur again within that same 24-hour period. For example, if the password was already changed earlier in the day. The schedule is then adjusted to run a few minutes later within the same day. In that case, another password change will not occur until 24 hours has passed since the last change.
- If you set the auto-change schedule to run once per week, for example, on a Thursday, and "Only change password when secret is expired" is checked. Even if the secret expires on a Monday, a password change would not occur until the secret has expired and the scheduled time on Thursday has passed.
- If you set the auto change schedule to run once per week on a Thursday and "only change password when Secret is expired" is not checked, the password would be changed every Thursday, regardless of the secret's expiration status.
- If a secret has an expiration period but auto change is not enabled, no password change occurs automatically. The expiration would only update when the password is manually updated or a remote password change is manually triggered through Secret Server.
- If you want to change a password more frequently than once per day, we recommend using some of the advanced security features at the secret level or controlling the change through a secret policy. Use the check out feature combined with "Change Password on Check In" on the Security tab of a Secret. You can specify a custom interval to check out the secret. After the password check out interval expires or a user manually checks in the secret, the password is automatically changed.

Important: For the configuration above, ensure that these accounts have a password-related group policy in Active Directory that specifies that the "Minimum Password Age" is set to 0. We recommend creating fine-grained password policies to achieve this. Add all the accounts that need rotation more frequently than once per day to an AD security group assigned to the fine-grained password policy. See [Password and account lockout policies on Azure Active Directory Domain Services managed domains](#) for more information.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret dependencies are items that rely on the username, password, or SSH private key stored in the secret. By adding them to the Dependencies tab, they are automatically updated when the secret's password is changed, ensuring they are up to date with the account on which they depend.

☰
?
👤

Secret Dependency Templates Designer

[Explain](#)

+ Create New Dependency Template
Filter by Dependency Type All ▼

DEPENDENCY TEMPLATE NAME	DEPENDENCY TYPE	ACTIVE	OPTIONS
Application Pool	Application Pool	Yes	
Application Pool Recycle	Application Pool Recycle	Yes	
COM+ Application	COM+ Application	Yes	
Remote File	Remote File (Regex Replace)	Yes	
Scheduled Task	Scheduled Task	Yes	
Windows Service	Windows Service	Yes	
SSH Key Rotation	SSH Script	Yes	
SSH Key Rotation Privileged	SSH Script	Yes	
ind	PowerShell Script	Yes	

Show Inactive

Back

Configure Dependency Changers

Adding a custom dependency template may require additional settings (these settings are described in the following section):

COM+ Dependency Scanner

The COM+ Dependency Scanner allows for an Active Directory domain discovery source to locate COM+ Applications running on machines on the domain that are being run by Domain Accounts.

Firewall concerns may be addressed by ensuring that Port 135 is open between the target machine being scanned and the machine that engine is installed on.

Requirements for Discovery

Windows Services

For all supported versions of Windows and Windows Server, ensure that **Remote Procedure Call (RPC)** and **Remote Procedure Call (RPC) Locator** services are running. To help prevent any errors that would stop the services, set the **Startup Type** to **Automatic**.

Component Services

For all supported versions of Windows and Windows Server, ensure that **NETWORK** has remote access permissions to the machine.

1. Open **Component Services** (dcomcnfg.exe).
2. Under **Console Root**, expand **Component Services** and the **Computers** folder.
3. Right-click **My Computer**
4. Select **Properties**.
5. Under the **Default Properties** tab, ensure that the **Default Authentication Level** is set to **Connect** and that **Default Impersonation Level** is set to **Identify**.
6. On the **COM Security** tab, for both the **Access Permissions** and **Launch and Activation Permissions** sections, click **Edit Limits** and then add **NETWORK**.
7. Check **Allow** for all Remote permissions.

Note: If the **Edit Limits** button is disabled, open the **Local Security Policy**. Under **Security Settings** expand **Local Policies** and select **Security Options**. There will be two **DCOM: Machine Access/Launch Restrictions**. Edit the one that corresponds to the disabled **Edit Limits** buttons, adding **NETWORK** and giving Remote permissions there.

Important: Editing or altering the existing permissions on the machine or editing the **Default Permissions** listed can have a negative impact on the machine.

COM+ Network Access

For all supported versions of Windows Server, ensure that COM+ Network Access is enabled by installing the Application Server Role. During the installation process, check the box next to **COM+ Network Access** under **Features**.

Versions Supported

- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008

- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Versions Not Supported

- Windows Vista and earlier versions of Windows
- Versions of Windows Server pre-2008
- Windows Server 2016

Configuring COM+ Discovery for a New Domain

1. Navigate to **Admin > Discovery**
2. Click **Edit Discovery Sources**.
3. Click **Create New**.
4. Select **Active Directory Discovery Source** and click **OK**.
5. In the wizard, click **Next**.
6. Select a **Site** that is set up with **Distributed Engine**
7. Click **Next**.

Note: The COM+ Dependency Scanner will only run when a Distributed Engine Site is applied to the Discovery Source. The Engine will need to be installed either on the Domain to be scanned, a Child Domain relative to the Domain being scanned, a Parent Domain relative to the Domain being scanned, or another Trusted Domain relative to the Domain being scanned.

8. Check the box next to **COM+ Application**.
9. Click **Next**.
10. Enter your **Fully Qualified Domain Name**.
11. Select or create a Secret for an Active Directory account that will scan for your COM+ dependencies.
12. Click **Next**.

Your new domain is now configured in Secret Server and Discovery will search for COM+ dependencies in it.

Configuring COM+ Discovery for an Existing Domain

1. Navigate to **Admin > Discovery**.
2. Click **Edit Discovery Sources**.
3. Click on the domain where you wish to search for COM+ dependencies.
4. Click the **Scanner Settings** tab.
5. Scroll down to the **Find Dependencies** section and click **Add New Dependency Scanner**.

6. Click the plus symbol to the left of **COM+ Application**.

You will be unable to make additional changes.

7. Click **OK** to proceed. Discovery will now search for COM+ dependencies.

Creating Custom Dependencies

If there are different dependency types that you want to manage that are not supported out of the box, new ones can be created based on a script. A custom dependency consists of two components:

- **Dependency Template:** The dependency template defines how a dependency is matched to discovered accounts and how it updates the target after a password change occurs on the account. To create a new dependency template, go to **Admin > Secret Templates** and click the **Dependency Templates** button.
- **Dependency Changer:** A dependency changer is a script and the associated parameters to be passed into the script. Dependency changers can be created and modified by going to **Admin > Remote Password Changing > Configure Dependency Changers**.

Note: Please see [Discovery](#) for comprehensive guidance on configuring and using dependency changers and dependency templates.

Dependency Groups

By default, all dependencies are updated in the order listed. There are cases where you may want to split out different sets of dependencies into separate groups. Typically, this is because a single service account may run services across different segregated networks that can communicate with the domain but not each other and have different distributed engine sites assigned. In this case you can create two dependency groups and assign them to different distributed engine sites to solve connectivity issues.

Dependency Settings and Information




Dependencies have the following settings:

Note: Not all dependency types have all these settings.

- **Change Fail Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that determines if Secret Server was unable to update the public key on the dependency.
- **Change Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that updates the public key on the dependency.
- **Change Success Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that determines if Secret Server was able to update the public key on the dependency.
- **Database:** For SQL script dependency types, the database name for the script.
- **Dependency Group:** Name of the group to run the dependency update in.
- **Description:** Description of the dependency for documentation purposes.
- **Enabled:** Whether Secret Server attempts to update the dependency. A disabled dependency is ignored by Secret Server.
- **File Path:** For Remote File Dependency types, this is the UNC file path on the remote server where the embedded password exists.
- **Machine Name:** Computer name or IP address on which the dependency is located.
- **Name:** Name of the dependency on the remote machine.
- **Port:** For SQL and SSH script dependency types, the port name for the script.
- **Privileged Account:** The account Secret Server authenticates as when changing the dependency's credentials, so it must have privileges on the remote machine to edit the dependency.
- **Public Key:** For SSH key rotation and SSH key rotation privileged dependency types, this text-entry field holds the value of the public key stored on the dependency.
- **Regex:** For Remote File Dependency types, the regular expression used to locate the password embedded in the configuration file.
- **Restart:** Determines if the dependency is restarted once the account has been updated.
- **Run Condition:** Allows the dependency to run conditionally depending on the outcome of the dependencies above it.
- **Script:** Name of the PowerShell script, SSH script, or SQL script in the scripts repository configured on the Dependency Template. The actual script selected can be previewed by clicking the eye icon.
- **Server Key Digest:** For SSH key rotation and SSH key rotation privileged dependency types, a text-entry field that serves as a security control for specifying the SHA1 hash of the SSH host key on the remote server.
- **Server Name:** For SQL script dependency types, the server name for the script.
- **SSH Key Secret:** An account with SSH Key that Secret Server uses to authenticate when executing the SSH Script or SSH Key rotation dependency types.
- **Template:** Whether the dependency is an IIS application pool, Scheduled Task, windows service, remote file, COM+ application. Custom dependencies can also be created using a SQL, SSH, or PowerShell script.
- **Verification Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that verifies that the new public key on the dependency matches the private key on the secret.
- **Wait(s):** Time in seconds that Secret Server pauses before changing the dependency.

Example values for a Windows service dependency on a remote computer might be: 192.11.158.99, Windows Service, aspnet_state, OR DOMAIN\admin.

The following operations can be performed in the Dependency grid:

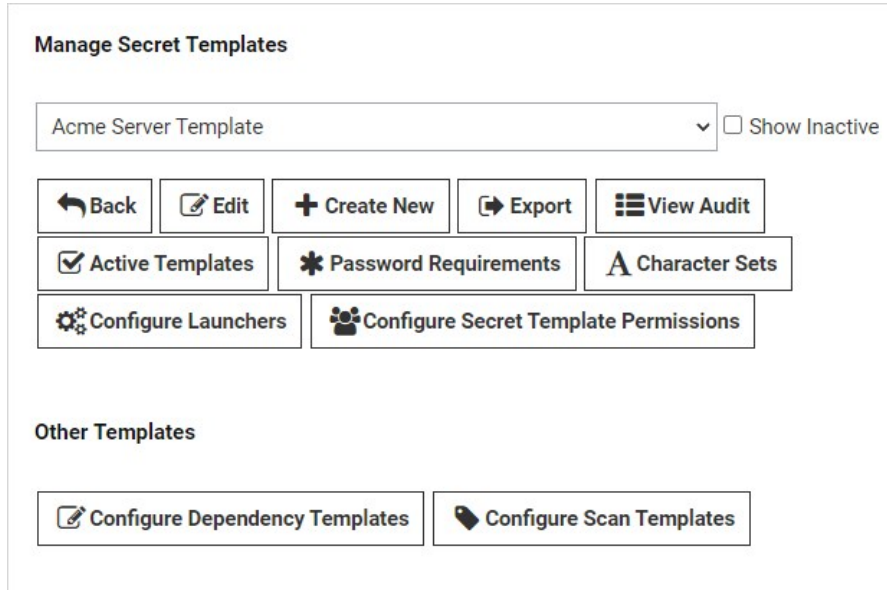
- **Delete:** Click the  icon to delete the dependency.
- **Edit:** Click the  icon to edit dependency text boxes. Cancel changes by pressing the Cancel button.
- **Run Dependency:** Click the second arrow icon to run the script on the selected dependency and set the password on the selected dependency to the current password for the secret
- **Test Connection:** Click the return arrow icon to test the dependency connection, the tests results are displayed afterward.
- **View Dependency History:** Click the  icon to view the activity logs for the dependency.

Note: Due to security constraints, scheduled tasks require an Active Directory domain user as the privileged account.

Manually Adding Dependencies

To manually add a dependency:

1. Click **Admin > Secret Templates**. The Manage Secret Templates page appears:

















2. Click the **Configure Dependency Templates** button. The Secret Dependency Templates Designer page appears:

Secret Dependency Templates Designer

[Explain](#)

+ Create New Dependency Template **Filter by Dependency Type** All ▼

DEPENDENCY TEMPLATE NAME	DEPENDENCY TYPE	ACTIVE	OPTIONS
Application Pool	 Application Pool	Yes	
Application Pool Recycle	 Application Pool Recycle	Yes	
COM+ Application	 COM+ Application	Yes	
Remote File	 Remote File (Regex Replace)	Yes	
Scheduled Task	 Scheduled Task	Yes	
Windows Service	 Windows Service	Yes	
SSH Key Rotation	 SSH Script	Yes	
SSH Key Rotation Privileged	 SSH Script	Yes	
PowerShell Multi-RPC	 PowerShell Script	Yes	 
SQL Service Dependency Changer	 PowerShell Script	Yes	 

Show Inactive

[← Back](#) [⚙️ Configure Dependency Changers](#)

- Click on the plus icon next to **Create New Dependency Template**. The New Dependency Template popup appears:

New Dependency Template ✕

Type: Powershell Script ▼

Scan Template: < Select a scan template > ▼ *

Dependency Changer: < Select a dependency changer > ▼

Name: *

Description:

Enabled:

4. Choose your dependency type from the **Type** dropdown list.
5. Click to select your desired scan template from the **Scan Template** dropdown list.
6. Click to select your desired dependency changer from the **Dependency Changer** dropdown list.
7. Fill in the dependency name, machine name, and other information depending on the dependency type.
8. Ensure the **Enabled** check box is selected.
9. Click the **OK** button to finish adding the dependency.

Secret Dependency Status

You can see a list and status of all dependencies for a secret when viewing that secret in the UI. For example:

Secret Dependencies > Local > Remote File Dependency\RegTestUser ☆

General Security Audit Remote Password Changing **Dependencies** Sharing Settings Metadata

Launchers ▾ Options ▾

This Secret is currently marked as Inactive. Certain actions such as password changing will be unavailable until the secret is activated. ✕

Find New Dependencies Dependency Templates **New Dependency**

▼ Group: RegTestUser Group, Site: Local (3 Total Enabled) 2 Not Run 1 Failed

☐ 3 Items All Templates ▾ All Run Results ▾ 🔍 Include Disabled

ORDER	TITLE	TYPE	MACHINE	ENABLED	RUN RESULT
1	c:\dependency\depe...	Remote File (Regex R...	jumphost.gamma.thy...	✔	✘
2	c:\dependency\depe...	Remote File (Regex R...	jumphost.gamma.thy...	✔	⚠
3	c:\dependency\depe...	Remote File (Regex R...	jumphost.gamma.thy...	✔	⚠

Account Dependence

You can use a single account (username and credential) for OS login and also for running Windows services, scheduled tasks, IIS application pools, and more. This is especially common for functional accounts. You can link the credentials of one account object to such usages. The link is independent of the actual system that the dependency is running on.

For example, given this scenario:

- The Windows domain account DOM\svc_app1 is managed on windows domain controller, where it is located.
- SRV1 is running a Windows service under the user DOM\svc_app1.
- SRV2 is running a scheduled task under the user DOM\svc_app1.
- When the account DOM\svc_app1 is changed, the dependencies of that account on SRV1 and SRV2 need to be updated too.
- The Windows service on SRV1 may have to be restarted for the password change to work.

The modeling of the dependency would take place in / with the master object, which in this case is the windows domain account DOM\svc_app1. When selecting the master account object in the UI, it shows the dependencies along with their status.

Account Clusters

You can group different account objects in clusters where the account name may be different but the credential and password are the same for all members of the cluster. Account clusters can also support the account dependency functionality from above. For example:

For example, given this scenario:

- Windows domain account DOM\svc_app1 has dependencies as outlined above.
- Linux OS login svc_app2 on LinSRV1 runs a daemon local to that system.
- Database login svc_db1 on DBSRV1 is consumed by a billing application.
- A dependency exists where the account's password can also be pushed into the configuration file of an application running as user

svc_app2 on LinSRV1.

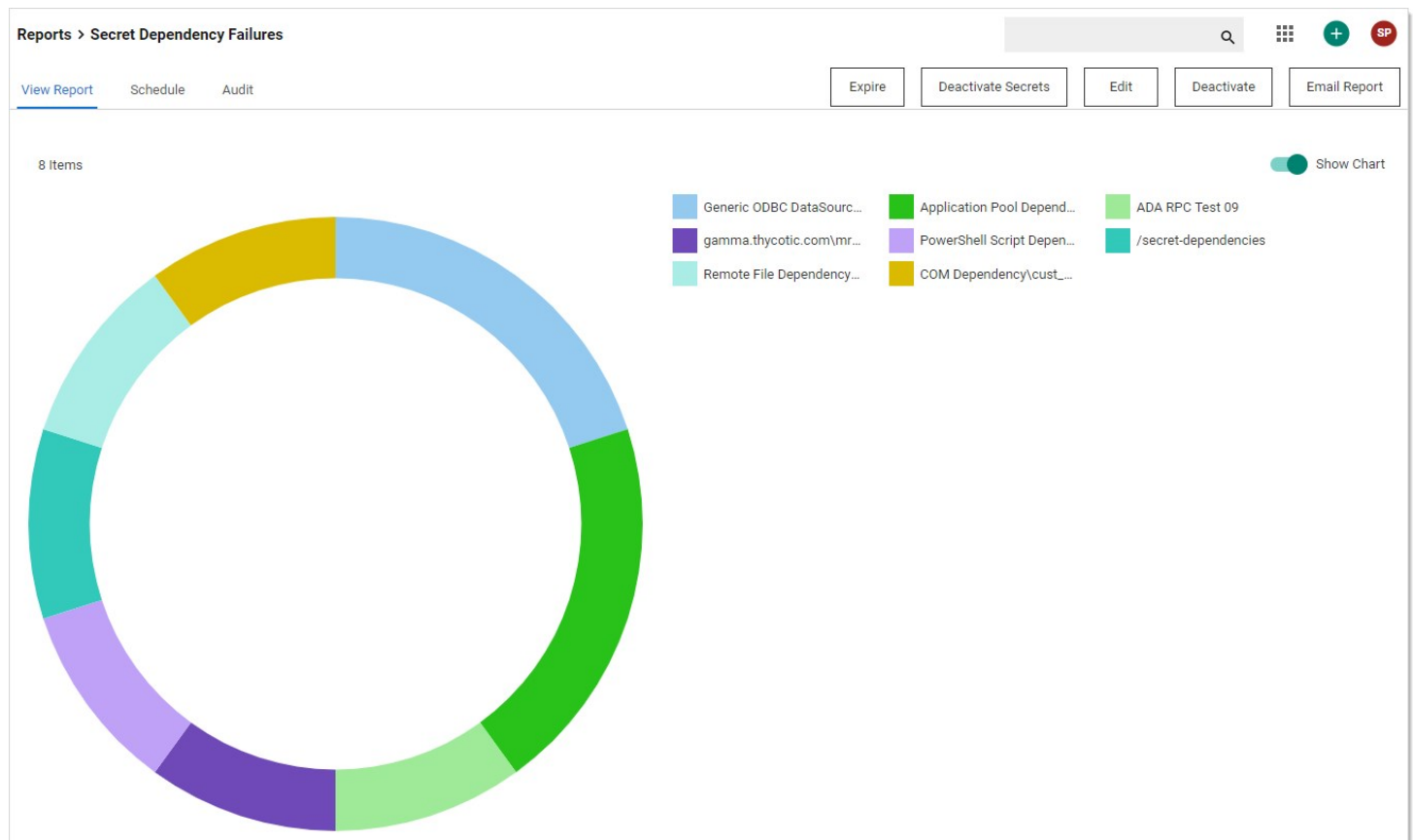
Accounts DOM\svc_app1, svc_app2 and svc_db1 would all have the same password when the password is rotated. The system can re-sync passwords of all members of the account cluster in case one of the members runs out of sync for whatever reason, such as restoring from a backup.

Viewing Dependency Status

We offer four reports for viewing your secret dependency status:

Secret Dependency Failures

Figure: Secret Dependency Failures Report



Secret Dependency Not Run

Figure: Secret Dependency Not Run Report

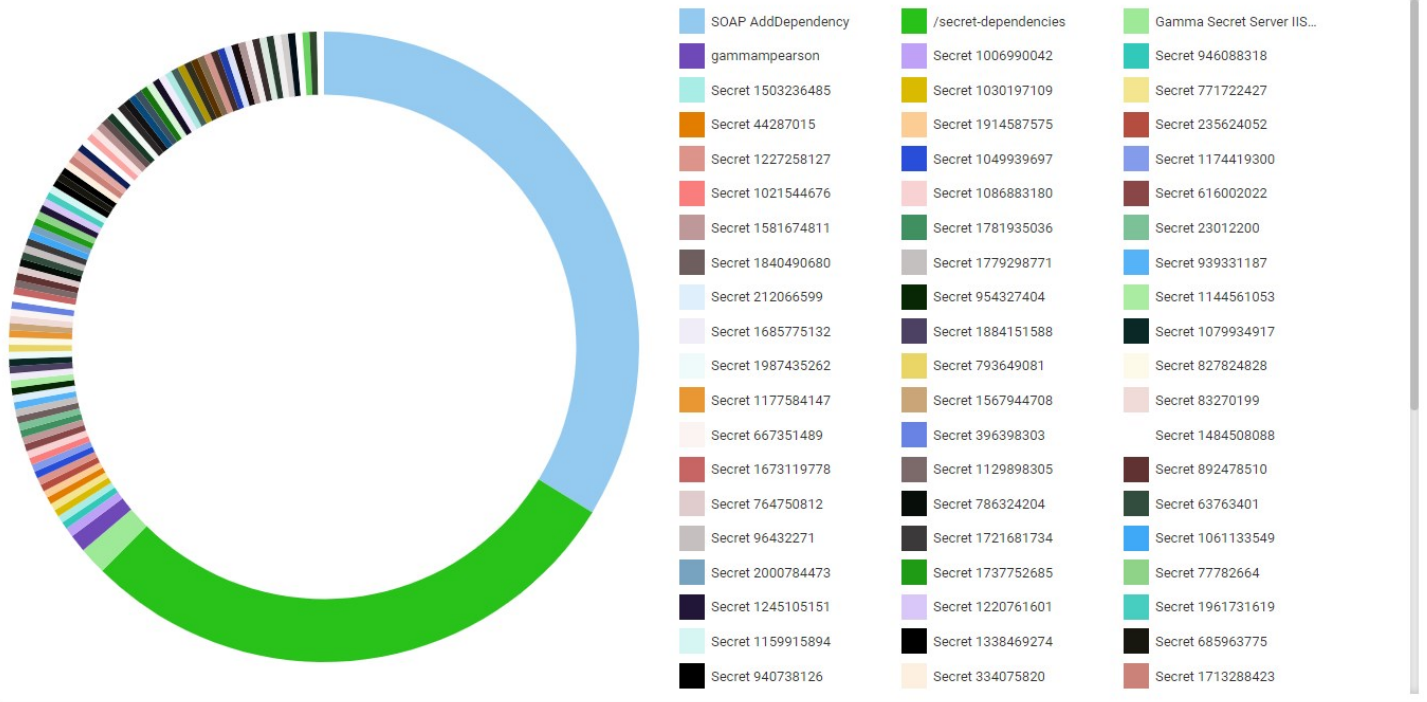
Reports > Secret Dependency Not Run

[View Report](#) [Schedule](#) [Audit](#)

[Expire](#) [Deactivate Secrets](#) [Edit](#) [Deactivate](#) [Email Report](#)

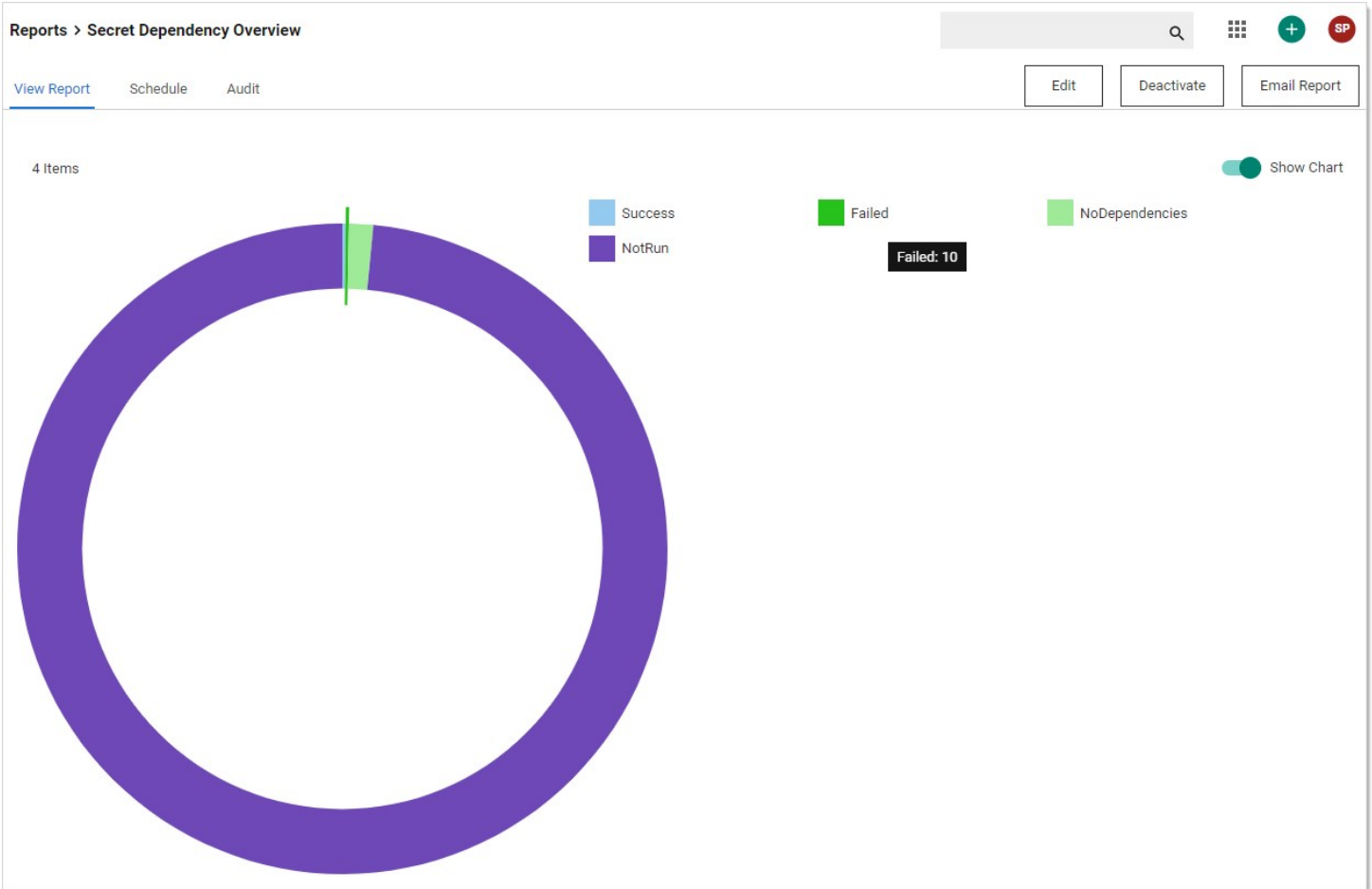
599 Items

Show Chart



Secret Dependency Overview

Figure: Secret Dependency Overview Report



Secret Dependency Status

Figure: Secret Dependency Status Report

Reports > Secret Dependency Status

View Report Schedule Audit Expire Deactivate Secrets Edit Deactivate Email Report

615 Items

SECRETNAME	DEPENDENCYGRO...	SITENAME	SUCCESS	FAILED	NOTRUN
\$domain\launcher	group		0	0	1
/secret-dependencies	Dependency Gr...		0	1	696
/secret-dependencies	Dependency Gr...	Gamma-Engines	0	0	8

Using Regex with Dependencies

Overview

In release version 7.8.00010 and later, Secret Server allows a secret to have file dependencies. File dependencies allow text files with embedded credentials to be changed via Regex.

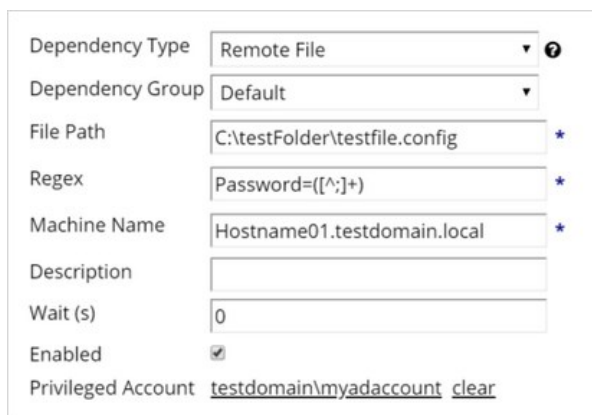
A Regular Expression (Regex) is a phrase in a language for matching text. For details on the .NET Regex language, see [.NET Framework Regular Expressions](#).

Secret Server replaces the contents of the first group (within parentheses) within the Regex.

Setting up a remote file dependency, requires:

- **File Path:** This is the file path on the remote server where the remote password exists. UNC paths do not work here. See [UNC Names](#).
- **Regex:** This regular expression to be used to locate the password embedded in the configuration file.
- **Machine Name:** Computer name or IP address where the dependency is located.
- **Privileged Account:** The account Secret Server will authenticate as when changing the dependency. It must have privileges on the remote machine.

A typical filled in New Dependency page looks something like this:



Dependency Type	Remote File
Dependency Group	Default
File Path	C:\testFolder\testfile.config
Regex	Password=([^\;]+)
Machine Name	Hostname01.testdomain.local
Description	
Wait (s)	0
Enabled	<input checked="" type="checkbox"/>
Privileged Account	testdomain\myadaccount clear

UNC Names

UNC names, such as:

\\BARAKA\SHARE\test.txt OR

\\192.168.1.154\SHARE\test.txt

do **not** work in the file path. You can, however, put the machine name or IP address in the Machine Name text box, and put the rest of the path in the file path. For example:

In the **File Path** text box:

\SHARE\test.txt OR

SHARE\test.txt

In the **Machine Name** text box:

192.168.1.154 OR

BARAKA

Examples

The following are some examples of using Regex within file dependencies:

XML Configuration Files

Example One

Source

```
<Configuration>
  <User>
    <UserName>Bob</UserName>
    <Password>Password1</Password>
  </User>
  <User>
    <UserName>Sam</UserName>
    <Password>DontChangeThisOne</Password>
  </User>
</Configuration>
```

Regex

```
<UserName>Bob</UserName>\s*<Password>([^\<]+)</Password>
```

Example Two

Source

```
<Configuration>
  <User name="Bob" password="Password1" />
  <User name="John" password="Password1" />
</Configuration>
```

Regex

```
<User name="Bob" password="([^\"]+)" />
```

Windows Initialization (.ini) Files

Source

```
[owner]
name=John Doe
password=Password1
organization=Acme Widgets Inc.
```

Regex

```
name=John\sDoe\s*password=([^\r\n]+)
```

SQL Server Connection Strings

Source

```
Data Source=myServerAddress;Initial
Catalog=myDataBase;UserId=myUsername;Password=myPassword;Server=myServerAddress;Database=myDataBase;Trusted_Connection=False;
```

Regex

Password=([^\;]+)

Oracle Connection Strings

Example One

Source

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=MyHost)(PORT=MyPort)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));
```

```
User Id=myUsername;Password=myPassword;
```

Regex

Password=([^\;]+)

Example Two

Source

```
Data Source=username/password@//myserver:1521/[my.service.com](http://my.service.com);
```

Regex

username/([^\@/]+)

YAML

Source

```
receipt: Oz-Ware Purchase Invoice
```

```
date: 2007-08-06
```

```
user:
```

```
name: Dorothy
```

```
password: Password1
```

Regex

```
name:\s*Dorothy\s*password:\s*([^\r\n]+)
```

In most environments, we recommend using a separate password for each account for optimal security. However in environments where identical credentials are used in multiple secrets, we recommend using RPC to change the password on one primary parent account secret, and then using a PowerShell dependency script to update values in child secrets. The PowerShell script calls back to Secret Server's API, retrieves a list of comma-separated values representing child secret IDs, and updates the values stored in the child secrets. We recommend using this process for no more than 25 child secrets.

Requirements

- A Secret Server instance version 10.1.000000 or newer with a premium add-on or Enterprise Plus
- A PowerShell implementation enabled and working properly. See [Configuring WinRM for PowerShell](#)
- the [WellnessChecker tool](#)


For this procedure you will need to create the four types of user accounts listed below, and for each account you will need to create a corresponding secret in Secret Server with the account's login credentials and other information.

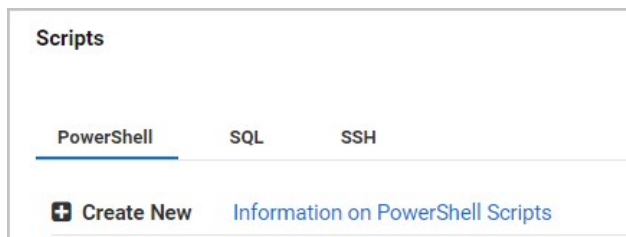
Create the user accounts and secrets described below:

- An API User account and a corresponding secret. This API User account will NOT take up a user license. Recommended templates for the secret include the Active Directory template and the Web Password template. Credentials may be a local account or an Active Directory service account assigned to the Synchronization group, but must be stored in Secret Server to be passed to the PowerShell script.
- A primary parent account and a corresponding secret that has RPC set up and the PowerShell dependency script from this page attached. The primary parent account credentials may be either a local account or an Active Directory service account assigned to the Synchronization group.
- Child accounts with a corresponding secret for each account containing the child secret ID, with edit permissions granted to the API User account.
- A privileged Active Directory account and a corresponding secret that can run PowerShell on the Secret Server machine.

To create a new dependency changer for synchronizing passwords during RPC, follow the procedure below.

1. Download the [WellnessChecker tool](#) ZIP file.
2. Extract the ZIP file and run this command:

```
PowerShell.WellnessChecker.exe -fixerrors
```
3. In Secret Server, browse to **Admin > Scripts**.
4. Click the  symbol next to **Create New**.



5. In the **New PowerShell Script** dialog, fill in the fields for **Name**, **Description**, and **Category**.
6. In the **Script** field, paste in the script provided at the bottom of this page.
7. Click **OK** to save the file.

New PowerShell Script ✕

Name !

Description *

Category Untyped ▼

Script

1		

8. Browse to **Admin > Configuration**.
9. On the **General** tab, make sure **Enable Webservices** is set to **Yes**.

Configuration

General
Login
SAML
Folders
Local User Passwords
Sec

APPLICATION SETTINGS

Allow Automatic Checks for Software Updates	Yes
Early Adopter	No
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	No
View Webservices	
Enable Webservices	Yes

10. Browse to the primary parent account secret and ensure that RPC is setup on it. See [link-to-page](#).
11. In the primary parent account secret, click the **RPC** tab.

12. Click **Edit**.
13. In the secret grid at the bottom, select the API User account secret you created. The API User account secret should be the only secret in the grid. If you have not yet created the PowerShell script, you will see no grid.
14. Browse to **Admin > Discovery** and click the **Configuration** tab.
15. Click **Discovery Configuration Options** and select **Extensible Discovery** from the drop-down list.

Admin > Discovery

Discovery Sources **Configuration** Discovery Logs Computer Scan Logs

Discovery [Edit](#)

Discovery is used to scan for machines, local accounts and dependencies on Active Directory, Unix systems, and VMware ESX servers, AWS, and GCP.

Discovery is easy to set up and provides a great range of customizations for specific network requirements.
[KB Link](#)

Discovery Configuration Options ▾

- Rules
- Extensible Discovery**
- Scanners
- Scan Templates
- Command Sets
- Domain Name Index

Enable Discovery

Discovery Interval Days

Discovery Interval Hours

Ignore Cluster Node Objects No

Discovery Scan Offset Hours 0

Days to Keep Operational Logs 30

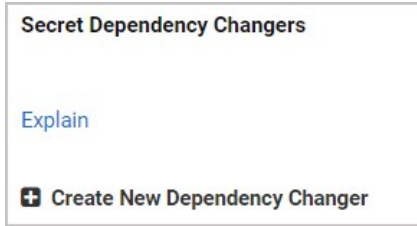
16. On the **Extensible Discovery Configuration** page, click **Configure Dependency Changers**.

Dependency Changers

Dependency Changers define the type of dependency and method for changing the dependency's password by utilizing information from the Secret and the the Scan Template. Dependency Changers are either built-in or Scriptable via PowerShell, SSH or SQL.

Configure Dependency Changers

17. On the **Secret Dependency Changers** page, click **Create New Dependency Changer**.



18. In the **New Dependency Changer** dialog, click the **Basic** tab and enter the following information:

New Dependency Changer

Basic Scripts

Explain

Type Powershell Script

Scan Template Computer Dependency (Basic)

Name PowerShell Multi-RPC

Description

Port

Wait (s) 0

Enabled

Create Template

Save Cancel

19. Click the **Scripts** tab and enter the following information:

New Dependency Changer

Basic **Scripts**

Explain
 Use advanced scripts ?

▼ Change Script RPC - Child Secrets ✓

Script: RPC - Child Secrets *

Arguments: `[1]$USERNAME [1]$PASSWORD $PASSWORD $NOTES [1]$DOMAIN`

✓ Save ✕ Cancel

20. In the **Arguments** field, paste the following:

```
[1]$USERNAME [1]$PASSWORD $PASSWORD $NOTES [1]$DOMAIN
```

The actions of the Arguments are as follows:

- `[1]$USERNAME` pulls the username from the first linked secret on the primary parent account, which will be used to execute the PowerShell script.
- `[1]$PASSWORD` pulls the password from the first linked secret on the primary parent account, which will be used to execute the PowerShell script.
- `$PASSWORD` pulls the password from the primary parent account, which will be set for all secrets listed in the **Notes** field.
- `$NOTES` pulls the **Notes** content from the primary parent account, and parses the comma separated list of secret IDs to find the other secrets to update.
- `[1]$DOMAIN` pulls the **Domain** field from the first linked secret on the primary parent account. For local accounts, leave the **Domain** field on the linked secret empty. It must be listed last because of the way PowerShell parses empty fields.

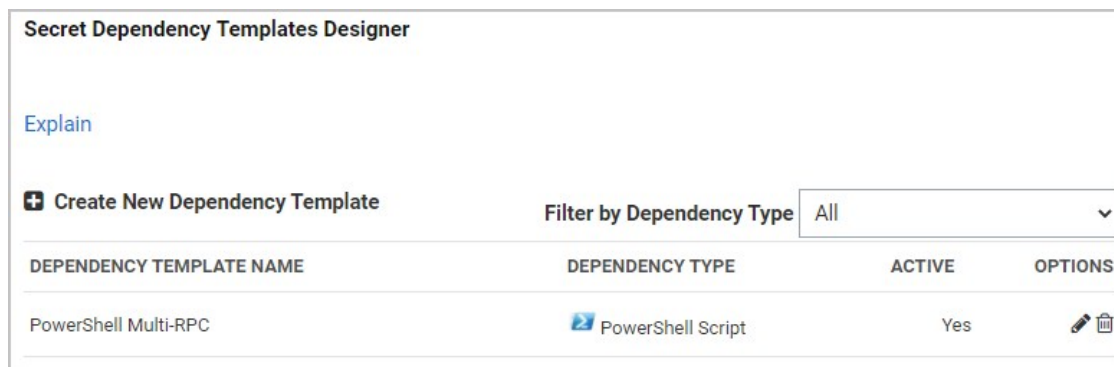
21. Browse back to the **Extensible Discovery Configuration** page and this time, click **Configure Dependency Templates**.

Dependency Templates

Dependency Templates link Scan Templates to Dependency Changers. Every Dependency Changer requires a Dependency Template.

[✎ Configure Dependency Templates](#)

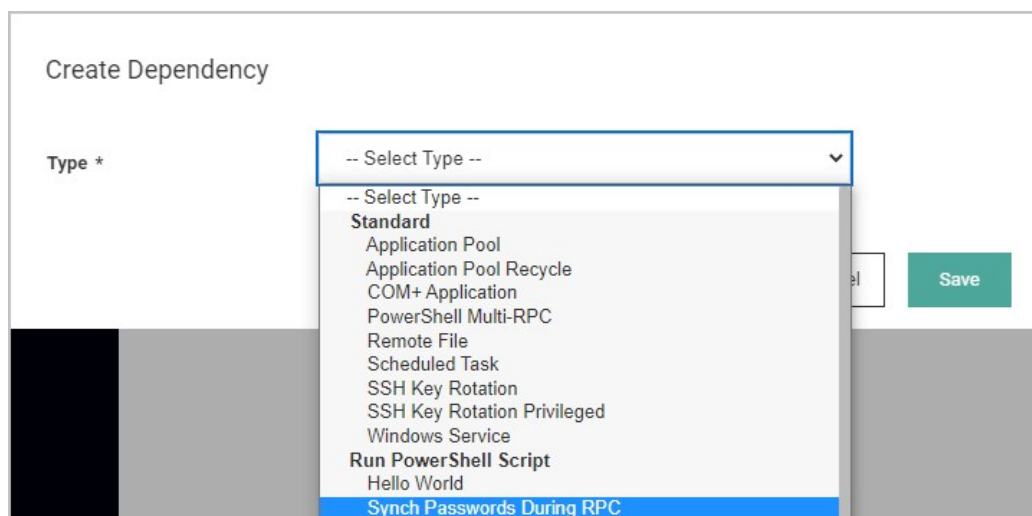
22. On the **Secret Dependency Templates Designer** page, select the new dependency changer you configured in the last step.



- Browse to the primary parent account secret and click the **Dependencies** tab.
- Click **New Dependency**.



- In the **Create Dependency** dialog, click the **Select Type** dropdown and select the PowerShell dependency template you created.



- In the **Edit Dependency** dialog, enter default in the **Machine Name** field.
- Select a privileged account (active directory account secret able to run PowerShell on the server)
- Add the following to the **Arguments** field: `${1}$USERNAME ${1}$PASSWORD $PASSWORD $NOTES ${1}$DOMAIN`
- In the primary parent account secret's **Notes** field, ensure that the child secret IDs appear in a comma-separated-values list, for example 19,39,81...

Now the dependency has been added and you can test the full process by running a remote password change on the primary parent account. All of the secrets listed by ID in the **Notes** field should be updated with the same password.

PowerShell Script

Replace \$url with the name of the machine hosting your Secret Server instance.

```
$url = 'http://MySecretServerURL/webservices/sswebservice.asmx';
$username = $Args[0]
$password = $Args[1]
$newpassword = $Args[2]
$secretIdArray = $Args[3]
$domain = $Args[4]
$proxy = New-WebServiceProxy -uri $url -UseDefaultCredential
$result1 = $proxy.Authenticate($username, $password, "", $domain)
if ($result1.Errors.Length -gt 0){
    $errors = $result1.Errors[0]
    Write-Debug "Errors result1: $errors"
    exit
} else {
    $token = $result1.Token
}
$secretIds = $secretIdArray -split ","
foreach($secretId in $secretIds){
    $result2 = $proxy.GetSecret($token, $secretId, $false, $null)
    if ($result2.Errors.Length -gt 0){
        $errors = $result2.Errors[0]
        Write-Debug "Errors result2: $errors"
    } else {
        $secretName = $result2.Secret.Name
        Write-Debug "Updating Secret: $secretName"
        foreach ($item in $result2.Secret.Items) {
            if($item.IsPassword) {
                $item.Value = $newpassword
            }
        }
        $secret = $result2.Secret
        $result3 = $proxy.UpdateSecret($token, $secret)
        if ($result3.Errors.Length -gt 0) {
            $errors = $result3.Errors[0]
            Write-Debug "Errors result3: $errors"
        } else {
            Write-Debug "Updated Secret: $secretName"
        }
    }
}
}
```


The Password Changers Configuration page can be accessed by navigating to **Admin > Remote Password Changing > Configure Password Changers**.

There are a few password changing types that allow the user to enter in specific commands that are sent to the computer where the password is changing. This enables the system to accommodate for differences in the standard password change procedure. For example: The Unix system that is being changed prompts for the current password twice instead of only once before asking for the new password.

Changing Ports and Line Endings

To change the port or line ending used on a password changer, click the password changer on the **Configure Password Changers** page and then click **Edit**. There, you can choose the line ending and port used by the device. By default, line endings are set to New Line (\n), however some devices and applications (such as HP iLO) use a different line ending system. The port defaults to 22 for SSH connections and 23 for Telnet connections.

For the built in Windows password changer there is a ports text-entry field available that can be filled in to help ensure a computer is listening. This can be used if DNS returns multiple IP addresses for a single box and only one is valid. For example, a laptop might get two IP addresses for an Ethernet and wireless connection, but if it is unplugged the Ethernet IP is invalid. In this case, Secret Server can do a reverse lookup and test each IP until it is able to connect on one of the specified ports. When it gets a response, it uses that IP for the password change.

Creating a Custom Password Changer for IBM AS/400

Note: Password changing on the IBM AS/400 can be performed through SSH, which is installed by default. If you are using an earlier version, you will need to install SSH.

To create a custom password changer for IBM AS/400 on newer systems such as i7, use the procedure for [Creating a Custom Password Changer](#) but be sure to use the following SSH command:

- Command: `system CHGUSRPRF $USERNAME PASSWORD($NEWPASSWORD)`
- Comment: Set Password on account
- Pause(ms): 2000

For additional information, see [Securing Communications with OpenSSH on IBM i5/OS](#).

Creating a Custom Password Changer for IBM AS/400 in Secret Server 10.5.

The procedure for creating password changers in Secret Server 10.5 for the IBM AS/400 terminal includes using the 5250 terminal connection and scripting to perform the password change and heartbeat.

To create this IBM AS/400 password changer, start with an existing z/OS Mainframe password changer as a baseline, then modify the changer commands. You also need to create an AS/400 secret template using the z/OS secret template as a baseline, then modify the template to use the new password changer.

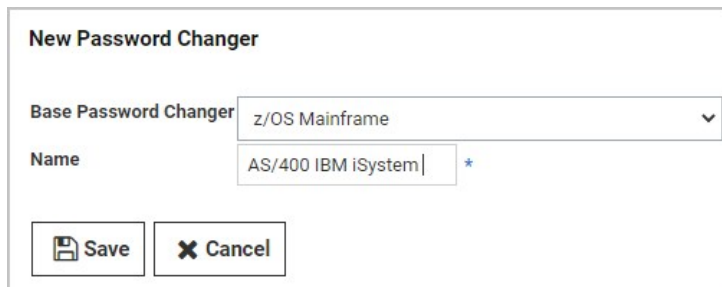
Note: Support for this feature, including script customization for advanced requirements, is available only through professional services.

Configuration

Follow the procedure below, in the sequence presented.

Create an AS/400 password changer from an existing z/OS Mainframe password changer:

1. Browse to **Admin > Remote Password Changing > Configure Password Changers**.
2. Scroll to the bottom and select **New**.
3. For the **Base Password Changer**, select the **z/OS Mainframe**.
4. In the **Name** field, enter AS/400 IBM iSystem



The screenshot shows a dialog box titled "New Password Changer". It has two main input fields: "Base Password Changer" which is a dropdown menu currently showing "z/OS Mainframe", and "Name" which is a text input field containing "AS/400 IBM iSystem" followed by an asterisk. At the bottom of the dialog, there are two buttons: "Save" with a floppy disk icon and "Cancel" with an 'X' icon.

5. Click **Save**.

Modify the AS/400 IBM iSystem password changer commands:

To add custom password changer commands to the AS/400, you must replace the existing standard z/OS mainframe command set.

1. Browse to **Admin > Remote Password Changing > Configure Password Changers**.
2. Click the **AS/400 IBM iSystem** password changer you just created.
3. On the **AS/400 IBM iSystem** page, scroll to the bottom and click the **Edit Commands** button. The commands that appear initially on the **Verify Password Changed Commands** page represent the standard z/OS Mainframe command set. You can use these commands as a baseline but you must customize them to suit your environment.

Verify Password Changed Commands

ORDER	COMMAND	COMMENT	PAUSE(MS)	
1	LOGON \$USERNAME/\$CURRENTPASSWORD NORECONNECT	Logon	2000	
2	<ENTER>	Enter	2000	
3	\$\$CHECKCONTAINS READY	Ready For Next Input	4000	
4	LOGOFF	##SESSIONLOGOFF	2000	
	<input type="text"/>	<input type="text"/>	<input type="text" value="2000"/>	

Password Change Commands

ORDER	COMMAND	COMMENT	PAUSE(MS)	
1	LOGON \${1}\$USERNAME/\${1}\$PASSWORD NORECONNECT	Logon	2000	
2	<ENTER>	Enter	2000	
3	\$\$CHECKCONTAINS READY	Ready For Next Input	4000	
4	<CLEAR>	Clear	2000	
5	alu \$USERNAME password(\$NEWPASSWORD) resume noexpire	Change Password	2000	
6	\$\$CHECKCONTAINS READY	Ready For Next Input	2000	
7	LOGOFF	##SESSIONLOGOFF	2000	
	<input type="text"/>	<input type="text"/>	<input type="text" value="2000"/>	

[Advanced Settings](#)

Back
 Configure Scan Template
 View Audit

4. Click the **Back** button when you have finished customizing your password changer commands, to return to the **AS/400 IBM iSystem** password changer page.

Modify the AS/400 password changer for 5250 emulation and commands:

1. On the **AS/400 IBM iSystem** page, scroll to the bottom and click the **Edit** button.
2. On the **Edit Password Changer** page, check the box next to **Use SSL** (recommended).
3. Set the **Custom Port** to 992.

Edit Password Changer

Name * AS/400 IBM iSystem

Line Ending New Line (\n)

Custom Port 992 (e.g. override the default value of 22 for SSH or 23 for Telnet with another value)

Request Terminal (If checked, the standard out and standard error data streams combine for \$\$CHECK* cor.

Connection String

Use SSL

Ignore SSL Verification

Active

Valid for Discovery Import

For extra troubleshooting assistance, you can add TRACE to the connection string to have a trace file written to the Secret Server website or engine.

Create an AS/400 template from the z/OS Secret Template:

1. Browse to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, select **z/OS Mainframe** from the drop-down menu and click **Edit**.
3. On the **Secret Template Designer** page, scroll to the bottom and click **Copy Secret Template**.
4. On the **Name New Secret Template** page, enter AS/400 IBM iSystem in the **Name** field.

Name new Secret Template x

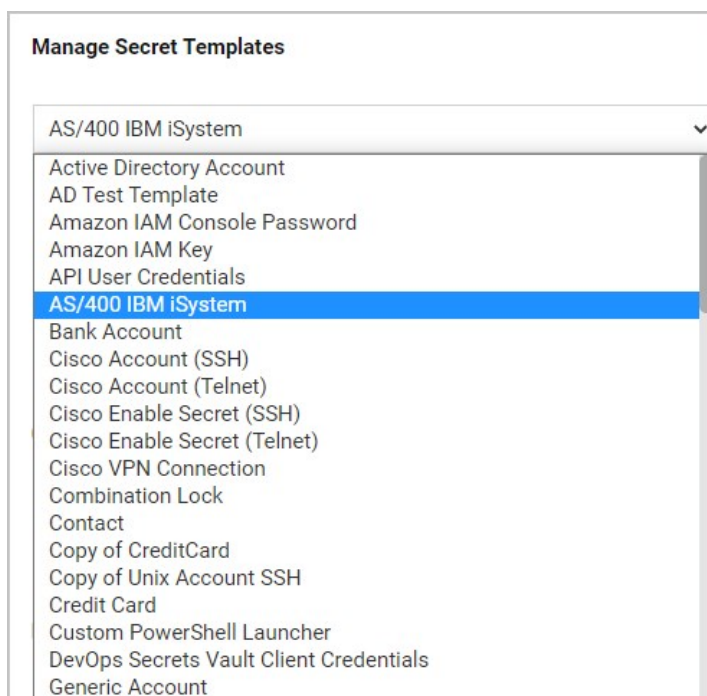
Name: AS/400 IBM iSystem

5. Click **OK**.
6. On the confirmation screen, click **Continue**.



Modify the AS/400 Secret Template to use the AS/400 Password Changer:

1. Browse to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, click the drop-down menu and select the new **AS/400 IBM iSystem** secret template you just created.





3. Click **Edit**.
4. On the **Secret Template Designer** page, scroll to the bottom and click **Configure Password Changing**.
5. On the **Secret Template Edit Password Changing** page, click **Edit**.

Secret Template Edit Password Changing

Enable Remote Password Changing Yes
Retry Interval 2 hours
Maximum Attempts 12
Enable Heartbeat Yes
Heartbeat Check Interval 1 day

Password Type to use z/OS Mainframe

PASSWORD TYPE	SECRET FIELD	SCRIPT VARIABLE
Machine Name	Machine	\$machine
Passphrase	Passphrase	\$passphrase
Password	Password	\$password
Port	Port	\$port
User Name	Username	\$username

- On the second **Secret Template Edit Password Changing** page, select **AS/400 IBM iSystem** from the **Password Type to Use** drop-down menu.

Secret Template Edit Password Changing

Enable Remote Password Changing

Retry Interval

Days

Hours

Minutes

Maximum Attempts

Enable Heartbeat

Heartbeat Check Interval

Days

Hours

Minutes

Password Type to use

Machine Name

Passphrase

Password

Port

User Name

7. Click **Save**.

You can now create secrets using your new template and password changer in Secret Server 10.5.

If you would like to test your template and password changer but you do not have access to an AS/400 IBM iSystem, you can use the website PUB400.com.

Creating a Custom Password Changer

To create a custom password changer, follow the procedure below.




Note: Before creating the password changer and attempting to change a password through Secret Server, we recommend that you test the command set you are using directly.

1. From the **Admin** menu, select **Remote Password Changing**.
2. Click **Configure Password Changers**, then scroll to the bottom of the page and click **+New**.

Remote Password Changing Configuration

Enable Remote Password Changing	Yes
Enable Password Changing on Check In	No
Enable Heartbeat	Yes

[Advanced \(not required\)](#)



  






3. Choose a **Base Password Changer** with a command set that most closely matches the type of password changer you are creating, as this determines which customizable parameters and test actions are available to you. To create a custom SSH password changer, choose a base password changer with a name that ends in (SSH).

New Password Changer

Base Password Changer

Name *

4. Give your new password changer a **Name** and click **Save**.
5. Edit the **Password Change Commands** to contain the command set you need.
 - o Use the Delete button to remove a row. 
 - o Use the Edit button to modify a row. 
 - o Use the up and down arrows to move a row.  
 - o Use the Plus button to save a row. 

New PW Changer

Verify Password Changed Commands

AUTHENTICATE AS

Username

Password

Key

Passphrase

ORDER	COMMAND	COMMENT	PAUSE(MS)
	<input type="text"/>	<input type="text"/>	2000 <input type="button" value="+"/>

Password Change Commands

AUTHENTICATE AS

Username

Password

Key

Passphrase

6. Edit the **Verify Password Changed Commands** to create the command set for checking that the password is valid. These commands are used by heartbeat and after a password change to verify that the change was successful.
7. When you are finished editing the commands, scroll to the bottom and click **Back** to return to the overview screen and access test actions for your new password changer. To edit advanced commands and settings, see the instructions below.

Advanced Post Change Commands

To modify advanced post change commands, do the following:

1. Scroll to the bottom of the page and click **Advanced Post Change Commands**.

[Advanced Post Change Commands](#)
[Advanced Settings](#)

2. Change the commands as desired in the under **Post Successful Change Commands** and **Post Failure Change Commands**.

Post Successful Change Commands ⓘ *Test Disabled*

AUTHENTICATE AS

Username

Password

Key

Passphrase

ORDER	COMMAND	COMMENT	PAUSE(MS)
	<input type="text"/>	<input type="text"/>	2000 <input type="button" value="+"/>

Post Failure Change Commands ⓘ *Test Disabled*

AUTHENTICATE AS

Username

Password

Key






Passphrase




ORDER	COMMAND	COMMENT	PAUSE(MS)
	<input type="text"/>	<input type="text"/>	2000 <input type="button" value="+"/>

Advanced Settings

To modify advanced settings, do the following:

1. Scroll to the bottom of the page and click **Advanced Settings**.
2. Change the settings as desired in the under **Post Successful Change Commands** and **Post Failure Change Commands**.

SETTING	VALUE	
Remote Password Changing Timeout (minutes)	5	
Bypass Verify After Password Change	No	
Heartbeat Unknown Error to Unable to Connect Translation (regex)		
Attempt Password Change with new password when error contains (regex)		
Advanced: Delay Verify After Password Change (seconds)		

 Back  Configure Scan Template  View Audit

Note: Before attempting to change a password through Secret Server using your new custom password changer, we recommend that you test the command set you are using.

A Note About Commands

Any term in these commands preceded by \$ will reference a secret template field. Any term preceded by \${1} refers to the Secret template field of a linked Secret. If you need to reference a secret template field, make sure you are using the exact secret template field name.

To use your new password changer, you will need to assign it to a secret template. See [Assigning a Password Changer to a Secret Template](#).

Creating a Custom Password Changer for Cisco ASA

To create a custom password changer using SSH for Cisco ASA 5505, 5515 and other models with IOS 12.2 and earlier that cannot use the copy command, follow the procedure for [Creating a Custom Password Changer](#). Make sure you choose a base password changer that ends with (SSH) with a command set similar to the one you are adding, and use the following settings:

Authenticate As

1. `$_[1]$USERNAME`
2. `$_[1]$PASSWORD`

Commands

1. Enter `enable`
2. Enter `$_CURRENTPASSWORD`
3. Enter `config terminal`
4. Enter `enable password $_NEWPASSWORD`
5. Enter `end`
6. Enter `wr mem`
7. Enter `exit`

Creating a Custom SSH Password Changer

To create a custom SSH password changer for a device that supports changing passwords via SSH, follow the procedure for [Creating a Custom Password Changer](#). Be sure to choose a base password changer with a name that ends in (SSH).

Deactivating Password Changers

To make a password changer unavailable for use and to hide it from view in your list of password changers, you must mark it inactive:

1. From the **Password Changers Configuration** page, click the password type name of the password changer you would like to make inactive.
2. Click **Edit**.
3. Uncheck the **Active** box.
4. Click **Save**.

To view inactive password changers, check the **Show Inactive** box at the bottom of the list of password changers. The Active column in the table indicates the status of the password changer.

Distributed Engines and RPC

Distributed Engines allow RPC, heartbeat and discovery to occur on networks that are not directly connected to the network that Secret Server is installed on. See the linked KB and its associated white paper for details on configuration and functionality.

Note: Distributed Engines were released in version 8.9.000000 and replaced remote agents.

Editing Custom Commands

The SSH type changers use the SSH protocol to access the machine. This type contains custom commands for password reset and can contain commands for the verify password functionality but most SSH type changers simply verify that a connection can be established with the username and password. The Telnet type changers use the Telnet protocol in order to access the machine and contain custom commands for both the password reset and the verify password functionality. The verify functionality is used in the heartbeat, as well as verifying that the password was changed successfully.

SSH key rotation type changers also include post-reset success and failure custom commands. These extra command sets are run after both the reset and verify functions are run and are used to either finalize the key rotation and password change (success) or clean up after a failure. If both the reset and verify functions are successful, the post-reset success command set is run. If either the reset or the verify fail, the post-reset failure command set is run.

To edit the custom commands, click on the **Edit** Commands button. This sets the command grids into Edit mode where you can add, update, or delete the commands in order to suit their purpose.

RPC-Mapped Text-Entry Fields

Prepend a \$ to any text-entry field name to access that field. For example, to echo the notes value for a secret, you would use this command: `echo $Notes`. Commonly accessed fields include:

- `$USERNAME` The username text-entry field mapped in RPC on the secret template.
- `$CURRENTPASSWORD` The password text-entry field mapped in RPC on the secret template.
- `$NEWPASSWORD` The next password (filled in Next Password textbox or auto-generated).
- `$PRIVATEKEY` The private key text-entry field mapped in RPC on the secret template.
- `$NEWPRIVATEKEY` The next private key (filled in Next Private Key text box or auto-generated).
- `$CURRENTPUBLICKEY` The public key text-entry field mapped in RPC on the secret template.
- `$NEWPUBLICKEY` The next public key (generated from the next private key).
- `$PASSPHRASE` The passphrase text-entry field mapped in RPC on the secret template.
- `$NEWPASSPHRASE` The next passphrase (filled in Next Private Key Passphrase text box or auto-generated).

Associated Reset Secrets

- `[$1]$` Adding this prefix to any text-entry field targets the associated reset secret with order 1.
- `[$1]$USERNAME` The mapped username of the associated secret, identified by order. Can also reference any other property on the associated secret. Common examples include:
 - `[$1]$PASSWORD`
 - `[$1]$CURRENTPASSWORD`
 - `[$1]$PRIVATE KEY`
 - `[$1]$PRIVATE KEY PASSPHRASE`
- `[$(SID:105)]` Adding this prefix to any text-entry field targets the associated reset secret with a secret Id of 105.
- `[$(SID:105)]$USERNAME` The mapped username of the associated secret, identified by secret id. Like referencing an associated secret by order, referencing by secret id can also access any text-entry field on the secret by name.

Note: Both the mapped text-entry fields and secret text-entry field names can be used.

Check-Result Commands

- `$$CHECKCONTAINS <text>` Checks that the response from last command contains `<text>`.
- `$$CHECKFOR <text>` Checks that the response from the last command equals `<text>`.
- `$$CHECKNOTCONTAINS <text>` Checks that the response from last command does not contain `<text>`.

Note: If these conditions are not met the process fails and immediately returns a result.

If you want to exit out of the command set early without triggering a failure, echo an "OK" on the line immediately preceding the `exit 0;` statement. "OK" must be the only text in the response from the server for this to work.

You can test out your password reset and verify password command sets by clicking on the **Test Action** buttons next to the relevant sections. All communication between Secret Server and the target machine is displayed when using these test buttons.

Enabling RPC

RPC is enabled under the Administration, Remote Password Changing page. Click **Edit** to enable RPC, secret heartbeat, and secret checkout. Once enabled, all secret templates with RPC configured are available to use with RPC.

Mapping Account Fields for RPC

All the secret templates with the prefix RPC have RPC configured by default. For creating a custom template that uses RPC it can be configured from the Secret Template Designer. **Enable Remote Password Changing** must be turned on for secrets created from the template to make use of this feature. Select the password type for the account and map the text-entry fields to be used for authenticating to the remote server. The secret fields must be mapped to the corresponding required text-entry fields based on the password change type. Do that in the **Secret Template Edit Password Changing** page for the secret template:

Secret Template Edit Password Changing

Enable Remote Password Changing	Yes
Retry Interval	1 hour
Maximum Attempts	10000
Enable Heartbeat	Yes
Heartbeat Check Interval	8 hours

Password Type to use Active Directory Account

PASSWORD TYPE	SECRET FIELD	SCRIPT VARIABLE
Domain	Domain	\$domain
Password	Password	\$password
User Name	Username	\$username
Domain Controller (DC)		\$domaincontroller
Default Privileged Account		< None >

Back
 Edit

The **Retry Interval** text box is the amount of time that a secret waits before once again attempting to change a password after a password change is unable to succeed.

The **Default Privileged Account** text box is the secret that is set as the privileged account for all new secrets that are created with this secret template. Changing this does not affect any existing secrets.

Mapping an SSH Key or Private Key Passphrase for Authentication

Some password changers may be customized to use SSH key authentication. Secret Server needs to know which text-entry fields contain the key and the passphrase. These text-entry fields can be specified after clicking **Edit** from the password changer page.

Unix Account Custom (SSH)

Verify Password Changed Commands [Test Action](#)

AUTHENTICATE AS

Username \$USERNAME
 Password \$CURRENTPASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
-------	---------	---------	-----------

Password Change Commands [Test Action](#)

AUTHENTICATE AS

Username \$USERNAME
 Password \$CURRENTPASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	passwd	Password Command	2000
2	\$CURRENTPASSWORD	Current Password	2000
3	\$NEWPASSWORD	New Password	2000
4	\$NEWPASSWORD	Confirmed Password	2000

[Advanced Post Change Commands](#)
[Advanced Settings](#)

[Back](#)
[Edit](#)
[Edit Commands](#)
[Configure Scan Template](#)
[View Audit](#)

The key and passphrase must be identified by a \$ sign and the secret text-entry field name, which can be obtained from the secret template.

To set which text-entry fields are your key and passphrase, go to the extended mappings for a secret template by clicking **Extended Mappings** from the **Secret Template Edit** page. Select the text-entry fields that correspond to the SSH private key and passphrase if applicable. No matter what you name your key text-entry field, Secret Server knows what it is. This is set up by default, so you should not need to do this unless you've created custom Unix templates you want to use keys with.

Once Secret Server knows which text-entry fields contain the private key and private key passphrases, it can automatically use them as a part of launchers.

Minimum Requirements for Windows Local Accounts

Due to a security issue ([MS KB3178465](#)), we do not allow Windows local accounts to change their own passwords. Instead, we recommend using the discovery privileged account to change them. Each privileged account should meet the following requirements:

- Must be a domain user
- Must be a member of the local administrator group on all target end points

Note: The discovery account for Secret Server can also be used for RPC.

Modifying Password Changers

To modify a password changer, click the password changer name under **Admin > Remote Password Changing > Configure Password Changers** and then use the **Edit** or **Edit Commands** buttons to make changes. For more information about editing the custom PowerShell password changer, see [Remote Password Changing with PowerShell](#).

Note: You can find the full, up-to-date list of password changers included with Secret Server by default in [List of Built-In Password Changers](#) (KB).

Password Changing Scripts

PowerShell scripts, SSH scripts, and SQL scripts for password changing (PowerShell only), dependencies, and discovery custom actions can be created by administrators with the role permission called Administer Scripts. The scripts can be accessed by going to **Administration > Remote Password Changing > Scripts**.

Note: Secret Server requires that WinRM is configured on the Web server. For instructions please see [Configuring WinRM for PowerShell](#).

Creating Scripts

On the **Scripts** screen, select desired script tab and click **Create New** to enter the name of the script, a description, and the commands to run, then click **OK**. The script now shows up in the grid. Scripts can be deactivated and reactivated from the grid.

Testing Scripts

All scripts run from the machine that Secret Server is installed on, or the site assigned to the secret. To test a script, click the **Test** button on the grid next to the corresponding script.

PowerShell scripts run as the identity of the secret, so enter in an Active Directory credential to run the script as or select a secret to pre-fill the run-as credentials. Then enter in any command line arguments that the script requires. The output of the script is displayed above the grid for debugging purposes. To test the script over an engine, select a site from the **Site** list. This helps in diagnosing server specific issues where engines are installed.

Using Scripts

To use a script as a password changer or Dependency, it must be wired up to the appropriate action under **Admin > Remote Password Changing** on the password changer or dependency changer.

Discovery scripting is done under **Admin > Discovery > Extensible Discovery**. For more information on configuring extensible discovery see the [Extensible Discovery Overview](#).

Viewing Audits

A full history of each PowerShell script is kept and can be downloaded from the audit trail. Click **View Audit** to view the audit trail for PowerShell. Each time a script is updated, the previous one can be downloaded from the corresponding audit record.

Note: For additional information on setting up PowerShell scripts, please read the following KB article: [Creating and Using PowerShell Scripts](#).

RPC Error Codes

The most common RPC errors are:

- **NERR_PasswordPolicySettings**: The password Secret Server attempted to set is a repeating password or one that does not meet domain password policy standards. A common reason is minimum password age, which is often defaulted to 24 hours.
- **ERROR_ACCESS_DENIED**: The user account's "Not Able to Change Password" setting could not be set or logon was denied.
- **ERROR_INVALID_PASSWORD**: Either the user does not exist (ensure the usernames match) or the password is not correct.

For more information about common error messages for Remote Password Changing, see [Remote Password Changing Errors](#) (KB).

RPC for Service Accounts and SSH Keys

Service Accounts

RPC can be performed on service accounts where the dependent services is automatically updated and restarted as the service account password is changed. Administrators are notified if a dependency fails to restart. The supported dependency types are IIS application pools, IIS application pool recycle, scheduled tasks, windows services, passwords embedded in .ini, .config, and other text files. Custom dependencies can be created using SSH, PowerShell, or SQL scripts. The application pool recycle only recycles the specified application pool, it does not update the password of the service account running the application pool. Secret Server attempts to unlock the service account should the account become locked during the dependency password change if there is a privileged account assigned to the secret.

SSH Keys

RPC can be performed on multiple public keys referencing the same private key in Secret Server. The dependency types for this situation are SSH key rotation and SSH key rotation privileged.

RPC Logs

The RPC logs for a specific secret can be accessed by clicking the **View Audit** button on Secret View page and ticking the check box at the bottom of the page for display password changing Log. The RPC logs for all secrets can be accessed by navigating to **Admin > Remote Password Changing**.

You can change the **Days to Keep Operational Logs** text box to set the period to keep RPC-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).


Running a Manual RPC

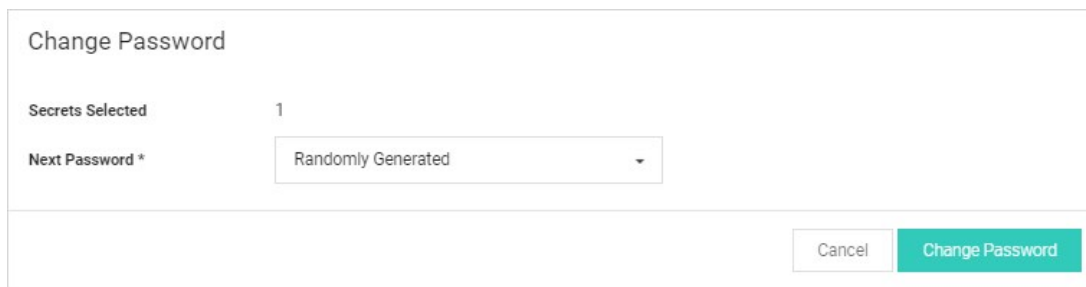
On the RPC tab there is a button called Change Password Remotely button that allows the user to change the password immediately instead of waiting for it to expire. When this button is clicked the user is taken to the Change Password Remotely page where they can enter in or generate the new password for the account. When the user clicks the Change button the secret enters the queue for having its password changed. The RPC Log found on the Remote Password Changing page details the results of the password change attempts and can be used for debugging.

If the secret is a Unix or Linux account and uses a password changer that supports SSH key rotation, the user can change the account's password, public and private keypair, and the private key passphrase. The user can enter or generate any of these items.

Note: If the password change fails, Secret Server continues to retry until it is successful, or the change is canceled by the user. To manually cancel the change, click Cancel Password Change on the RPC tab.

To run a manual RPC:

1. From **Dashboard**, click its check box to select secret you want to test.
2. Click the  Change Password Remotely button. The Change Password popup page appears:



The image shows a 'Change Password' popup form. At the top, it says 'Change Password'. Below that, there are two rows of information: 'Secrets Selected' with the value '1', and 'Next Password *' with a dropdown menu currently showing 'Randomly Generated'. At the bottom right of the form, there are two buttons: a 'Cancel' button and a 'Change Password' button.

3. Click to select the **Next Password** dropdown list and select **Manual** or **Randomly Generated**. If you chose manual:
 1. The Password text box appears.
 2. Type the new password in the **Password** text box.
 3. Click the **Change Password** button.

Otherwise, click the **Change Password** button. The password change is now queued.

4. You can verify that the password change completed either by unmasking the password on this screen (click the lock icon beside the password field) or by looking at the **Remote Password Changing** log. You can find the Remote Password Changing log by going to **Admin > Remote Password Changing**.

Treating Specific Heartbeat "Unknown Errors" as Connection Failures

Note: This setting was previously called "Password Change Error Code Translation (regex)."

The Secret Server "Heartbeat Unknown Error to Unable to Connect Translation (regex)" setting can translate UnknownError heartbeat errors into connection errors based on text, such as the error code, in the error message. Using a regular expression, which you define, Secret Server scans heartbeat UnknownError messages for specific text strings. When there is a match, Secret Server changes the UnknownError to an "Unable to Connect" heartbeat error. This setting is useful if a custom command error is interpreted as UnknownError but the message indicates it actually was unable to connect. The translated connection error will cause Secret Server to attempt another heartbeat.

Figure: Heartbeat Unknown Error to Unable to Connect Translation (regex) Setting

The screenshot shows the configuration page for an Active Directory Account. It includes sections for 'Verify Password Changed Commands', 'Password Change Commands', and 'Password Change By Admin Credentials Commands', each with a 'Test Action' button and a message: 'This process is done through internal commands. The commands cannot be edited.' Below these is a 'Hide Advanced Settings' section containing a table of settings.

SETTING	VALUE
Heartbeat Unknown Error to Unable to Connect Translation (regex)	<input type="text"/>
Attempt Password Change with new password when error contains (regex)	<input type="text"/>

At the bottom of the page are buttons for 'Back', 'Configure Scan Template', and 'View Audit'.

Note: The UnknownError error is very common when running scripts and commands, making the regex discrimination desirable.

Logic:

(RPC UnknownError) AND (Regex match in error message) => RPC status changed to "Unable to Connect"

Example:

. *error code is 10060.* (any error with the code 10060 changes the RPC status to "Unable to Connect")

Procedure

To configure the unknown errors to trigger connection failures:

1. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears:

Remote Password Changing Configuration

Enable Remote Password Changing	Yes
Enable Password Changing on Check In	No
Enable Heartbeat	Yes

[Advanced \(not required\)](#)

Days to Keep Operational Logs	30
-------------------------------	----

[← Back](#) [✎ Edit](#) [✎ Configure Password Changers](#) [⚙️ Configure Dependency Changers](#)

[🏗️ Distributed Engine Configuration](#) [📄 View Audit](#)

Logs

[Password Changing](#) [Heartbeat](#)

[▶ Run Now](#)

Record Count 0 Page 1 / 1 << Prev Next >>

i No results matching the current filter.

2. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:

Password Changers Configuration

PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes

3. Click the link for the desired password type. Its Account page appears:

Active Directory Account

Verify Password Changed Commands Test Action Password Change Commands Test Action

This process is done through internal commands. The commands cannot be edited. *This process is done through internal commands. The commands cannot be edited.*

Password Change By Admin Credentials Commands Test Action

This process is done through internal commands. The commands cannot be edited.

[Hide Advanced Settings](#)

SETTING	VALUE
Heartbeat Unknown Error to Unable to Connect Translation (regex)	<input type="text"/>
Attempt Password Change with new password when error contains (regex)	<input type="text"/>

Back Configure Scan Template View Audit

4. If necessary, click the **Advanced Settings** link.

5. Click the pencil icon next to **Heartbeat Unknown Error to Unable to Connect Translation (regex)**. The Value text box appears.

6. Determine the desired text string to search for.

7. Type the desired regex in the **Value** text box.

8. Click the **Save** icon next to the text box

Triggering an RPC When Defined Errors Occur

When the "Attempt Password Change with new password when error contains (regex)" setting is enabled, Secret Server generates a new password to use during the next RPC attempt when the defined error is returned. Using a regular expression, which you define, Secret Server scans the error message for specific text strings. When there is a match, Secret Server generates and sets a new next password for the secret that will be used in the next RPC attempt, which will occur based on the templates RPC interval. To keep this process from generating too many next passwords, it is restricted to five attempts while failing RPC.

Note: Only the password field is updated. Passcodes and SSH keys are left alone.

Figure: Attempt Password Change with new password when error contains (regex) setting

Active Directory Account

Verify Password Changed Commands Test Action

Password Change Commands Test Action

This process is done through internal commands. The commands cannot be edited.

Password Change By Admin Credentials Commands Test Action

This process is done through internal commands. The commands cannot be edited.

Hide Advanced Settings

SETTING	VALUE
Heartbeat Unknown Error to Unable to Connect Translation (regex)	
Attempt Password Change with new password when error contains (regex)	

Back Configure Scan Template View Audit

Logic:

(RPC Error) AND (one or more regex matches) AND (five or fewer attempts) = > New password generated

Examples:

.*UnknownError.* (any unknown error)

.* (any error)

.*minimum.* (minimum password length requirement error)

.*0x80072035.* (server rejects password error)

.*0x80072035.*I.*minimum.* (server rejects password or password length error)

Procedure

To configure RPC in response to specific unknown errors:

1. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears:

Remote Password Changing Configuration

Enable Remote Password Changing	Yes
Enable Password Changing on Check In	No
Enable Heartbeat	Yes

[Advanced \(not required\)](#)

Days to Keep Operational Logs	30
-------------------------------	----

[Back](#) [Edit](#) [Configure Password Changers](#) [Configure Dependency Changers](#)
[Distributed Engine Configuration](#) [View Audit](#)

Logs

[Password Changing](#) [Heartbeat](#)

[Run Now](#)

Search... 50 90 minutes [Refresh](#) Record Count 0 Page 1 / 1 [Prev](#) [Next](#) [Download](#)

i No results matching the current filter.

1. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:

Password Changers Configuration

PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes

1. Click the link for the desired password type. Its Account page appears:

Active Directory Account

Verify Password Changed Commands Test Action Password Change Commands Test Action

This process is done through internal commands. The commands cannot be edited. *This process is done through internal commands. The commands cannot be edited.*

Password Change By Admin Credentials Commands Test Action

This process is done through internal commands. The commands cannot be edited.

[Hide Advanced Settings](#)

SETTING	VALUE
Heartbeat Unknown Error to Unable to Connect Translation (regex)	
Attempt Password Change with new password when error contains (regex)	

[Back](#) [Configure Scan Template](#) [View Audit](#)

1. If necessary, click the **Advanced Settings** link.
2. Click the pencil icon next to **Attempt Password Change with new password when error contains (regex)**. The Value text box appears.
3. Determine the desired text string to search for.
4. Type the desired regex in the **Value** text box.
5. Click the **Save** icon next to the text box.

Overview

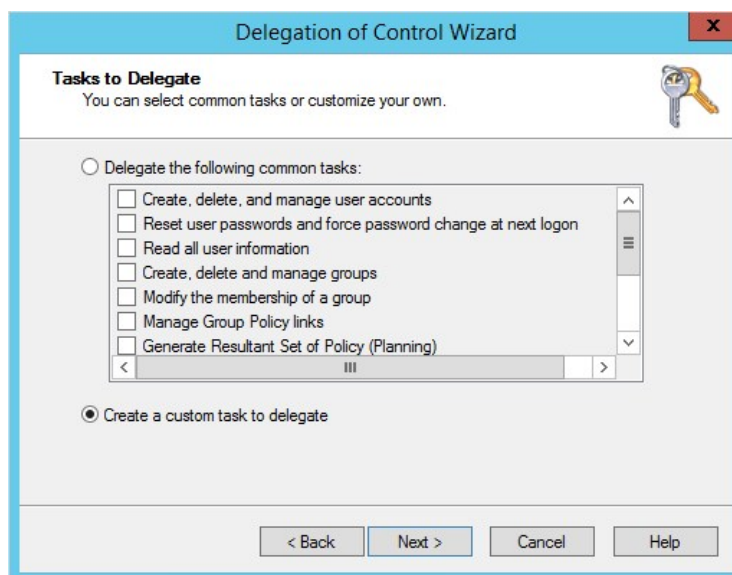
Secret Server requires proper permissions to perform remote password changing. The privileged secret used for remote password changing of an Active Directory (AD) account secret must have the following minimum permissions:

- Change password
- Reset password
- Write lockoutTime
- Write pwdLastSet
- Write UserAccountControl
- Read all properties on CN=System,CN=Password

Settings Container and all child containers or objects, or read all properties on any other fine-grained password policy objects (this is completed through ADSIedit).

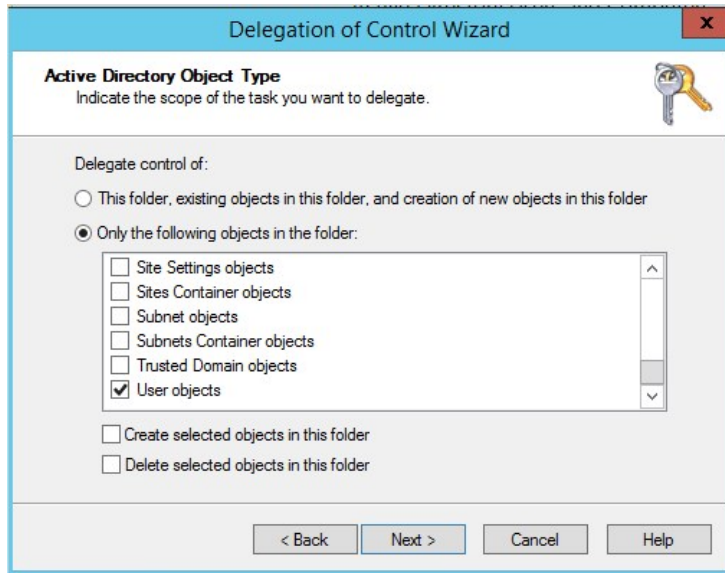
Setting Permissions

1. Open the Active Directory Users and Computers administrative console.
2. Right-click the Organizational Unit (OU) or the top-level domain you want to configure and select **Delegate Control...** from the context menu. The Delegation of Control Wizard appears on the Welcome page.
3. Click the **Next** button. The **Users or Groups** dialog appears (not shown).
4. Click the **Add...** button to enter and search Active Directory for the appropriate users or groups. The Select Users, Computers, or Groups dialog box appears (not shown).
5. When done with the selection, click the **OK** button.
6. Click the **Next** button. The **Tasks to Delegate** dialog box appears:



7. In the **Tasks to Delegate** dialog, click to select the **Create a custom task to delegate** selection button.

8. Click **Next**. The Active Directory Object Type dialog box appears:



9. Click to select the **Only the following objects in the folder** selection button.

10. Scroll to bottom of the list.

11. Click to select the **User objects** check box.

12. Click the **Next** button. The Permissions dialog box appears (not shown).

13. Click to select the **General** check box.

14. Locate and select the following in the **Permissions** list:

- Change password
- Reset password

15. Click to deselect the **General** check box.

16. Click to select the **Property-specific** check box.

17. Locate and select the followings in the **Permissions** list:

- Write lockoutTime
- Read lockoutTime
- Write pwdLastSet
- Read pwdLastSet
- Write UserAccountControl
- Read UserAccountControl

18. Click the **Next** button.

19. Click the **Finish** button.

This document explains how to configure Oracle Database 19c for heartbeat and remote password changing (RPC) with Secret Server. It consists of installing the Oracle Database Access Components (ODAC), configuring Secret Server, and configuring one or more distributed engines.

Note: This document is not updated with every release—many releases do not affect the guide's contents and thus do not warrant a document update.

This Delinea technical configuration knowledge base article is relevant to and has been tested on:

- Secret Server 10.7 on Windows Server 2016 Standard (64-bit)
- Distributed engine 10.7 on Windows Server 2016 Standard (64-bit)
- Oracle Database 19c on Windows Server 2019 Standard (64-bit)

Introduction

This document explains how to configure Oracle Database 19c for heartbeat and remote password changing (RPC) with Secret Server. The process consists of installing the Oracle Database Access Components (ODAC), configuring Secret Server, and configuring one or more distributed engines.

Procedure

Task One: Installing the Oracle Database Access Components

1. Navigate to [ODAC Runtime Downloads](#) in your browser.
2. Download the latest version of the ODAC OUI file with the same major number as your database version.
3. Unzip the file.
4. Right click and run setup.exe as a Windows administrator. The setup wizard appears.
5. Click to select **Use Windows Built-in Account**.
6. Click the **Next** button.
7. Type the desired installation path, such as c:\oracle.
8. Click the **Next** button.
9. Confirm the default product components are selected. If not, click to select **Reset Defaults**.
10. Click the **Next** button.
11. Leave the **DB Connection Configuration** fields as they are (empty).
12. Click the **Next** button. The setup runs some pre-installation configuration tests.
13. When the tests are completed, click the **Install** button.
14. Reboot your machine.

Task Two: Configuring Secret Server

1. Navigate to the ODAC directory, such as c:\oracle.

2. Copy the C:\oracle\product\19.x\client_1\odp.net\bin\4\Oracle.DataAccess.dll file to the bin directory of your Secret Server directory, for instance, c:\inetpub\wwwroot\SecretServer\bin.
3. Navigate to the C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config directory.
4. Open the machine.config file.
5. Copy and paste the line below into the DbProviderFactories section:

```
<add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
```

The section should look like this:

```
<system.data>
<DbProviderFactories>
  <add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</DbProviderFactories>
</system.data>
```

Note: There may be additional <Add> sections, such as for Microsoft SQL Server. Leave them as is.

Task Three: Configuring a Secret Server Distributed Engine

1. Install ODAC on the machine hosting the distributed engine using the same procedure as Task One.
2. Navigate to the ODAC directory on the distributed engine machine, such as c:\oracle.
3. Copy the C:\<ODAC_Directory>\odp.net\bin\4\Oracle.DataAccess.dll file to the Distributed Engine directory, for instance, C:\Program Files\Thycotic Software Ltd\Distributed Engine.
4. Navigate to the C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config directory.
5. Open the machine.config file.
6. If necessary, create a <DbProviderFactories> section within the <system.data> section.
7. Copy and paste the line below into the <DbProviderFactories> section:

```
<add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
```

The section should look like this:

```
<system.data>
<DbProviderFactories>
  <add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</DbProviderFactories>
</system.data>
```

Troubleshooting

Log Files

The errors below may appear in these files:

- SS-BWSR.log (Secret Server)
- SSDE.log (distributed engine)

These files are located within the log subdirectory of the application's directory. Typically, these are:

- Secret Server: c:\inetpub\wwwroot\secretserver\log\SS-BWSR.log
- Distributed engines: c:\program files\thycotic software ltd\distributed engine\log\SSDE.log

Errors

Oracle.DataAccess.Client.OracleException: The provider is not compatible with the version of Oracle client

This error occurs when the Oracle ODAC driver does not match the Oracle database version.

Uninstall the ODAC, and then re-install the correct version. You can uninstall ODAC using the universal installer that is included with the ODAC installation that resides in the ODAC directory.

Oracle.DataAccess.Client.OracleException (0x80004005): ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

This error occurs when the secret's database field does not match the Oracle SERVICE_NAME database.

Note: The default "Oracle Account" secret template's database field is looking for the Oracle SERVICE_NAME database. You can find that database's location by reading the tnsnames.ora configuration file on your Oracle database server.

System.ArgumentException: Unable to find the requested .Net Framework Data Provider. It may not be installed.

This error occurs when Oracle parameters are missing from the section in the machine.config file.

System.Configuration.ConfigurationErrorsException: Unrecognized element

This error occurs when the section, located in the machine.config file, is not properly formatted.

Overview

Secret Server includes many pre-configured password changers that are used by Remote Password Changing (RPC). The following are commonly used password changers, and the list is always growing.

Note: To see the latest list, go to Admin > RPC > Configure RPC.

Note: Secret Server can use scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to conduct password changes to other platforms.

List

The followings are the current built-in password changers:

- Active Directory Account
- Amazon IAM Console Password Privileged Account
- Amazon IAM Key
- Blue Coat Account Custom (SSH)
- Blue Coat Enable Password Custom (SSH)
- Cisco Account Custom (SSH)
- Cisco Account Custom (Telnet)
- Cisco Enable Secret Custom (SSH)
- Cisco Enable Secret Custom (Telnet)
- ESX/ESXi (API)
- F5 BIG-IP Root Account (SSH)
- Generic Discovery-Only Credentials
- Generic ODBC (DataSource)
- HP iLO Account Custom (SSH)
- IBM iSeries Mainframe
- Juniper Account Custom (SSH)
- LDAP (Active Directory)
- LDAP (DSEE)
- LDAP (OpenLDAP)
- MySQL Account
- Office365 *
- Oracle Account
- Oracle Account (AS SYS)
- Oracle Account (DataSource)
- PostgreSQL Account (x64)
- PowerShell Script **
- SAP Account **
- SonicWall NSA Web Admin Account
- SonicWall NSA Web Local User Account
- SQL Server Account
- SSH Key Rotation **
- SSH Key Rotation (No Password) **
- SSH Key Rotation Privileged Account **
- SSH Key Rotation Privileged Account (No Password) **
- Sybase Account
- Unix Account (SSH)

- Unix Account (Telnet)
- Unix Account Custom (SSH)
- Unix Account Custom (Telnet)
- Unix Account SU Takeover (SSH)
- Unix Account SUDO Takeover (SSH)
- Unix Root Account Custom (SSH)
- WatchGuard Custom (SSH)
- Web User Account (built-in support for AWS, Google, Salesforce)
- Windows Account
- z/OS Mainframe

* Does not require an Advanced Scripting Add-On License. Will require PowerShell installation. ** Professional Edition add-on/Platinum Edition only

Other platforms that Secret Server can change passwords on include:

- AS/400
- Linux / Mac
- Check Point
- Enterasys
- Dell DRAC

Overview

You can create custom SQL password changers based on an ODBC driver. The Secret Server machine must have the corresponding ODBC driver installed. These can be downloaded from the corresponding database vendor sites.

ODBC connection strings vary depending on product. See [Example Connection Strings](#) below for sample ODBC connection strings for Microsoft SQL server and PostgreSQL.

Create an ODBC Password Changer

1. In Secret Server, go to **Admin > Remote Password Changing**.
2. Click **Configure Password Changers**, and then scroll to the bottom of the page.
3. Click the **New** button.
4. Select **Generic ODBC (DataSource)** in the **Base Password Changer** dropdown list.
5. Type a name for your new custom password changer.
6. Under **Password Reset Commands**, type a command to reset a password (see below).

Note: Secret field variables can be used in a way similar to how they are used in a Linux or UNIX password changer, with the exception that they can be specified as ODBC parameters, assuming the command allows it. To parameterize a secret field variable, prefix it with the @ symbol instead of a \$.

Example Reset Commands

Parameterized SQL server command:

```
EXEC sp_password @CURRENTPASSWORD, @NEWPASSWORD
```

Note: If the command does not support using parameters, the secret field values can be substituted into the command.

Substitution PostgreSQL command:

```
ALTER ROLE "$USERNAME" WITH ENCRYPTED PASSWORD '$NEWPASSWORD'
```

Substitution MySQL command:

```
ALTER USER '$USERNAME' IDENTIFIED BY '$NEWPASSWORD';
```

Adding Connection Strings

Each ODBC password changer requires a connection string. This can be specified within the password changer settings or in the secret itself.

Adding Connection Strings to Password Changer Settings

Add a connection string to password changer settings:

1. In Secret Server, go to **Admin > Remote Password Changing**.
2. Click **Configure Password Changers**.

3. Click the name of your password changer.
4. Click the **Edit** button.
5. Type your database ODBC connection string in the **Connection String** text box.
6. Click the **Save** button.

Adding Connection Strings to Secrets

The Connection String can also be specified on the secret by adding a new field to the template and mapping it to the **Data Source** property on the template's **Remote Password Changing** configuration. Otherwise, that mapping field can be left blank.

Note: See [Creating or Editing Secret Templates](#) for more information about adding fields to secret templates.

Example connection strings:

SQL 2012:

```
Driver={SQL Server Native Client 11.0};Server=$SERVER;Database=master;Uid=$USERNAME;Pwd=$PASSWORD;
```

PostgreSQL (x64):

```
Driver={PostgreSQL ANSI(x64)};Server=$SERVER;Port=$PORT;Database=$DATABASE;Uid=$USERNAME;Pwd=$PASSWORD;
```

Troubleshooting

A common problem experienced with ODBC drivers is they require the IIS application pool to be set to either 32-bit or 64-bit mode to match the specified ODBC driver. When not set correctly, you will see an error in the system log when running heartbeat for a secret using that password changer.

PostgreSQL with 64-bit drivers will throw the following error if the IIS application pool is in 32-bit mode:

```
ExpiredSecretMonitor - System.Data.Odbc.OdbcException (0x80131937): ERROR [IM002] [Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified
```

PostgreSQL with Distributed Engines

A machine with a distributed engine installed requires the corresponding ODBC driver. In some cases, additional configuration may be necessary. For example, PostgreSQL requires adding an additional host entry:

1. Install the latest PostgreSQL ODBC drivers on the agent computer.
2. Modify the `pg_hba.conf` (for example: `/PostreSql/9.3/pg_hba.conf`) file to have a host entry for the agent computer IP address. For example, where 192.168.60.147 is the IP address of the distributed engine:

```
host all all 192.168.60.147/32 md5
```

By default, RPC uses the credentials on secret option, using the credentials stored in the secret to invoke a password change. For Windows and Active Directory accounts, a privileged account can be used instead by selecting the Privileged Account Credentials option and selecting an Active Directory secret with permission to change the account's password.

For secret templates with a custom commands password type, any number of associated reset secrets can assign for use in the custom commands.

When a secret is wired up with a privileged account or reset secrets, the ability to edit the username, host, domain, or machine is restricted if the user does not have access to those associated secrets. On the RPC tab, the user sees "You do not have access to View this secret" for the secret name and on the Edit page all text-entry fields mapped for RPC except the password is disabled. This added security prevents the user from changing the username and resetting another account's password.

Note: If the user does not have access to the privileged account or reset secrets, the ability to edit all secret text-entry fields mapped for RPC except the password text-entry field is restricted to prevent changing the password on another account.

Note: To appear in searches, privileged accounts must have RPC enabled in their originating template or the Active Directory secret.

Overview

This topic lists some of the common errors experienced when setting up Remote Password Changing (RPC) for an account.

To view the errors, navigate to **Admin > Remote Password Changing** in Secret Server and look for the name of the secret you are testing.

Errors

The user name cannot be found

For local Windows accounts, ensure you only have the username in the username field (do not include the machine name). The machine name should go in the Machine field only. If the RDP Launcher stops working when you remove the machine name from the username field, see [RDP Proxy Configuration](#).

Change password failed: Unknown. (ERROR_CANT_ACCESS_DOMAIN_INFO)

For RPC on local Windows accounts, this error can be deceptive because the built-in Windows method used to change a password takes either a machine or domain name, so if the machine is not found, it thinks a domain was passed in and throw a domain error.

For RPC on Active Directory accounts, this error may occur if the account does not have permission to perform the password change or the domain name is wrong or abbreviated. Verify by checking the account properties in Active Directory or log in to the account and try to change the password manually or use privileged secret to perform the RPC.

Note: The RPC process uses information from the secret, not a central configuration for resetting the password. The Active Directory configuration settings are used for user synchronization only, so ensure the information on the secret is correct, including the Active Directory domain.

Common causes include:

- The machine name is wrong or abbreviated. For example: comp3 is entered as the machine name instead of comp3.yourdomain.com. Try replacing the machine name in the secret with the IP address of the machine and seeing if you still receive the domain error.
- The firewall is blocking the ports. See [Secret Server Ports](#).

Monitor activity to see if the authentication is accepted on the machine by viewing the security log:

1. Run secpol.msc from the Run prompt.
2. Click on Local Policies, Audit Policy.
3. Turn on "Audit account logon events" and "Audit logon events" for both Successes and Failures.
4. View the logs at Administrative Tools > Event Viewer. Check the Security Logs to determine whether the requests are getting through to the computer.

Note: The RPC log looks different if the firewall denies the connection, and will show ERROR_ACCESS_DENIED in some cases.

Firewall settings also apply to changing passwords on the local machine that Secret Server is running on because net authentication is used.

Change password failed: Unknown. (NERR_PasswordPolicySettings)

Cause: repeating password or a password that does not meet domain standards.

Check the minimum password age. When performing RPC on Active Directory accounts, this error may occur due to a minimum password age policy on the domain. If the minimum password age is set to 1 day or greater, and due to testing, the password has already been changed once, a follow up password change will violate the domain policy.

If you need to change the user's password more than the policy allows, change their policy so they are not subject to minimum password ages, or use the privileged account option in the Remote Password Changing tab on the secret. Privileged account will perform a password reset instead of changing the password using the accounts credentials.

Change password failed: Unknown. (ERROR_ACCESS_DENIED)

Cause: User account is set to Not Able to Change Password, firewall denial, or login incorrect. May also occur when the userWorkstations attribute on the user is set.

Change password failed: Unknown. (ERROR_INVALID_PASSWORD)

Cause: Either the user does not exist (ensure the usernames match) or the password is not correct.

Change password failed: Unknown. (ERROR_ACCOUNT_LOCKED_OUT)

Cause: User account is locked out.

ExpiredSecretMonitor - Unspecified error

Cause: Firewall issue or ports are blocked.

DirectoryEntry.Invoke SetPassword - The network path was not found.

Cause: Domain can not be found from the computer. Check the machine can ping the domain.

Secret '

Cause: the password changer for the secret template this secret is based on is looking for an "associated secret." Associated secrets are additional accounts that are needed in the password change process.

You can view the commands being used for the password change and add the associated Secret by going to the Remote Password Changing tab of the Secret in question and clicking Edit (you may also need to click Show Commands).

Error changing password - Check Out is enabled on associated Secret.

Cause: The secret has a Privileged Account Credentials option selected for performing the password change and the privileged account secret has the Require Check Out option enabled. This configuration causes an error with the remote password change process because the Required Check Out option is not intended for use by the system to avoid conflict from user's request, which is the intended usage.

Overview

This address uses a Secret Server privileged account to change SQL Server accounts. This enables taking over those accounts without knowing their password.

Creating the Account

1. Open SQL Server Management Studio.
2. Connect to your database server.
3. Expand the root-level security folder.
4. Right-click the **Logins** folder and select **New Login**.
5. Give the account a log on name.
6. Select SQL authentication.
7. Go to Secret Server.
8. Create a secret using the **SQL Server Account** template.
9. Assign it the desired username .
10. Click the **Generate** button on the secret password field to create a password.
11. Copy that password to the account creation wizard in SQL Server Management Studio.
12. Click the **OK** button to save the secret.

Assign Permissions

1. In SQL Server Management Studio, go to **Security > Logins** in the object explorer.
2. Right click on the SQL login object and select **Properties**. The Login Properties dialog box appears.
3. Select **Securables** in the **Select a page** list.
4. Find the **Alter any login** permission on the **Explicit** tab at the bottom of the dialog box.
5. Click to select the **Grant** check box for that permission.
6. Click the **OK** button.
7. Similarly, enable the **Control Server** permission. This is for changing the target logins that are members of the **sysadmin** fixed server role or grantees of this permission.

Using the Account

1. In Secret Server, open the SQL Server secret that you created.
2. Click the **Remote Password Changing** tab.
3. Click the **Edit** link.
4. Click to select **Privileged Account Credentials** in the **Change Password Using** selection buttons. The Privileged Account section appears.
5. Click the **No Secret Selected** link.

6. Select the secret you created earlier. The secret appears in the Privileged Account section.
7. Click the **Save** button.
8. Click the **Change password remotely** button.
9. Provide or generate a new password.
10. Click the **Change** button. You have now successfully changed a SQL Server account password using a privileged account.

Note: You can also assign the account for use by multiple secrets by creating a secret policy and applying that policy to a folder.

Overview

As of version 8.8, Secret Server supports running PowerShell scripts for Remote Password Changing (RPC) and heartbeat. Below are the steps for creating an Active Directory (AD) password changer that uses PowerShell scripts. The example is meant as a simple guide for how to wire-up the template to scripts as a proof of concept. Your actual PowerShell password changer scripts may be more complex depending on your environment and needs.

Important: Before you begin, please ensure password changing and heartbeat are enabled in **Admin > Remote Password Changing** and review the information on [Configuring CredSSP for use with WinRM/PowerShell](#), which will be necessary for most PowerShell password changing tasks.

Procedure

The PowerShell scripts are created and accessed through the **Admin > Scripts** page. To create a PowerShell password changer, you need to create two scripts. The first script verifies the account's current password. The second script changes the account's password. These two scripts are linked to a new secret template.

Task 1: Creating the Active Directory Verify Password Script

1. Navigate to **Admin > Scripts**.
2. Click the **+ Create New** button on the **PowerShell** tab.
3. Type the following information in the dialog:
 - o **Name:** Active Directory Verify
 - o **Description:** Script used to verify an Active Directory account
 - o **Category:** Heartbeat
 - o **Script:**

```
$domain = "LDAP://"+$Args[0];  
$dn = New-Object System.DirectoryServices.DirectoryEntry($domain, $Args[1], $Args[2]);  
if ($dn.name -eq $null){ throw "Authentication failed - please verify your username and password." };
```

4. Click the **OK** button to save the script.

Task 2: Creating the Active Directory Change Script

1. On the **PowerShell** tab, click the **+ Create New** button.
2. Type the following information in the dialog:
 - o **Name:** Active Directory Change
 - o **Description:** Script used to change the password of an Active Directory account
 - o **Category:** Password Changing
 - o **Script:**

```
$Domain = $args[0]  
$UserToChange = $args[1]  
$NewPassword = $args[2]  
$P_User = $args[0] + "\" + $args[3]  
$P_PWord = ConvertTo-SecureString -String $args[4] -AsPlainText -Force  
$Creds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $P_User, $P_PWord  
$pwd = ConvertTo-SecureString $NewPassword -AsPlainText -Force;  
Set-ADAccountPassword -Server $Domain -Identity $UserToChange -NewPassword $pwd -Reset -Credential $Creds
```

3. Click the **OK** button to save the script.

Task 3: Testing the Scripts

For the AD verification script:

1. Go to **Scripts > PowerShell tab**.
2. Click the Run Script arrow icon on the AD verify script. The Test Script popup appears.
3. Type the arguments (separated by spaces) in the **Arguments** text box: domain name (for you), username (yours), password (yours).
For example: my.company.com ssadmin FD#@789Uik4\$
4. Type your domain name for the script-running account in the **Domain** text box.
5. Type the username in the **Username** text box for account that can run PowerShell scripts on the domain.
6. Type that user's password in the **Password** text box.
7. Click the **OK** button to test your script the with provided parameters.

For the Active Directory change script:

1. Go to **Scripts > PowerShell tab**.
2. Click the Run Script arrow icon on the AD change script. The Test Script popup appears.
3. Type the arguments (separated by spaces) in the **Arguments** text box: domain name (for you), username (yours), new password (yours), domain admin username, domain admin password. For example: my.company.com ssuser 08sSKthsoidPW ssadmin FD#@789Uik4\$
4. Type your domain name for the script-running account in the **Domain** text box.
5. Type the username in the **Username** text box for account that can run PowerShell scripts on the domain.
6. Type that user's password in the **Password** text box.
7. Click the **OK** button to test your script the with provided parameters.

Note: If successful, this will change the password on the account that is used for testing.

The remaining steps depend on the version of Secret Server you are using. In Secret Server 10.0.000006 we introduced the ability to create multiple PowerShell password changers, each with their own set of password change and verify scripts. These password changers can be assigned to different scan templates to automatically assign different PowerShell password changer scripts to different types of local accounts when creating local account import rules in discovery. For more information about how scan templates and password changers are used in discovery and local account import rules, please see [Discovery](#). Prior to 10.0.000006, there was only one PowerShell password changer and the scripts were assigned on the secret template.

Task 4: Configuring a Password Changer for Secret Server Version 10.0.000006 and Later

In Secret Server versions 10.0.000006 and later, after the scripts are tested and working correctly, the next step is to create a PowerShell password changer.

1. Go to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. Click the **New** button.

4. In the **Base Password Changer** dropdown list, select **PowerShell Script**.
5. Type the name of the new password changer.
6. Click the **Save** button. On the next page you will select the scripts to use for password changing and verification (heartbeat).
7. Under **Password Change Commands**:
 1. Select the script that you created to do password changes.
 2. Type the following in the **Script Args** text box: \$DOMAIN \$USERNAME \$NEWPASSWORD \${1}\$USERNAME \${1}\$PASSWORD.
 3. Click the **Save** button next to the **Script Args** text box.
8. Under **Verify Password Changed Commands**:
 1. Select the script that you created to do heartbeats and verification.
 2. Type the following in the **Script Args** field: \$DOMAIN \$USERNAME \$PASSWORD.
 3. Click the **Save** button next to the **Script Args** text box.

Note: When Secret Server runs the script, it replaces the fields with the matching secret field values. \$NEWPASSWORD is a special case for the new password that is generated by Secret Server or specified by the user when performing a password change. For more information see [Using Secret Fields in Scripts](#).

Important: You must specify scripts for both sections and you must click the Save button next to each one for both to save.

Task 5: Creating a Secret Template

The next step is to create the secret template:

1. Go to **Admin > Secret Templates**.
2. Click the **Create New** button.
3. Name the template **PowerShell Active Directory**.
4. Create the following new fields:
 - o Domain Field Type: Text
 - o Username Field Type: Text
 - o Password Field Type: Password
 - o Notes Field Type: Notes
5. Click the **Configure Password Changing** button.
6. Click the **Edit** button.
7. Click to select the **Enable Remote Password Changing** and **Enable Heartbeat** checkboxes.

Task 6a: Finishing the Secret Template Configuration for Secret Server 10.0.000006 and later

Note: Complete either 6a or 6b, not both.

1. Select the password changer created in the previous section from the **Password Type to use** dropdown list.
2. Click to select **Domain** next to the **Domain** field.
3. Click to select **Username** next to the **User Name** field.
4. Click to select **Password** next to the **Password** field.

5. Click the **Save** button to save the mapping.

Task 6b: Finishing the Secret Template Configuration for Secret Server 8.8.000000 to 10.0.000000

Note: Complete either 6a or 6b, not both.

1. Select **PowerShell Script** from the **Password Type to use** dropdown.
2. Click to select **Domain** next to the Domain field.
3. Click to select **Username** next to the User Name field.
4. Click to select **Password** next to the Password field.
5. Click to select **Active Directory Change** next to the **Remote Password Change Script** field.
6. Enter the following to the **Remote Password Change Args** field: `$DOMAIN $USERNAME $NEWPASSWORD ${1}$USERNAME ${1}$PASSWORD.`
7. Click to select **Active Directory Verify** next to the **Heartbeat Script** field.
8. Type the following next to the **Heartbeat Args** field: `$DOMAIN $USERNAME $PASSWORD.`

Note: When Secret Server runs the script, it replaces the fields with the matching secret field values. `$NEWPASSWORD` is a special case for the new password that is generated by Secret Server or specified by the user when performing a password change.

9. Click the **Save** button to save the mapping.

Task 7: Creating Secrets Using PowerShell Remote Password Changing

Create the AD account secret PowerShell account:

1. Create three secrets (The first two **must** be different secrets):
 - One that is an Active Directory Account that has the necessary rights to run PowerShell on your domain
 - One that is an Active Directory Account that has the necessary rights to run a password change on your domain
 - One that is based on the new PowerShell Active Directory Template.
2. Create the Active Directory account secret PowerShell account.
3. On the dashboard, use the dropdown on the **Create Secret** widget and select **Active Directory Account**. Use the following parameters:
 - **Secret Name:** PowerShell Admin
 - **Domain:** Domain that the account exists on
 - **Username:** Account name that can run PowerShell scripts in the domain
 - **Password:** Password for the account
4. Click the **Save** button to save your secret and verify that it passes heartbeat.
5. Click the **Home** button to return to the dashboard.

Create the AD account secret for password changing:

1. On the dashboard, use the dropdown on the **Create Secret** widget and select **Active Directory Account**. Use the following parameters:
 - o **Secret Name:** Password changing Admin
 - o **Domain:** Domain that the account exists on
 - o **Username:** Account name that can change passwords in the domain
 - o **Password:** Password for the account
2. Click the **Save** button to save your secret and verify that it passes heartbeat.
3. Click the **Home** button to return to the dashboard.

Create the PowerShell Active Directory secret:

1. On the dashboard, use the dropdown on the **Create Secret** widget and select **PowerShell Active Directory Account**. Use the following parameters:
 - o **Secret Name:** PowerShell AD user
 - o **Domain:** Domain that the account exists on
 - o **Username:** samAccountName of the account to be managed
 - o **Password:** Password for the account
2. Click the **Save** button to save your secret and verify that it passes heartbeat.
3. Click the **Remote Password Changing** tab for the secret.
4. Click the Edit button.
5. Click to select **Privileged Account Credentials** in **Execute PowerShell**. The Privileged Account selector appears.
6. Click the **No Selected** Secret link.
7. Locate click on the **PowerShell Admin** secret.
8. Click the **Home** button to return to the dashboard.
9. In the **The following Secrets are available to be used in Custom Password Changing Commands and Scripts** section:
 1. Click the **No Selected Secret** link.
 2. Select your AD account secret for password changing.
 3. Click on the **Save** button.

Everything should now be configured for heartbeat and RPC on the Secret. Run **Heartbeat** (from the **General** tab in the Secret) to confirm that it works and run an RPC ** (from the **Remote Password Changing** tab of the secret) to confirm that it also works.

Errors

If you receive the "The term 'Set-ADAccountPassword' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again." error, install the AD-Domain-Services in Powershell. To do this start PowerShell as an administrator then run the following command:

```
Install-windowsfeature -name AD-Domain-Services -IncludeManagementTools
```

Additionally you may need to install the Remote Server Administration Tools for your version of Windows and then in PowerShell run:

Import-Module Servermanager

Secret Server supports Office 365 in version 8.8 and later. This does not require the advanced scripting add-on license. To perform heartbeat checks and remote password changes on secrets using the Office 365 password changer for user accounts, follow the steps below:

Procedure

Note: This applies to both the Secret Server Web server or distributed engines.

1. Run Windows PowerShell as an admin. This opens an elevated Windows PowerShell command prompt.
2. Run this command: `Install-Module -Name AzureAD`.
3. Recycle the application pool.

Note: These steps are required once on the subject computer, not every time you connect. However, you may want to update the module periodically as a security best practice using the command: `Update-Module -Name AzureAD`

Troubleshooting

1. Uninstall AzureAD using command `remove-module AzureAD`.
2. Reinstall using the above procedure.
3. Ensure the Secret Server application pool setting *Load User Profile* is set to "True".
4. Recycle the application pool.

As of version 8.6, Secret Server supports password changing for Salesforce.com accounts.

The password changer can be enabled on secrets that were created using the default Web Password Secret template or any custom template that is configured to use the Web User Account Password type.

The Secret Server Web server's outbound IP Address must be added to the IP Address white list for your Salesforce.com organization. Please refer to the Salesforce.com documentation for instructions on how to set this up. See [Restricting Login IP Ranges for Your Organization](#).

In cases where this is not set up correctly, you may see the follow error in the Remote Password Changing logs:

Login failed: LOGIN_MUST_USE_SECURITY_TOKEN: Invalid username, password, security token; or user locked out.

Please note:

- Secret Server can only communicate to the following Salesforce default Login URLs: <https://test.salesforce.com> and <https://login.salesforce.com>.
- Having the domain URL in the secret will not work and will throw this exception: Login failed: INVALID_LOGIN: Invalid username, password, security token; or user locked out. Only those two URLs work.
- There are three required Salesforce configurations:
 - Go to **Setup > Administration > Users > Profile**. Choose the user profile. Make sure that **Enabled API** is checked. This option is not available in all versions of Salesforce. Other versions will not have this enabled by default. Please see this "[Enable API" not available](#) article. If this setting is not enabled in salesforce you will get one of these errors: ERROR: Secret 'Salesforce Test' (Id = 1063) on Site 'EARTH' returned (LoginFailed). Exception: Login failed: API_DISABLED_FOR_ORG: API is not enabled for this Organization OR Partner, System.Web.Services.Protocols.SoapException: API_DISABLED_FOR_ORG: API is not enabled for this Organization or Partner.
 - Configure network access and allowlist the distribute engine or Secret Server IP address. If this is internal, use the public IP address.
 - Go to **Setup > Company Settings > My Domain**. Edit my domain settings and make sure that **Prevent login from <https://login.salesforce.com>** is unchecked.

You can enable Secret Server to perform heartbeat and change passwords on SAP accounts by following the procedures as indicated below (Premium Add-on or Enterprise Plus Edition Only).

First, create a new privileged SAP account administrator secret, typically for the SAP or DDIC account that is used to log on to SAP for administrative tasks. Select the **SAP Account** template and enter all required information to create the new SAP account administrator secret. By default, the **Instance Number** will be 00 and the **Client Number** will be 001.

Note: the default **System ID** for SAP is also NSP.

Second, create the account you are planning to change. Follow the same method as before and enter the current account password in the password field.

Third, in your new SAP account administrator secret, set the privileged account on the "Remote Password" changing tab.

Note: for an account to have its password changed, even a privileged account changing its own password, it requires permissions in SAP.

For Secret Server 8.8.000000 and Higher

Download [SAP .Net Connector 3.0](#) and install it using the following procedure:

1. Navigate to service.sap.com/connectors.
2. Enter your credentials for the SAP Marketplace.
3. Click on SAP Connector for Microsoft .NET.
4. Download the .NET 4.0 Option with the proper bitness for your application pool (64-bit mode for most customers).
5. Install the downloaded file.
6. Copy the `sapnco.dll` and `sapnco_utils.dll` files into the bin folder of your web application.
7. Recycle the application pool.

Once these steps are complete, heartbeat and password changing should be working.

Note: Accounts can change their own SAP passwords just once per day. This is a restriction in the SAP software that cannot be changed. If an account needs its password to change more than once a day, use a privileged account to perform the reset.

If performing a Heartbeat on an SAP Secret fails with the error, `Exception: PASSWORD_EXPIRED`, it most likely means an administrator has reset the SAP account's password, and the account must log in and change its own password in SAP.

For Secret Server 8.7.000000 and Below

1. Change your Secret Server application pool to run in 32-bit mode.
2. Download the SAP GUI version 720 from the [SAP Community website](#).
3. Extract the downloaded ZIP file. Depending on the version, the extracted download will have a GUI or Frontend Tools directory.
4. Copy that directory over to the machine running Secret Server.
5. Run the installer inside the directory. The install should take only a couple of minutes.

Overview

This document enables you to password change SQL accounts using a privileged account. This allows you to take over accounts without knowing their password.

Task 1: Creating an Account

1. Open SQL Server Management Studio.
2. Connect to your database server.
3. Expand the root-level security folder.
4. Right click the **Logins** folder.
5. Click **New login**.
6. Name the account.
7. Click **SQL Authentication**.
8. Go to Secret Server.
9. Create a secret using the **SQL Server Account** template. Give it the same username as the login name you are creating.
10. For best security, click the **Generate** button on the secret password field.
11. Copy the generated password to the account creation wizard in SQL Server Management Studio.
12. Click **OK**.
13. Save your secret.

Task 2: Assigning Permissions

1. Right click on the SQL login and click **Properties**.
2. Select **Securables** in the left column.
3. Select **Grant** for **Alter any login**.

Task 3: Using the Account

1. In Secret Server, select the SQL account secret you are going to have represent your new privileged account.
2. Select the **Remote Password Changing** tab and click **Edit**.
3. Click the **Change Password Using** selection button and select **Privilege Account Credentials**.
4. Click the **No Selected Secret** link.
5. Find and select the secret created for the privileged account in the first task.
6. Click the **Save** button.
7. Click the **Change password remotely** button.
8. Provide or generate a new password.
9. Click the **Change** button. You have now successfully changed a SQL account password using a privileged account.

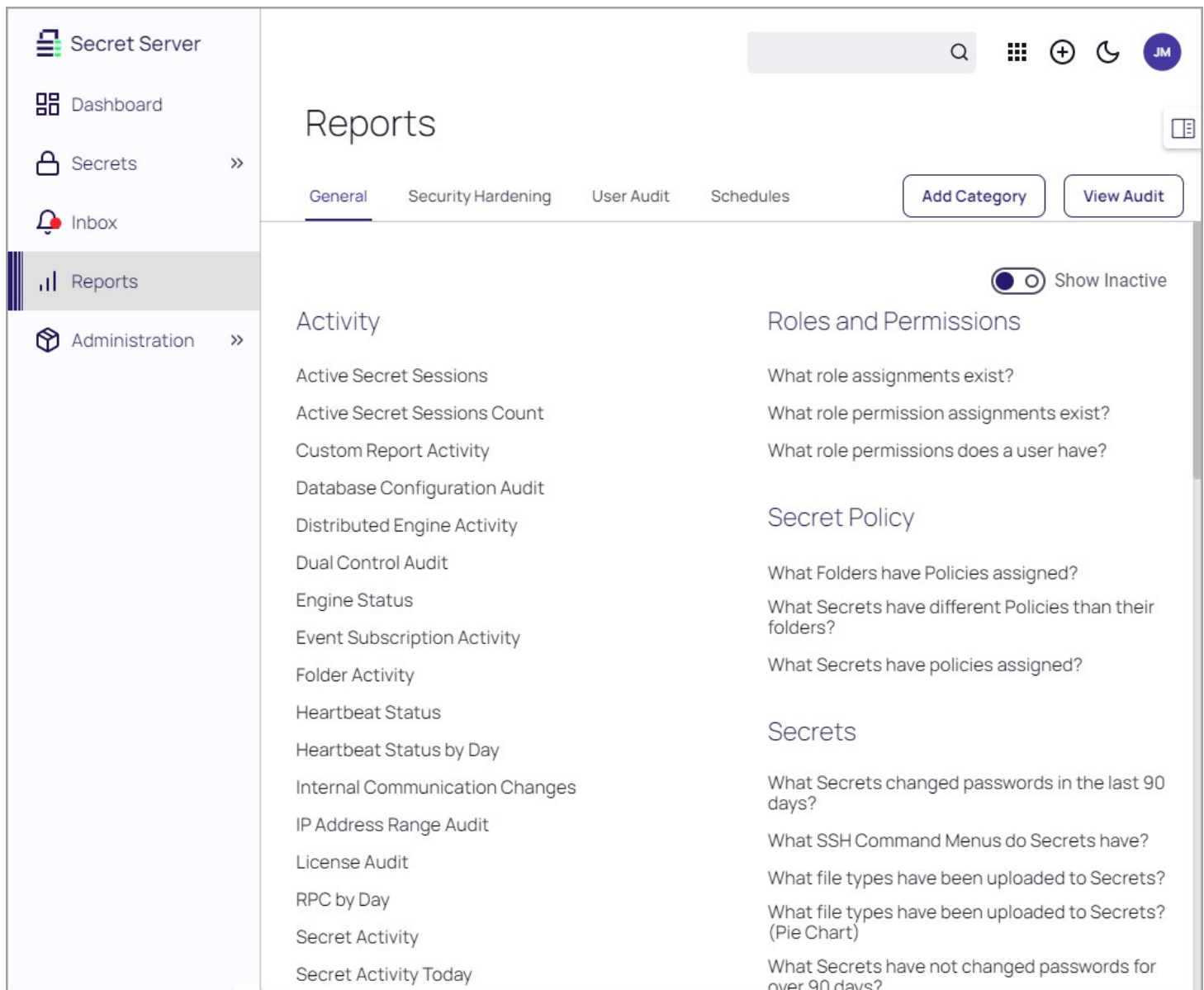
Note: You can also assign the account for use by multiple secrets by creating a secret policy and applying that policy to a folder.

Reports

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The reporting interface comes with a set of standard reports. These reports include a variety of 2D and 3D charting and graphing components and a full grid of data. Some of the reports are purely data detailed and have no charts. You can also create your own reports based on any Secret Server data, such as user, audit, permissions, and folders. You can create report categories to aid in the organization of your reports. Reports can be arranged to provide access to auditors and meet compliance requirements. These reports can be accessed in the **General** tab on the **Reports** page.

Figure: Reports Page



The *Security Hardening Report* checks aspects of Secret Server to ensure security best practices are being implemented. While Secret

Server runs with all the items failing, administrators should be aware of possible security issues within an installation. For details on this, see [Reports Security Hardening Tab](#).

The User Audit Report shows all secrets accessed by a user during a specified period.

Secret Server includes many pre-configured reports that you can run or use as templates for creating custom reports. Below are the reports shipped with current release of Secret Server:

Note: Unless otherwise designated, reports listed are available in all editions. However, older releases may not include all reports listed here.

Activity

- Active Secret Sessions
- Active Secret Sessions Count
- Custom Report Activity
- Database Configuration Audit
- Distributed Engine Activity
- Dual Control Audit
- Engine Status
- Event Subscription Activity
- Folder Activity
- Heartbeat Status
- Heartbeat Status by Day
- Internal Communication Changes
- IP Address Range Audit
- License Audit
- RPC by Day
- Secret Activity
- Secret Activity Today
- Secret Activity Yesterday
- Secret Template Activity
- Session Recording Errors
- Unlimited Administrator behavior
- Users Activity

Discovery Scan

- Discovery Scan Status
- GCP Discovery: What Instances do Service Accounts have access to?
- What computers have been successfully scanned?
- What computers in Active Directory no longer exist?
- What computers that exist have not been successfully scanned?
- What Secrets are pending import by Discovery?
- What Secrets failed to import by Discovery?

Folders

- ConnectWise API Config
- What folder permissions exist for groups?
- What folder permissions exist?
- What folders can a user see?
- What folders can all users see?

Groups

- Group Membership
- Group Membership By Group

Legacy Reports

- Secret Expiration Health
- Secret Server Usage
- Secret Template Distribution
- Top Ten Viewers

Password Compliance

- Secret Password Compliance Statuses
- What Secrets Do Not Meet Password Requirements?

Report Schedules

- Report Schedules

Roles and Permissions

- What role assignments exist?
- What role permission assignments exist?
- What role permissions does a user have?

Secret Policy

- What Folders have Policies assigned?
- What Secrets have different Policies than their folders?
- What Secrets have policies assigned?

Secrets

- Secret Count per Site
- Secret Dependency Failures
- Secret Dependency Not Run
- Secret Dependency Overview
- Secret Dependency Status
- Secret Permissions Mismatch.
- Secret Templates without an expiration field?
- Secrets Failing Heartbeat
- Secrets Pending Heartbeat
- Secrets with Failed Password Change
- What file types have been uploaded to Secrets?
- What file types have been uploaded to Secrets? (Pie Chart)
- What Secret permissions exist for a group?
- What Secret permissions exist for a user?
- What Secret permissions exist?
- What Secrets are expiring this week?
- What Secrets can a user see?
- What Secrets can all users see?
- What Secrets changed passwords in the last 90 days?

- What Secrets Do Not Have Distributed Engines?
- What Secrets don't require approval?
- What Secrets have been accessed by a user?
- What Secrets have been accessed by an impersonated user?
- What Secrets have been accessed?
- What Secrets Have Distributed Engines?
- What Secrets have Expiration?
- What Secrets have failed Heartbeat?
- What Secrets have not changed passwords for over 90 days?
- What Secrets require approval?
- What Secrets require Comments?
- What SSH Command Menus do Secrets have?

System Reports

- Folder Permissions Report
- Folder Secrets Report
- Group Lookup Report
- Permission Lookup Report
- Privileged Behavior Analytics Configuration Activity
- Role Permissions Report
- User Access Report

User

- Active Users Custom Report
- Failed login attempts
- Secret Template Permissions by User
- Session resumed
- What SSH Command Menus do users have access to?
- What users have had an admin reset their password?
- Who hasn't logged in within the last 90 days?

Note: You can find additional reports in the [Custom Report Gallery](#).

There are two ways to create a Report. From the Reports Edit page, click the **Add New** link at the bottom of a Report Category. Or alternatively, from the Reports View page, click the **Create it** link at the bottom of that page.

Creating a Custom Report

1. Click the **+** icon on the right side of the **Reports** menu item. The Report Edit page appears:

The screenshot shows the 'Report Edit' page with the following fields and controls:

- Report Name:** A text input field.
- Report Description:** A larger text input field with a vertical scrollbar.
- Report Category:** A dropdown menu with the text '- Select Report Category'.
- Chart Type:** A dropdown menu with the text 'None'.
- Page Size:** A dropdown menu with the text '15'.
- Use Database Paging:** A checkbox that is checked.
- Report SQL:** A large text area with a vertical scrollbar, containing the number '1' in the first line.

At the bottom of the form, there are three buttons: **Save** (with a floppy disk icon), **Preview** (with an eye icon), and **Cancel** (with an 'X' icon).

Below the form, there are two links: [Dynamic SQL Parameter KB Article](#) and [Show Secret Server SQL database information](#).

2. Type the report name in the **Report Name** text box. This is the name that is displayed on the Reports page as a link underneath its containing category.
3. Type a description in the **Report Description** text box. This is displayed in the Report View page. It is also used as the Tooltip for the Report name on the Reports page.

4. Click the **Report Category** dropdown list to select the category the report will appear in on the Reports page.
5. Click the **Chart Type** dropdown list to select the type of chart to use for displaying the results. If set to None, a grid displays.
6. Click the **Page Size** dropdown list to select the page size limit for the data displayed in the grid.
7. Click to select the **Use Database Paging** check box if desired. See [Database Paging](#).
8. Paste your script in the Report SQL text box. See [Report SQL Scripts](#). Our completed report looks like this:

Report Edit

[Report Definition](#) [Report Preview](#)

Report Name Active Users Custom Report

Report Description This displays a user list with all active user activity on view and returns a user id. This defaults to the current logged in user.

Report Category User

Chart Type None

Page Size 15

Use Database Paging

Report SQL

```
1 SELECT
2   tau.UserIdAffected,
3   tau.[Action],
4   tau.Notes,
5   tau.DateRecorded,
6   tau.IpAddress,
7   tau.MachineName,
8   tau.DatabaseName,
9   tu.UserId,
10  tu.UserName
11 FROM tbAuditUser tau INNER JOIN tbUser tu ON tau.UserId=tu.UserId WHERE tu.UserId=#USER
```

[Dynamic SQL Parameter KB Article](#)
[Show Secret Server SQL database information](#)

9. (optional) Click the **Preview** button to see your report before creating it. Our preview looks like this:

Report Edit

Report Definition [Report Preview](#)

Active Users Custom Report

This displays a user list with all active user activity on view and returns an user id. This defaults to the current logged in user.

User

Show Inactive Users

[▶ Update Report](#)

Save To File | Show All < 1 to 15 of 194 >

USERID AFFECTED	ACTION	NOTES	DATERECORDED	IPADDRESS	MACHINENAME	DATABASENAME	USERID	USERNAME
6	LOGIN SUCCESS		6/25/2019 02:33 PM	192.168.113.18	QA-CUST- SQL-01	SS_Playground	6	
6	LOGIN FAILED	Attempted to use session key from a different IP address. Expected: 192.168.113.18	7/16/2019 03:36 PM	192.168.113.8	QA-CUST- SQL-01	SS_Playground	6	
6	LOGIN SUCCESS		7/16/2019 03:37 PM	192.168.113.8	QA-CUST- SQL-01	SS_Playground	6	

10. Click the **Report Definition** tab to return to your editing.

11. Click the **Save** button. The new report's page appears:

Reports > Active Users Custom Report

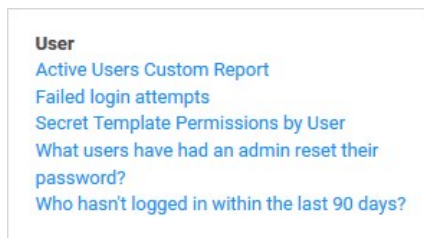
Filter Schedule Edit Delete View Audit Email Report

194 Items

USERIDAFF...	ACTION	NOTES	DATERECOR...	IPADDRESS	MACHINEN...	DATABASEN...	USERID	USERNAME	
6	LOGIN SUC...		6/25/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Attempted t...	7/16/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN SUC...		7/16/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Attempted t...	7/30/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN SUC...		7/30/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Attempted t...	8/7/2019 0...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN SUC...		8/7/2019 0...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Attempted t...	8/13/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGOUT		8/13/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Authenticati...	8/13/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN SUC...		8/13/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGIN FAIL...	Attempted t...	8/26/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGOUT		8/26/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGOUT		8/26/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	
6	LOGOUT		8/26/2019 ...	192.168.11...	QA-CUST-S...	SS_Playgrou...	6	[REDACTED]	

[Load More](#)

12. The new report now appears on the Reports page:



Editing Reports

To edit a report:

1. Click the **Reports** menu item. The Reports page appears, listing all the reports.
2. Click the name of the report, which is a link. That report's page appears.

3. Click the **Edit** button. The Report Edit page appears. See [Creating a Custom Report](#) for details about the parameters.

Note: The SQL script text cannot be edited for non-custom (built-in) reports.

Report SQL Scripts

Overview

The best way to create SQL scripts is to view existing ones and the Secret Server database structure. Click any existing report's link to arrive at its page. Then click the **Edit** button. The SQL appears in the Report SQL text box.

Note: Even though you are pressing the Edit button, you cannot edit non-custom reports. You can view their parameters, including their SQL script.

Dynamic Parameters

Reports support the embedding of certain parameters into the SQL so you can dynamically change the resulting data set. Another option available for custom reports is to apply a different color to returned rows dependent on certain conditions. For more information as well as examples, see the [Using Dynamic Parameters in Reports](#) topic.

Viewing Secret Server SQL Database Information

You can show Secret Server's SQL database information to assist with creating custom reports. By selecting the SQL Table from the list, the details of the table's columns display in a grid. Click the **Show Secret Server SQL database information** link to see the SQL Table list and SQL Table Columns grid. The link is also available on the Report Edit page.

You can click **Preview** button at the bottom of the page to see a preview of the chart. The resulting chart displays in the Report Preview section at the bottom of the page.

Database Paging

Database paging allows the database to load large reports more quickly. We recommend database paging if the query is expected to pull large amounts of data for the report. Implementing database paging may not work if the SQL query uses some keywords, including TOP, OPTION, INSERT, UNION, WITH, or aliases containing the word FROM.

Example queries:

- Works using database paging: `SELECT * FROM tbSecret WHERE NAME LIKE 'Test%'`
- Does not work using database paging: `SELECT TOP 10 * FROM tbSecret WHERE SecretName LIKE 'Test%'`

To delete or undelete a report.

- **Delete:** To delete a report, click the **Delete** button.
- **Undelete:** To undelete a report, you must navigate to the Reports Edit page as deleted reports are not visible on the Reports View page. On the Reports Edit page, click the **Show Deleted** button. This displays a Deleted Report category, which contains all the deleted reports. Either drag the report to a report category that is not deleted or click the report name to go into its Report View page. In there, click the **Undelete** button.

For details on the Show Deleted button, see [Deleting and Undeleting Reports](#).

- **Rearrange:** Any item with the icon can be dragged and dropped to a new location. Report categories can be moved anywhere within the page. Reports can be moved from one report category to another.
- **Create New:** Click **Create Report Category** and specify a category name and description on the following page. Note that the Report Category Description is used as the tooltip for the report category on the Reports View page.
- **Delete:** Click the icon next to the report category name. This deletes all the reports in the category. To undelete the reports, see [Deleting and Undeleting Reports](#).
- **Edit:** Click the icon next to the report category name to change the name or description of the category.

Reports General Tab

See [Built-In Reports](#) for the most up-to-date list of reports included.

The reports are listed under the report categories. To view a report, click on its name. This takes you to the **Report View** page.

You can view a record of all the actions performed on reports by clicking on the **View Audit** button.

For details on the **Edit** button, see [Creating and Editing Reports](#).

The **Create it** link is a shortcut for creating a new report.

You can adjust the look of the Reports View page. The report categories as well as the reports can be rearranged on the page. To do this, click **Edit** on the Reports page.

Reports Security Hardening Tab

The Security Hardening Tab configures aspects of Secret Server to ensure security best practices are being implemented. While Secret Server runs with all the items failing, administrators should be aware of possible security issues within an installation. Below is an explanation of the different features:

Configuration Section

- **Allow Approval for Access from Email:** This is a convenience option that allows users to approve or deny a secret access request by clicking a link in the request email sent by Secret Server. Allow Approval From Email does not require a user to authenticate with Secret Server when approving access to a secret. This can be a security concern if the approver's email account becomes compromised. Turn Allow Approval From Email off to get a pass result. Secret Server.
- **Browser AutoComplete:** Browser AutoComplete allows Web browsers to save the login credentials for the Secret Server login screen. These credentials are often kept by the Web browser in an insecure manner on the user's workstation. Allowing AutoComplete also interferes with the security policy of your Secret Server by not requiring the user to re-enter their login credentials on your desired schedule. To prevent the AutoComplete feature, disable the Allow AutoComplete option on the Configuration page.
- **File Attachment Restrictions:** File attachment restrictions allows administrators to configure what kind of file attachments can be uploaded to secrets. This helps protect users from being tricked into downloading a malicious secret attachment. The file extension and maximum file size can be specified, such as:

*.7z, *.bmp, *.ca-bundle, *.cer, *.config, *.crt, *.csr, *.csv, *.dat, *.doc, *.docx, *.gif, *.gz, *.id-rsa, *.jpeg, *.jpg, *.json, *.key, *.lic, *.p7b, *.pcf, *.pdf, *.pem, *.pfx, *.pkey, *.png, *.ppk, *.pub, *.tar, *.tif, *.tiff, *.tpm, *.txt, *.vdx, *.vsd, *.vsdx, *.xls, *.xlsx, *.xml, *.zip

This security check will fail if the file attachment restrictions is not enabled. This check will return warnings if a potentially dangerous file extension is allowed, maximum file size is not specified, or maximum file size is greater than 30 MB.

- **Force Password Masking:** Password masking prevents over-the-shoulder viewing of your passwords by a casual observer (when masked, passwords show as *). To activate this option, click to select the **Force Password Masking** option on the **Configuration** page.
- **Frame Blocking:** Frame blocking prevents the Secret Server site from being placed in an iFrame. This is to prevent clickjacking attacks. There may be legitimate reasons for placing Secret Server in a frame, such as embedding the UI in another site. To turn frame blocking on, enable the setting under the Security tab in Configuration.
- **Login Password Requirements:** Login passwords can be strengthened by requiring a minimum length and the use of various character sets. A minimum password length of 8 characters or longer is recommended. In addition, all character sets (lowercase, uppercase, numbers and symbols) are required to get a pass result. Turn on these login password settings on the Configuration page.

- **Maximum Login Failures:** The maximum number of login failures is the number of attempts that can be made to login to Secret Server as a user before that user's account is locked. A user with user administration permissions is then required to unlock the user's account. The maximum failures allowed should be set to 5 or less to get a pass result. Change the "Maximum Login Failures" settings on the Configuration page. Secret Server
- **Remember Me:** Remember Me is a convenience option that allows users to remain logged in for up to a specific period. This setting can be a security concern as it does not require re-entry of credentials to gain access to SS. Turn Remember Me off on the Configuration page to get a pass result. It must be set to be valid for 1 day or less to not get a fail result.
- **Secure Session and Forms Auth Cookies:** Cookies contain potentially sensitive information that can allow users to log onto application. By default, cookies are not marked with the secure attribute. That is, **they are transmitted unencrypted when a user accesses Secret Server through HTTP instead of HTTPS.**

Note: For more information about how to secure your cookies, see [Secure ASP Session and Forms Authentication Cookies](#) (KB).

- **Web Service HTTP Gets Allowed:** Web service HTTP get requests are allowed. Allowing HTTP GET requests allows REST-style calls to many Secret Server Web service methods. This can be a security concern because simply clicking a link to the Web service, created by a malicious user, would cause it to be executed.
- **Zero Information Disclosure Error Message:** Replace all error messages with a custom "contact your admin" message. Error messages can be very helpful when diagnosing installation and configuration issues. However, having errors displayed to a potential attacker can provide him or her with the critical information they need to perform a successful attack.

Database Section

- **SQL Account Using Least Permissions:** Use the fewest Secret Server permissions as possible in the SQL Account used to access the database. We recommend using a least permission approach where the account only has dbOwner. See [Installing and Configuring SQL Server](#).
- **SQL Server Authentication Password Strength:** SQL Server authentication requires a username and password. The password must be a strong password to get a pass result. Strong passwords are 8 characters or longer and contain lowercase and uppercase letters, numbers and symbols. The SQL Server authentication credentials in use can be changed by going to the installer (installer.aspx) and changing them on Step 3. A pass result is also given if Windows authentication is used to authenticate to SQL Server.
- **SQL Server Authentication Username:** The SQL Server authentication username should not be obvious. The use of "sa", "Secret Server" or "secretserver" triggers a fail result. The SQL Server authentication credentials in use can be changed by going to the installer (installer.aspx) and changing them on Step 3. A pass result is also given if Windows authentication is used to authenticate to SQL Server.
- **Windows Authentication to Database:** Windows authentication takes advantage of Windows security to provide secure authentication to SQL Server. The SQL Server authentication options can be changed by going to the installer (installer.aspx) and changing them on Step 3. Please see [Installation](#) for instructions on configuring Windows authentication to SQL Server.

Environment Section

- **Application Pool Identity:** The Application Pool identity GAMMA\ss_iis_svc appears to be a member of the administrators group on the system. This puts the system at risk by giving more access than necessary.
- **DPAPI or HSM Encryption of Encryption Key:** Encrypt your Secret Server encryption key, and limit decryption to that same server. Data Protection API (DPAPI) is an encryption library that is built into Windows operating systems. It allows encryption of data and configuration files based on the machine key. Enabling DPAPI Encryption in Secret Server protects the Secret Server encryption key by using DPAPI, so even getting access to the Secret Server encryption key is not enough to be useful—the machine key is required. If you enable this option, back up your encryption key first, as a DPAPI encrypted file can only be used by the machine it was encrypted on.

SSL Section

Note: SSL needs to be running with at least a 128-bit key size to get a pass result. A warning result indicates your key size is less than 128 bits. A fail result indicates you are not using SSL.

- **Require SMTP SSL:** SMTP SSL is required to ensure that all communication between Secret Server and the email server is encrypted. Enable the "Use SSL" option in Secret Server to get a pass result.
- **Require SSL:** Secure Sockets Layer (SSL) is required to ensure that all communication between the Web browser and Secret Server is encrypted and secure. Once the SSL certificate is installed, Force HTTPS/SSL in Configuration to get a pass result. Please see the [Installing a Self-Signed SSL/HTTPS Certificate](#) Knowledge Base article for instructions.
- **SSL/TLS Hash:** Check the digest algorithm of the certificate. If the algorithm is SHA1, this check returns a warning because SHA1 is being phased out. If the digest algorithm is MD2, MD4, or MD5, the check will fail because they are not secure. SHA256, SHA384, and SHA512 will pass. This check fails if Secret Server cannot be loaded over HTTPS.
- **SSL/TLS Key:** Check the key size of the HTTPS certificate used. If it is RSA or DSA, the key must be at least 2048-bit to pass. If the signature algorithm of the certificate is ECDSA, the key size must be at least 256-bit to pass. If the algorithm of the certificate is unknown, the result shows "unknown. This check fails if Secret Server cannot be loaded over HTTPS.
- **SSL/TLS Protocols:** Check for legacy SSL or TLS protocols, which should not be used in a secure environment. If the server accepts SSLv2 or SSLv3 connections, this check will fail. SSLv2 is not considered secure for data transport, and SSLv3 is vulnerable to the POODLE attack. If this server does not support TLSv1.1 or TLSv1.2, this check will give a warning because they are recommended. The SSL certificate used may affect what protocols can be used, even if they are enabled. This check will fail if Secret Server cannot be loaded over HTTPS.
- **Using HTTP Strict Transport Security:** HTTP Strict Transport Security (HSTS) is an additional security layer for SSL. HSTS allows SS, Password Reset Server, or Group Management Server to inform browsers that it should only be accessible over HTTPS. With this setting enabled, visitors are automatically are redirected by their browser to the HTTPS-enabled site.

Reports User Audit Tab

User Audit Reports show all secrets accessed by a user during a specified period. For a more detailed explanation, see [Built-in Reports](#).

If there are requirements around protecting potentially personally identifying information when running reports or viewing recorded sessions, you can enforce that another user has authorized you by enabling dual control for a secret or Report. When enabled a user in the approver group must enter in their credentials before a report or session can be viewed.

Once the approver has entered their credentials, the resource can be accessed.

To configure dual control, perform the following steps:

1. Click **Administration > Diagnostics, Logs, Security > Dual Controls**.
2. On the Dual Controls page, click the **Create New** button.

The screenshot displays the 'Dual Controls' page in the Delinea interface. The left sidebar contains navigation links: Secret Server, Dashboard, Secrets, Inbox, Reports, and Administration. The main content area shows the 'Dual Controls' page with a breadcrumb 'Reports > Dual Controls'. The page title is 'Dual Controls', and there are tabs for 'Dual Controls' and 'Audit'. A 'Create New' button is visible in the top right. Below it is a 'Show Inactive' toggle. A table lists one item:

TYPE ↑	NAME	ACTIVE	⚙️ ↓
	Access User Audit Report	✔️	

3. On the **Create New** page in the **Type** drop-down box, select the report type you wish to modify.

Create New

Type: Access Report

Report: Access Report

Active: Secret Session Access

Valid Approvers: Create Report

All Secrets Session Access

No users or groups have been selected

Add: All

Search for groups or users

- (lol)
- 1778
- 2003tasktest
- 0007T-11

Cancel Create New

- In the drop-down list below **Add**, click the box next to each party you'd like to add as a Valid Approver. Their identities will automatically appear in the **Valid Approvers** field.

Create New

Active

Valid Approvers

Items (3)

Cert Publishers	Remove
Cesar Velez	Remove
Developers	Remove

Add

All

- demo 2
- Denied RODC Password Replication Group
- Developers
- DevOps1
- Discovery
- Distributed COM Users
- Dns + Admins

When you are finished, click the **Create New** button. The Dual Control page appears, displaying the parameters you have set.

Dual Control


Require approval from at least one other user to access this resource.


Type	Access Report
Report	Active Secret Sessions
Active	Yes
Valid Approvers	Cert Publishers, Developers, Cesar Velez

1. Click the **Reports** menu item. The Reports page appears:

2. Click the link for the desired report. Its page appears:

EXPIRATION DATE	FOLDER PATH	SECRET NAME	SECRET TEMPLATE
6/6/2022 11:38 AM	\\RPC/Heartbeat\Oracle Account\Engine	Oracle Account 02	Oracle Account
6/6/2022 11:38 AM	\\Secret Expiration	Secret Expiration 06	Active Directory Account
6/6/2022 11:39 AM	\\Secret Expiration	Secret Expiration 05	zzAD - 1 Day
6/6/2022 11:47 AM	\\Secret Expiration	Secret Expiration 19	Active Directory Account

3. Click the  button in the top right of the item table. The download popup appears:



Download

Would you like to download this table data as a CSV file?

Export

Date Format

Time Zone

4. Click the **Export** dropdown list to choose which data to download.
5. Click the **Date Format** dropdown list to choose the date format.
6. Click the **Time Zone** dropdown list to choose the time zone.
7. Click the **Download** button. The reports save as a CSV file to your browser's default download location.

Creating New Schedules for Reports

1. To create a schedule for a report, click the **Schedules** tab on the **Report View** page:

The screenshot shows the 'Reports > RPC by Day' interface. At the top, there are navigation tabs: 'View Report', 'Schedule' (which is underlined and highlighted in blue), and 'Audit'. To the right of these tabs are three buttons: 'Edit', 'Delete', and 'Email Report'. Below the tabs, there is a large green button labeled 'Create Schedule'. At the bottom left, it says '0 Items', and at the bottom right, there is a toggle switch labeled 'Include Deleted' which is currently turned off.

2. Click the **Create Schedule** button. The Report Schedule page appears:

The screenshot shows the 'Reports > RPC by Day > Schedules >' configuration page. It has the same navigation tabs as the previous page: 'View Report', 'Schedule', and 'Audit'. The 'Schedule' tab is active. Below the tabs are 'Edit', 'Delete', and 'Email Report' buttons. The main section is titled 'Report Schedule' and contains the instruction 'Define when the report should run.' The configuration fields are: 'Schedule Name *' (a text input field), 'Health Check' (a checkbox that is unchecked), 'Start recurring on' (a date field set to '3/10/2021' with a calendar icon and a time field set to '3:01 PM' with a clock icon), 'Schedule Type' (a dropdown menu currently set to 'Daily'), and 'Recur every' (a text input field followed by the word 'Days'). At the bottom right, there are 'Cancel' and 'Save' buttons.

3. Configure the report settings as listed in [Editing Schedule Settings](#)
4. Click the **Save** button. The page for the new report schedule appears:

Reports > RPC by Day > Schedules > Test

Detail History

Delete

Report Schedule [Edit](#)

Define when the report should run.

Schedule Name *	Test
Health Check	No
Start recurring on	3/10/2021 03:01 pm
Schedule Type	Daily

Recur every 1 Days

5. The report will now be saved for you, saving only one report at a time. If you want more saved or want it emailed to you according the the schedule:

1. Click the **Edit** link in the **Report Distribution** section. The section becomes editable:

Number of Saved Reports Save All

Format

Send Email

2. Either click to select the **Save All** check box or type a new number in the **Number of Saved Reports** text box. Remember, saving reports can use a lot of disk space.
3. Click the **Format** dropdown list to select the report format:
 - HTML: Save the report as an HTML file.
 - CSV: Save the report as a comma separated value file for importation into a spreadsheet.

Important: The CSV feature is part of the early release of Secret Server 10.11. The general release is not till April 12, 2021 (on-premises version) and April 12, 2021 (cloud version).

1. Click to select the **Send Email** check box to have the report emailed to you at the reporting interval. An email section appears:

The screenshot shows a configuration panel with the following sections:

- Send Email**: A checked checkbox with a green checkmark.
- Send Email With High Priority**: An unchecked checkbox.
- Selected groups (0)**: A text box containing the message "No users or groups have been selected".
- Report Subscribers**: A section containing:
 - Add groups**: A dropdown menu currently set to "All" and a search input field labeled "Search for gro".
 - A list of groups with checkboxes:
 - Access Control Assistance Operators
 - Account Operators
 - admin
 - Administrators
- Additional Email Recipients**: An empty text input field.

2. Click the **All** dropdown list in the **Report Subscribers** section to choose the domain to look for users and groups.
3. Click the check boxes next to the users or groups you want to send the report to in the **Report Subscribers** section. You can also search for the same in the provided search box at the top of the section. The users or groups appear in the Selected Groups text box.
4. Type any additional email addresses in the **Additional Email Recipients** text box.
5. Click the **Save** button.

Viewing Existing Report Schedules

1. To view existing schedules for a report, click **Schedule** on the Report View screen. A list of existing schedules for the report appear in the grid.
2. To view the details of a schedule, click the schedule name in the grid.
3. (Optional) Deleted schedules can be made visible by checking the **Show Deleted** box at the bottom of the grid.
4. Click the **View** link in the History column of the grid to view the history of all generated reports for that schedule.

Editing Schedule Settings

When viewing a report, click Schedule and then the name of the report schedule to modify it. The following configuration options are available:

- **Schedule Name:** This is the name of the schedule for the report. This name must be unique to the Secret Server installation.
- **Health Check:** This sends an email notification only when the report contains data.
- **Recurrence Schedule:** This specifies the schedule runs every X number of days, weeks, or months, with the option to specify days of the week or month as well. The date and time that the report schedule is effective can be specified in this section as well.
- **Save Generated Reports:** This saves the history of generated reports in the database for later viewing. Enabling this setting also allows you to specify the number of generated reports to save.
- **Send Email:** Secret Server sends an email containing the generated report every time the schedule runs. Enabling this setting also allows you to specify whether the email is sent with the high priority flag and a list of Secret Server users or groups that receive the generated report email. Add additional email recipients in the text box below the subscribers, separating recipients with a semi-colon.

The following configuration options appear if the report being scheduled contains at least one dynamic parameter in the SQL of the report:

- **User Parameter Value:** Value of the #USER parameter to set in the report when it is generated.
- **Group Parameter Value:** Value of the #GROUP parameter to set in the report when it is generated.
- **Start Date Parameter Value:** Value of the #STARTDATE parameter to set in the report when it is generated.
- **End Date Parameter Value:** Value of the #ENDDATE parameter to set in the report when it is generated.

Note: As version 7.0, Secret Server allows creation of Reports using custom SQL.

Reporting supports embedding certain parameters into the SQL to give the viewer controls to dynamically change the report. The supported parameters are:

Primary Parameters

#STARTDATE

This displays a calendar picker on view and returns a date. This defaults to beginning of the year and truncates the hours and minutes to 12:00 AM.

Example: display all users who have logged in after a certain date:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
  LEFT JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin > #STARTDATE
```

#ENDDATE

This displays a calendar picker on view and returns a date. This defaults the current day and truncates the hours and minutes to 11:59 PM.

Example: display all users who have logged on a certain date:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
  LEFT JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin > #STARTDATE
AND
  LastLogin < #ENDDATE
```

#USER

This displays a user dropdown list with all active users on view and returns an user id. This defaults to the current logged in user.

Example: display all audit entries for a certain user:

```
SELECT
  tau.UserIdAffected,
  tau.[Action],
  tau.Notes,
  tau.DateRecorded,
  tau.IpAddress,
  tau.MachineName,
  tau.DatabaseName,
  tu.UserId,
  tu.UserName
FROM tbAuditUser tau INNER JOIN tbUser tu ON tau.UserId=tu.UserId WHERE tu.UserId=#USER
```

#ORGANIZATION

This is an internal parameter that returns the current instance's organization code. This is only useful for Secret Server Online (a legacy product, which is *not* the same as Secret Server Cloud). Do not use this parameter in your reports for either Secret Server On-Premises or

Secret Server Cloud.

Note: As of Secret Server 7.8.000048 the #GROUP parameter is also available.

#GROUP

Displays a group dropdown list with all active groups on view and returns a group id. This defaults to the Everyone group.

Example: display the group details of the selected group:

```
SELECT
  GroupID,
  GroupName,
  Active
FROM tbGroup
WHERE GroupID = #GROUP
```

#FOLDERID

Displays a folder picker that shows all Folders and returns a folder id.

Example: Display secret names in a selected folder:

```
SELECT
  s.SecretName
FROM tbSecret s
WHERE s.Folderid = #FOLDERID
```

#FOLDERPATH

Displays a folder picker that shows all folders and returns the path of the folder.

Example: display folders that are child folders of the selected path:

```
SELECT *
FROM tbFolder f
WHERE FolderPath LIKE '%'+ #FOLDERPATH + '%'
```

#CUSTOMTEXT

Displays a text input where a user can put in arbitrary free text for searching.

Example: display secrets that have names that contain the text input:

```
SELECT *
FROM tbFolder f
WHERE FolderPath LIKE '%'+ #CUSTOMTEXT + '%'
```

Additional Parameters

The following additional parameters can be used to make your report more dynamic:

Parameters

Table: Additional Parameters

#ENDCURRENTMONTH	The last day of current month
------------------	-------------------------------

#ENDCURRENTYEAR	December 31st of the current year
#ENDPREVIOUSMONTH	The last day of the previous month at 11:59:59 PM
#ENDPREVIOUSYEAR	December 31st of the previous year
#ENDTODAY	End of today at 11:59:59 PM
#ENDWEEK	End of the current week (Sunday) at 11:59:59 PM
#ENDYESTERDAY	End of Yesterday at 11:59:59 PM
#STARTCURRENTMONTH	The first day of current month
#STARTCURRENTYEAR	January 1st of the current year
#STARTPREVIOUSMONTH	The first day of the previous month at 12:00 AM
#STARTPREVIOUSYEAR	January 1st of the previous year
#STARTTODAY	Beginning of today at 12:00 AM
#STARTWEEK	Beginning of the current week (Monday) at 12:00 AM
#STARTYESTERDAY	Beginning of yesterday at 12:00 AM

Example

For example, the following script would give you a list of all users who have logged on during the last calendar month:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
LEFT JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin BETWEEN #STARTPREVIOUSMONTH AND #ENDPREVIOUSMONTH
```

Note: As of Secret Server 7.8.000048, the #STARTWEEK and #ENDWEEK parameters are available.

Coloring Your Reports

Another option when creating reports is to include a Column in your SQL query called "Color" this will give the row that particular color. See [HTML Color Names](#).

For example, to show users who haven't logged in within 90 days in Red:

```
SELECT DisplayName
CASE
  WHEN LastLogin < GetDate() - 90 THEN 'Red'
  ELSE 'White'
END AS Color
FROM tbUser
```

To view auditing for all actions performed on a report, do the following:

1. Click Reports from the left navigation panel.
2. Click the report for which you wish to view an audit.
3. At the far right end of the page, click the button labeled **View Audit**.

The audit information for the record appears in a pop-up message.

Audit

Date Recorded	2022-06-06T13:38:33.783 (6/6/2022 09:38 AM)
Full Name	John Melody
Report	Changed90DaysReportName
Action	VIEW
Details	

OK

For more information, See [Built-in Reports](#).

On this page you see the graph, chart, grid, and more for the report. To see a grid representation of the report, click the **Show Data** link to expand that area. If there is no data, then no graph is visible and the text "There are no items" displays in the Show Data section.

Some reports use dynamic values like user, start date, and end date. Adjust these values to generate the report you need. Click the **Update Report** button to generate the new report.

The **Edit** button allows you to alter the report to fit your requirements. See the Creating and Editing a Report topic for details.

Roles

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Modeled after the role-based access control (RBAC) mechanism, role-based security (RBS) is Secret Server's method of regulating permission to system access. Each user and group must be assigned to a role. Secret Server ships with three roles: Administrator, User, and Read-Only User. Each role contains various permissions to match the job function of the user. With RBS, strict granular access to Secret Server is ensured. A list of role permissions and their descriptions can be found in the [Secret Server Role Permissions List](#).

You can assign multiple permissions to a role. For example, you could assign Administer Users, Edit Secret, Own Secret, and View Active Directory permissions to a role. That role can then be assigned to a user or group.

Note: The Unlimited Administrator permission allows the user to have unlimited administrator rights when Unlimited Administrator is enabled in the configuration. By default, it is disabled.

Note: to see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

To assign roles to a user, click the **Assign Roles** button on the main **Roles** page. Depending on which tab is selected, this page allows you either view the roles that are assigned to users or view the users that are assigned to roles. To change these settings, click the **Edit** button. Now select a role from the list and assign or unassign users to the role. In the **By User or Group** tab, you can select a user or group from the list and assign or unassign roles to them in the selectable list boxes.

You can create roles from the Roles page. To get to the Roles page, navigate to **Administration > Roles**. Click the **Create New** button to add the role.

To add or remove permissions to an existing role, click the role name of the role you wish to edit.

On this Role View page, permissions can be added and removed from the role by clicking the **Edit** button. Use the arrow buttons to move permissions into and out of the current role. If needed, a role can also be enabled or disabled from this page. If you have finished with your changes, you must click the **Save** button to have the changes take effect.

Overview

Secret Server uses role-based access control (RBAC) to regulate permissions. The roles are assigned to users or groups. A complete list of the permissions available to roles appears below:

Note: to see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

Adding a jumpbox route to a target secret: A user must have owner permissions on a secret to assign, change, or remove that secret's jump server route. Additionally, users are only able to pick from a list of routes where they have at least list permission on the first jump route server. Editing Jumpbox Routes: Users must have the "Administer Jumpbox Route" permission to create, edit, or deactivate jump server routes. Users with the "View Jumpbox Route" permission can view the details of all jump server routes in the Admin Jumpbox Route page, but they cannot make any changes.

Complete List

Access Offline Secrets on Mobile

Allows a user to cache their Secrets in the Secret Server mobile application for offline use. This permission does not automatically come with the Administrator role.

Add Secret

Allows a user to create new Secrets. The Add permission no longer include the role permission View Secret.

Add Secret Custom Audit

Allows a user to make a custom audit entry when accessing a Secret using the web services API.

Administer Active Directory

Allows a user to view domains, edit existing domains, delete domains, and add new domains. Also allows a user to force synchronization or set the synchronization interval.

Administer Automatic Export

The user can do everything the other automatic export permissions allow *and* edit the automatic export configuration.

Administer Backup

Allows a user to view and configure automated backups for Secret Server. Users with this role permission can change the backup path, disable backups, and set the backup schedule.

Administer Configuration

Allows a user to view and edit general configuration options. For example, a user with this role permission can turn on "Force HTTPS/SSL" and disable "Allow Remember Me".

Administer Configuration Proxying

Allows a user to view and edit SSH Proxy settings.

Administer Configuration SAML

Allows a user to view and edit SAML integration settings on the Login tab of Configuration settings.

Administer Configuration Security

Formerly "Administer Security Configuration," allows a user to view and edit security configuration options in Secret Server. Currently, these include enabling FIPS compliance mode and protecting the encryption key.

Administer Configuration Session Recording

Allows a user to view and edit session recording settings on the Session Recording tab of Configuration settings.

Administer Configuration Two Factor

Allows a user to change the configuration settings of the two factor authentication that are available for users logging into Secret Server.

Administer Configuration Unlimited Admin

Formerly "Administer Unlimited Admin Configuration," allows a user to turn on Unlimited Admin Mode. When this mode is enabled, users with the "Unlimited Administrator" role permission can view and edit all Secrets in the system, regardless of permissions. Note that you can assign "Administer Unlimited Admin Configuration" to one user and "Unlimited Administrator" to another user. This would require one user to turn on the mode and another user to view and edit secrets.

Administer ConnectWise Integration

Allows a user to view and edit configuration options for synchronizing with ConnectWise. This can be accessed through the "Folder Synchronization" link on the Administration page. Note that you need at least view access on the sync folder in order to set up or edit the ConnectWise integration.

Administer Create Application Accounts

Formerly "Create Application Account", allows a user to create application user accounts to be used exclusively for accessing Secret Server via the API.

Administer Create Users

Allows a user to create new local users in Secret Server, but not edit them once created.

Administer Custom Password Requirements

Allows a user to view and edit custom password requirements that can be configured under the Security tab for individual Secrets.

Administer Data Retention

Can manage audit data retention, such as editing and running now. This permission does not automatically come with the Administrator role.

Administer DevOps Secrets Vault Tenants

Add, remove, and edit DSV tenants that automatically synchronize with Secret Server on a schedule.

Administer Disaster Recovery

Allows a user to configure instances as data sources or replicas for Disaster Recovery. Also allows user to initiate or test Data Replication and view related logs and audits.

Administer Discovery

Allows a user to view and import computers and accounts that are found by Discovery.

Administer Distributed Engine Configuration

Allows a user to update the Distributed Engine configuration.

Administer DoubleLock Keys

Allows a user to view, edit, create, and disable DoubleLock keys. A DoubleLock key acts as a separate encryption key to protect your most sensitive secrets. This option allows users to access and use the "DoubleLocks" link on the Administration page.

Administer Dual Control

Allows a user to view, edit, create, and disable Dual Control settings for reports and recorded sessions.

Administer Event Subscriptions

Allows a user to view, edit and create event subscriptions.

Administer Export

Allows a user to view the export log. Also allows users to export Secrets to which they have access to a clear text, CSV file.

Administer Folders

Allows a user to view, edit, create, move, and delete folders. Users still need the relevant view, edit, and owner permissions on the folders to perform these tasks.

Administer Inbox

Administer notification settings for the inbox.

Administer Groups

Allows a user to view, edit, create, and disable groups. Also allows users to assign users to groups and remove users from groups.

Administer HSM

Allows a user to change configuration or disable the use of a Hardware Security Module (HSM).

Administer IP Addresses

Allows a user to create, edit, and delete IP Address Ranges. These ranges are used to restrict certain users to specific IP Addresses.

Administer Jumpbox Route

Allows a user to create, edit, or deactivate jump server routes.

Administer Key Management

Allows a user to enable, change, or disable the Key Management (Secret Server Cloud only).

Administer Languages

Allows a user to change the default language of Secret Server.

Administer Licenses

Allows a user to view, edit, install, and delete licenses.

Administer Lists

Add, remove, and modify lists and list contents in Admin > Lists.

Administer Metadata

Manage metadata fields and sections added to secrets and users in Secret Server.

Administer Nodes

Allows a user to view and edit server nodes and clustering settings.

Administer OpenID Connect

Allows a user to manage OpenID connections.

Administer Password Requirements

Allows a user to view and edit character sets and password requirements.

Administer Pipelines

Allows a user to create, edit, and remove event pipelines and event pipeline policies.

Administer Platform Integration

Allows a user to manage the Secret Server connection to the Delinea platform.

Administer Remote Password Changing

Allows a user to turn Heartbeat and Remote Password Changing on and off globally. Also allows users to create new password changers and install password changing agents on remote machines.

Administer Reports

Allows a user to view, edit, delete, and create reports. Also allows users to customize report categories.

Administer Role Assignment

Allows a user to view which users and groups are assigned to which roles. Also allows users to assign users and groups to different roles.

Administer Role Permissions

Allows a user to view, edit, create and delete roles. Also allows users to assign different permissions to each role.

Administer Scripts

Allows a user to view, edit, and add PowerShell, SQL, and SSH scripts on the Scripts Administration page.

Administer Search Indexer

Allows a user to view and edit search indexer options. These options control how searching in Secret Server works. For example, a user with this role permission could enable search indexing, which allows users to search on fields within a secret.

Administer Secret Policy

Allows a user to create and edit Secret Policies."

Administer Secret Templates

Allows a user to view, edit, disable, and create Secret Templates.

Administer Security Analytics

Allows a user to view and edit the settings for Privilege Behavior Analytics.

Administer Session Monitoring

Allows a user to view and terminate active launcher sessions.

Administer SSH Menus

Allows a user to edit and create SSH Menus, used in allowlisting commands that can be used on a SSH session.

Administer System Log

Allows users to view and clear the System Log, which shows general diagnostics information for Secret Server.

Administer Teams

Users can create, delete, and view all teams.

Administer Template Custom Columns

Allows a user to enable the "Expose for Display" setting of a Secret template field to make it available for use in Dashboard custom columns.

Administer Users

Allows a user to create, disable, and edit users in the system.

Note: This permission also allows a user to create and edit SDK/CLI rules.

Administer Workflows

Allows users to manage workflows (advanced access management).

Advanced Import

Allows a user to import Secrets from an XML file. Users with the this permission can import groups, folders, site connectors, sites, and secret templates, without having to create a secret. Users must have the Secret Server permissions needed for the objects listed in the XML.

Allow Access Challenge

Allows a user be challenged by Privileged Behavior Analytics if their behavior deviates from their normal behavior and meets certain requirements set by Privileged Behavior Analytics. Administrators do not have this permission by default.

Allow List Secret Access for Assigning Policy

Allows users with list access to a secret to assign policies. Users need the view permission if they do not have this one.

Approve Via Duo Push

Allow a user to approve access requests via Duo push notifications. Administrators do not have this permission by default.

Assign Pipelines

Allows the user to assign an event pipeline policy to secret policies, or folders.

Assign Secret Policy

Allows a user to assign Secret Policies to folders and secrets.

Browse Reports

The "Browse Reports" role allows access to reports restricted by permissions. Permissions are configurable at the category and report levels and share a similar inheritance model to secrets and folders. You can define users or groups with "view" or "edit" permissions for each category or report.

Note: Users with the existing "view reports" and "edit reports" roles are not restricted by the permissions set.

Bypass Direct API Authentication Restriction

Allows users to ignore the PreventDirectApiAuthentication advanced setting and log in via the API with a non-application account.

Bypass SAML Login

Allows a user to login with local account without using SAML.

Copy Secret

Allows a user to copy secrets when that user also has Own Secret role permission.

Create Root Folders

Allows a user to create new folders at the root level of the folder structure.

Deactivate Secret

Allows a user to mark secrets as deactivated.

Delete Secrets from Reports

Allows a user to run the delete Secrets action from a report.

Download Automatic Export

The user can view all of the automatic export tabs *and* download exports from cloud storage (cloud customers only).

Edit Secret

Allows a user to edit secrets. Note that they still require the "Edit" or "Owner" permissions on the individual secrets they are editing.

Erase Secret

Allows a user to permanently erase (as opposed to deactivate, which is reversible) a secret.

Expire Secrets from Reports

Allows a user to expire Secrets listed in a report."

Force Check In

Allows a user to force a secret that is checked out by another user to be checked in.

Own Group

Allows a user to be an owner of a group. This permission is in the default Group Owner role, which is automatically assigned when that user is set as owner of a group.

Own Secret

Formerly "Share Secret", allows a user to share secrets with other users. Also allows users to perform more advanced tasks on secrets of which they are "Owners", such as configuring expiration schedules, configuring the web launcher, converting secret template, and copying secrets (when a user also have the Copy Secret role permission.)

Own User

Allows the user to become a user owner, used to configure specific users without the Administer Users permission.

Personal Folders

Allows a user to have personal folder when the global personal folders configuration options is enabled.

Privilege Manager Administrator

Allows the user to have the "Administrator" role for Privilege Manager, giving full access to the system.

Privilege Manager Helpdesk User

Allows the user to have the "Help Desk" role for Privilege Manager, giving full access to approve or deny escalation requests.

Privilege Manager MacOS Admin

Allows the user to have the MacOS "Administrator" role for Privilege Manager, giving full access to the system.

Privilege Manager Windows Administrator

Allows the user to have the Windows "Administrator" role for Privilege Manager, giving full access to the system.

Privilege Manager User

Allows the user to have the "User" role for Privilege Manager, giving read and write permissions to most items, but not rights to modify security permissions. Administrators do not have this permission by default.

Rotate Encryption Keys

Allows a user to start a process that rotates the Secret encryption keys.

Run Automatic Export

The user can view all of the automatic export tabs and run the export manually by clicking the Run Export button.

Run Disaster Recovery Data Replication

Allows user to initiate or test Data Replication.

Session Recording Auditor

Grants access to the session recording of a secret to a user with at least "List Access" permission on the secret. Administrators do not have this permission by default.

Note: Users also need the "View Session Monitoring" permission to view the recordings in Secret Server.

Unlimited Administrator

Allows a user to view and edit all secrets in the system, regardless of permissions, when Unlimited Admin Mode is on. Note that another user with the "Administer Unlimited Admin Configuration" role permission would still need to turn this mode on.

Unrestricted by Teams

Users can view all users, groups, and sites, regardless of team affiliation. Essentially, teams do not exist for the users with this permission, and the Teams page is not available to them. The default user role has this permission.

User Audit Expire Secrets

Allows a user to view the "User Audit" report, which shows all secrets that have been accessed by a particular user in a specified date range. Also allows the user to force expiration on all these secrets, which would make Secret Server automatically change the password.

View About

Allows a user to view the "About" page from the Help menu, which links to external resources such as Technical Support and the Delinea blog.

View Active Directory

Allows a user to view, but not edit, the Active Directory settings in the system.

View Advanced Dashboard

Allows a user to view advanced dashboard. Without this permission, users will only be able to view basic dashboard.

View Advanced Secret Options

Allows a user to view the Remote Password Changing, Security, and Dependency tabs on a Secret they have access to.

View Automatic Export

The user can view all of the automatic export tabs.

View Backup

Allows a user to view, but not edit, the automated backup settings.

View Configuration

Allows a user to view, but not edit, general configuration settings.

View Configuration Proxying

Allows a user to view, but not edit, SSH Proxy settings.

View Configuration SAML

Allows a user to view SAML integration settings on the Login tab of Configuration settings.

View Configuration Security

Formerly "View Security Configuration," allows a user to view the security configuration of Secret Server.

View Configuration Session Recording

Allows a user to view session recording settings on the Session Recording tab of Configuration settings.

View Configuration Two Factor

Allows a user to view the configuration settings of the two factor authentication that are available for users logging into Secret Server.

View Configuration Unlimited Admin

Formerly "View Unlimited Admin Configuration," allows a user to view the Unlimited Admin Mode configuration. Also allows a user to view the Unlimited Admin Mode audit log.

View ConnectWise Integration

Allows a user to view, but not edit, the ConnectWise integration settings.

View Data Retention

Can view retained audit data. This permission does not automatically come with the Administrator role.

View Deleted Secrets

Allows a user to view Secrets that have been deleted in the system.

View DevOps Secrets Vault Tenants

View (not edit) the DSV tenants set to synchronize with Secret Server.

View Disaster Recovery

Allows a user to view configuration, logs and audits for Disaster Recovery.

View Discovery

Allows a user to view, but not edit, computers and accounts that are found by Discovery.

View Distributed Engine Configuration

Allows a user to view the Distributed Engine configuration.

View DoubleLock Keys

Allows a user to view which DoubleLock keys exist in the system.

View Dual Control

Allows a user to view configured Dual Control settings for reports and Secret sessions.

View Event Subscriptions

Allows a user to view event subscriptions.

View Export

Allows a user to view the export log of the system to see when users exported secrets. Does not allow a user to export.

View Folders

Allows a user to view, but not edit, folders in the system.

View Group Roles

Allows a user to see which groups and users are assigned to which roles. Does not allow a user to change these assignments.

View Groups

Allows a user to see which groups exist in the system. Also allows a user to see which users belong to each group.

View HSM

Allows a user to view the Hardware Security Module (HSM) configuration settings.

View IP Addresses

Allows a user to view IP Address Ranges that have been created to restrict access. Does not allow a user to edit these ranges.

View Jumpbox Route

Allows a user to view the details of all jump server routes in the Admin Jumpbox Route page but not make any changes.

View Key Management

Allows a user to view the Key Management settings (Secret Server Cloud only).

View Launcher Password

Allows a user to unmask the password on the view screen of secrets with a launcher. Typically, this includes Web Passwords, Active Directory accounts, Local Windows accounts, and Linux accounts.

View Licenses

Allows a user to view, but not edit, the licenses in the system.

View Lists

View lists and list contents in Admin > Lists.

View Nodes

Allows a user to view, but not edit, the Secret Server web server nodes.

View OpenID Connect

View OpenID Connect integration settings in the Configuration Login tab. This replaces the Delinea One equivalent.

View Password Requirements

Allows a user to view character sets and password requirements.

View Pipelines

Allows a user to view event pipeline policies and policy activities.

View Platform Integration

Allows a user to view the Secret Server connection to the Delinea platform.

View Remote Password Changing

Allows a user to view, but not edit, Heartbeat and Remote Password Changing settings.

View Reports

Allows a user to view, but not edit, reports. See "Browse Reports."

View Roles

Allows a user to view roles in the system. Also allows a user to see which groups are assigned to which roles.

View Scripts

Allows a user to view PowerShell, SQL, and SSH scripts on the Scripts Administration page.

View Search Indexer

Allows a user to view, but not edit, search indexer settings.

View Secret

Allows a user to only view which Secrets exist in the system.

View Secret Audit

Allows a user to view Secret Audit.

View Secret Password and Private Key History

Allows a user to see the history of passwords, private keys, or passphrases in both old and new UI.

View Secret Policy

Allows a user to view, but not edit, Secret Policies.

View Secret Templates

Allows a user to view, but not edit, Secret Templates.

View Security Analytics

Allows a user to view, but not edit, settings for Privilege Behavior Analytics.

View Security Hardening Report

Allows a user to view the Security Hardening Report.

View Session Monitoring

Allows a user to view active launcher sessions.

View Session Recording

Allows a user to view recorded sessions within Secret Server.

View SSH Menus

Allows a user to view existing SSH Menus, used in allowlisting commands that can be used on a SSH session.

View System Log

Allows a user to only view the System Log, which shows general diagnostics information for Secret Server.

View Teams

Users can view all teams. This is essentially a read-only Administer Teams.

View User Audit Report

Allows a user to view, but not edit, the User Audit Report.

View Users

Allows a user to view which users exist in the system.

View Workflows

View (not edit) workflows used for multi-tier secret-access approvals and secret erase requests.

Web Services Impersonate

Allows a user to send an approval request to act as another user within their organization when accessing Secret Server programmatically. Administrators do not have this permission by default.

Secret Checkout

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The Secret Server *checkout* feature forces accountability on secrets by granting exclusive access to a single user. If a secret is configured for check out, a user can then access it. If **Change Password on Check In** is turned on, after check in, Secret Server automatically forces a password change on the remote machine. No other user can access a secret while it is checked out, except unlimited administrators. This guarantees that if the remote machine is accessed using the secret, the user who had it checked out was the only one with proper credentials at that time.

Note: The exception to the exclusive access rule is unlimited administrators. If Unlimited Administration is enabled, users with Unlimited Administrator role permission can access checked out secrets.

Note: Secrets with a doublelock cannot be configured for check out.

Each secret must be individually set to require check out:

1. From the **Secret View** page, click the **Security** tab to modify a secret's **Check Out** setting.
2. You must configure RPC before **Change Password on Check in** can be set.
3. Enable **Require Check Out** to force users to check out the secret before gaining access.
4. Enable **Change Password on Check In** to have the password change after the secret is checked in.

Overview

In addition to changing the password on check in, secret owners can also specify administrator-created PowerShell scripts, called *hooks*, to run before or after checkout and check in. These are accessed from the **Hooks** tab of the secret, which only shows if checkout is enabled and PowerShell scripts have been created by an admin.

To specify a before- or after-checkout hook, click **Create New Hook** and specify the following settings:

- **Before/After:** Whether the PowerShell script should run before or after the event action.
- **Event Action:** The hook runs at either check in or checkout.
- **Name:** A descriptive name for the hook.
- **Description:** An extended description for the purpose of the hook.
- **PowerShell Script:** Administrator-created PowerShell script to run.
- **Arguments:** Any command line arguments to pass to the PowerShell script.
- **Stop on Failure:** If enabled, Secret Server prevents the event action if the script returns an error. For example, if "Stop on Failure" is selected for a checkout action, then Secret Server prevents the user from checking out the secret if the script fails.
- **Privileged Account:** If needed, the script can run as another secret's identity.

Checkout User Variables for Scripts

Checkout user variables for scripts are special code variables that return information about the user or automated task making the checkout request, rather than system or secret information. For example, the \$USERNAME variable returns one or more user IDs related to a specific secret, whereas the \$SECRETSERVERUSERID checkout user variable returns the user ID of the logged-on user or automated task.

Note: These variables may also be useful for Active-Directory-related scripts.

The variables are:

Table: Checkout User Variables for Scripts:

\$SECRETSERVERUSERID	Logged-on user's ID	-1
\$SECRETSERVERUSERNAME	Logged-on user's name	"System"
\$SECRETSERVERDISPLAYNAME	Logged-on user's display name	"System"
\$SECRETSERVEREMAILADDRESS	Logged-on user's email address	Empty string

[Unexpected Link Text](#)

Note: You can find the regular "system" variables in the [Editing Custom Commands](#) subsection of the Custom Password Changers section.

To configure password checking on check in, navigate to the **Remote Password Changing Administration** page and set **Enable Password Changing on Check In**. If RPC is turned off, enable it before configuring checkout. Once RPC and checkout are enabled, secrets can be configured for interval that specifies how long a user has exclusive secret access.

Remote Password Changing Configuration

Enable Remote Password Changing	Yes
Enable Password Changing on Check In	Yes
Check Out Interval	30 minutes
Enable Heartbeat	Yes

[← Back](#) [✎ Edit](#) [✎ Configure Password Changers](#) [⚙️ Configure Dependency Changers](#)

[🏠 Distributed Engine Configuration](#) [📄 View Audit](#)

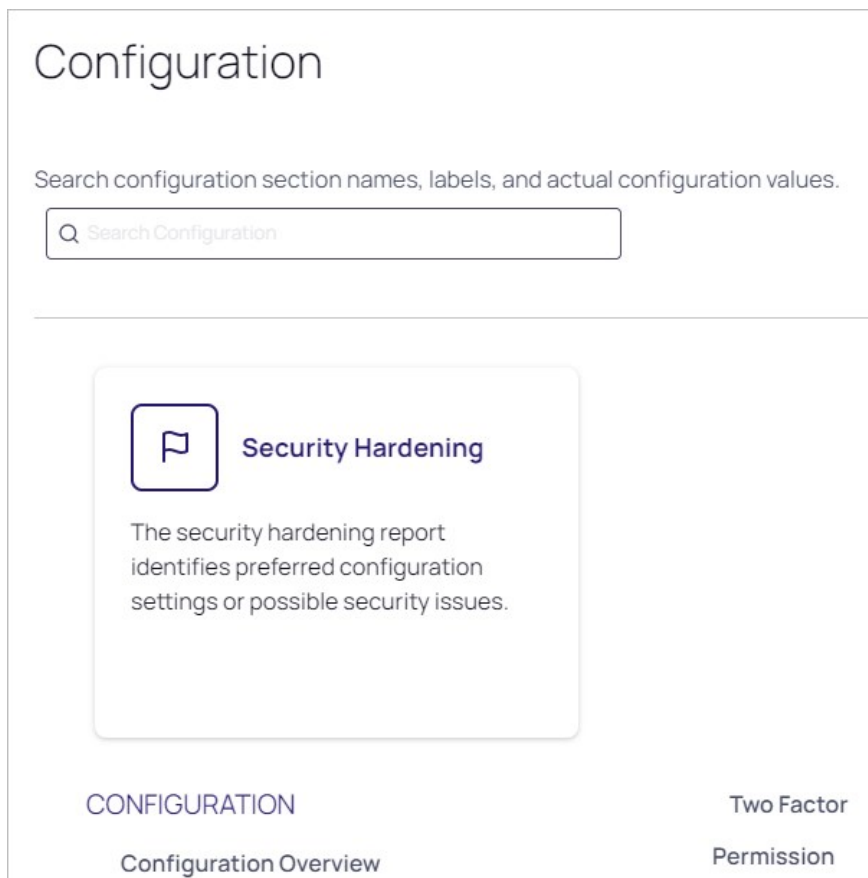
Any user attempting to view a checked-out secret is directed to a notification dialog informing them when the secret is available. Secret Server automatically checks in secrets after either 30 minutes or the interval specified on the secret. Users can check in the secret earlier from the secret's page.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Checkout expiration notifies users with a checked-out secret via their Secret Server inbox and inbox rules when a defined checkout period percentage has elapsed. For example, setting checkout expiration to 80% notifies users when 20% of the checkout interval remains.


This setting is available under "user experience" in the configuration page. To set checkout expiration:

1. Go to the **Admin** menu item and click > >. The Admin slideout appears.
2. Type `configuration` in the search text box and click the link when it appears below. The Configuration page appears:



Configuration

Search configuration section names, labels, and actual configuration values.

 **Security Hardening**

The security hardening report identifies preferred configuration settings or possible security issues.

CONFIGURATION **Two Factor**

Configuration Overview Permission

3. Click the User Experience link at the bottom of the page. The User Experience page appears:

User Experience

Define timezone, default date/time formats, timeout and general application experience.

Edit

Search Delay *	700
Force Inactivity Timeout	Yes
Force Inactivity Timeout Minutes *	60
UI Inactivity Timeout *	5
Time Zone	Eastern Standard Time
Default Date Format	M/d/yyyy
Default Time Format	hh:mm tt
Require Folder For Secrets	No
Secret Password History	1
All Secret Password History	No
Secret View Interval Minutes	5
Checkout Notification Threshold	0
Enable Secret Check Out Extension	No
Application Language	English
Default New User Role	User

- Click the **Edit** button.
- Click the **Checkout Notification Threshold** spinner to select or enter a notification threshold percentage between 0 and 99. This sets what percentage of the checkout period elapses before an expiration notification occurs. That is, if set to 70, a notification is sent when 30% of the checkout period is left.
- Click the **Save** button.

Secret DoubleLocks

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.


Secret Server's *doublelock* is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server.

A shortcut way of thinking about doublelocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets.

1. Navigate to the secret you wish to doublelock by clicking **Secrets** on the main menu.
2. Either drill down to the desired secret in the folders on the main menu, or click the secret in the **All Secrets** table to arrive at the secret's page:

Personal Folders > Will > Mudfin Gmail ☆

[General](#) [Security](#) [Audit](#) [RPC](#) [Dependencies](#) [Sharing](#) [Settings](#)

Secret Name *	Mudfin Gmail	Edit
Template	Web Password	Edit
URL *	https://mail.google.com	Edit
UserName *	smedlymufin	Edit
Password *	***** Show	Edit
Notes		Edit
Launchers	 Web Password Filler	

[Show Advanced](#) [Edit all fields](#)

3. Click the **Security** tab.

Personal Folders > Will > Mudfin Gmail ☆

General **Security** Audit RPC Dependencies Sharing Settings

CHECK OUT Edit	
Require Check Out	No
APPROVAL Edit	
Require Approval	No
OTHER SECURITY Edit	
Require Comment Users will be prompted for comment and ticket number when accessing a Secret.	No
Enable DoubleLock	No
Hide Launcher Password	No

4. Click the **Edit** link for the **Other Security** section. The section becomes editable:


OTHER SECURITY	
Require Comment Users will be prompted for comment and ticket number when accessing a Secret.	<input type="checkbox"/>
Enable DoubleLock	<input type="checkbox"/>
Hide Launcher Password	<input type="checkbox"/>
Cancel Save	

5. Click to select the **Enable DoubleLock** check box. The DoubleLock dropdown list appears.

6. Click to select the doublelock you created earlier.


Important: Enabling doublelock on this secret only grants users access if they have access to to the doublelock and enter their doublelock password. Enabling doublelock disables the RPC features for the secret.











7. Click the **Save** button. The doublelock is now enforced for the secret.

1. Click the  icon at the top right of Secret Server. Your My Profile page appears:

My Profile


GENERAL

Display Name	Will
User Name	Will
Email	
Domain	Local
Last Login	Mon, 13 May 2019 15:26:42 GMT

-  Account Settings
Notifications, theming, password masking, and language settings.
-  Change Password
Change local Secret Server user password.
-  Change DoubleLock Password
Change the doublelock password associated to this user account
-  Reset DoubleLock Password
Reset the doublelock password associated to this user account
-  Manage Secret Access Requests
Approve or deny Secret access requests.
-  Application Access Requests
Manage applications requesting access to Secret Server on your behalf.
-  Sessions
View all active sessions for the current user.
-  Notifications
View all alert notifications for the current user.
-  View in Classic UI
Switch to use a classic Secret Server theme.
-  Logout
Log the current user out of Secret Server

2. Click the **Change DoubleLock Password** button. The Change DoubleLock Password page appears:

Change DoubleLock Password

 Please enter a new DoubleLock password and press the Change Password button. This will allow you to access Secrets with DoubleLock.

Current Password *

New Password *

Confirm Password *

3. Type your current doublelock password in the **Current Password** text box.

Note: You cannot create a doublelock password until you are associated with a doublelock. When you access your first doublelock, you are prompted to create a password.

4. Type your desired doublelock password in the **Password** and **Confirm Password** text boxes.

Important: It is critical that you remember or securely store this password. It cannot be recovered.

5. Click the **Change Password** button. The password is created.

1. Navigate to **Admin > See All**. The Administration page appears:

What are you looking for?

Search for an admin option



Simplified View ▾



Actions

Secret Server features that perform important jobs



Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

2. Type and then click **DoubleLock** in the search text box. The DoubleLock Management page appears:

Admin > DoubleLock Management

DoubleLocks Audit

Manage DoubleLock Password Create New DoubleLock

DOUBLELOCK GROUP MANAGEMENT Show Inactive

DOUBLELOCK GROUP NAME	NUMBER OF SECRETS	NUMBER OF USERS	CREATED	ACTIVE
Main DoubleLock	1 Secrets	1 Users	11/11/2019	✓
System Admin DoubleLock	1 Secrets	1 Users	1/31/2001	✓
Test DoubleLock	3 Secrets	1 Users	11/11/2019	✓

3. Click the desired doublelock. Its page appears:

Admin > DoubleLock Management > My DoubleLock

Doublelock Information

Summary information for this DoubleLock including when it was created and whether or not it is active / enabled. Once a DoubleLock is disabled any associated Secrets will be unable to be accessed.

Assign Users [Add or Remove](#)

Defines the users that are able to access Secrets using this DoubleLock or assign this DoubleLock to other Secrets. Note that only users who have already created a DoubleLock password can be added to a DoubleLock. Users must be part of a DoubleLock group to edit the users in the group. A user can not remove themselves from the DoubleLock

Accessible Secrets Associated with DoubleLock

Secrets that can only be accessed by using this DoubleLock. Note: Only Secrets to which you have access will show in this list.

DoubleLock Name My DoubleLock [Edit](#)

Date Created 11/14/2019

Enabled Yes [Edit](#)

Associated Secrets 0

Users Assigned to DoubleLock

No Secrets are currently using this DoubleLock.

4. Click the **Add or Remove** link in the **Assign Users** section. An Add Users to DoubleLock section appears:

Add Users to DoubleLock


All Search

No users / groups match the search criteria that have created DoubleLock passwords. If a user is expected, ensure that they have created a DoubleLock password.

Cancel Save

5. (Optional) Click the dropdown list to limit the user search to a specific domain.
6. Type the user's name in the search text box. The matching users appear below the search text box:

Add Users to DoubleLock

 John Smith

Cancel Save

7. Click to select the check box next to the desired user. The users that appear do not already have a doublelock assigned to them.
8. Repeat the process for other users.
9. Click the **Save** button.

1. Navigate to **Admin > See All**. The Administration page appears:

What are you looking for?

Search for an admin option



Simplified View ▾



Actions

Secret Server features that perform important jobs



Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

2. Type and then click **DoubleLock** in the search text box. The DoubleLock Management page appears:

Admin > DoubleLock Management

DoubleLocks Audit

Manage DoubleLock Password Create New DoubleLock

DOUBLELOCK GROUP MANAGEMENT Show Inactive

DOUBLELOCK GROUP NAME	NUMBER OF SECRETS	NUMBER OF USERS	CREATED	ACTIVE
Main DoubleLock	1 Secrets	1 Users	11/11/2019	✓
System Admin DoubleLock	1 Secrets	1 Users	1/31/2001	✓
Test DoubleLock	3 Secrets	1 Users	11/11/2019	✓

- Click the **Create New DoubleLock** button. If you have never created a doublelock before, you will have to create a doublelock password first:

Enter DoubleLock Password

Password *

Confirm *

[Forgot DoubleLock Password?](#)

Important: It is critical that you remember or securely store this password. It cannot be recovered.

Type the doublelock password in the **Password** and **Confirm** text boxes, and then click the **Verify Password** button.

Otherwise, you go directly to the Create New DoubleLock popup page because you already have a doublelock password in the system:

Create New DoubleLock

Name

Note: Because it is a secondary password, your doublelock password does not have to (but can) meet the same strong requirements as regular Secret Server passwords (as defined by your admin). Think of it as more of a PIN than a password.

Note: A new doublelock and doublelock password are created together. In fact, it is impossible to create a doublelock password without immediately assigning it to a doublelock. For an existing doublelock, you are assigned access to it by its creator. Upon first accessing it, you must create *your* doublelock password for it. At least one other user will already have created their password for the same doublelock—the creator plus anybody else they granted access to.

4. Type the new doublelock's name in the **Name** text box.
 5. Click the **OK** button. The new doublelock's page appears:
-

Admin > DoubleLock Management > My DoubleLock

DoubleLock Information

Summary information for this DoubleLock including when it was created and whether or not it is active / enabled. Once a DoubleLock is disabled any associated Secrets will be unable to be accessed.

DoubleLock Name My DoubleLock [Edit](#)

Date Created 11/14/2019

Enabled Yes [Edit](#)

Associated Secrets 0

Assign Users [Add or Remove](#)

Defines the users that are able to access Secrets using this DoubleLock or assign this DoubleLock to other Secrets. Note that only users who have already created a DoubleLock password can be added to a DoubleLock. Users must be part of a DoubleLock group to edit the users in the group. A user can not remove themselves from the DoubleLock

Users Assigned to DoubleLock

No users are currently assigned to this DoubleLock.

Secrets Associated with DoubleLock

Secrets that can only be accessed by using this DoubleLock. Note: Only Secrets to which you have access will show in this list.

No Secrets are currently using this DoubleLock.

Important: It is critical that you remember or securely store this password. It cannot be recovered.

6. Click the **Create Password** button. The password is created, and the DoubleLocks page reappears.

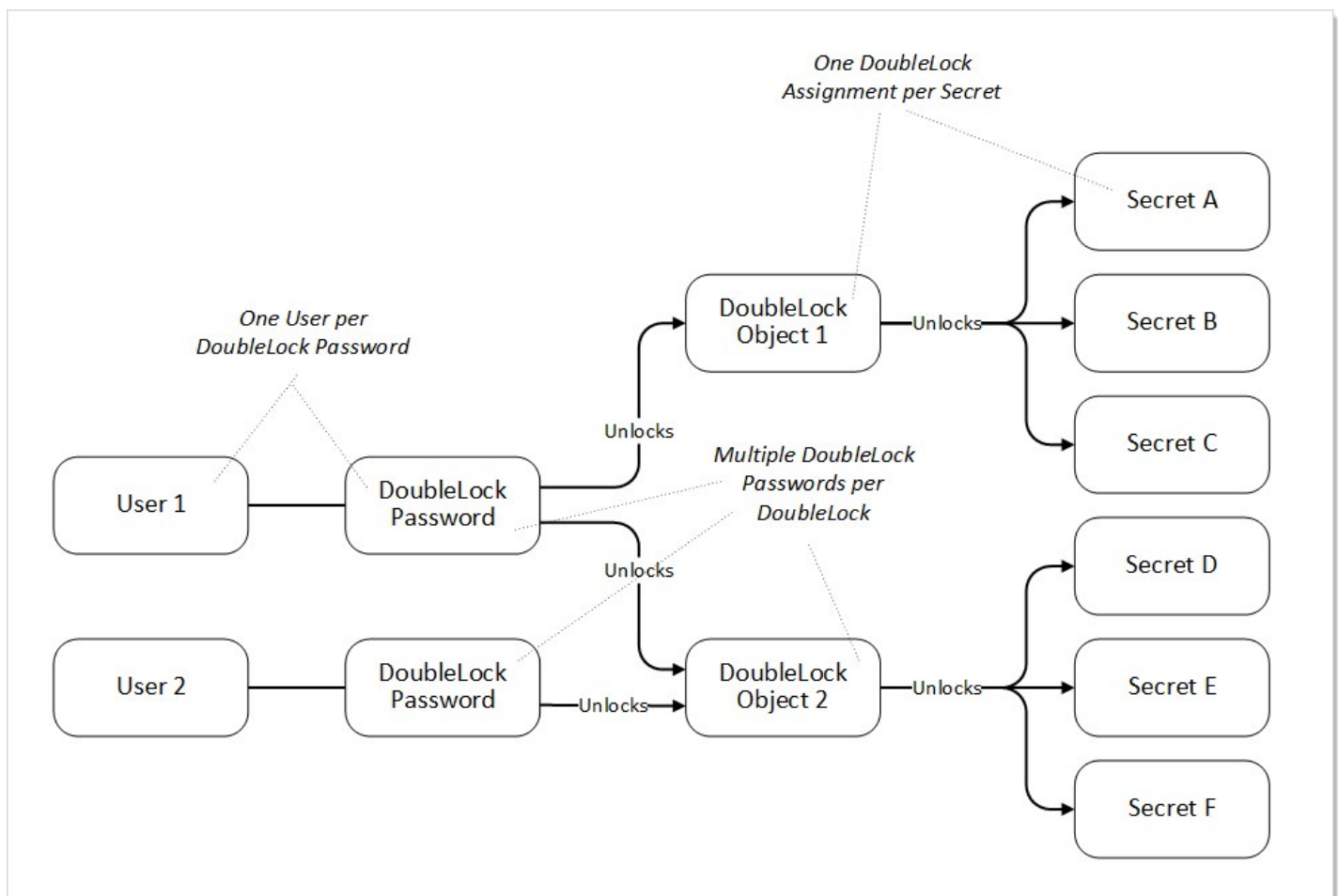
Note: The newly created doublelock does **not** appear on the page.

Note: A new doublelock and doublelock password are created at the same time. In fact, it is impossible to create a doublelock password without immediately assigning it to a doublelock. For an existing doublelock, you are assigned access to it. Upon first access, you must create a doublelock password if you do not already have one.

The doublelock system is a group of interrelated objects (see the following diagram):

- **Doublelock object:** A named object that is associated with one or more secrets and one or more users (via password objects). Doublelock objects, or simply *doublelocks*, point to secrets (what can be accessed) and doublelock password objects (who can access it).
- **Doublelock password object:** An encrypted password that is associated with one user. The same doublelock password object, or simply *doublelock password*, is used for all doublelocks to which a user has access. Once a user is assigned to a doublelock, that user has access to any secret using that doublelock, using a single password. A doublelock password has nothing to do with the user's Secret Server access password.
- **Secret:** A secret that has a single doublelock assigned to it. Multiple secrets can have the same doublelock assigned to them.
- **User:** A Secret Server user, which can have a single doublelock password assigned to it.

Figure: DoubleLock Object Relationships



Because users with access to a given doublelock each have their own separate password. Users that forget their doublelock password cannot simply ask another person using that doublelock for the password. Instead, one of the other users must reassign that forgetful user to the doublelock, and the user must choose a new password. This must occur for every doublelock the user was associated with. If no other doublelock users are available for the assignment to a given secret (there is only one associated doublelock password), the forgetful user is out of luck, and the secret will be destroyed when the user receives a new doublelock password.

When users forget their doublelock passwords, there are multiple steps and considerations. Data loss may or may not result from resetting:

1. When you forget your doublelock password, you typically come to that realization when attempting to access a secret protected by that doublelock:

DoubleLock - Mudfin Gmail

Please enter your DoubleLock password to gain access to the requested resource.


Password *

Forgot DoubleLock Password?

Cancel Continue


2. Click the **Forgot DoubleLock Password?** link. The Reset DoubleLock Password page appears:

Reset DoubleLock Password

 Resetting your forgotten DoubleLock password is irreversible and could result in permanent loss of the data. In the case you are the only user with access to the DoubleLocked Secrets, the data will be lost and the Secrets deleted. If another user has access to the Secret, they will need to re-assign you to the DoubleLock. Please review the DoubleLocks and Secrets that will be impacted on reset.

DOUBLELOCKS

Will

 No one will have access to these Secret(s). The data will be permanently lost and Secrets deleted.

NAME	TEMPLATE	FOLDER	CREATED
Mudfin Gmail	Web Password	Will	2019. 05. 07.

Please enter your login password to confirm the DoubleLock reset, and acknowledge the Secrets will be lost.

Login Password *

✔ Reset DoubleLock Password ✕ Cancel

3. At this stage, there are two possibilities:

- You are the only one with access to the doublelocked secret: When you reset the doublelock password, the secret and its data is deleted. **This is permanent.**
- Others have access to the secret via that doublelock: You can reset the doublelock, and you lose access to the secret, but it is not deleted. You must ask one of those others to re-assign you to the doublelock after you reset it.

4. Type your main Secret Server password in the **Login Password** text box.

5. Click the **Reset DoubleLock Password** button. The password is reset, and if you are the only one with access to it, the secret is deleted.
6. (Optional) Ask one of the others with the doublelock password to re-assign you to the doublelock.

As an admin, to use doublelocks on a secret, you must first create complete these steps for a new doublelock:

1. One time: Create a doublelock password (one time per user). This is automatically required of you when you create a doublelock or access a secret with an existing one (that somebody else assigned to you). You can also create one manually ahead of time.
2. One time: Create a doublelock, which can be used on multiple secrets by multiple users.
3. One or more times: Assign the doublelock to a secret or secret template.
4. One time per user: Assign the user to that doublelock. Users without an existing doublelock password are required to create one.
5. Unlimited times: A user unlocks the doublelock with his doublelock password, which in turn gives the user access to the secret associated with the doublelock (every time the user wants access to the secret).

Secret Heartbeats

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server's *heartbeat* feature allows secrets to have their entered credentials automatically tested for accuracy at a given interval. Using heartbeat on secrets ensures those credentials are up-to-date and can alert administrators if the credentials are changed outside of Secret Server. Heartbeat helps manage secrets and prevent them from being out of sync.

On the **Preferences** page, the **Send Email Alerts when Heartbeat Fails for Secrets** setting can be enabled to email the user when heartbeat fails for any secret the user has view access to.

Heartbeat is configured from the secret template designer. The heartbeat interval determines how often the secret credentials are tested.

To enable heartbeat, ensure it is enabled on the **Remote Password Changing Configuration** page:

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Edit** button.
3. Click to select the **Enable Heartbeat** check box.
4. Click the **Save** button.

Note: Heartbeat must also be enabled on the secret template by setting the **Enable Remote Password Changing Heartbeat** setting.

The heartbeat logs for a specific secret can be accessed by clicking the **View Audit** button on the **Secret View** page and clicking to enable the **Display Password Changing Log** check box. The heartbeat logs for all secrets can be accessed by navigating to **Administration > Remote Password Changing** and scrolling down to the second set of logs.

- **Failed:** The credentials in the secret failed authentication with the target system.
- **Success:** The credentials in the secret authenticated successfully with the target system.
- **UnableToConnect:** Secret Server was unable to contact the target system. Ensure that the domain, IP address, or hostname is correct and resolvable from the server that Secret Server is installed on.
- **UnknownError:** Check the Heartbeat log on the Remote Password Changing page for details, and contact [Support](#) for assistance.

For the most up-to-date list of account types supported by RPC, see [this KB article](#).

Heartbeat runs in a background thread to check each secret where it is enabled. If the credential test fails, the secret is flagged as heartbeat failed and out of sync. To avoid locking out the account, heartbeat no longer runs on that secret until the secret items are edited by the user. If the machine is determined to be unavailable, the secret is flagged as heartbeat unable to connect and the secret continues to be checked on the heartbeat interval.

To manually use heartbeat to check the credentials, the **Secret View** page has a **Heartbeat Now** button. The button marks the password as heartbeat pending. The background thread processes the secret in the next 10 seconds, and when the page is refreshed the heartbeat status is updated.

Note: Heartbeat does not work on Windows accounts on the server that is running Secret Server. These accounts are flagged with an "Incompatible Host" status.

To run heartbeat for a secret:

1. From **Dashboard**, click the secret you would like to test.
2. Click the **View** button. The **Last Heartbeat** field of the secret shows the last date and time that Heartbeat ran for this secret.
3. To run Heartbeat once more, click **Run Heartbeat** at the bottom of the Secret.
4. Monitor the **Last Heartbeat** field to see the updated status. This may take a few seconds to complete.

If you receive any Heartbeat status code aside from Success, you can check the Heartbeat log for details. To view the entry, Go to **Admin > Remote Password Changing** and then search for the secret name in the **Search** field of the **Heartbeat Log**.

Secret Import and Export

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secrets are imported or exported as a comma-separated-value (CSV) file or as XML:

- The CSV file is easily read and edited in Excel or other spreadsheet application. The file is grouped by secret template and each cluster of secrets has a header row that contains the template text-entry field names followed by all exported secrets based on that template.
- The XML file is useful for migrating data from one Secret Server installation to another or even from a third-party application to Secret Server.

Secrets are exported in the exact same structure as a secret Import.

This topic has three subtopics:

- [Exporting Secrets](#)
- [Importing Secrets](#)
- [Secret Server Migration Tool](#)

Import and export include:

- Folders (and their permissions)
- Secret templates
- Secrets (and their permissions)

The import or export does **not** include users, groups, launchers, configuration, and others.

Note: Folders and secret templates are only exportable from Secret Server 10.0 and later.

Important: To ensure permissions are applied correctly, you must recreate your users and groups on the target Secret Server before importing.

The following secret template settings **are** transferred with the export or import:

- Edit requires
- Field slug names
- Hide on view
- Is required?
- Keep secret name history
- One-time password settings
- Secret template icons
- Type descriptions
- Validate password requirements on create or edit

The following secret template settings are **not** transferred:

- Associated secrets
- Launcher settings
- Password changing settings
- Session recording enabled

If you use XML import and export to migrate from Secret Server on-premises to cloud, the major release version (x.x) must be the same. Otherwise, you need to upgrade before you can migrate. Additionally, the **Allow Duplicate Secret Names** check box on the **General** tab of the **Admin Configuration** page should be disabled in Secret Server Cloud before importing.

You can use XML import and export to transfer between on-premises and cloud editions.

This feature allows you to automatically export secrets on a schedule to an external location in an encrypted, password-protected archive.

The secret export settings and XML export format matches the existing Export/Import tool (Admin > Export / Import). This feature lets you automate that export process.

To access this feature, your user must have at least one automatic export permission where you can then find it at **Admin > Automatic Export**.

The export is performed using the permissions of the user last that set up the automatic export, this means only secrets that user can access can be exported.

All actions, successful or not, related to this feature are audited and logged.

Export Process

Note: The actual export of secrets to XML is exactly the same as the standard Export / Import tool—only the triggering differs.

The automatic export follows this process in order:

1. Either a user clicks the Run Export button on the Automatic Export tab or the configured frequency days have passed.

Note: Users can manually run the automatic export to determine if the export performs as desired.

2. Secret Server determines if the user who set up the automatic export has either the Run Automatic Export or Administer Run Automatic permission.
3. Secret Server exports secrets to XML, subject to your export settings.
4. Secret Server compresses the XML export into an encrypted, password-protected ZIP archive. The filename for both the XML file and the zipped XML file includes the export date and time—both are named the same, except the file extension. The encryption is 256-bit AES, and the password comes from the Export Password configuration setting.
5. Secret Server saves the archive to the export path (on-premises customers) or to cloud storage (cloud customers).
6. Secret Server logs the results, successful or not, noting any errors. You can view the logs on the Automatic Export Log tab.

Considerations and Settings

Configurations and Returned Data

The Automatic Export page shows the following information and configurations:

Those unique to automatic exports:

- **Enabled:** Whether the feature is enabled.
- **Last Exported:** The last time an automatic export successfully ran.
- **Export Path:** The path the export archives are saved to.

Note: On-premises Secret Server customers must have write permissions to the directory.

- **Export User:** The user the secret export runs as. Thus, only secrets this user has access to can be exported. This setting is updated automatically to the last user who saved automatic export configuration changes.

- **Frequency (Days):** Number of days between automatic exports.

Those in common with the secret export tool:

- **Export Child Folders:** Whether the export should include child folders, performing a recursive export.
- **Export Folder Paths:** Whether the export should include folder paths.
- **Export Password:** The secret whose password value will be used to password-protect the export archive.
- **Export TOTP Settings:** Whether the export should include Time-based One-Time Password (TOTP) settings.
- **Folder:** The folder the exported secrets are in. If no folder is selected, all secrets are exported. If Export Child Folders is enabled, the folder's subfolder's secrets are included.

Export Storage

Once the XML export is encrypted and archived in a ZIP file, the file is stored at an external location, which differs for on-premises customers and cloud customers. Only the 10 most recent exports are retained. Older export archives are purged as new exports are stored. This applies to both on-premises customers and cloud customers.

On-Premises storage export archives are saved in the directory in the Export Path configuration. Secret Server must have write permissions to this folder.

Cloud export archives are listed for viewing and downloading on the Automatic Export Storage tab. You can automate downloading these export archives using the REST API Automatic Export. See the [REST Web Services API Reference](#).

Security

Because this feature moves sensitive data outside of Secret Server, it is very important to understand that **anyone with access to the export archive and the export password has access to all exported secrets**. This bypasses Secret Server security features and may result in a user having access to secrets they would not have access to in Secret Server.

The export archive is a password protected, 256-bit AES-encrypted zip file. Thus, the only thing preventing a possible breach is the password, so it is important you use a cryptographically strong password to foil brute force attacks.

Permissions

The following permissions relate to automatic secret export:

- **Administer Automatic Export:** The user can do everything the other permissions allow *and* edit the automatic export configuration.
- **Download Automatic Export:** The user can view all of the automatic export tabs *and* download exports from cloud storage (cloud customers only).
- **Run Automatic Export:** The user can view all of the automatic export tabs *and* run the export manually by clicking the Run Export button.
- **View Automatic Export:** The user can view all of the automatic export tabs.

Event Subscriptions

The following automatic export events are available for event subscriptions:

- **Download:** When an export archive is downloaded (cloud customers only).
- **Edit:** When changes are made to the automatic export configuration settings.

- **Export:** When an automatic export executes, automatically or manually.

Note: This event also triggers when a *fatal* error occurs during the export causing the export to fail. Non-fatal errors do not trigger this event, but they are logged on the Automatic Export Log tab.

- **Run Export:** When an automatic export is executed manually by a user clicking the Run Export button.

Note: If you subscribe to both this event and the Export event, both events trigger at once.

To set up an automatic export:

Note: Only the secrets the user has view access to are exported.

Important: File attachments on the original secret are not exported into the XML file and require using the API to migrate. Secret audits and history are not preserved during the migration.

Note: If you later use the export to import into another Secret Server instance, be sure to first create the AD groups and users using the permissions on the secrets in the original Secret Server instance. Otherwise, they will not be created when the secrets are imported into the new instance.

1. Create a new secret to store the export password in.
2. Go to **Admin > See All**. A popup menu appears.
3. Click the **Automatic Export** link. The Automatic Export (configuration) tab appears:

Admin > Automatic Export

Automatic Export Log Audit

Run Export

Automatic Export [Edit](#)

Setup an automatic Secret export to XML that is encrypted and password protected then stored at the specified location.

The export is performed using the permissions of the export user, this means only secrets they can access will be exported.

The export user is set as the last user to save configuration changes here.

[Automatic Export Documentation](#)

Enable Automatic Export	No
Last Exported	Never
Export User	(not set)
Export Path *	None
Export Password *	No Secret Selected
Folder	< All Folders >
Export Folder Paths	Yes
Export Child Folders	Yes
Export TOTP Settings	Yes
Frequency (Days)	7

- Click the **Edit** link next to **Automatic Export**. The page becomes editable:

[Run Export](#)

Enable Automatic Export

Last Exported Never

Export User (not set)

Export Path *

Export Password * [No Secret Selected](#)

Folder [No Folder Selected](#)

Export Folder Paths

Export Child Folders

Export TOTP Settings

Frequency (Days)

5. Click to select the **Enable Automatic Export** check box.
6. There is no need to set the export user—Secret Server automatically notes who is logged on when you save the export.
7. Type the path to export the secret to in the **Export Path** text box (on-premises users only). On-premises Secret Server customers must have write permissions to the directory.
8. Click the **No Secret Selected** link. The Select Secret popup appears.
9. Click the secret you want to store the password in. Your choice appears as a link next to Export Password:

Export Password * [My Test Secret](#)
[Clear](#)

10. Click the **No Folder Selected** link to choose a folder to export. By default, all secrets are exported if a folder is not selected. The Select Folder popup appears. Click on the desired folder. Your chosen folder appears as a link:

Folder My Folder Clear

- (Optional) Click to select the **Export Folder Paths** check box. This adds the full folder path to the export. Folder paths in the export file provide organizational structure if secrets need to be imported later.
- (Optional) Click to select the **Export Child Folders** check box. This option includes any subfolders of the one you chose earlier.
- (Optional) Click to select the **Export TOTP Settings** check box if you want to include time-based one-time password settings in the export.
- Type the number of days between automatic exports in the **Frequency (Days)** text box.
- Click the **Save** button. Any error messages, such as secrets with doublelocks, appear. The page leaves edit mode, and the Run Export button appears.
- Click the **Run Export** button to text the export. A confirmation popup appears:

Confirm Run Automatic Export Now

Are you certain you want to run this Automatic Export?

Cancel Run Now

- Click the **Run Now** button. A notification appears:

Running export, check the log for results Run Export

And then disappears.

- Click the **Log** tab to confirm the export worked:

Automatic Export		
EXPORT DATE	SUCCESSFUL	NOTES
7/15/2021 12:30 PM	Yes	Successfully automaticall...

- Using the stored password, open the zip file to confirm its contents.

Overview

If you use Secret Server Cloud, you can use a REST API to download exports and view your export storage list.

The automatic export feature has the following endpoints available for cloud customers only. API usage is fully audited.

A typical use of these API endpoints is to automate downloading exports to your backup solution outside of Secret Server (for redundancy).

Note: Any permission errors when using the API will return a 403 forbidden status code and an API_AccessDenied error message.

Viewing the Storage List

Get a list of the exports currently in storage. Your session must be authenticated, and the authenticated user must have Automatic Export view permissions.

Sample Request

GET: <http://sample.secretservercloud.com/api/v1/configuration/auto-export-storage>

Sample Response

```
{
  "records": [
    {
      "id": "123",
      "autoExportConfigurationId": 1,
      "storageDate": "2021-07-01T07:00:02.27",
      "filename": "secret-server-export-20210707070002",
      "canDownload": true
    },
    ...
  ],
  ...
}
```

The response is a JSON object with a records property whose value is the list of all the exports in storage. Each list entry has the following properties:

- **id:** The ID for this export in storage, which is used with the Download Export endpoint to download the export.
- **autoExportConfigurationId:** The ID for the automatic export configuration this export belongs to. This may be useful in the future if we support multiple export configurations, but for now it is only used internally.
- **storageDate:** The date and time the export was stored.
- **filename:** The filename for the export archive and the export XML file inside it.
- **canDownload:** Whether the user can download this export archive.

Downloading Secret Exports

Download an export in storage by its ID. Your session must be authenticated and the authenticated user must have automatic export download permissions.

Sample Request

GET: <http://sample.secretservercloud.com/api/v1/configuration/auto-export-storage/item/{id}>

Where is the ID of the export you want to download. This value is obtained from the **Storage List** endpoint.

Sample Response

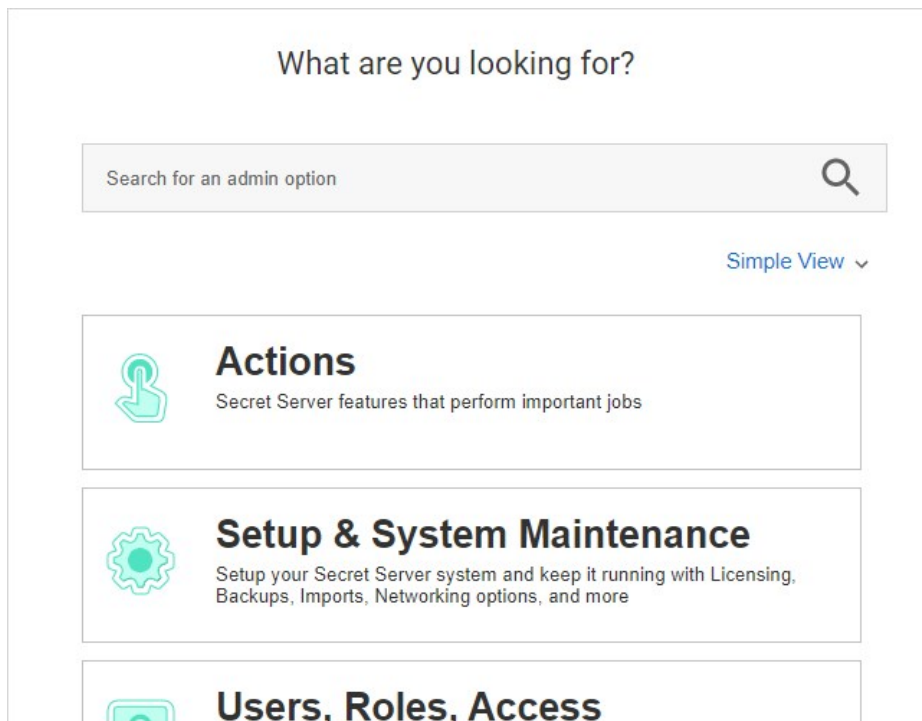
A stream of bytes representing the export archive.

To export a secret, either CSV or XML:

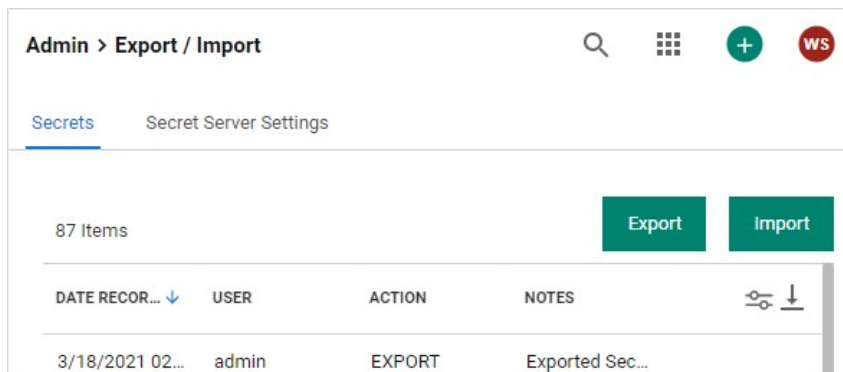
Note: Only the secrets the user has view access to are exported.

Imported: File attachments on the original secret are not exported into the XML file and require using the API to migrate. Secret audits and history are not preserved during the migration. Be sure to first create the AD groups and users using the permissions on the secrets in the original Secret Server instance. Otherwise, they will not be created when the secrets are imported into the new instance.

1. Go to **Admin > See All**. The admin menu appears:




2. Hover the mouse pointer over the **Setup and System Maintenance** panel and select **Export / Import**. The Export / Import page appears:



3. Click the **Export** button on the **Secrets** tab. The Export page appears:

Export

Please enter your password for security purposes.

Folder  < All Folders >

Password *


Export with Folder Path

Export Child Folders

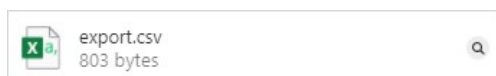
Export TOTP Settings

Export Format CSV XML

Enter any additional notes or explanations for the export.

 **Export**

4. Click the folder icon to choose a folder to export. By default, all secrets are exported if a folder is not selected.
5. Type your password in the **Password** text box. The administrative password must be entered, as it is a security measure to verify the permission of the user performing the export.
6. (Optional) Click to select the **Export with Folder Path** check box. This adds the full folder path to the export. Folder paths in the export file provide organizational structure if secrets need to be imported later.
7. (Optional) Click to select the **Export Child Folders** check box. This option includes any subfolders of the one you chose earlier.
8. (Optional) Click to select the **Import with TOTP Settings** check box if you want to include time-based one-time password settings in the export.
9. Click the **Export Format** selection button to choose the type of export. CSV is for Excel and the like, and XML is for migrating to other Secret Server instances.
10. Click the **Export** button. The Export Secrets popup appears. Any error messages, such as secrets with doublelocks, appear.
11. Click the Close button. When the exportation is finished, an `export.csv` file appears in your browser's queue:

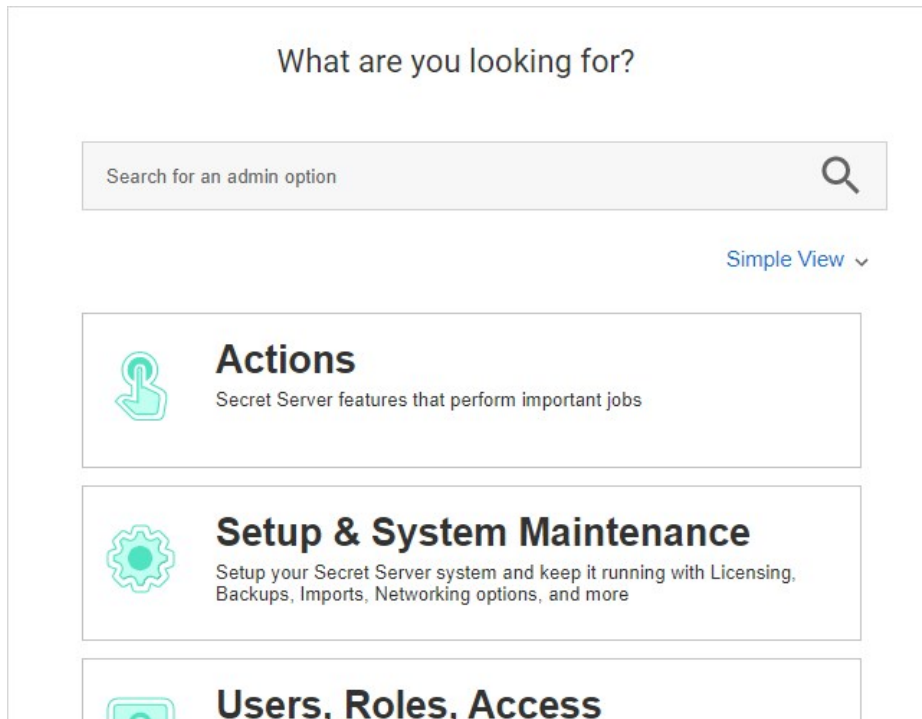


Important: Take care with the file—it contains unencrypted passwords.

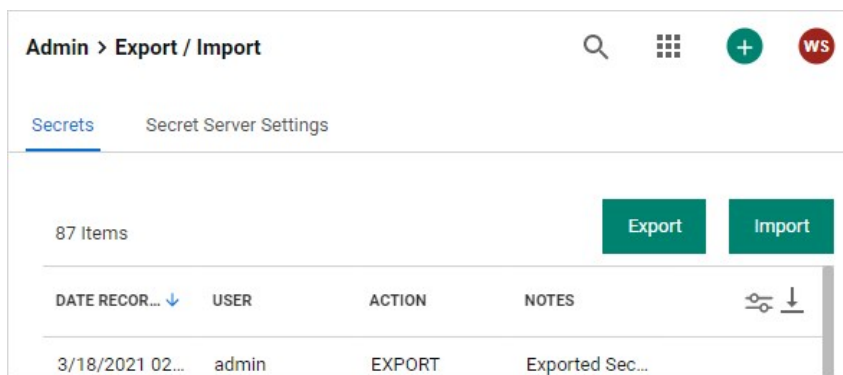
Secret Server's importation feature simplifies integration with legacy systems and allows users to easily add large numbers of secrets from an Excel or comma-separated values (CSV) file. Secrets are batch imported by template, so multiple types of input data need to be imported in several batches. The [Secret Server Migration Tool](#) supports easy addition of existing passwords from other third-party password-storing applications.

Importing CSV Data

1. Go to **Admin > See All**. The admin menu appears:



2. Hover the mouse pointer over the **Setup and System Maintenance** panel and select **Export / Import**. The Export / Import page appears:



3. Click the **Import** button on the **Secrets** tab. The Choose Secret Template page appears:

Choose Secret Template

What type of Secret do you want to import?

< Select > *

[← Back](#) [→ Continue](#)

Additional Options

[Download](#) the Secret Server Migration Tool.
[Upload XML File](#) for an advanced import to add Folders, Secret Templates, and Secrets.

4. Click the **What type of Secret...** list box to select the type of secrets you intend to import.
5. Click the **Continue** button. The Import Secrets page appears:

Import Secrets

Paste your Secrets directly from Microsoft Excel® or in comma/tab separated format and click 'Next'.
Do not include a header line.
Secret Name must be included but others fields can be blank.
Fields containing commas or tabs must be surrounded with double quotes.
It is permissible to include quotes if they are escaped with a \ (for example, pa\"word comes out as pa"word)
Fields must be in the following order:

Secret Name,Full Name,Card Number,Expiration Date,Card Type,Notes

[← Back](#) [→ Next](#)

Allow Duplicate Secrets Import With Folder Import With TOTP Settings

6. Paste the secrets for importation from MS Excel or a CSV file directly into the text box in the **Import Secrets** page. The order of the imported fields is based on the template selected. Consider the following:
 - o Do not include a header line. The field names are determined by the order, not a header line.
 - o The fields **must** be in this order: Secret Name, AccessKey, SecretKey, Username, SecretId, and Trigger.

- Secret names must be included, but other text-entry fields can be blank unless the secret template indicates that the text-entry field is required
 - Fields containing commas or tabs must be surrounded with double quotation marks
 - If you have to include double quotation marks inside your data, escape all of them with a \ character so the importer does not get confused.
7. Click to select the **Allow Duplicate Secrets** check box if you wish to import a secret with the same name as an existing one.
 8. Click to select the **Import with Folder** check box if you included an additional field in the importation text with a fully qualified folder name for the secret to be created in.
 9. Click to select the **Import with TOTP Settings** check box if you want to include time-based one-time password settings in the import.
 10. Click the **Next** button. Secret Server displays a preview.
 11. If you are happy with what you see, click the **Yes, Import these Secrets** button.

Importing Secrets with XML

Advanced XML importation adds folders, secret templates, and secrets based on an XML file. Permissions can be specified on the folders and secrets or the default is to inherit permissions. This import can only be done by administrators with proper role permissions.

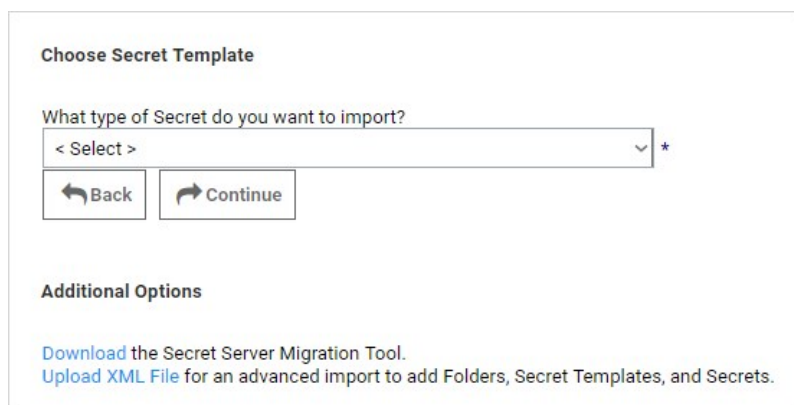
Important: Migration is **not** supported by Delinea Technical Support.

Procedure

1. Ensure your XML is formatted correctly. If coming from a Secret Server export, you should be good to go. See [Example XML File](#).

Important: Do not edit the XML file with Windows Notepad. Instead, use Notepad++, Visual Studio Code, or Atom to make your edits. Windows Notepad can add invisible characters that can prevent importation.

2. Go to **Admin > See All**.
3. Click the **Import Secrets** link. The Import page appears:



The screenshot shows a web interface titled "Choose Secret Template". It features a dropdown menu with the text "What type of Secret do you want to import?" and a placeholder "< Select >". Below the dropdown are two buttons: "Back" with a left-pointing arrow and "Continue" with a right-pointing arrow. Underneath the buttons is a section titled "Additional Options" containing two links: "Download the Secret Server Migration Tool." and "Upload XML File for an advanced import to add Folders, Secret Templates, and Secrets."

4. Click the **Upload XML File** link. The Advanced Import page appears:

Advanced Import

i The Advanced Import will add Folders, Secret Templates, and Secrets using data in the xml file. Permissions can be specified on the Folders and Secrets. Items without permissions specified in the xml, will have permissions set based upon the option chosen in the 'Secrets without permissions' dropdown. For details on the XML file, see KB Article [Advanced Import with XML](#).

Secrets without permissions Inherit Permissions from folder v

Browse... No file selected. ➔ Import XML File

5. Click the **Secrets without permissions** dropdown list to choose how you want secrets without permissions to get them assigned.
6. Click the **Browse** button to choose the XML file.
7. Click the **Import XML File** button.

Example XML File

The XML file should look like the example below, the comments are for explanation only and may be removed before importing, if desired.

Important: Migration is **not** supported by Delinea Technical Support.

Notes

- Leaving the <Permissions> tag empty for a folder will cause that folder to inherit permissions from its parent folder.
- Leaving the <Permissions> tag empty for a secret will cause it to inherit permissions from its folder.
- To add a line-break within a Notes field use ##BR##.

Note: Please do **not** edit the XML file with Windows Notepad. Use Notepad++, Visual Studio Code, or Atom to make your edits. Using Notepad increases you chances of importation failure.

Sample XML

```
<?xml version="1.0" encoding="utf-16"?>
<ImportFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema";
<Folders>
  <Folder>
    <FolderName>Customers</FolderName>
    <FolderPath>Customers</FolderPath>
    <Permissions>
      <Permission>
        <View>true</View>
        <Edit>true</Edit>
        <Owner>true</Owner>
        <UserName>admin</UserName>
      <!-- Either UserName or GroupName is required in permissions -->
    </Permission>
    <Permission>
      <View>true</View>
      <Edit>>false</Edit>
      <Owner>>false</Owner>
      <GroupName>Auditors</GroupName>
    </Permission>
  </Permissions>
```

```

</Folder>
<Folder>
  <FolderName>Customer A</FolderName>
  <FolderPath>Customers\Customer A</FolderPath>
  <Permissions />
<!-- Empty Permissions will cause folder to inherit from parent -->
</Folder>
</Folders>
<!-- Groups are optional -->
<Groups>
  <Group>
    <GroupName>Other Administrators</GroupName>
    <GroupMembers>
      <GroupMember>
        <UserName>admin2</UserName>
      </GroupMember>
      <GroupMember>
        <UserName>DomainAdmin</UserName>
        <Domain>http://testdomain.test.com</Domain>
      </GroupMember>
    </GroupMembers>
  </Group>
  <Group>
    <GroupName>Domain Administrators</GroupName>
    <Domain>http://testdomain.test.com</Domain>
    <GroupMembers>
      <GroupMember>
        <UserName>DomainAdmin</UserName>
        <Domain>http://testdomain.test.com</Domain>
      </GroupMember>
    </GroupMembers>
  </Group>
</Groups>
<SecretTemplates>
<!-- You can have multiple secret type entries -->
<secrettype>
  <name>Windows Account</name>
  <active>true</active>
  <fields>
    <field isexpirationfield="false">
      <name>Resource URL</name>
      <mustencrypt>false</mustencrypt>
      <isurl>false</isurl>
      <ispassword>false</ispassword>
      <isnotes>false</isnotes>
      <isfile>false</isfile>
      <passwordlength>0</passwordlength>
      <historylength>0</historylength>
      <isindexable>false</isindexable>
    </field>
    <field isexpirationfield="false">
      <name>Username</name>
      <mustencrypt>false</mustencrypt>
      <isurl>false</isurl>
      <ispassword>false</ispassword>
      <isnotes>false</isnotes>
      <isfile>false</isfile>
      <passwordlength>0</passwordlength>
      <historylength>0</historylength>
      <isindexable>false</isindexable>
    </field>
    <field isexpirationfield="false">
      <name>Password</name>
      <mustencrypt>true</mustencrypt>
      <isurl>false</isurl>
      <ispassword>true</ispassword>
      <isnotes>false</isnotes>
      <isfile>false</isfile>
      <passwordlength>12</passwordlength>
<!-- Use this number for 'All' history -->
      <historylength>2147483647</historylength>
      <isindexable>false</isindexable>
    </field>
    <field isexpirationfield="false">
      <name>Notes</name>
      <mustencrypt>false</mustencrypt>

```

```

<isurl>>false</isurl>
<ispassword>>false</ispassword>
<isnotes>>true</isnotes>
<isfile>>false</isfile>
<passwordlength>0</passwordlength>
<historylength>0</historylength>
<isindexable>>true</isindexable>
</field>
</fields>
<expirationdays>0</expirationdays>
</secrettype>
</SecretTemplates>
<Secrets>
<Secret>
<SecretName>Test Secret</SecretName>
<SecretTemplateName>Windows Account</SecretTemplateName>
<FolderPath>Customers\Customer A</FolderPath>
<Permissions>
<Permission>
<View>>true</View>
<Edit>>true</Edit>
<Owner>>false</Owner>
<GroupName>IT Admins</GroupName>
</Permission>
<Permission>
<View>>true</View>
<Edit>>true</Edit>
<Owner>>true</Owner>
<UserName>admin</UserName>
</Permission>
</Permissions>
<SecretItems>
<SecretItem>
<FieldName>Resource URL</FieldName>
<Value>10.10.0.25</Value>
</SecretItem>
<SecretItem>
<FieldName>Username</FieldName>
<Value>Administrator</Value>
</SecretItem>
<SecretItem>
<FieldName>Password</FieldName>
<Value>D*KGY#$5</Value>
</SecretItem>
<SecretItem>
<FieldName>Notes</FieldName>
<Value>Just some notes##BR##...and some more notes on a new line. </Value>
</SecretItem>
</SecretItems>
</Secret>
<Secret>
<SecretName>Another Test Secret</SecretName>
<SecretTemplateName>Windows Account</SecretTemplateName>
<FolderPath>Customers\Customer A</FolderPath>
<!-- Empty Permissions causes secret to inherit from folder -->
<Permissions />
<SecretItems>
<SecretItem>
<FieldName>Resource URL</FieldName>
<Value>10.10.0.25</Value>
</SecretItem>
<SecretItem>
<FieldName>Username</FieldName>
<Value>JSmith</Value>
</SecretItem>
<SecretItem>
<FieldName>Password</FieldName>
<Value>DKud3()DS</Value>
</SecretItem>
<SecretItem>
<FieldName>Notes</FieldName>
<Value>This line has an empty line##BR####BR##in between this line.</Value>
</SecretItem>
</SecretItems>
<SecretDependencies>
<!-- Secret dependencies are optional, and there can be multiple ones -->

```

```
<SecretDependency>
  <Active>true</Active>
  <Restart>true</Restart>
  <Description>Some Dependency</Description>
  <MachineName>192.168.99.1</MachineName>
  <DependencyName>Some Service</DependencyName>
  <Type>Windows Service</Type>
<!-- Leave this blank to not use a PrivilegedAccount -->
  <PrivilegedAccount>Some Account</PrivilegedAccount>
  <WaitBeforeSeconds>10</WaitBeforeSeconds>
</SecretDependency>
</SecretDependencies>
</Secret>
</Secrets>
</ImportFile>
```

Secret Server offers a migration utility for importing secrets from other applications. Currently, the migration tool supports the following applications:

- KeePass (version 1 and 2)
- Password Corral
- Password Safe
- Passwords MAX

Note: This is done with another exportation tool that creates a single XML file. Please contact Delinea Support for details.

[Download the Tool](#)

Secret Launchers and Protocol Handlers

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

A secret *launcher* launches applications on end-user machines and automatically logs on using credentials stored in Secret Server. In general, there are three types of launchers: RDP, SSH, and Custom. This provides a convenient method to open RDP and PuTTY connections, but it also circumvents users needing to know their passwords—a user can still gain access to a needed machine but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser.

A *protocol handler* is an application on an end-user's machine. It enables communication between Secret Server and that client machine. It also provides the files needed by launchers. When a Secret Server user starts a launcher:

1. The protocol handler bootstraps the client-side application.
2. The protocol handler communicates with Secret Server over HTTP(S) to ensure that it is the latest version. If not, it begins an upgrade process.
3. The protocol handler bootstraps the target launcher type and begin the process of securely logging in the user. Beyond HTTP(S) transport protection, credentials are retrieved securely from Secret Server using signed AES-256-encrypted messages.

Secret Server has a convenience feature that eliminates the need to manually enter a su or sudo command's password when using a proxied SSH session to a Unix or Linux server. When a user manually types a su or sudo command with a valid secret ID, the SSH proxy automatically provides the username and password to use. The user does not need to know either.

For su, the connection procedure is as follows:

1. Secret A is created to contain the username and password for the su privilege elevation. Any potentially elevated users must have access to this secret.
2. Using secret B, a user (with access to secret A) starts a Secret Server proxied SSH session.
3. When the user types su at the command prompt, the SSH proxy detects it, determines the user has access to secret A, and augments the command with the secret ID for secret A via a command line argument. Any other arguments the user may have typed are left as is.
4. The user runs the su command, and the secret ID is replaced with the user and credential from secret A.
5. With the elevated permissions (temporarily as another user) from secret A, the user completes the desired tasks.
6. When finished, the user uses an exit command to return to their non-elevated status based on secret B.

Note: The added argument appears as `--secret-id <secret ID>` OR `-id <secret ID>`, such as `su --secret-id ElevationSecret`, which is replaced by a username and password when the command runs.

Note: Sudo does not take either secret argument and automatically types the current user's password.

Secret Server launchers, supported by protocol handlers, come in three primary types:

- **Remote Desktop:** Launches a Windows Remote Desktop session and automatically authenticates the user to the machine.
- **PuTTY:** Opens a PuTTY session and authenticates the user to a Unix system.
- **Web Password Filler:** Uses a Chrome extension to automatically log the user into a website with secret credentials.
- **Web Launcher:** An alternative method to automatically log on websites. See [Web Launchers](#).

Problem

When a Remote Desktop Launcher fails to log into a machine, it is sometimes because the machine is configured to have a default domain name (other than the local machine name). To determine whether the machine is configured to have such a default domain name, check one of the following:

Registry Setting: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\DefaultDomainName

Group Policy (Windows 2008 and higher): "Assign a default domain for logon" under Computer Configuration\Administrative Templates\System\Logon

Solution

Add a key to `web-appSettings.config` that will cause the RDP launcher to use the machine name as the domain name when authenticating using a local Windows account.

1. Run a text editor as an administrator on the server running Secret Server.
2. Open the `web-appSettings.config` file located in the Secret Server application directory (typically `C:\inetpub\wwwroot\secretserver`).

3. Add the following key within `<AppSettings>`

```
<add key="RDPUseComputerForDomain" value="true" />
```

4. Perform an IIS reset
5. Test the launcher.

The following instructions describe how to set up a custom launcher using SecureCRT:

Step 1: Creating the Custom Launcher

1. Navigate to **Administration > Secret Templates**.
2. Click **Configure Launchers**. The Launcher Types page appears.
3. Click the **New** button. The Launcher page appears:

Launcher

GENERAL SETTINGS

Launcher Type

Process

Launches the process on the user's machine and replaces \$ parameters with values from the Secret and its associated Secret. For more information see this [KB Article](#)

Launcher Name

*

Active

Use Additional Prompt

Launcher Image

Use Custom Image?



To prevent parameter injection in **Process Arguments** fields below, quotation marks can be inserted around custom parameters.

Example:

`$USERNAME` becomes "`$USERNAME`" prior to launch.

Wrap custom parameters with quotation marks

Record Multiple Windows

Record Additional Processes

WINDOWS SETTINGS

Process Name

ex. powershell

[How do I configure process arguments?](#)

Process Arguments

ex. -user \$USERNAME -pwd \$PASSWORD -f

Run Process As Secret Credentials

Use Operating System Shell

[Advanced](#)

MAC SETTINGS

Process Name

ex. /Applications/TextEdit.app

[How do I configure Mac process name and arguments?](#)

Process Arguments

ex. -user \$USERNAME -pwd \$PASSWORD -f

Save

Cancel

4. Click the **Launcher Type** list box and select one of the following:
 - **Process:** If you would like to use secret credentials to connect directly to the remote host.
 - **Proxied SSH Process:** If you have SSH Proxy enabled. This will prevent Secret credentials from being passed to the client by connecting to Secret Server's proxy to interact with the remote host.
5. Type the name `Secure CRT Proxied Process` in the **Launcher Name** text box.
6. Type the location and filename of the executable (`C:\program files\acme software\clients\securect.exe`) in the **Process Name** text box in the **Windows** section. The location must be on the client machine (the machine that will run the launcher).
7. Type the following custom command-line parameters in the **Process Arguments** text box:


```
/ssh2 /AUTH keyboard-interactive /PASSWORD $PASSWORD /P $PORT /L $USERNAME $HOST
```
8. Click the **Save** button. The new launcher appears:

Launcher

GENERAL SETTINGS

Launcher Name Secure CRT Proxied Process

Active Yes

Launcher Image 

Wrap custom parameters with quotation marks Yes

Record Multiple Windows Yes

Record Additional Processes < None >

WINDOWS SETTINGS

Process Name C:\program files\acme software\clients\securecrt.exe

[How do I configure process arguments?](#)

Process Arguments /ssh2 /AUTH keyboard-interactive /PASSWORD \$PASSWORD /P \$PORT /L \$USERNAME \$HOST

Run Process As Secret Credentials No

Load User Profile No

Use Operating System Shell No

MAC SETTINGS

Process Name

[How do I configure Mac process name and arguments?](#)

Process Arguments

[Back](#) [Edit](#) [View Audit](#)

Step 2: Creating a Custom Secret Template (optional)

See [Creating and Editing Secret Templates](#) for details on creating a custom secret template.

Step 3: Associating the Launcher with a Secret Template

1. Navigate to **Administration > Secret Templates**.
2. Select the desired template from the drop-down menu.
3. Click **Edit**. The Secret Template Designer appears.
4. Click **Configure Launcher**. The **Secret Template Edit Launcher Configuration** page appears.

5. Click the **Add New Launcher** button.
6. For **Launcher Type to Use**, select your custom launcher.
7. Leave **Domain** set to **< blank >**.
8. For **Password** and **Username**, select **Password** and **Username**, respectively. The result should look like this:

Secret Template Edit Launcher Configuration

Launcher Type to use: Secure CRT Proxied Process ▼ *

Host: Machine ▼ *

Password: Password ▼ *

Port: <use default> ▼ * Default Value: 22 *

Username: Username ▼ *

9. Click the **Save** button. You can now launch SecureCRT whenever you use the launcher for secrets based off of this template.

In addition to the built in PuTTY and Remote Desktop launchers, Secret Server supports custom launchers. You can customize these process launchers to work with any application that can be started by command-line. Custom launchers pass values to the command-line from the secret text fields. For process launchers to work, the client machine needs to have the program installed and typically needs the program folder in the PATH environment variable.

Note: For more information on launcher arguments see [Custom Launcher Process Arguments](#).

Like the built in launchers, custom launchers run on the users machine not on the web server. Launcher Processes can be set to run either using the credentials of the logged in user or the credentials of the secret. The "Run Process as Secret Credentials" check box is used to switch between these two options.

There are three types of custom launchers to choose from:

- **Process:** Launch a process on the client machine that connects directly to the target system from the client.
- **Proxied SSH Process:** Launch a process on the client machine that proxies its connection to the target system through Secret Server. This applies to an SSH client other than PuTTY (which is a built-in launcher), for example, SecureCRT.

Note: See [Configuring SSH Proxies for Launchers](#).

- **Batch File:** Launch a batch file from the client machine that uses Secret Server information.

Creating Custom Launchers

Procedure

Note: See [Custom Launcher Errors](#) if errors arise.

To create a new custom launcher:

1. Select **Secret Templates** from the **Admin** main menu item. The Manage Secret Templates page appears:

Manage Secret Templates

Active Directory Account Show Inactive

[Back](#) [Edit](#) [+ Create New](#) [Export](#) [View Audit](#) [Active Templates](#) [* Password Requirements](#)

[A Character Sets](#) [Configure Launchers](#) [Configure Secret Template Permissions](#)

Other Templates

[Configure Dependency Templates](#) [Configure Scan Templates](#)

Import Secret Templates

Please paste your XML from the online [Secret Templates Gallery](#) into the box below to add your new Secret Template.

[Import](#)

2. Click the **Configure Launchers** button. The Launcher Types page appears:

Launcher Types	
LAUNCHER TYPE NAME	ACTIVE
Remote Desktop	Yes
PuTTY	Yes
Website Login	Yes
Powershell Launcher	Yes
SQL Server Launcher	Yes
Sybase isql Launcher	Yes
z/OS Launcher	Yes
IBM iSeries Launcher	Yes

Show Inactive

[← Back](#) [+ New](#)

3. Click the **New** button. The Launcher page appears:

Launcher

GENERAL SETTINGS

Launcher Type

Process ▼

Launches the process on the user's machine and replaces \$ parameters with values from the Secret and its associated Secret. For more information see this [KB Article](#)

Launcher Name

*

Active

Use Additional Prompt

Launcher Image

Use Custom Image?



To prevent parameter injection in **Process Arguments** fields below, quotation marks can be inserted around custom parameters.



Example:

`$USERNAME` becomes "`$USERNAME`" prior to launch.

Wrap custom parameters with quotation marks

Record Multiple Windows

Record Additional Processes

Use SSH Tunneling with SSH Proxy

WINDOWS SETTINGS

Process Name

ex. powershell

[How do I configure process arguments?](#)

Process Arguments

ex. -user \$USERNAME -pwd \$PASSWORD -f

Run Process As Secret Credentials

Use Operating System Shell

[Advanced](#)

MAC SETTINGS

Process Name

ex. /Applications/TextEdit.app

[How do I configure Mac process name and arguments?](#)

Process Arguments

ex. -user \$USERNAME -pwd \$PASSWORD -f

Save

Cancel

4. Configure the page. See the following section for details on the settings.
5. Click the **Save** button.

Settings

Note: Not all of the following are available for all types of launchers.

General Settings

The following settings are available in the General Settings section:

- **Active:** Whether the launcher is active for use.
- **Additional Prompt Field Name:** Name of the text field that is prompted for when the user uses the launcher. This value can be referenced in the process arguments with a \$ prefix.
- **Launcher Image:** Upload a custom image for the launcher.
- **Launcher Name:** Name of the launcher that is displayed to the user.
- **Launcher Type:** Select Process, Proxied SSH Process, or Batch File.
- **Record Additional Processes:** Add a comma-separated list of additional process names to record if they are running. When a launcher is in progress and recording, any visible windows from the listed processes are also recorded. This only applies to processes running in your session—other users running the same process are not recorded. The processes themselves are not affected—they remain running after the launch is finished. This setting is only active if Record Multiple Windows is enabled too.
- **Record Multiple Windows:** Records all visible windows of the primary process, not just the primary window of the primary process. This helps record applications with multiple windows or dialog boxes. In addition, if the primary process (or one of its children) spawns child processes, any visible windows are recorded too. For example, if you run the cmd.exe process (the command prompt and then run notepad.exe (Notepad) from the command prompt (cmd.exe), notepad is recorded along with the command prompt. This check box is enabled by default. Enabling this setting is a prerequisite for Record Additional Processes.
- **Use Additional Prompt:** User is prompted for additional information when using the launcher. When selected, the "Additional Prompt Field Name" text box appears. This option is mutually exclusive with "Use SSH Tunneling with SSH Proxy."
- **Use SSH Tunneling with SSH Proxy:** When enabled and the launcher is mapped to a secret template, the user can select host and port fields, and you can provide a default port, which can be referenced using \$HOST and \$PORT in the process arguments. If the launching secret has proxying enabled and a Secret Server SSH proxy is available, those process arguments are replaced with with SSH tunnel values (localhost [127.0.0.1] and a random port).

This causes the launched process to connect to the local tunnel, and traffic then flows from the client to the Secret Server SSH proxy, which connects to the real endpoint. This is useful when users are not allowed to directly connect to the endpoint but Secret Server or distributed engines can. The check box is disabled by default. This option is mutually exclusive with "Use Additional Prompt."

Note: To use this feature, you must first map the host and port fields when you map the launcher to a template. To do so, edit the secret template and click Configure Launcher.

- **User Secret Server RDP Client:** Use the RDP client.
- **Wrap Custom Parameters with Quotation Marks:** Wraps the variables in the process arguments fields with quotation marks. This is a security and disambiguation feature. For example:

Given these process arguments:

```
--host=$HOST --port=$PORT --username=$USERNAME --password=$PASSWORD
```

With no quotation mark wrap, the process arguments for a launcher mapped to a secret might look like this:

```
--host=xyz --port=123 --username=user --password=x x x
```

The final parameter would be ambiguous—the last three characters might be misinterpreted with the process thinking a single "x" is the password. Also, text could be injected, causing the value to be interpreted as another parameter, causing a security issue. Wrapping the parameter values fixes these potential problems:

```
--host="xyz" --port="123" --username="user" --password="x x x"
```

The check box is selected by default.

Windows Settings

The following settings are available in the Windows Settings section:

- **Batch File:** As an alternative to opening a process, upload a .bat file that is downloaded and executed on the client when the user runs a launcher. The file is deleted from the client after execution.
- **Process Arguments:** Process arguments depend on the process that is being launched. View the built-in SQL Server launcher for examples on how the text-entry fields are substituted. For greater flexibility, other secrets can be linked on the Launcher tab on the secret. The text-entry field values from those secrets can also be used in the process arguments using the same prefix `$_[1][FieldName]` syntax as the SSH custom commands. There is a launcher specific token `$SESSIONKEY` that can be passed to the command line. This passes an identifier to the customer launcher that can be used to anonymously check in the secret using the `CheckInSecretByKey` Web service method. Example: `-user $USERNAME -pwd $PASSWORD -f`. See [Custom Launcher Process Arguments](#) for details.
- **Process Name:** Name of the process that is launched. Example: `powershell`
- **Run Process as Secret Credentials:** The process authenticates with the secret credentials (username, domain, and password) instead of the client user that is using the launcher. This can be overridden at the secret level to use a privileged account to run the process.
- **Use Operating System Shell:** Use the OS shell for the launcher. Useful for processes requiring UAC confirmation.

The following settings are available in the **Advanced Windows Settings** section, which is accessible by clicking the **Advanced** link:

- **Escape Character:** The character to use as an escape character in passwords. Escape characters are required to allow the use of characters that are otherwise not allowed in passwords because they have special meaning to the launcher's target application.
- **Characters to Escape:** The characters that require escaping for the target application.

Mac Settings

The following settings are available in the Mac Settings section:

- **Process Name:** Name of the process that is launched. Example: `/Applications/TextEdit.app/Contents/MacOS/TextEdit`
- **Process Arguments:** Process arguments depend on the process that is being launched. View the built-in SQL Server launcher for examples on how the text-entry fields are substituted. For greater flexibility, other secrets can be linked on the Launcher tab on the secret. The text-entry field values from those secrets can also be used in the process arguments using the same prefix `$_[1][FieldName]` syntax as the SSH custom commands. There is a launcher specific token `$SESSIONKEY` that can be passed to the command line. This passes an identifier to the customer launcher that can be used to anonymously check in the secret using the `CheckInSecretByKey` Web service method. Example: `-user $USERNAME -pwd $PASSWORD -f`. See [Custom Launcher Process Arguments](#) for details.

Custom Launcher Errors

Common errors when creating custom launchers:

The process (process name) was not found

The application has not been installed on the machine. If the application was installed, the program folder will need to be added to the path.

The stub received bad data (1783)

The process is set to launch as the credentials of the secret but the username or domain is not correct on the secret or the client machine cannot find the user or domain credentials specified.

Error(740): The requested operation requires elevation

When using "Run process as Secret credentials," even though the credentials have admin privileges, the process cannot be run with elevated privileges from the command prompt using runas. Instead, configure the process launcher as follows (substituting your .exe for program.exe):

- Process Name: cmd.exe
- Process Arguments: /C start /B program.exe /wait

Custom Launcher Process Arguments

Custom launcher process arguments can use a combination of parameters from:

- A field value from the secret.
- A field value from a linked secret.
- User input obtained from a prompt prior to launching.
- `$Host` and `$Port` (for use with a proxied SSH process or SSH tunneling)

Note: For more information, see the [Dependency Token List](#).

Syntax

Parameters are prefixed with a dollar sign `$`. To obtain a value from the secret being launched, use `FieldName`. To obtain a value from a prompt, use `PromptName`. To obtain a value from a linked secret being launched, use `[$n]FieldName` (where `n` represents the `n`th linked secret). Linked secrets can be configured in the Launcher tab.

Examples

```
-user $UserName -color ${1}$Color -Location $LocationPrompt
```

```
-ssh $UserName@$Host -pw $Password -P $Port
```

Creating a Custom TOAD Launcher

You can create a custom launcher for TOAD by entering custom command line parameters in the "Process Arguments" field.

1. Navigate to **Admin > Secret Templates**,
2. Click **Configure Launchers**,
3. Click **New**,
4. Select a Launcher Type:
 - Use **Process** if you would like to use Secret credentials to connect directly to the remote host.
 - Use **Proxied SSH Process** if you have SSH Proxy enabled, to prevent passing Secret credentials to the client by connecting to Secret Server's proxy to interact with the remote host.

Launcher

GENERAL SETTINGS


Launcher Type Process ▼
Launches the process on the user's machine and replaces \$ parameters with values from the Secret and its associated Secret. For more information see this [KB Article](#)

Launcher Name * TOAD

Active

Use Additional Prompt

Launcher Image Use Custom Image?



i To prevent parameter injection in **Process Arguments** fields below, quotation marks can be inserted around custom parameters.

Example:
 \$USERNAME becomes "\$USERNAME" prior to launch.

Wrap custom parameters with quotation marks

Record Multiple Windows

Record Additional Processes

WINDOWS SETTINGS

Process Name C:\Del\Toad for Oracle\Toad.exe ex. powershell

[How do I configure process arguments?](#)

Process Arguments -C \$USERNAME/\$PASSWORD@\$SERVER:\$PORT/\$DATABASE ex. -user \$USERNAME -pwd \$PASSWORD -f

Run Process As Secret Credentials

Use Operating System Shell

[Advanced](#)

5. Enter a **Launcher Name** of your choice.
6. For **Process Name**, enter the location and the Toad executable. The location must exist on the client machine that will run the Toad launcher.
7. For **Process Arguments**, enter your own custom command line parameters, or the following:
 -C \$USERNAME/\$PASSWORD@\$SERVER:\$PORT/\$DATABASE
8. When finished, click **Save**.

Creating and Implementing an Ultra VNC Custom Connection Launcher

Follow the steps below to create an Ultra Virtual Network Computing (VNC) custom connection launcher using Secret Server on a Windows machine.

Create an Ultra VNC Custom Connection Launcher

1. Open Secret Server and click, **Admin > Secret Templates > Configure Launchers**

Manage Secret Templates

Active Directory Account Show Inactive

Active Templates
 Password Requirements
 Character Sets

2. Click the **+New** button at the bottom of the window.

LAUNCHER TYPE NAME	ACTIVE
Remote Desktop	Yes
PuTTY	Yes
Website Login	Yes
Powershell Launcher	Yes
SQL Server Launcher	Yes
Sybase isql Launcher	Yes
z/OS Launcher	Yes
IBM iSeries Launcher	Yes
Secure CRT Proxied Process	Yes
Putty With Port Prompt	Yes
Secure CRT (Proxied)	Yes
Custom PowerShell Launcher	Yes

Show Inactive

3. Under **General Settings**, enter the settings as described below, then click **Save**.

Launcher Type: Process

Process Arguments: \$USERNAME \$PASSWORD \$HOST

Process Name: C:\Program Files (x86)\UltraVNC\vncviewer.exe

Parameters: /user \$USERNAME /password \$PASSWORD -connect \$HOST

Note: You may need to change the Process Arguments if the names of these fields in your Secret Template are something other than "Username" "Password" and "Machine"

GENERAL SETTINGS

Launcher Type

Process 

Launches the process on the user's machine and replaces \$ parameters with values from the Secret and its associated Secret. For more information see this [KB Article](#)

Launcher Name

* VNC

Active



Use Additional Prompt



Launcher Image

Use Custom Image?



To prevent parameter injection in **Process Arguments** fields below, quotation marks can be inserted around custom parameters.

Example:

\$USERNAME becomes "\$USERNAME" prior to launch.

Wrap custom parameters with quotation marks



Record Multiple Windows



Record Additional Processes

WINDOWS SETTINGS

Process Name

ex. powershell

[How do I configure process arguments?](#)

Process Arguments

ex. -user \$USERNAME -pwd \$PASSWORD -f

Run Process As Secret Credentials



Use Operating System Shell



[Advanced](#)

MAC SETTINGS

Process Name

C:\Program Files (x86)\UltraVNC\vncviewer.exe

ex. /Applications/TextEdit.app

[How do I configure Mac process name and arguments?](#)

Process Arguments

\$USERNAME \$PASSWORD \$HOST

ex. -user \$USERNAME -pwd \$PASSWORD -f



Save



Cancel

Assign the Launcher to a Template

Assign the new launcher to an appropriate existing template. To build a new template specifically for VNC connections, see [Creating or Editing Secret Templates](#))

1. Open the template and click the "Configure Launcher" button.



2. Click **+Add New Launcher**



3. Select the VNC custom launcher you just created
4. Map the username and password fields accordingly.
5. Click **Save**.

Overview

A Common Access Card (CAC) or Personal Identity Verification (PIV) smart card is a physical card with an embedded electronic chip that uses a certificate-key pair to authenticate users. The certificate is issued by an authorized organization. The user has a PIN that should be known only to that user, which serves a second factor for two-factor authentication—access requires physical possession of the card, as well as the PIN. The user inserts the card into a card reader, which prompts for the PIN.

Secret Server launchers can pass smart card credentials through Remote Desktop Protocol (RDP) sessions. This is useful when a user needs to authenticate through an RDP session to a resource that requires smart card authentication, for example, a secured network drive that the user attempts to open while using the RDP session.

Currently, you can enable this either globally, via user settings, or per secret:

Enabling Globally with User Settings


1. In Secret Server, click the user icon and select **User Preferences**. The User Preferences page appears.
2. Click the **Settings** tab.
3. In the **Launcher Settings** section, click to enable the **Allow Access to Smart Cards** toggle. The change is automatically saved.

Enabling on a Specific Secret

1. On a Secret with an RDP launcher, click the **Settings** tab.
2. Click the **Edit** link on the **Under RDP Launcher – Personalized User Settings** title bar. The page changes to edit mode.
3. Click to select the **Allow Access to Smart Cards** check box.
4. Click the **Save** button.

Introduction

By default, the launcher is enabled by the **Enable Launcher** setting under **Admin > Configuration**.

The launcher (protocol handler) can be deployed in two ways—with the ClickOnce (the default) or MSI-installable applications. This can also be set in the configuration settings. The latter method allows the launcher to be used in virtualized environments or any environment in which the user does not have access to a Windows Temp directory. The Protocol Handler can be downloaded by clicking the  button on the Dashboard and selecting **Launcher Tools**:

Note: A ClickOnce application is any Windows Presentation Foundation (.xbap), Windows Forms (.exe), console application (.exe), or Office solution (.dll) installed with ClickOnce technology in one of three ways: from a Web page, from a network file share, or from media. See [ClickOnce Security and Deployment](#) for details.

Launcher Tools

LOGIN ASSIST CHROME EXTENSION

Preferred solution for logging into websites from Chrome.

Offers similar functionality to the Web Password Filler, but for a wider range of websites.

Install the Login Assist extension by adding it to the browser from the Chrome web store:
[Chrome Web Store - Secret Server Login Assist](#)

WEB PASSWORD FILLER

Quick, Convenient and Secure logging into Websites.

Install the Web Password Filler by adding this link to your web browser's bookmark bar:
[Secret Server Web Password Filler](#)

- Log into most websites with a single click.
- Click while on a website.
- Automatically fill in the Username and Password.

PROTOCOL HANDLER INSTALLER

Allows launcher to function in virtualized environments. For more information [click here](#).

The MSI can be installed directly or through group policy. A reboot may be necessary on certain operating systems.

[Download Protocol Handler MSI \(64 bit\)](#)

[Download Protocol Handler MSI \(32 bit\)](#)

[Download Protocol Handler PKG \(Apple OSX\)](#)

[Back](#)

MSI Installer

To use the MSI installer (protocol handler installer) following steps below:

1. Go to **Admin > Configuration**.
2. Click the **General** tab.
3. Set the **Launcher Deployment Type** to "**Protocol Handler**".
4. Go to **Tools > Launcher Tools** to download the launcher application.
5. Click the **Download Protocol Handler MSI** link for the operating system you want to install on.
6. Run the MSI file with admin privileges.

Note: The session is kept in check for security reasons with the session process pinging back to Secret Server to ensure it is still valid. This checks secret settings, such as checkout and secret access. If that check fails or the callback times out, Secret Server errs on the side of security and kills the sessions, ensuring access is not allowed.

Installing by Group Policy

The Protocol Handler application runs without requiring any input from the user. The installation may be pushed to your network without any special configuration. For details, see [Installing Protocol Handler through Group Policy](#) (KBA).

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Adding a Program Folder to the Windows PATH

If a launcher does not automatically add the program's folder to the Windows PATH:

1. Right click on **Computer** and go to **Properties**.
2. In the Properties window, click Advanced System Settings.
3. On the **Advanced** tab, click the **Environment Variables** button.
4. In the **System Variables** section scroll to **Path**.
5. Click **Edit** then at the very end of the text box, paste the full path to the folder where the program file is located, but make sure not to replace any existing entries. The list is semi-colon separated.
6. Click **OK** to close the dialogs.

Common Launcher Errors

Two of the most common launcher errors:

- **The process (process name) was not found:** The application has not been installed on the machine. If the application was installed, the program folder needs to be added to the path.
- **The stub received bad data (1783):** The process is set to launch as the credentials of the secret but the username or domain is not correct on the secret or the client machine cannot find the user or domain credentials specified.

Configuring Launchers on the Secret

Custom and SSH launchers provide additional settings on the Launcher tab of the secret for customizing authentication to the target.

- **Run Launcher using SSH Key:** If there is an SSH key set on the secret, it is used by default for authenticating to the target. Alternatively, you can specify a key from a different secret.
- **Connect As:** When an SSH secret is proxied, you can choose to connect as another user and then do an **su** to the current secret's user. This is a common practice for connecting with a lower privileged account and then switching to the root user.

Configuring SSH Proxies for Launchers

Launchers using an SSH connection can alternatively use Secret Server as a proxy rather than the launcher connecting directly to the target system from the machine it is being launched from.

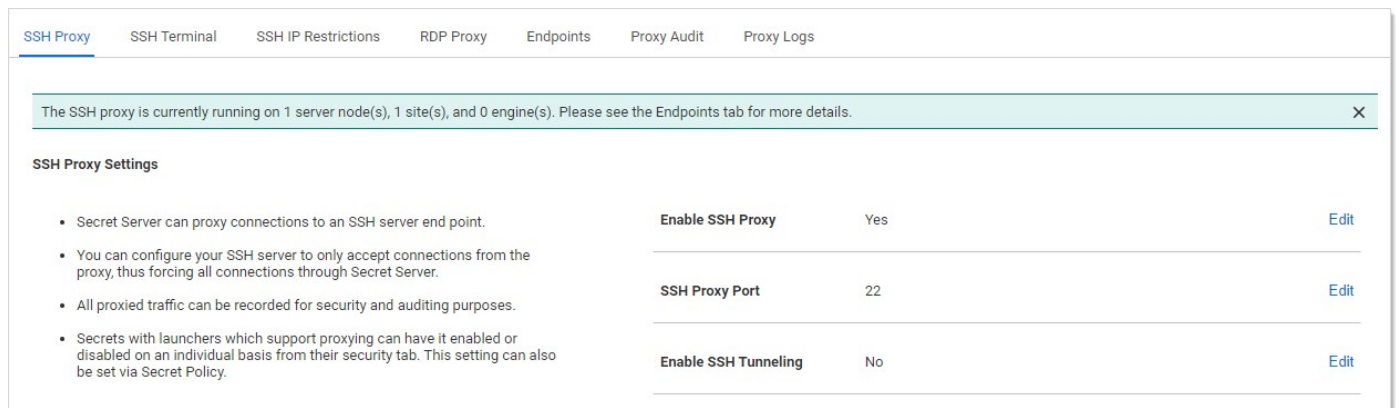
Note: Remote Desktop Services (RDS) is a special version of Secret Server Protocol Handler (SSPH) that can record keystrokes on its own, if configured in Secret Server. See [Session Connector](#) for details.

When proxying is enabled, all RDS sessions are routed through Secret Server. You can configure your SSH server to only accept connections from the proxy, thus forcing all connections through Secret Server. All proxied traffic can be recorded for security and auditing. You can enable or disable proxying for individual launchers. You can also do this using a secret policy.

In Secret Server Cloud, the distributed engine service also supports acting as a proxy for session launchers for greater network flexibility and offloading connections from the Secret Server instance.

To configure this:

1. Select **Admin > Proxying**. The SSH Proxy tab of the Proxying page appears:



The screenshot shows the 'SSH Proxy' tab in the 'Proxying' section. At the top, there are navigation tabs: SSH Proxy (selected), SSH Terminal, SSH IP Restrictions, RDP Proxy, Endpoints, Proxy Audit, and Proxy Logs. A status message at the top indicates: 'The SSH proxy is currently running on 1 server node(s), 1 site(s), and 0 engine(s). Please see the Endpoints tab for more details.' Below this, the 'SSH Proxy Settings' section contains a list of informational bullet points on the left and a settings table on the right. The settings table includes 'Enable SSH Proxy' (Yes), 'SSH Proxy Port' (22), and 'Enable SSH Tunneling' (No), each with an 'Edit' link.

Setting	Value	Action
Enable SSH Proxy	Yes	Edit
SSH Proxy Port	22	Edit
Enable SSH Tunneling	No	Edit

The settings are on the right:

Enable SSH Proxy	Yes	Edit
SSH Proxy Port	22	Edit
Enable SSH Tunneling	No	Edit
Proxy New Secrets By Default	No	Edit
Enable SSH Proxy Inactivity Timeout	No	Edit
SSH Proxy Banner	==== BE A COOL KID AND USE TERMINAL NEXT TIME =====	Edit
Hide passwords from SSH keystroke capture	No	Edit
SSH Proxy Host Fingerprint	SHA1 - d1:0a:f6:0c:be:7c:4a:7a:7b:f1:cc:a8:b6:c2:81:5e:4e:c3:39:66 SHA512 - ed:35:ac:3c:be:ab:7a:ca:2f:61:77:be:1c:7a:d3:1c: da:f1:de:d0:d9:70:94:2a:e4:7c:bf:a8:27:8b:10:57: 55:ce:c1:74:b3:e9:d7:b5:cb:f8:1e:65:5e:4f:95:af: 1c:21:01:95:2d:54:82:b5:91:b8:ce:dd:4f:14:d3:5c MD5 - 04:40:47:4d:51:38:72:b2:78:a9:b7:d3:34:a9:cd:ce	Edit Generate
Days to Keep Operational Logs	30	Edit

2. Scroll down and click the desired **Edit** links to enter your SSH proxy configuration settings.

The **SSH Proxy Settings** are:

- **Enable Proxy:** Enable or disable SSH proxying.
- **SSH Proxy Port:** The port to proxy through. Changing this setting closes all active SSH proxy connections.
- **Enable SSH Tunneling:** SSH Tunneling allows Remote Desktop Sessions to be proxied using the same proxy configuration settings.
- **Proxy New Secrets by Default:** Enable proxying for applicable secrets when you create them.
- **SSH Proxy Banner:** Users connecting through SSH proxy see this text banner. This is not the same as the SSH Terminal Banner.
- **Hide Passwords from SSH Keystroke Capture:** By default proxying records keystrokes. This disables that.
- **SSH Proxy Host Fingerprint:** The Secret Server SSH private key. This can be generated using the **Generate** link.
- **Days to Keep Operational Logs:** Number of days to store operational audit logs.

The **SSH Block List Settings** are:

- **Enable Block Listing:** Block incoming SSH proxy clients that connect and fail to authenticate.
- **Auto Block Max Attempts:** How many times authentication can fail before the connection is blocked.
- **Auto Block Time Frame (minutes):** How long to block connections after authentication tries are exhausted.

Note: Once the period has passed, the address must still be manually removed from the blocked list.

The **Client Override IP Address Ranges** are IP address ranges that you can configure to always allow or always block the incoming connection. Click the **Add** link to add one. Examples:

- 192.168.3.12
- 192.168.42.147-192.168.42.194
- 192.168.3.52/22

Default Launcher Requirements

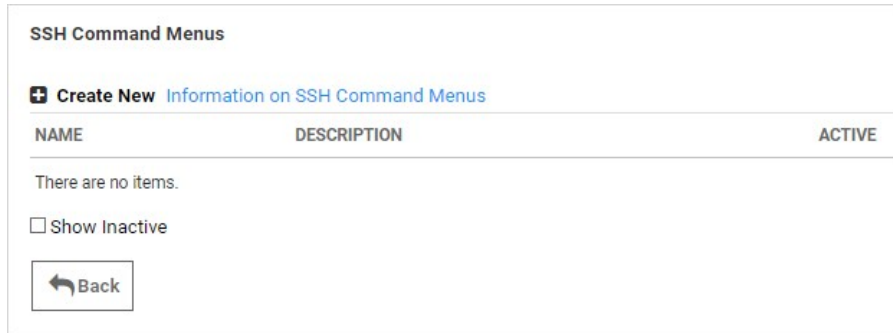
- **IBM iSeries Launcher:** For Secret Server itself, nothing is required. For distributed engines, the `ws3270.exe` file, found in the Secret Server directory, must be copied to the distributed engine folder on the engine, which is typically found at `C:\Program Files\Thycotic Software Ltd\Distributed Engine`.
- **PowerShell Launcher:** Requires PowerShell to be installed. When installed, the program is automatically added to the PATH.
- **SQL Server Launcher:** Requires SQL Server Management Studio to be installed. When installed, the program is automatically added to the PATH.
- **Sybase iSQL Launcher:** Requires that `isql.exe` is installed.
- **z/OS Launcher:** For Secret Server itself, nothing is required. For distributed engines, the `ws3270.exe` file, found in the Secret Server directory, must be copied to the distributed engine folder on the engine, which is typically found at `C:\Program Files\Thycotic Software Ltd\Distributed Engine`.

Managing Superuser Privilege

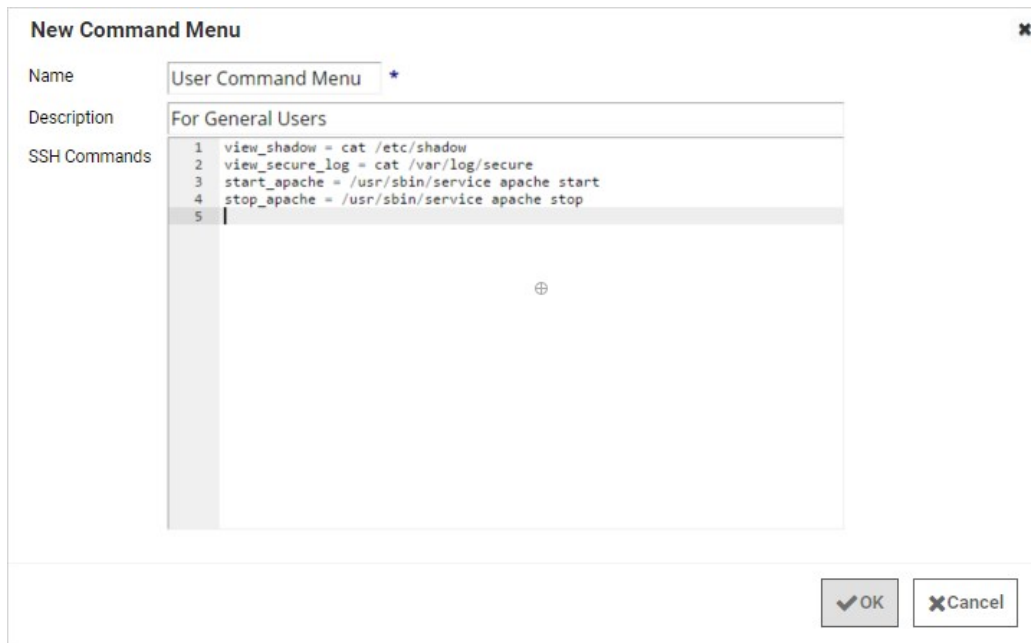
Administrators can create command menus for use with a proxied SSH connection to restrict what commands can be run by users or groups on the connected server. This feature requires an additional license. To add a command menu:

Note: For details, see [SSH Command Menus](#).

1. Navigate to **Admin > All**.
2. Click the **SSH Command Menus** button.



3. Click the **Create New** button.
4. Type a name, description and the SSH commands:



Once one or more command menus have been created, access can be controlled to individual Unix SSH secrets.

On the **Security** tab of a secret that can use a proxied PuTTY session, proxy must be enabled as well as command menu restrictions. If **Allow Owners Unrestricted SSH Commands** is enabled, any user who is an owner of the secret has unrestricted use of the PuTTY session, that is, that user is able to type in commands as in a normal session. Additionally, other groups can be assigned the Unrestricted role as well.

In the following example, the "admin" group is unrestricted, while everyone who is not in the admin group is restricted to only being able to run the commands that are enumerated in the user command menu, created above.

SSH Unix Secret (Unix Account (SSH))

General
Personalize
Expiration
Launcher
Security
Dependencies

Require Check Out

Enable DoubleLock
(You have not created a DoubleLock password.)

Enable Requires Approval for Access

Require Comment

Enable Proxy

Hide Launcher Password

Enable SSH Command Restrictions

Allow Owners Unrestricted SSH Commands

Name	SSH Command Menu		
admin	Unrestricted		🗑️
Everyone		👁️	🗑️

Add New

--Groups-- ▾

Customize Password Requirement

Save
Cancel

A user who is subject to SSH Command Restrictions are presented with a screen similar to the following when connecting to an SSH session:

```
Using username "729ddaef-38d0-48e0-b9dc-d4911e76d0c1".
1. User Command Menu
   ?. Show Command Menus
   exit. Exit session
Last login: Thu Mar 17 12:38:46 2016 from 192.168.60.153
[runscripts@centostestserver ~]$ █
```

The user simply enters the number of the command menu to see available commands, or types "?" to display the options again.

```
Using username "729ddaef-38d0-48e0-b9dc-d4911e76d0c1".

1. User Command Menu

    ?. Show Command Menus
    exit. Exit session
Last login: Thu Mar 17 12:38:46 2016 from 192.168.60.153
[runscripts@centostestserver ~]$ 1

1. view_shadow = cat /etc/shadow
2. view_secure_log = cat /var/log/secure
3. start_apache = /usr/sbin/service apache start
4. stop_apache = /usr/sbin/service apache stop

    up. Return to Command Menu selection. You may also type ..
    ?. Show Commands
    exit. Exit session

[runscripts@centostestserver ~]$ █
```

Only the commands listed can be run by this user. The user can either enter the number of the command to be run, or the name of the command, which is the word to the left of the equal (=) sign. Other options are available (as shown) to navigate through the available command menus, display help, or exit the session.

Session Recording and Launchers

Session recording provides an additional level of security by recording a user's actions after a launcher is used. Session recording works for any launcher, including PuTTY and SSH, Windows Remote Desktop, Microsoft SQL Management Studio, and custom executables. The resulting movie is viewable from the secret audit. Session recording can be toggled on or off globally on the Configuration page and set for individual secrets on the Security tab. Detailed information on supported codecs can be found in [Session Recording](#). When a user launches a session with session recording enabled, a brief message is displayed to inform the user that their actions are recorded.

Note: When multiple Launchers are enabled for a secret template, enabling session recording for a secret applies the setting to all launchers for that secret.

On the Secret View page, clicking the Launcher icon launches the Remote Desktop, PuTTY, or custom session directly from the browser or log into the website. The mapped text fields are passed to the launcher for automatic authentication.

If the machine is set for Remote Desktop, the console launches and allows the machine to be specified from the RDP dialog.

If the Host is set to <user>, a prompt asks for the specific machine before launching the PuTTY session.

For some browser security levels, you might need to click **Allow** for the launcher application to open.

Note: The View Launcher Password permission can be removed to prevent users from viewing the credentials but can still use the authentication session to access the computer.

The settings under the Launcher tab are used for secrets that are enabled for SSH and custom launchers.

Note: This capability applies to Secret Server 10.9 or later. That is Secret Server Protocol Handler 6.0.0.28 or later.

You can limit the domains that a launcher connects to. If this is not set, then nothing changes—the launcher can connect to any domain. If it is set, however, Secret Server refuses to connect to any domains that are not explicitly allowed.

This setting is done via a Windows Group Policy Object (GPO) administrative template XML file (.admx). The file specifies the registry key that are changed when the GPO is edited. Download that file here: [LimitLauncherDomainPolicyDefinitions.zip](#).

For details on using these files, see [How to create and manage the Central Store for Group Policy Administrative Templates in Windows](#) on the Microsoft site. The settings are present in both user and machine configurations in the group policy editor. If both are specified, then only the machine configuration is used (the user configuration is completely ignored). This is because the user configuration is stored in part of the registry that does not require administrator access to edit, so the machine configuration should be used in most cases.

The Group Policy valid values are just domain names, like `example.com`, or IP addresses, like `192.168.1.2`. No port should be specified, and no scheme. A value like `https://example.com` is not valid, because it has `https://` in the front. Ports are also invalid, so `example.com:885` will not match. The correct value would simply be `example.com`. Wildcards are not supported, but subdomains matter, so a value of `example.com` will not match `something.example.com`.

In an organization running multiple instances of Secret Server, some users might find themselves having to repeatedly uninstall and reinstall different versions of the protocol handler to match the different instances of Secret Server.

To enable Secret Server to simultaneously support multiple versions of the protocol handler, you just need to disable the protocol handler auto-update function using the procedure below.

Note: Disabling auto-update for forward and backward compatibility is supported on the protocol handler only. The ClickOnce launcher and the Mac protocol handler do not support disabling auto-update.

Prerequisites

- Secret Server Cloud 10.8+
- Secret Server On-Premises 10.8+
- Protocol Handler 6.0.0.23 or higher on your PC

Setup Steps and Configuration

1. In Secret Server, click **Admin > Configuration**, then click the **General** tab.
2. Scroll to the bottom of the page and click **Edit**.
3. In the **Launcher Settings** section, uncheck the box next to **Enable Protocol Handler Auto-Update**.

LAUNCHER SETTINGS (RUNTIME)	
The settings in this section apply only to launchers that connect to this server. To change the Protocol Handler from this server, see the next section.	
Enable Launcher	<input checked="" type="checkbox"/>
Launcher Deployment Type Explanation	
Launcher Deployment Type	Protocol Handler ▾
Enable Protocol Handler Auto-Update	<input type="checkbox"/>
Send Secret URL to Launcher	<input checked="" type="checkbox"/>

4. Click **Save**.
5. Ensure that the **Enable Protocol Handler Auto-Update** function is now labeled, **No**.


LAUNCHER SETTINGS (RUNTIME)	
The settings in this section apply only to launchers that connect to this server. To the Protocol Handler from this server, see the next section.	
Enable Launcher	Yes
Launcher Deployment Type	Protocol Handler
Enable Protocol Handler Auto-Update	No

You can re-enable protocol handler auto-update at any time by following the steps above and re-checking the box next to **Enable Protocol Handler Auto-Update**. When you re-enable auto-update, users will be required to install the latest instance.

Note: While protocol handler auto-update is disabled, each user must manually update their installed protocol handler as necessary on a machine-by-machine basis. The MSI can be installed directly or through Group Policy. A reboot may be necessary on certain operating systems.


Manually Updating Protocol Handler

1. In Secret Server, click **Admin > See All**. A page opens with **What are you looking for?** at the top.
2. Click **Tools & Integrations**, then **Launcher Tools**



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

TOOLS & INTEGRATIONS

- Launcher Tools
- Connection Manager
- SDK Client Management
- Privilege Manager
- Privileged Behavior Analytics
- DevOps Secrets Vault
- Slack Integration

3. On the **Launcher Tools** page, click **Download Protocol Handler (64-bit)** to download the file.

The MSI can be installed directly or through Group Policy. A reboot may be necessary on certain operating systems.

- [Download Protocol Handler \(64-bit\)](#)
- [Download Protocol Handler \(32-bit\)](#)
- [Download Connection Manager PKG \(macOS\)](#)
- [Download Protocol Handler Group Policy Templates](#)

[Hashes](#)

4. Follow the steps in the installation wizard.

The Secret Server protocol handler has several administrative settings that you can configure through Microsoft's Group Policy Objects (GPOs) or through Secret Server itself.

Important: We **strongly** recommend using GPOs instead of Secret Server.

Available Settings

Allowed Secret Server Domains

This setting controls which domains or IP addresses the protocol handler installation may connect to. If the setting is unset or disabled, then the protocol handler is allowed to connect to any domain. If one or more comma-separated values are provided, then the protocol handler is blocked from accessing any domains or IP addresses not included in the list.

The protocol handler performs a string match against the URL it receives. It does not attempt to resolve domain names to IP addresses. Values in this list should match only the domain or IP address portion of the actual URL used to access Secret Server. For example, if users access your installation via `https://example.com/SecretServer`, then `example.com` should be added to the list. If `example.com` resolves to the IP address `192.168.1.5`, then adding that IP address *will not* allow access to the domain if users actually access it via `example.com`.

Wildcards are not supported, but subdomains do matter. The above entry for `example.com` would not allow `www.example.com`; the two may need to be added separately depending on your configuration. Ports and protocols are also unnecessary—only the domain portion is checked. For example, do not include an entry in the list like `https://example.com` or `example.com:885` as both are invalid. Simply using `example.com` covers these scenarios.

Disable Auto-Update

This setting ensures protocol handler will never auto-update itself, even if told to by the Secret Server installation that it connects to. When the setting is enabled, protocol handler installations need to be updated either manually or as part of your organization's regular program-update process.

Configuration Methods

Choosing the Configuration Method

Important: If your domain is configured to use GPOs, we **strongly recommend** using that to configure the protocol handler.

Why use GPOs instead of Secret Server?

- GPOs are more resilient, as Windows reapplies settings if they are deleted from the registry. Settings applied through Secret Server have no such resilience.
- GPOs are centrally managed along with other settings for machines in your domain.
- For security reasons, Secret Server's configuration can only be applied during the initial installation of Secret Server. If you change these settings within Secret Server, users must reinstall the program before they will be applied. GPOs do not have this restriction.

Configuring GPOs

You can download GPO definitions for your version of Secret Server from the Launcher Tools page of the Admin section of Secret Server. For details about using these policy definition files, see [How to create and manage the Central Store for Group Policy Administrative Templates in Windows](#). Both machine and user configurations are available as needed, but machine configurations will always override user configurations—if a machine configuration is configured, the user configuration is completely ignored.

Settings are available in the group policy editor under **(Computer/User) Configuration > Administrative Templates > Secret Server Protocol Handler**.

Configuring Settings During Secret Server Installation

If you do determine that using Secret Server's settings are necessary, you can configure them via the Configuration page in the Admin section of Secret Server, in the "Protocol Handler Settings (Install-Time)" section. Enabling these settings causes downloads to generate a zip rather than an MSI file. The zip file contains a batch file that configures the install-time settings. These settings only update when the protocol handler is manually reinstalled or updated—changing them later on through Secret Server has no effect on protocol handlers that are already installed on user machines.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Adding Remote Desktop Launchers

1. Click **Add New Launcher** to add a launcher to the template.
2. On the following page, select a launcher type from the drop-down menu. The text-entry fields below reflect the text-entry fields necessary to map to the launcher. In the case of a custom launcher, these text-entry fields are used to run the launcher process if the launcher is configured to run as secret credentials.
3. Choose a secret text-entry field in the drop-down menu on the right to map to each launcher value on the left. See the following section for further details on editing launcher configuration.
4. Click the **Save** button to add the launcher to the template.

Browser Configuration

Remote Desktop (RD) launchers require the following:

- **Firefox Configuration:** Firefox requires a helper add-on application to run the RD and PuTTY launchers. The Microsoft .Net Framework Assistant add-on and .NET framework version .NET 4.8+ needs to be installed.
- **Chrome Configuration:** If using ClickOnce, Chrome requires a Helper Add-on application to run the RDP and PuTTY Launcher. The ClickOnce add-on for Google Chrome Add-on needs to be installed. The launcher requires .NET framework version .NET 4.8+ as well.
- **SSL Certificates:** SSL must be set up properly for the RD launcher to work correctly. If Secret Server is using SSL certificates, they must be trusted at the user's computer. This is only an issue with self-created certificates.

Editing RD Launchers

Click **Edit** to modify the settings for a launcher that has already been added to the template. For a launcher to work properly, Secret Server requires credentials to be taken from secret text-entry fields. Fields must be assigned their corresponding credentials from the list. In addition to the secret fields, the domain can be mapped to <blank>, which passes an empty string to be used with local accounts, and the machine or host can be mapped to <user input>, which prompts the user for a specific machine to be used with domain accounts.

In cases where there are multiple endpoints to connect to, such as with a domain account, the machines can be restricted to a set list. Under the **Advanced** section of the secret template launcher configuration, enable **Restrict User Input**. When that option is on, the launcher shows a drop down of machines to connect to, based on a comma-separated list in the specified secret field.

Enforcing a Secret's RDP Launcher Policy

As a secret owner, you can create an RDP launcher policy to control who changes RDP launcher settings, such as clipboard access, drives, and more for that secret. The policy applies to all users of the secret. In prior versions, users with access to a secret could set their own preferences for RDP launcher settings. Secret owners can now implement a consistent access policy for all users.

To enforce RDP settings for all users of your secret:

1. Click the desired secret name link for a secret you own on the default **All Secrets** page. The General tab appears for that secret:

Personal Folders > [User] > My RDP Launcher ☆

General Security Audit Sharing Settings List Fields Metadata Options ▾

Basic Information [Edit All](#)

Contains general information, such as the secret's template type, the domain, the username and password, and other basic information. Depending on permissions, you may not be able to see or edit these fields.

Secret Name *	My RDP Launcher	Edit
Secret Template	Windows Account RDP Launcher (Prompt, White List)	Edit
Computer	Test	Edit
Username	Test	Edit
Password	*****	Edit
Notes	None	Edit

Launchers

Provides a launcher to easily access an account using your secret's credentials.

[RDP Launcher](#)

No Active Sessions

Expiration and Heartbeat

Sets when a secret's credentials are confirmed to work (heartbeat) and must be changed (expiration). Administrators use these settings to enforce your organization's security policy. Expiration is set in the secret template.

Expiration	Expires in 29 days. (Expires every 30 day(s))
-------------------	---

2. Click the the **Settings** tab:

General	Security	Audit	Sharing	<u>Settings</u>	List Fields	Metadata	Options ▾						
<p>Email Notifications Edit</p> <p>Choose your personalized email settings for this specific Secret.</p> <table border="0"> <tr> <td>Send Email When Viewed</td> <td>No</td> </tr> <tr> <td>Send Email When Changed</td> <td>No</td> </tr> <tr> <td>Send Email When Heartbeat Fails</td> <td>No</td> </tr> </table>								Send Email When Viewed	No	Send Email When Changed	No	Send Email When Heartbeat Fails	No
Send Email When Viewed	No												
Send Email When Changed	No												
Send Email When Heartbeat Fails	No												
<p>RDP Launcher Edit</p> <p>RDP launcher settings can be configured under user preferences and per Secret. The settings defined will be passed to the launcher to configure the RDP session.</p> <table border="0"> <tr> <td>Enforce these RDP Settings for All Users</td> <td>No</td> </tr> <tr> <td>Connect to Console</td> <td>No</td> </tr> <tr> <td>Allow Access to Printers</td> <td>No</td> </tr> </table>								Enforce these RDP Settings for All Users	No	Connect to Console	No	Allow Access to Printers	No
Enforce these RDP Settings for All Users	No												
Connect to Console	No												
Allow Access to Printers	No												

3. Click the **Edit** button In the **RDP Launcher** section. The settings become editable:

RDP Launcher															
<p>RDP launcher settings can be configured under user preferences and per Secret. The settings defined will be passed to the launcher to configure the RDP session.</p>	<table border="0"> <tr> <td>Enforce these RDP Settings for All Users</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Connect to Console</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Allow Access to Printers</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Allow Access to Drives</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Allow Access to Clipboard</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Allow Access to Smart Cards</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Use Custom Window Size</td> <td><input type="checkbox"/></td> </tr> </table>	Enforce these RDP Settings for All Users	<input type="checkbox"/>	Connect to Console	<input type="checkbox"/>	Allow Access to Printers	<input type="checkbox"/>	Allow Access to Drives	<input type="checkbox"/>	Allow Access to Clipboard	<input checked="" type="checkbox"/>	Allow Access to Smart Cards	<input type="checkbox"/>	Use Custom Window Size	<input type="checkbox"/>
Enforce these RDP Settings for All Users	<input type="checkbox"/>														
Connect to Console	<input type="checkbox"/>														
Allow Access to Printers	<input type="checkbox"/>														
Allow Access to Drives	<input type="checkbox"/>														
Allow Access to Clipboard	<input checked="" type="checkbox"/>														
Allow Access to Smart Cards	<input type="checkbox"/>														
Use Custom Window Size	<input type="checkbox"/>														
<div style="text-align: right;"> <input type="button" value="Cancel"/> <input type="button" value="Save"/> </div>															

4. Click to select the **Enforce these RDP Settings for All Users** check box.

Note: If you do not see this check box, you are not an owner of the secret.

5. Enable or disable the RDP Launcher settings as desired for the secret. "Use Custom Window Size" displays additional text boxes for

entering dimensions.

6. Click the **Save** button. Now, those restrictions are applied to all using the secret.

When RDP Launcher settings are enforced for all users, all non-owner users of the secret see the settings defined by the owner but are unable to edit them. These settings are applied to the RDP session when launched using the default RDP launcher.

Setting Up Secret Templates for RD Launchers

Launchers can be accessed from any secret created from a properly configured template.

By default, the templates Windows Account, Active Directory Account, Cisco Account (SSH), HP iLO Account (SSH), Unix Account (SSH), Web Password, and SQL Server Account have the launcher configured.

Secrets can be configured for the launcher from within the Secret Template Designer page.

Clicking **Configure Launcher** displays the options available.

To upgrade or apply a fix to the Secret Server Mac Launcher, you must remove the version that is already installed. But first you must prevent the launcher from restarting, and terminate all processes related to the Delinea Launcher.

1. Open **Terminal** and type `launchctl remove com.thycotic.thycoticD`

This step should remove the ThycoticDaemon and prevent the launcher from restarting, but you might need to perform the step more than once.

2. Open **Activity Monitor**.
3. Force Quit the **ThycoticLauncher** process and all related processes. See [Quit an app or process in Activity Monitor on Mac](#).
4. Open **Finder**.
5. Navigate to **Application > Delinea**.
6. Right-click **ThycoticLauncher** and select **Move to Trash**.
7. Empty your trash.

Overview

Normally, Secret Server requires installing additional software such as Connection Manager or Secret Server Protocol Handler (SSPH) on the end-user computers to launch secrets, such as RDP, SSH, or custom, and optionally record the session.

With Secret Server Session Connector (SSSC) installed on a Remote Desktop Services (RDS) server, anyone who can download and launch a standard Remote Desktop Protocol (RDP) shortcut file can have the same experience. The RDS server itself runs a special SSPH for RDS—SSPH (RDS) as a remote app to record the sessions, so end-users do not need to install any additional software.

The SSSC feature is largely scalable and can be set up using a single RDS server, a load-balanced cluster of RDS servers, or multiple load-balanced clusters of RDS servers. Before you set up the SSSC feature, there are some baseline requirements for those RDS servers and on your domain.

Note: SSPH (RDS) is sometimes referred to as RDPWin in this topic. RDPWin is the main executable that SSPH runs to launch and record sessions.

Table: Terms and Definitions

RDP	<i>Remote Desktop Protocol.</i> A Microsoft protocol for remote control of computers.	
RDPWin	The primary executable for SSPH.	
RDS	<i>Remote Desktop Services.</i> Remote control services (using RDP) provided by a dedicated server or servers.	
SSPH	<i>Secret Server Protocol Handler.</i> SSPH is an application on an end-user's machine. It enables communication between Secret Server and that client machine. It also provides the files needed by secret launchers.	*Secret Server *
SSPH (RDS)	<i>Secret Server Protocol Handler, RDS Version.</i> A special SSPH for use with SSSC that enables optional keystroke recording.	
SSSC	<i>Secret Server Session Connector.</i> SSSC is the subject of this topic.	

[Unexpected Link Text](#)

Connection Sequences

Figure: Session Connector Connection Sequences for an RDS Server.

Session Connector Connection Sequence

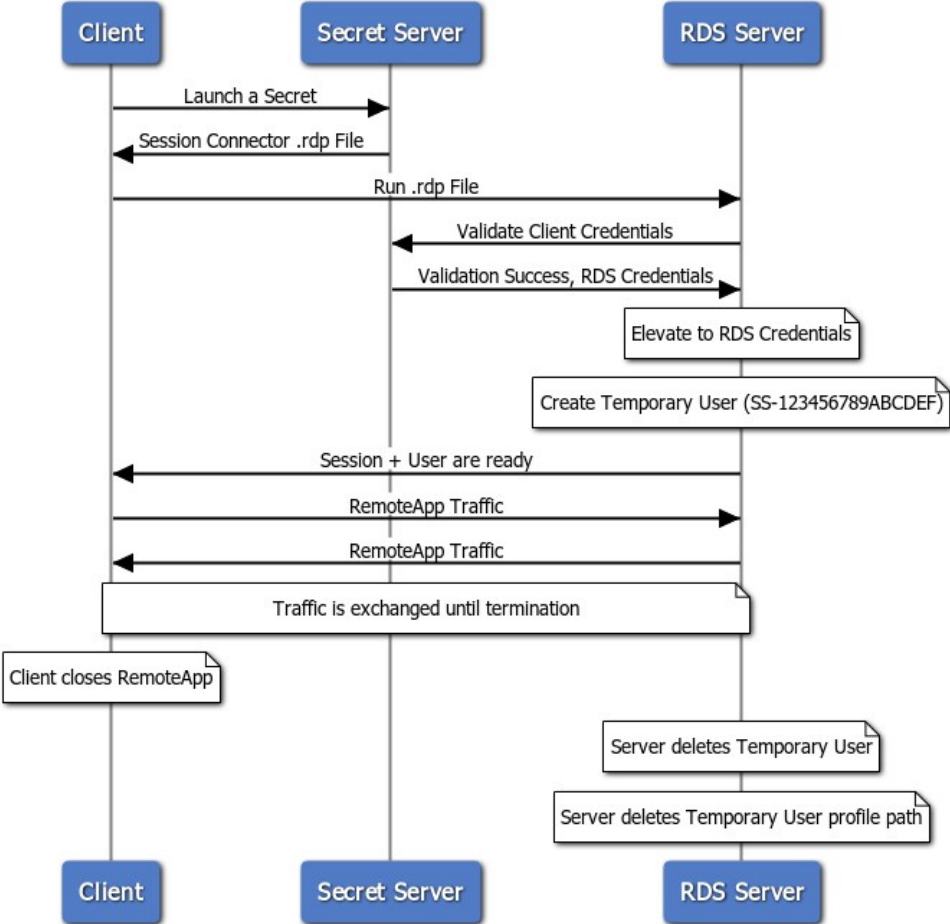
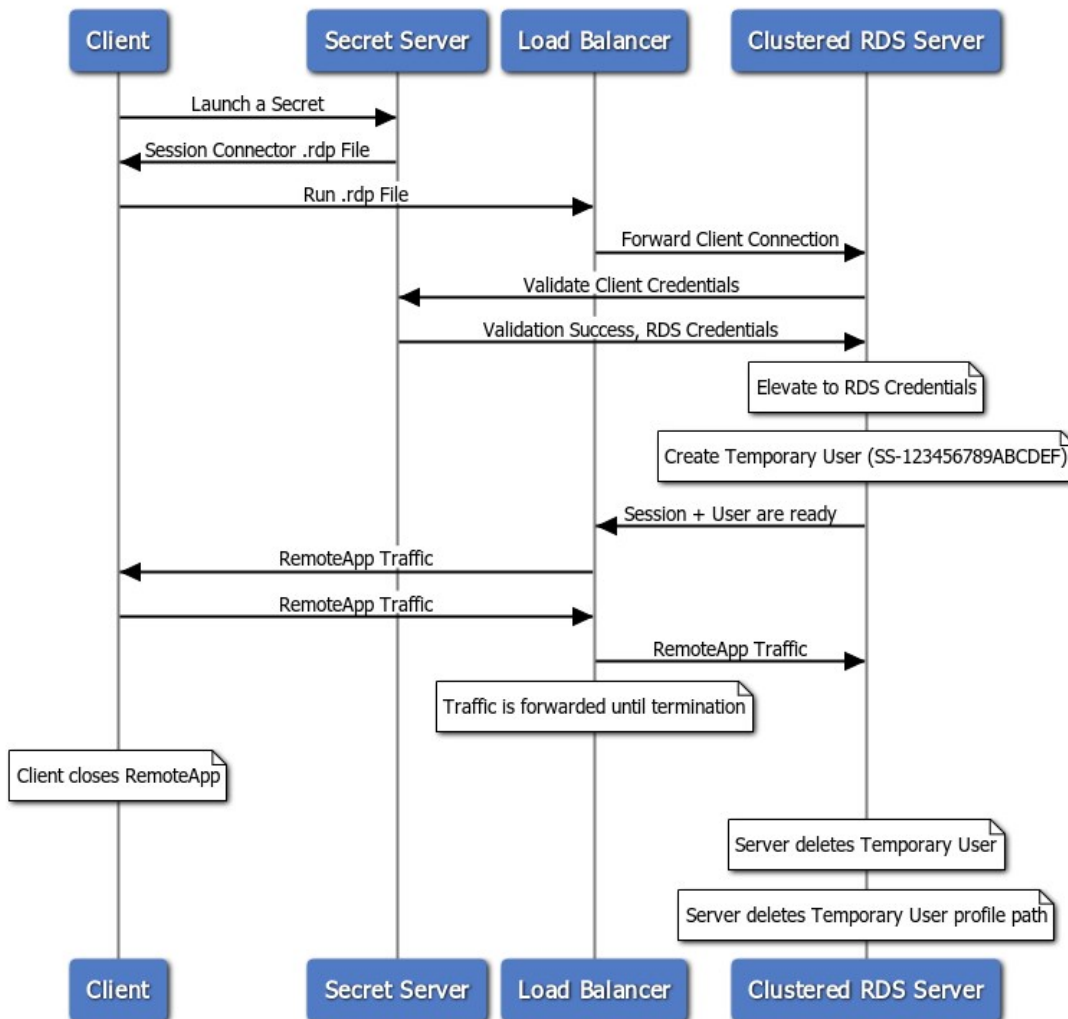


Figure: Session Connector Connection Sequences for Clustered RDS Servers.

Session Connector Connection Sequence



Download

Session Connector is downloaded separately from Secret Server. Go to [Session Connector Downloads](#) for download links and hashes.

Configuration

Note: To comply with Microsoft licensing requirements, there is an additional constraint on which Microsoft Windows Server version you can use as the RDS server for session connector.

If you use Microsoft User Client Access Licenses (CALs), you cannot use Windows Server 2019. You must use Windows Server 2012 or 2016. If you use Microsoft Device CALs, you can use any supported version of Windows Server.

Task 1: Reviewing RDS Server Prerequisites

- Each RDS server should be a 64-bit installation of Windows Server 2012, 2016 or 2019.
- You **MUST** have access to the console session (non-RDP) to install the SSSC integration. This is in case of any of any errors during

installation, which may disable RDP access to the server.

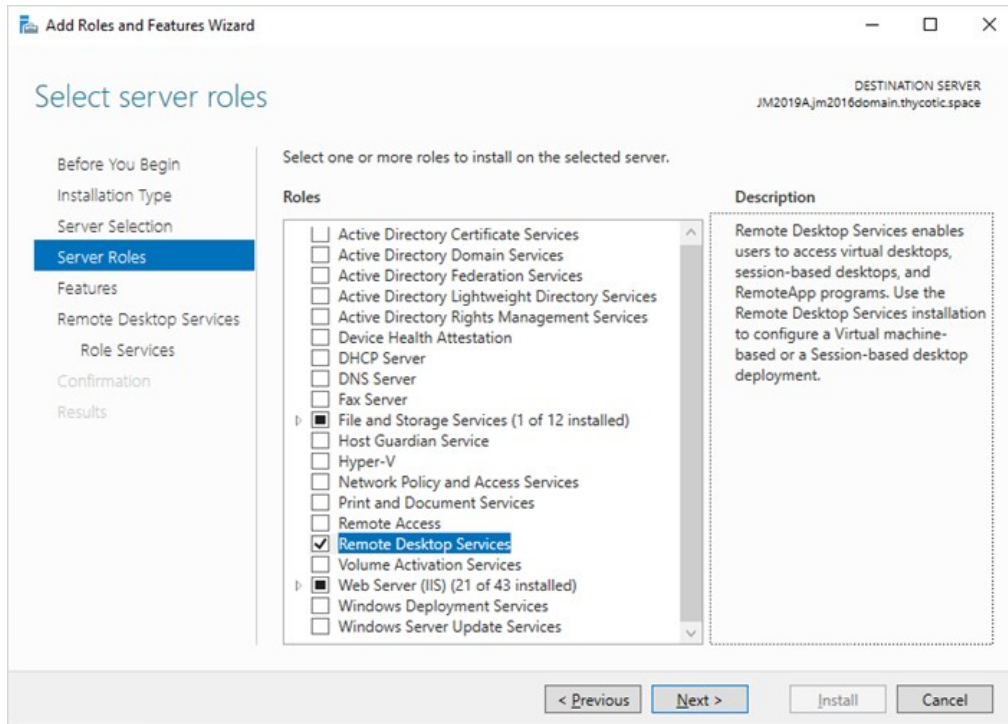
- Each RDS server must be domain joined. Configuration of the RDS feature requires being logged in as a domain user.
- Each RDS server needs to have a recent version of the C++ redistributable installed (v14.26.28720 or higher, May 2020):
 - Download: https://aka.ms/vs/16/release/vc_redist.x64.exe
 - More info: [The latest supported Visual C++ downloads](#)
- Each RDS server needs to have a credential available to manage temporary users. This credential should be able to create and delete local users and add users to the Remote Desktop Users group. If you plan to use one or more load-balanced clusters of RDS servers, this credential should be a domain user and will be used for all servers inside of a cluster. We recommend one domain user per cluster. This credential will be referred to as the **RDS Credential**
- Each RDS server needs to have the RDS Session Host Windows feature installed. See the next section.

Task 2: Setting up RDS Services

Step 2.1: Installing Remote Desktop Services—Remote Desktop Session Host

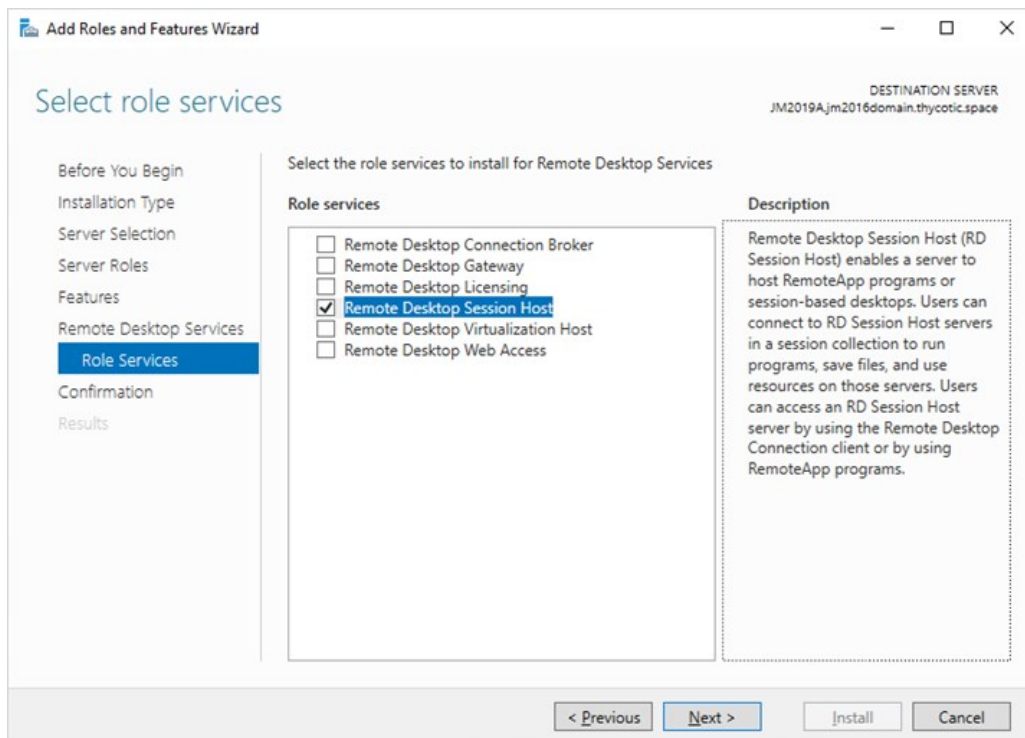
Note: SSSC cannot function without this feature and will refuse to install if it is not present. **RDS requires additional remote desktop licensing from Microsoft.** This may also require installing the remote desktop licensing feature if you do not already have a licensing server available in your environment. See [Activate the Remote Desktop Services license server](#) for details.

1. In Server Manager, click **Add roles and Features**. The Add Role and Features wizard appears.
2. Click the **Next >** button. The Installation Type page appears.
3. Select **Role-based or feature-based installation**.
4. Click the **Server Roles** menu item (or press **Next >** twice). The Select server roles page appears:

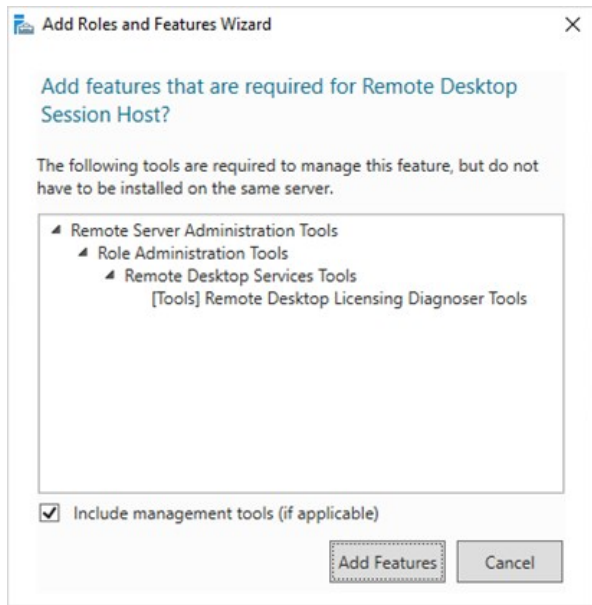


5. Click to select the **Remote Desktop Services** check box.

6. Click the **Next >** button. The Select role services page appears:



7. Click to select the **Remote Desktop Session Host** check box.
8. Click the **Next** > button. The Add features... page appears:



9. Click the **Add Features** button. A "Confirm installation selections" page appears.
10. Click the **Install** button.

Step 2.2: Setting up RDS in \$Secret Server\$2

1. Enable the **Session Connector** advanced configuration setting. For more instructions on this please follow the steps under **Configuring Session Connector Settings** below.
2. Go to **Admin > Configuration > General** tab.
3. Ensure the **Secret Server Custom URL** setting is set to a valid URL for your Secret Server. This URL is given out to SSPH launches (including SSSC and RDS SSPH) to ensure it knows how to connect back to Secret Server. Use HTTPS for maximum security. In fact, as of Secret Server version 10.9, SSPH and SSSC both refuse to connect to HTTP.
4. Create a Secret for the **RDS Credentials** mentioned above. If the credential is a local account, use a Windows Local Account secret, and if it is a domain user, use an Active Directory secret.
5. Create application users in Secret Server, one for each of the RDS server machines. See **Creating RDS Application Accounts** for details.
6. Share the secret created for the RDS credential mentioned above with the RDS application accounts that will be used by the RDS server(s). See **Application Account RDS Credential Sharing**.
7. Create SSSC custom launchers. For example, if you wanted to run an RDP session on the RDS server, you should configure a custom SSSC launcher that uses the built-in RDP launcher as its child launcher. See **Configure Session Connector Custom Launchers**.
8. Assign your SSSC custom launchers to the secret templates you want to launch from. See **Assign Session Connector Custom Launchers to Secret Templates**.
9. Configuration and setup is finished for Secret Server, but there are still some things you need to do inside of the RDS servers before

setup is complete.

Step 2.3: Configuring Session Connector Settings

Enable SSSC:

1. Go to <https://<your Secret Server location>/ConfigurationAdvanced.aspx>.
2. Click the **Edit** button at the bottom of the page.

Important: Do not change any other settings on this page without consulting Delinea Support. Your Secret Server installation could malfunction.

3. Set **Session Connector** to **True**.
4. (Optional) Set **Session Connector Session Timeout** if you do not want to use the 900-second default (15 minutes). SSSC .RDP files are valid for this many seconds (only for a single use). If set to 0 or below, the default is used.
5. (Optional) Set **Session Connector Allow Connection Sharing** to **True**. This changes the value of "disableconnectionssharing" in the output SSSC RDP files. If true, this speeds up concurrent launches into the same RDS server quite a bit by re-using the existing Windows sessions, at the risk of sometimes causing errors if launching a new session while an old session is in the middle of closing. The default is false.
6. Click the **Save** button.

Task 3: Setting up RDS

Step 3.1: Installing the Secret Server RDS Protocol Handler

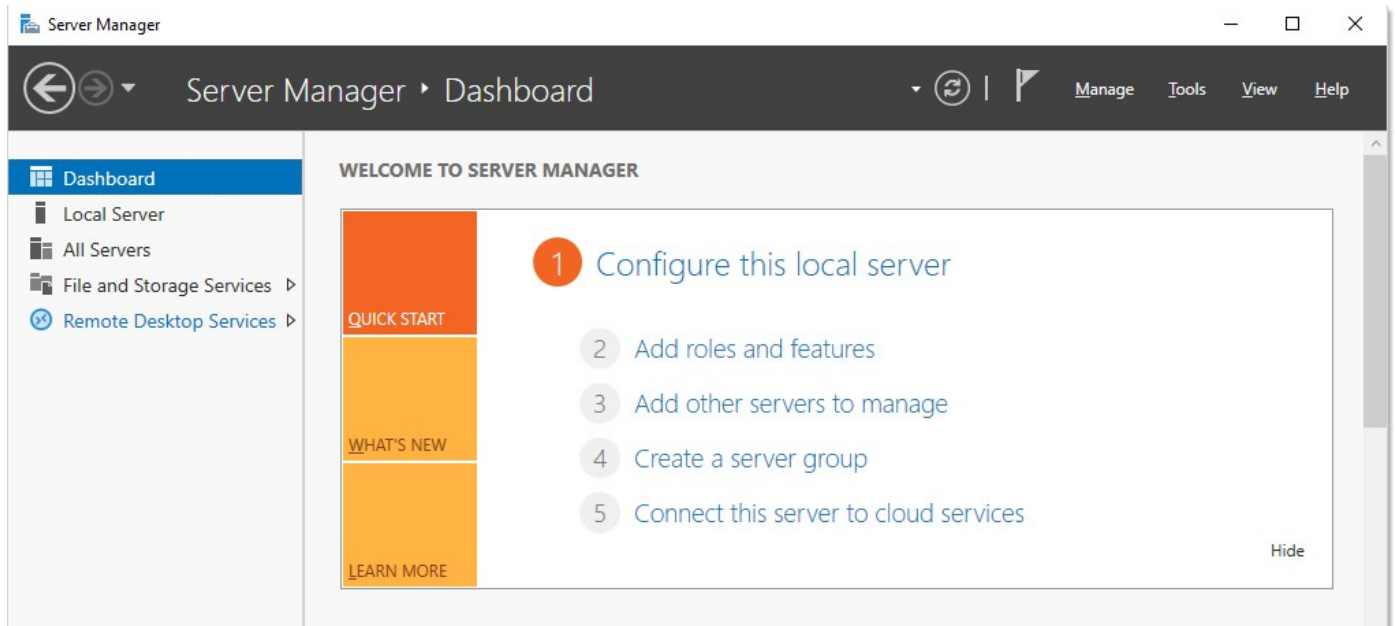
1. Go to the [Session Connector Downloads](#) page.
2. Download the SSPH (RDS) installer file, SSProtocolHandlerRDS.msi .
3. (Optional) Ensure the listed hash value matches that for the file.

Note: SSPH (RDS) is a special version of SSPH that can record keystrokes on its own, if configured in Secret Server. Due to this optional keystroke recording, you may need to allowlist the RDPWin.exe file (the primary executable for SSPH) in any antivirus software running on the server. This is not currently necessary with Windows Defender.

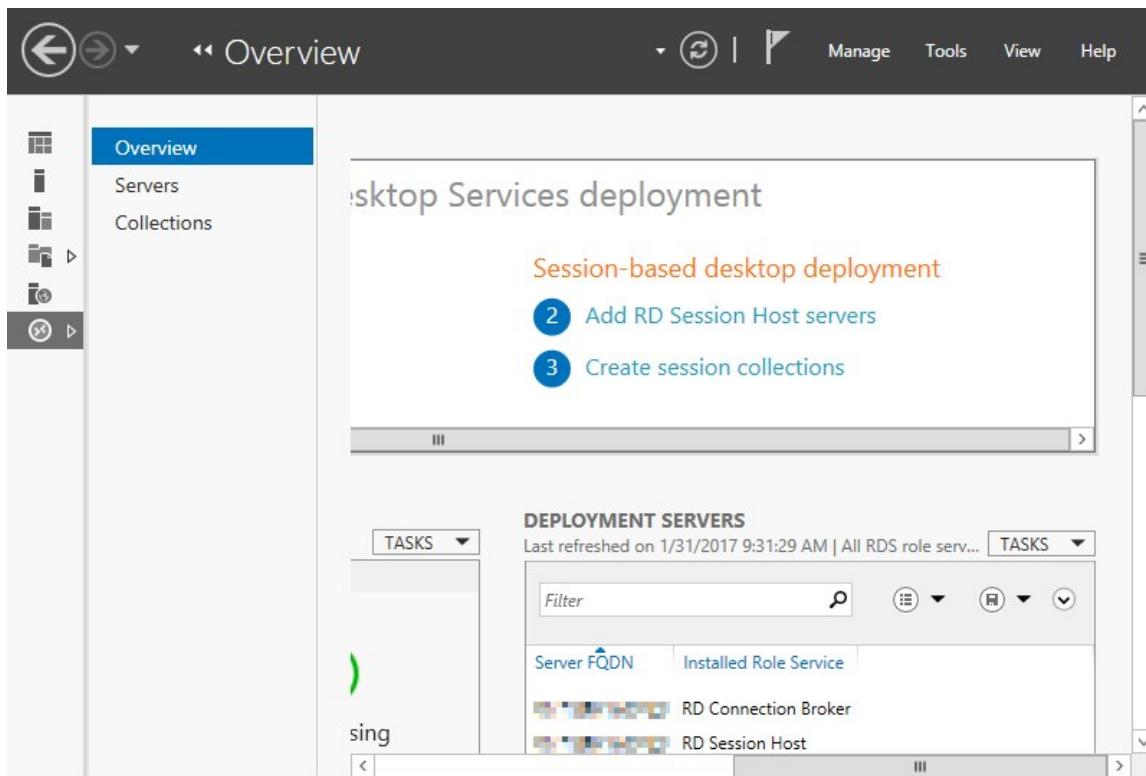
Note: SSPH (RDS) does not auto-update itself, unlike SSPH, because this could cause problems with multiple users running it at once on a single RDS server. Older SSPH (RDS) versions will continue to work with new Secret Server releases until updated, but a manual update is required on the RDS server(s) to take advantage of any future SSPH (RDS) features.

Step 3.2: Adding the Remote Desktop Collection and Application

1. While logged in as a domain user, go to Server Manager:
-



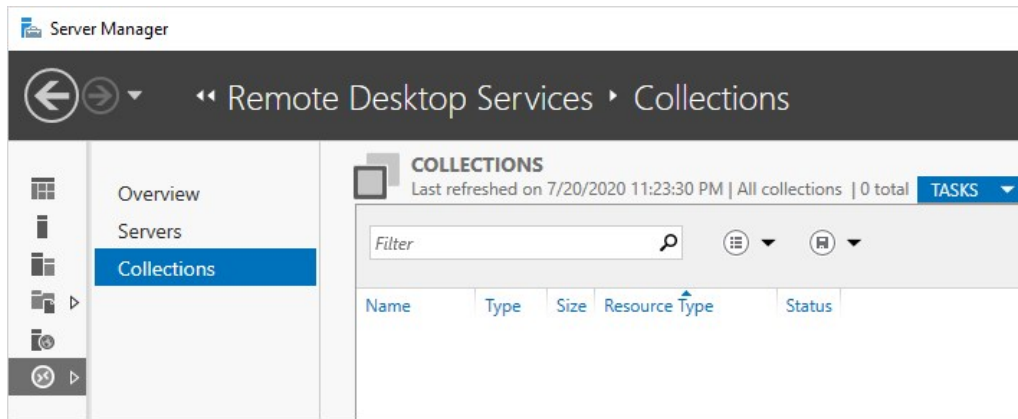
2. Click the **Remote Desktop Services** menu item on the left. The Overview page appears:



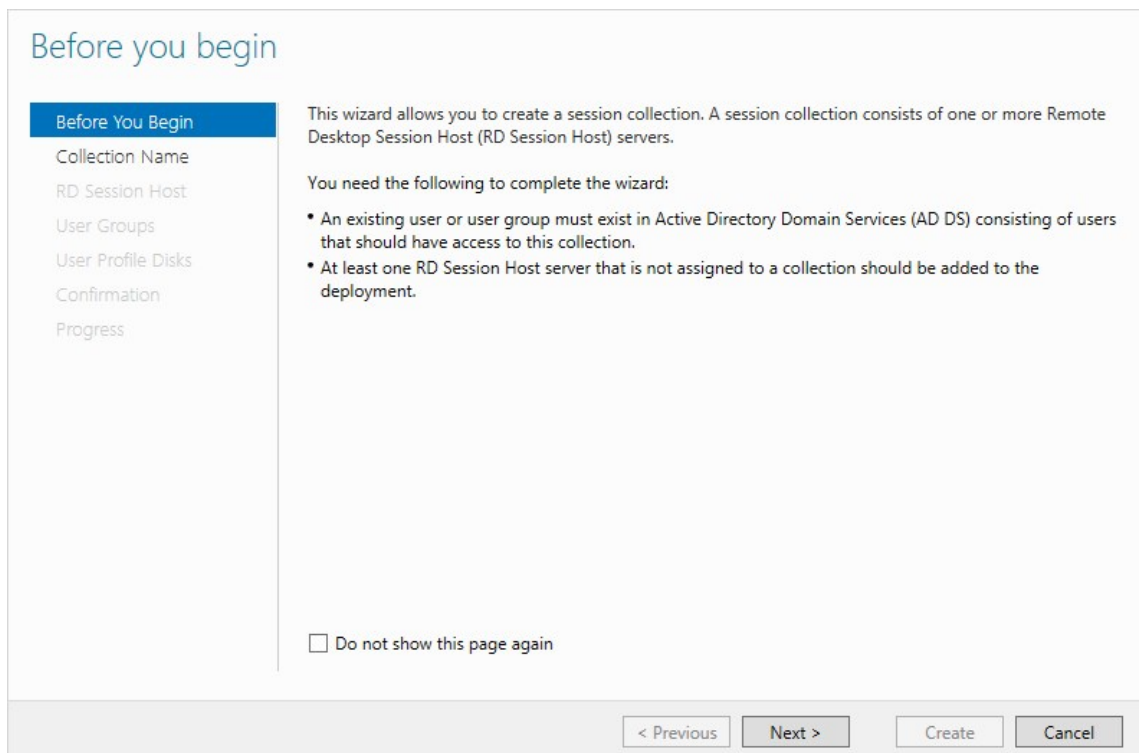
Note: If you logged on as a local user, you will see this error and be unable to configure RDS. You must be logged on as a domain user.

You are currently logged on as local administrator on the computer. You must be logged on as a domain user to manage servers and collections.

3. Click the Collections menu item. The Collections page appears:



4. Click the **Tasks** dropdown list and select **Create Session Collection**. The **Create Collection** wizard appears on the Before You Begin page:



5. Click the **Next >** button to arrive at the Collection Name page:

Name the collection

Before You Begin

- Collection Name
- RD Session Host
- User Groups
- User Profile Disks
- Confirmation
- Progress

A session collection name is displayed to users when they log on to a Remote Desktop Web Access server.

Name:

Description (optional):

< Previous Next > Create Cancel

6. Type Session Connector in the **Name** text box.
7. Click the **Next >** button. The Specify RD Session Host Servers page appears:

Specify RD Session Host servers

Before You Begin

- Collection Name
- RD Session Host
- User Groups
- User Profile Disks
- Confirmation
- Progress

Select the RD Session Host servers from the server pool to add to this collection.

Server Pool

Filter:

Name	IP Address	Operating S

< ||| >

1 Computer(s) found

Selected

Computer

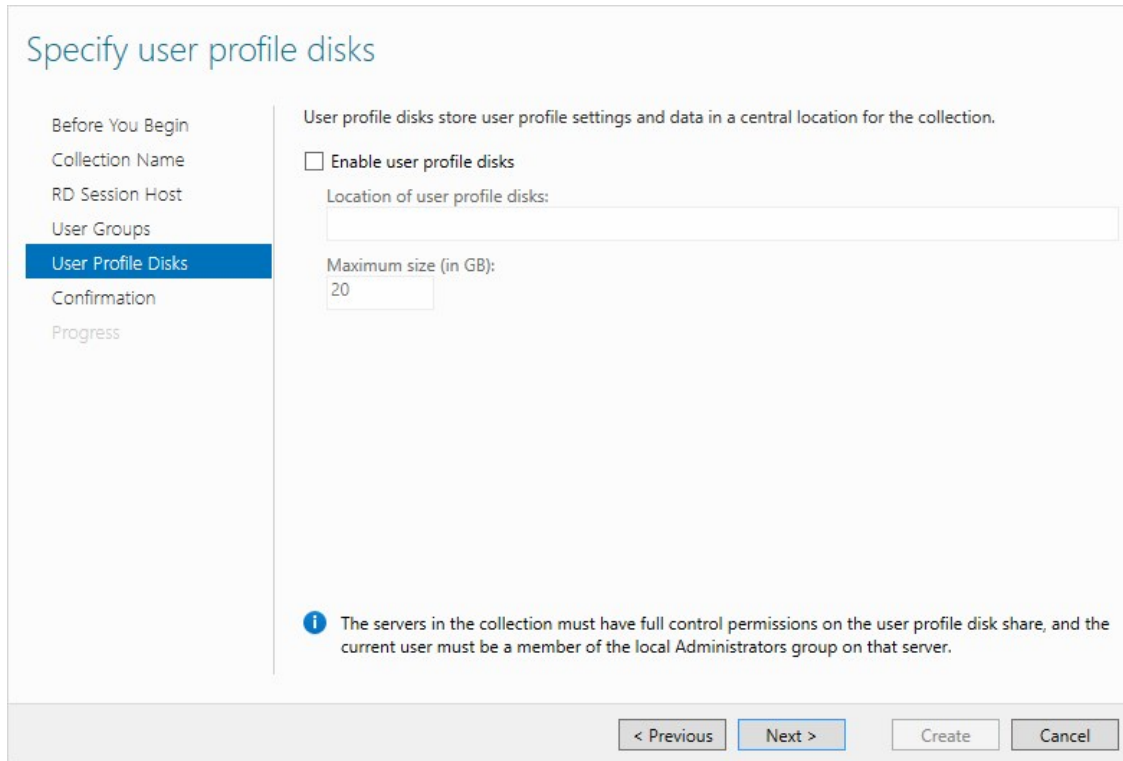
1 Computer(s) selected

< Previous Next > Create Cancel

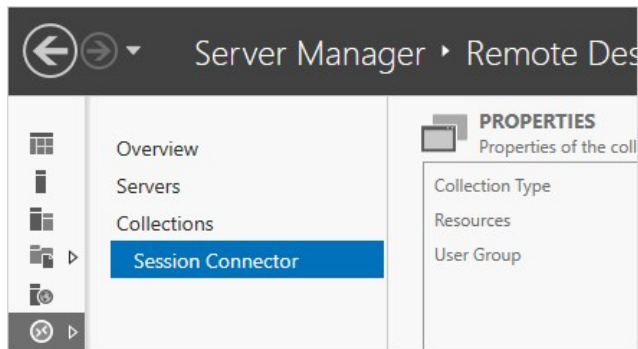
8. Add the local server from the left side (**Server Pool**) to the right side (**Selected**).
9. Click the **Next >** button. The User Groups page appears:

The screenshot shows a wizard window titled "Specify user groups". On the left is a navigation pane with the following items: "Before You Begin", "Collection Name", "RD Session Host", "User Groups" (highlighted in blue), "User Profile Disks", "Confirmation", and "Progress". The main area contains the instruction "Add the user groups that should have access to connect to the collection." Below this is a "User Groups:" label and an empty list box. To the right of the list box are two buttons: "Add..." and "Remove". At the bottom of the window are four buttons: "< Previous", "Next >" (highlighted in blue), "Create", and "Cancel".

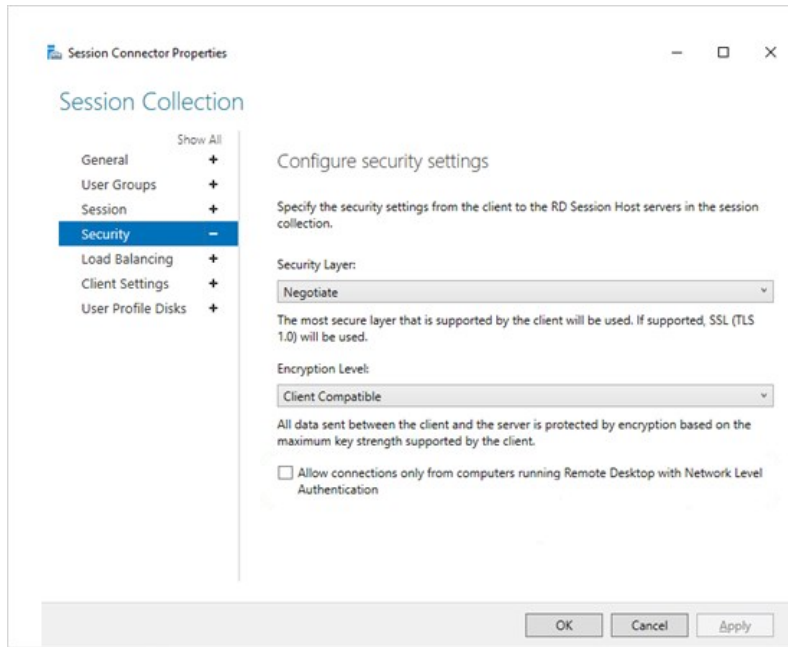
10. Select **Domain Users** in the **User Groups** list. This is not actually used by SSSC (it creates temporary local users), but RDS requires that something is selected.
11. Click the **Next >** button. The User Profiles page appears.



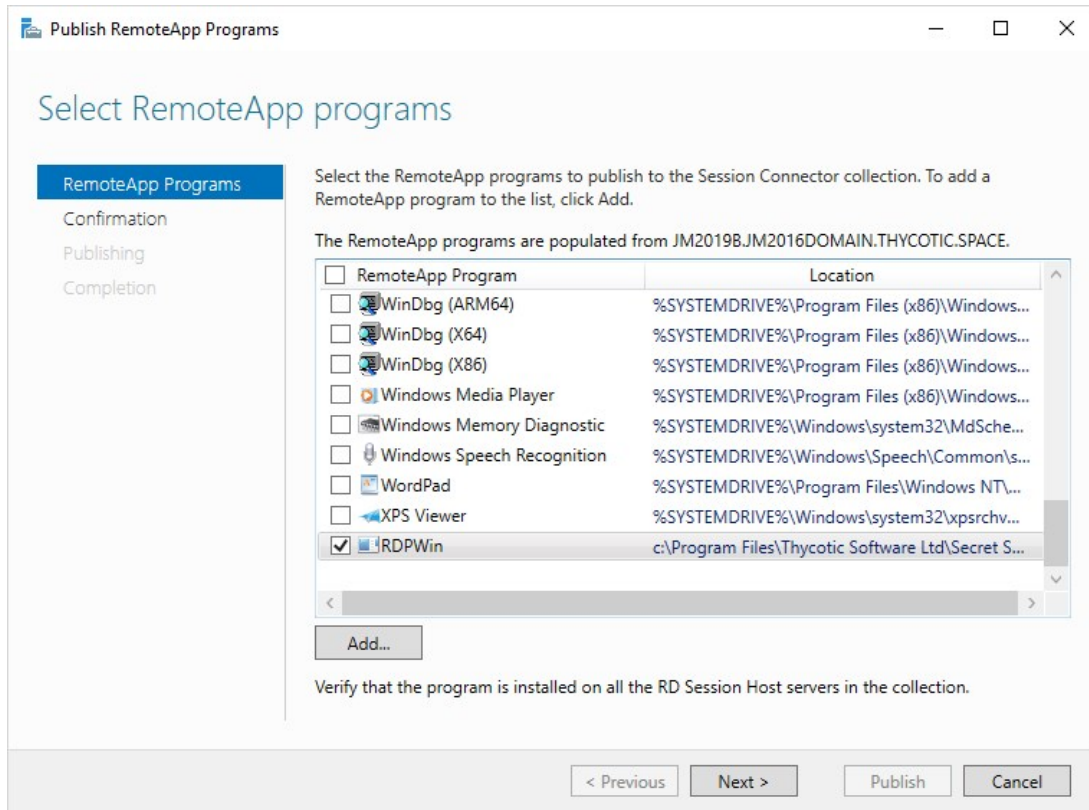
12. Click to select the **Enable user profile disks** check box. SSSC does not use user profile disks. We select the check box to enable the Create button.
13. Click the **Create** button. The collection is created, and the wizard disappears. The SSSC is now listed under Collections:



14. Click the **Tasks** dropdown list in the **Properties** section and select **Edit Properties**. The Properties popup appears.
15. In the left menu, click **Security**:



16. Click to deselect the **Allow connections only from computers...** check box. This necessary because SSSC uses temporary one-time use local users that do not exist until a connection is authenticated with Secret Server, making them incompatible with network-level authentication.
17. (Optional) If you want to restrict what can be mapped at the server level, such as drives, you can do so on the **Client Settings** tab. This is also configurable in Secret Server on each secret.
18. Click the **OK** button. The popup disappears.
19. Click the collection name in the menu on the left.
20. In the **RemoteApp Programs** section, click the **Tasks** dropdown and select **Publish RemoteApp Programs**.
21. Click the **Add...** button to add the RemoteApp for RDS protocol handler. A dialog box appears.
22. Navigate to C:\Program Files\Thycotic Software Ltd\Secret Server Protocol Handler.
23. Select RDPWin.exe.
24. Click the **Open** button. The dialog closes, and RDPWin (SSSH (RDP)) is now selected in the list:



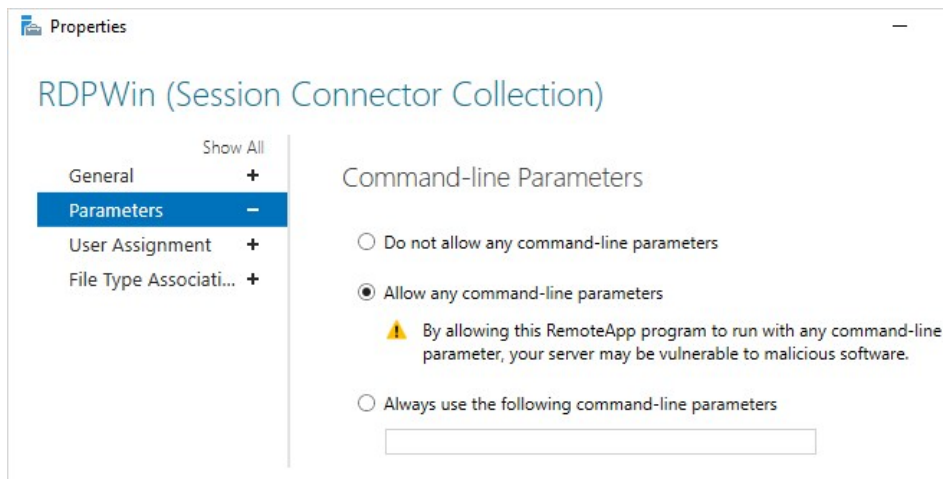
25. Click the **Next >** button. The Confirmation page appears.

26. Click the **Publish** button to save.

27. Click the **Close** button.

28. On the RemoteApp Programs page, right click **RDPWin RemoteApp** and select **Edit Properties**. A property page appears.

29. Click the Parameters menu item on the left:



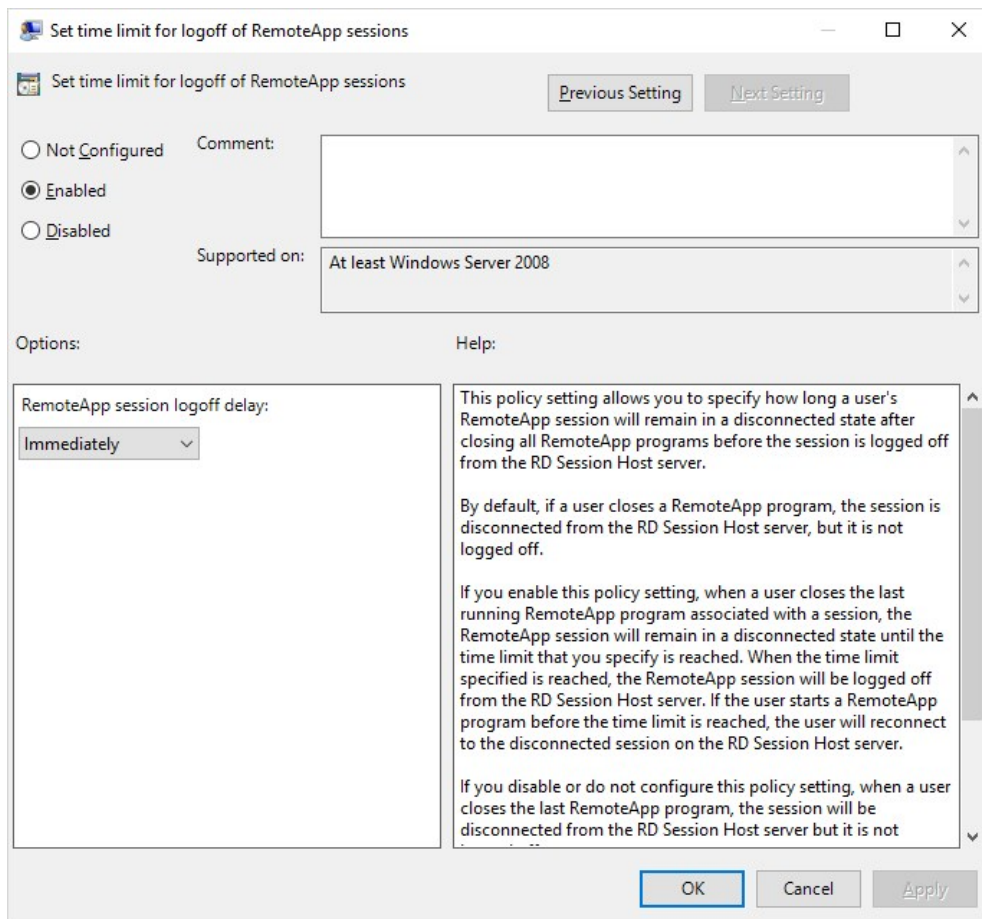
30. Click to select the **Allow any command-line parameters** selection button.

31. Click the **OK** button.

Step 3.3: Configuring RDS-related Group Policy Settings

To configure on a single server:

1. Run the Group Policy Editor (gpedit.msc).
2. Go to **Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits**.
3. Click **Set time limit for logoff of RemoteApp sessions** to edit it. Its properties appear:



4. Click the selection buttons to select **Enabled**.
5. Click the **RemoteApp session logoff delay** dropdown list and select **Immediately**.
6. Click the **OK** button to save.

Task 4: Installing the Secret Server Session Connector

1. Go to the [Session Connector Downloads](#) page.

2. Download the `SSSessionConnector.msi` SSSC installer file.
3. (Optional) Ensure the listed hash value matches that for the file.
4. Run the file.
5. When prompted, type the Secret Server URL and application account username and password you previously configured. The Secret Server URL must start with `https://` for security reasons, or the installer will not proceed.
6. When the installation is finished, you are prompted to reboot the server. This is to restart all of the remote desktop and terminal service services.
7. Once the server reboots, it will be a SSSC listening on the RDP port (TCP 3389). You can now use the SSSC custom launchers connected to it.

Task 5: Updating API Credentials

The credentials for the SA application account are saved encrypted in the registry. The credentials are restricted to the NETWORK SERVICE account, which Remote Desktop runs under using DPAPI-NG.

If those application account credentials change in the future, follow these steps to update them:

1. Run the Windows Registry Editor, `Regedit.exe`.
2. Navigate to `HKLM\SOFTWARE\Thycotic\SessionConnector`.
3. Set **CredentialsEncrypted** to 0.
4. Set **SecretServerUsername** to the plain text new username.
5. Set **SecretServerPassword** to the plain text new password.

These credentials are encrypted upon their first use, either the next time someone launches a SSSC session that hits this server, or if you reboot the entire server. Once this happens, returning to the Registry Editor, `CredentialsEncrypted` will be set back to "1," and an encrypted version of the username and password will be visible.

Task 6: Launching Session Connector Sessions

Now that it has been configured and installed, you should be able to launch SSSC sessions.

Once configured, the SSSC custom launchers appear just like any other launcher on the associated secret template types. When clicked, a Remote Desktop shortcut (.RDP) file is downloaded. This .RDP file can then be opened by standard Remote Desktop clients, such as `mstsc.exe` in Windows or RoyalITS in OSX.

When launched, the end-user will connect to the RDS host configured on the SSSC custom launcher. The RDS host then launches the RDS protocol handler and connects to the actual destination machine.

Subprocedures

Creating RDS Application Accounts

1. Go to **Admin > Users**.
2. Click the **Create New** button. The Edit User page appears:

Edit User

User Name

Display Name

Email Address

Domain

Password

Confirm

Two Factor

Enabled

Locked Out

[Advanced](#)

3. Type in or set the account details.
4. Ensure that the **Enabled** check box is selected.
5. Click the **Advanced** link. Additional parameters appear:

Edit User

User Name

Display Name

Email Address

Domain Local

Password Weak

Confirm

Two Factor

Enabled

Locked Out

Advanced

Application Account *As an application account, the user will only be able to log in through [Article](#)*

Managed By

Note: Leave the Two Factor and Managed By controls set to their defaults, < None > and User Administrators. Because this account is for SSSC and not a human being, 2FA is not appropriate.

6. Click to select the **Application Account** check box. As an application account, the user can only log on through the application account API and does not require a separate user license.

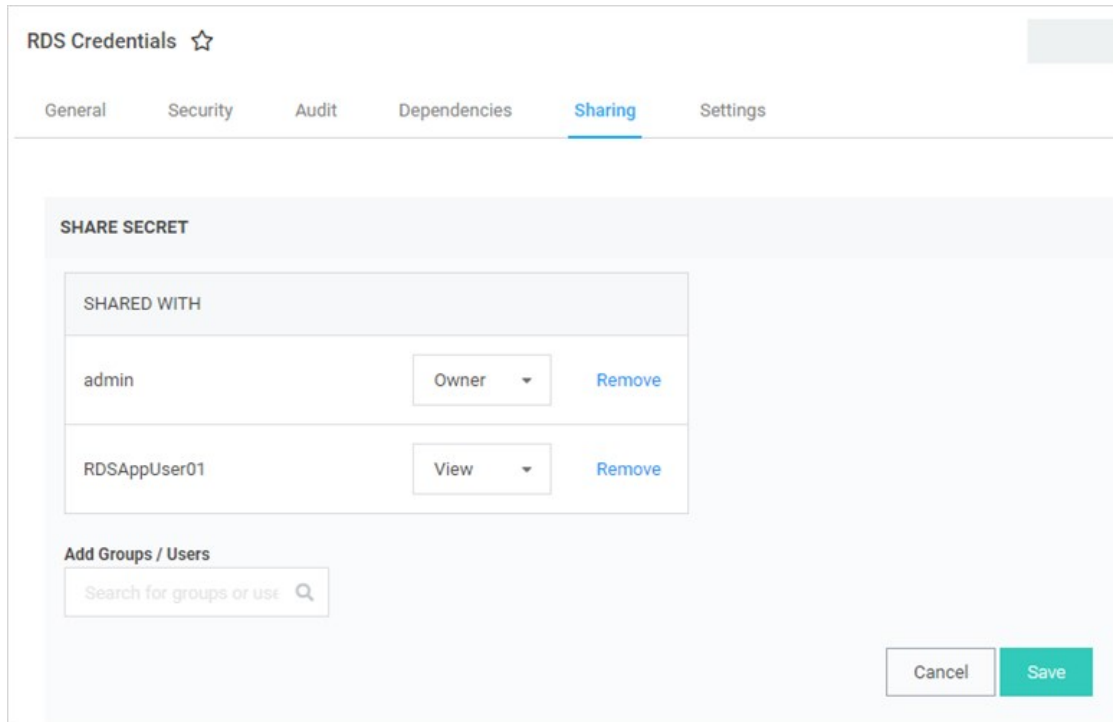
Note: We recommend application account users because only API access is required by SSSC, and they do not consume regular user licenses. You may want to name the users to make it obvious which server they belong to. We recommend one user per RDS server for auditing purposes and to avoid one server with invalid credentials locking out the user, impacting all the other servers. See [REST Web Services API Reference and Download](#) for more about the API.

7. Click the **Save** button.
8. Repeat this process for each RDS server if you are clustering more than one.

Enabling Application Account RDS Credential Sharing

Each RDS application account must have view access to the RDS Credential that the RDS server(s) use to manage the temporary Windows local accounts:

1. Go to the RDS credential secret you created earlier.
2. Click the **Sharing** tab:



3. Grant view access to the applicable application account users. That is usually one Secret Server user account per RDS server. If you are using a cluster, this secret would be an Active Directory secret for a domain credential that all the RDS servers can use, and you would share it with each of the RDS application accounts for each RDS Server in the cluster using in their SSSC configuration.

Configuring Session Connector Custom Launchers

You must create a custom launcher for each combination of and RDS server cluster and custom launcher type:

1. Go to **Admin > Secret Templates**.
2. Click the **Configure Launchers** button. The Launcher Types page appears.
3. Click the **New** button. The Launcher page appears:

Launcher

GENERAL SETTINGS

Launcher Type Session Connector Launcher ▾

Allows for downloading and running an RDP file to launch into a Remote Desktop Server with Protocol Handler installed, so end-user client machines do not need to install anything. Recommended only for advanced users. For more information see this [KB Article](#)

Launcher Name *

Active

Record Keystrokes

Child Launcher Type Remote Desktop ▾

RDS Server Hostname *

RDS Server Port *

RDS Server Credentials [RDS Credentials](#) [Clear](#) [Create New Secret](#) ↗

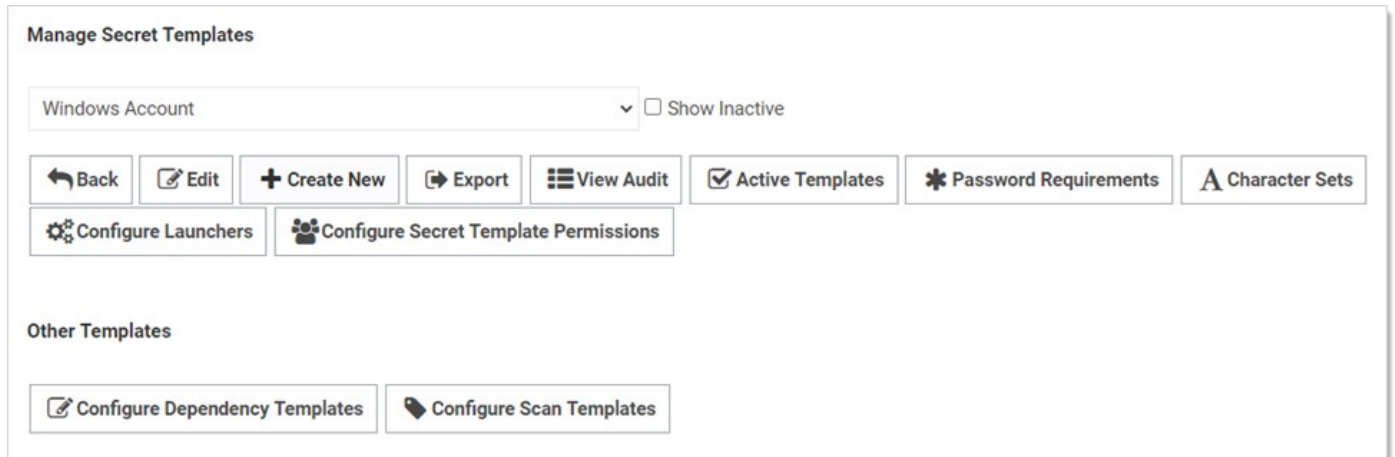
4. Type or set the parameters as follows:

- **Launcher Type:** Session Connector Launcher. This launcher type will not be visible until the Configuration Advanced Setting is enabled.
- **Active:** Ensure this is selected.
- **Record Keystrokes:** Check to record keystrokes in addition to video on related secrets with session monitoring enabled.
- **Child Launcher Type:** Click to select the launcher type, such as Remote Desktop or PuTTY. This is the real launcher type that runs on the RDS server to connect to the secret.
- **RDS Server Hostname:** IP or hostname for the RDS server or cluster.
- **RDS Server Port:** Type the port. The default RDP port is TCP 3389.
- **RDS Server Credentials:** Click the **RDS Credentials** link to pick the Secret configured above for credentials that can create and delete local users. If RDS Server Hostname points to a cluster, all servers must be able to use these credentials.

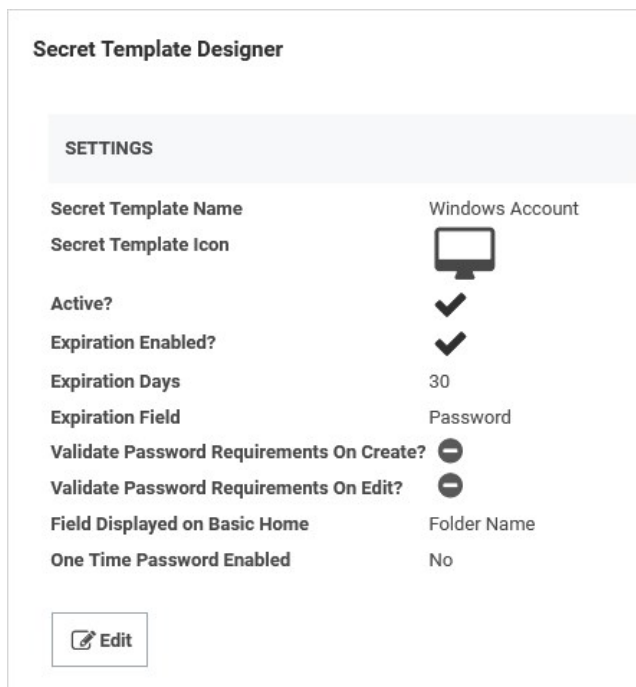
5. Click the **Save** button.

Assigning Session Connector Custom Launchers to Secret Templates

1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:




2. Click the unlabeled dropdown to select a secret template that you want to allow SSSC to launch from.
3. Click the **Edit** button to view that secret template. The Secret Template Designer page appears:



4. Click the **Configure Launchers** button at the bottom of the page to open the launcher mappings. The Secret Template Edit Launcher Configuration page appears:

Secret Template Edit Launcher Configuration



Launcher Type to use Remote Desktop

Computer Machine

Domain <blank>

Password Password

Username Username

5. Click the **Add New Launcher** button. A page of the same name appears:

Secret Template Edit Launcher Configuration

Launcher Type to use Remote Desktop

Computer <user input> *

Domain <blank> *

Password Machine *

Username Machine *

ADVANCED SETTINGS

Restrict User Input

6. Click to select the desired parameters for the launcher:

Secret Template Edit Launcher Configuration

Launcher Type to use: RDS RDP

Computer: Machine *

Domain: <blank> *

Password: Password *

Username: Username *

ADVANCED SETTINGS

Restrict User Input ?

Save Cancel

7. All secrets using this template are now ready to run SSSC launches.

When launching a downloaded .RDP file, if SSSC rejects the session due to any issues (including being expired based on the "Session Connector Session Timeout" setting), the user's Remote Desktop client will receive a generic error about the RemoteApp being invalid.

In the ss.log file, you can search for "SessionConnector" to find details about why sessions may have been rejected.

Session Connector will also log to the file C:\Program Files\Thycotic Software Ltd\Secret Server Session Connector\log\SS-SC.log on each RDS server. That is, if the RDS server has trouble using the supplied RDS credential to create a local user, it is logged to this file

Secret Server Session Connector can be removed from "Add/Remove Programs" or "Apps & Features." Once uninstalled, a reboot is required to restore the default Remote Desktop behavior.

Any related SSSC custom launchers need to be un-associated with any secret templates they were previously tied to.

Note: It is not currently possible to delete a custom launcher in Secret Server, but if it is unassociated with all secret templates, it will not appear on any secrets.

Note: The install files (.msi) are zipped to make them more download friendly All hashes listed below are run on the install files, not the zips of those files.

Session Connector

File:

[SSSessionConnector.msi](#) (v1.0.0.0)

SHA1 Hash:

C9A4B274455159FFB792AB6C497B752399921EC8

SHA256 Hash:

3ED64AB39FDAE0E0DD99513C395FE7690BFDAD0D6CA26DD77844AFFEDA7ADDF0

SSPH (RDS)

Note: SSPH (RDS) is a special Remote Desktop Services version of Secret Server Protocol Handler that can record keystrokes on its own, if configured in Secret Server. See [Session Connector](#) for details.

Note: .NET Framework 4.8 is required for SSPH (RDS).

File: [SSProtocolHandlerRDS.msi](#) (v6.0.3.11)

SHA1 Hash:

E32C60C91286354E35653871250E8618667243BF

SHA256 Hash:

A072C24C36C322BD67825DFAC61D5E59320FFA44127A674E413E7B24D9A05E54

Overview

Connect As Command is an advanced setting for the PuTTY launcher type where SSH proxy automatically runs the su command for a Unix root account secret after the user launches a PuTTY session. This provides a user elevated privileges without allowing a remote root connection or giving the user direct access to the credentials.

The connection procedure is as follows:

1. An admin uses this instruction to set up secret A (a Unix root account secret) to use secret B (a regular Unix account secret) as its "connect as" secret.
2. A user launches secret A.
3. SSH proxy connects using secret B's credentials.
4. SSH proxy issues the su command to switch the user back to secret A.

The procedure is performed once at the beginning of the session.

As noted, to implement this feature, you typically use a Unix *root* account secret and a Unix *regular* account secret. The session usually launches as the Unix *regular* account secret that is specified in the **Secret To Use** field on a Unix *root* account secret's **Settings** page.

Setting up SSH Proxy to Use the Connect As Feature

This procedure explains how to set a "connect as" secret when using SSH Proxy to allow connecting with a less privileged account and then using sudo or su to elevate privileges.

1. Make sure SSH proxy is enabled in Secret Server's global configuration settings.
2. Open a secret based on a template with SSH proxy enabled that specifies PuTTY as the launcher type to use.

Note: For this feature, we recommend building a custom secret based on a copy of the built-in **Unix Root Account (SSH)** template, and associating the PuTTY launcher with it.

3. Click the secret's **Settings** tab.
4. Next to **Connect Using**, select **Credentials on another Secret**.
5. Next to **Secret to Use**, click **No Selected Secret**.

The screenshot shows the 'Settings' page with the following sections:

- EMAIL NOTIFICATIONS** (with an 'Edit' link):
 - Send Email When Viewed: No
 - Send Email When Changed: No
 - Send Email When Heartbeat Fails: No
- SSH LAUNCHER**:
 - Connect Using: Credentials on another Secret (dropdown menu)
 - Secret To Use: No Secret Selected

Buttons: Cancel, Save

6. Navigate to **Admin > Secret Templates**.
7. Select a template from the drop-down and click **Edit**.
8. Scroll down the page and click **Configure Launcher**.
9. Locate the PuTTY launcher type and click **Edit**.
10. Verify that the commands in the **Connect As Command** field are correct.
11. If you change anything, click **Save**.

The screenshot shows the configuration fields for the Connect As Command:

- Connect As Command**: su - \$USERNAME
- Connect As Command Response**: Password:\$PASSWORD
- Line Ending**: New Line (\n)

Buttons: Save, Cancel

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Web launchers are a separate login method from the Web password filler and provide a convenient click to automatically log on simpler websites. Web launchers do not work on complex login pages that rely on JavaScript. For those login pages, use the browser extension for the Web password filler. By default, Web launchers are enabled on the Web Password Secret template, but they can be enabled on custom templates as well, as described in [Enabling Launchers](#).

Configuring Web Launchers for Secrets

Once enabled on the template, a Web launcher needs to be configured for the secret. Each website login is unique and requires the secret text-entry fields to be mapped to the form controls. For a new secret the Launcher icon appears and clicking on it takes the user to a configuration screen. The user can also view and access the configuration screen from the Launcher tab. Depending on whether other secrets with the same website have been configured, the user has different options.

Note: Configuring the Secret for use with the Web Launcher requires the user to have Owner permission on the Secret.

First, there is the option of downloading the setting from Delinea.com. When the Configure Web Launcher page is loaded, Secret Server checks online at Delinea.com for pre-approved matching websites. If any are found, they are downloaded and made available to pick from in the dropdown list.

Note: This functionality can be disabled in Secret Server in the Configuration Settings.

The list displays all downloaded configurations and other secrets' configuration for the same domain that the user has permission to view. Select one from the list and click **Next** to create a copy of the settings for the secret.

There is also an option to create a configuration that allows the Web launcher to be used on most websites and not rely on published configuration settings. To use this, select the last item in the dropdown list and click **Next**. The next section discusses the create process.

Creating a Configuration

When configuring the Web Launcher:

- **Entering the Login URL:** Secret Server needs to know the exact URL used to login to be able to figure out the controls and perform the automatic login. Some example login URLs:
 - <https://login.yahoo.com/config/login>
 - <https://MyServer/Billing/login.aspx>
 - <https://firewall07/login/>

Note: The Login URL is typically a secure site with a prefix of `https://`. If allowed to access the site, Secret Server automatically detects if `https` should be used to ensure the credentials are passed securely.

- **Providing the Page Source:** If Secret Server is not allowed access to sites, or the login URL is not accessible by an external site, the page source needs to be provided for the Web launcher controls to be obtained. Ensure the login URL is correct when the page source is taken. If the site can be accessed by Secret Server the page source is automatically obtained and this step is not present.
- **Choosing the Form:** The page is read, and the exact login form needs to be identified. The page forms are listed in the list with the most likely selected. If no forms or no likely forms are found, the user needs to update the URL or page source, as configuration must have at least one textbox and one password box.
- **Wiring Up the Fields to Controls:** In most cases, Secret Server automatically wires up the Username and Password text fields to the correct page controls. If not, the user completes the control mapping on the Launcher tab.

Launching to a Website

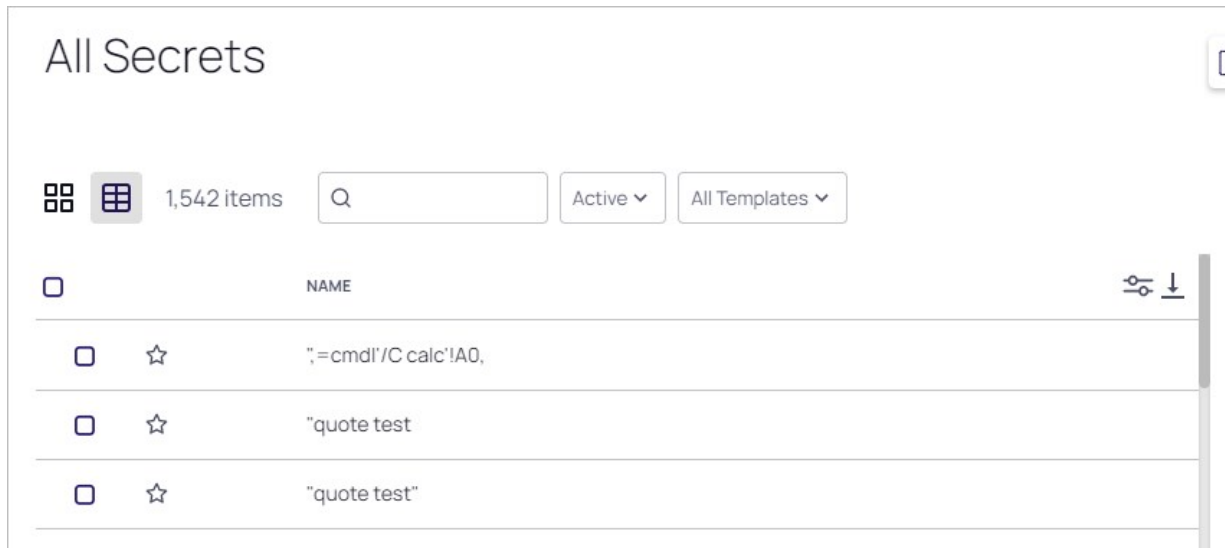
The Web launcher can be used by clicking the Launcher icon on the Secret View page. The Web launcher opens a new window in the browser, which attempts to login to the site using the credentials on the secret. The launcher can also be used with the Test Launcher button on the Launcher tab. Testing the Launcher creates a dialog to offer troubleshooting help and means to upload the configuration to Delinea.com. The uploaded configuration is reviewed and published by Delinea for all Secret Server customers to use with the check online feature. No secret or identifiable information is uploaded to Delinea.com. Only the website URL and control names are sent.

Secret Management

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.


Secrets are individually named sets of sensitive information. Secrets address a broad spectrum of secure data, each type represented and created by a *secret template*. You can centrally manage secret security through sharing settings for each secret. Additionally, using folder structure, you can allow one or more secrets to inherit permissions from their parent folder. All secret text-entry field information is securely encrypted before being stored in the database, including a detailed audit trail for access and history.

All Secrets is a master table of the secrets stored on Secret Server. It is a one-stop, searchable location for examining the status and properties of secrets. It is a supplement to, not a replacement for, the [secret folder-tree panel](#). It lists and you can sort by secret template, heartbeat status, sync status, machine, access date, username, and much more.



Click the **Secrets** menu item in the left menu to see the All Secrets Page. Click the > > icon to see the All Secrets folder tree.

Secret Columns

You can customize which columns are displayed by clicking the  on the right side of the title bar. The sortable columns available are (the ones displayed by default are bolded):

- Auto Change Enabled
- Checked Out
- Checkout Enabled
- Created
- Days until Expiration
- Deleted
- DoubleLock Enabled
- **Folder**
- **Heartbeat**
- Hide Password
- Inherits Permissions
- **Last Accessed**
- Machine
- **Name**
- Notes
- **Out of Sync**
- Requires Approval
- Requires Comment
- **Secret Template**
- Username

Quick Launch and Actions

You can quickly access several secret actions from the All Secrets table by hovering the mouse pointer over the secret's row:

- Quick Launch: Secrets that display a rocket icon have secret launchers associated with them. Click the icon and a Select Launcher page appears. Click on the desired launcher from the list.
- More Actions: Click the ... icon to
 - Audit: Brings you to the Audit tab of the secret's page.
 - Details: Brings you to the General tab of the secret's page.
 - Enter Comment: Brings you to a page for entering a comment before you are allowed to view the secret. Comment ask for information such as reason for viewing and ticketing system.
 - Share: Brings you to the Sharing tab of the secret's page.

Other Features

The All Secrets page also allows you to:

- Select multiple secrets for [bulk actions](#) by clicking the secret's check box. You can also download multiple secrets in a CSV file by clicking the download icon at the top of the table.
- Select the secret as a favorite by clicking the star icon.
- View supplemental information about the secret by clicking any blank area on the secret's row.
- Click the secret's name to go to that secret's page.
- Filter the secrets shown in the table by template, search text, or activity status.
- Click the [Secret Navigation Slideout](#).

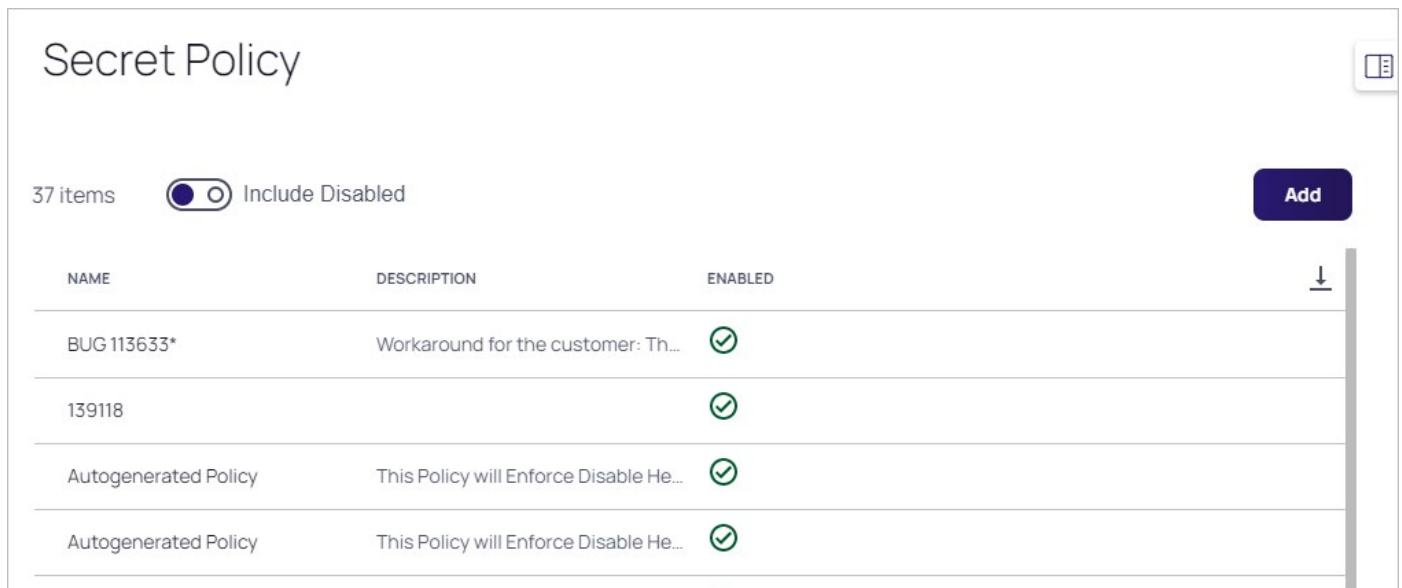
Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Creating Secret Policies

A secret policy is a set of rules that you can apply all at once to multiple secrets. For example, a secret policy could include rules about remote password changing or security settings, and you could apply all of the rules as a single policy to multiple secrets, whether the secrets reside in the same folder or different folders.

Follow the procedure below to create a secret policy:

1. Click **Administration > Secret Policies**. The Secret Policy page appears:

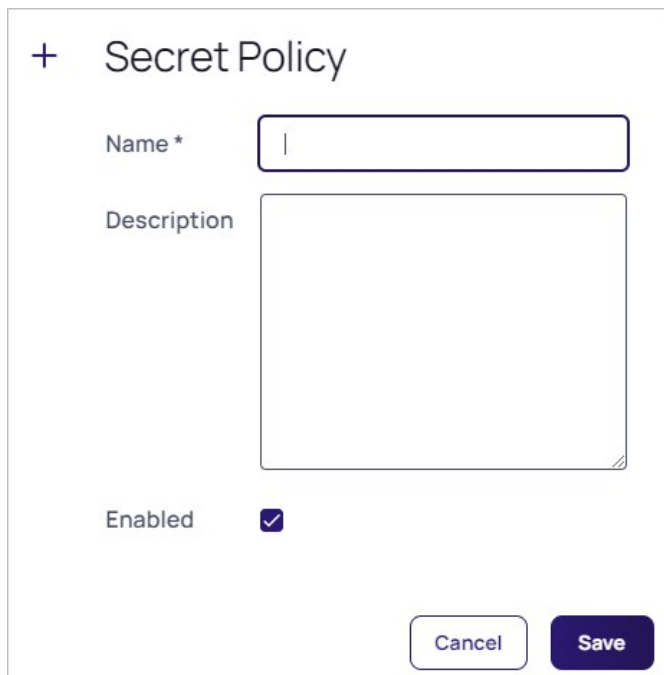


Secret Policy

37 items Include Disabled Add

NAME	DESCRIPTION	ENABLED
BUG 113633*	Workaround for the customer: Th...	✓
139118		✓
Autogenerated Policy	This Policy will Enforce Disable He...	✓
Autogenerated Policy	This Policy will Enforce Disable He...	✓

2. In the **Secret Policy** window, click the **Add** button. A popup appears:



+ Secret Policy

Name *

Description

Enabled

Cancel Save

3. Type a **Secret Policy Name** and **Description** for your new security policy.
4. Click to select the **Enabled** check box.
5. Click the **Save** button. The new policy's page appears on the Policy tab and Summary subtab:

My Secret Policy

Policy
Audit

Summary
General
Security
Remote Password Changing
Launcher

Secret Policy Summary

The name of the Secret Policy will show for users when assigning a policy to a Secret or folder. Disabling a Secret Policy will not change values on any Secrets. Default policy values will only be assign when creating a new Secret. Removing a Secret Policy from a Secret will not revert values on the Secret.

Edit

Name *	My Secret Policy
Description	Mine, all mine.
Enabled	Yes

Because you are creating a brand-new secret policy, the value in the setting column for many policy items is *(Not Set)*.

6. Click each subtab in turn and click the **Edit** button to set which parameters are enforced. See the various topics for specifics about each parameter.

Creating Secrets

To create a secret:

1. Click the **+** icon and select **New Secret**. The Create New Secret page appears:

+ Create New Secret

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Personal Folders/Will Sprunk Change Clear

Choose a Secret Template

Search for template name

- [GTS] Unix Account - SUDO SU (SSH)
- 1-Bug AD Copy
- A Small Template
- Active Directory Account
- Active Directory Account (Custom Launchers)
- Active Directory Account RDP Launcher (No Prompt, No Restrictions)
- Active Directory Account RDP Launcher (Prompt, Black List)
- Active Directory Account RDP Launcher (Prompt, Black List, Exclude Dependency)
- Active Directory Account RDP Launcher (Prompt, No Restrictions)
- Active Directory Account RDP Launcher (Prompt, White List)
- Active Directory Account RDP Launcher (Prompt, White List, Exclude Dependency)
- Active Directory with Allow List
- Active Directory with Lists
- AD BugCopy
- AD Template CC

Cancel Create Secret

2. Click the **Choose a Secret Template** list or type a template name in the **Search for template name box** to choose a template from which to create the secret .



Note: If you do not find a suitable template available, you can create a custom template.

3. Click the **Create Secret** button. A Create New Secret page appears.

Note: These pages differ significantly, based on the secret template you chose. For this instruction, we chose the frequently used Web Password template.

+ Create New Secret

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Secret Template	Web Password Change
Folder	Personal Folders/Will Sprunk X
Secret Name *	<input type="text"/>
URL *	<input type="text"/>
Username *	<input type="text"/>
Password *	<input type="password"/>  Generate
Notes	<div style="border: 1px solid #ccc; height: 150px; width: 100%;"></div>
Site	<input type="text" value="Local"/> 
Auto Change Enabled	<input type="checkbox"/>

4. Complete the text boxes and selection controls on the page.

Note: The password generator is governed by a password requirement, which is usually set via the secret template. However, you can override the template for this secret and set the requirement to something different in the Password Requirements section of the Security tab, *after* you create the secret.

5. Click the **Generate** button to create a strong password that meets the requirements for that type of secret. You can also add your own.

Note: Click the eyeball icon to see a list of the current requirements and what you have already typed. The maximum

password length is 1024.

The bar below the text box indicates the strength of the password you enter—it stays red but gets longer as the password quality

My Web Secret

https://example.com

Bubba

mybadpassword

Generate

Password should include:

- At least 12 characters.
- At least 1 Lower case letters (a-z)

At least 1 Symbols (Symbols)

At least 1 Numbers (0-9)

At least 1 Upper case letters (A-Z)

Password should exclude:

- Username

improves.

As you meet each requirement, a green checkmark appears next to it.

Password should include:

- At least 12 characters.
- At least 1 Lower case letters (a-z)

Once you have met all the requirements, the color bar is full length and green.


6. Click the **Sites** dropdown list to select a site the secret belongs to.
7. (Optional) Click to select the **Auto Change Enabled** check box to enable automatic remote password changing (RPC) for the secret.
8. Click the **Create Secret** button. The new secret's page appears.
9. Click any of the tabs to further configure the secret.

Note: It is possible to import data as secrets. See [Importing Secrets](#).

Customizing the All-Secrets Page

On the main menu, there is a **Secrets** folder tree. When you click on the root or any subfolder, you see a list of all the secrets in that folder with multiple columns. You can customize what you see in one of three ways:

Customizing Visible Columns

You can display additional columns on the grid by clicking the  icon. This data can be either secret metadata or template text-entry fields that have been set to be available for viewing. To select additional columns to display, click the **Advanced** link and then the **Column Selection** link. You can display the following metadata fields:

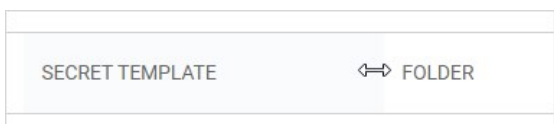
- Auto Change Enabled
- Checked out
- Checkout Enabled
- Created
- Days until Expiration
- Deleted
- Double Lock Enabled
- Expiration Field Changed
- Folder
- Inherits Permissions
- Heartbeat
- Hide Password
- Last Accessed
- Machine
- Notes
- Requires Approval
- Requires Comment
- Secret Template
- Username

Filtering Search Results

You can filter secret search results by selecting a folder on the left, either by clicking it or using the search text-entry field above the folder tree. On the right side of the widget, secrets can be filtered further by specifying search criteria in the top text box. The Advanced section allows filtering by secret template and status, as well as the option to include secrets contained in subfolders. Advanced criteria only remain in effect while those options are expanded (visible).

Sizing Columns

You can resize any of the columns by hovering the cursor over the border between them till it turns into a double arrow:



Click and drag to resize the column.

Deactivating and Reactivating Secrets

Note: Deactivating (previously called "deleting") a secret is *not* the same as erasing one—the former hides it but it can still be viewed or undeleted by administrators—the latter is a permanent removal of data and requires more effort, including an access request. Deleting secrets is common. Erasing them is rare, only needed in special circumstances. See [Erasing Secrets](#) for details.

Important: We strongly recommend against deactivating or erasing large numbers of secrets, both of which negatively affect performance over time. Secret Server is not a transactional system—it is not designed to handle large numbers of deactivations or erasures. Deactivated or erased secrets continue to use database table resources forever. Even erased secrets leave a database record even though the secret data is permanently deleted. Erasure is for regulatory compliance and not database clean up.

To deactivate a secret:

1. Navigate to the secret **View** page by searching or drilling down the folder tree.
2. Click the **Options** dropdown list and select **Deactivate**. A confirmation appears.
3. Click the **Confirm Deactivate** button.
4. The secret is logically deleted and hidden from users who do not have a role containing the View Deactivated Secrets permission.

Note: Secret Server uses deactivations to maintain the audit history for all data. However, deactivated secrets are still accessible by administrators (like a permanent Recycle Bin) to ensure that audit history is maintained and to support recovery. A user must have the View Deactivated Secrets permission in addition to Owner permission on a secret to access the secret View page for a deactivated secret. For more information about these permissions, see [Roles](#) and [Sharing a Secret](#).

To reactivate a secret:

1. Navigate to the secret view page.
2. Click the **Active** menu link and select **Inactive**. The secret list now shows inactive secrets.
3. Click the name link for the desired secret. Its secret page appears.
4. Click the **Options** button and select **Activate**.

Note: Secrets can also be deactivated and reactivated in bulk. See [Running Dashboard Bulk Operations](#).

Duplicating Secrets

The secret duplication function allows for easier, automatic secret duplication. Any user with the Owner Secret permission on a secret can click to select **Duplicate** in the **Options** dropdown list to create a new secret with information based on the original secret. Secret text-entry field information, launcher settings, secret settings, double locks, email settings, and permissions are copied over. Audit records are written to the source secret and target secret to indicate that a copy operation took place. Currently, file attachments are not copied.

Editing Secrets

Note: If using the Dashboard, see [Secret Server Dashboard](#).

To edit a secret:

1. Navigate to the secret's **View** page by searching or drilling down the folder tree.
2. Click the desired tab for the secret configuration.
3. Click the **Edit All Fields** link. All text-entry fields become editable.

Note: The password generator is governed by a password requirement, which is usually set via the secret template. However, you can override the template for this secret and set the requirement to something different in the Password Requirements section of the Security tab after you create the secret.

4. For passwords, you can create a random password with the **Generate** button (on the General tab). This generates a password according to the rules set at the template level (see secret templates for more information about password requirements).
5. Click the **Save** button.

Erasing Secrets

Note: Deactivating (previously called "deleting") a secret is *not* the same as erasing one—the former hides it but it can still be viewed or reactivated by administrators—the latter is a permanent removal of data and requires more effort, including a secret erase request. Deactivating secrets is common. Erasing them is rare, only needed in special circumstances. See [Deactivating and Reactivating Secrets](#) for related information.

Important: We strongly recommend against deactivating or erasing large numbers of secrets, both of which negatively affect performance over time. Secret Server is not a transactional system—it is not designed to handle large numbers of deactivations or erasures. Deactivated or erased secrets continue to use database table resources forever. Even erased secrets leave a database record even though the secret data is permanently deleted. Erasure is for regulatory compliance and not database clean up.

Note: This instructions assumes you know the basics of access requests, groups, roles, and permissions. We also suggest reading the introductory material for [Workflows](#) if you are not familiar with it.

Task 1: Configuring Secret Erase

Note: If secret erasure is already configured on this server and you are in the Secret Erasers group, you can skip to Task 2.

1. Ensure that you have a workflow license for Secret Server.
2. Go to **Admin > Roles** in Secret Server:

ROLE NAME	ENABLED	CREATED
Administrator	Yes	5/15/2019 05:39 AM
APITesting	Yes	12/10/2019 11:12 AM
Basic User	Yes	5/15/2019 05:39 AM

3. Create a new role named "Secret Erase Requester" or "Secret Erase Administrator" (see [Creating Roles](#) for details), assigning it the "Erase Secret" permission:

Note: You can name the role anything you desire, but we recommend the above for clarity.

Role Edit

Role Name *

Enabled

Created

Permissions Assigned

Erase Secret

Permissions Unassigned

- Access Offline Secrets on Mobile
- Add Secret
- Add Secret Custom Audit
- Administer Active Directory
- Administer Automatic Export
- Administer Backup
- Administer Configuration
- Administer Configuration Proxying
- Administer Configuration SAML
- Administer Configuration Security
- Administer Configuration Session Recording
- Administer Configuration Two Factor
- Administer Configuration Unlimited Admin
- Administer ConnectWise Integration
- Administer Create Application Accounts

Save
 Cancel

The "Erase Secret" role permission allows users with the role to create secret erase requests and view secret erase administration pages.

4. Go to **Admin > Groups**. The Groups tab of the User Management page appears:

Admin > User Management 🔍 🏠 + WS

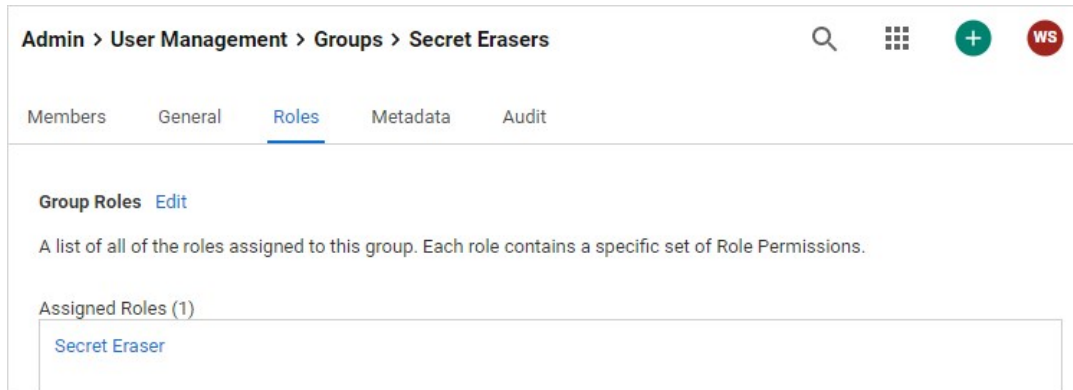
Groups Users Audit

Manage Directory Groups Create Group

78 Items All Domains ▾ 🔍 Include Disabled

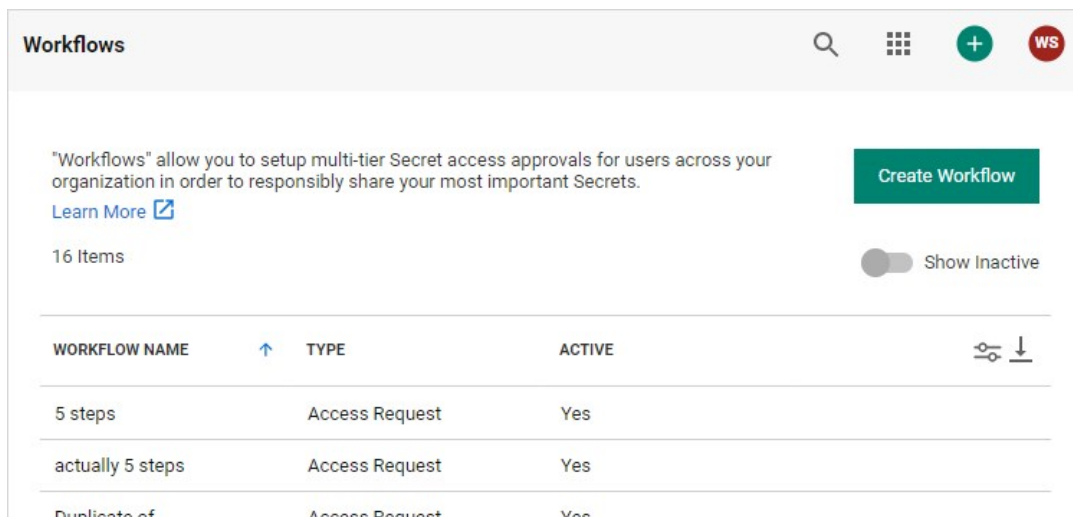
GROUP NAME	↑	ENABLED	MEMBER COUNT	CREATED	⚙️ ↓
(lol)		Yes	25	13 days, 1 hour ago	
Access Control As...		Yes	0	1 year, 11 months ...	
Account Operators		Yes	0	1 year, 11 months	

5. Create a group named "Secret Erasers" and give it the "Secret Eraser" role:



6. Click the **Members** tab to add yourself to the **Secret Erasers** group.

7. Go to **Admin > Workflows**:



8. Create a "Secret Erase Requests" workflow template, assigning it the Secret Erase Request type. The Designer tab for the new workflow appears:

Workflows > Secret Erase Requests

Designer Audit Duplicate Deactivate

Workflow Designer

Workflow Type Secret Erase

Name * Secret Erase Requests

Description

Active

- Assign one or more users or groups as approvers by typing each in the search text box in the **Add Groups / Users** section and then clicking your choice when it appears. It then appears in the Approvers list box.

Note: We chose to have the approvers be the same group as those that can make the requests, but you can choose any groups or users you like or make a group just for approvals. The important thing is the same *user* cannot both make the request and approve it—that way, a single person cannot make an irreversible, potentially very harmful, mistake.

- Click the **Save** button. The result looks like this:

Workflows > Secret Erase Requests

[Designer](#) [Audit](#)

Workflow Designer [Edit](#)

Workflow Type	Secret Erase
Name *	Secret Erase Request
Description	None
Active	Yes

Step 1

Name	Step 1
-------------	--------

Approvers

[Secret Erasers](#)

Number of approvers required	1
If approved	Approve The Request

11. Go to **Admin > Configuration:**

The screenshot shows the 'Configuration' page with the 'Security' tab selected. The 'APPLICATION SETTINGS' section is visible, containing the following items:

Setting	Value
Allow Automatic Checks for Software Updates	Yes
Early Adopter	No
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	No View Metric Data

12. Click the **Security** tab.
13. Click the **Edit** button at the bottom of the page. The page becomes editable.
14. Click to select the **Enable Secret Erase** check box in the **Secret Erase** section. The Secret erase Workflow dropdown list appears:

The screenshot shows the 'SECRET ERASE' configuration section. It includes the following elements:

- Enable Secret Erase**: A checked checkbox.
- Secret Erase Workflow**: A dropdown menu currently set to '< None >'.
- Save** and **Cancel** buttons at the bottom.

15. Click the **Secret Erase Workflow** dropdown list and select **Secret Erase Request**.
16. Click the **Save** button. Secret Erase is now set up.

Task 2: Erasing a Secret

1. Ensure the following requirements are met for the secret you intend to erase—ensure the secret:
 - Is inactive
 - Is owned by you
 - Does not have a pending secret erase request
 - Is not double-locked
 - Is not checked out by another user
 - Is not a discovery secret
 - Is not a domain sync secret

2. For purposes of this instruction, create a secret for testing in your personal folder. For now, do not use an existing one to ensure all the requirements are met.
3. You can erase the secret via a dashboard bulk operation or from the **Options** button on the **General** tab of the secret itself. For a bulk operation, erase is accessed by the **More Bulk Options** button. **Erase Secrets** is in the **Security** section of the **More Bulk Options** popup. See [Running Dashboard Bulk Operations](#).



Note: If the "Erase Secrets" link does not appear in the Security section (when erasing from the dashboard) or "Erase" is not available on the Option button (when erasing from the General tab) you may have not properly configured secret erase (see Task 1) or the secret might not meet one of the requirements above.

4. When you click the **Erase Secrets** link, the Erase Secrets popup appears:

Erase Secrets

Permanently delete all data from the selected Secrets.
Deletion will occur once the Erase After time has passed.

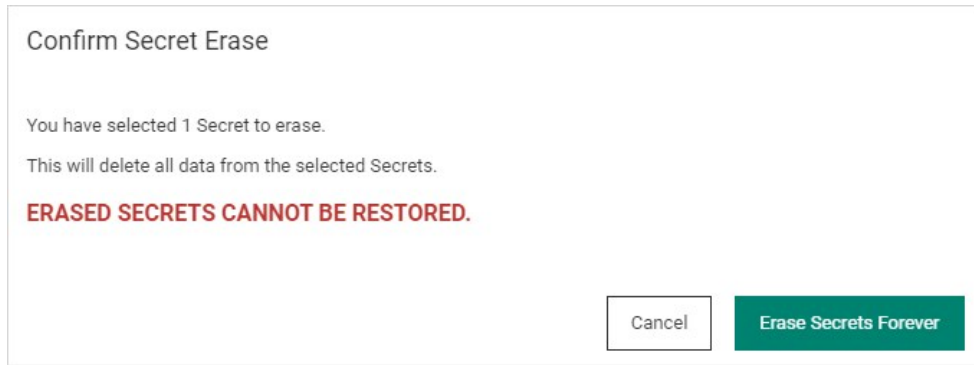
Secrets Selected 1

Erase After Date * 7/15/2021  2:30 PM 

Reason *

Here, you are essentially setting up a erase secrets request. The access request is sent to the users or user group you designated earlier.

5. Use the calendar and time widgets to set the **Erase After Date**. It must be minimum of 24 hours away to give the erase secrets request time to process. If you set it to less than that, you cannot continue the process.
6. Type your reasoning for permanently erasing the secret or secrets in the **Reason** text box. This is not tedium—the granter will need this to decide whether to let you take this irreversible, destructive action. Specifically, explain why a deactivation is not sufficient.
7. Click the **Erase** button. A confirmation popup appears:



8. Pause a second, and make sure you are sure.
9. Click the **Erase Secrets Forever** button.
10. When the erase request is approved, the secret or secrets will be erased by an automated process after the "erase after" date and time arrives.

Overriding the Secret Template's Password Requirements

All secrets inherit a set of password requirements (see [Template Password Requirements](#)) from their parent secret template. After you create a secret, you can choose to use a different password requirement for this one secret, which leaves other secrets based on the template as they were. To choose a different password requirement for the secret:

1. Navigate to the secret **View** page by searching or drilling down the folder tree.
2. Click the secret to open the secret's page.
3. Click the **Security** tab.
4. Click the **Edit** Link in the **Password Requirements** subsection in the **Other Security** section. The Edit Password popup appears.
5. Click the Password Requirement dropdown list to select the password requirement you desire.
6. Click the **Save** button.

Setting Up Password Masking

Password masking prevents over-the-shoulder viewing of your passwords by a casual observer (passwords show as *****). For security, the number of asterisks does not relate to the length of the password.

As an administrator, you can force all the secret password text-entry fields in the system when viewed to be masked. To do this, enable the **Force Password Masking** setting in the **Configuration** settings. Only secret text-entry fields marked as a password text-entry field on the secret template is masked.

There is also a user preference setting that forces password masking on all secret password text-entry fields viewed by the user. This Mask passwords when viewing secrets setting is found in the **Profile > Preference** section for each user. If the configuration setting discussed above is enabled, this user preference setting is overridden and cannot be disabled.

Sharing Secrets

Sharing passwords is crucial for information technology teams. Due to the sensitive nature of sharing secure information, Secret Server ensures shared passwords are tracked and guarded.

Permissions

There are three permission levels to choose from when sharing secrets with another user or group:

- **View:** User may see all secret data, such as username and password, and metadata, such as permissions, auditing, history, and security settings.
- **Edit:** User may edit the secret data. Also allows users to move the secret to another folder unless the Inherit Permissions from Folder setting is turned on, in which case the user needs Owner permissions to move the secret.
- **List:** User may see the secret in a list, such as a list returned by running a search, but not to view any more details about a secret or edit it.
- **Owner:** User may change all the secret's metadata.

Note: Password text-entry fields are not visible if a secret has a launcher and the Hide Launcher Password setting is on or the user does not have the View Launcher Password role permission. See the table below.

Secrets can be shared with either groups or individual users. The Secret Sharing section allows secrets to be configured for access.

Password Visibility

Password visibility in the password text box depends on secret access permission, role permissions, and secret security policy settings. The following table shows the possible combinations and their password visibility result.

Table: Password Visibility Determinants

Owner	No	On	No
Owner	Yes	On	Yes
Owner	Yes	Off	Yes
Owner	No	Off	Yes
Edit	Yes	On	No
Edit	No	On	No
Edit	Yes	Off	Yes
Edit	No	Off	Yes
View	Yes	On	No
View	No	On	No
View	No	Off	No

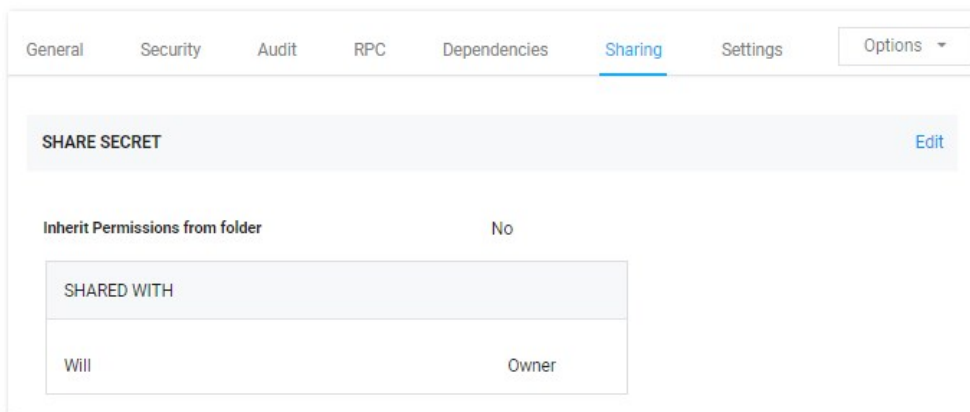
View	Yes	Off	Yes
List	Yes	On	No
List	No	On	No
List	Yes	Off	No
List	No	Off	No

Procedure

To add or remove secret sharing:

Note: To simplify the sharing process, new secrets automatically inherit the settings from the folder they are stored in. That is, we enable the **Inherit Permissions from Folder** check box on the **Sharing Edit** page by default, so secrets inherit all the parent folders' sharing settings. As long as this check box is selected, you cannot set the permissions for the secret. For more on folder security, see the [Folders](#) section.

1. [View the secret](#) you want to share.
2. Click the **Sharing** tab.



3. Click the **Edit** link. The page becomes editable:

SHARE SECRET

Inherit Permissions from folder

SHARED WITH		
Will	Owner ▾	Remove

Add Groups / Users

Search for groups or users 🔍

Cancel Save

4. Click the **Remove** link next to any share you want to delete.
5. Type any user or group you want to share with in the **Add Groups / Users** search text box.
6. When the user or group appears in the dropdown list, click to select it. The user or group appears in the **Shared with** table.
7. Click the unlabeled permission dropdown list box to select the desired permission.
8. Repeat the process for additional users or groups.
9. Click the **Save** button to commit the changes.

You can also modify sharing settings for users or groups that already have sharing enabled for the secret. If a user or group is not displayed, they do not have access to the secret.

Viewing Secrets

To view the information contained in a secret:


1. Locate the desired secret in one of these ways:
 - On the main menu, drill down the folders tree to select the secret.
 - Click the **Secret** menu item on the main menu and find the secret in the **All Secrets** table. You can filter the list or click the magnifying glass icon to search for the secret.
2. Click on the secret's name link. The secret's view page opens to the General tab.
3. Click the desired tab to view specific information. For example, click the General tab and go to the Expiration and Heartbeat section to see if the secret's password has expired and what its expiration interval is. You can check the history of the secret by clicking the Audit tab.
4. [Edit the secret](#) if desired.

Searching for Secrets

To search for secrets:

1. Click the **Secrets** menu item in the main menu. The All Secrets page appears:

<input type="checkbox"/>	NAME	SECRET TEMPLATE	FOLDER
<input type="checkbox"/> ☆	10.60.19.69	Unix Account (SSH)	Everyone/Test has been renamed
<input type="checkbox"/> ☆	AD with Deny List	Active Directory Account	Everyone
<input type="checkbox"/> ☆	adminOne	Bank Account	Playground
<input type="checkbox"/> ☆	adminTwo	Bank Account	Playground
<input type="checkbox"/> ☆	Approval Required	Credit Card	aaaaa/Everyone
<input type="checkbox"/> ☆	bob'spassword	Bank Account	Everyone/Test has been renamed
<input type="checkbox"/> ☆	bob'spassword	Bank Account	Everyone/Test has been renamed
<input type="checkbox"/> ☆	BSD (Key Only) (Engine Proxy)	Unix Account (SSH)	RPC/Heartbeat/Local
<input type="checkbox"/> ☆	BSD (session recording)	Unix Account (SSH)	RPC/Heartbeat/Local
<input type="checkbox"/> ☆	CentOS (Key Only) (Engine Proxy)	Unix Account (SSH Key Rot...	RPC/Heartbeat/Local
<input type="checkbox"/> ☆	CentOS Privileged Key Only	Unix Account (SSH Key Rot...	RPC/Heartbeat/Local

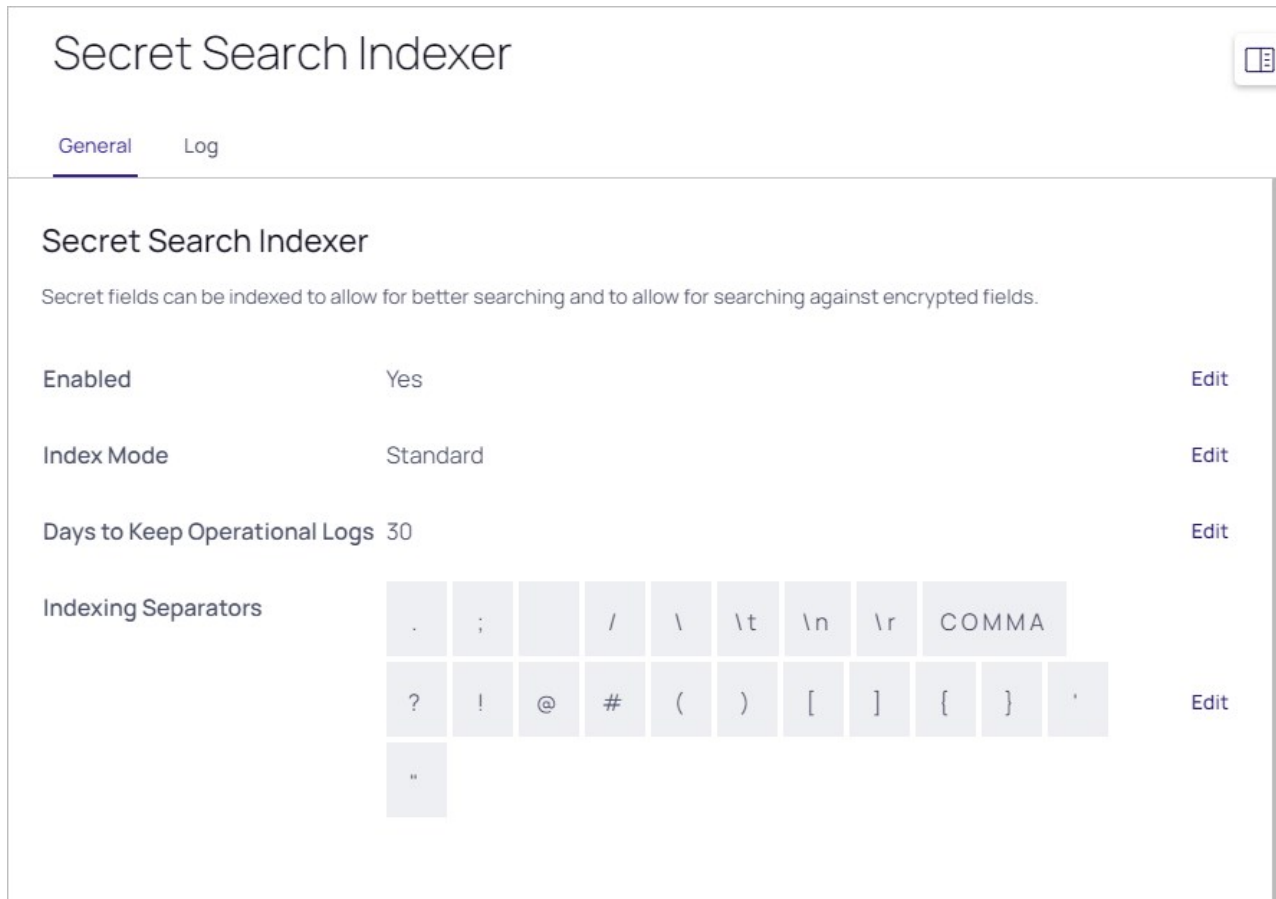
2. Type the secret name or other text in the unlabeled search text box at the top of the page.
3. Click the  button. The All Secrets table only displays matching secrets. Searches search for all text-entry fields that are configured as searchable on the secret's template if the extended search indexer is enabled.

Important: If the search indexer is not enabled, searches are only performed on the **Secret Name** text field.

Search Indexer

The *search indexer* allows searching on all text-entry fields set to searchable on the template. To enable and configure the search indexer:

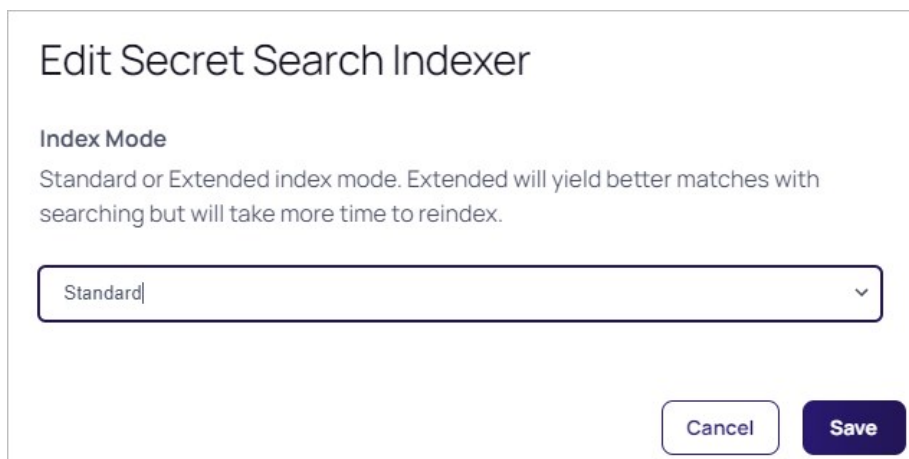
1. Click the **Admin** button on the main menu.
2. Click the >> icon to view the Admin slideout.
3. Type and then click Search Indexer in the Search text box. The Indexing Service page appears:



The screenshot shows the 'Secret Search Indexer' configuration page. At the top, there are tabs for 'General' and 'Log'. Below the tabs, the page title 'Secret Search Indexer' is displayed, followed by a descriptive sentence: 'Secret fields can be indexed to allow for better searching and to allow for searching against encrypted fields.' The configuration is presented in a table-like format with 'Edit' links for each row.

Enabled	Yes	Edit
Index Mode	Standard	Edit
Days to Keep Operational Logs	30	Edit
Indexing Separators	. ; / \ \t \n \r COMMA ? ! @ # () [] { } ' "	Edit

4. Click the **Edit** link next to it to ensure the **Enabled** check box is selected. Click **Save** in the popup if you changed anything.
5. Click the Edit link for the Index Mode. A popup appears:



The screenshot shows a popup titled 'Edit Secret Search Indexer'. It features a section for 'Index Mode' with a descriptive text: 'Standard or Extended index mode. Extended will yield better matches with searching but will take more time to reindex.' Below this text is a dropdown menu currently set to 'Standard'. At the bottom of the popup, there are two buttons: 'Cancel' and 'Save'.

6. Click the dropdown list to select either **Standard** or **Extended**:

- *Standard search mode* is the default and searches on whole words in a field value. For example, a field value of "My AWS Secret" would match when you search for *My AWS Secret*, *My*, or *Secret*.
- *Extended search mode* searches for whole words or a partial words by up to twelve characters. For example, a field value of "My AWS Secret" would match when you search for *My AWS Secret*, *My*, *Secret*, *WS*, or *ecret*. This is more useful, but may impact search performance and creates a larger index table.

Note: Indexing separators are used to split the text text-entry fields into search terms. By default, the separators are semi-colon, space, forward slash, back slash, tab (`\t`), new-line (`\n`), return (`\r`), and comma. Changes to the indexing separators require a full rebuild of the search index.

7. Click the **Save** button.

8. Click the Edit link to change the **Days to Keep Operational Logs** spinner to set the period to keep indexing-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).

9. Click the **Save** button. The Indexing Service page reappears, and the indexing begins in the background. Depending on the size of the Secret Server installation, it may take awhile. Progress is shown on the Progress bar.

10. If you changed the indexing separators, click the **Rebuild Now** link.

Common Configuration Options

These are the configuration options that are common to every secret:

- **Convert Template:** Change which template is being used to store and display information in this Secret.
- **Copy Secret:** Create a duplicate copy of the secret, which may also be renamed and modified.
- **Delete:** Delete the secret.
- **Edit:** Edit the secret parameters.
- **Favorite:** Click the star from the Dashboard or check this box on the Secret View page to mark the Secret as a favorite. It then displays in the Favorite Secrets widget.
- **Folder:** Folder location of the secret. The secret inherits permissions of this folder, depending on the Default Secret Permissions setting in the Secret Server Configuration options.
- **Share:** Configure the sharing settings, or permissions, for the secret.
- **View Audit:** View the secret audit log to see which users have accessed the secret and the actions that have been performed.

Advanced Configuration Options

These are the buttons, fields, and icons that are available for more advanced secrets:

- **Expire Now:** Expire the secret manually.
- **RDP Launcher Icon:** Click to open the Remote Desktop Protocol (RDP) Launcher. See further details in the Launcher section.
- **Run Heartbeat:** Initiate heartbeat, which attempts to verify that the secret credentials can authenticate.
- **Site:** Edit the secret to set the distributed engine site. This determines where password changing, heartbeat, and proxied sessions run from.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret expiration is a core Secret Server feature. Any template can be set to expire within a fixed time interval. For a secret to expire, a text field must be selected as the target of the expiration. For example, a secret template for Active Directory accounts might require a change on the password text field every 90 days. If the password remains unchanged past the length of time specified, that secret has expired and appears in the Expired Secrets panel on either the Dashboard's Expired secrets widget or the Home page.

Secret expiration provides additional security by reminding users when sensitive data requires review. This assists in meeting compliance requirements that mandate certain passwords be regularly changed. When expiration is combined with RPC, Secret Server can completely automate the process of regularly changing entire sets of passwords to meet security needs.

Forcing Expirations

To force expiration:

1. Navigate to the **Secret View** page for the desired secret.
2. Click the **Expire Now** button. This forces the secret to expire immediately regardless of the interval setting. The expiration date displays "Expiration Forced."

Resetting Expired Secrets

To reset an expired secret, you must change the text field that has expired and is required to change. For example, if the text field set to expire is the password text field and the current password is "asdf," then a change to "jklh" resets the expiration interval and thus removes the expiration text on the Secret View page.

If you do not know which text field is set to expire:

1. Go to the secret template that the secret was created from.
2. Navigate to **Admin > Secret Template**.
3. Select the template.
4. Click the **Edit** button.
5. On the next page, click the **Change** link. In the **Change Required On** text box you can see the text field that is set to expire.

Setting up Secret Templates for Secret Expiration

To set up expiration on a secret, you must first enable expiration on the template from which the secret is created.

To enable secret expiration for a secret template:

1. Navigate to **Admin > Secret Templates**.
2. On the **Secret Templates** page, click the template name. The page for that template appears.
3. Scroll down to the **Template Expiration** section.
4. Click the **Edit** button.
5. On the **Secret Template Designer** page, click on the **Change** link. The **Expiration Enabled?** check box appears.
6. Click to select the check box. Two more fields appear.
7. Type the number of days desired in the **Days until Expiration** text box.
8. Click the Change Required on dropdown list to select which parameter you want updated upon expiration. Your choices are:
 - Computer
 - Domain
 - My Server List
 - Password
 - Server
 - Username
9. Click the **Save** button.

Note: You can override the interval setting for individual secrets.

Note: Enabling expiration for a template enables expiration for all the secrets that were created using that template.

Setting up Secrets

Once you enable expiration for the template, expiration is also enabled for secrets that were created using that template as well as secrets created in the future. The Expiration tab appears on the Secret View page and requires the user to have Owner permission on the secret.

To set a custom expiration at the secret level, you adjust the expiration interval for the secret by clicking the **Expiration** tab in the **Secret View** page. There, you can set the secret to expire using the template settings (default), a custom interval, or a specific date in the future.

The Secret Navigation Slideout is a set of useful links to secret. Its tab appears on the right side of all top-level pages:



Click the tab and the slideout appears:



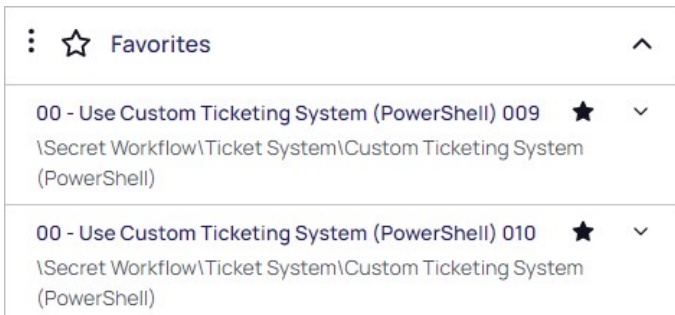
There are four dropdowns:

Favorites

These are the secrets you set as favorites by clicking the star icon on the secret's row on the All Secrets table.

Dropdown Quick List

When you click the dropdown, you see a list of your favorite secrets:



The initial view shows the folder path to the secret and a "favorite" toggle icon. You may need to hover over the blank space to see an unselected toggle.

Note: Some changes may require you to refresh the list by clicking the stacked dots icon.

Each list item has its own dropdown. When you click it, a set of information about the secret appears, including a list of users or roles with access to (shared with) the secret.

Note: You may also have to enter a comment or check out the secret.

Most of the datums can be copied with a single click of the copy icon next to each. You can hover the mouse pointer over the "Shared with" list to see the type of access for that user or role, or you can click "See All" to see the entire list.

Favorites Page

Alternatively, in the slideout, you can click the Favorites title to navigate to the full Favorites page.

The screenshot shows the 'Favorites' page interface. At the top, there is a title 'Favorites' and a search bar. Below the search bar, there are two filter buttons: 'Active' and 'All Templates'. The main content is a table with the following columns: 'NAME', 'SECRET TEMPLATE', and 'FOLDER'. There are three items listed in the table, each with a star icon and a checkbox. The first two items have the same name: '00 - Use Custom Ticketing System ...' and the same folder: 'Sec.../Custom Ticketin'. The third item has a different name: 'GCP Rotate For Me Service Account' and a different folder: 'RPC/Heartbeat/GCP IA'.

<input type="checkbox"/>	NAME	SECRET TEMPLATE	FOLDER		
<input type="checkbox"/>	★ 00 - Use Custom Ticketing System ...	Simple Template	Sec.../Custom Ticketin		
<input type="checkbox"/>	★ 00 - Use Custom Ticketing System ...	Simple Template	Sec.../Custom Ticketin		
<input type="checkbox"/>	★ GCP Rotate For Me Service Account	Google IAM Service Accou...	RPC/Heartbeat/GCP IA		

Recent

These are the secrets you have opened recently. The dropdown provided for each list item is exactly the same as that provided in "favorites," as is accessing the Recent page via clicking the title.

Most Used Secrets

These are the secrets you have opened the most often. The dropdown provided for each list item is nearly the same as that provided in "favorites," as is accessing the Recent page via clicking the title. The only difference is each list item states the time since last access.

Shared with Me

These are the secrets others have shared with you. No folder path is provided.

The dropdown provided for each list item provides a shortcut to request approval for the secret:

The screenshot shows a 'Shared Secret 1' card. At the top, there is a title 'Shared Secret 1' and a small upward arrow icon. Below the title is a large key icon. Underneath the key icon, the text 'Shared Secret 1' is displayed. Below that, a message reads: 'This Secret requires approval to access.' At the bottom of the card, there is a dark blue button with the text 'Request Approval' in white.

Sharing passwords is crucial for information technology teams. Due to the sensitive nature of sharing secure information, Secret Server ensures shared passwords are tracked and guarded via permissions.

Note: Depending on your configuration, folder settings can affect the permissions of secrets contained in that folder (and subfolders). Secrets and folders are not visible to users that do not have at least View (can see details) or list (can see a list of secrets) permission. See [Secret Folder Permissions](#) for more information.

Note: To simplify sharing, new secrets automatically inherit the settings from the folder they are stored in. That is, we enable the "Inherit Permissions from Folder" check box on the "Sharing Edit" page by default, so secrets inherit all the parent folders' sharing settings. As long as this check box is selected, you cannot set the permissions for the secret. For more on folder security, see the [Folders](#) section.

There are four permission levels when sharing secrets with another user or group:

- **View:** User may see all secret data, such as username and password, and metadata, such as permissions, auditing, history, and security settings.
- **Edit:** User may edit the secret data. Also allows users to move the secret to another folder unless the Inherit Permissions from Folder setting is turned on, in which case the user needs Owner permissions to move the secret.
- **List:** User may see the secret in a list, such as a list returned by running a search, but not to view any more details about a secret or edit it.
- **Owner:** User may change all the secret's metadata.

Note: Password text-entry fields are not visible if a secret has a launcher and the Hide Launcher Password setting is on or the user does not have the View Launcher Password role permission.

Secrets can be shared with either groups or individual users. The Secret Sharing section allows secrets to be configured for access.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Dependencies Tab

The settings inside the Dependencies tab are used for secrets that have RPC enabled.

See [Manually Adding Dependencies](#) for details.

Secret Expiration Tab

Inside the Expiration tab, the expiration period can be modified. The following options are available:

- **Template Interval:** Default expiration period configured for new secrets based on the current template.
- **Custom Interval:** A custom expiration period in days.
- **Custom Date:** A custom expiration date in month/day/year format.

Note: See [Secret Expiration](#) for details.

Secret Launcher Tab

The Launcher tab appears for secrets that use either a custom launcher or Web launcher.

If a custom launcher is associated with a secret template, a secret owner can configure associated secrets or a privileged secret to run the launcher process. The associated secret can be tied in to the command line parameters on the custom launcher, and the privileged secret is the identity that kicks off the launcher process.

If a Web launcher is associated with a secret template, the launcher tab displays how the Web launcher is configured for that secret. The following options are available:

- **Edit Fields:** Modify which secret text-entry fields are mapped to the HTML input controls on the target website.
- **Reconfigure Web Launcher:** Reset the Web launcher configuration.
- **Test Launcher:** Test the current Web Launcher configuration.
- **Use Web Password Filler:** Use the Web password filler rather than the Web launcher.

Note: See [Web Launcher](#) for details.

Secret Personalize Tab

These settings only apply to the user who is editing the settings. They do not apply to the other users who have View, Edit, or Owner permission to the secret.

To use the settings in the Email Notifications section, you must have email configured correctly in your configuration settings. You also need a valid email address entered for each user account to use these settings. This can be set in the **Administration > Users** section.

The following email notification settings are available:

- **Send Email When Changed:** Email the user when the secret is edited by any user.
- **Send Email When Heartbeat Fails:** Email the user when a heartbeat function fails for the secret. The email contains the secret name, error code and details.
- **Send Email When Viewed:** Email the user when the secret is viewed by any user.

The Personalize tab also contains settings that pertain to the type of launcher configured for a secret. If the launcher type is Remote Desktop Protocol (RDP), the following settings are available:

- **Connect to Console:** Remote Desktop (RD) may connect to the console session.
- **Allow Access to Printers:** RD may access local printers.
- **Allow Access to Drives:** RD may access drives connected to the local machine.
- **Allow Access to Clipboard:** RD may access the clipboard of the local machine.
- **Use Custom Window Size:** Users may specify custom window height and width. Use Preferences refer to the user's settings under **Profile > Preferences** in the **Launcher** tab.

Users may enable or disable these settings or to defer to what is configured in their user settings by selecting **Use Preferences**.

Secret RPC Tab

The settings inside the Remote Password Changing tab are used for secrets that are Remote Password Changing (RPC) enabled:

- **Auto Change:** Enable or disable auto change for the secret.
- **Next Password:** Specify the next password

Note: See [Remote Password Changing](#) for details.

Secret Security Tab

The Security tab contains settings that can be enabled to increase security for a secret. The settings listed below may or may not be visible, depending on your configuration settings:

Checkout

- **Require Check Out:** Only one user at a time has access to a secret. When enabled users must checkout a secret before they can access it. Checkout prevents other users from accessing the Secret while it is checked out. See [Secret Checkout](#) for details.
- **Check Out Interval:** The default time a checkout lasts.

Approval

Require Approval Type: Require users to get approval before accessing this secret. Define whether standard users, editors, or everyone has to have approval and the workflow through which approval occurs.

Password Requirements

- Defines which password rules apply to password fields on this secret. By default, a secret uses the password requirements as defined on the secret template and can be overridden per secret as needed.
- **Validation:** Lists requirements from the template.
- **Password:** Either confirms the default or lists the chosen override.
- **Private Key Passphrase:** The password required to access a private key. Either confirms the default or lists the chosen override.

Other Security

- **Require Comment:** Users must enter a comment before being granted access to view the secret. The comment is stored in the audit log for that secret.
- **Enable DoubleLock:** User must enter a doubleLock password to decrypt and view a secret.
- **Enable Proxy:** Enable or disable proxying. When enabled, SSH and RDP launcher traffic passes through Secret Server.
- **Enable Session Recording:** Record the Launcher session. This applies to secrets with a launcher associated with the secret template.
- **Viewing Password Requires Edit:** To view the password, the user must have at least edit permission.

Secret Server Cloud

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This section contains information that is exclusive to Secret Server Cloud.

Introduction

Secret Server protects your secrets using a master encryption key, as well as an additional intermediate encryption key that is unique for each secret. These effectively act as internal passwords that Secret Server itself needs to unlock your data, for example any time you view or update a secret.

Key Management in Secret Server Cloud (SSC) allows you to add an additional layer of encryption using a third-party provider to protect these encryption keys for added protection and control. To do this, you must first set up your own encryption key with a third party that you fully control, and then provide Secret Server limited access to it. This external encryption key is used to protect the Secret Server encryption keys. You can revoke Secret Server's access at any time if the need arises, rendering Secrets unusable.

Important: Once enabled, beware that if you delete your external third-party encryption key, or the credentials you gave Secret Server no longer work. *You will not be able to access your existing Secrets, and even Delinea will not be able to help!*

You can change your key management configuration through Secret Server's Web interface or by using the REST API. If key management has already been enabled, you can switch to a new configuration or disable key management completely. To make any change, your existing key management configuration **must still be valid**, so your secrets and the master encryption keys can be converted to the new configuration. Your new settings are validated before they can be saved.

Secret Server Cloud currently supports Amazon's Key Management Service.

Configuring Key Management

Overview

To enable key management, you will first create an encryption key with your third-party provider, then an API account that Secret Server will use in order to access the key. After the external encryption key is setup, you will update Secret Server with the details.

Important: Changing your key management settings will trigger "maintenance mode" and a secret key rotation that will re-encrypt all your secret keys. No one will be able to access secrets until the rotation finishes, and it must finish successfully before further key management changes can be made.

Navigate to Secret Server's key management page by clicking **Admin > All > Key Management**.

Here you can change your key management settings, as well as view the audit history showing all key management updates.

Key Management Providers

SSC currently supports one provider, AWS Key Management Service. More providers may be added over time. Azure's KeyVault service is not a viable provider at this time due to slow speed limits when using strong encryption keys (such as 4096-bit RSA with HSM).

AWS Key Management Services Pricing

Please see [AWS Key Management Service Pricing](#).

SSC requires one AWS Key ("CMK"), and the number of requests per month will vary depending on how often secrets are accessed.

Procedure

Important: Changing your key management settings triggers SSC maintenance mode and a secret key rotation that re-encrypts (or de-encrypts) all your secret keys! No one can access secrets until the rotation finishes, and it must finish successfully before further key management changes can be made.

Task 1: Setting up the Encryption Key and IAM User in AWS

1. Log into the AWS Console website at <https://console.aws.amazon.com/>.
2. Under **Services**, search for **IAM** (Identity and Access Management). This is where you will configure both your encryption key and an IAM user for Secret Server to use to access the encryption key.
3. Click the **Users** button on the left menu.
4. Click **Add User** button.
5. Type a name (such as *SecretServerCloud*) in the **User Name** text box.
6. Click to select the **Programmatic Access** check box in the **Access Type** section.
7. Click the **Next: Permissions** button. The Permissions page appears.
8. Click the **Next: Review** button (on the Permissions page, no special permissions are needed). The Review page appears.
9. Click the **Create User** button. A Success page appears confirming the user was created. Both the access key ID and the secret access key appear (click the **Show** link).
10. Click the **Download .csv** button to save the credentials

Important: Be sure to **save both the access key ID and the secret access key!** If you lose them, you can never view the secret access key again. Even after you enter them in SSC, you cannot retrieve the secret access key.

11. Click the **Encryption Keys** button in the left menu.
12. Click the **Region** dropdown list to select a region. We recommend picking **US East (Virginia)** if you are using *.secretservercloud.com, or **EU (Frankfurt)** if you are using *.secretservercloud.eu.
13. Click the **Create key** button. The Create Alias and Description page appears.
14. Type an alias in the **Alias** text box (such as *SecretServerCloud*).
15. (Optional) Type a description in the **Description** text box.
16. In the **Advanced Options** section, you can either let Amazon create a new random key for you (default), or you can provide your own key, which is beyond the scope of this guide. Click the **Learn More** link for more information. Either way, Amazon will have access to the key because they are providing the encryption and decryption services.
17. Click the **Next Step** button. The Add Tags page appears.
18. Click the **Next Step** button again, unless you want to add optional tags. The Define Key Administrative Permissions page appears.
19. Click the **Next Step** button again. The Define Key Usage Permissions page appears.

Important: Do **not** give access to the user you created earlier for SSC. It is unnecessary for Secret Server to have administrative access to the key.

20. Click to select the check box next to SecretServerCloud to give that user access to the key.
21. Click the **Next Step** button. The Preview Key Policy page appears.
22. Click the **Finish** button. The new key appears in your Encryption Keys list.
23. Click to select the new key in the list. The Summary section on the key's page appears.

24. Copy and save the contents of the read-only **ARN** text box. You will need it later.

Note: AWS supports automatically rotating this key every year. You can change that setting on this page in the **Key Rotation** section (select the "Rotate this Key every year" check box). Once rotated, the key management settings in Secret Server will not require further changes, and your existing secrets can still be accessed by the old encryption settings. However, only new secrets will be created under the new version of the encryption key, and you must perform a secret key rotation inside SSC if you want to update all secrets to use the new version of the AWS key.

Note: As a security best practice, we recommend performing a secret key rotation inside of SSC on a regular basis to refresh the encryption keys on your Secrets. Go to **Admin > Configuration > Security**, and click **Rotate Secret Keys**.

Task 2: Adding Encryption Key and User Details in Secret Server

1. In SSC, go to **Administration > Key Management**. The Key Management page appears.
2. Click the **Edit** button. The page becomes editable.
3. Click the **Key Management Type** dropdown list to select **Amazon KMS**.
4. Type your AWS key details that you saved earlier in the remaining four text boxes.
5. Click the **Save** button.

Task 3: Secret Key Rotation

1. Once you save your changes, your new settings are validated and a secret key rotation is triggered.
2. View the progress of the rotation:
 1. Go to **Admin > Configuration**.
 2. Click the **Security** tab.
 3. Go to the **Key Rotation** section.
3. Later you can repeat the process to change the AWS encryption key, or you can select **None** for the **Key Management Type** to disable it completely.

Secret Server Key Management via the REST API

SSC has a REST API for retrieving or updating your key management configuration. For details:

1. Log on your SSC instance.
2. Click the question mark icon in the top right corner and select **Secret Server REST API Guide**.
3. Click on the **Documentation for REST API** document link for your authentication style, normal tokens or Windows Integrated Authentication.
4. Search for KeyManagement to view that section of our API.

Important: When changed via the API, maintenance mode and a secret key rotation still occur.

Introduction

This topic addresses migrating a Secret Server Online instance to a Secret Server Cloud instance using Secret Server's XML import and export.

Note: Secret Server Online does not support for Active Directory (AD) groups or users. This instruction assumes that you will use them in your new Secret Server Cloud instance.

Note: Migrating Secret Server is considered a professional service under the Delinea support policy. Technical support can assist in troubleshooting errors but cannot walk you through the whole migration process

What Is Migrated

The following are migrated from Secret Server Online to Secret Server Cloud:

- Folders
- Secret templates
- Secrets
- Users

The following are not:

- Deleted secret policies
- File attachments
- Log history (System log, RPC log, and audit logs)
- Object IDs

Note: All IDs are remapped. **Changing secret IDs requires that you update any links to secrets.**

- Secret audit data
- Secret field history
- User audit data

Procedure

Task One: Exporting from Secret Server Online

To export XML from Secret Server Online:

1. Ensure you have an XML editor, such as [Notepad++](#), installed.
2. Ensure you have admin rights in both Secret Server Online and Secret Server Cloud instances.
3. Enable [unlimited administration mode](#) for both Secret Server Online and Secret Server Cloud instances.
4. Remove the Add Secret, Edit Secret, and Share Secret role permissions from user roles in Secret Server Online to prevent users from modifying data after the migration is complete:

1. Go to **Admin > Roles**.
2. Click to edit the **user** role.
3. Remove the **add secret**, **edit secret**, and **share secret** role permissions.
4. Save your changes.
5. Recreate local users and groups on Secret Server Cloud that are equivalent to those on Secret Server Online.
6. If you have customized any of the default templates, rename them to identify the templates as custom—for instance "Active Directory - Custom." This ensures that your edits do not cause conflicts with the templates in the new environment.
7. If you have added custom fields to default secret templates, you need to recreate those fields in Secret Server Cloud *before* the migration. Failing to do so will prevent the secret template and any secrets based on it from being imported.
8. If you enabled duplicate secrets in Secret Server Online, rename any duplicates prior to exporting. Importing creates new IDs for the imported secrets and does not recognize duplicate secret names. If not renamed before migration, duplicate secrets are not imported.
9. If you enabled secrets outside of folders in Secret Server Online, manually move the found folder-less secrets to a folder prior to export. You can use a bulk operation to move the secrets into a temporary or new folder.

Note: Any advanced items, such as dependencies, custom scripting, or custom reports. May require the Delinea professional service team, which may incur an additional charge.

10. Go to **Admin > Export**. The Export page appears:

Export

Please enter your password for security purposes.

Folder

Password *

Export with folder path

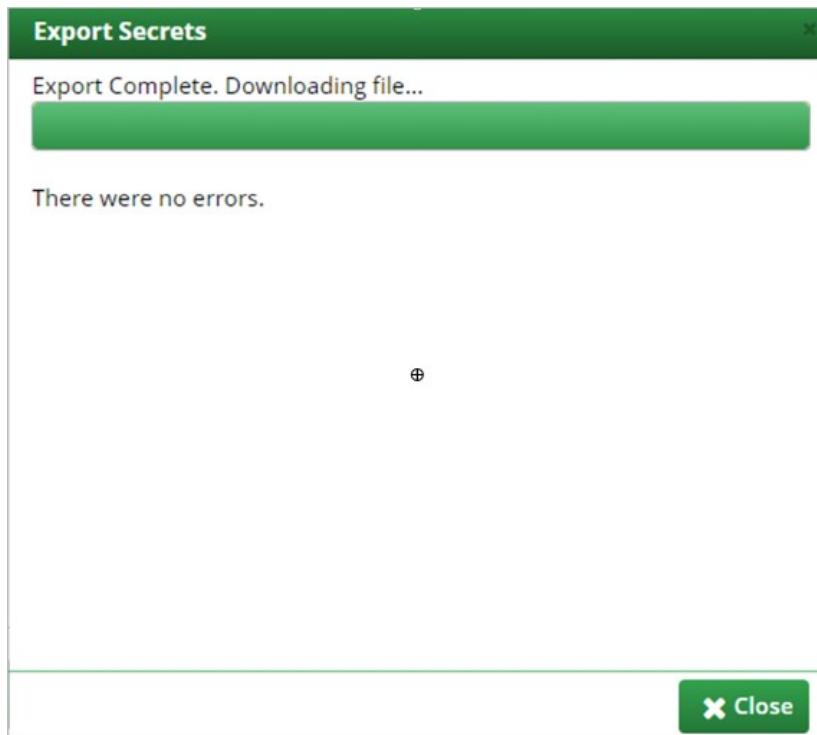
Export Child Folders

Export Format CSV XML

Enter any additional notes or explanations for the export.

11. Do not select a folder if you want to export everything. This exports all Secrets, including secrets in personal folders. Select a specific folder to limit the export scope.

12. Type your password in the **Password** text box.
13. Ensure the **Export with Folder Path** check box is selected.
14. Ensure the **Export Child Folders** check box is selected.
15. Click to select **XML** for the **Export Format** selection button.
16. Click the **Export** button to start the export. When it is complete, the Exports Secrets confirmation appears:



17. Click the **Close** button.
18. Open the XML file in your XML editor.
19. If you had any custom templates, confirm that the field names in the source templates match those in the destination templates.

Task Two: Importing to Secret Server Cloud

1. Log on your Secret Server Cloud.
2. In Secret Server Cloud, go to **Admin > Configuration > General Tab > Permissions**.

PERMISSION OPTIONS	
Default Secret Permissions	Secrets inherit permissions from folder
Allow Duplicate Secret Names	No
Force Require Approval for Owners on Approval Secrets	No
Force Require Approval for Editors on Approval Secrets	Yes
Allow Approval For Access from Email	Yes
Allow View User To Retrieve Auto-Change Next Password	No

3. Review the permissions to ensure they match those in the above screen grab.
4. Import the XML file. See [Importing Secrets with XML](#) for instructions. Ensure **Secrets without Permissions** is set to **Inherit Permissions from Folder**.
Note: We recommend you copy the import secret log before continuing.
5. Validate the migrated data and adjust as required.

Task Three: Finishing up

1. Turn off [unlimited administration mode](#) when completed.
2. Recreate custom launchers, password changes, and password requirements and relink them to the new equivalent of their original secret template in Secret Server Online.
3. Recreate any even subscriptions you want to keep from Secret Server Online.
4. Recreate roles and their assignments to match what was in Secret Server Online.
5. Reconfigure settings in Secret Server Cloud to match those you changed in Secret Server Online.

Overview

Secret Server Cloud (SSC) is a scalable, multi-tenant cloud platform that provides the same features as the on-premise Secret Server Professional edition. With the SSC platform, all backend services, databases, and redundancy are securely managed by Delinea and hosted on the Microsoft Azure platform. Customers do not have direct access to the databases or application file system.

Cloud Versus On-Premise Secret Server

For documentation purposes, SSC is the same as the corresponding on-premise edition. However, there are some feature differences:

- **Site Connectors:** On-premise versions can use multiple site connectors to manage engine connections, such as RabbitMQ or MemoryMQ. The cloud version manages this for you as an Azure service and is not configurable.
- **CRM Integration:** On-premise versions can integrate with CRMs via direct database connections or the ConnectWise API. This is not currently available in SSC.

Getting Started

This section walks you through an initial configuration of your cloud instance. To see additional documentation for Secret Server features, please refer to the support resources section at the end of this document.

System Requirements

A distributed Engine server is required to communicate with SSC. Distributed engine server recommended specifications:

- Windows Server 2012 or Above
- CPU: 4-core 2 GHz (minimum)
- Memory: 4 GB of RAM (minimum)

Engine Connectivity

[SSC's Architecture Diagram](#) shows the network topology of your cloud instance. Your on-premises distributed engines do not need any inbound TCP/IP ports open (unless using RADIUS authentication). If you do not have outbound firewall policies in place, no firewall configuration is necessary. If you do, the distributed engines need outbound access to:

- SSC's multi-tenant front-end Web server
- A shared service bus
- A customer-specific service bus
- A Content Delivery Network (CDN)

The protocols and endpoint details are in the architecture diagram mentioned above.

Initial Setup

After you sign up for a trial, you can choose your URL name and provision your instance:

Note: To see additional documentation for Secret Server features, please refer to the support resources section at the end of this document.

1. After you sign signed up for a SSC trial, you received an email from Delinea Sales titled "Secret Server Cloud Trial." Click the **Cloud Portal** link in that email to begin your setup. The Setup Page appears in your browser.
2. Choose your location in the **Cloud Environment** dropdown list.

3. Click the **Continue** button. The Delinea One Portal appears.
4. Create the password for your first user account with administrator credentials. This account will be assigned to the email address you entered to request the trial.
5. After confirming the password, click the **Set Password and Login** button. The Delinea log on page appears.

Note: This is the backup admin account that you may need in a "break the glass" or unlimited admin situation. Delinea recommends you store the password in a secured physical location such as a safe or locked file cabinet. You can reset the password using an email reset, but **if this password is forgotten or you no longer have access to the email account, Delinea cannot reset this password.**

6. Click the blue button that matches the location you just chose. A setup page appears.
7. Type a name for your subdomain. Do not use special characters or spaces.
8. Read the End User License Agreement.
9. Click to select the check box to signify agreement.
10. From the dropdown, select **Yes** or **No** to signify your organization's oversight of EU information.
11. Click the **Accept** button. It may take several minutes for your new SSC to spin up.
12. When initialization is complete, click go to your SSC URL and click the **Login with Delinea One** button. You are automatically redirected to your new SSC dashboard.

Distributed Engine

All interaction between the SSC tenant and your on premises network uses our distributed engine service to communicate. The work tasks that distributed engine completes includes Active Directory authentication, password changing, and heartbeat. The machine where the engine is installed must be able to communicate outbound on port 443.

Note: For more information, see the [Distributed Engine Overview](#).

To install the Distributed Engine:

1. Navigate to **Admin > Distributed Engine**
2. Click the **Download Engine Installer** button for either 64-bit or 32-bit.

Note: You can install distributed engine on your workstation or laptop for testing purposes, but for production installs, the distributed engine server should be installed on a server. Secret Server uses the distributed engine to communicate with your domain, so if your machine is turned off, users cannot log on with their domain accounts, and heartbeat and remote password changing will fail.

3. Run setup.exe as an administrator to install the engine service. This will install into Thycotic Software Ltd\Distributed Engine.
4. Go to **Admin > Distributed Engine**.
5. Click **Manage Sites**.
6. Click **Manage New Engines**. There should be a new engine available.
7. Click the **Assigned Site** dropdown list and select **Default**.
8. Approve it by clicking the check box to the right.

9. Validate the engine's connectivity:

1. Go to **Admin > Distributed Engine > Manage Sites**.
2. Click the Default site.
3. Click the **Validate Connectivity** button to test the communication between the engine and Secret Server. It may take several minutes for the engine to register. If it does not immediately validate wait a few minutes and try again.

Configure Active Directory Integration

Active Directory integration allows users to log in with their domain credentials. Connections to your domain are routed through the distributed engine service running in your network.

1. On the dashboard, create a new Active Directory secret from the create secret widget in the upper right hand corner.

Note: The domain account should be able to read users and groups from the domain you want to sync. For detailed information on the rights required, please see [Active Directory Rights for Synchronization Account](#).

2. Type the domain, username, and password in the **Create Secret** form.
3. Save the secret.
4. Navigate to **Admin > Active Directory**.
5. Click **Edit** and check the boxes for **Enable Active Directory Integration** and **Enable Synchronization of Active Directory**.
6. Click the **Save** button.
7. Click the **Edit Domains** button.
8. Click the **Create New** button.
9. Type your FQDN and a friendly domain name that users will see on the login page.
10. Click **Sync Secret** to select the secret you just created.

Note: The domain site is set to default. This means that the Active Directory authentication and synchronization will run through the distributed engine service installed on your network.

Note: Do **not** select "Enable Login from AD." If you do, you cannot set the domain groups later in this instruction.

11. Click the **Save and Validate** button.
12. Click the **Back** button.
13. Click the **Edit Synchronization** button. The Synchronization Edit page appears.
14. In the **Available Groups** list, click each domain group that you want to log on in the SSC instance and click the the ◀ button to move the group to the **Synchronized Groups** list.
15. Click the **Save** button.
16. Click the **Synchronize Now** button to start the user and group synchronization immediately. The synchronization process runs automatically, but to get immediate results, you can start it manually.

Test Heartbeat and Remote Password Changing

Heartbeat ensures the secrets you have stored have the correct password, and Remote Password Changing (RPC) changes passwords on demand or a schedule.

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Edit** button.
3. Click to select the **Enable Remote Password Changing** and **Enable Heartbeat** check boxes.
4. Click the **Save** button.
5. Click the **Run Now** button in the **Remote Password Changing and Heartbeat Log** sections. This runs the heartbeat and RPC processes immediately.
6. Go to the secret you created for domain synchronization in the previous section or create a new test secret to use.
7. A brand new secret's **Last Heartbeat** status should be pending or processing. Once heartbeat completes you should see one of these statuses:
 - **Unable to Connect:** Secret Server could not reach the target machine. This could be a firewall issue or the machine name or IP address is wrong.
 - **Failed:** Secret Server could connect but could not authenticate. This likely means the password on the secret is incorrect.
 - **Success:** Secret Server successfully connected with the username and password.
8. You can test password changing by viewing a secret and clicking the **Change Password Remotely** button.

Note: This will change the password on the target system.
9. You can view the status of password changes and heartbeats in the log at **Admin > Remote Password Changing**.

Next Steps

- Add another user to the Administrator role in Secret Server. This allows you to have another administrator besides the initial user account created. To assign roles, go to **Admin > Roles** and click the **Assign Roles** button.
- Add a folder and share it with the group you synchronized from Active Directory. Create and edit folders from the Folder Tree View on your Dashboard.
- Create a secret in that folder for other users to see. When creating a secret, you can click the **Folder** link to save it to another folder.
- Have other users log on. Any users synchronized to Secret Server through the domain synchronization can log on with their domain credentials.
- Enable Google two-factor authentication by going to **Admin > Users**, editing the specific user, and assigning a two-factor option.

Troubleshooting and Resources

Get Error: "Site (Default) engines are not currently online" When Saving Domain

This can occur when Secret Server was not able to complete a round trip with the installed engine service. This validation may take several minutes for Secret Server to perform after the engine has been approved and assigned to the site. To address the issue:

1. On the server you installed engine on, check the logs in the install directory `C:\Program Files\Thycotic Software Ltd\Distributed Engine\log`.
2. If you see a message for "Could not configure, trying in 30 seconds" or a "Bus Broken Down Error" verify that the engine is approved and assigned to your default site.
3. Go to the site under **Admin > Distributed Engine > Manage Sites**.
4. Click the **Validate Connectivity** button.

5. If a success message appears and the engine status shows as online, try saving the domain again.

Secret Server End User Guide

This guide is for regular, non-administrative, users of Secret Server. It is mostly a set of links to a subset of the greater corpus of Secret Server documentation. For Secret Server Cloud, see the [Secret Server Cloud Quick Start](#).

Secret Server is a privileged access management (PAM) system. Essentially that means it manages who can access what, when, and under whose authority—all without introducing weak points, such as weak passwords or stale user accounts, and discovering those that potentially exist. For large organizations, this is a huge undertaking. It only takes one security breach to cause huge problems, and there are seemingly countless ways for those breaches to occur. PAM systems, such as Secret Server, are invaluable in getting this situation under control. Better still, Secret Server can make your day-to-day work environment safer and easier to manage too.

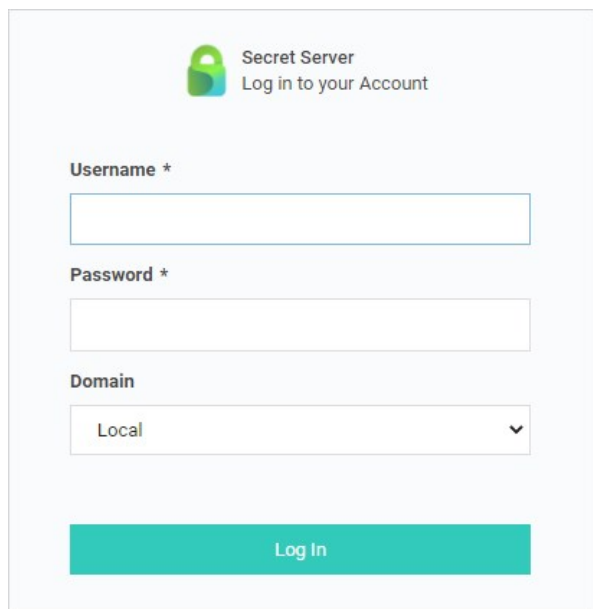
Secret Server is a powerful, advanced product with a wide range of capabilities. Even so, it is very easy to use for regular day-to-day operations for non-technical people. The key to this is knowing what to ignore and understanding the bits you do need to know. This guide is designed to help you do just that. It provides links to only what you need to know. You can add other topics later as needed.

- Technical Support: Please contact your organization's help desk.
- [Self-Help Resources](#)
- [Secret Server Glossary](#)
- [Document Conventions](#)

Important: When using this User Guide, it is easy to get lost in the ocean of Secret Server documentation. To avoid that, we recommend using **<Ctrl >** + click to access the links here. That way, the page you are going to will open to a new browser window, leaving this one as is, making it much easier to get back to. You can also simply use the browser back button to return, but that can get tiresome because many pages link to others.

Depending on how your administrators configured Secret Server, you can log on with either your Active Directory account or a local account.

1. In your browser, go to the URL for your organization's Secret Server.



Secret Server
Log in to your Account

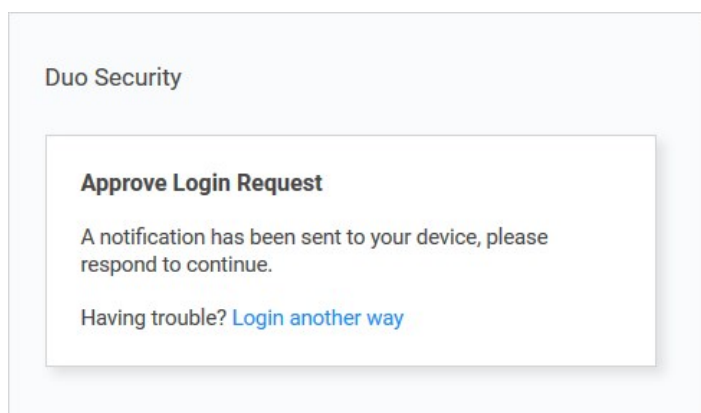
Username *

Password *

Domain
Local

Log In

2. On the login screen, enter your:
 - Active Directory username (or local one if you do not have one)
 - Active Directory password (or local one if you do not have one)
3. Select the your domain from the **Domain** dropdown list. If you do not have an AD domain, select **Local** instead.
4. Click the **Log In** button. If you have Duo two-factor authentication, this appears:



Duo Security

Approve Login Request

A notification has been sent to your device, please respond to continue.

Having trouble? [Login another way](#)

Your cell phone receives a notification you have to approve to access Secret Server.

Note: Secret Server also supports other two-factor authentication methods (depending on what your organization configured), such as text or email codes that Secret Server prompts you for.

Note: After you log on with your local account for the first time, you are immediately prompted to change your password .

5. Click the **Login** button. The Secret Server Dashboard appears.

Secrets are individually named packets of sensitive information, such as passwords. Secrets address a broad spectrum of secure data, each type represented and created by a *secret template* that defines the parameters of all secrets based on it. Secrets are very powerful and provide many ways of controlling and protecting their data, such as:

- Ensuring passwords are long, complex, and frequently changed.
- Relieving users of having to remember numerous complex passwords or when to change them. You only need to remember your password to access Secret Server. All of your secret passwords are managed for you.
- Automatically changing passwords at set intervals with no user intervention.
- Defining who has access to the secret.
- Ensuring the person accessing Secret Server or a secret is indeed you.
- Recording who actually accessed a secret.

All secret text-entry field information is securely encrypted before being stored in the database, including a detailed audit trail for access and history.

Some important basic information about secrets:

- [Viewing Secrets](#) (includes checking expiration and history)
- [Creating Secrets](#)
- [Secret Configuration Options](#)
- [Editing Secrets](#) (includes manually changing passwords, instead of waiting for expiration)
- [Deleting and Undeleting Secrets](#)

Secret folders allow you to create containers of secrets based on your needs. They help organize your customers, computers, regions, and branch offices, to name a few. Folders can be nested within other folders to create sub-categories for each set of classifications. Secrets can be assigned to these folders and sub-folders. Folders allow you to customize permissions at the folder level, and all secrets within can inherit the folder's permissions. Setting permissions at the folder level ensures future secrets placed in that folder have the same permissions, simplifying management across users and groups.

- [Creating Folders](#)
- [Adding and Moving Secrets Between Folders](#)

Please set up Web Password Filler (WPF) in the following order:

1. Ensure you can log in to Secret Server the conventional way.
2. If necessary, create a folder in Secret Server where the WPF secrets will reside.
3. [Install the WPF browser extension](#).
4. [Configure WPF to point to Secret Server](#).
5. [Login to Secret Server via WPF] ([https://docs.thycotic.com/wpf/2.0.0/getting-started/login-Secret Server.md](https://docs.thycotic.com/wpf/2.0.0/getting-started/login-Secret%20Server.md)).

The Secret Server *check-out* feature grants exclusive access to a single user. If a secret is configured for check out, a user can then access it. No other user can access a secret while it is checked out, except unlimited administrators. This guarantees that if the remote machine is accessed using the secret, the user who had it checked out was the only one with proper credentials at that time. See [Secret Checkout](#) for details.

Secret Server records specific events, including expired secrets, and optionally sends you alerts when they happen. See the [Inbox](#) and [Creating Event Subscriptions](#) for details.

We created a [Getting Started Tutorial](#) for technical users. While it covers many things you do *not* need to know right now, you may later find it helpful if you want to get a deeper understanding of Secret Server.

Secret Templates

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret templates are used to create secrets and allow customization of the format and content of secrets to meet company needs and standards. Examples include: Local Administrator Account, SQL Server Account, Oracle Account, Credit Card and Web Password. Templates can contain passwords, usernames, notes, uploaded files, and drop-down list values. New Secret templates can be created, and all existing templates can be modified.

A custom password-exclusion dictionary is a list of words that you do not want users to choose as part of a password, for example, your company name. The dictionary becomes an option when creating or editing a password requirement object. Those, in turn, appear as options when creating a secret template. Finally, when a secret is created based on that template, the words in the dictionary are not allowed when creating a password (the "weak" warning appears).

Creating a Custom Dictionary

To create a new custom password-exclusion dictionary for use by secret templates:

1. Create a text file containing the words you want to exclude, one word per line. These words cannot be used as part of a password on applicable secrets.
 2. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:
-

Manage Secret Templates

Active Directory Account

Show Inactive

Back

Edit

+ Create New

Export

View Audit

Active Templates

* Password Requirements

A Character Sets

Configure Launchers

Configure Secret Template Permissions

Other Templates

Configure Dependency Templates

Configure Scan Templates

Import Secret Templates

Please paste your XML from the online [Secret Templates Gallery](#) into the box below to add your new Secret Template.

Import

3. Click the **Password Requirements** button. The Password Requirements page appears:

Password Requirements

NAME	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	DEFAULT
Default	The default password requirement, which uses the alpha-numeric character set and requires one lowercase, one uppercase, one number, and one symbol.	12	12	Yes
SAP	SAP Password Requirement	12	12	No
Mainframe	Mainframe Password Requirement	8	8	No
FSQA		3	10	No


[← Back](#)
[+ Create New](#)
[Edit Custom Dictionaries](#)

4. Click the **Edit Custom Dictionaries** button. The Password Dictionaries page appears:

Admin > Secret Templates > Password Requirements > Password Dictionaries

+
WS

[Add Password Dictionary](#)



No Items Found

There is no information available to display on this screen.

5. Click the **Add Password Dictionary** button. The Add Dictionary popup page appears:

Add Dictionary

Name

File [Change](#)

Cancel
Upload

6. Type the name of the dictionary in the **Name** text box.
7. Click the **Change** link to locate your dictionary text file. The name of the file appears on the popup.
8. Click the **Upload** button. The popup disappears, and the file appears on the Password Dictionaries page:

Admin > **Secret Templates** > **Password Requirements** > **Password Dictionaries** 🔍 🏠 + WS

Add Password Dictionary

1 item

Test	Upload Updated Version	Download	Delete
------	--	--	--

9. Now, when defining a password requirement, the custom dictionary you created ("test") appears as a prevention option:

PASSWORD VALIDATION

- Prevent Username In Password
- Prevent Common Dictionary Words
- Prevent Dictionary Words from Dictionary 'Test'
- Prevent Spatial Terms In Password
- Prevent Sequences In Password

When a user attempts to include one of the excluded words in the dictionary in a secret based on the template using the password requirement, the "weak" warning appears and the user cannot save the password. For example, our dictionary contains the word (string) xxyy. The user enters a strong password that contains the string, and Secret Server rejects it anyway:

Note: The excluded words are not case sensitive. xxyy would have triggered a password rejection too.

Secret Template: Web Password Copy

Secret Name: My Secret

URL:

UserName:

Password: 0329!@#50IUyJwlrjufdopisufxxyy Weak ⚠ Generate

Notes:

When you hover the mouse pointer over the password strength bar, the disallowed string appears in red:

Password should include:

- ✓ At least 12 characters.
- ✓ At least 1 Lower case letters (a-z)
abcdefghijklmnopqrstuvwxyz
- ✓ At least 1 Symbols (Symbols)
!@#%*&*()
- ✓ At least 1 Numbers (0-9)
1234567890
- ✓ At least 1 Upper case letters (A-Z)
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Password should exclude:

- Dictionary words
Password includes: 'xxyy'
- ✓ Username

Editing a Custom Password-Exclusion Dictionary

To edit a custom password-exclusion dictionary for use by secret templates:

1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:

Manage Secret Templates

Active Directory Account

Show Inactive

 Back

 Edit

 Create New

 Export

 View Audit

Active Templates

 Password Requirements

 Character Sets

 Configure Launchers

 Configure Secret Template Permissions

Other Templates

 Configure Dependency Templates

 Configure Scan Templates

Import Secret Templates

Please paste your XML from the online [Secret Templates Gallery](#) into the box below to add your new Secret Template.

 Import

2. Click the **Password Requirements** button. The Password Requirements page appears:

Password Requirements


NAME	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	DEFAULT
Default	The default password requirement, which uses the alpha-numeric character set and requires one lowercase, one uppercase, one number, and one symbol.	12	12	Yes
SAP	SAP Password Requirement	12	12	No
Mainframe	Mainframe Password Requirement	8	8	No
FSQA		3	10	No

[← Back](#)
[+ Create New](#)
[Edit Custom Dictionaries](#)

3. Click the **Edit Custom Dictionaries** button. The Password Dictionaries page appears:

Admin > Secret Templates > Password Requirements > Password Dictionaries

[Add Password Dictionary](#)


No Items Found
 There is no information available to display on this screen.

4. Click the **Add Password Dictionary** button. The Add Dictionary popup page appears:

Admin > Secret Templates > Password Requirements > Password Dictionaries

[Add Password Dictionary](#)

1 item

Test	Upload Updated Version	Download	Delete
------	--	--------------------------	------------------------

5. Click the **Download** link for the desired dictionary.

6. Save the file to your computer.
7. Edit the text file as desired. Do not change the name of the file.
8. Click the **Upload Update Version** link to locate and upload your dictionary text file. The existing dictionary, of the same name, is overwritten.

To use the privileged password security policy template, follow the steps below:

1. Download the privileged password security policy [template](#).
2. Open the template as a Microsoft Word document.
3. Remove the "About this Template" and "Customizing the Template" instructions and other author comments.
4. Replace the term "Company X" with the name of your organization.
5. Replace the current logo or add your company logo in the upper left corner.
6. Update all of the company-specific contact information (highlighted yellow).
7. Update the effective date.
8. Revise any policy guidelines to meet your organization's policies.
9. Revise the Violations section to meet your organization's policies.
10. Save your changes.
11. Obtain your management and auditors' approval of the completed policy.
12. Distribute the policy according to your management guidance.

Secret Server includes many pre-configured secret templates.

Built-in Secret Templates Available Out-of-the-box

- Active Directory Account
- Amazon IAM Console Password
- Amazon IAM Key
- Bank Account
- Cisco Account (SSH)
- Cisco Account (Telnet)
- Cisco Enable Secret (SSH)
- Cisco Enable Secret (Telnet)
- Cisco VPN Connection
- Combination Lock
- Contact
- Credit Card
- DevOps Secrets Vault Client Credentials
- Generic Discovery Credentials
- Healthcare
- HP iLO Account (SSH)
- IBM iSeries Mainframe
- MySql Account
- Office365 Account
- OpenLDAP Account
- Oracle Account
- Password
- Pin
- Product License Key
- SAP Account
- SAP SNC Account
- Security Alarm Code
- Social Security Number
- SonicWall NSA Admin Account
- SonicWall NSA Local User Account
- SonicWall NSA Web Admin Account
- SonicWall NSA Web Local User Account
- SQL Server Account
- SSH Key
- Sybase Account
- Unix Account (Privileged Account SSH Key Rotation - No Password)
- Unix Account (Privileged Account SSH Key Rotation)
- Unix Account (SSH Key Rotation - No Password)
- Unix Account (SSH Key Rotation)
- Unix Account (SSH)
- Unix Account (Telnet)
- Unix Root Account (SSH)
- Update Secret From Script Template
- VMware ESX/ESXi
- WatchGuard
- Web Password

- Web Password with TOTP
- Windows Account
- Windows LiveAccount
- z/OS Mainframe

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Activating and Deactivating Templates

If a template is no longer relevant or outdated, it can be inactivated. This can be done from the specific template's Secret Template Edit page.

1. Go to **Admin > Secret Templates**.
2. Click the template name in the **Secret Templates** column. That template's page appears on the General tab.
3. Click **Edit** next to Template Status. The Secret Template Detail Status section enters edit mode:

Template Status

Indicates how many Secrets use this template and whether or not the template is active.

Template Usage	0
Active	<input checked="" type="checkbox"/>

Cancel
Save

Template Settings

Configure settings for this template.

Secret Template Name *	My Secret Template
Name Pattern	None
Description	None
All History	No
Secret Name History Length *	0
Validate Password Requirements On Create	No
Validate Password Requirements On Edit	No

4. To deactivate a template, uncheck the **Active** checkbox.
5. To activate a template, check the **Active** checkbox.

Templates can also be inactivated in bulk from the Manage Secret Templates page. Click the **Active Templates** button to navigate to the Set Active Secret Templates page. This screen displays all the secret templates in Secret Server. Each secret template can be set as active or inactive. Once the secret templates are chosen as active or inactive, then saving changes brings the secret templates into effect immediately. Inactivating a secret template does not inactivate any secrets using that secret template—those secrets still exist, but users are not able to create new secrets using an inactivated secret template.

Changing a Secret's Template

To convert secrets from one secret template to another:

1. View a secret and click on the **Convert Template** button.
2. Click to select the target template from the **Secret Template** list.
3. Map each text-entry field to a new field:
 1. Go through each list and select the target text-entry field for each source text-entry field on your secret.
 2. If you want to remove the value for a text-entry field instead of converting it, then select the <Remove> option on the list for that text-entry field.
 3. When you are done selecting, you can choose a folder.
4. Click **Save**.

The Convert Template button is only available to users and groups with the "Owner" permission to the secret.

Note: To preserve audit data, when a secret is converted from one type to another, the old secret is deleted, and a new secret is created. An admin can view old secret by searching for deleted secrets on the dashboard. A user needs "Add Secret," "Edit Secret," "Delete Secret," and "Own Secret" role permissions in order to convert a secret to a new template.

Configuring Secret Template Permissions

You can assign users and groups to specific secret templates so they can either manage or create secrets based on those templates. This allows you to have more granular control over what secret templates are seen by users and groups when they are managing the templates or creating secrets. To configure permissions:

1. Select **Admin > Secret Templates**. The Secret Templates page appears:

SECRET TEMPLATES ↑	TOTAL SECRETS	ACTIVE
[GTS] Unix Account - SUDO ...	0	<input checked="" type="checkbox"/>
1-Bug AD Copy	1	<input checked="" type="checkbox"/>
A Small Template	1	<input checked="" type="checkbox"/>
Active Directory Account	447	<input checked="" type="checkbox"/>
Active Directory Account (C...	4	<input checked="" type="checkbox"/>
Active Directory Account RD...	6	<input checked="" type="checkbox"/>
Active Directory Account RD...	2	<input checked="" type="checkbox"/>

2. Click the name of the template for which you wish to configure permissions. The Secret Template Detail page appears on the General tab:

Active Directory Account

General Fields Mapping Permissions Audit Export Duplicate

Template Status

Indicates how many Secrets use this template and whether or not the template is active. Edit

Secret Templates [↗](#)

Template Usage	447
Active	Yes

Template Settings

Configure settings for this template. Edit

3. Click the Permissions tab:

Active Directory Account

General Fields Mapping Permissions Audit Export Duplicate

Secret Template Permissions

Configure any specific permissions to allow creating specific types of Secrets or owning and editing the template definition. Edit

0 Items

4. Click the **Edit** button in the **Secret Template Permissions** section. The Secret Template Permissions section enters edit mode:

Secret Template Permissions

Configure any specific permissions to allow creating specific types of Secrets or owning and editing the template definition.

Add

Add users or groups by selecting the add button. This will define who can create Secrets of this type or manage this Secret template.

Cancel **Save**

5. To change the permissions of a user or group that has already been assigned permissions, select a new permission from the **Administrator** drop-down list. Template Create Secret allows a user or group to create secrets based on the selected secret template. Template Owner allows a user or group to edit a secret template and create secrets based on the selected secret template. By default, the Everyone group that targets all users of Secret Server can create secrets based on any secret template.

Note: User's secret template permissions are based on the permissions directly assigned to them, as well as the permissions assigned to all of the groups the user belongs to. If a user or group does not have Template Create secret or Template Owner permissions, they are unable to create a secret based on that secret template or see that it exists in Secret Server.

6. To add permissions for a new user or group, click the **Add** button. The Users popup appears:

Users

Search...

All ▾ Users & Groups ▾

0 selected

- #symbolgroup (testparent.thycotic.com)
- (lol) (testparent.thycotic.com)
- /groups
- \leading spaces (testparent.thycotic.com)
- _test
- a1
- a10
- a11

Cancel **Add**

7. Click to select the check box for the user or group.
8. Click the **Add** button.

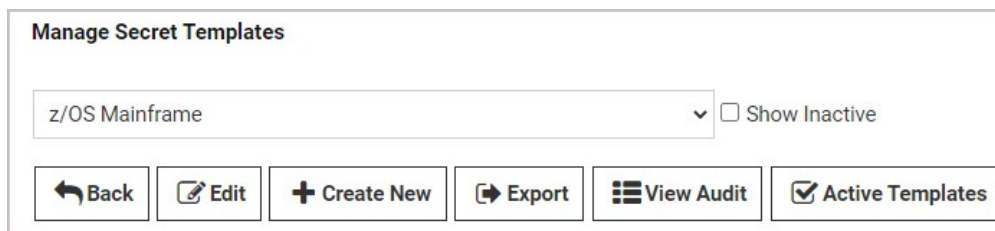
Create and Customize an IBM iSystem (AS/400) Template to use the new IBM iSeries (AS/400) Password Changer

The IBM iSeries (AS/400) Terminal password changer is based on the z/OS Mainframe password changer. It uses the 5250 terminal connection and scripting to perform the password change and heartbeat. You can modify the script for any advanced configuration requirements, and Delinea Professional Services is available to help you.

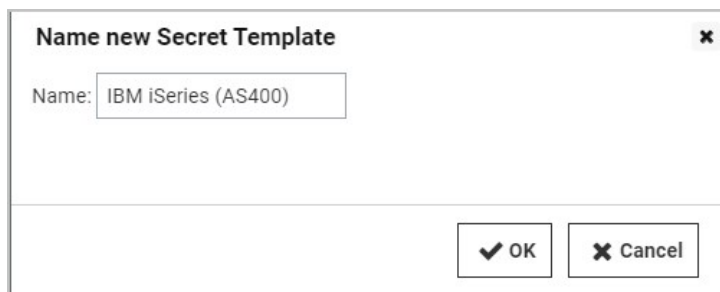
Note: You can also change passwords on the AS/400 using SSH. See [Creating a Custom Password Changer for IBM AS/400](#).

Create an AS/400 Secret Template

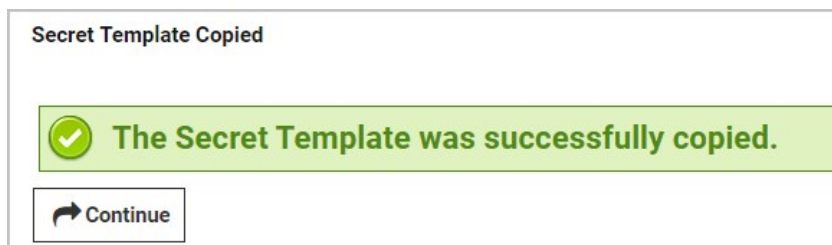
1. Navigate to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, select the **z/OS Mainframe** template from the drop-down list.
3. Click the **Edit** button.




4. On the **Secret Template Designer** page, click the **Copy Secret Template** button.
5. On the popup page, type IBM iSeries (AS400) in the **Name** text box.



6. Click the **OK** button.
7. On the confirmation page, click the **Continue** button.



Optional: on the **Secret Template Designer** page, you can deactivate the **Passphrase** field by clicking the deactivate icon  to the right of the **Passphrase** row. Unlike the z/OS, the iSeries does not need an additional passphrase and will not have an option for it unless adjusted. Unless your environment specifically requires the passphrase text-entry field, we recommend deactivating it.

Modify Your AS/400 Secret Template to use the AS/400 Password Changer

1. On the **Secret Template Designer** page, click the **Configure Password Changing** button.
2. On the **Secret Template Edit Password Changing** page, click the **Edit** button. The page becomes editable.

Secret Template Edit Password Changing

Enable Remote Password Changing	Yes
Retry Interval	2 hours
Maximum Attempts	12
Enable Heartbeat	Yes
Heartbeat Check Interval	1 day

Password Type to use z/OS Mainframe

PASSWORD TYPE	SECRET FIELD	SCRIPT VARIABLE
Machine Name	Machine	\$machine
Passphrase	Passphrase	\$passphrase
Password	Password	\$password
Port	Port	\$port
User Name	Username	\$username

[← Back](#) [✎ Edit](#)

3. Next to **Password Type to Use**, click the drop-down list and select **IBM iSeries Mainframe**.

Secret Template Edit Password Changing

Enable Remote Password Changing

Retry Interval

Days

Hours

Minutes

Maximum Attempts

Enable Heartbeat

Heartbeat Check Interval

Days

Hours

Minutes

Password Type to use

Machine Name

Password

Port

User Name

4. Make required changes, if any, to the text boxes and lists.
5. Click the **Save** button. The page is no longer editable.
6. Click the **Back** button.
7. On the **Secret Template Designer** page, create secrets based on the new template as desired.

Customize Your AS/400 Password Changer for Your Environment

Note: For the default IBM iSeries (AS/400) systems, the default password changer configuration requires no adjustment. However, additional parameters and connection string options are available.

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. On the **Password Changer Configuration** page, click the **IBM iSeries Mainframe** link.
4. On the **IBM iSeries Mainframe** page, scroll to the bottom and click **Edit**.
5. On the **Edit Password Changer** page, adjust ports and other parameters as desired.

Edit Password Changer

Name *

Line Ending

Custom Port (e.g. override the default value of 22 for SSH or 23 for Telnet with another value)

Request Terminal (If checked, the standard out and standard error data streams combine for \$\$CHECK* commands, else \$\$CHECK* will only check standard out and standard error will cause an error)

Connection String

Use SSL

Active

Valid for Discovery Import

6. Click the **Save** button.

Note: The *trace* function can be a powerful tool for troubleshooting and debugging, especially for complex RPC implementations in unique environments. The trace function logs emulator input, mainframe output, and ASCII screenshots of what is happening on the terminal GUI. To write a trace file to the Secret Server website or engine, just add TRACE to the connection string.

Additional Functions, Adjustments, and Parameters

For unique IBM iSeries environments, the IBM iSeries password changer offers extra features, options, adjustments and parameters for customization, including the commands in the table below. To implement these commands successfully, it helps to keep in mind that the password changer is emulating user input. Some of these commands are designed for very fine emulations of unique IBM iSeries environments, and Delinea Professional Services can help you with these. Other commands are implemented and tested on a base environment, so before implementing them in a production environment, you should verify that they are working as expected through testing or by using the trace function.

Note: The commands below are followed by an <ENTER> command by default. To prevent this, you must add ##NOENTER in the *comment* of the previous command. For example:

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	\$USERNAME	Username ##NOENTER	2000
2	<TAB>	Tab to password	2000
3	\$CURRENTPASSWORD	Password	2000

Table: Secret Template Commands

< Backtab >	Tab to the previous input field.
-------------	----------------------------------

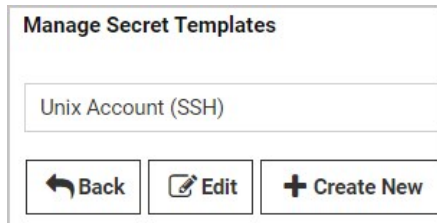
<Clear>	Clear the screen.	Mostly used for trace.
<Close>	End the session to the mainframe.	
<Delete>	Delete a character under the cursor; can be used with <MoveCursor(#, #)>	
<DeleteField>	Delete the entire text input or field.	
<DeleteWord>	Delete the current word if available, otherwise delete the previous word.	
<Disconnect>	Disconnect the password changer's connection to the mainframe.	
<Down>	Move cursor down.	
<Enter>	Send the Enter key press command.	
<Erase>	Erase previous character on a selected text input.	<Erase>
<EraseEOF>	Erase end-of-field of current text input.	<EraseEOF>
<Execute()>	Execute commands in shell.	<Execute(USRMGR)>
<HexString(#)>	Insert a control character in a text field or string.	<HexString(41)>
<Key(#)>	Execute named iSeries keys.	Execute unique keys via hex, character code, or key symbol.
<Left>	Move cursor left.	
<PF(#)>	Execute program function.	Program function keys 1 to 24
<PA(#)>	Execute program attention.	Program attention functions 1 to 3
<MoveCursor(#, #)>	Move the cursor by row and column.	<MoveCursor(10,2)>
<Right>	Move cursor right.	
<Tab>	Tab to the next line.	
<Up>	Move cursor up.	

Creating a Unix Account Secret Template that Uses Key Authentication Instead of a Password

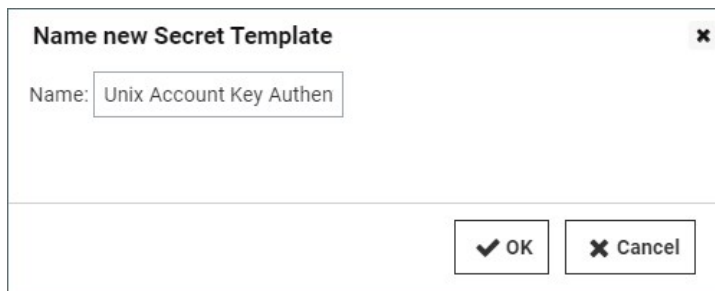
To create a Unix account secret template that uses key authentication only instead of a password, begin by using an existing **Unix Account (SSH)** template as a baseline.

Create the New Template

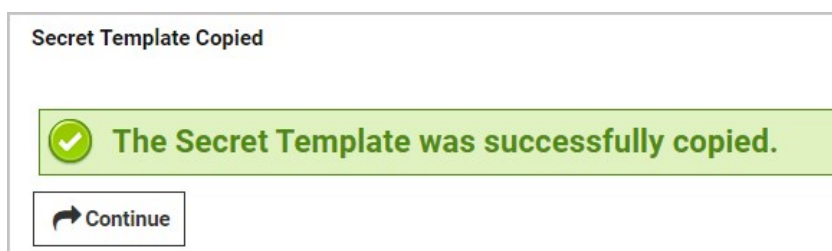
1. Go to **Admin > Secret Templates**.
2. Select the built-in **Unix Account (SSH)** template from the drop-down menu and click **Edit**.



3. On the **Secret Template Designer** page, scroll to the bottom and click **Copy Secret Template**.
4. Give the new template an appropriate name, such as *Unix Account Without Password (SSH)* or *Unix Account Key Authentication Only (SSH)*.



5. Click **OK**.
6. On the **Secret Template Copied** confirmation page, click **Continue**.



7. On the **Secret Template Designer** page, scroll down to the **Fields** section and under **Field Name**, find the **Password** row.

FIELDS
FIELD NAME
Machine
Username
Password

- At the right end of the **Password** row, click the **Edit this field** icon 
- In the **Password** row under **IS REQUIRED**, uncheck the box. Optionally, you can also select **Not Editable** from the **Edit Requires** drop-down list.

IS REQUIRED
<input type="checkbox"/>

- At the right end of the **Password** field, click the **Save this field** icon 

You now have a Unix account (SSH) Secret template that displays key authentication fields instead of a password field.

Disable

Your new template has inherited characteristics from the **Unix Account (SSH)** template you based it on, including having **Remote Password Changing** and **Heartbeat** enabled by default. But because your new template has no password, it cannot be remotely changed and heartbeat cannot validate on an empty password. Therefore, you must disable these features by editing your new template using the procedure below:



- In the **Secret Template Designer** window, scroll to the bottom and click **Configure Password Changing**.
- In the **Secret Template Edit Password Changing** window, click **Edit**.

Secret Template Edit Password Changing

Enable Remote Password Changing Yes
 Retry Interval 1 hour
 Maximum Attempts 10000
 Enable Heartbeat Yes
 Heartbeat Check Interval 8 hours

Password Type to use Active Directory Account

PASSWORD TYPE	SECRET FIELD	SCRIPT VARIABLE
Domain	Domain	\$domain
Password	Password	\$password
User Name	Username	\$username
Domain Controller (DC)		\$domaincontroller
Default Privileged Account		< None >



 

3. In the next **Secret Template Edit Password Changing** window:

1. Uncheck **Enable Remote Password Changing**.
2. Uncheck **Enable Heartbeat**.

Secret Template Edit Password Changing

Enable Remote Password Changing
 Enable Heartbeat

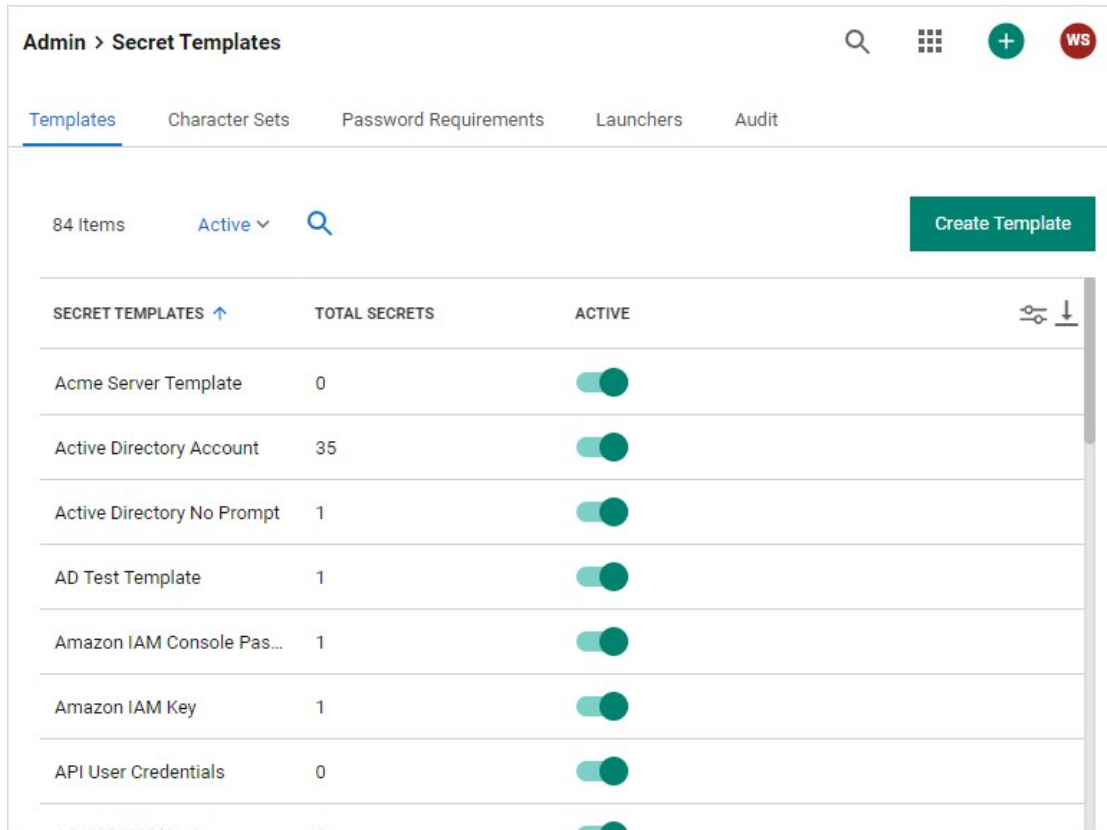
4. Click **Save**.

Note Some Secrets based on the **Unix Account (SSH)** might display the **Password** field as well as the **Private Key** and **Private Key Passphrase** (key authentication) fields. If a user signs in using this template with correct credentials in the key authentication fields but a blank or incorrect password in the Password field, the default PuTTY launcher will use key authentication to connect.

Creating or Editing Secret Templates

Editing Secret Templates

1. Select **Admin > Secret Templates**. The Secret Templates page appears:



2. Click the template name in the Secret Templates column. That template's page appears:

Admin > Secret Templates > Acme Server Template

General Fields Mapping Permissions Audit

Export Duplicate

Template Status [Edit](#)

Indicates how many Secrets use this template and whether or not the template is active.

[Secret Templates](#) [↗](#)

Template Usage	0
Active	Yes

Template Settings [Edit](#)

Configure settings for this template.

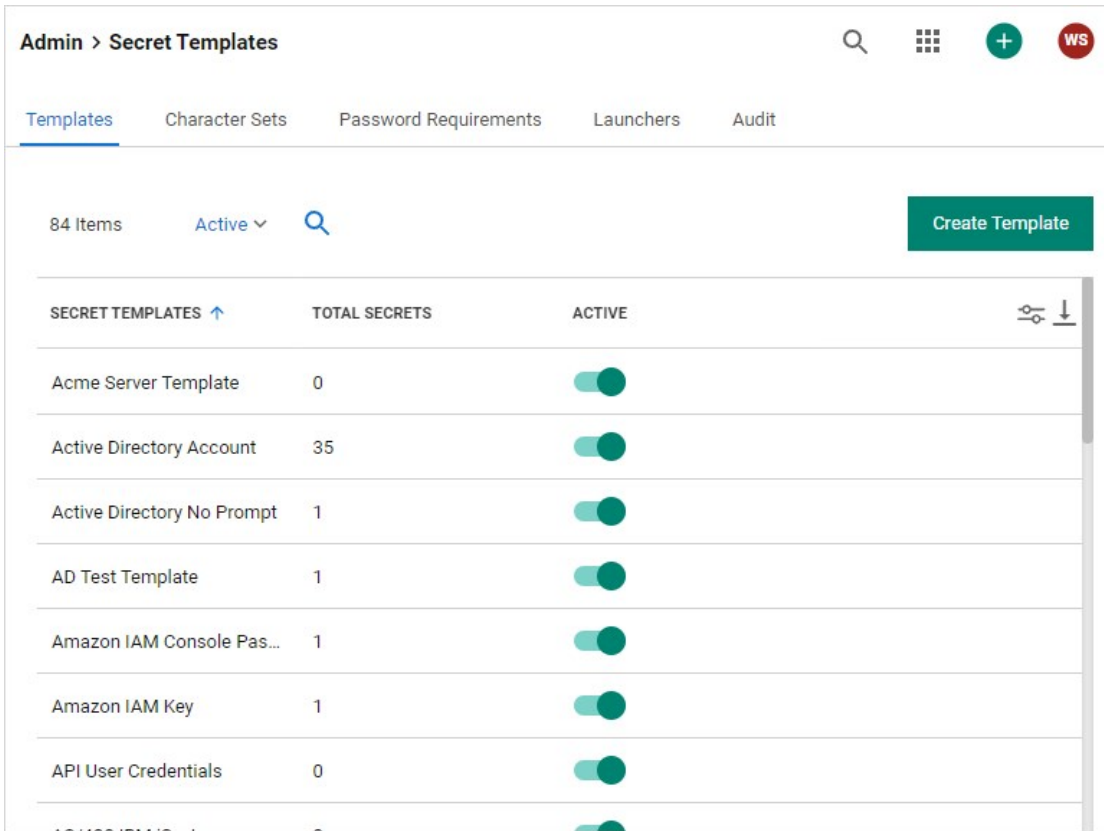
Secret Template Name *	Acme Server Template
Name Pattern	None

3. Click the desired tab for the configuration you want to change. See the Creating or Importing a New Template section for details.

Creating or Importing a New Template

Task 1: Creating the Template

1. Select **Admin > Secret Templates**. The Secret Templates page appears:



1. Click the **Create Template** button. The Create Template pop-up page appears:

The 'Create Template' form contains the following elements:

- Title:** Create Template
- Subtitle:** Name your new Secret Template, or import one from an XML file.
- Radio Buttons:** 'New / Import *' with 'New' selected (indicated by a blue box) and 'Import XML'.
- Text Box:** 'Template Name *' with an empty input field.
- Buttons:** 'Cancel' and 'Save'.

2. If importing the template, click to select the **Import XML** selection button.
3. Type the name of the new template in the **Template Name** text box.
4. Click the **Save** button. The new template's setup page on the General tab appears:

Admin > Secret Templates > My Secret Template

General Fields Mapping Permissions Audit

Export Duplicate

Template Status [Edit](#)

Indicates how many Secrets use this template and whether or not the template is active.

[Secret Templates](#) [↗](#)

Template Usage	0
Active	Yes

Template Settings [Edit](#)

Configure settings for this template.

Secret Template Name *	My Secret Template
Name Pattern	None
Description	None
All History	No
Secret Name History Length *	0
Validate Password Requirements On Create	No
Validate Password Requirements On Edit	No

The page provides all the options for configuring a secret template, as well as which text-entry fields appear on any secret created from that template.

Task 2: Adding General Settings and Setting an Expiration or One-Time Password

1. On the **General** tab, click the **Edit** link in the **Template Settings** section. The section becomes editable:

The screenshot shows a configuration form for a secret template. It includes the following fields and controls:

- Secret Template Name ***: Text box containing "My Secret Template".
- Name Pattern**: Empty text box.
- Description**: Empty text box.
- All History**: Check box, currently unchecked.
- Secret Name History Length ***: Text box containing "0".
- Validate Password Requirements On Create**: Check box, currently unchecked.
- Validate Password Requirements On Edit**: Check box, currently unchecked.
- Buttons**: "Cancel" and "Save" buttons at the bottom right.

2. Edit the setting as desired. They include:

- **Secret Template Name** text box.
- **Name Pattern** text box. See [Template Naming Patterns](#).
- **Description**: An optional description for the template.
- **All History** check box: If this check box is enabled, Secret Server keeps all entries for viewing. This feature creates a record of every name used when a new secret is created.
- **Secret name History Length** text box: If All History is disabled, Secret Server keeps this number of entries for viewing.
- **Validate Password Requirements on Create?** check box: Ensure requirements are met on secret creation.
- **Validate Password Requirements on Edit?** check box: Ensure requirements are met when editing secret.

3. Click the **Save** button.

4. Click the **Edit** link for the **Template Expiration** section:

The screenshot shows the "Template Expiration" configuration section. It includes the following elements:

- Section Header**: "Template Expiration".
- Description**: "Secret Expiration applies to one field of a Secret Template (most commonly the password field) and may trigger a password change for that Secret if Auto-Change is configured for Remote Password Changing."
- Expiration Enabled**: Check box, currently unchecked.
- Buttons**: "Cancel" and "Save" buttons at the bottom right.

Secret expiration applies to one field of a secret template (most commonly the password field) and may trigger a password change for that secret if auto-change is enabled for RPC.

5. Click to select the **Expiration Enabled?** check box. New controls appear:

Expiration Enabled

Days until Expiration *

Change Required On *

6. Type the days till expiration in the **Days until Expiration** text box.
7. Click the Change Required On dropdown list to select the field to choose the field the expiration is applied to.
8. Click the **Save** button.
9. Secret Server begins providing alerts if the secret text-entry field is not changed within the specified expiration requirements.
10. Click the **Edit** link for the **One Time Password** section if you want the secret to have a one-time password that the user must change:

One Time Password

Configure this secret for one time password. **One Time Password Enabled**

11. Click to select the **One Time Password Enabled** check box. Additional controls appear:

One Time Password Enabled

One Time Password Length

One Time Password Duration

One Time Password Hash

12. Type or select the options.
13. Click the **Save** button.

Task 3: Defining Fields for the Template

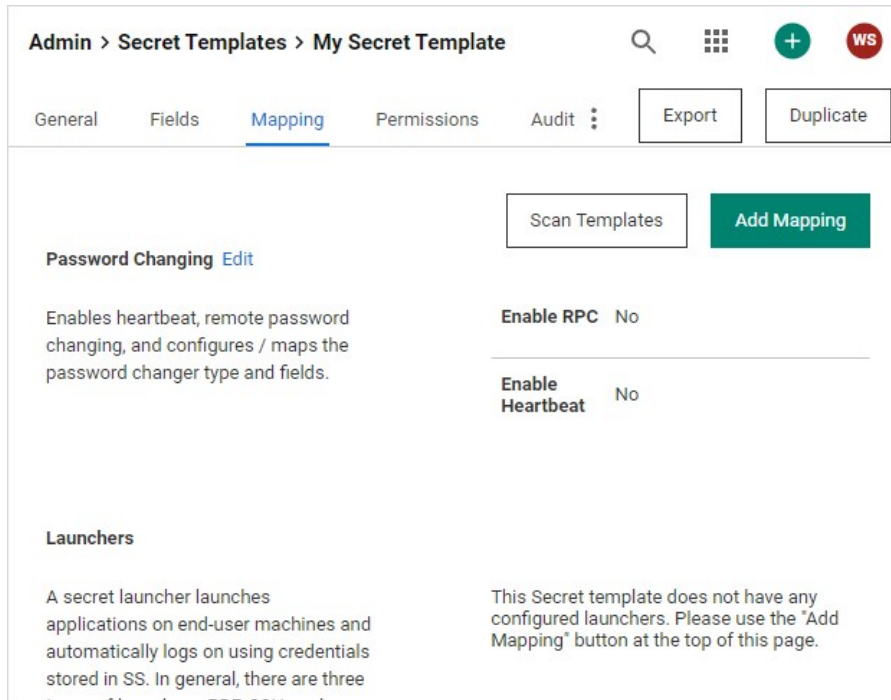
1. Click the **Fields** tab to add template fields as desired. See [Secret Template Fields](#).

Note: To use a custom SSHRPC port, add a field named "Port" to your secret template. Empty port fields are equivalent to

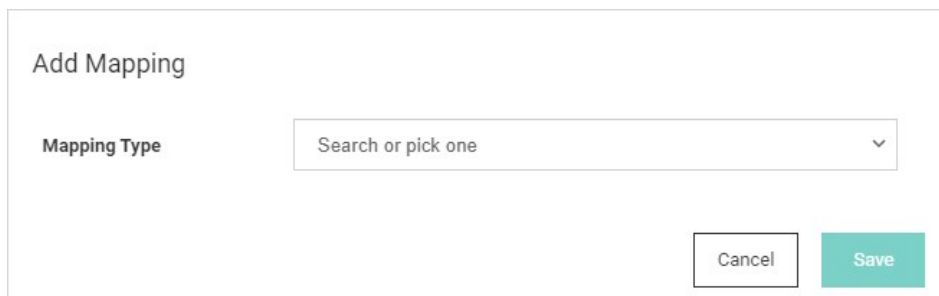
the default port, 22.

Task 4: Mapping Launchers and RPC Type

1. Click the **Mapping** tab to configure launchers and RPC:



2. Click the Edit link in the **Password Changing** section to enable RPC on secrets based on this template. This enables heartbeat, RPC, and configures the password changer type and fields. For details, see [Remote Password Changing](#).
3. Click the **Add Mapping** button to add a secret launcher or extended mapping. The Add Mapping popup appears:



4. Click the **Mapping Type** combination list to search for or select a mapping type:

Launchers:

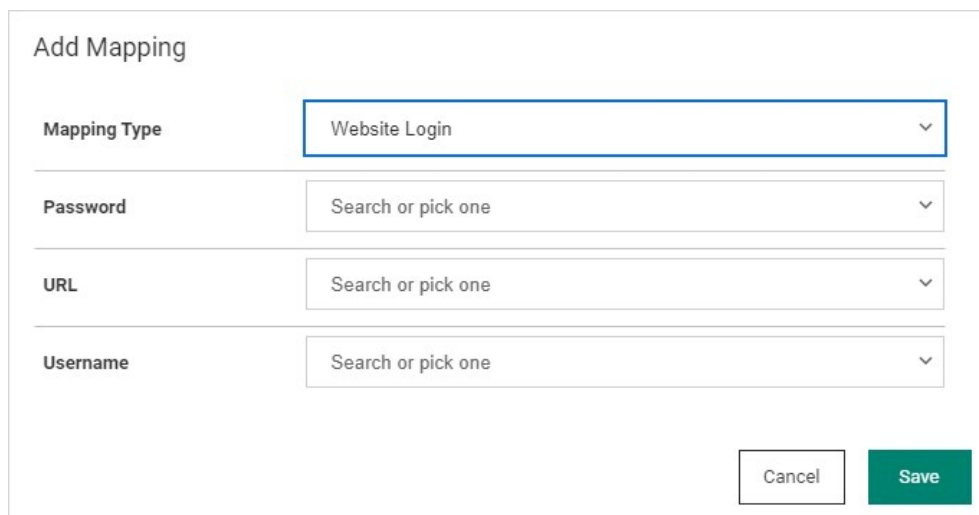
- Batch Launcher
- Custom PowerShell Launcher
- IBM iSeries Launcher
- Notepad
- PowerShell Launcher

- PuTTY
- PuTTY With Port Prompt
- Remote Desktop
- Secure CRT (Proxied)
- Secure CRT Proxied Process
- Secure CRT Proxied with Van Dyke File Path
- SQL Server Launcher
- Sybase iSQL Launcher
- Ultra VNC

Website Logins:

- OATH Secret Key
- Regex List
- Remote Server SSH Key for Validation
- SSH Private Key
- Username and Password
- z/OS Launcher

The popup changes to accommodate your choice. For example, for a Web server:



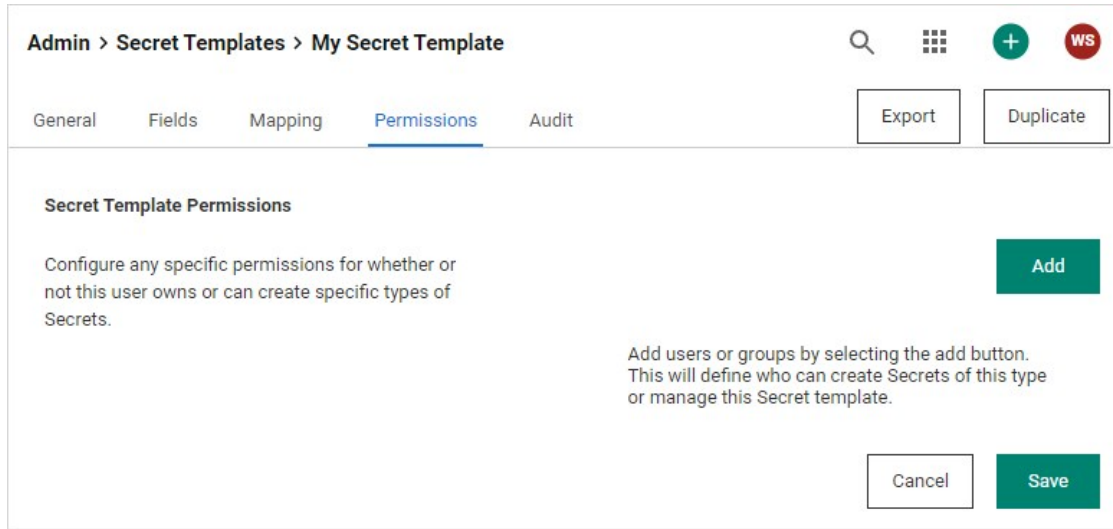
The screenshot shows a dialog box titled "Add Mapping". It contains four rows of input fields, each with a label on the left and a search or selection box on the right. The first row is "Mapping Type" with a dropdown menu showing "Website Login". The second row is "Password" with a search box containing "Search or pick one". The third row is "URL" with a search box containing "Search or pick one". The fourth row is "Username" with a search box containing "Search or pick one". At the bottom right of the dialog are two buttons: "Cancel" and "Save".

Note: A secret launcher launches applications on other machines and automatically logs on using credentials stored in Secret Server. In general, there are three types of launchers: RDP, SSH, and Custom. In addition to user convenience, launchers can circumvent users needing to know their passwords—a user can still gain access to a needed machine but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser.

5. Click to select or type to search the desired dropdown lists.
6. Click the **Save** button.

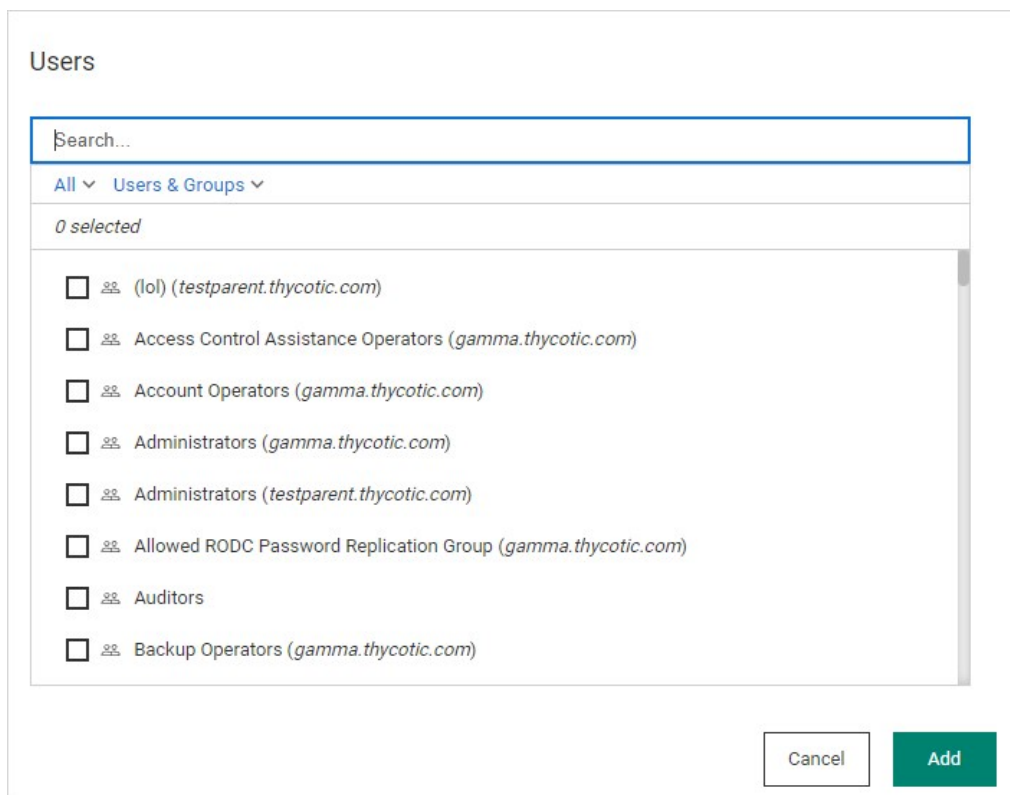
Task 5: Adding Permissions

1. Click the **Permissions** tab:



This defines who can create secrets of this type or manage this secret template.

2. Click the **Add** button. The Users popup appears:



3. Type the name of the user or group you want to add in the **Search** text box. Note that the groups are by domain.
4. Click to select the user or group's check box for those you desire.

5. Click the **Add** button. The selected users or groups appear on the Permissions tab:

6. Click the dropdown list next to each to define if the user or group has the Template Create Secret or Template Owner permission. More than one owner is allowed.

7. Click the **Save** button the users or groups now appear in a small table, along with their roles (permissions).

8. To remove a user or group:

1. Click the **Edit** link for the **Secret Template Permissions** section. The table of users and groups disappears, and the dropdown lists reappear.
2. Click the dropdown list for the user or group you want to delete and select **< None >**.
3. Click the **Save** button

Task 6: Viewing the Template's Audit Trail

1. Click the **Audit** tab to view activity on the secret template:

DATE ↓	DISPLAY NAME	ACTION	NOTES	IP ADDRESS	⚙️ ↓
11/18/2021 01:33 PM		OWNERS_MODIFI...	(Revoked access)...	172.25.115.98	
11/18/2021 01:30 PM		OWNERS_MODIFI...	(Revoked access)...	172.25.115.98	
11/18/2021 01:27 PM		OWNERS_MODIFI...	(Granted access) ...	172.25.115.98	
11/17/2021 03:13 PM		Create		172.25.115.101	

You cannot drill down on the entries, but you can define what columns to see by clicking the slider icon on the right. You can also click the download icon to download a text file version of the table:

Download

Would you like to download this table data as a CSV file?

Export	All Data	▼
Date Format	ISO (1994-11-05T13:15:30.939Z)	▼
Time Zone	UTC	▼

Cancel
Download

Settings for Specific Template Types

Oracle Account as SYS

Settings for an Oracle Account secret template to work with Oracle connecting as SYS in SysDBA:

- Set **Oracle Account** as the type.
- Set **Oracle Account (AS SYS)** as the password type.
- Create a secret based on the new template to test the template.

SQL Windows Authentication Account Secret Template and Launcher

Settings for an Active Directory template that is specifically for SQL:

Note: You can copy the existing AD template that you have. However, if you copy an existing template that has launchers attached to it, you may need to delete those launchers on the newly created template.

- Set **Active Directory** as the type.
- If necessary, create a field called **Server**.
- Add the following parameters for Windows settings (see [Creating Custom Launchers](#)):
 - Name: SQL Server Launcher - Windows Authentication
 - Active: Yes
 - Process Arguments: -E -S \$\$Server (\$Server should match the field name you created or observed earlier)
 - Run Process as Secret Credentials: Yes
 - Load User Profile: Yes
 - Use Operating System Shell: No
 - Use Additional Prompt (in General Settings): No

Introduction

The "SAP SNC Account" secret template is an expansion on the original "SAP Account" secret template. It takes advantage of SAP's Secure Network Communication (SNC), which is a protocol that encrypts communication between Secret Server and an SAP Server. The SAP SNC Account template includes all the original fields from the SAP Account secret, adding a few more as well.

New Template Fields

The following is an introduction to the new template fields (in addition to those also found in the SAP Account secret template):

Note: Please see the [SAP .NET Connector 3.0 Programming Guide](#) for additional information.

- **SNC Partner Name:** Matches the snc/identity/as value set in your SAP Server configuration.
- **SNC My Name:** For most SAP configurations, you can ignore this. See the connector programming guide for cases where it may be required.
- **SNC Quality of Service:** Dropdown list to select the service quality or protection used for SNC communication. Choose one of the following protection options:
 - Authentication Integrity (includes authentication)
 - Authentication Integrity Privacy (includes integrity protection and authentication)
 - Authentication Only
 - Default Protection
 - Maximum Protection
- **SNC Single Sign On:** Dropdown list to set to true if you wish to use single sign on. If you set this to false, you authenticate with your username and password on the secret.
- **X.509 Certificate:** Click the **Change** link to upload an X.509 certificate for authentication.

Server-Side Setup

Prerequisites

SAP Server Setup

Follow the latest SAP documentation for configuring the SAP server and your SAP users to use SNC. For example:

1. SSH into your SAP server.
2. Edit the configuration file: `/sapmnt/<SystemID>/profile/profilename.pfl`
3. Add the SNC settings to the end of this file. For example:

```
snc/enable = 1

snc/gssapi_lib = /usr/sap/NPL/SYS/exe/run/libsapcrypto.so

snc/identity/as = p:CN=vhcalnplci,OU=Test,O=Thycotic,C=US

snc/accept_insecure_cplic = 1

snc/accept_insecure_gui = 1

snc/accept_insecure_r3int_rfc = 1
```



```
snc/accept_insecure_rfc = 1
snc/permit_insecure_start = 1
snc/extid_login_diag = 1
snc/extid_login_rfc = 1
snc/data_protection/min = 1
```

4. Verify that the library file path exists on your server (make sure that `libsapcrypto.so` is actually in that directory).
5. Reboot your server.
6. When the server is finished, reconnect and restart the SAP server with these commands:

```
su npladm
startsap all
```

SAP NCO Files

As with the original SAP Account template, you include the `SAPNCO.dll` and `SAPNCO_UTILS.dll` files in your Secret Server or distributed engine installation. See [SAP Heartbeat and Password Changing](#) for more information.

SAP Cryptographic Library

In addition to the SAP NCO DLL files, you need to obtain the SAP Cryptographic Library. This should include the library DLL (`sapcrypto.dll`), the license ticket, and the configuration tool (`sapgenpse.exe`). Add the DLL file to your Secret Server or distributed engine installation following the same steps as the SAP NCO files. For more information on this library, see the [SAP Identity Management Configuration Guide](#).

SAP Server Certificate

1. Open SAP Trust Manager (STRUST).
2. Download your SAP's server certificate from the STRUST transaction. Assuming you setup your SAP server correctly, this should be located in the **SNC SAPCryptolib** folder.
3. If nothing exists under SNC SAPCryptolib, right click on the folder and select **Create** to create a new PSE under **SNC SAPCryptolib**.
4. Open the PSE.
5. Enter a password if prompted. (If not, use the **Password** button to set a password).
6. Click the SNC SAPCryptolib folder.
7. Double click the **Subject** under **Own Certificate**.
8. Confirm the certificate details appear in the **Certificate** section.
9. Click the **Export Certificate** icon button at the bottom to open a dialog box, which allows you to download the certificate.

Note: If the button is not enabled, you may need to click the Display or Edit (pencil and glasses) button and click the Base64 selection button when prompted and then the green checkmark button to complete the download.

Personal Security Environment Setup

As with your SAP server setup, you should consult the latest SAP documentation for more information when setting up your Personal Security Environment (PSE). These instructions are provided to illustrate the options to configure the SAP SNC Account secret template in

Secret Server, but SAP's documentation may provide more information about your options pertaining to the creation of a PSE. To set up your PSE:

1. In your client environment (your Secret Server or distributed engine server), create a directory to stage your setup. For example, I used C:\SAPSNC.
2. Add the two SAP NCO files (sapnco.dll and sapnco_utils.dll), the SAP Cryptographic library (sapcrypto.dll), your ticket license file, and sapgenpse.exe to this directory.
3. Copy the server certificate you exported from your SAP instance to this directory.
4. Add two system environment variables to your server:
 - o SECUDIR should be the directory you just created (for instance C:\SAPSNC)
 - o SNC_LIB should be the full path of the SAP Encryption library (for instance C:\SAPSNC\sapcrypto.dll).
5. Following SAP's instructions, use SAPGENPSE (or other tools that SAP may provide) to generate the PSE, including the cred_v2 file and the X.509 certificate. See [Configuring the Use of the SAP Cryptographic Library for SNC](#). For example, you could run these commands from a command prompt window with Administrator permissions in the C:\SAPSNC directory:

```
sapgenpse get_pse -p target.pse -x <PASSWORD> <DISTINGUISHED NAME>
```

```
sapgenpse seclogin -p target.pse -x <PASSWORD> -O <DOMAINUSER>
```

```
sapgenpse maintain_pk -a <CERT FILE FROM SAP GUI> -p target.pse -x <PASSWORD>
```

```
sapgenpse maintain_pk -v -l -p target.pse -x <PASSWORD>
```

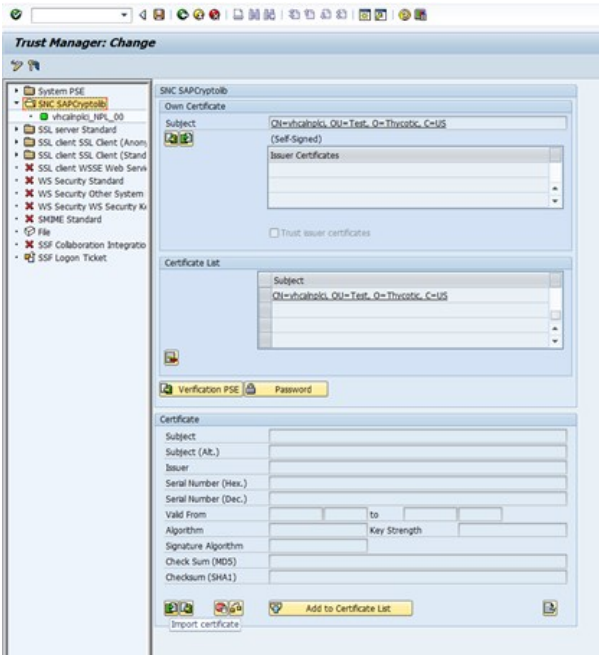
```
sapgenpse export_own_cert -o target.crt -p target.pse -x <PASSWORD>
```

6. When you create the server credentials with the sapgenpse "seclogin" command, specify a Windows or Active Directory user for the credentials. You have two options here:
 - o Specify the same user who runs your Secret Server or distributed engine as the one who is allowed to use the PSE you just setup. This is the easier option. Secret Server
 - o Specify a different Windows or Active Directory user. If you choose this option, you need to also create a secret for that user in Secret Server as either a Windows or Active Directory secret. Add this secret to your SAP SNC secret's associated secrets.

Note: For more information about using the SAPGENPSE tool, see [Creating the Server's Credentials Using SAPGENPSE](#).

Importing PSE information to the SAP GUI

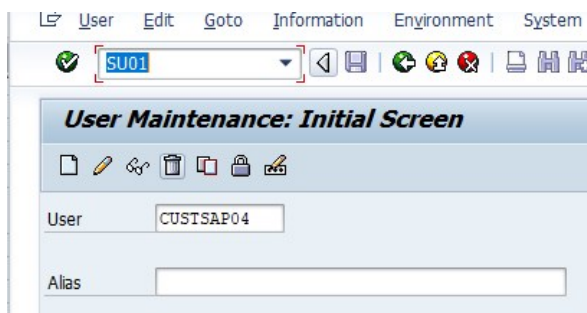
1. As above, refer to SAP's documentation for details on getting your PSE recognized by your SAP server. This is just an example.
2. Import the certificate you created above ('target.crt' in my example) through the STRUST transaction in the SAP GUI:
 1. Go to **STRUST \> SNC SAPCryptolib**.
 2. Click the entry below the SNC SAP Cryptolib folder. In the example below it is vhcainplici_NPL_00.



3. If prompted for a password, type it and then click the green checkmark button.
4. Click the Import Certificate icon on the far left on the bottom (hover over). The Import Certificate dialog box appears.
5. Type your certificate's file path.
6. Click the green checkmark button. The dialog disappears.
7. Confirm the certificate details are now in the **Certificate** section.
8. Click **Add to Certificate List**.

Note: If the button is not enabled, you may need to click the Display or Edit (pencil and glasses) button.

9. Confirm the certificate now appears in the **Certificate List** section.
 10. Save and exit.
3. Go to the **SU01** function.
 4. Type your SAP user's name in the **User** text box.



5. Click the pencil icon to edit.

- In the **SNC** tab, define the SNC name using the syntax: p:<YOUR USER'S DISTINGUISHED NAME>.

The screenshot shows the 'Maintain Users' transaction in SAP. The user 'IBARUCCOMH' is selected. The 'SNC' tab is active, showing the following configuration:

- SNC Status:** SNC is active on this application server. Unsecured logon is generally permitted.
- SNC Data:**
 - SNC name: p:CN=vhcalnplc,OU=Test,O=Thycotic,C=US
 - Canonical name defined:
 - Allow password logon for SAP GUI (user-specific):
- Administrative Data:**
 - Created: CUSTSAP03, 27.04.2021, 19:37:13
 - Modified: CUSTSAP03, 27.04.2021, 21:05:32
- Other SAP Users with Same SNC Names:**

Client	User	SNC name
001	INDJUM01	p:CN=vhcalnplc,OU=Test,O=Thycotic,C=US
001	INDJUM02	p:CN=vhcalnplc,OU=Test,O=Thycotic,C=US
001	OC	p:CN=vhcalnplc,OU=Test,O=Thycotic,C=US

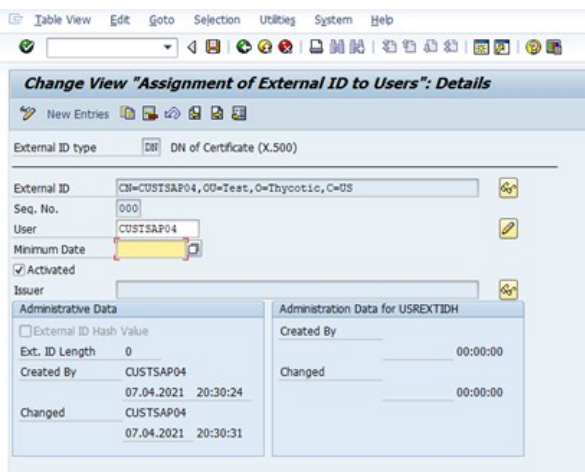
- Save and exit the **SU01** transaction.
- Go to the **SM30** transaction.

The screenshot shows the 'Edit Table Views: Initial Screen' transaction in SAP. The 'Table/View' field contains 'VUSREXTID'. The 'Restrict Data Range' section has 'No Restrictions' selected. The bottom navigation bar includes buttons for 'Display', 'Maintain', 'Transport', and 'Customizing'.

- Type VUSREXTID in the **Table/View** text box.
- Click the **Maintain** button. A dialog box appears:



11. Select **DN** as the work area.
12. Click the check mark icon button.



13. Click the **New Entries** button. The Change View "Assignment of External ID to Users" panel appears:

Change View "Assignment of External ID to Users": Overview

New Entries

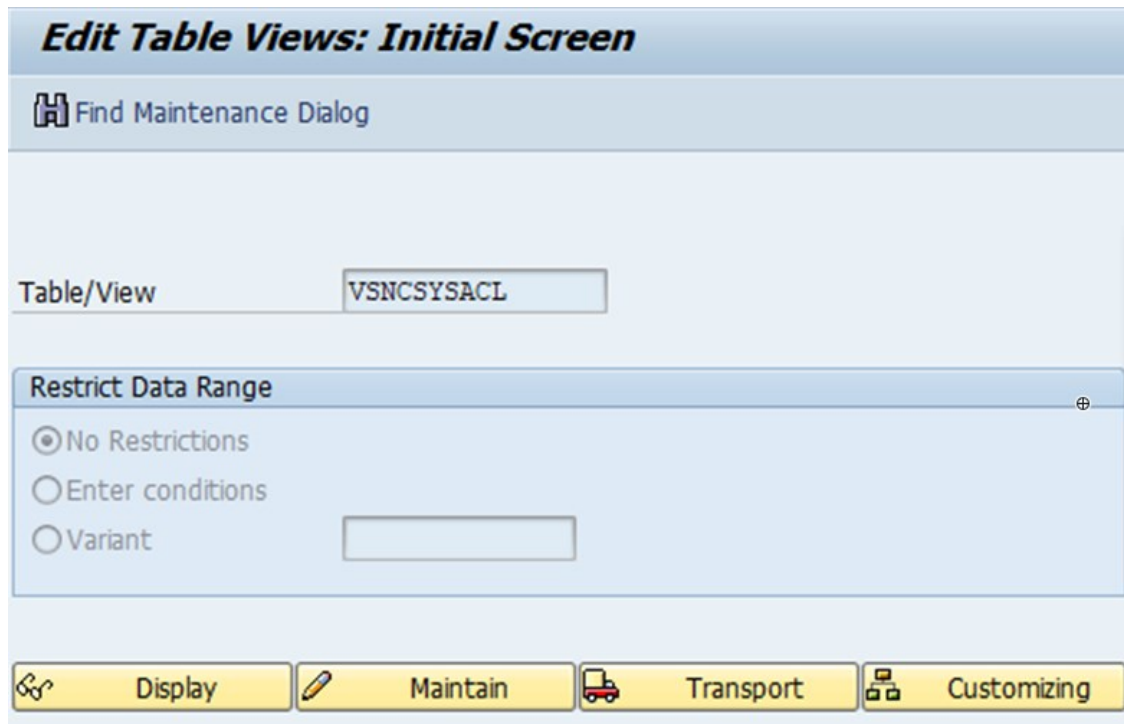
External ID type DN of Certificate (X.500)

Assignment of External ID to Users

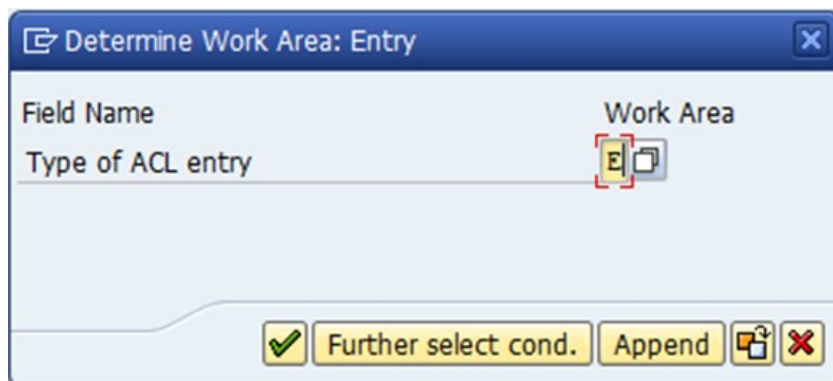
E..	External ID	User	Act.
<input type="checkbox"/>	CN=INDIUM01, OU=Test, O=Thycotic, C=US	INDIUM01	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=OC, OU=Test, O=Thycotic, C=US	OC	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=OCX509, OU=Test, O=Thycotic, C=US	OCX509	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=target, OU=Test, O=Thycotic, C=US	PASS_NO_SSO	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=target, OU=Test, O=Thycotic, C=US	CUSTSAP08	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=vhcalnplci, OU=Test, O=Thycotic, C=US	TEST123	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=vhcalnplci, OU=Test, O=Thycotic, C=US	INDIUM01	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=vhcalnplci, OU=Test, O=Thycotic, C=US	INDIUMADM	<input checked="" type="checkbox"/>
<input type="checkbox"/>	CN=vhcalnplci, OU=Test, O=Thycotic, C=US	CUSTSAP05	<input checked="" type="checkbox"/>

Position... Entry 1 of 9

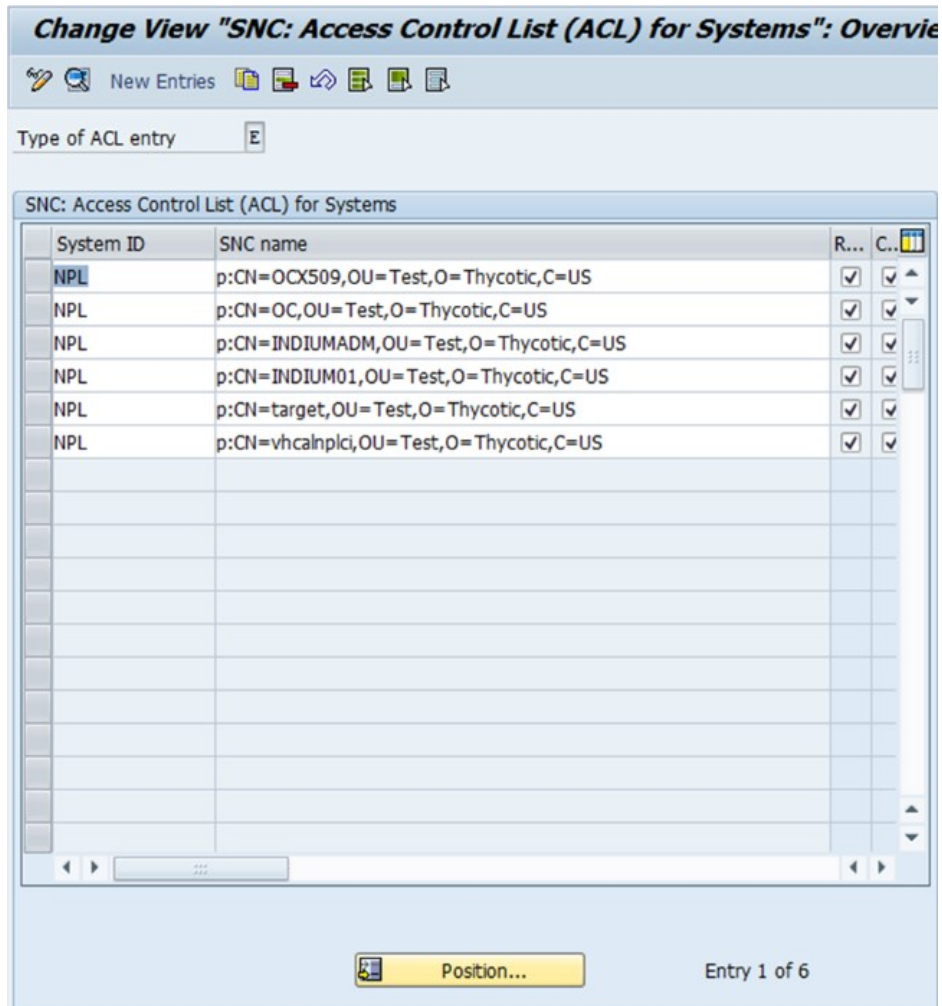
14. Click the Details (magnifying glass) icon. A details panel appears.
15. Fill out the fields as follows:
 1. Replace the **External ID** with your own
 2. Click to select the **Activated** check box.
 3. Type your SAP username in the User text box.
 4. Type a sequence number in the **Seq. No.** (sequence number) text box. For example, 000.
 5. Save and exit.
16. Return to the SM30 function.



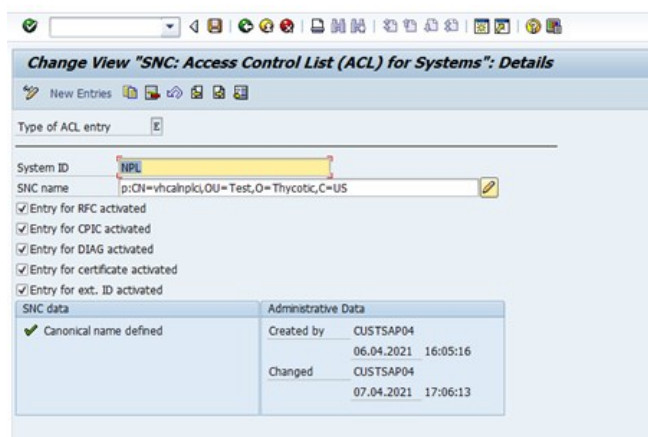
17. Type vsncysacl in the **Table/View** text box.
18. Click the Maintain button. A dialog box appears:



19. Select **E** as the work area.
20. Click the check mark icon button. The dialog box disappears.
21. Click the **New Entries** button. The Change View "SNC: Access Control List (ACL) for Systems" panel appears:



22. Click the Details (magnifying glass) icon. A details panel appears.



23. The **System ID** should match the system ID of your SAP instance. The **SNC name** should be the distinguished name of the server. There should only be one entry in this table for the server.

24. Confirm that a "Canonical name defined" message appears.
25. Save and exit

Creating an SAP SNC Secret in Secret Server

SAP SNC Account secrets are created in the same way as the original SAP Account secrets but have additional fields, as described above. For details that apply to both the SAP Account and SAP SNC Account secrets, see [SAP Heartbeat and Password Changing](#).

If your PSE was created for a Windows or Active Directory user other than the one who runs Secret Server or distributed engine, you need to add that user to your SAP SNC Account secret's associated users. To do this, add your user as either a Windows Account or an Active Directory secret. Next, open your SAP SNC Account secret and navigate to the Remote Password Changing tab to add that secret as an Associated Secret

If you do not use single sign-on or if you choose to use the username and password without the X.509 certificate for authentication, the X.509 certificate may be omitted.

Troubleshooting

SAP Account Secret Work but SAP SNC Secrets Do Not

SNC uses port 4800 to communicate. If the original SAP Account secrets work but SAP SNC secrets do not, be sure that port 4800 is not blocked by your firewall or VPN.

Client-Side Errors

If you experience client-side errors (such as generating a client certificate), right click on your SAP DLL files (sapcrypto.dll, sapnco.dll, or sapnco_utils.dll), and make sure that they are not blocked by your OS.

Distinguished Name Errors

If you run into an error message a distinguished name (DN) error, such as Exception: LOCATION CPIC (TCP/IP) with Unicode ERROR GSS-API(maj): No credentials were supplied Unable to establish the security context target="p:CN=vhcalnplci, OU=Test, O=Thycotic, check your spacing in the distinguished name. SAP can be strict about adding or removing the spaces after commas in the DN.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Field Slug Names

A *field slug name* in Secret Server is a unique human-readable identifier for a data field in a Secret Server template. The field slug name is available for integrating with third-party applications via API calls. Slug names are programmatically available for API calls but are not visible to template users (secret creators). Instead, they are displayed as references in secret templates.

Note: If you are not planning to access Secret Server with an API, slug field names are not for you—leave the suggested name as is.

Figure: Field Slug Name in a Secret Template

FIELDS		
FIELD NAME	FIELD SLUG NAME	FIELD DESCRIPTION
Public Key	public-key	The SSH public key.
Private Key	private-key	The SSH private key.
Private Key Passphrase	private-key-passphrase	The passphrase for decrypting the SSH private key.
Notes	notes	Any additional notes.
* <input type="text"/>	* <input type="text"/>	<input type="text"/>

Field slug names are automatically generated, based on the field name, when the field is created. For example, "User Name" became "user-name." Characters that are potentially problematic for programming, such as spaces, are swapped out. The automatically generated name is unchangeable by human users, unlike the field name. If API calls were based on the field name, human users with access to the template could break those calls, simply by changing the name.

With Secret Server 10.7.X+, The generated field slug names are now user-definable. You can edit the generated names to:

- Conform to a naming convention used in your API calls.
- Maintain the same name for a field across secret templates to simplify coding by developers.

The only requirement is that each slug field name is unique to that template.

Note: If you are wondering how Secret Server internally uniquely identifies fields, there is an internal ID that is not accessible by users or APIs. It is not available read-only (for API use) because we want to futureproof integrations from internal changes to Secret Server.

Note: The user-definable field slug names are also automatically generated when you upgrade from a version of Secret Server that did not have user-defined field slug names. If there are two fields with the same field name, the second (and later) generated field slug name has an incremented number appended to it.

Secret Template Fields

Note: If you want to programmatically manipulate fields, see [Field Slug Names](#).

Note: To use a custom SSH RPC port, add a field named "Port" to your secret template. Empty port fields are equivalent to the default port, 22.




Field Types

Template fields can be specified as one of several different types to enhance customization:

- **File:** File attachment link. File attachments are stored in the Microsoft SQL Server database.
- **General List:** Preconfigured selectable list for launcher enhancement or general use. See [Secret Template List Fields](#).
- **Notes:** Multi-line text-entry field.
- **Password:** Password type text-entry field.
- **Text:** Single-line text-entry field.
- **URL:** Clickable hyperlink.
- **URL List:** Preconfigured selectable list for general use. See [Secret Template List Fields](#).

Editing Fields

The secret template designer provides several settings to customize secret template text-entry fields:

- To add a secret text-entry field, fill out the values and click the + button.
- To delete a text-entry field, click the  icon. There is a confirmation dialog box before deletion takes place.
- To edit a text-entry field, click the  icon. Click either the  icon to save or the **X** icon to discard the changes.


Text-Entry Field and Control Settings

The settings available for text-entry fields are:

- **Field Name:** Name of the text-entry field. This name is used for the Create New drop-down list on either the Dashboard's Create Secret Widget or Home page.
- **Field Description:** Description of the text-entry field.
- **Field Type:** Type of the text-entry field. See below for a description of the different text-entry fields.
- **Is Required:** Whether the text-entry field should require a value. These check boxes are checked for correct content when the user attempts to create this secret. A validation error is displayed if not entered correctly.
- **History:** Number of values to keep in the text-entry field's history of values.
- **Searchable:** Whether that text-entry field should be indexed for searching. By default, passwords are not indexed. File attachments and history cannot be indexed for searching.
- **Edit Requires:** Minimum permissions on the secret needed in order to edit the value on the secret. The options are Edit, Owner and Not Editable. This enables the secret text-entry field to be locked down at a more granular level than other text-entry fields on the template.
- **Hide on View:** If checked, this text-entry field is not displayed to users when viewing the secret. The text-entry field is only be displayed when the secret is in Edit mode.
- **Expose for Display:** If checked, this text-entry field is available to be displayed as a Custom Column on the Secret Server Dashboard.

Note: All text-entry fields that are set to "Expose for Display" are **not** encrypted in the database. Only check this value if the secret text-entry field data is not considered privileged information.

The order of the text-entry fields in the Template Designer grid is the same as those that appear when the user views or edits a secret created from the template. The order can be modified through the up and down arrows on the grid.

Default values can be specified on each text-entry field by clicking the edit defaults  button. These added values appear as a list on any

secret created from this template.

Secret Template List Fields

Overview

With secret template list fields, administrators can create new lists that can be shared by multiple secrets. Clicking on an existing list goes to the details page for that list where the user can set the list's name, description, and the options available in the list.

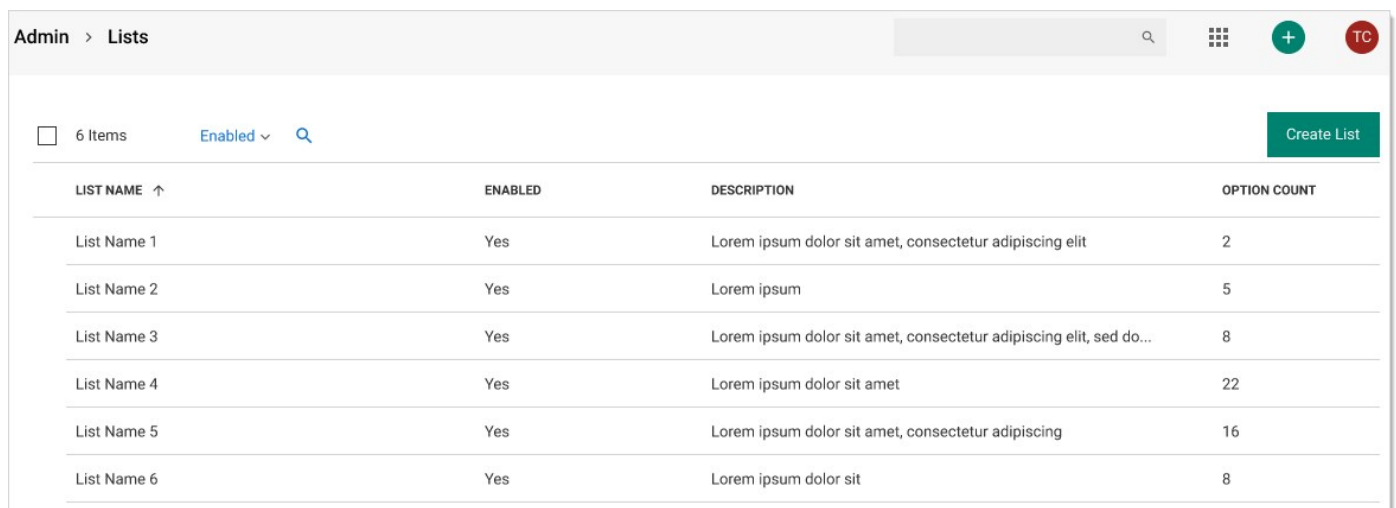
You can optionally group list options by category, which make using very large lists easier. For instance, a list of machines might have the machines categorized by function, such as "Web Server" or "Database Server." You could also use categories for locations, such as "London," "New York," or "Tokyo."

List categories are displayed on the secret and on the launcher dialog with the options sorted alphabetically within categories, which are also sorted alphabetically. Options can be duplicated in multiple categories and will show up in each one. In addition to manually adding categories and options, you can upload a file containing the list options. Teams (Admin > Teams) - The team details page

Adding a New List Field

Task 1: Create the List

1. Go to **Admin > Lists**. The List page appears:



The screenshot shows the 'Admin > Lists' page. At the top, there is a breadcrumb 'Admin > Lists', a search bar, and navigation icons. Below the header, there is a summary row with a checkbox, '6 Items', 'Enabled' with a dropdown arrow, a search icon, and a green 'Create List' button. The main content is a table with four columns: 'LIST NAME' (with an upward arrow), 'ENABLED', 'DESCRIPTION', and 'OPTION COUNT'. The table contains six rows of data.

LIST NAME ↑	ENABLED	DESCRIPTION	OPTION COUNT
List Name 1	Yes	Lorem ipsum dolor sit amet, consectetur adipiscing elit	2
List Name 2	Yes	Lorem ipsum	5
List Name 3	Yes	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do...	8
List Name 4	Yes	Lorem ipsum dolor sit amet	22
List Name 5	Yes	Lorem ipsum dolor sit amet, consectetur adipiscing	16
List Name 6	Yes	Lorem ipsum dolor sit	8

2. Click the **Create List** button. The Create List popup appears:

Create List

Name *

Description

3. Type the name in the **Name** text box.
4. (Optional) Type a description in the **Description** text box.
5. Click the **Save** button. The configuration page for the new list appears:

Admin > Lists > Web Servers

List Detail Audit

List Status [Edit](#)

Disable a list to prevent it from being used. **Enabled** Yes


List Detail [Edit](#)

List name must be unique. **List Name *** Web Servers

Description None

List Options

Add options organized by category, or an uncategorized list of options can be created. 0 Items **Uncategorized** **Add**



No Items Found

There is no information available to display on this screen.

6. Click the **Add** button in the **List Options** section and select **Create Category**. The Create Category popup appears:

Create Category

Category *

Cancel **Save**

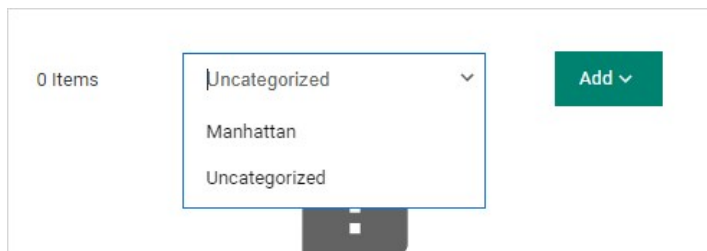
Note: If you want a list with no categories, choose Uncategorized for category, and follow these same instruction for

adding options.

Note: You can also create categories from a comma-delimited list in a text file. Select the **Add** button and select **Add from File**. This can be either a list of options, one option per line, or a list of comma-delimited values in the format option,category with one pair per line. Files can also combine these formats, and any line without a comma will be treated as a option without a category.

7. Type the name for the category in the **Category** text box. We typed "Manhattan."

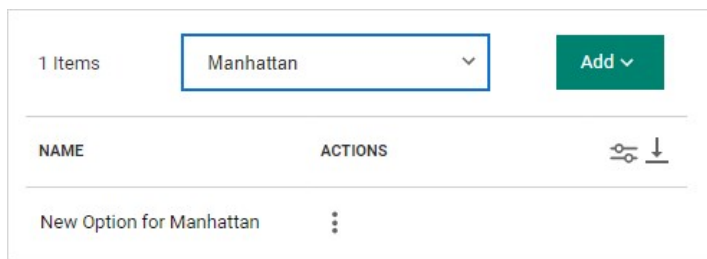
8. Click the **Save** button. The category name now appears in the dropdown list:



The screenshot shows a user interface for adding a category. On the left, it says "0 Items". In the center, there is a dropdown menu with "Uncategorized" selected. Below the dropdown, the options "Manhattan" and "Uncategorized" are visible. On the right, there is a green "Add" button with a downward arrow.

9. Add another category the same way. We added "Albany."

10. Click the dropdown list to select a category you just added. We chose "Manhattan." A table appears below the dropdown list:

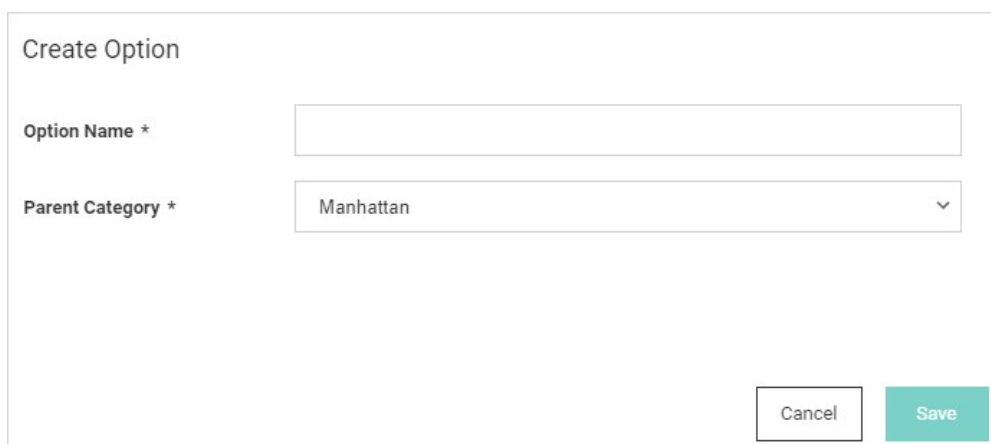


The screenshot shows the user interface after selecting a category. On the left, it says "1 Items". In the center, the dropdown menu now shows "Manhattan" selected. On the right, there is a green "Add" button. Below the dropdown, a table is displayed with the following structure:

NAME	ACTIONS	
New Option for Manhattan		⋮

Note that there is no option (list item) listed.

11. Click the **Add** button and select **Create Option**. The Create Option popup appears:



The screenshot shows the "Create Option" popup form. It has two input fields: "Option Name *" and "Parent Category *". The "Parent Category *" field is set to "Manhattan". At the bottom right, there are two buttons: "Cancel" and "Save".

12. Type the name for the Option in the **Option Name** text box. We typed "Web Server."

13. Click to select the category the new option will belong to in the **Parent Category** dropdown list. We selected "Manhattan."

14. Click the **Save** button. The new option appears in the list:

NAME	ACTIONS
New Option for Manhattan	⋮
Web Server	⋮

15. Add another option the same way. We added "Database Server."

16. Repeat the process for the Albany category. The list new table looks like this:

NAME	ACTIONS
Database Server	⋮
New Option for Albany	⋮
Web Server	⋮

17. For future reference, click the three vertical dots button in the **Actions** column for one of the options. Three actions appear:

- Update Option: Rename the option
- Move to Category: Move the option to another category in the same list
- Delete Option: Remove the option from the category.

For now, we will not use any of them.

18. You now have a new categorized list available for secrets (via a secret template with the list).

Note: If you ever want to view past changes to a list or category, click the Audit tab for the list.

Task 2: Create a Template Using the List

1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:

Manage Secret Templates

Active Directory Account

Show Inactive

[← Back](#) [✎ Edit](#) [+ Create New](#) [↶ Export](#) [☰ View Audit](#)

Active Templates [* Password Requirements](#)

[A Character Sets](#) [⚙️ Configure Launchers](#)

[👤 Configure Secret Template Permissions](#)

2. Click the **Create New** button. The Create New Secret Template page appears:

Create New Secret Template


Name of the New Secret Template? *


[+ Create](#) [✕ Cancel](#)

3. Type the template name in the **Name of the New Secret Template?** text box. We typed "Acme Server Template."
4. Click the **Create** button. The Secret Template Designer page for that new template appears:

Secret Template Designer

SETTINGS

Secret Template Name Acme Server Template
Secret Template Icon 
Active?
Expiration Enabled?
Validate Password Requirements On Create?
Validate Password Requirements On Edit?
Field Displayed on Basic Home Folder Name
One Time Password Enabled No

 Edit

FIELDS

FIELD NAME	FIELD SLUG NAME	DESCRIPTION

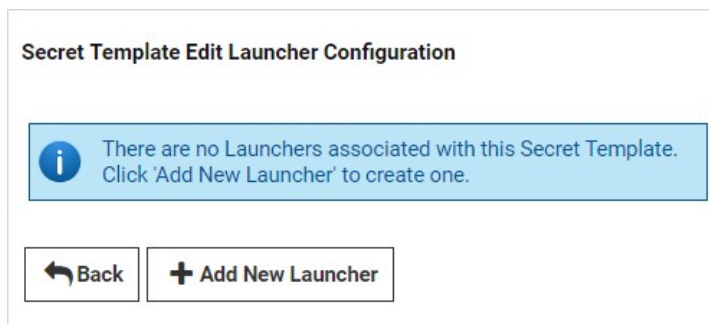
5. Go to the **Fields** section:

FIELDS

FIELD NAME	FIELD SLUG NAME	DESCRIPTION	TYPE	IS
* <input style="width: 90%;" type="text"/>	* <input style="width: 90%;" type="text"/>	<input style="width: 95%;" type="text"/>	Text ▼	

6. Complete the following steps for
7. Type My Server List in the **Field Name** text box for the first (and currently only) field.
8. Click the **Type** dropdown list for the field and select **List**.
9. Click to select the **Is Required** check box.
10. Click the **+** on the far right of the table row.
11. Type Password in the **Field Name** text box for the second field.
12. Click the **Type** dropdown list for the field and select **Password**.
13. Click to select the **Is Required** check box.
14. Click the **+** on the far right of the table row.

15. Type Username in the **Field Name** text box for the third field.
16. Click the **Type** dropdown list for the field and select **Text**.
17. Click to select the **Is Required** check box.
18. Click the **+** on the far right of the table row.
19. Type Computer in the **Field Name** text box for the fourth field.
20. Click the **Type** dropdown list for the field and select **Text**.
21. Click to select the **Is Required** check box.
22. Click the **+** on the far right of the table row.
23. Type Domain in the **Field Name** text box for the fifth field.
24. Click the **Type** dropdown list for the field and select **Text**.
25. Click to select the **Is Required** check box.
26. Click the **+** on the far right of the table row.
27. Click the **Configure Launcher** button. The Secret Template Edit Launcher Configuration page appears:



28. Click the **Add New Launcher** button. The Secret Template Edit Launcher Configuration page appears:

Secret Template Edit Launcher Configuration

Launcher Type to use: Remote Desktop

Computer: <user input> *

Domain: <blank> *

Password: <blank> *

Username: *

ADVANCED SETTINGS

Restrict User Input ?

Save Cancel

29. Click the **Launcher Type to Use** dropdown list and select **Remote Desktop**.
30. Click each of the four following dropdown lists and select the matching value.
31. Click to select the Restrict User Input check box. A new section appears:

ADVANCED SETTINGS

Restrict User Input ?

i In order to restrict user input the secret template must have at least one field of type "List"

Allow List: None ?

Include Machines From Dependencies: ?

Deny List: None ?

32. Click the **Allow List** dropdown list to select **My Server List**.
33. Click the Save button.
34. Click the **Home** button in the main menu to return to the Secret Server dashboard.

Task 3: Create a Secret

1. Click the **+** next to **Secrets** on the main menu to create a new secret based on the template you just created. The Create New Secret popup appears:

Create New Secret

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Personal Folders/Will Sprunk [Change](#) [Clear](#)

Choose a Secret Template

- Acme Server Template
- Active Directory Account
- Active Directory No Prompt
- AD Test Template
- Amazon IAM Console Password
- Amazon IAM Key
- API User Credentials
- AS/400 IBM iSystem
- Bank Account
- Cisco Account (SSH)

[Cancel](#) [Create Secret](#)

2. Click the secret template you just created in the **Choose a Secret Template** list. Another Create New Secret popup appears:

Create New Secret

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

Secret Template Acme Server Template [Change](#)

Folder [Personal Folders/Will Sprunk](#) [Clear](#)

Secret Name *

Server

Site

- Note that one of the dropdown lists has the same name as the list field you created earlier. Click it, and you see the list categories you created. The list is available for that secret's launcher. In addition, you can use a list to provide the allow and deny lists for restricted user input based on the user's selection of the list in the secret.

With this Secret Server feature, admins can use private SSH keys for PuTTY launcher sessions as well as for RPC tasks (configurable through password changer settings) and Unix and Linux discovery. Passphrases can additionally be stored, if necessary, to decrypt the private keys for additional security. The Unix Account (SSH) secret template includes text-entry fields for the private key and passphrase by default:

The SSH Key template is included by default and can be used to store SSH keys that can later be selected for use in RPC, discovery or launcher authentication for other secrets:

Note: Starting with version 10.1.000000, Secret Server also supports SSH key rotation on secrets.

The **Unix Account (SSH Key Rotation)** and **Unix Privileged Account (SSH Key Rotation)** secret templates use password changers that change the public key in the account's `authorized_keys` file as well as change the password on the account. Secret Server ships with a password changer and custom command sets that allow an account to change its own public key and password, and a password changer and custom command sets that changes a user's public key and password using a privileged account. These scripts can be customized for different Unix environments.

For more information about SSH Key Rotation, see the [SSH Key Rotation](#) (KBA) and [SSH Key Rotation Quick Start](#) (KBA).

Character sets are a collection of distinct characters that are used in password requirements and password rules. Custom sets can be created, and both ASCII and Unicode are supported. For more information on setting up compliance checks and password generation standards, see [Password Requirements](#). The five standard character sets are:

- Lower Case (a-z)
- Upper Case (A-Z)
- Numeric (0-9)
- Non-Alphanumeric (!@#\$%^&* ())
- Default – Includes all the above

To manage character sets, click the **Character Sets** button on the **Administration > Secret Templates** page. Only character sets which are not currently used by a password requirement can be deleted.

Secret Server supports naming patterns for secret templates. Naming patterns are a way for administrators to maintain consistency for secret names and can help ease both browsing and grouping secrets by name. Patterns are created using regular expressions. Regular expressions are a formal set of symbols commonly used to match text to patterns. For example, the regular expression `^\\w+\\w+$`, allows `NTDOMAIN01\\USER3454` but not `USER3454` on `NTDOMAIN01`.

Note: Regular expressions are beyond the scope of this document. They are very powerful and can get quite complex—books have been written on the topic. Microsoft offers a good overview at their [Regular Expression Language Quick Reference](#) Web page.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Overview

A password requirement is a stored Secret Server object that defines the requirements on a password text-entry field to validate user-entered passwords or make auto-generated passwords conform to set specifications. You can have multiple password requirements, but only one can be set to the default.

A password requirement is made up of a minimum and maximum length, a set of characters, and optional rules such as "At least three upper-case characters" or "The first character must be lower-case". The default password requirement is 12 characters from the default character set, with at least one upper-case, lower-case, numeric, and symbol character.

Creating a Custom Password Requirement

To create a new password requirement:

1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:

SECRET TEMPLATES ↑	TOTAL SECRETS	ACTIVE
Acme Server Template	0	<input checked="" type="checkbox"/>
Active Directory Account	39	<input checked="" type="checkbox"/>
Active Directory No Prompt	1	<input checked="" type="checkbox"/>
AD Test Template	1	<input checked="" type="checkbox"/>

2. Click the **Password Requirements** tab:

Secret Templates

Templates Character Sets Password Requirements Launchers Audit

4 Items Custom Dictionaries Create

NAME ↑	DESCRIPTION	DEFAULT	⚙️ ↓
Default	The default password requirement, ...	Yes	
FSQA		No	
Mainframe	Mainframe Password Requirement	No	
SAP	SAP Password Requirement	No	

3. Click the **Create** button. A popup appears:

Create Password Requirement

Name *

Description

Minimum Password Length *

Character Set ▼

Is Default

Prevent Username

Prevent Spatial Pattern

Prevent Sequential Pattern

Cancel Save

4. Type the name of the new password requirement in the **Name** text box.

5. (Optional) Type a description of the new password requirement in the **Description** text box.
6. Click the **Minimum Password Length** spinner to select or type a minimum allowed password length.
7. Click the Character Set dropdown list to select a character set for the password. The out-of-the-box default is abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#%&^*().
8. If you want the password requirement to become the new default, click to select the **Is Default** check box.
9. Click to select the desired password no-no check boxes. The options are:
 - **Prevent Username:** Do not allow the username to be part of the password.
 - **Prevent Spatial Pattern:** Do not allow strings of characters based their order on the keyboard, such as qwerty or asdfg.
 - **Prevent Sequential Pattern:** Do not allow strings of characters based on their order in the character set, such as abcd or 5678.
10. Click the **Save** button. The popup closes and the page for the new requirement appears (containing the choices you just made for the details and generation sections):

Admin > Secret Templates > Password Requirements > My PW Requirement

My PW Requirement

Password Requirement Details

Password Requirements set requirements on a password text-entry field to validate user-entered passwords or make auto-generated passwords conform to set specifications.

Name	My PW Requirement
Description	None
Is Default	No
Example	rnobrdkjhym

Password Generation

Set parameters for password generation under this requirement.

Minimum Password Length * 8

11. Scroll down to the **Password Validation** section.
12. Click the **Edit** button. The section expands:

Password Validation

Set validation rules for password generation under this requirement.

Prevent Username

Prevent Dictionary Words

Prevent Spatial Pattern

Prevent Sequential Pattern

Prevent Words from Dictionary 'Test'

13. Most of the validation rules are ones you have already set with these two exceptions, which you can now set:
 - **Prevent Dictionary Words:** Do not allow everyday English words in the password.
 - **Prevent Words from Dictionary ...:** Do not allow words that appear in the named dictionary. In our example, the dictionary is named "Test."
14. Go to the **Starting and Ending Character Validation** section.
15. Click the **Edit** button. The section expands:

Starting and Ending Character Validation

Set validation rules that passwords must start with a specified number of characters from a Character Set, end with a specified number of characters from a Character Set, or both.

Require Specific Starting Characters

Require Specific Ending Characters

16. To require specific starting characters, click to select the **Require Specific Starting Characters** check box. Two hidden controls appear:

Require Specific Starting Characters

characters from

17. Type or click the spinner to set the number of required starting characters.
18. Click the **characters from** dropdown list to select the character set to draw the characters from.
19. Repeat the procedure for any desired ending characters.
20. Click the **Save** button. An edit button now appears for the Character Count Validation section.
21. To set the minimum character count from each character set:
 1. Click the **Edit** button for the **Character Count Validation** section. The section expands:

2. Click the **Add Rule** button. Two hidden controls appear:

and another hidden control appears:

3. Use the exact same procedure as earlier to define the rule.
4. Click the Add Rule button again to add rules for other character sets.
5. If you want to require how many of the rules must apply, type the number in the **Minimum Required Character Count Rules** text/dropdown list box or select **All**, which is the default.
6. Click the **Save** button.

Note: The explicit character rules cannot conflict with the implicit ones you created earlier or you will get an error when saving. For that reason, we suggest leaving the Password Requirements Details character set set to the default. Carefully consider any other conflicts if you get an error.

Note: To set a custom password requirement for a specific secret, use the "Customize Password Requirement" in the Security tab of a secret.

Note: You can enable or disable the validation of manually entered passwords at the secret template level via the "Validate Password Requirements on Create" and "Validate Password Requirements on Edit" settings.

Note: The "What Secrets Do Not Meet Password Requirements" report shows secrets containing a password that does not meet

the password requirements set for its secret template.

Note: Password requirements cannot include rules with overlapping character sets. For example, if an attempt is made to add both a "Minimum of 1 upper-case" rule and a "Minimum of 3 Default" rule to a new password requirement, an error displays.

Security and Hardening

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: Please see [Running the IIS Application Pool As a Service Account](#) for additional information.

Introduction

Integrated Windows Authentication (IWA) requires:

- Installing a SQL Server instance
- Creating a new domain service account
- Granting access to SQL Server database
- Registering a service account to run IIS and ASP.NET
- Assigning an account as an application pool identity

Note: For instructions on Creating the SQL account or Installing SQL Server see [Installing and Configuring SQL Server](#) (KBA).

Creating a Domain Service Account

The account needs access to the application server and database server. Ensure password expiration is not enabled or the account could lock you out of Secret Server.

Granting Access to SQL Server database

1. Connect to the Database instance using SQL Management Studio.
2. Right click on the Security node (ensure this is the top most security node under the instance and not under the database name itself) and select **New > Login**.
3. Enter the Login name as Domain\Username.
4. Ensure **Windows Authentication** radio button is selected.
5. If you have already created the database, then under **User Mappings** select the database and grant dbOwner permission. Otherwise, if you plan to have the Database created for you, under **Server Roles** select dbCreator.
6. Click the **OK** button.

Assigning Account as Identity of Application Pool

1. Open IIS (Run command inetmgr).
2. Click the Application Pool node.
3. Select Secret Server's Application Pool (default is SecretServerAppPool).
4. On the Right panel, Click .
5. Scroll down to the **Identity** row under **Process Model**.
6. In the popup, select **Custom Account > Set**.
7. Type the user as domain\username.
8. Type the password.
9. Click the **OK** button.

10. Recycle the application pool by clicking the **Recycle..** button under the **Application Pool** tasks.

The Common Criteria Standard

The Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), known as "Common Criteria," is an international standard for certifying security of computer systems, networks, and application software. Certification ensures that claims about the security attributes of the evaluated product have been independently verified in the specified, evaluated configuration and environment. The certification does not validate any security claims when the product is used outside of that evaluated configuration or environment.

Common Criteria for Secret Server

Secret Server

Secret Server releases 10.4 and later provide security features enabling users to achieve Common Criteria certification. To ensure that each parameter setting matches those evaluated and certified as secure by Common Criteria standards, follow these guidelines in their entirety.

Audience

This Common Criteria documentation is for users with administrative access to configure their organization's operational environment; knowledge of their organization's network infrastructure and applicable policies; and responsibility for installing, configuring, and operating enterprise infrastructure for their organization.

Note: This documentation is not updated with every Secret Server release—some releases do not affect the documentation's contents and thus do not warrant a document update.

Enabling Common Criteria Security Hardening

Secret Server releases 10.4 and later provide security features enabling users to achieve Common Criteria certification. Due to the stringency of the standards and the need for customized user configuration, not all of these features are automatically set by the installer application. Instead, an administrator must configure some settings, and this documentation provides the information to perform those configurations to comply with the Common Criteria evaluated configuration.

This documentation is closely associated with the Security Hardening Report in Secret Server (Click Reports > Security Hardening). Follow the checklists in Secret Server to adjust parameters to achieve Common Criteria certification.

Security Hardening Checklist

After installing Secret Server, navigate to the **Reports > Security Hardening** tab, and follow the checklist to ensure your environment is as secure as possible:

1563386730221

Evaluated Configuration

The **Target of Evaluation (TOE)** is Delinea Secret Server Government Edition v10.1, a software application used in the enterprise settings to define and maintain user credentials within a large organization. The evaluated configuration consists of the software application running on Windows Server 2016 (or later) with Microsoft .NET Framework 4.6.2 (or later) and Microsoft SQL Server 2016 (or later) database, installed locally.

Delinea Secret Server – Government Edition can manage any IT asset compatible with the following types of credentials:

- Windows Accounts
- Active Directory accounts
- Unix accounts (SSH)

These credential types define a broad range of compatible ESM products. Generally, Linux 2.6.32 or later, Windows Server 2008 R2 or later, Windows 7 Enterprise or later would be compatible.

Assumptions and Operational Environment

There are specific conditions that are assumed to exist in the TOE's Operational Environment. The following table lists assumptions about the Operational Environment as specified by the Protection Profile:

Assumption Name	Assumption Definition
A.CRYPTO	The TOE will use cryptographic primitives provided by the Operational Environment to perform cryptographic services.
A.ENROLLMENT	There will be a defined enrollment process that confirms user identity before the assignment of credentials.
A.ELECTRONIC SUPPORT MEASURES (ESM)	The TOE will be able to establish connectivity to other ESM products in order to share security data.
A.FEDERATE	Third-party entities that exchange attribute data with the TOE are assumed to be trusted.
A.MANAGE	There will be one or more competent individuals assigned to install, configure, and operate the TOE.
A.SYSTIME	The TOE will receive reliable time data from the Operational Environment.

The following table identifies the organizational security policies applicable to the TOE as specified by the Protection Profile:

P.BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
----------	---

Table: Organizational Security Policies

Configuring TLS

To achieve Common Criteria certification on Secret Server, you must enable Transport Security Layer (TLS).

Manually Disabling TLS Version 1.0

TLS 1.0 is no longer considered secure, so it is important to disable this version of the protocol on Secret Server. To do this, follow the instructions in **Manually Disabling TLS v1.0**.

TLS Diffie-Hellman Hardening Overview

For information on configuring your servers with stronger Ephemeral Diffie-Hellman hardening, see the **TLS Diffie-Hellman Hardening Overview**.

Restricting Server Cipher Suites for TLS

Allowed Suites

Common Criteria certification requires restricting the cipher suites configured on your server to only:

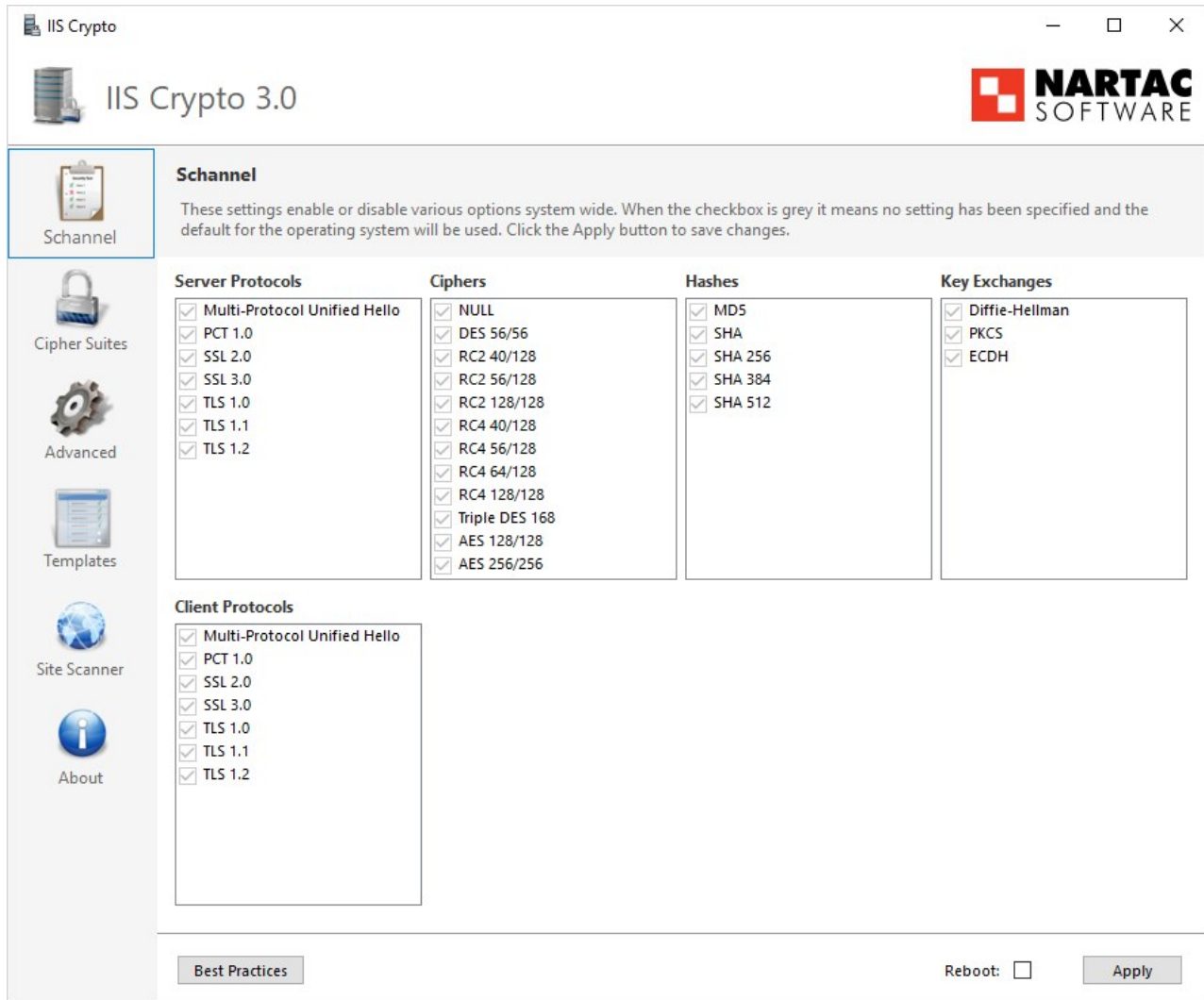
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA256

Restricting them can cause communication issues with other servers if they are not able to communicate using any of the above ciphers. In that case, you need to modify those servers to include these cipher suites to securely communicate according to Common Criteria guidelines.

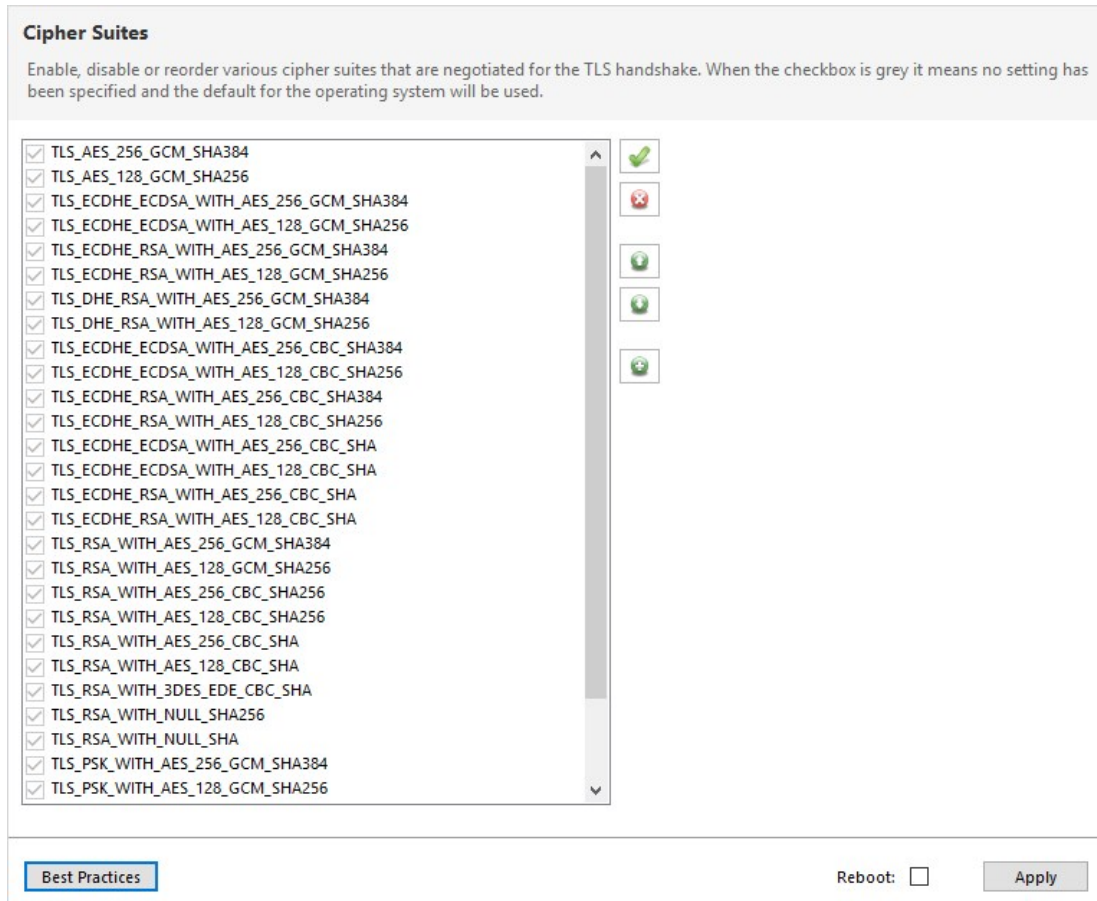
Changing Cipher Suites with the IIS Crypto Tool


One way to change the cipher suites on a computer is to use the free IIS Crypto tool:

1. Download the GUI version of the [NARTAC IISCrypto](#) tool.
2. Run the tool:



3. Click **Cipher Suites** button on the left. The Cipher Suites window appears:



4. Click the  **Uncheck All** button to uncheck all cipher suites
5. Find and click to select the suites in the list above.
6. Click the **Apply** button.

Configuring TLS with IIS

Common Criteria certification requires using HTTPS/SSL for all connections to the Secret Server Web page. To do this, follow the instructions in **TLS Configuration with IIS**.

Enabling TLS Auditing

To have Secret Server audit TLS connections and connection failures, follow the instructions in **Configuring Auditing for TLS Connections**.

Configuring TLS with Active Directory

To ensure that TLS is configured with Active Directory Follow the instructions in **Configuring TLS with Active Directory**.

Note: If you have any existing domains configured in Secret Server, you must edit them and enable LDAPS on each one.

Configuring TLS with Syslog

To configure TLS with Syslog, follow the steps in **Configuring Syslog/CEF External Audit Server**.

Additional Common Criteria Configurations

Configuring X.509v3 Certificates

For instructions on installing and configuring certificates on the Secret Server Web servers, see **Configuring X.509v3 Certificates**.

Enabling DPAPI

The Windows Data Protection API (DPAPI) is a pair of functions that allow access to operating-system-level data protection services to protect the master encryption key file, encryption.config.

To enable DPAPI, follow the instructions in the **Verify DPAPI Setting Is Enabled** section.

Enabling FIPS Mode

To configure your server and Secret Server to use the Federal Information Processing Standard (FIPS), follow the instructions in the **Verify FIPS Mode Is Enabled**.

Note: Also see: [Enabling FIPS Compliance](#)

Ensuring Zero Information Disclosure

To comply with Common Criteria requirements, you must configure Secret Server to not display any unnecessary information. This applies to unhandled errors as well as the application version number.

Configuring Custom Error Messages

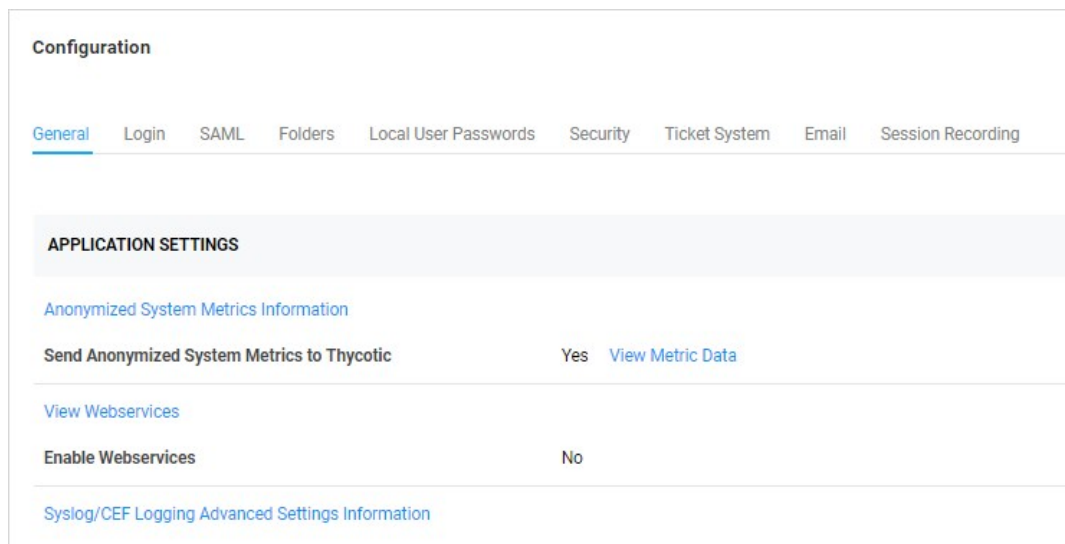
To hide detailed error messages and display a custom message when an unhandled error occurs:

1. Open `https://<your_secret_server_url>/ConfigurationAdvanced.aspx` in your browser.
2. Scroll to the bottom of the page and click the **Edit** button.
3. Type the message you want displayed to users in the **Zero Information Disclosure Message** text box.
4. Click the **Save** button.

Hiding the Application Version Number

To hide the application version number in the application header and footer:

1. Go to **Admin > Configuration > Security**.
2. Click the **Edit** button. The Configuration page appears:



3. Click the **Security** tab.
4. In the **Web Services** section of the page, click to select the **Hide Secret Server Version Numbers** check box.
5. Click the **Save** button.

Note: For diagnostic purposes, the application version number is still displayed on the Diagnostics page. Make sure that permissions to this page is limited to employees that may need to access this page when contacting Delinea technical support.

Configuring the Login Banner

For Common Criteria compliance, when a user first logs in, the login banner must reveal the user policy agreement and force that user to agree to the policy before logging into Secret Server. To configure the Login Banner according to Common Criteria guidelines, follow the instructions in **Configuring the Login Banner**.

Configuring Account Lockout

To access Secret Server, users must login with local or domain credentials. To comply with Common Criteria, Secret Server must use "account lockouts" to prevent repeated unsuccessful login attempts. Configurable by an Secret Server admin, an account becomes inaccessible after a defined number of unsuccessful authentication attempts until an admin unlocks the user's account.

To configure settings for account lockouts:

1. Navigate to **Admin > Configuration**.
2. Click the **Login** tab.
3. Click the **Edit** button.
4. Adjust the number in the **Maximum Login Failures** text box. The default is five attempts.

To Unlock a user's account:

1. Navigate to **Admin > Users > Select the User**.
2. Click the **Edit** button.
3. Click to deselect the **Locked Out** check box.
4. Click the **Save** button.

Disabling "Remember Me" Logins

A browser's "remember me" login function stores the user's login name and password so the user does not need to enter it again on that browser, which is both convenient and insecure. To disable "Allow Remember Me" during logins, follow the instructions in **How to Disable Allow Remember Me during Logins**.

Configuring SQL Server

To meet Common Criteria requirements, Microsoft SQL Server must be installed on the local machine—the same as the Secret Server Web server. During the install process for MS SQL, ensure that you use Windows authentication mode.

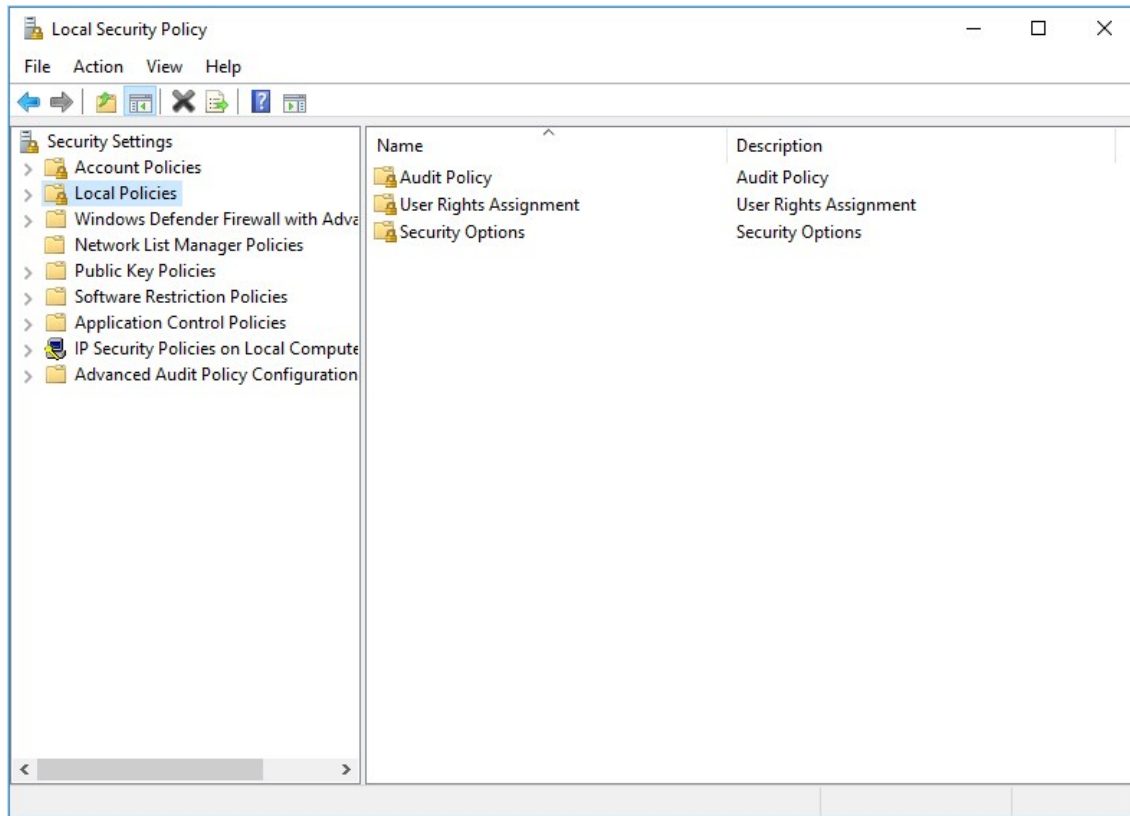
If you have an installed instance of Secret Server that does not meet this requirement, you can migrate the remote database to the server hosting Secret Server. If MS SQL Server is not installed and configured on the Secret Server, you must install it. The server must be configured with enough RAM, storage space, and processors to support running MS SQL Server and the Web site simultaneously. After copying the database, you can go to **Admin > Database** to point Secret Server to the new database location.

Note: Because the database must be installed locally with the Secret Server Web application for Common Criteria compliance, Secret Server is not fully compliant when running multiple nodes.

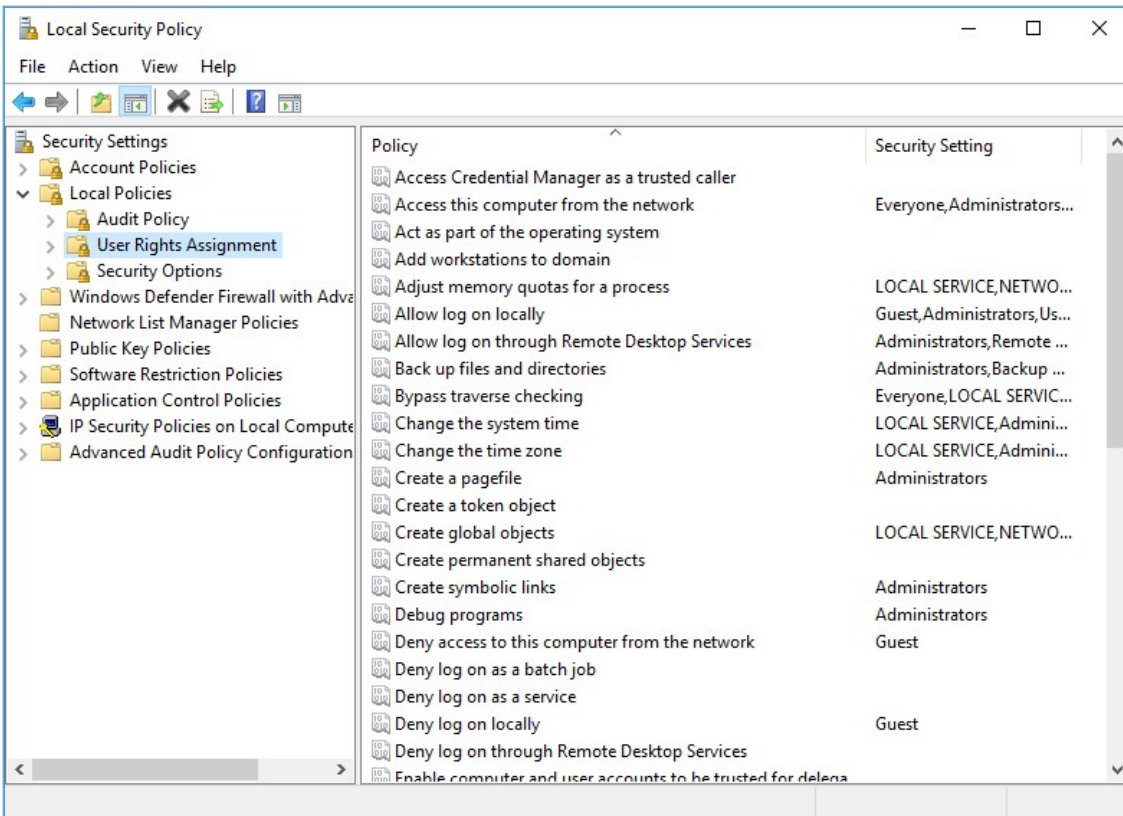
Running the IIS Application Pool with a Service Account

To use Windows authentication to access the SQL database, you should create a service account. To run the Secret Server IIS Application Pool with a service account:

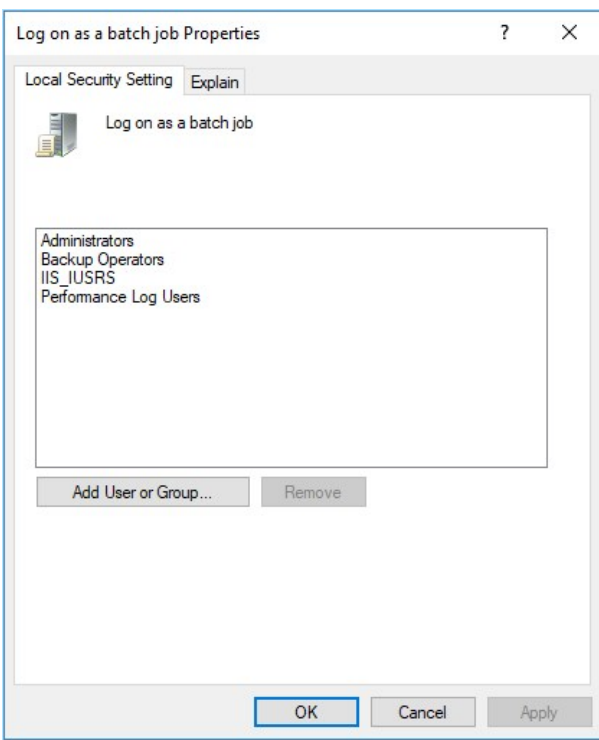
1. Open a command prompt window, change the directory to your .NET framework installation directory (usually C:\Windows\Microsoft.NET\Framework...) using the `cd` command.
2. Type `.\aspnet_regiis -ga <user_name>` and press **<Enter>**. The username is from the MS SQL Server user account.
3. Give your service account "modify" access to C:\Windows\TEMP.
4. Open the Local Security Policy App from your start menu.
5. Grant batch logon permissions to your service account:
 1. Open the Local Security Policy Console (search for and open `secpol.msc`):



1. Expand the **Local Policies** folder (not shown).
2. Click to select the **User Rights Assignment** folder.



3. Right-click **Log on as a batch job** in the right panel and select **Properties**.



4. Click the **Add User or Group** button.
5. Add your service account.
6. Click the **OK** button.

Note: If you use group policy to enforce "Log on as a batch job" and have group-managed service accounts, that will overwrite any local permissions to "Log on as a batch job" on all computers that have the policy applied. Using the local security policy is a safer option if you are not sure about your usage across your domain.

1. Grant "Impersonate a client after authentication" permission to the service account under **User Rights Assignment** the same way "Log on as a batch job" was assigned above.
2. If you now get a "Service Unavailable" error after applying "Log on as a batch job" permissions:
 1. Update your group policy settings (**Start > Run > Cmd** and type `gpupdate /force`) and restart the Windows Process Activation service.

Note: For more information, see [Running the IIS Application Pool As a Service Account](#).

Assigning Common Criteria Roles and Permissions

See the "Common Criteria Roles and Permissions" section of the Common Criteria Hardening Guide.

Managing User Passwords

See [Managing User Passwords](#).

Configuring Secret Templates

To enable only the secret templates that are certified Common Criteria compliant and to set Common Criteria-compliant password policies on those templates, see [Configuring Secret Templates](#) and [Configuring Password Policy for Secret Templates](#).

Setting Authentication Strength for Non-Password Credentials

See [Authentication Strength for Non-Password Credentials](#).

Configuring Remote Password Changing for SSH Key Rotation

See [Configuring Remote Password Changing for SSH Key Rotation](#).

Configuring External Auditing

Connecting to an External Audit Server

To connect to an external syslog/CEF audit server, see [Security—Connecting to an External Audit Server](#) and [Configuring Syslog/CEF External Audit Server](#).

Configuring Local Windows Event Log Auditing

See [Configuring Local Windows Event Log Auditing](#).

Installing Your System

Prerequisites

Minimum Hardware

- **CPU:** Intel 2.4 GHz 4 core 64-bit
- **RAM:** 8 GB
- **Data storage:** 60 GB of free space
- **Network:** Gigabit Ethernet adapter

Recommended Software

- Windows Server 2016 or later
- Microsoft .NET Framework 4.6 or later
- IIS v10 or later
- Microsoft SQL Server 2016 or later

For more information on SQL licensing, see Microsoft's 2016 [editions and components list](#). Use of SQL Server Express, although included in the installation package, was not covered by the evaluation.

Windows Server 2016 ships with .NET 4.6.2 included; .NET 4.6.2 is backwards compatible with all previous .NET 4.x versions. Microsoft ensures that new .NET framework versions are backwards compatible. The .NET framework is updated through Windows Update, where Microsoft dictates specific release of the .NET framework on a fully patched system.

Operational Environment

- **Audit Server:** syslog-ng 3.9 or later
- **Domain Controller:** Active Directory version 2008 R2 or later

More detailed System Requirements can be found at [System Requirements for Secret Server](#).

SQL Installation and Configuration

To maintain Common Criteria Compliance, *SQL must be installed locally* on the Secret Server web application machine before running the Secret Server installer. During the install process for SQL, ensure that you use *Windows authentication mode*.

The steps that follow walk you through setup and configuration for SQL Server 2016 Standard Edition. At the end of these steps you will have:

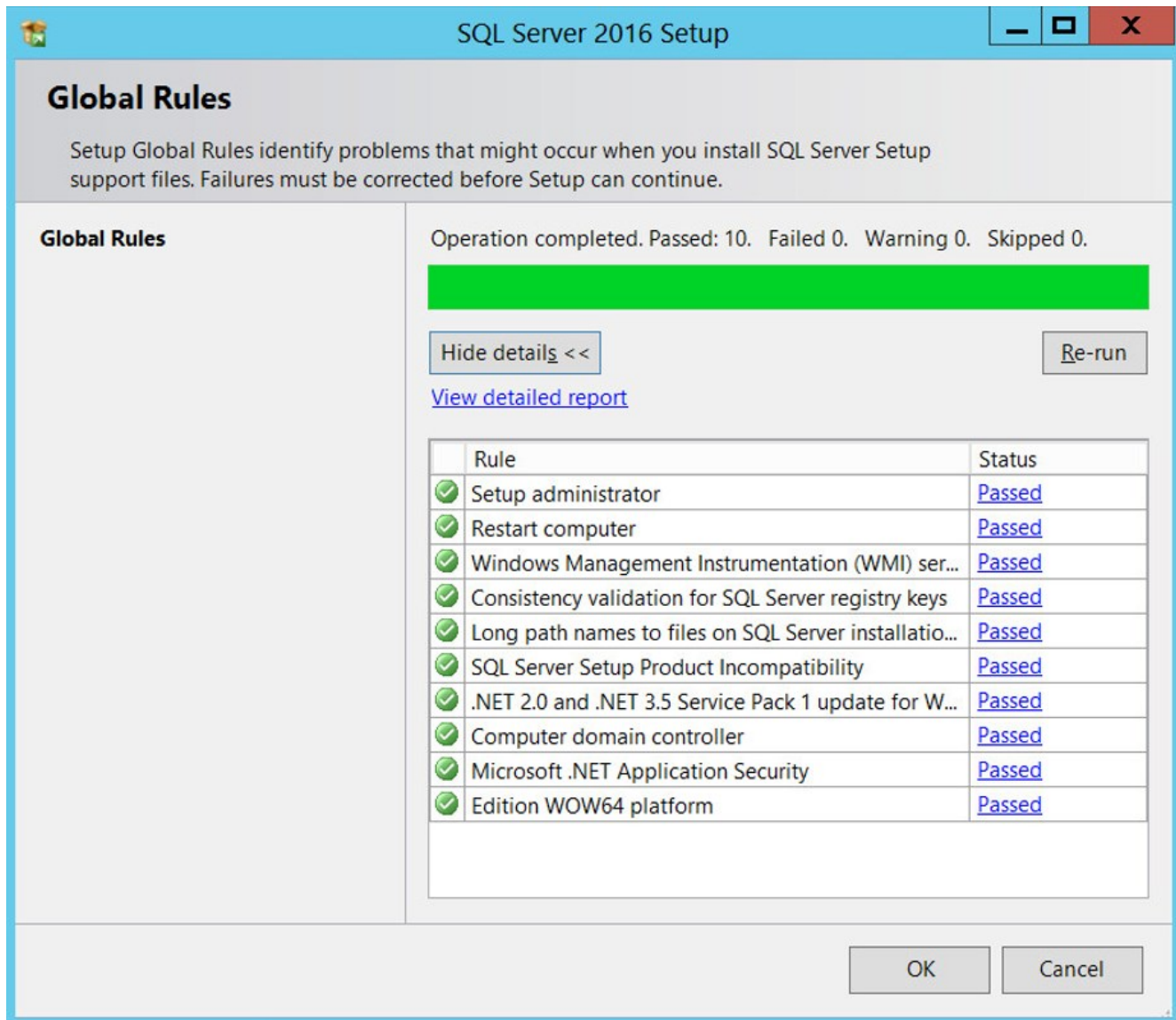
- Installed a basic stand-alone instance of SQL Server 2016 Standard with the minimum features necessary for SQL Server.
- Installed SQL Server Management Studio for managing the local database.
- Created a local windows account to be used as the database account for the Secret Server installation.

Installing SQL Server 2016 Standard Edition

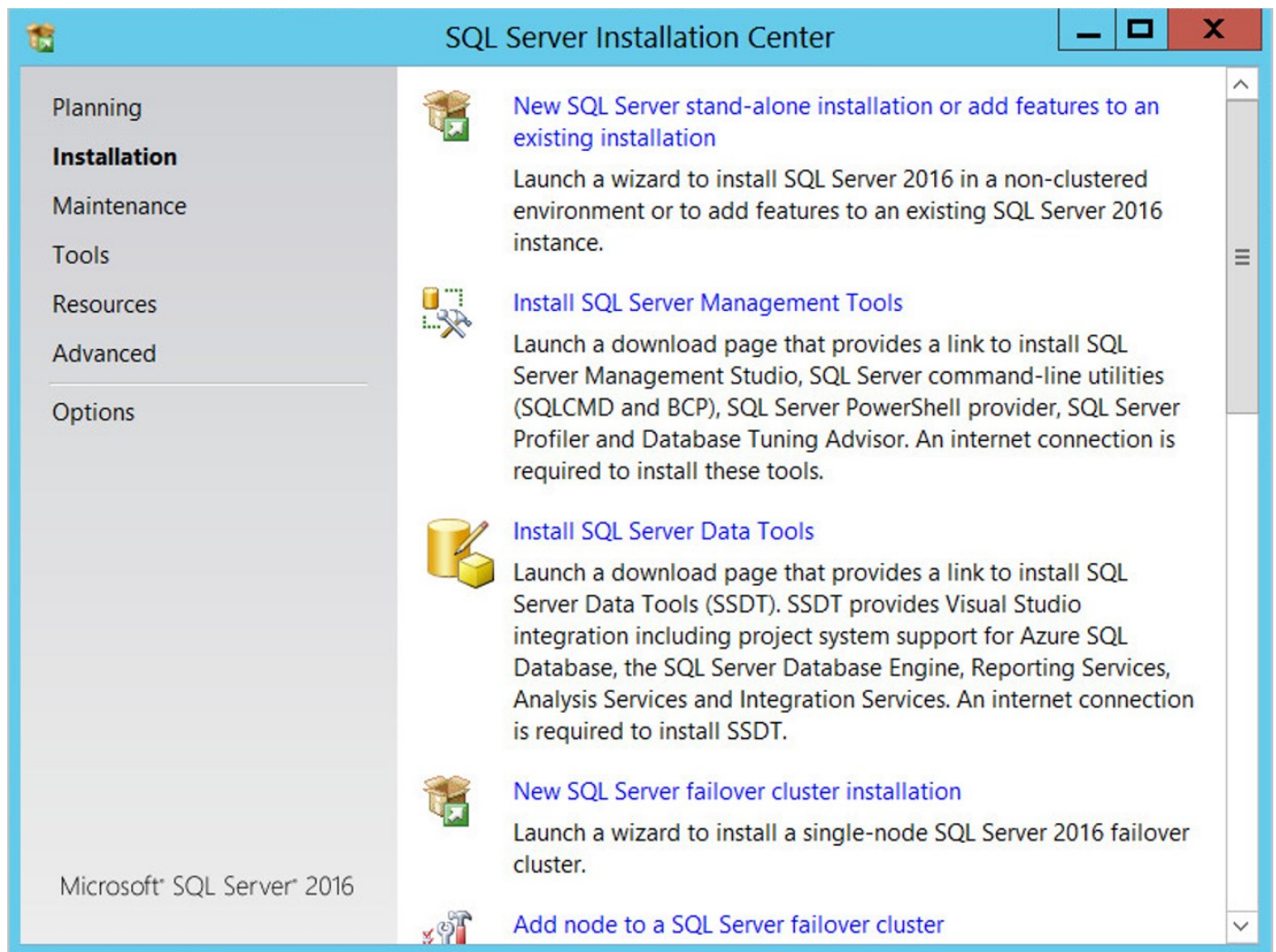
1. Launch the SQL Server installer from CD or file download.



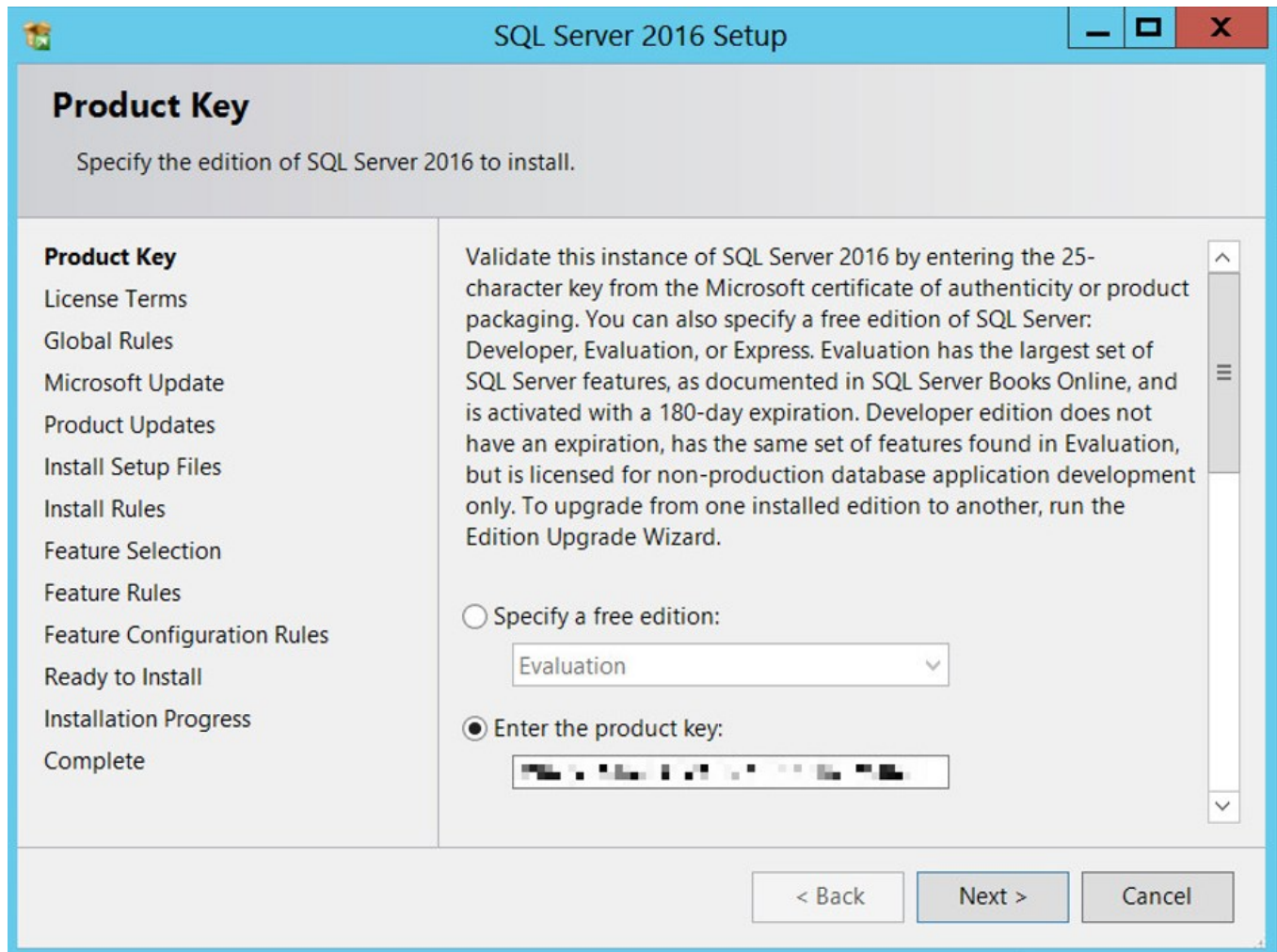
2. Click **System Configuration Checker** to run a utility that checks for conditions on your server that could prevent SQL Server from installing.
3. When the tool launches, click the **Show details** button. A successful scan should look like the one shown below. If you encounter any issues, look at the detailed report, resolve the reported issues, and rerun the scan.



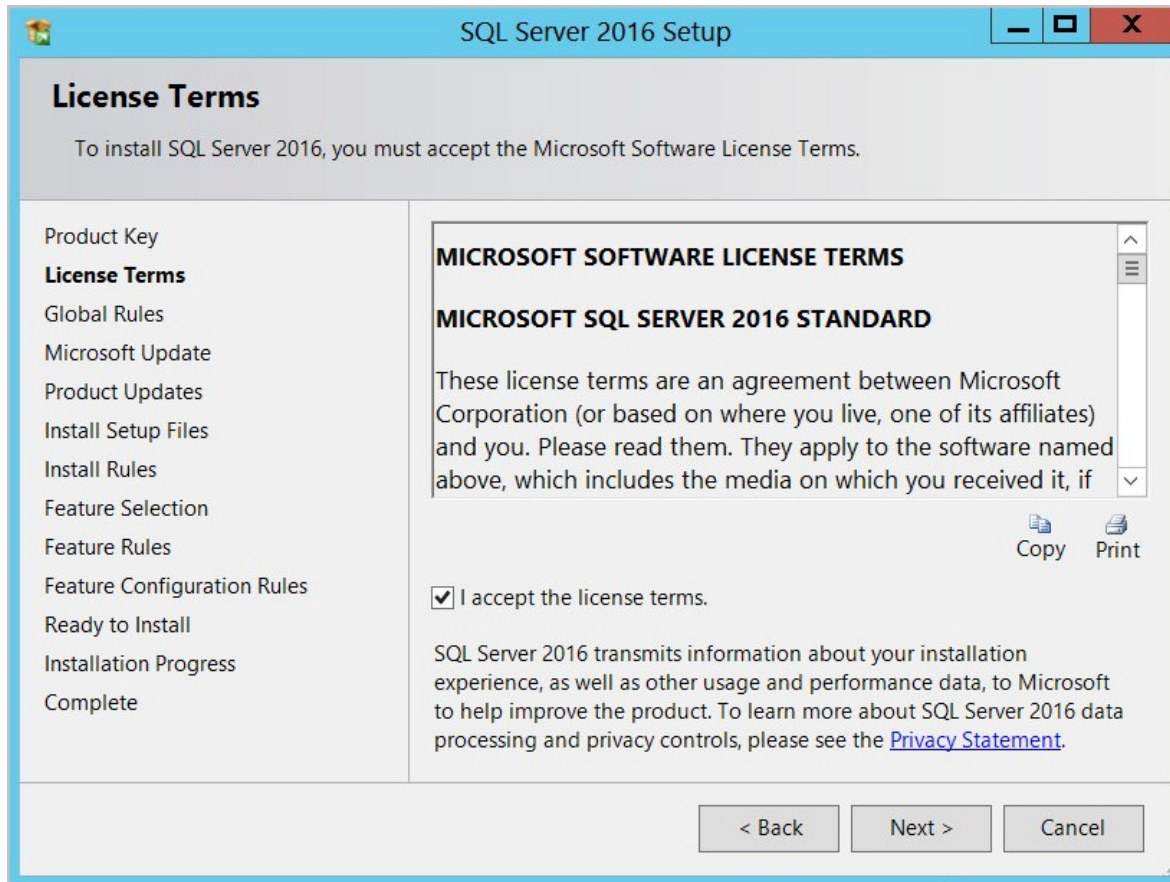
4. Click **OK** when done to return to the SQL Server Installation Center window.
 5. In the SQL Server Installation Center window, click **Installation**.
 6. Click **New SQL Server stand-alone installation or add features to an existing installation**.
-



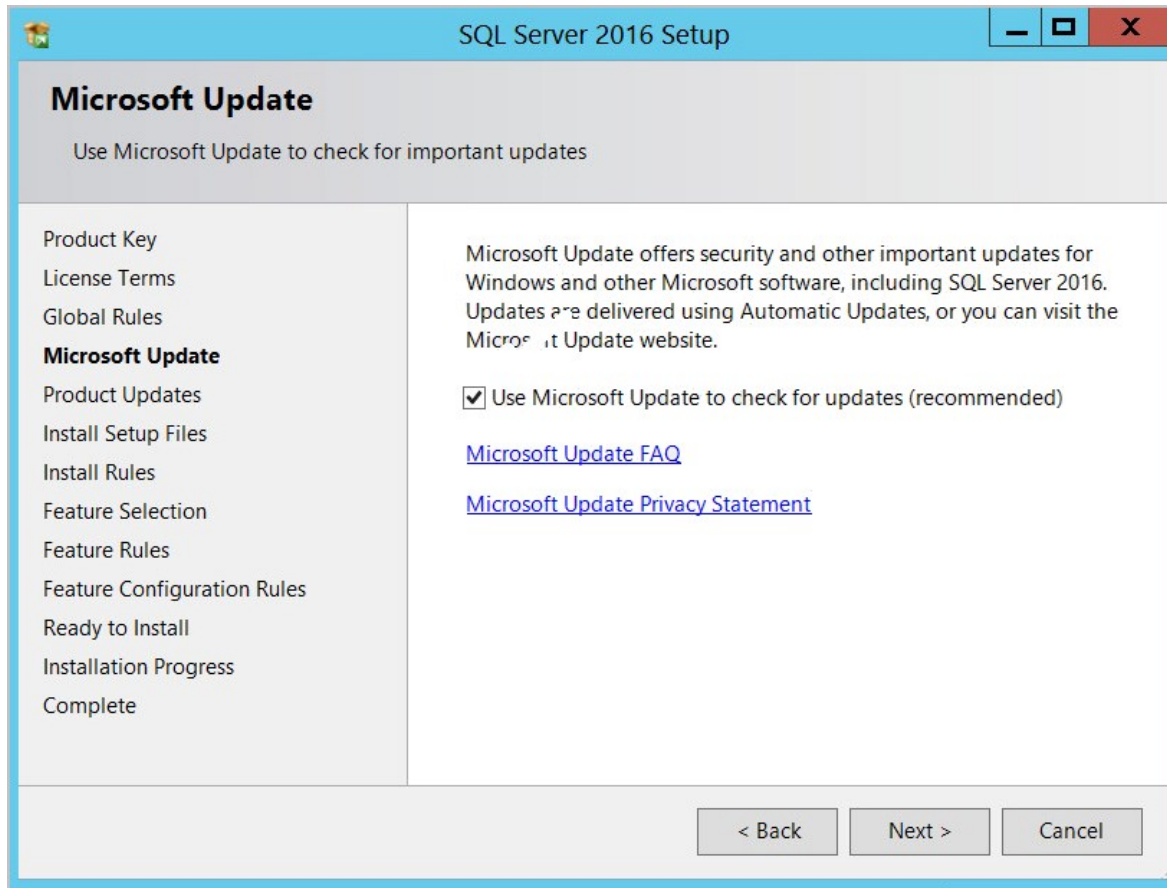
7. Enter your product key and click **Next**.



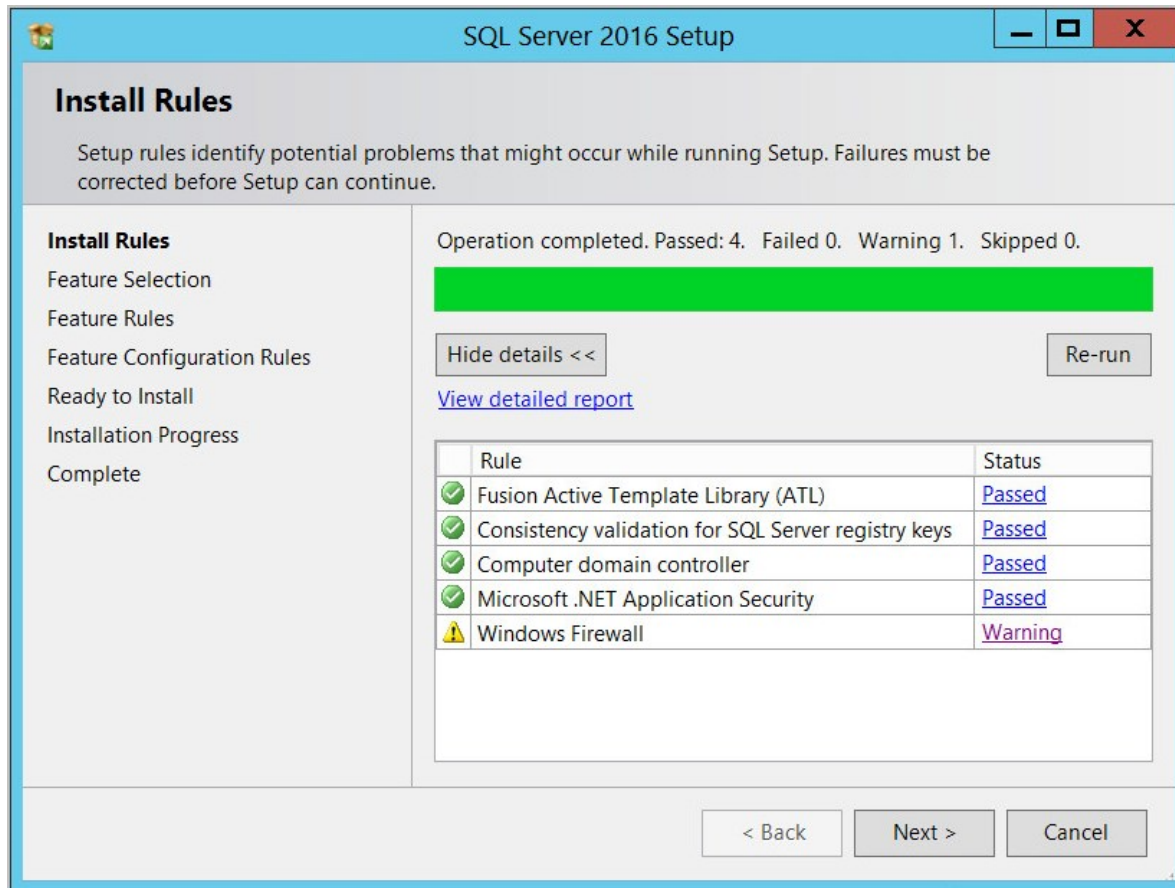
8. Check the **I accept the license terms** checkbox and click **Next**.



9. Wait for the Rule Check to run.
10. At the Microsoft Update step, check **Use Microsoft Update to check for updates (recommended)** if your software update process does not utilize automatic updates from Microsoft, skip this step.
11. Click **Next**.

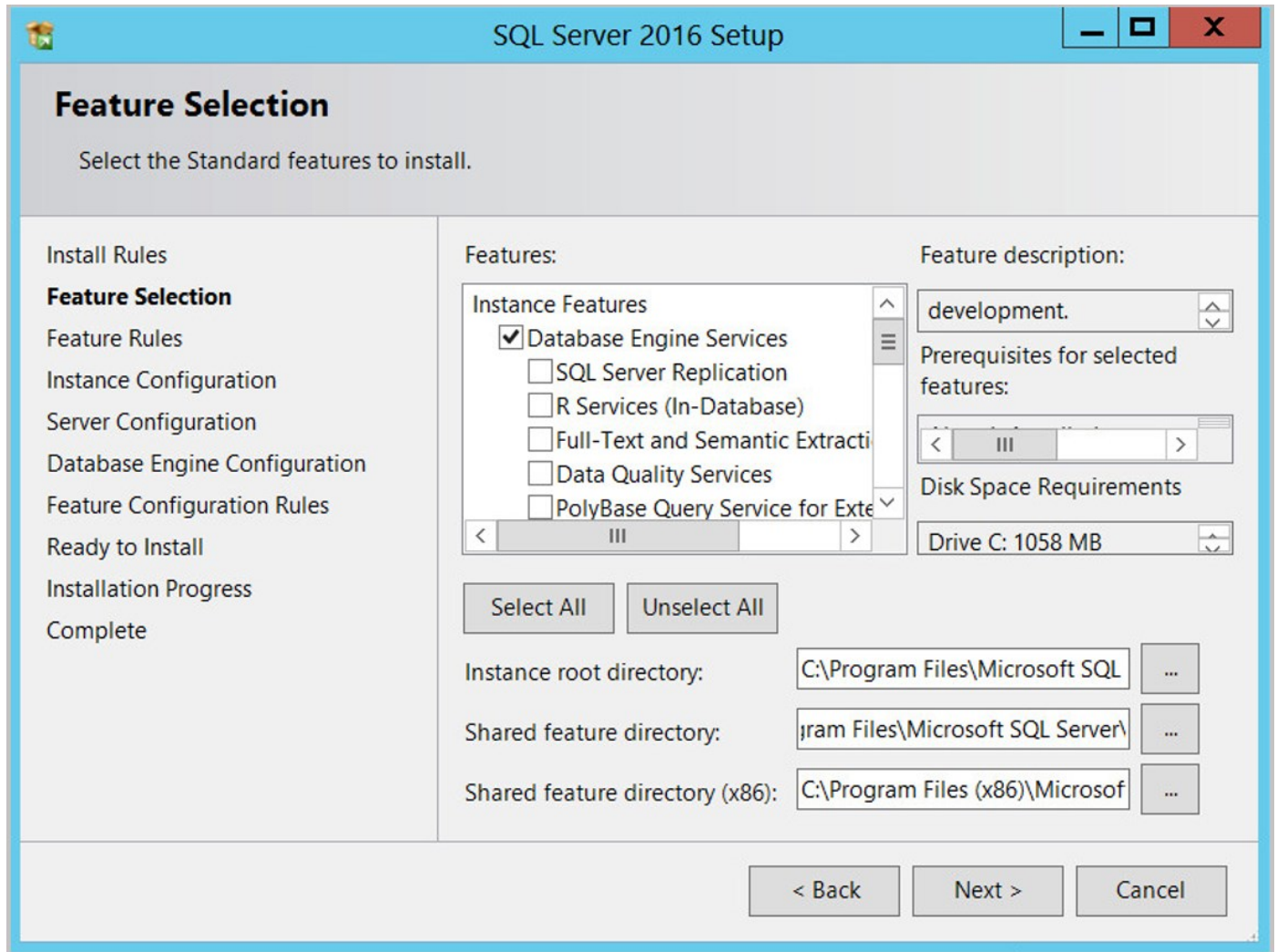


12. At the Product Updates step leave all defaults and click **Next**.
13. Wait for the "Install Setup Files" step to complete. Check that all operations pass (with the exception of Windows Firewall, which may return a warning status since the database will not be accessed from outside the firewall). The "Install Rules" screen (below) will appear upon completion. When the installation is successful, click **Next**.

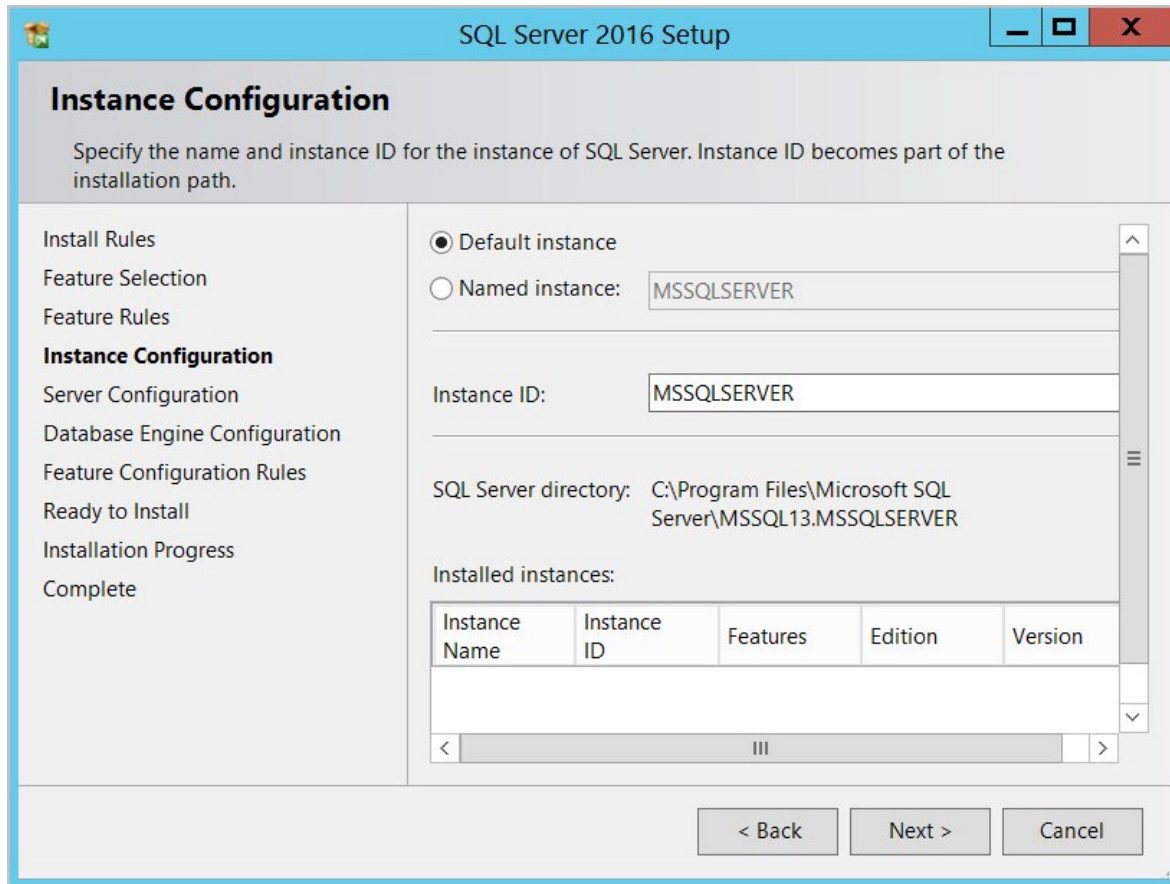


The only feature in Feature Selection necessary for Secret Server is Database Engine Services (see screenshot below).

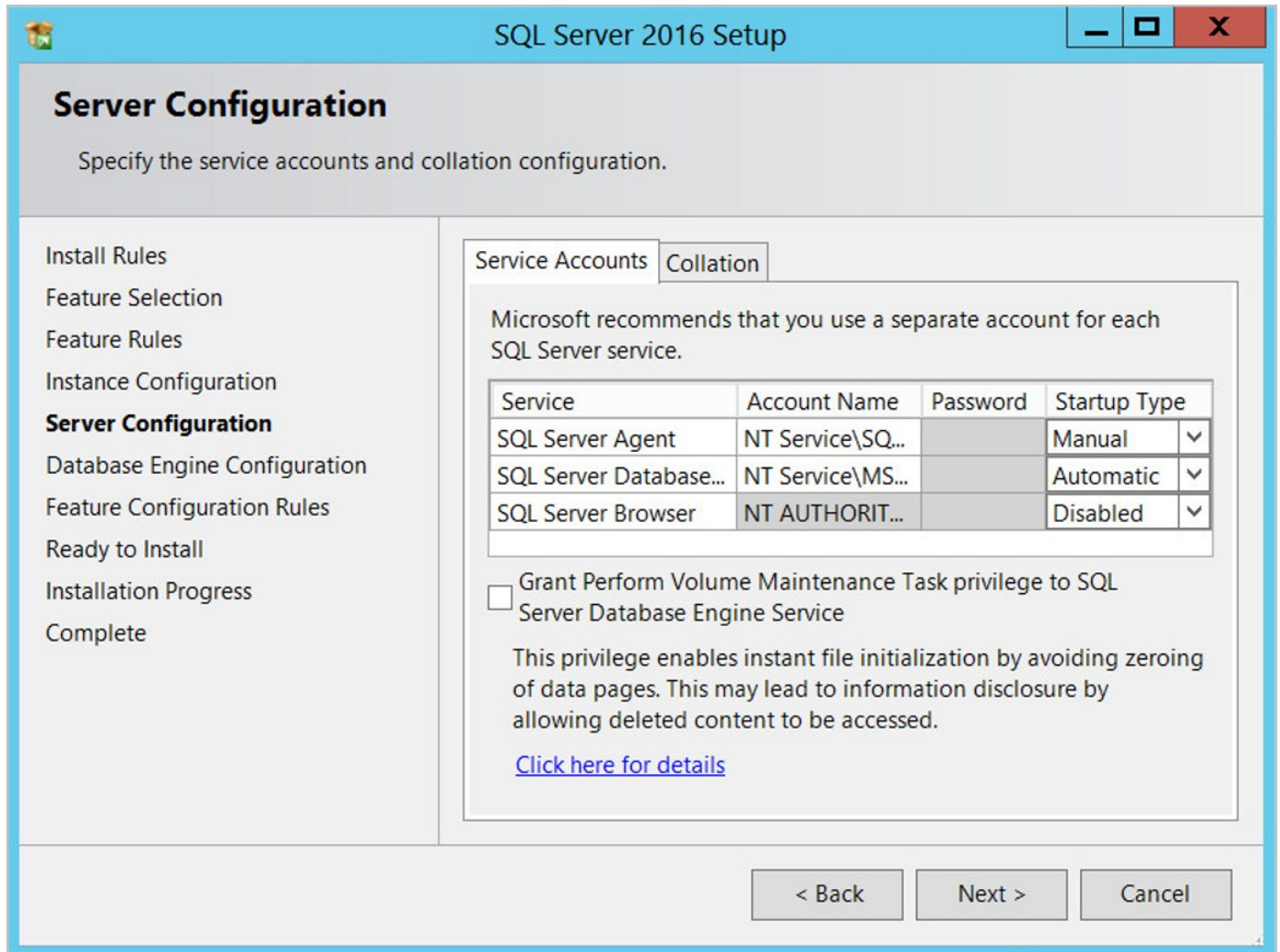
Unless you are using Geo-Replication (this is not in scope for Common Criteria standards) you can leave everything else unchecked. Leave the directory locations unchanged and click **Next**.



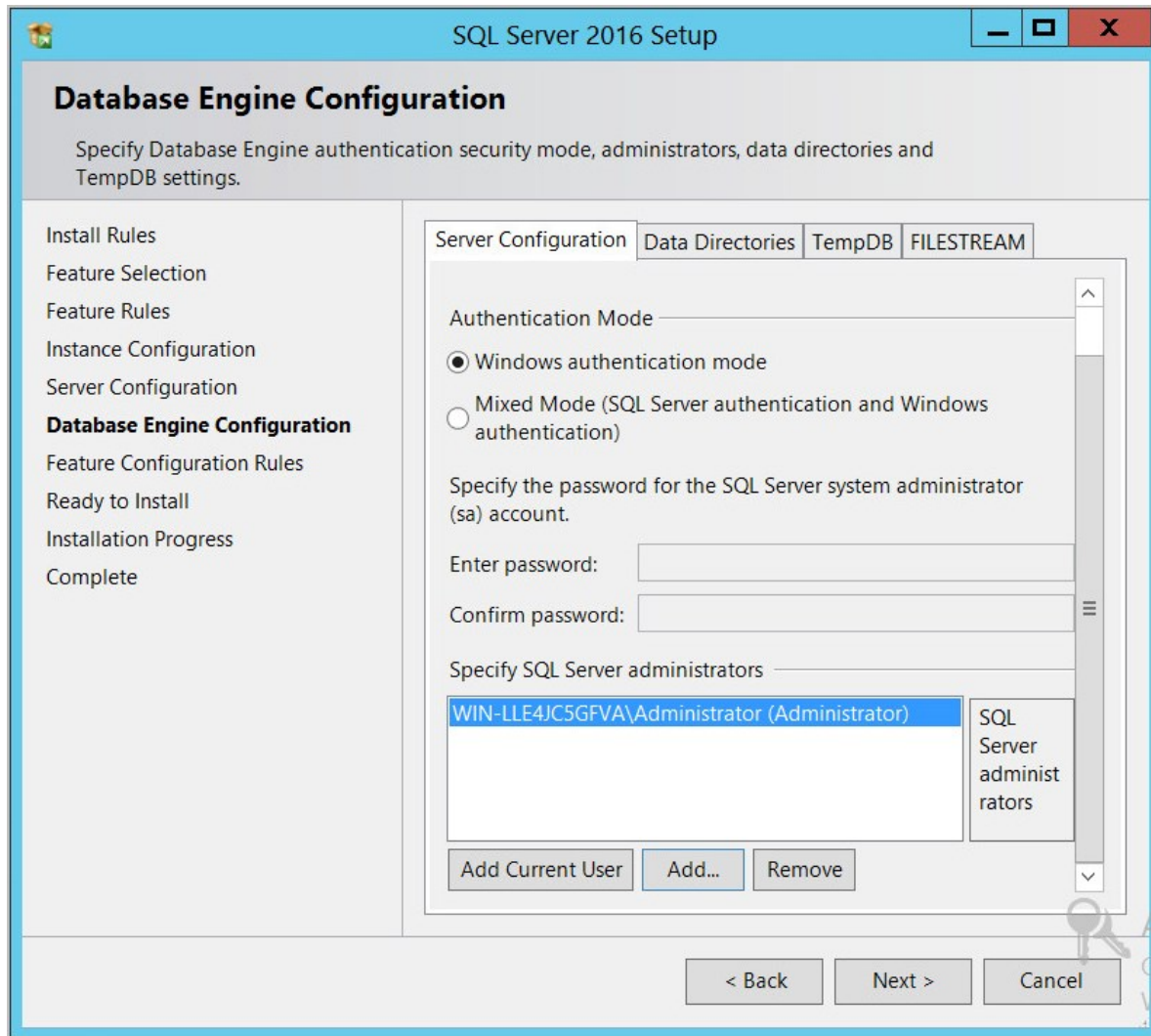
14. At the Instance Configuration step leave **Default instance** selected and click **Next**.



15. At the Server Configuration step (screenshot below) leave the defaults and click **Next**.

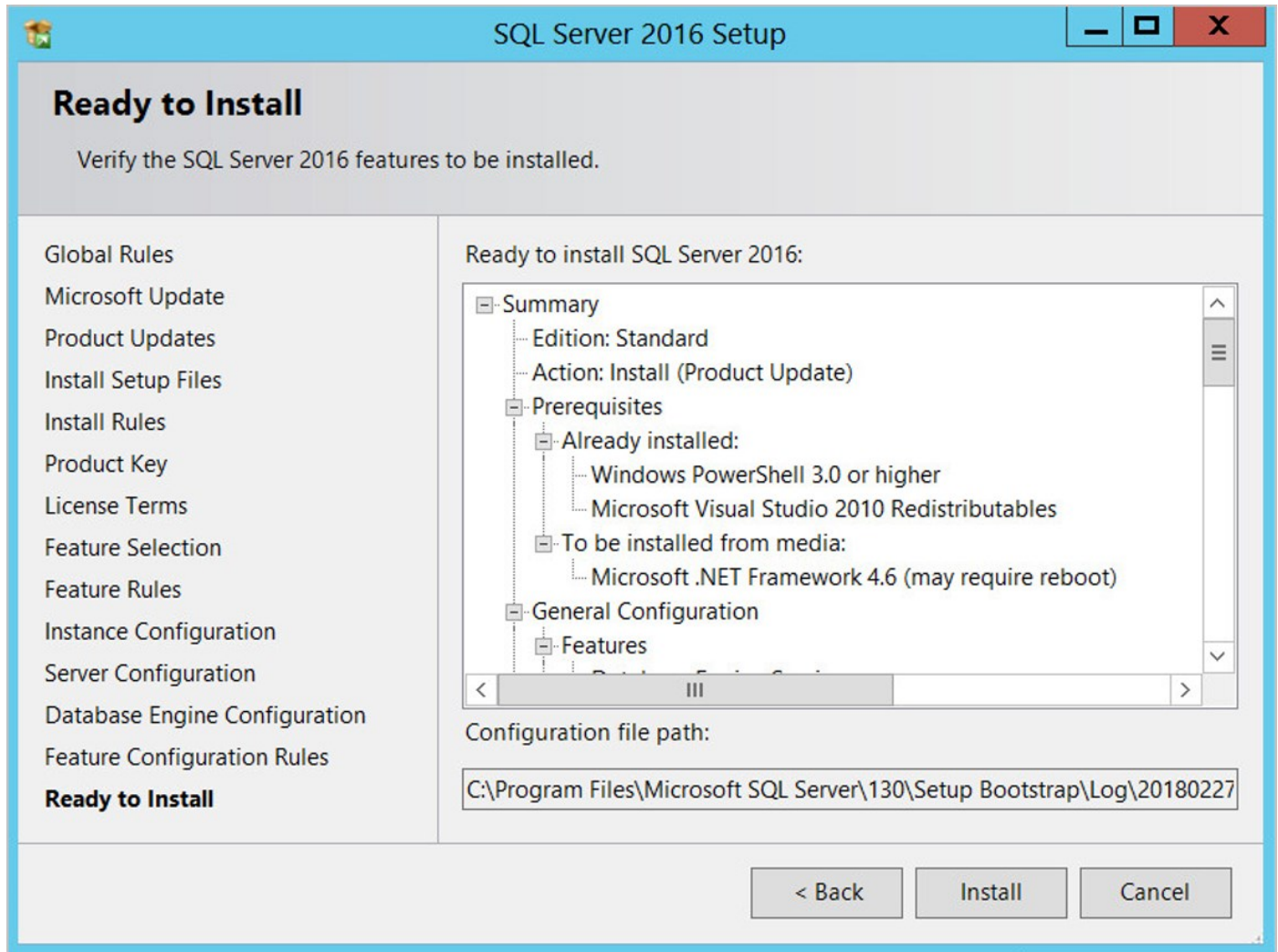


16. At the Database Engine Configuration step, make sure that **Windows authentication mode** is selected on the Server Configuration tab.
17. Scroll down and add at least one Local Windows or Active Directory user or group in Specify SQL Server administrators.
18. Click the **Add Current User** button to find and then **Add** the Administrator account.



You can leave the options in the remaining tabs at their default values, or to store the database and log data in a different drive or directory, change the file locations in the **Data Directories** and **TempDB** tabs.

19. Click **Next**.
20. Wait for the setup wizard to progress to the Ready to Install step and click **Install**.



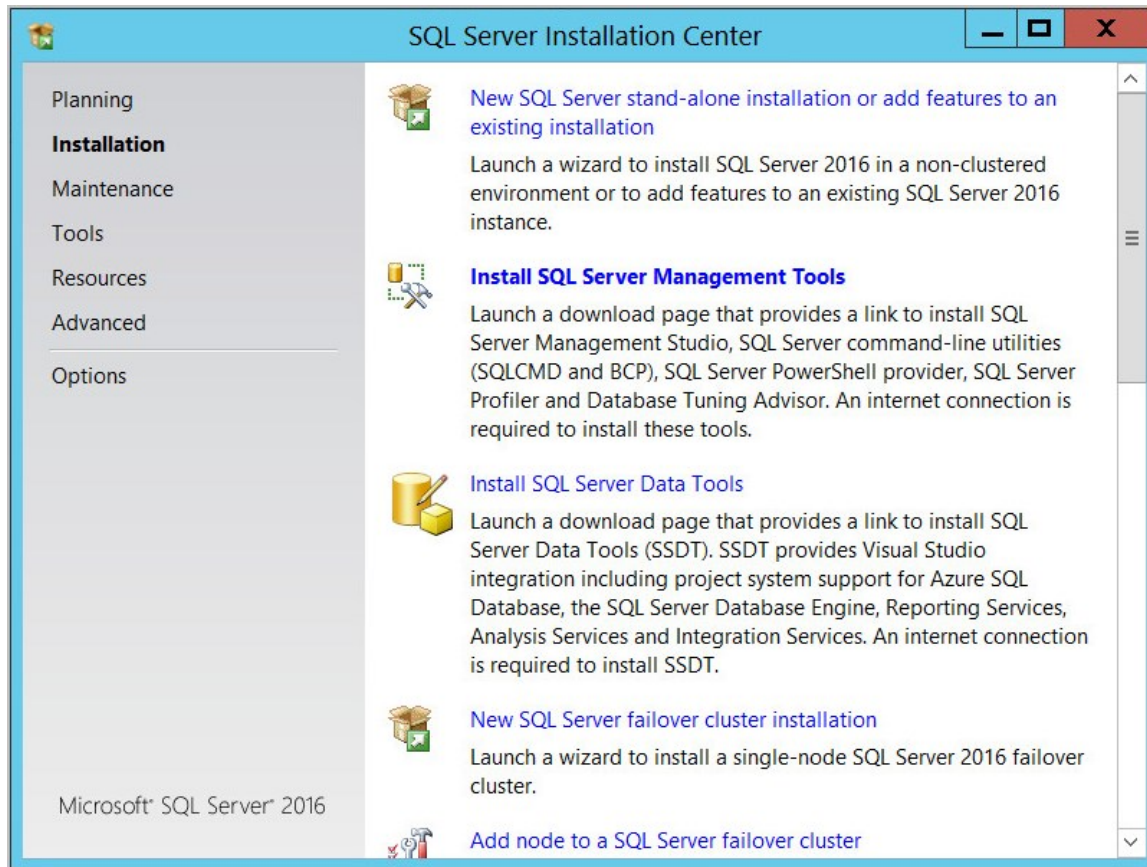
21. Wait for the installation to complete. This may take several minutes.

22. Click **Close**.

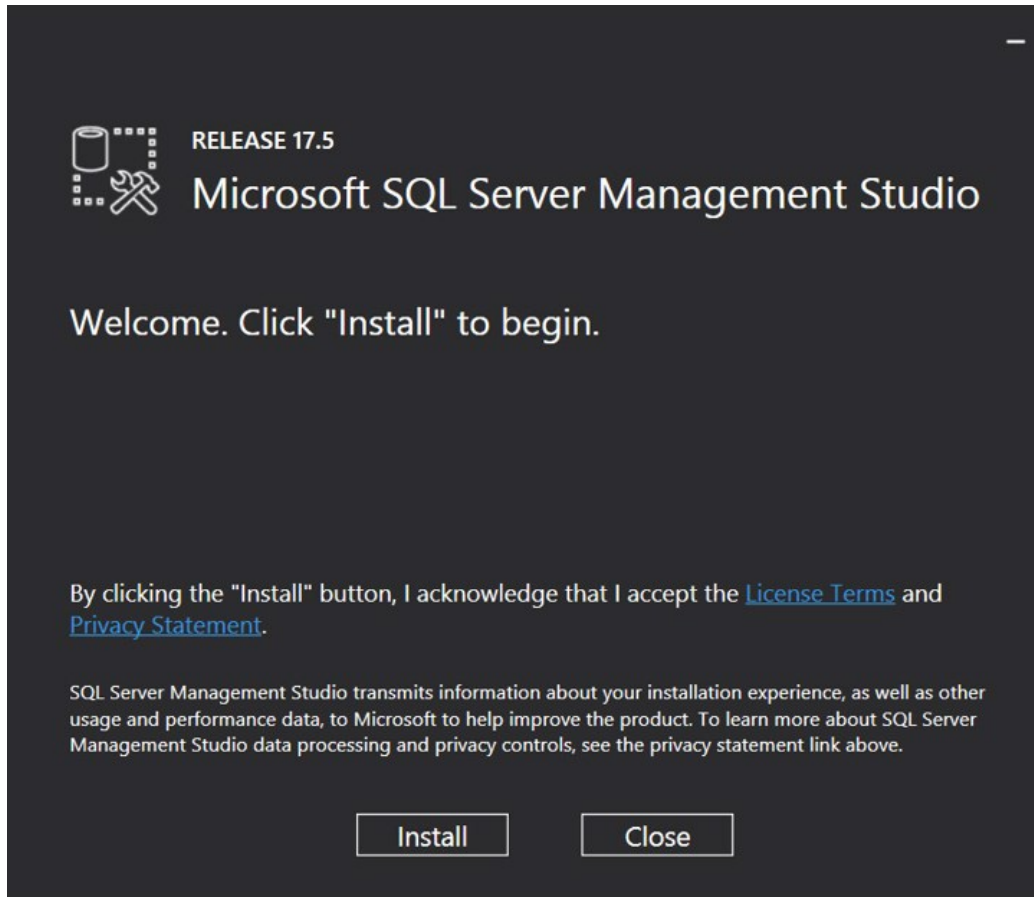
You may be prompted to reboot the computer before proceeding.

Installing SQL Server Management Studio

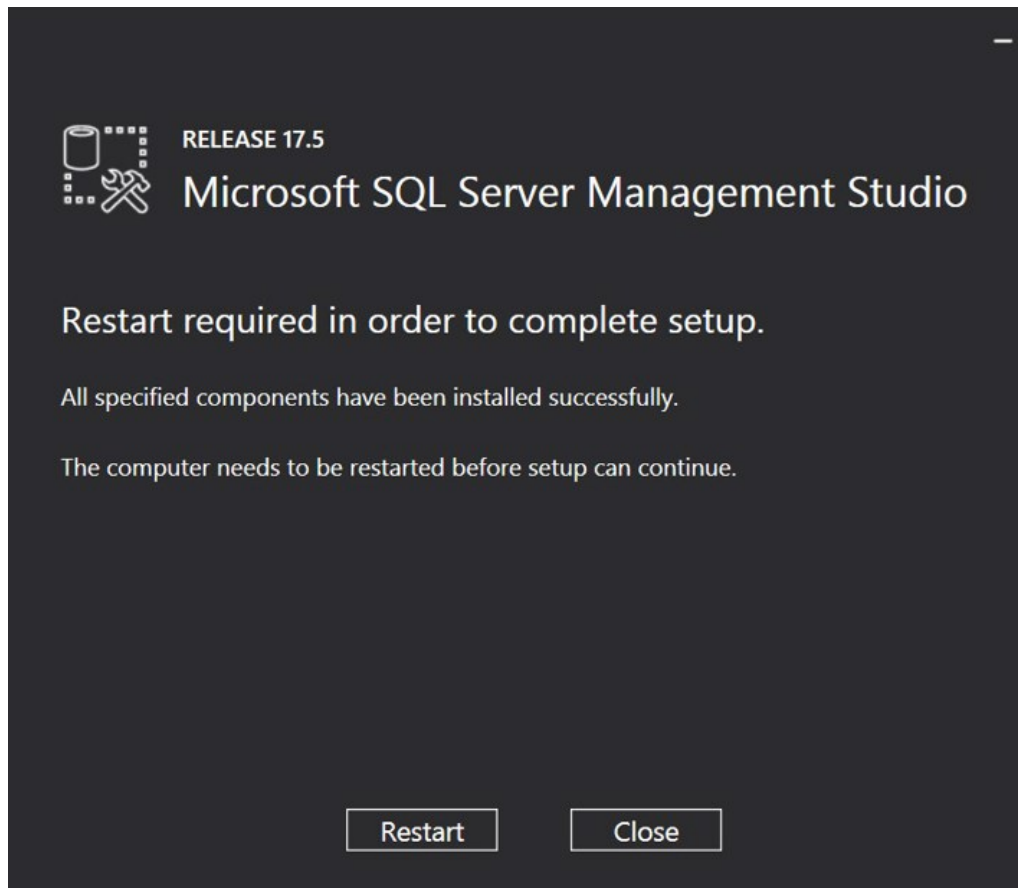
1. Back in the SQL Server Installation Center window, click **Installation**.
2. Click **Install SQL Server Management Tools** and wait for the web page to load.



3. Click the **Download SQL Server Management Studio 17.6** link (note: actual version number may be higher).
4. After the download is complete, click or double-click the file to begin the installation process.
5. When the SQL Server Management Studio installer starts, click **Install**.
6. Wait for the installer to complete. This may take several minutes.



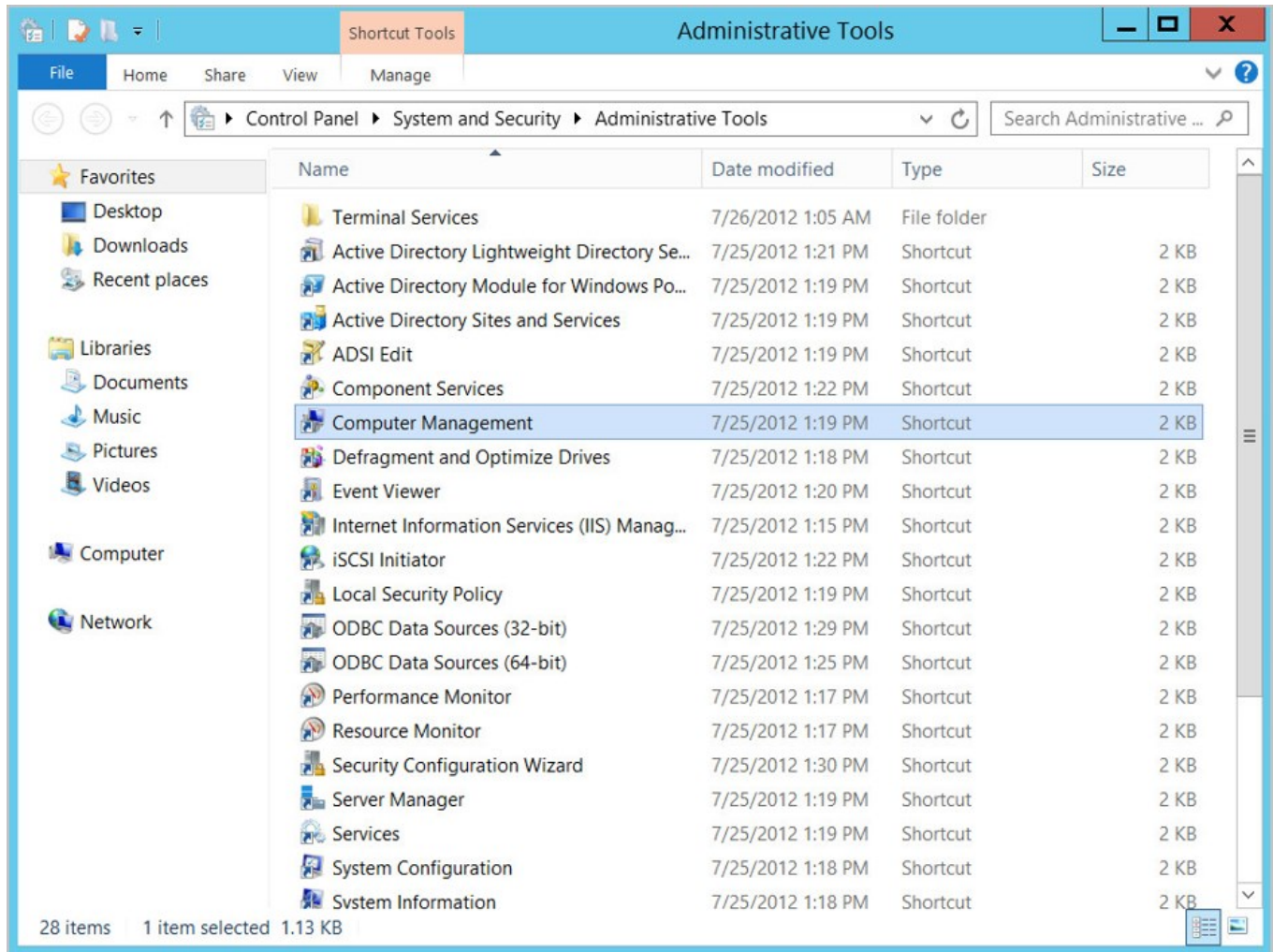
7. When the installation has completed, click **Restart** if prompted. Otherwise, click **Close**.



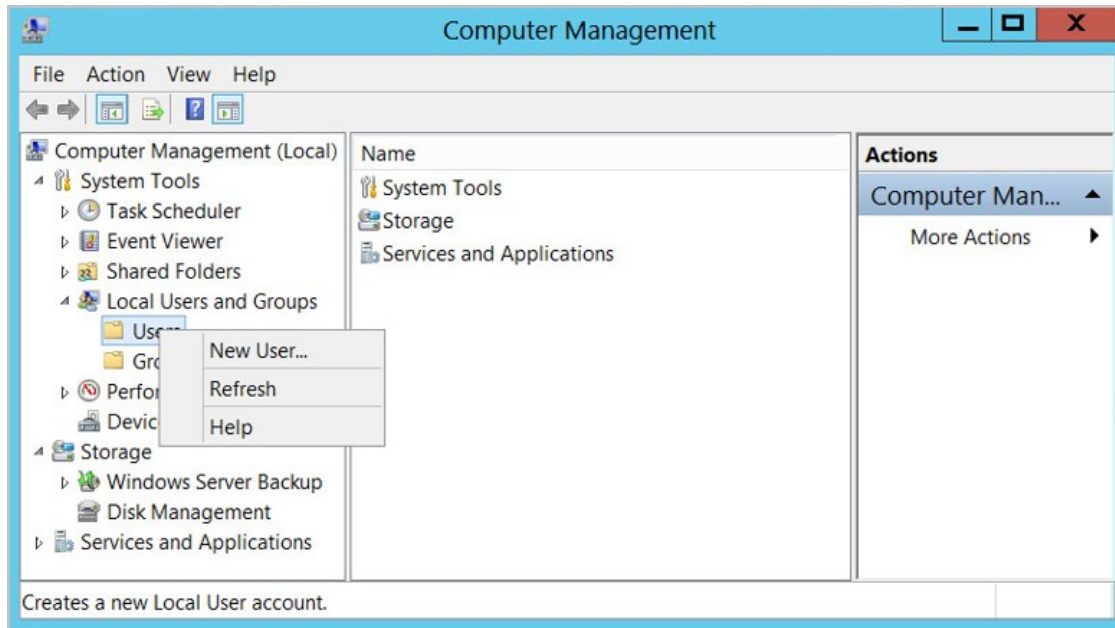
Configuring SQL Server for Secret Server Installation

Create the SQL Server User Account

1. Open **Control Panel | System and Security | Administrative Tools**.
2. Double-click **Computer Management**.



3. Expand the Local Users and Groups folder.
4. Right-click **Users**.
5. Select **New User....**



6. Give the user *any desired user name and a very strong password*.
7. Uncheck the default **User must change password at next logon**.
8. Check **Password never expires**.

This account will be added as a user in **SQL Server Management Studio** in the next step and its credentials will be saved to an encrypted file during Secret Server setup.

9. Click **Create** followed by **Close** when done.

New User ? X

User name: SecretServer_db

Full name: SecretServer_db

Description: Database user account for Secret Server

Password: ●●●●●●

Confirm password: ●●●●●●

User must change password at next logon

User cannot change password

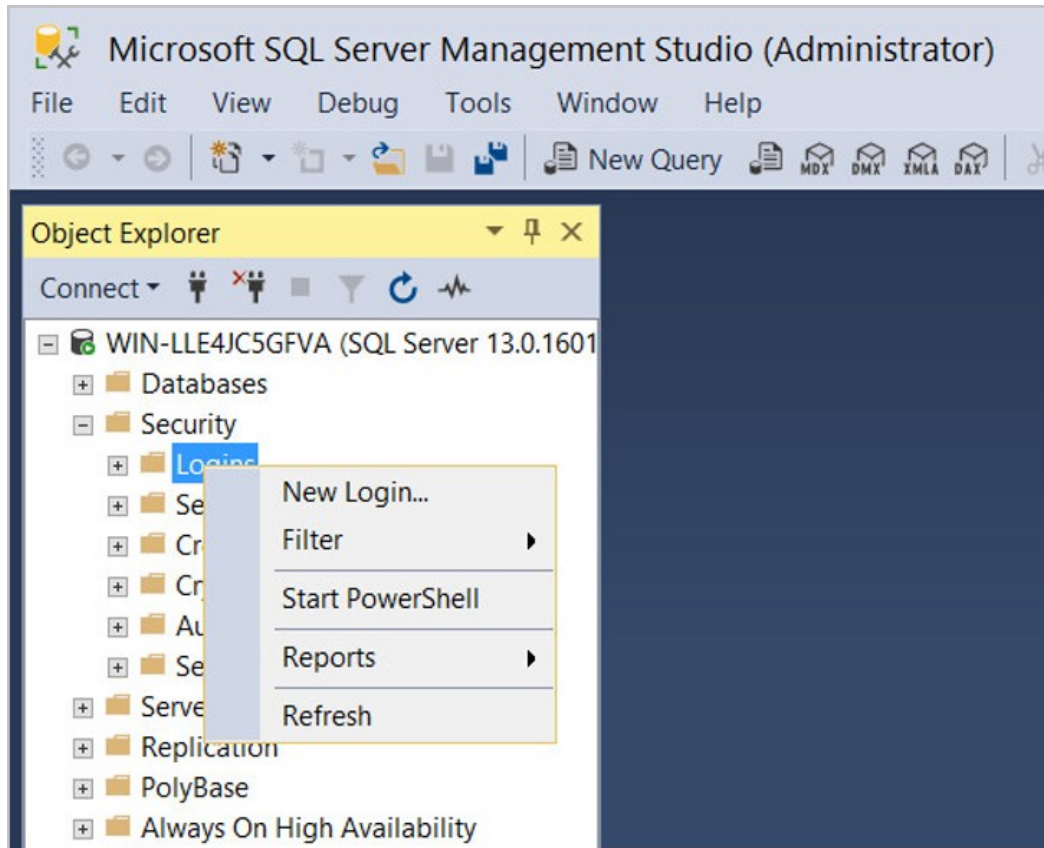
Password never expires

Account is disabled

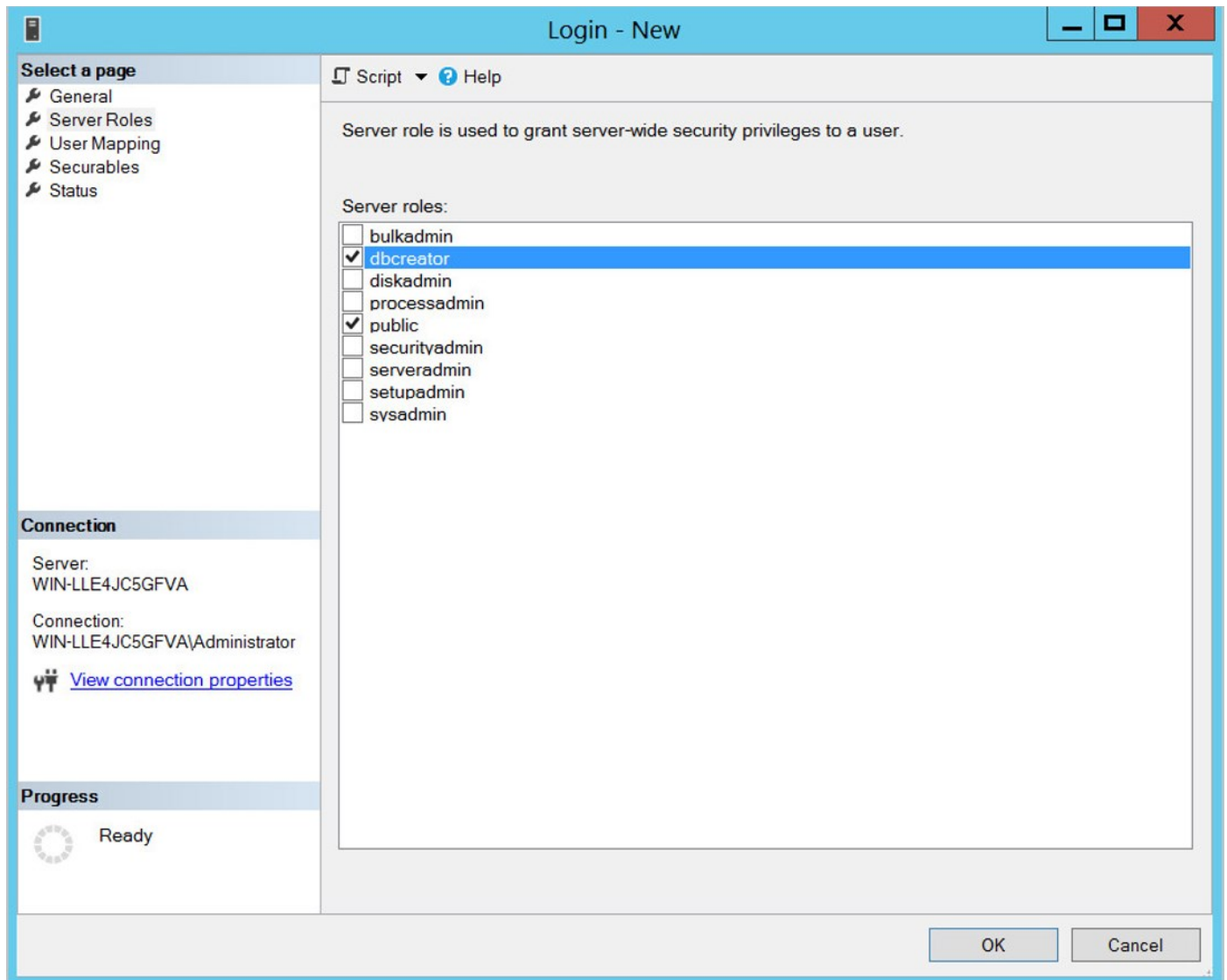
Help Create Close

Add the User Account to SQL Server

1. Start SQL Server Management Studio. This may take several minutes the first time the application is started.
2. Click the drop-down box beside **Server name**.
3. Click **Browse for more...**
4. Select the name of the database installed.
5. Click **Connect** to login using **Windows Authentication** (login as the user that you selected in the Database Engine Configuration step of the SQL Server installation).
6. Open the Security folder.
7. Right-click **Logins**.
8. Select **New Login...**



9. In the Login – New dialog, click the **Search** button.
 10. Enter the full name of the Windows account you created in the previous step.
 11. Click **Check Names**. If entered correctly, the user name should be underlined. If it is not, correct the name and repeat. When the name is entered correctly, click **OK**.
 12. Back in the Login – New dialog, click **Server Roles**.
 13. Check **dbcreator**.
 14. Click **OK**.
-

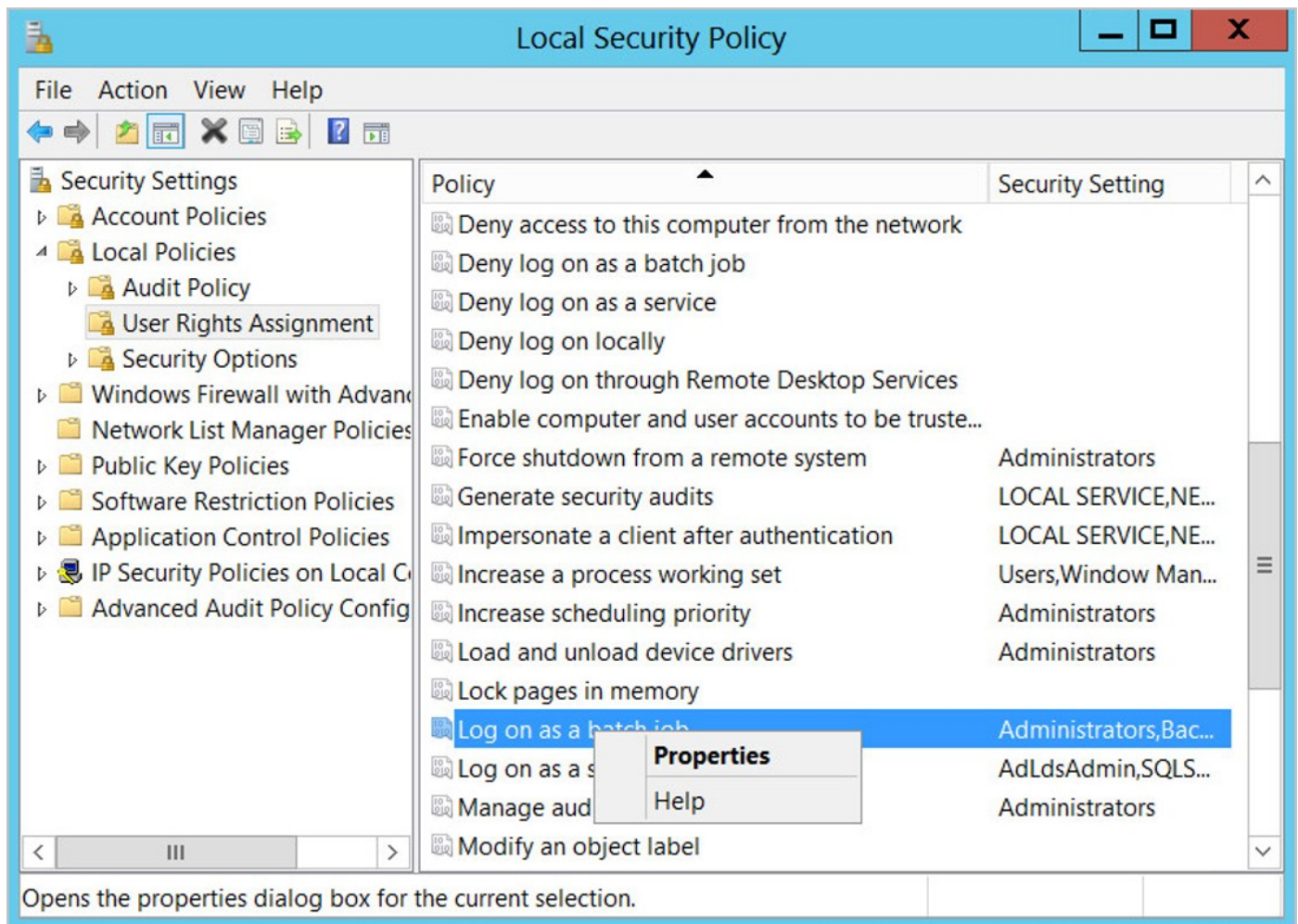


Additional Pre-Requisites for Secret Server

For Windows Authentication to access the SQL database, it is recommended that you create a service account.

To run the Secret Server IIS Application Pool with a Service Account

1. Open a command prompt window
2. Change the directory to your .NET framework installation directory using the "cd" command (either `C:\Windows\Microsoft.NET\Framework\v4.0.30319` or `C:\Windows\Microsoft.NET\Framework64\v4.0.30319`)
3. Type in `.\aspnet_regiis -ga < user name>` and press enter. Replace the field with the relevant value from the SQL Server user account created in section 2.2.3.1.
4. Give your service account "modify" access to C:\Windows\TEMP
5. Next open the Local Security Policy App from your start menu. Grant batch logon permissions to your service account:



1. Open the Local Security Policy Console (Search- > secpol.msc- > [Enter]).
2. Expand **Local Policies**.
3. Click **User Rights Assignment**.
4. Right-click **Log on as a batch job**.
5. Click **Properties**.
6. Click **Add User or Group** and add your service account.
7. Click **OK**.

NOTE: If you utilize Group Policy to enforce the "Log on as a batch job" and have group managed service accounts, take note that this will overwrite any local permissions to the "Log on as a batch job" on all computers that have the policy applied. Utilizing the local security policy is a safer option if you are not sure about your usage across your domain.

6. Grant **Impersonate a client after authentication** permission to the service account under "User Rights Assignment" the same way "Log on as a batch job" was assigned above.
7. If you now get a "Service Unavailable" after applying "Log on as a batch job" permissions, then you need to update your group policy settings (start- > run- > cmd, type in gpupdate /force) and restart the Windows Process Activation Service.

For information about configuring a Windows account for database access, please see [Accessing MS SQL Server with IWA](#).

Secret Server Installer

Downloading and Verifying the Installer

You can download the **Government Edition Installer** files from the Delinea Portal Downloads page after logging into the Delinea Support Portal.

To meet Common Criteria standards, you must run the ThycoticSetup.exe file as an Administrator on a 'clean' Secret Server web server, with a full, licensed version of SQL on the same database.

Installer Verification and Authentication

Delinea generates SHA1 and SHA256 hashes for each software release, and they are available for download from a link in the [Secret Server Setup](#) chapter of the Secret Server documentation.

To verify that the installer has not been altered from our official release, run the Get-FileHash PowerShell command against your copy of the installer and check to make sure that the hash generated from that file matches the official hash value we publish. Assuming the copy of the installer you are checking is in the C:\temp\ folder, the command you would run is:

```
Get-FileHash -path c:\temp\gov\ThycoticSetup.exe -algorithm SHA256
```

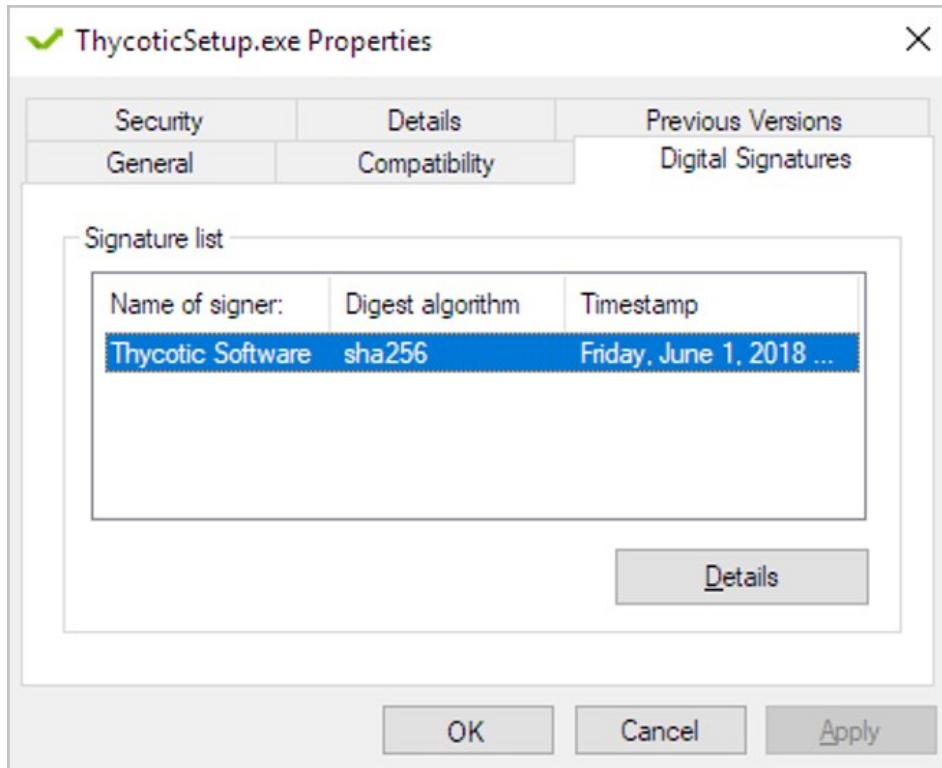
This will generate the hash and output the results. The output will look something like this:

Algorithm Hash Path

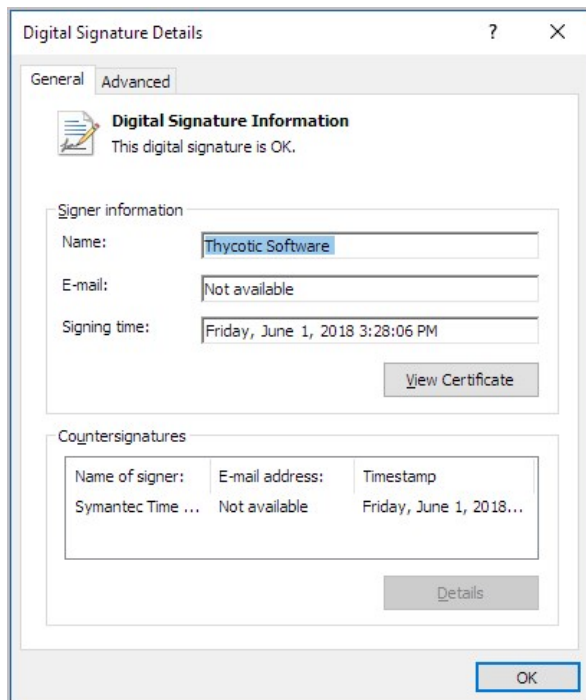
```
SHA256 588D130AE766C34E53072DFA4B900A372A99B5E413B3D1667CB17C2738ACD046 C:\temp\gov\ThycoticSetup.exe
```

The second step to take when checking the authenticity of the installer is to check the code signing certificate to confirm that it is signed by Delinea and has a valid certificate chain. To do this:

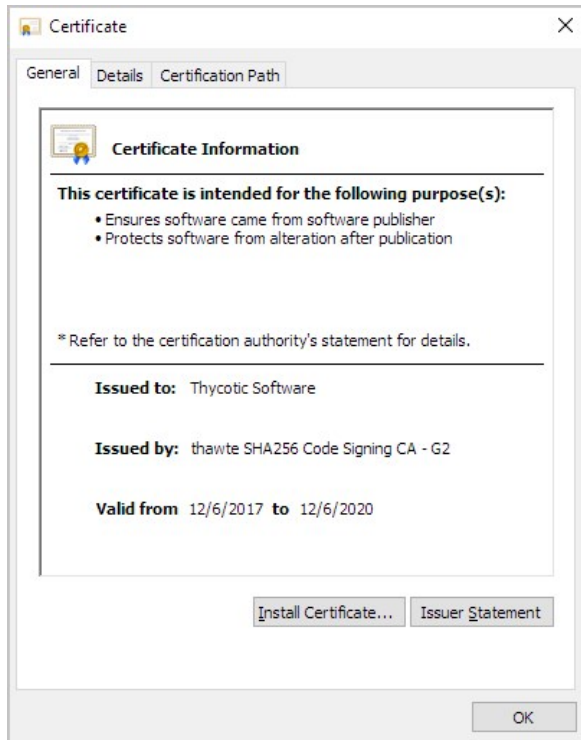
1. Right-click on ThycoticSetup.exe and select "Properties" from the context menu.
2. Switch to the "Digital Signatures" tab.



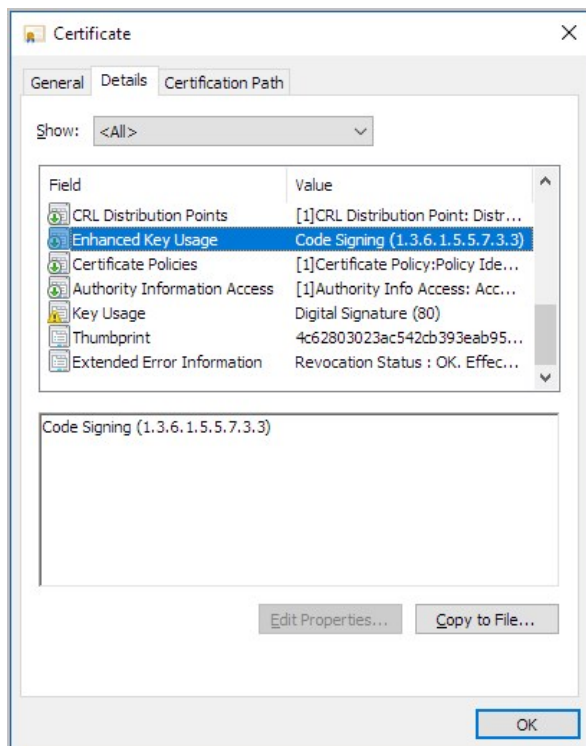
3. Verify that the name of signer is "Delinea Software" and make note of the date in the Timestamp. Select that entry in the Signature list and click "Details."



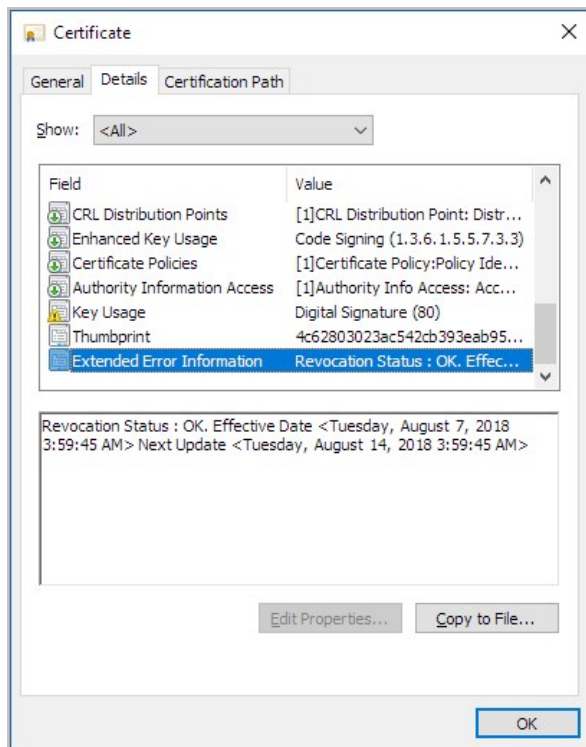
4. Click "View Certificate" to view the signing certificate.



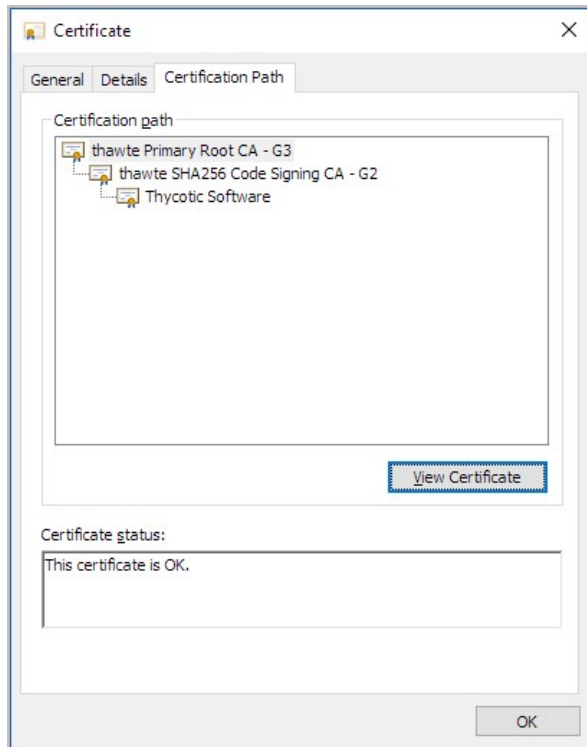
5. Verify that the certificate is issued to "Delinea Software" and that the signing timestamp from the signature list is within the "Valid from" date range.
6. Switch to the "Details" tab.



7. Verify that the “Enhanced Key Usage” field lists that the key is valid for code signing, as shown above.
8. Select the “Extended Error Information” field.



9. Verify that the certificate's revocation status is “OK”
10. Switch to the **Certification Path** tab.



11. View each certificate in the chain above “Delinea Software” and verify that the certificates are all valid.

Running the Installer

If the following components are not already installed in your environment the Delinea Installer will automatically set them up for you:

- Microsoft .NET 4.6.2+ Framework or newer
- Application and Web Server roles enabled in IIS
- SQL License (Login with *dbcreator* rights needed)
- SSL certificate (Troubleshooting self-signed certificates is not supported by Delinea, we recommend a Certificate Authority for the certificate)

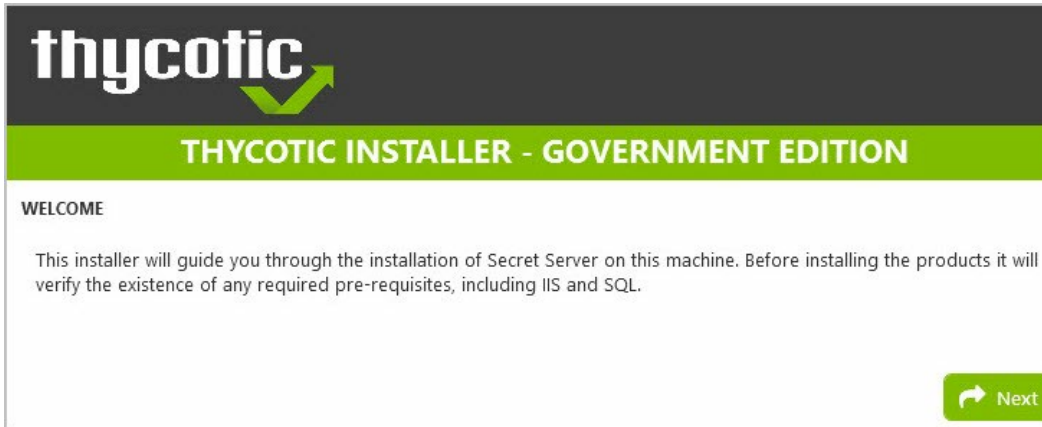
We strongly recommend that your Web Server is “clean,” meaning that you should have no other running applications installed on this server to avoid complications between multiple applications

If an existing SQL server does not exist in your environment already, the installer will give you the option to install SQL Express, however, *using SQL Express is ONLY intended for trials or sandbox environments and is not acceptable for production use.*

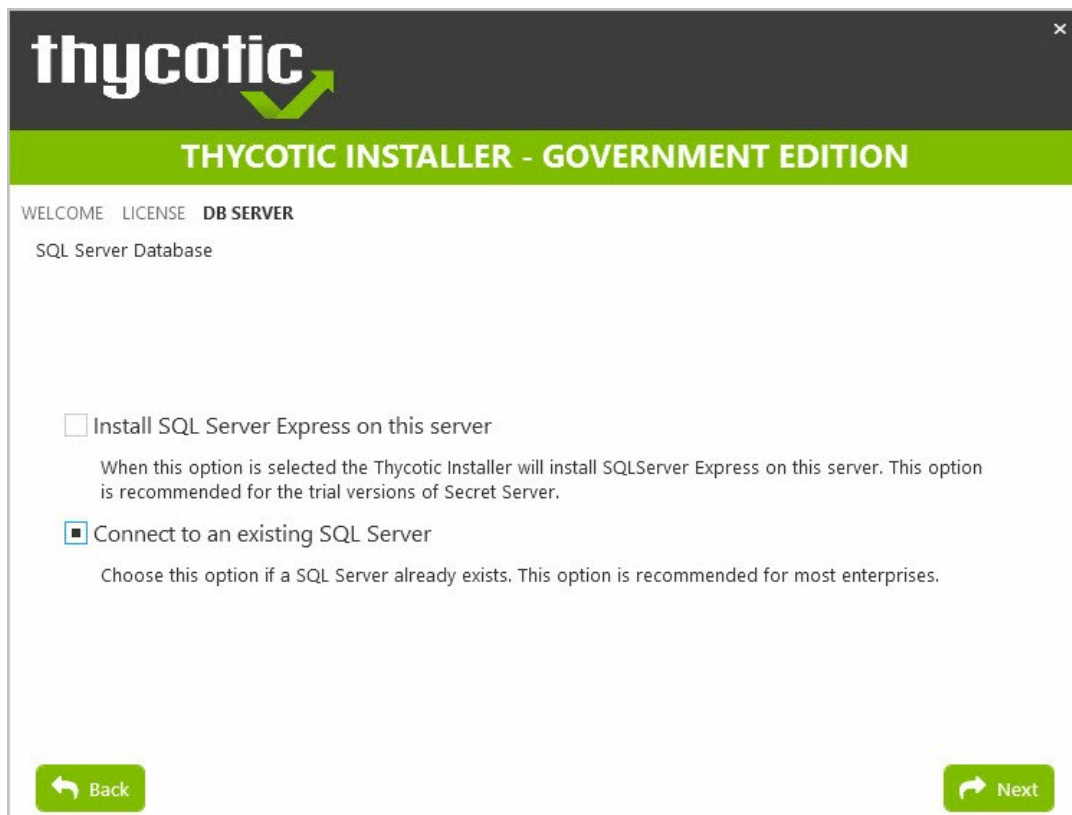
If your web server does not already have an SSL certificate, the installer will create and install a self-signed certificate, however, *self-signed certificates are NOT recommended for production environments.*

The installer will walk you through the steps required to perform your **Secret Server – Government Edition** install. The following steps comprise a shortened version for the install, highlighting Common Criteria Compliance requirements. For a comprehensive step-by-step guide see our KB on **Installing Secret Server – Government Edition** at:

<https://thycotic.force.com/support/s/article/SS-Installing-Secret-Server-Government-Edition> **INTERNAL ONLY SUGGESTS**
<https://thycotic.force.com/support/s/article/System-Requirements-for-Secret-Server>

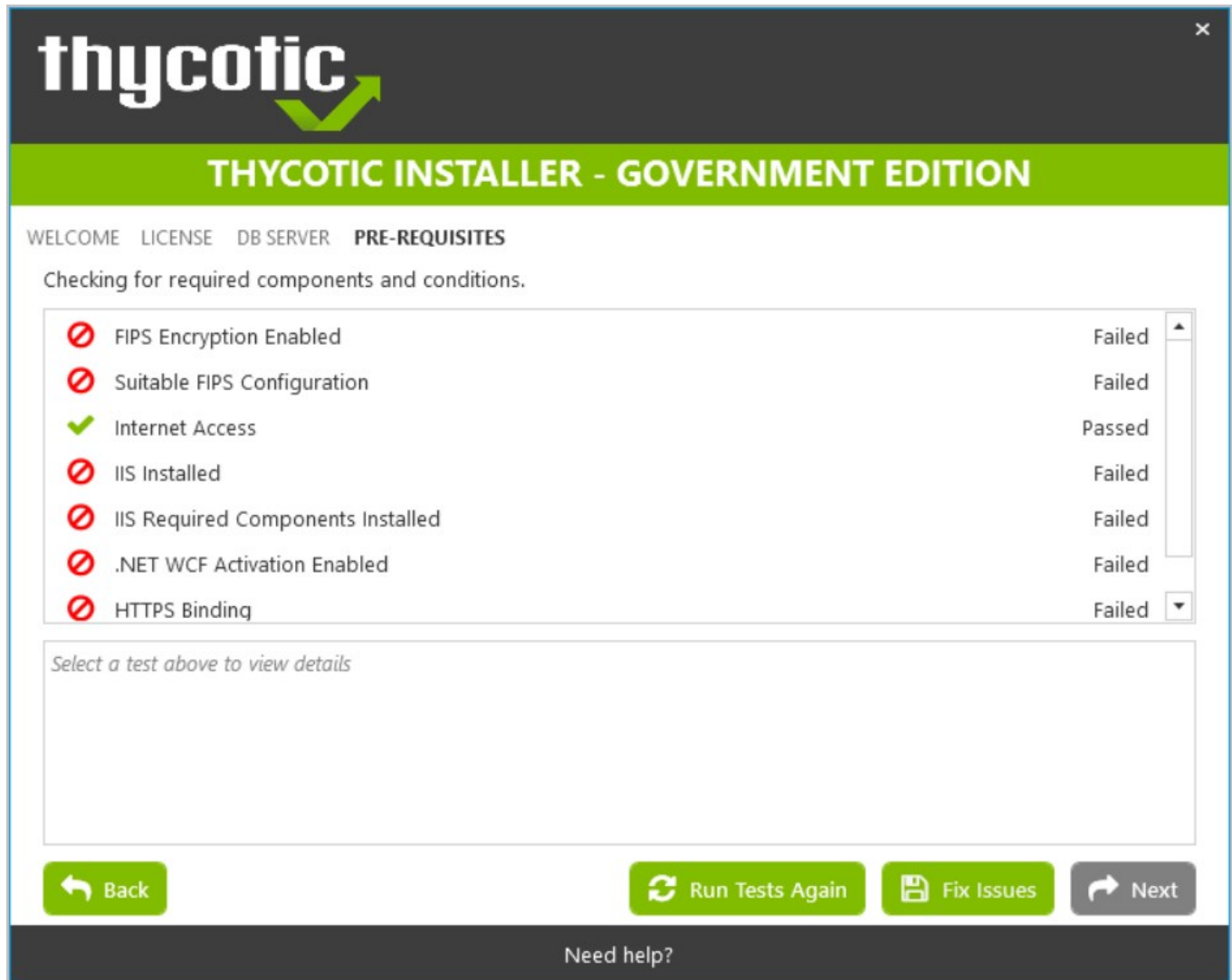


1. Click on the **Government Edition** of Secret Server to download installer (see section 2.3.1 for link) and execute the installer on your joint Web Application + Database Server. The first installer page you will be presented with is the **Welcome** splash screen. Click **Next**, then read and **Accept** the End User Licensing Agreement.



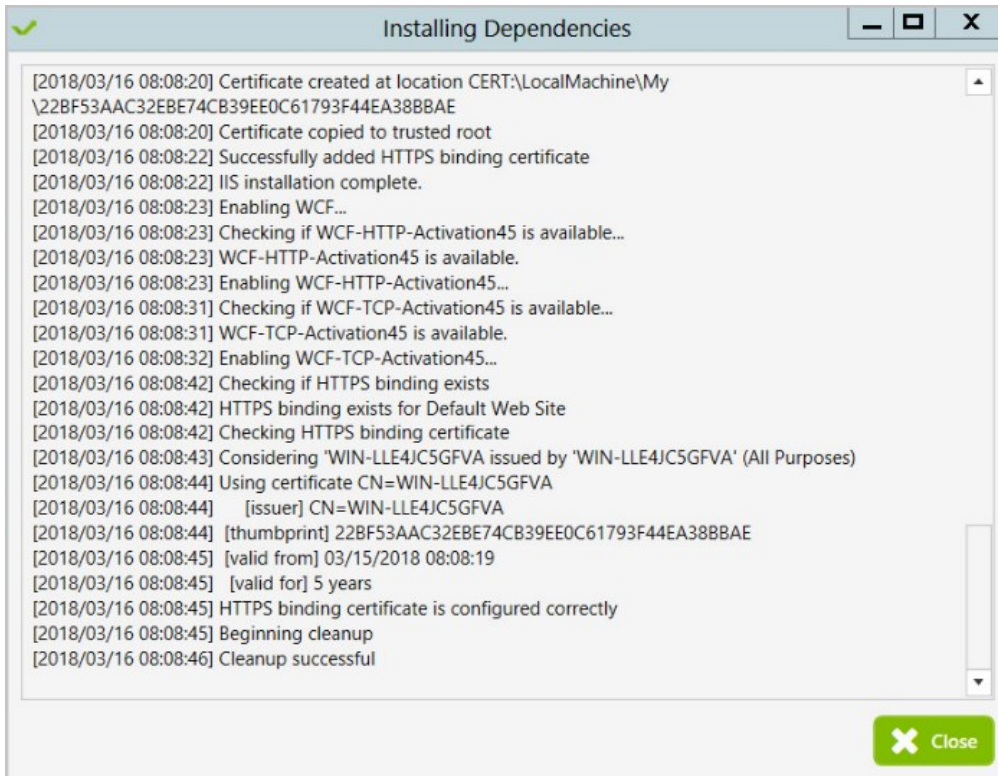
2. For Common Criteria compliance, select **Connect to an existing SQL Server**. To uphold compliance a full licensed version of SQL must be installed locally (e.g. on the same server where Secret Server is installed).

Note: Installing SQL Express is only intended for the trial version of Secret Server and is *NOT* suitable for production environments. If you choose "Install SQL Server Express on this server," the Delinea Installer will automatically install a fresh instance of SQL Express locally, but your instance will not be supported and will not be up to standard for production use.



- When you arrive at the **Pre-Requisites** window the Delinea Installer will check to make sure that all required conditions are met. If there are issues or failures, choose "**Fix Issues**" to automatically install the necessary Pre-Requisites. This may take several minutes. If an "Installing Dependencies" windows pops up during this process, close to continue. When successful, click **Next**.

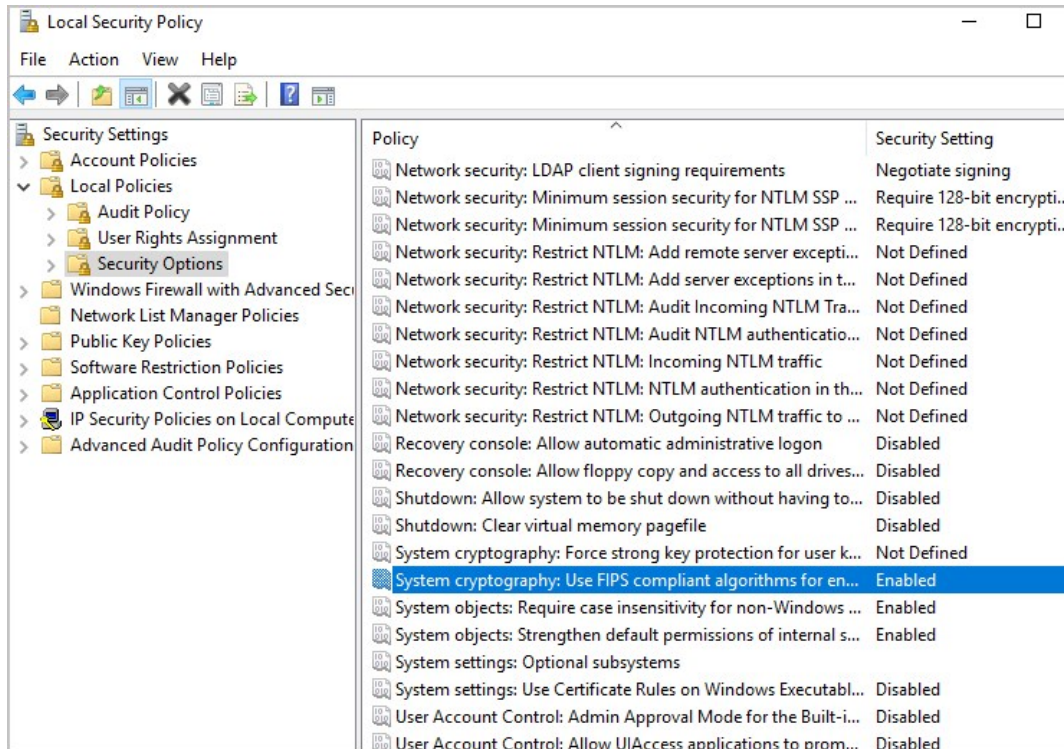
Troubleshooting:



Troubleshooting Note: If the **FIPS Encryption Enabled**, **Suitable FIPS Configuration**, or **HTTPS Binding** settings *Fail* during this Pre-Requisites check, click the **Fix Issues** button. After the tests run again, if the settings still list as "Failed," close the installer as directed, and then re-open the Installer. **Run the installer again** and the settings should pass.

If the FIPS Encryption Enabled setting continues to fail the pre-requisite check, you can manually turn on FIPS mode, Enabling FIPS Compliance in Windows by following the steps below:

1. Go to **Windows Security Policy editor** (secpol.msc) from your search bar
2. Navigate on your left pane to **Security Settings | Local Policies | Security Options**
3. Find and go to the property of **System Cryptography: Use FIPS Compliant algorithms for encryption, hashing, and signing...**
4. Choose **Enable** and click **OK**.
5. Restart your IIS server with **iisreset**.



- **FIPS Encryption Enabled** means that FIPS-compatible encryption algorithms (i.e. AES128, AES256) are available
- **Suitable FIPS Configuration** means that Windows is in FIPS mode on the local server
- For more information about enabling FIPS compliance, refer to the following:
 - [Enabling FIPS Compliance](#)
 - [FIPS 140 Validation](#).

NOTE: If installing SQL Express you may need to manually install .NET 3.5 through the Server Manager Roles and Features wizard. If this is the case, the installer will flag it for you in the details section. Remember that SQL Express is not compatible for Common Criteria compliance and is only intended for trial or sandbox environments.*

1. Provide the Server Name or IP address for your Database server. According to Common Criteria compliance, SQL should already be installed locally on this machine.

NOTE: Using SQL Server Authentication with Administrator SQL credentials in mixed mode is *NOT* compliant with Common Criteria. Instead, select **Windows Authentication using Service Account**. Then click **Next**.

SQL Authentication

SQL Server Authentication (SQL Server authentication requires Mixed Mode)

User name

Password

Windows Authentication using Service Account

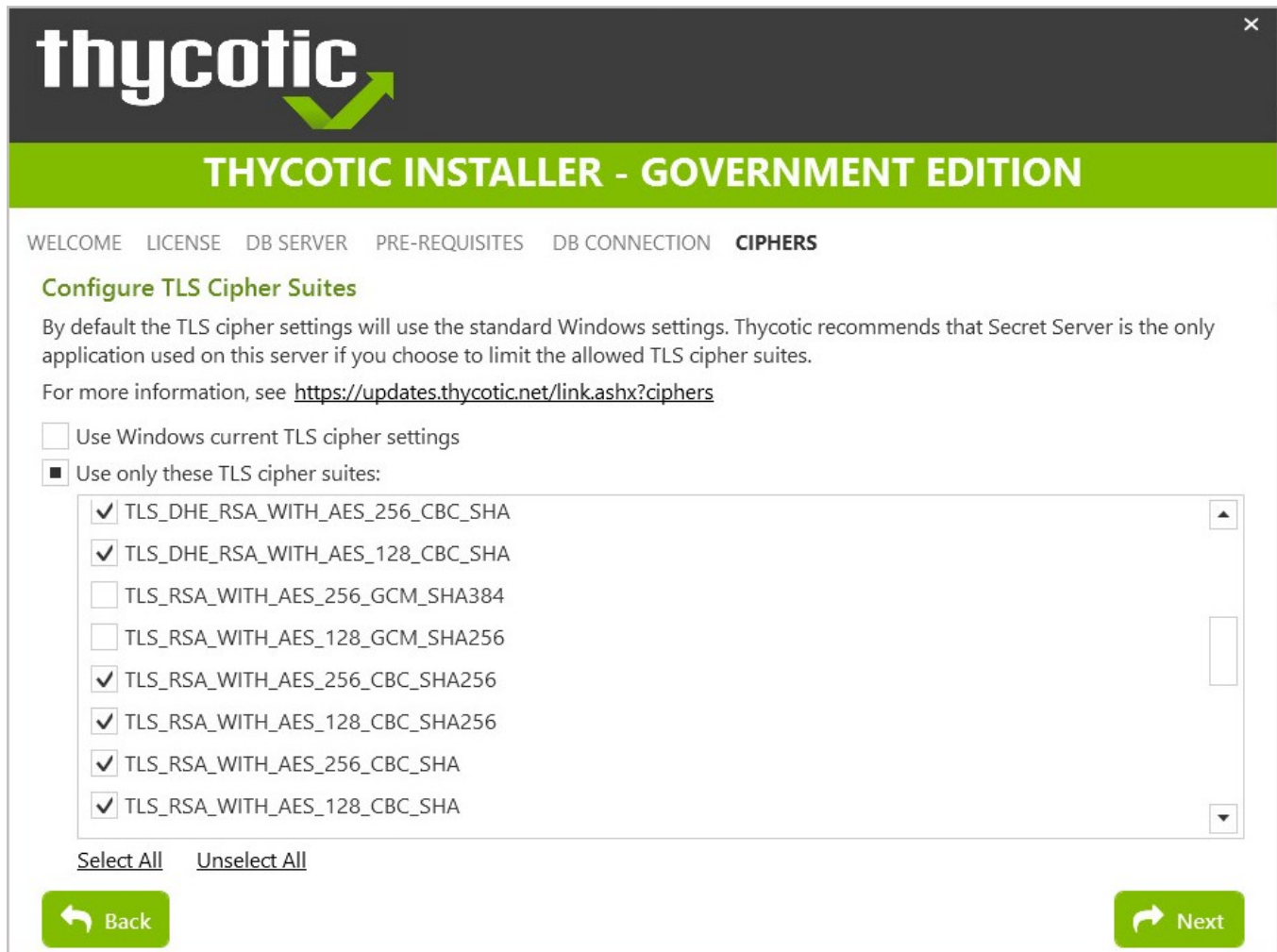
2. Choosing **Windows Authentication using Service Account** will bring you to a new tab called **Account**. This tab will prompt you for the **DOMAIN\User Name** and **Password** of your pre-configured service account which we will assign to the application pool

identities in IIS when the site is created. This account must have the **db_creator** role.

NOTE: To move between the Secret Server web platform and the SQL database, **your licensed SQL database must be maintain Compliance according to Common Criteria and ensure that encryption standards are upheld by communications installed locally (e.g. on the same server that the installer is running on).**

- If your database does not yet exist when you click "**Test Connection**" the Delinea Installer will create an instance of SQL Express, which is only intended for trial use and not suitable for production environments (see step 4).

3. Select **Validate Credentials**. When the connection has been tested successfully, click **Next**.



4. Verify that "Use only these TLS cipher suites" is selected, along with the six checked cipher suites. If not, deselect the toggle for "Use Windows current TLS cipher settings" and select **Use only these TLS cipher suites**.

The following TLS cipher suites are required to make sure your Secret Server instance's encryption standards are as secure as possible:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_RSA_WITH_AES_128_CBC_SHA256
 TLS_RSA_WITH_AES_256_CBC_SHA256

5. **Enter credentials** for your Secret Server Administrator account. Please remember these credentials as they are necessary to login to the web application after you complete installation. According to Common Criteria Compliance regulations, **this local administrator user password must be 15 characters or greater in length and include both numbers and special characters** – see section **9.1 Password Requirements for User Authentication** for more details.
6. If you have an **Email Server** that your organization uses for notifications, enter its information in the Email Server Window and then click **Next**. If this does not apply to your organization, select **Skip Email**
7. In the **Review** window (screenshot below), verify that all settings are correctly listed. If anything needs to be changed feel free to modify the section by clicking "Modify" on the righthand side. You may also hover over the left-hand question mark symbols for descriptions of each point. Once reviewed, click **Install**.

Note: Installation time will vary, most installs last between 10-40 minutes.

thycotic

THYCOTIC INSTALLER - GOVERNMENT EDITION

WELCOME LICENSE DB SERVER PRE-REQUISITES DB CONNECTION ACCOUNT CIPHERS CREATE USER EMAIL **REVIEW**

Review the installation options and then click Install to continue.

Name	Setting	
? SQL Server	Using Existing SQL Server	Modify
? Database Connection	SecretServer_db on [redacted]	Modify
? Encryption Configuration	[redacted]	Modify
? Site Name	[redacted]	Modify
? Secret Server Destination	[redacted]	Modify
? Secret Server App Name	SecretServer running as [redacted]	Modify
? Create User	[redacted]	Modify
? Email Server	[redacted]	Modify
? TLS Ciphers	[redacted]	Modify

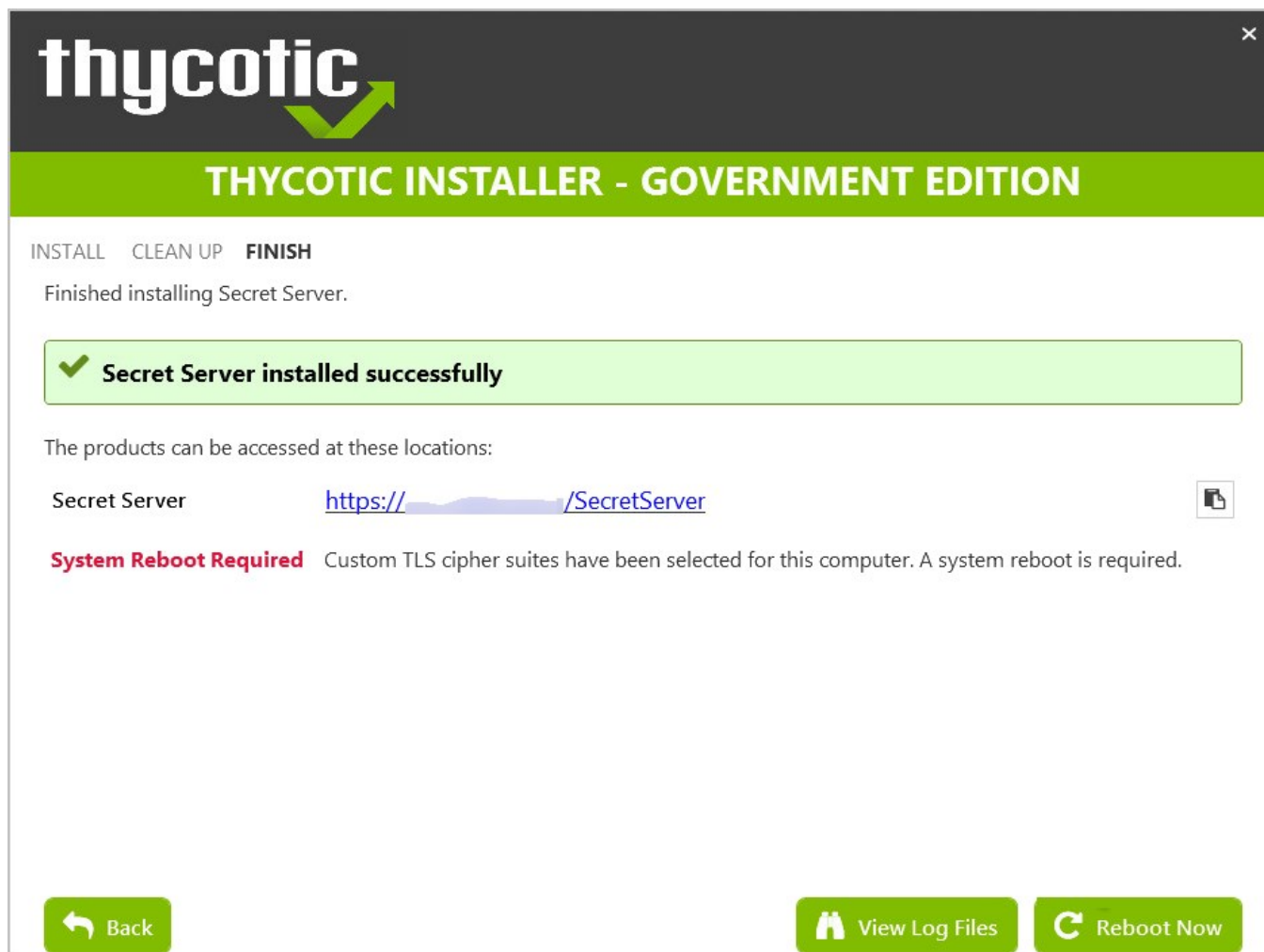
[Back](#) [Install](#)

[Need help?](#)

- When the install has successfully completed, the **Clean Up** tab will direct you to move your backup file path location. Follow these instructions and move your backup encryption file to a secure location on a different machine for disaster recovery purposes. Then click **Next**.



9. Because you specified TLS ciphers in step 6, after successfully completing the Secret Server Government Edition install you will be directed to fully reboot your Windows server so that the ciphers will take effect. Click **Reboot Now** to finish (screenshot below).



You can click on the provided URL to navigate directly to your setup landing page. Login using the Username and Password that you created on the "Create User" installer screen.

Additional Permissions for Secret Server Folder

1. After installing Secret Server, check permissions on the folder by navigating to the folder in which the web site was created (by default **C:\inetpub\wwwroot\SecretServer**)
2. Right-click the folder name and select **Properties | Security tab | Edit** button, then click **Add...** to add the **local user account created for the database** in step **2.2.3.1 Add the User Account to SQL Server**.
3. If not already checked, check the **Allow** boxes beside the **"Read and Execute," "List Folder Contents,"** and **"Read"** access permissions above, click **Apply**.

Accessing Your System

Browser Compatibility

Secret Server is compatible with the following browsers and versions. Some functionality, as noted below, is not available in all browsers.

Firefox	version 23+	Copy to Clipboard requires add-on. Click-once Launcher requires Microsoft add-on: https://www.microsoft.com/en-us/download/details.aspx?id=9923 (Protocol Handler supported)
Chrome	version 25+	Copy to Clipboard requires add-on. Click-once Launcher not supported (Protocol Handler supported)
Edge	all versions	Web filler not currently supported. Copy to Clipboard not currently supported.
Safari	version 5.0+	Copy to Clipboard requires add-on (Snow Leopard OSX or better; not supported on Windows). Click-once Launcher not supported (Protocol Handler supported)
Opera	not supported	

Table 4: Browser Compatibility

Note: Click-Once launcher is not supported on any non-Windows platform.

Web Interface

To access Secret Server, navigate to your organization's URL and login using your administrator credentials. The first time you login your screen will look like this:

*For information on setting up your Secret Server Local Administrator account, see section 3.2.

After enforcing the Login Policy Banner (for instructions, see section 5.4.1), the login screen will appear this way:

Login

Username *

Password *

Access to this system is restricted to authorized users. Users (authorized or unauthorized) have no explicit or implicit expectation of privacy. Any or all secrets may be modified, copied, audited, inspected, and disclosed to company management, law enforcement personnel, and other authorized individuals. I understand that I am responsible

I accept these terms

[Login](#)

Once user accounts are configured, both Local accounts and Active Directory users will access Secret Server through this login screen. To configure Active Directory users, see section 7.1.

You will be directed to Secret Server's landing page for Administrator accounts.

Thycotic Secret Server 10.0
Government Edition

Search Secrets [Search](#) [HOME](#) [TOOLS](#) [ADMIN](#) [REPORTS](#)

Browse [+](#)

Find Folder [+](#) [+](#) [+](#) [+](#)

Search in results [Advanced](#)

Secret	Folder	Template
Page 1 of 0		
15 records to view		
< Select Bulk Operation >		

Create Secret [+](#) [Content](#)

Create New
< Select >

Recent Secrets [+](#) [Content](#)

Favorite Secrets [+](#) [Content](#)

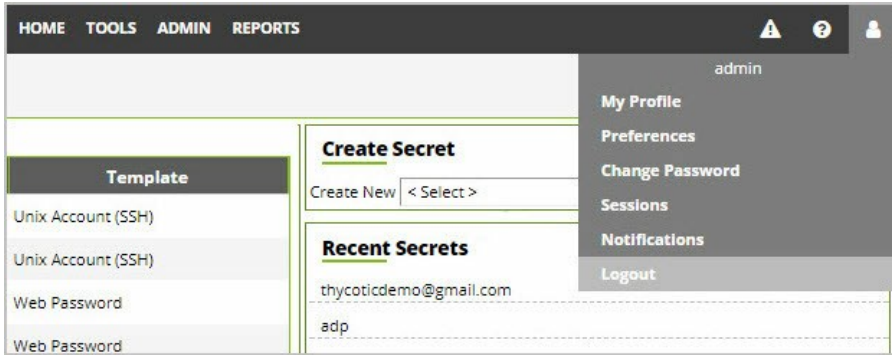
Expired Secrets [+](#) [Content](#)

Welcome [+](#) [Content](#)

Getting Started with Dashboard

Dashboard has a rich array of widgets with a customizable design. Create Tabs, add widgets, and arrange your own layout using drag and drop. See all the new features in Dashboard in this movie.

[View Movie](#) [Dismiss](#)



To logout, navigate to the **profile icon** in the upper right-hand corner and select **Logout** from the dropdown list. Once user accounts are created in Secret Server from the Admin profile, users can login using user-level credentials by navigating to your organization's login URL.

More detailed information for setting up Secret Server including customizing Secret Server your dashboard and creating users can be found in Secret Server's User Guide:

https://updates.thycotic.net/secretserver/documents/SS_UserGuide.pdf **CHANGE TO INTERNAL LINK(S)**

Local Admin Account

To create Secret Server's Local Administrator Account you will be prompted for information through the installer (step 7 in section 2.3.2 of this guide). Save the information you provide during this step so that you can login to the administrator account when the installer finishes setting up your instance.

The screenshot shows the 'THYCOTIC INSTALLER - GOVERNMENT EDITION' window. The title bar includes the Thycotic logo and a close button. Below the title bar, there are navigation tabs: WELCOME, LICENSE, DB SERVER, PRE-REQUISITES, and CREATE USER. The main content area is titled 'Please enter credentials for your initial Secret Server Administrator.' and contains the following fields:

- User Name:** Admin
- Display Name:** Lois Lane
- Email:** loislane@helpdesk.heroes
- Password:** [Redacted]
- Confirm Password:** [Redacted]

Below the form, there is a yellow warning box with the following text:

IMPORTANT: KEEP THIS INFORMATION
 This user name and password is used to administer Secret Server. It is the initial administrator and will be used to install licenses, create users, and setup advanced features. If you lose the password for this account, you will be unable to use these credentials in the event of an emergency.

I understand

At the bottom of the window, there are two buttons: 'Back' and 'Next'.

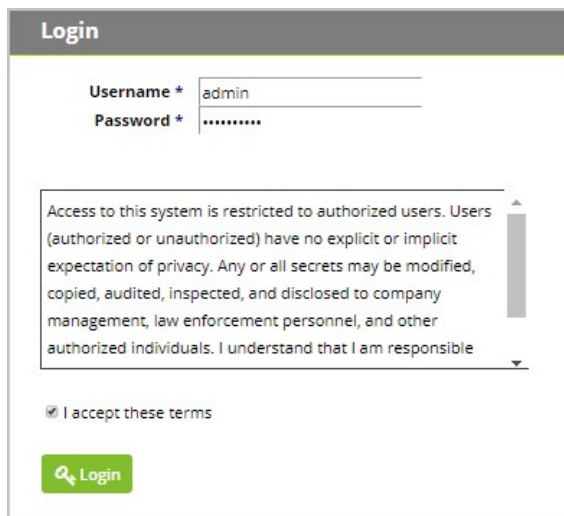
Installing License Keys

After logging into Secret Server with Secret Server's local administrator account, the first thing you will need to do is install your organization's license keys. You will need three different license keys, one for the Secret Server Edition being used, one for Support, and one for Users. Install your keys by navigating to **Admin | Licenses**, click **Install New License**. To install multiple licenses at once, click the **Bulk Entry Mode** button. Add your licenses according to the example format and click **Add Multiple Licenses** button, enter the required informational fields included in your trial or purchase license email and then click **Activate**.

To Activate newly installed licenses, click the **License Activation** images/a44f4a741a52b81b178cd16a325d8116.png)

*For information on setting up your Secret Server Local Administrator account, see section 3.2.

After enforcing the Login Policy Banner (for instructions, see section 5.4.1), the login screen will appear this way:




Login

Username * admin

Password *

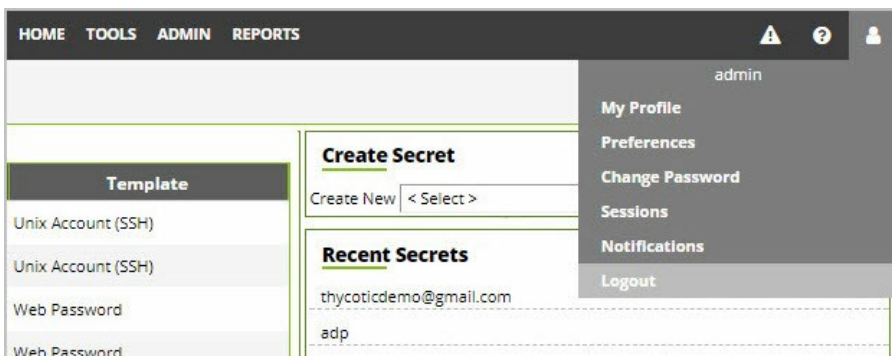
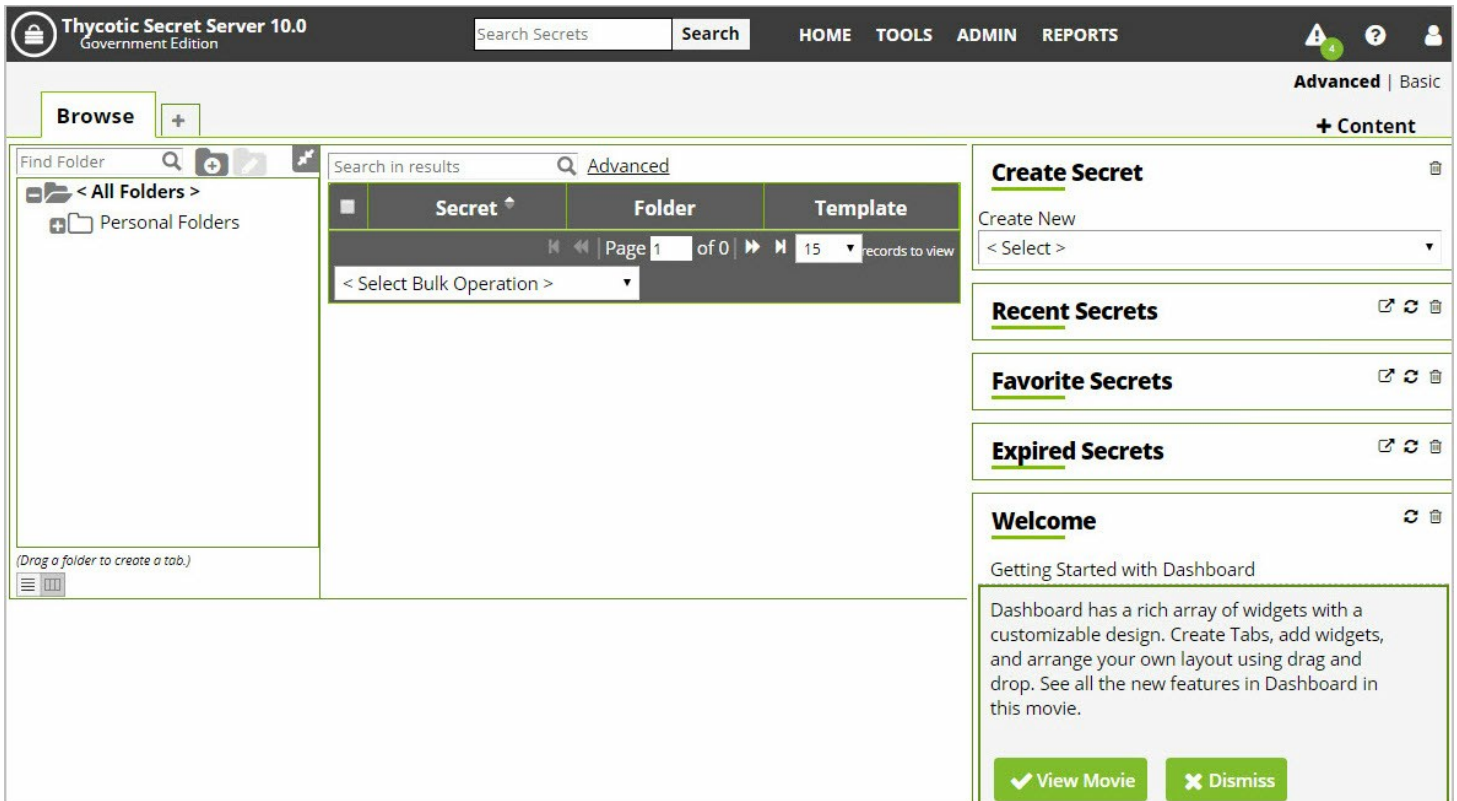
Access to this system is restricted to authorized users. Users (authorized or unauthorized) have no explicit or implicit expectation of privacy. Any or all secrets may be modified, copied, audited, inspected, and disclosed to company management, law enforcement personnel, and other authorized individuals. I understand that I am responsible

I accept these terms

 Login

Once user accounts are configured, both Local accounts and Active Directory users will access Secret Server through this login screen. To configure Active Directory users, see section 7.1.

You will be directed to Secret Server's landing page for Administrator accounts.




To logout, navigate to the **profile icon** in the upper right-hand corner and select **Logout** from the dropdown list. Once user accounts are created in Secret Server from the Admin profile, users can login using user-level credentials by navigating to your organization's login URL.

More detailed information for setting up Secret Server including customizing Secret Server your dashboard and creating users can be found in Secret Server's User Guide:

https://updates.thycotic.net/secretserver/documents/SS_UserGuide.pdf **CHANGE TO INTERNAL LINK(S)**

Local Admin Account

To create Secret Server's Local Administrator Account you will be prompted for information through the installer (step 7 in section 2.3.2 of this guide). Save the information you provide during this step so that you can login to the administrator account when the installer finishes setting up your instance.


×

THYCOTIC INSTALLER - GOVERNMENT EDITION

WELCOME LICENSE DB SERVER PRE-REQUISITES **CREATE USER**

Please enter credentials for your initial Secret Server Administrator.

User Name

Display Name

Email

Password

Confirm Password

⚠ IMPORTANT: KEEP THIS INFORMATION

This user name and password is used to administer Secret Server. It is the initial administrator and will be used to install licenses, create users, and setup advanced features. If you lose the password for this account, you will be unable to use these credentials in the event of an emergency.

I understand

Installing License Keys

After logging into Secret Server with Secret Server's local administrator account, the first thing you will need to do is install your organization's license keys. You will need three different license keys, one for the Secret Server Edition being used, one for Support, and one for Users. Install your keys by navigating to **Admin | Licenses**, click **Install New License**. To install multiple licenses at once, click the **Bulk Entry Mode** button. Add your licenses according to the example format and click **Add Multiple Licenses**. button, enter the required informational fields included in your trial or purchase license email and then click **Activate**.

To Activate newly installed licenses, click the **License Activation**

Updating Secret Server

Checking Software Version

Verify Software Version and Components

A shorthand representation of the software version appears in the upper left-hand corner of any user screen, along with the product edition. The full software version number is listed in the **Secret Server Environment** section by navigating to **Admin | Diagnostics**:

The screenshot shows the top navigation bar of the Thycotic Secret Server 10.0 Government Edition. The main content area displays the 'Secret Server Environment' section with the following details:

Secret Server Url	http://[redacted]/SecretServer/
Search Indexer Background Processor	
AD Synchronization Background Processor	Normal
LDAP Provider	Thycotic.AppCore.LdapProvider2, Thycotic.AppCore, Version=1.0.0.0, Culture=neutral, PublicKeyToken=[redacted]
Product Version	10.0.1[redacted]000
Internet Connectivity	Connection to [www.google.com] SUCCEEDED. Connection to [www.yahoo.com] SUCCEEDED. Connection to [updates.thycotic.net] SUCCEEDED.
Proxy Configuration	
Max Degrees of Parallelism	1
Maintenance Mode	Disabled

Downloading Software Updates

By default, Secret Server is configured to notify you immediately when software updates are available. To customize this behavior, navigate to **Admin | Configuration | General** tab and **Edit**, then check the box next to **Allow Automatic Checks for Updates**. This setting will create a banner at the top of your Administrator account's user screen when a new update is available for download.

The screenshot shows the 'Edit Configuration' page with the 'General' tab selected. Under the 'Application Settings' section, the checkbox for 'Allow Automatic Checks for Software Updates' is checked.

If your Secret Server web server has no outbound access, you can elect to receive updates through Delinea's Support Portal by logging into your Portal account at **Delinea.force.com**, **CHANGE TO WHAT?** navigating to your Account Settings, and signing up for the Delinea Mailing List.

After receiving a notification for the latest software release, navigate to <https://thycotic.force.com/support/s/article/Download-Secret-Server> in order to download the latest .zip file.

To perform an upgrade without outbound access, follow the steps in [Upgrading Secret Server Without Outbound Access](#).

Configuring Authentication and Login

The way that Secret Server authenticates interactive users depends on the type Secret Server of authentication that your organization has configured. Secret Server can use both **Local Accounts** and/or Active Directory **Domain Accounts** for authentication into Secret Server.

Local Authentication

A local user account is stored and managed by Secret Server. To successfully authenticate a local user must login with a matching username and password. When using local login, user credentials are checked against the internal authorized users' database.

To **create, edit, or remove** a local user account you must navigate to **Admin | Users** and locate any users whose Domain is listed as "Local" or **Create New** users.

Domain Authentication

A domain user account is stored and managed by Secret Server, but subject to changes made in Active Directory. To successfully authenticate a user must login with an Active Directory account that exists in Secret Server with matching Secret Server credentials. When using domain login, the TOE (Secret Server) initiates an authentication request to the external domain controller (Active Directory) using LDAP over TLS, and only allows access after receiving a successful result message.

For more information on syncing Secret Server with Active Directory credentials see section **7.1 Configuring Active Directory Sync**

Account Lockout Configuration

To access any information or functionality within Secret Server, users must login with correct local or domain credentials. To comply with Common Criteria regulations, Secret Server must be configured to prevent repeated unsuccessful attempts at logging in. **Account Lockouts** are used for this purpose. Configurable by the Secret Server Administrator, an account becomes inaccessible after a limited number of unsuccessful authentication attempts until an Administrator unlocks the user's account.

To configure settings for Account Lockouts, navigate to **Admin | Configuration | Login** tab, then click **Edit** and adjust the number for **Maximum Login Failures**. Default for this setting is five attempts.

To Unlock a user's account, navigate to **Admin | Users | Select the User** and click **Edit**. Change the value for "**Locked Out**" from Yes to **No**, then click **Save**.

Lockout Window

Secret Server is programmed with a default Lockout Window of 60 minutes. This means that once a user has locked out their account, they will be able to login with the correct credentials after a period of one hour has passed. Account lockouts are designed to prevent brute force attacks.

How to Disable "Allow Remember Me" during Logins

By default, the Secret Server installer will disable the "Allow Remember Me" caching feature during logins, however, to ensure this is feature is disabled, navigate to **Admin | Configuration | Login** tab, and verify that the first setting "**Allow Remember Me**" is set to **No**. If this is set to Yes, **Edit** the page and uncheck the toggle, then **Save**. Audits for this setting are logged under **Admin | Configuration | General** tab, by clicking the **View Audit** button. To filter log results, search for "**AllowRememberMe**" in the search bar.

Configuring the Login Banner

To configure the Login Banner, navigate to **Admin | Configuration | Login** tab, scroll to the bottom of the page and click the **Login Policy Agreement** button, then **Edit** and check the **Enable Login Policy** and **Force Login Policy** boxes.

Enabling these boxes will 1) Reveal the **User's Policy Agreement** on the Login page, and 2) Force users to **Agree** to the policy when logging into Secret Server.

How to Modify the Login Banner Messaging

To modify the display text from this **Login Policy Agreement**, go to Secret Server's Web Server and open **File Explorer**. Navigate to a text file called "**policy.txt**" (default file path location is **C:\inetpub\wwwroot\SecretServer\policy.txt**)

Open this file in Notepad and adjust the text according to your organization's policy requirements, then **Save**. The default text reads as follows:

Access to this system is restricted to authorized users. Users (authorized or unauthorized) have no explicit or implicit expectation of privacy. Any or all secrets may be modified, copied, audited, inspected, and disclosed to company management, law enforcement personnel, and other authorized individuals.

I understand that I am responsible for protecting the confidentiality of company secrets and will comply with the company Information Security Policy. Unauthorized or improper use of this system may result in administrative disciplinary action, civil and/or criminal penalties. By continuing to Login, I am indicating my awareness of and consent to these terms and conditions of use.

**** CLOSE THIS SITE IMMEDIATELY if you do not agree to these conditions ****

Configuring Session Timeouts

To configure Session Timeouts, navigate to **Admin | Configuration | General** tab, click **Edit**. Under the User Experience section, check the **Force Inactivity Timeout** check box, then adjust the number of minutes of inactivity before an active session in Secret Server will timeout and force users to login again. Click on the **Save** button to save your changes.

Configuring IP Address Restrictions

To configure IP address restrictions, navigate to **Admin | More... | IP Addresses** and click **Create New**. Then provide the **IP Address User/Network Name** and the **IP Address Range**, click **Save**.

IP address restrictions can then be set at a user-level by navigating to **Admin | Users** and clicking on a user name, then click **Change IP Restrictions** and click on the Restriction checkbox for the restriction(s) you would like to enable for that user. **Save** changes.

To restrict group access into Secret Server by a specific IP address or IP address range, simply configure your IP address or range as listed above, then navigate to **Admin | Groups | Create New**. Select all employees or groups of employees to impose this restriction on and move left into the **Members** box. Click **Save**. Next click the button **Change IP Restrictions** and check the box for your desired IP Address/Range. **Save** changes to apply. Now users will now be restricted from accessing Secret Server outside of the designated IP Address Range.

Managing Local Users

Create New Local Users

To create a new local user in Secret Server, navigate to **Admin | Users** and select the **Create New** button. Provide the requested information on the **Edit User** page.

Manage Local Users

Click on the **Username** for details about a specific local user. Secret Server will provide information including **User** and **Display Names**, **email address**, **Domain** affiliated, whether the user has **Two Factor** verification set up on their account, whether the user is **Enabled**, **Locked Out** from accessing their account, or whether the account is not a User account but an **Application Account**. The **View User** page also provides information about **Groups** and **Roles** for this user.

To edit these settings, click the **Edit** button. By editing a local user you may change their **display name**, **email address**, allow **Two Factor**, **disable/enable** the account, or **lock/unlock** the account.

You also can **create a new, one-time password** for the user's account. If you change a user's password they will be prompted for a new password after logging in with the Administrator-created password.

Managing Domain Users

Configuring Active Directory Sync

Secret Server can integrate with Active Directory by allowing users to use their Active Directory credentials to login to Secret Server. According to Common Criteria compliance, Active Directory relies on **LDAPv3 (RFC 2251) protocol**, which is not configurable by users.

In order to setup Active Directory in Secret Server, you will need to:

1. **Create a Sync Secret**
2. **Specify the domain to authenticate against**
3. **Configure TLS with Active Directory**
4. **Set Synchronization Groups**
5. **Turn on Active Directory Sync**

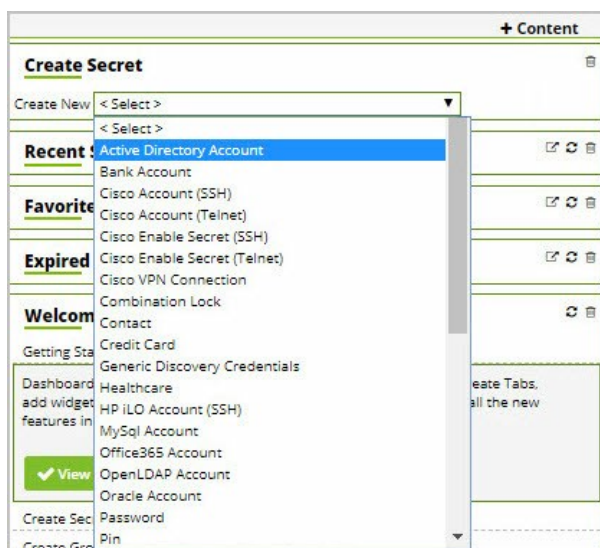
Secret Server relies on a primary “Sync” secret to connect to the LDAPv3 server in Active Directory. Once connection has been made to Active Directory through this secret, Secret Server needs to know which domain within Active Directory to authenticate against, and within that domain which specific Active Directory Group(s) to synchronize with. The Active Directory Sync in Secret Server targets Secret Server user account credentials from Active Directory. Secret Server will categorize users according to group information from Active Directory, but **Secret Server does not create, delete, or alter Active Directory Group Policies**.

Create a Sync Secret

Before synchronizing users, you must first create a secret to be used as the Sync Secret. This secret should contain **Domain Admin credentials** (or an account with appropriate permissions for **Read Access** to all your organization’s AD objects).

From Secret Server’s dashboard you can create this secret through the Create Secret Widget.

1. Next to “Create New,” select **Active Directory Account** from the dropdown list.



2. Add a **Secret Name** and provide the **Domain Name, Username,** and **Password** for the Sync Secret that will be able to access Active Directory with Admin credentials. **Save.**

New

This folder is for work related Secrets only. Do not store personal non-work Secrets, such as your Online Banking password, in this folder.

General

Secret Template: Active Directory Account

Secret Name: * Sync Secret

Domain: * testlab.com

Username: * TESTLAB\Administrator

Password: * [masked] **Generate** Strong ✓

Notes: [empty text area]

Folder: \Personal Folders\admin **Clear**

Inherit Secret Policy:

Secret Policy: < No Policy >

Save **Save and Share** **Save and Add New** **Cancel**

Specify the Domain & Enable Active Directory Integration

Specify which domains Secret Server will be able to authenticate against. Secret Server can synchronize with any number of domains.

1. Once logged into Secret Server – Click on **Admin | Active Directory**

Search Secrets **Search** **HOME** **TOOLS** **REPORTS** **ADMIN** **HELP** **LOGOUT**

Active Directory | Licenses

Backup | Remote Password Changing

Configuration | Roles

Diagnostics | Scripts

Discovery | Secret Policy

Distributed Engine | Secret Templates

DoubleLock | Session Monitoring

Event Subscriptions | SSH Proxy

Export | System Log

Folders | Users

Groups | More...

Secret	Folder	Template	Heartbeat
10.12.10.6\localadmi...	\Infrastructure\Serve...	Unix Account (SSH)	Success
10.12.10.6\sysman (...	\Infrastructure\Serve...	Unix Account (SSH)	Success
amazon.com (Web Fi...	\Personal Folders\Le...	Web Password	
GORO.testlab.com\A...	\Infrastructure\Serve...	Windows Account	Heartbeat F
GORO.testlab.com\S...	\Infrastructure\Data...	SQL Server Account	Success
goro\DBAdmin (SQL ...	\Infrastructure\Data...	SQL Server Account	Success
REPTILE\Administrat...	\Infrastructure\Work...	Windows Account	Success

2. From the AD Configuration page, click **Edit**.
3. Check **Enable Active Directory Integration** in the Active Directory Integration section
4. Select **Enable Synchronization of Active Directory** in the Active Directory User Synchronization section
5. Wait a few seconds for the screen to update
6. Next to **User Account Options** select **User status mirrors Active Directory (Automatic)** from the drop-down. This allows

Secret Server to mirror any changes made to Active Directory automatically.

7. Click **Save**

Edit Active Directory Configuration

Active Directory Integration

Enable Active Directory Integration

Enable Integrated Windows Authentication Requires advanced IIS settings (See [KB Article](#))

Active Directory User Synchronization

Enable Synchronization of Active Directory Enable to synchronize users by Active Directory Group. Once saved set the group with Edit Synchronization button.

Synchronization Interval for Active Directory Days: 0
Hours: 1
Minutes: 0

User Account Options:
 Users are enabled by default (Manual)
 Users are enabled by default (Manual)
 Users are disabled by default (Manual)
 User status mirrors Active Directory (Automatic)

Save Cancel

8. Click **Edit Domains**

Thycotic Secret Server 10.0 Government Edition

Search Secrets Search HOME TOOLS ADMIN REPORTS

Active Directory Configuration

⚠ Active Directory Integration is enabled but no Domains have been added. Please click Edit Domains to Add the domain

Active Directory Integration

Enable Active Directory Integration Yes

Enable Integrated Windows Authentication No

Active Directory User Synchronization

Enable Synchronization of Active Directory Yes

Synchronization Interval for Active Directory 1 hour

User Account Options User status mirrors Active Directory (Automatic)

Back Edit Edit Domains Edit Synchronization

9. Click **Create New**

10. Provide the **Fully Qualified Domain Name**

11. Provide a **Friendly Name**

12. Ensure that the box next to **Active** is checked.

13. Check the box next to **Allow Logins From Domain**.

14. Select your Sync Secret by clicking **No Selected Secret**
15. Search for the secret created earlier.
16. Click **Save and Validate**.

The screenshot shows the 'Active Directory Domain' configuration page in the Thycotic Secret Server 10.0 Government Edition. The page has a dark header with the product name and a search bar. Below the header, the main content area is titled 'Active Directory Domain'. It contains several configuration options: 'Fully Qualified Domain Name' and 'Friendly Name' are text input fields; 'Active' and 'Allow Logins From Domain' are checked checkboxes; 'Sync Secret' is a dropdown menu currently showing 'No Selected Secret' with a 'Create New Secret' link to its right; 'Site' is a dropdown menu showing 'Distributed Engines are Disabled in the Configuration'; and 'Enable Discovery' is a dropdown menu showing 'Not Enabled' with a note: 'Discovery is not enabled and will not be run. To enable it, go to the Discovery section under Administration.' Below these fields is a link for 'Advanced (not required)'. At the bottom of the form are two buttons: 'Save And Validate' and 'Cancel'.

The Active Directory Sync Secret will be used to synchronize users and groups, **it will require permission to search and view the attributes of the users and groups**. If you plan on using **Discovery (NOTE: Discovery is not under Common Criteria's scope)**, the account will also need permissions to scan computers on the network for accounts.

Configuring TLS with Active Directory

To ensure that TLS is configured with Active Directory:

- From the **Admin | Active Directory | Edit Domains | Create New** page (continuing from previous section), after entering the requested information, click **Advanced (not required)** and check the **Use LDAPS** box to enable. Click **Save And Validate** to save this domain.

This screenshot shows the 'Advanced (not required)' configuration page. It includes a link 'What is LDAPS?' and a checked checkbox for 'Use LDAPS'. Below it is a dropdown menu for 'Auto-Enable two factor for new Users' currently set to 'Google Authenticator'. At the bottom are 'Save And Validate' and 'Cancel' buttons.

**If the TLS connection to Active Directory fails, the user will be notified and the failure will be logged. Secret Server does not automatically retry to connect to TLS but will retry the next time a user attempts to connect to AD.*

More information for setting up Active Domain with LDAPS can be found at

<https://blogs.msdn.microsoft.com/microsoftserverteam/2017/04/10/step-by-step-guide-to-setup-ldaps-on-windows-server/>

"We're no longer updating this content regularly."

Setup Synchronization Groups

Once the domain has been added, click the "**Edit Synchronization**" button on the Active Directory Configuration page.

Synchronization Edit

testlab.com ▼

Synchronized Groups

SecretServer

Available Groups
(Search Results are limited to 100 groups. Use * for wildcards, ex: Admin*)

To view groups, click Search **Q Search**

Access Control Assistance Operators
Access-Denied Assistance Users
Account Operators
Administrators
Allowed RODC Password Replication Group
Backup Operators
Cert Publishers
Certificate Service DCOM Access
Cloneable Domain Controllers
Cryptographic Operators

Save **Cancel**

The Available Groups represent all accessible groups on the specified Active Directory domain. The user membership can be previewed with the Group Preview control.

- Select the desired group from the Available Groups that contains the Active Directory accounts for users you would like to create in Secret Server and move it into the **Synchronized Groups** area, click **Save**. This allows you to tailor specific Active Directory users to have domain accounts in Secret Server.

Managing Domain Credentials

Updating Domain Credentials

Updating Active Directory (AD) Credentials and passwords happens directly through Active Directory and syncs with Secret Server according to a schedule. You can synchronize your Active Directory accounts at any time by navigating to **Admin | Active Directory** and clicking the **Synchronize Now** button. *You cannot add or edit Active Directory credentials through Secret Server directly.*

When a user logs into Secret Server using Active Directory credentials or uses an AD account to launch a session, the credentials are sent to the Domain Controller in real time for authentication verification. Therefore, if an AD user account has been updated or removed, changes will be reflected immediately in Secret Server.

Failed Domain Authentication

If the connection between Secret Server and the AD domain breaks, domain users Secret Server will fail to authenticate into Secret Server until the connection is re-established. Secret Server will log all failed authentication attempts by users.

To find these audit logs, navigate to **Admin | Users | select a user | View Audit** or refer to section **12.0 Local Auditing**.

Common Criteria Roles and Permissions

User Roles can be edited by navigating to **Admin | Roles** and choosing **Assign Roles**, then **Edit**. On the Role Assignment page you will be able to select which users should be assigned to which role.

The **User Roles** that comply with Common Criteria standards are:

- **Administrator**
- **User**
- **Read Only User**

By default, Administrators have all possible Role Permissions. To view the list of permissions and security attributes that associate to each User Role, navigate to **Admin | Roles** and select each role to view the list of permissions. Each user session will be limited by the Role Permissions assigned to that user.

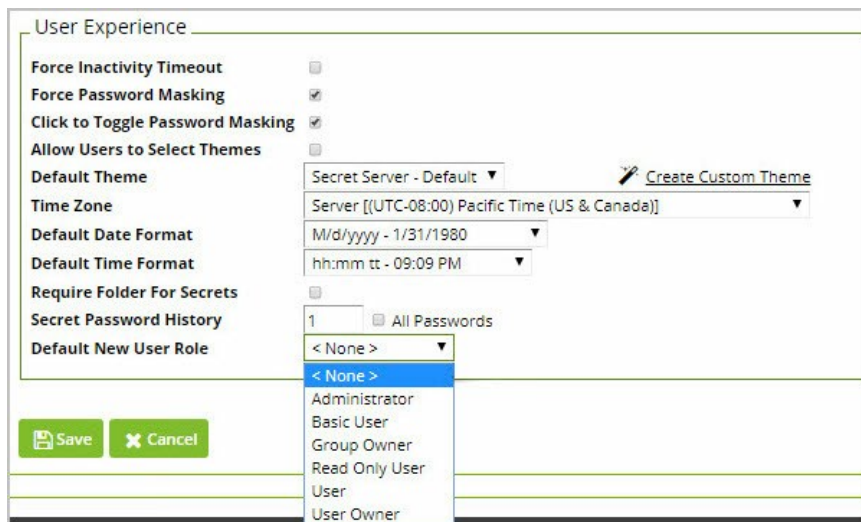
Each can be associated with individual user identities, or can be associated with either a local Group or an Active Directory Group. To assign a Role to a User or Group, navigate to **Admin | Roles | Assign Roles**.

To administer configurations required for Common Criteria standards, an Administrator requires the default permissions that are included for the Administrator Role.

Assigning Roles to Users

The default role assignment for new users is set by navigating to **Admin | General** tab | **User Experience** section (scroll to bottom). To ensure that no new users are created or imported with any extra privileges, make sure that this setting is set to "None."

To edit this default setting, click **Edit** and select "**None**" from the dropdown list as shown by the screenshot below. Then click **Save**.



The screenshot shows the 'User Experience' configuration page. The 'Default New User Role' dropdown menu is open, displaying the following options: '< None >', 'Administrator', 'Basic User', 'Group Owner', 'Read Only User', 'User', and 'User Owner'. The '< None >' option is currently selected and highlighted in blue. Below the dropdown menu, there are 'Save' and 'Cancel' buttons.

To assign users to Roles that are compliant with Common Criteria standards to specific users or groups, navigate to **Admin | Users** and select the user, OR navigate to **Admin | Roles**; then click **Assign Roles**.

View User

User Name	Diana Prince
Display Name	dprince
Email Address	dprince@helpdesk.heros
Domain	Local
Two Factor	< None >
Enabled	Yes
Locked Out	No
Application Account	No

IP Address Restrictions
None

Groups For User

There are no groups for this user.

Roles For User

Save To File < 1 to 1 of 1 >

Role Name
User

Back
Edit
View Audit
Change IP Restrictions
Assign Groups
Assign Roles

Under the **By User Or Group** tab, select a user or group from the dropdown, then click **Edit**.

The Roles supported in Common Criteria include **Administrator**, **User**, and **Read Only User**.

To apply one of these roles, select it from the righthand list and move it to the left side under the Assigned box. Ensure that every user only is assigned only one of these Roles. If both "Administrator" and "Read Only User" are assigned to the same user, the user will maintain full Administrator access to Secret Server. Click **Save Changes**:

Role Assignment

i Please note that changing role assignment could remove your access to Role Administration.

By Role

By User Or Group

User/Group dprince

<p>Assigned</p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <p>Read Only User</p> </div>	<div style="display: flex; flex-direction: column; justify-content: center; gap: 5px;"> << < > >> </div>	<p>Unassigned</p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <p>Administrator Basic User Group Owner User User Owner</p> </div>
--	--	--

Save Changes
Discard Changes

Management Functions Based On Role

This section describes management activities and corresponding roles of the evaluated security functionality.

Read-only User	Search and list Secrets
User	Use Secret/Launch session
User	Request access to Secret
Administrator	Create, view, expire, edit, and assign Secrets
Administrator	Perform bulk operations on Secrets
Administrator	Create and manage groups
Administrator	Create and manage roles, assign roles to users
Administrator	Create and manage containers (folders)
Administrator	Create and manage Secret policy
Administrator	Configure TOE SF (see Table 5)
Administrator	Create, manage, and unlock local accounts
Administrator	Configure IIS, SQL, syslog
Administrator	Update TOE

Table 5: Management Functions and Roles

By default each User Role is attached to Permission Sets. To view the specific permissions that each role is attached to, navigate to **Admin I Roles** and click into the user roles listed to see the list of permissions.

Organizations can tailor these user roles to maintain whatever permissions settings are required for your specific user environment.

8.3 Common Criteria Management Activities Based On Role

The following table specifies the user role required to allow each management activity listed according to Common Criteria Standards.

ESM_EAU.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_EID.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_ICD.1	Definition of identity and credential data that can be associated with users (activate, suspend, revoke credential, etc.)	Administrator
	Management of credential status	Administrator
	Enrollment of users into repository	Administrator

ESM_ICT.1	Configuration of circumstances in which transmission of identity and credential data (and object attributes, if applicable) is performed	Administrator
FAU_STG_EXT.1	Configuration of external audit storage location	Administrator
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	Administrator
	Management of actions to be taken in the event of an authentication failure	Administrator
FIA_USB.1	Definition of default subject security attributes, modification of subject security attributes	Administrator
FMT_MOF.1	Management of sets of users that can interact with security functions	Administrator
FMT_SMR.1	Management of the users that belong to a particular role	Administrator
FTA_SSL.3	Configuration of the inactivity period for session termination	Administrator
FTA_TAB.1	Maintenance of the banner	Administrator
FTP_ITC.1	Configuration of actions that require trusted channel (if applicable)	Administrator
FTP_TRP.1	Configuration of actions that require trusted path (if applicable)	Administrator

Table 6: Management Functions and Roles by Common Criteria Requirement

Managing User Passwords

Password Requirements for User Authentication

The Secret Server password policy for your user accounts is determined either through Active Directory or through Local User Passwords settings. To configure settings for Local User Passwords, navigate to **Admin | Configuration | Local User Password** tab and click **Edit**.

To meet Common Criteria Compliance regulations, passwords must be a **minimum of 16 characters** and include any subset of the following requirements (as demonstrated below):

- **Upper case letters**
- **Lower case letters**
- **Numbers**
- **These Special Characters: ! @ # \$ % ^ & * ()**

Edit Configuration

General | Login | Folders | **Local User Passwords** | Security | Ticket System

Allow Users to Reset Forgotten Passwords

Symbols Required for Passwords

Lowercase Letters Required for Passwords

Uppercase Letters Required for Passwords

Numbers Required for Passwords

Minimum Password Length Minimum Password Length !

Enable Local User Password Expiration

Days

Hours

Minutes

Enable Minimum Local User Password Age

Enable Local User Password History

Passwords in Secret Server are randomly generated according to admin-defined password complexity policies. By default the generator requires at least one upper case letter, one lower case letter, one number, and one symbol. The number of each type of character can be modified, and custom character sets can be created and used in password policies.

i **Example:** h)Wa7zNPw2j57%i7

Name

Description

Is Default Yes (All new Secret Templates will use this Password Requirement for password fields.)

Generate Password

Prevent Username In Password

Length between * and *

Using [Character Set](#).

Password Rules

Minimum of <input type="text" value="1"/> from <input type="text" value="Lower Case (a-z)"/>	✕
Minimum of <input type="text" value="1"/> from <input type="text" value="Symbol"/>	✕
Minimum of <input type="text" value="1"/> from <input type="text" value="Numeric (0-9)"/>	✕
Minimum of <input type="text" value="1"/> from <input type="text" value="Upper Case (A-Z)"/>	✕
Minimum of <input type="text" value="1"/> from <input type="text" value="Select..."/>	+

Save
Cancel
View Audit

Resetting User Authentication

To reset an Active Directory user account password, you must go *through Active Directory*.

To reset a Local User Password, follow these steps:

1. Navigate to **Admin | Users**.
2. Select the user who needs a reset.
3. Select **Edit**.
4. Type a new password twice. This is a *temporary password* that you must provide to the user. Immediately after the user logs into Secret Server they will be prompted to change their password.

If you are locked out of the Secret Server Local administrator account and you cannot request a reset through a linked administrator email account, contact the Delinea Support Team to request a password reset, and have your organization's **security pin code** at hand.

Setting Local User Password History Requirements

Ensure that local users cannot re-use old passwords to an administrator-settable number of past passwords used by that user.

1. Navigate to **Admin | Configuration | Local User Passwords**
2. Click **Edit**.
3. Check the toggle box next to **Enable Local User Password History**.
4. Set the number of historic passwords you would like to block for Secret Server Users.
5. Click the toggle next to **All** to block users from ever re-using any previous password.
6. Click **Save** to save changes.

Edit Configuration

General | Login | Folders | **Local User Passwords** | Security | Ticket System | Email | Session Recording | HSM

Allow Users to Reset Forgotten Passwords

Symbols Required for Passwords

Lowercase Letters Required for Passwords

Uppercase Letters Required for Passwords

Numbers Required for Passwords

Minimum Password Length

Enable Local User Password Expiration

Days

Hours

Minutes

Enable Minimum Local User Password Age

Enable Local User Password History

Number of Passwords: All

Managing Secrets

Creating Secrets

Each object that is stored within Secret Server is referred to as a **Secret**. Usually, a Secret will be a username and password combination. Other examples of Secrets can include SSH keys, contact information, or safe combinations.

To create a Secret, use the “Create Secret” widget.

The screenshot displays the Thycotic Secret Server 10.5 Enterprise Plus Edition interface. The top navigation bar includes 'HOME', 'TOOLS', 'ADMIN', and 'REPORTS'. The main content area is divided into several sections:

- Left Panel:** A 'Browse' section with a 'Find Folder' search bar and a tree view of folders including Personal Folders, Active Directory, AWS, Azure, Local Admins, Network, Service Accounts, and SQL.
- Center Panel:** A table titled 'Search in results' with columns for 'Secret', 'Folder', and 'Template'. It lists 15 secrets, including entries like 'internal.thycotic.training\b...', 'Admin Account 1', and 'sql\sa'.
- Right Panel:** A 'Create Secret' widget with a 'Create New' dropdown menu. Below it are three lists: 'Recent Secrets', 'Favorite Secrets', and 'Expired Secrets', each with a refresh icon.
- Bottom Panel:** A 'Welcome' widget with a 'Getting Started with Dashboard' link and a brief description of the dashboard's capabilities.

Select the type of Secret you want to create.

Thycotic Secret Server 10.5 Enterprise Plus Edition

Search Secrets Search HOME TOOLS ADMIN REPORTS

Advanced | Basic + Content

Browse +

Find Folder

- < All Folders >
 - Personal Folders
 - Active Directory
 - AWS
 - Azure
 - Local Admins
 - Network
 - Service Accounts
 - SQL

(Drag a folder to create a tab.)

Search in results Advanced

Secret	Folder	Template
<input type="checkbox"/> internal.thycotic.training\b...	Development	Active Directory Account
<input type="checkbox"/> Admin Account 1	Administrator	Active Directory Account
<input type="checkbox"/> Computer\Administrator	Local Admins	Windows Account
<input type="checkbox"/> internal.thycotic.training\Ad...	Active Directory	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\ad...	Active Directory	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\Ad...	Active Directory	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\Ce...	Domain Admins	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\Fu...	Domain Admins	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\js...	Standard Policy	Active Directory Account
<input type="checkbox"/> internal.thycotic.training\Ne...	Domain Admins	Active Directory Account
<input type="checkbox"/> Jumphost-10\jumphostadmin	Local Admins	Windows Account
<input type="checkbox"/> Secret	Australia	SSH Key
<input type="checkbox"/> sql\sa	SQL	SQL Server Account

Page 1 of 1 15 View 1 - 13 of 13

< Select Bulk Operation >

Create Secret

Create New

- < Select >
- < Select >
- Active Directory Account
- AS/400 IBM iSystem
- Bank Account
- Cisco Account (SSH)
- Cisco Account (Telnet)
- Cisco Enable Secret (SSH)
- Cisco Enable Secret (Telnet)
- Cisco VPN Connection
- Combination Lock
- Contact
- Credit Card
- Generic Discovery Credentials
- Healthcare
- HP iLO Account (SSH)
- Infoblox
- MySQL Account
- Office365 Account

Enter the data for the Secret you want to save, and make sure to choose the Folder that the Secret will be saved into. Hit Save, and your Secret has been created.

Thycotic Secret Server 10.5
Enterprise Plus Edition

Search Secrets

HOME TOOLS ADMIN REPORTS

New

General

Secret Template: Active Directory Account ▾

Secret Name: ! example name

Domain: ! example domain

Username: ! example Username

Password: ! Weak !

Notes: example notes

Folder: \Active Directory

Inherit Secret Policy:

Secret Policy: < No Policy >

Site: Local ▾

AutoChange?:

Notes
Any additional notes.

Configuring Secret Templates

Secret Server manages privileged account credentials through a highly configurable system of secrets. **Secret Templates** are used to create secrets and define object attributes for secrets. Templates can be configured according to account requirements.

The following is a list of Secret Templates that are compliant with Common Criteria standards available in the Government edition of Secret Server:

- **Active Directory Account**
- **Bank Account**
- **Combination Lock**
- **Contact**
- **Credit Card**
- **Password**
- **Pin**
- **Product License Key**
- **Security Alarm Code**
- **Social Security Number**
- **Unix Account (SSH Key Rotation)**

To follow Common Criteria standards, navigate to **Admin | Secret Templates** and click the **Active Templates** button. Ensure that only the templates listed above are selected in the Active column. Then **Save**. Users only can create Secrets using templates marked as Active.

To view object attribute data in a template from the dropdown list of Active Templates, **select a template** from the dropdown list and then click **Edit**.

Each of the listed templates below only hold object attribute data that are defined locally by Secret Server. The objects, secrets, and

templates are as follows:

Object

Attribute

Objects in Secret Server are defined by attribute data. All object attribute data is locally defined by Secret Server.

Secret

Attributes Inherited from Template

Secret templates dictate the fields each secret contains, the launchers for each secret, and the remote password changer used. They also provide a default expiration, which can be changed on a per-secret basis.

Command Restrictions

Command restrictions are limited to SSH sessions and allow administrators to create multiple-choice command menus that users can follow. If SSH command menus are enabled, users cannot issue commands directly to the target system.

Field Data

Each field in a secret template is either "Text," "Password," or "Notes." Password fields can be hidden from users and are updated when a password change has occurred on that secret's account. Text fields are the standard field type and may include information such as "Domain" or "Username."

Folder

Folders in Secret Server store individual secrets. Permissions applied to folders dictate which users can view the secrets inside that folder and which users can see the folders.

Password Requirements Rule Override

Password requirements can optionally be enforced on a per-template basis. If the password requirement is not enforced, users can manually type any password they want and will be notified if they enter a weak password that does not meet the password requirements. If the requirements are enforced, users cannot save the new password until it meets the requirements.

Policy Identifier

Secret policies are a collection of security and password settings that are applied to individual secrets or folders. Each setting of a secret policy can be configured as either default or enforced. Default allows users to later change the setting. Enforced locks the settings and cannot be modified on a per-secret basis unless the secret is moved out of the folder that has the secret policy attached. Secret policy settings include items such as "Remote Password Changing Auto Change," and "Requires Approval for Access."

Subject Identifier

The secret template of each secret identifies the type of store password or other data. This allows users to see the intended usage of each template.

Secret Name

The secret name is the label or title that describes the content of each secret. When Secret Server creates secrets automatically, the default naming convention is "host/account," or "domain/account."

Template

Field Parameters

Field parameters include username, password, and type. Secrets include a combination of field parameters that vary with user input. Secret type is defined by the template for each secret, such as "Bank Account" or "Active Directory Account." The number of field parameters also are defined at the secret template level.

Password Change Policy

Password rotation is enabled or disabled on a per-secret or per-template basis. Password rotation frequency depends entirely on the expiration period for each secret. The default expiration time is 30 days for all secret templates.

Password Strength Policy

A password policy is a set of instructions on how each new password is created. Administrators must choose a minimum and maximum password length, which character sets are used, and how many characters from each set are required. See section 9.0 for details on setting password requirements.

Secret Expiration Policy

When a secret reaches its expiration date, it is flagged as "Expired." If automatic password rotation is enabled for that secret, expiration triggers a remote password change. Expiration can be changed on each secret, but the default expiration period is set at the secret template level.

Secret Name Pattern

When Secret Server names a secret via the discovery import process, it uses the naming convention "hostname/username," or "domain/username." This is not enforced, so users can name secrets whatever they want.

Template Description

Secret template descriptions allow administrators to describe the purpose of a template when they create a new template, which is a best practice to avoid confusion.

Template Name

The secret template name is what users see in the drop-down menu when they create a new secret. It is also on the Secret Browse page so that users can see which secret template is associated with each secret.

Template Status

Like secrets, users can disable secret templates to make them invisible to users, unless they chose to view inactive templates. The status is either "enabled" or "disabled."

Secret Access Policy

If users have view (or greater) rights, they can see whether a secret exists and can open the secret to view its data. If a user does not have view rights, the secret is invisible.

Secret Modification Policy

Each secret carries individual access permissions that are typically inherited from the secret's folder. These permissions determine which users can view the secret and which users can edit the secret's data. A user can have view, edit, or owner permission. With the view permission, a user can view but not modify a secret. The edit permission allows the user to modify field data. The owner permission allows users to grant or revoke access to other users.

Configuring Password Policy for Secret Templates

When creating and rotating passwords for secrets inside of Secret Server, it is important to uphold strong requirements and to use Secret Server to manage changing requirements effectively.

For example, Secret Server allows administrators to set the minimum password length to 6 characters, observe that a 7-character password and a 16-character password are both accepted, then change the minimum length to 8, observe that a 7-character password is then rejected but that a 16-character password is accepted.

In Secret Server password requirements can be set and applied *at the Secret Template level*. To adjust requirements:

1. Navigate to **Admin | Secret Templates**.

The screenshot displays the 'Manage Secret Templates' page. At the top, there is a dropdown menu showing 'MySql Account' and a 'Show Inactive' checkbox. Below this, a row of buttons includes 'Back', 'Edit', 'Create New', 'Export', 'View Audit', 'Active Templates', 'Password Requirements', and 'Character Sets'. A second row of buttons includes 'Configure Launchers' and 'Configure Secret Template Permissions'. The page is organized into three main sections: 'Manage Secret Templates', 'Other Templates', and 'Import Secret Templates'. The 'Other Templates' section contains 'Configure Dependency Templates' and 'Configure Scan Templates'. The 'Import Secret Templates' section has a text box for pasting XML and a link to the 'Secret Templates Gallery'.

2. Select the **type of template** you want to adjust from the dropdown menu.
 3. Click **Password Requirements** to open the Password Requirements' page.
-

Password Requirements				
Name	Description	Minimum Length	Maximum Length	Default
Default	The default password requirement, which uses the alpha-numeric character set and requires one lowercase, one uppercase, one number, and one symbol.	12	12	Yes
SAP	SAP Password Requirement	12	12	No
Mainframe	Mainframe Password Requirement	8	8	No

[← Back](#)
[+ Create New](#)

- To edit *all* your organization's default password requirements on secret templates, select the **Default** password requirement from the list.
 - To create a new password requirement specific to this template, select **Create New**.
4. Adjust the **Password Length** and **Character Set** requirements to the needs of your organization. You may assign your new requirement to any Secret Template or templates.

i **Example:** SS(vLjg5yli4!xic)

Name

Description

Is Default *Yes (All new Secret Templates will use this Password Requirement for password fields.)*

Generate Password

Prevent Username In Password

Length between * and *

Using [Character Set.](#)

Password Rules

Minimum of <input type="text" value="1"/>	from	<input type="text" value="Lower Case (a-z)"/>	🗑️
Minimum of <input type="text" value="1"/>	from	<input type="text" value="Symbol"/>	🗑️
Minimum of <input type="text" value="1"/>	from	<input type="text" value="Numeric (0-9)"/>	🗑️
Minimum of <input type="text" value="1"/>	from	<input type="text" value="Upper Case (A-Z)"/>	🗑️
Minimum of <input type="text" value="1"/>	from	<input type="text" value="Select..."/>	+

📁 Save

✖ Cancel

[Show Usages](#)

Note: Password Requirements set for these templates will be enforced for both human-generated and auto-generated secrets.

Authentication Strength for Non-Password Credentials

Secret Server uses RSA keys of 2048 bits or higher for secure authentication. These SSH keys are non-password credentials that can be managed by Secret Server. To ensure that these are maintained up to encryption standards, make sure that your **Unix Account (SSH Key Rotation)** Template is configured with an SSH Key Bit Size of 2048 or higher.

By default, the Government Edition of Secret Server will set this Template setting to 2048.

To adjust the bit size:

1. Navigate to **Admin | Secret Templates**.
2. Select **Unix Account (SSH Key Rotation)** from the dropdown list.
3. Click the **Edit** button.

- From the Secret Template Designer page, click **Edit** to adjust settings. You can increase the bit size to 4096 from the default setting of 2048 if you choose, but do *not* lower this setting to 1024.

To create, store, and manage SSH keys in Secret Server, users must engage this Unix Account (SSH Key Rotation) Template. That means an SSH key will be created only when a standard encryption is enforced.

Configuring Remote Password Changing for SSH Key Rotation

Security Overview for SSH Key Rotation and PuTTY Launcher

SSH Key Rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. The public/private key pair is regenerated and the private key is encrypted with a new passphrase any time a secret's password changes, either manually or automatically. The public key is then updated on the Unix machine referenced on the secret.

To use default SSH Key Rotation commands, the machine being managed must meet the following minimum requirements:

- SSH Key logins in OpenSSH format should be enabled on the target using keys. A secret can be created with keys in PuTTY format but they will be converted to OpenSSH when the key is rotated.
- Public keys should be stored in [~userhome]/.ssh/authorized_keys (not authorized_keys2).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH Key Rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password, as well as permissions to edit the user's authorized keys file with sudo.

If a system does not meet these requirements it may still be possible to do key rotation by modifying the key rotation command sets.

Creating a Unix Account (SSH Key Rotation) Secret

Under Secret Server's Common Criteria compliance standards, you can set up Secret Server to rotate SSH Keys for Unix Accounts.

To setup a Launcher, you will need a **Unix Account (SSH Key Rotation)** Secret that is connected to a remote machine

To create a Unix Account (SSH Key Rotation) Secret:

1. From the **Home** Dashboard select the **Unix Account (SSH Key Rotation)** Template from the **Create Secret** widget.
2. Enter a **Secret Name**, **Machine Name**, **Username**, and **Password** for the Unix/Linux Account.
3. Select the SSH **Private Key** by browsing to it on your machine.
4. Enter the **Private Key Passphrase** for the SSH Key. If there is a corresponding Public Key, upload that as well.
5. Specify which Secret Server folder you want to store the Key in if it differs from the default folder path. Likewise, determine what secret policies you want to assign to this SSH key, if any.

General

Secret Template: Unix Account (SSH Key Rotation)

Secret Name: * Test SSH Rotate

Machine: * .180

Username: * ec2-user

Password: * [masked] **Generate** Strong ✓

Private Key: Choose File ProductMana...Pair-01.pem Generate New SSH Key

Private Key Passphrase: [locked] Choose File No file chosen **Generate**

Public Key: Choose File

Notes: Here is an SSH Key for a Unix Account--Rotate it regularly!

Folder: \Personal Folders\ssadmin [Clear](#)

Inherit Secret Policy:

Secret Policy: < No Policy >

AutoChange?:

Save **Save and Share** **+ Save and Add New** **✕ Cancel**

To automatically rotate the private and public SSH key pairing upon clicking **Save**, click **Generate New SSH Key**. This action is a security measure to ensure that no one can access your SSH key unless they are doing so through Secret Server's vault.

Enabling the Launcher

By default, the Launcher is enabled.

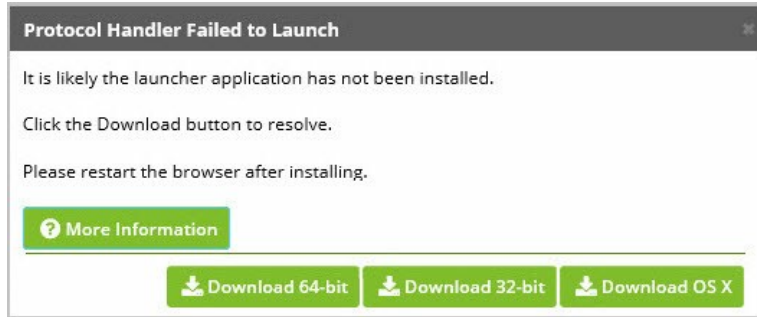
To verify this, click on **Admin | Configuration**. Check the **Enable Launcher** Setting to ensure that this is set to **Yes**.

Using the Launcher

Note: SSL Certificates must be properly configured or the Launcher will fail to connect. So when you try to connect to Secret Server you receive a certificate error such as host name mismatch, you must resolve the cause of the error before the putty launcher will function. To directions on setting up certificates, see section **CREATE AN INTERNAL LINK** 11.2 Configuring X.509v3 Certificates.

1. From your Home screen, click the SSH secret
2. Click **View Secret**.
3. Click the **PuTTY Launcher** icon.

Note: The first time you perform this task you will receive a "**Protocol Handler Failed to Launch**" message. Select the type of launcher you need and **Run** the .msi file. Secret Server will download a very small process called a Protocol Handler that facilitates the connection between your machine and the endpoint. Once the Protocol Handler is downloaded, close out the "Failed to Launch" window and *refresh your browser page*.



4. Click the launcher icon on the secret, and provide the machine name if prompted. The credentials will be passed along to the launcher and will open the appropriate launcher PuTTY session.

Enable Remote Password Changing

Note: To maintain compliance with Common Criteria regulations, Remote Password Changing is applicable for SSH Key Rotation ONLY.

1. Navigate to **Admin | Remote Password Changing**.
2. Click **Edit**
3. Check to select **Enable Remote Password Changing**.
4. Click **Save**.



Rotate SSH Key Remotely

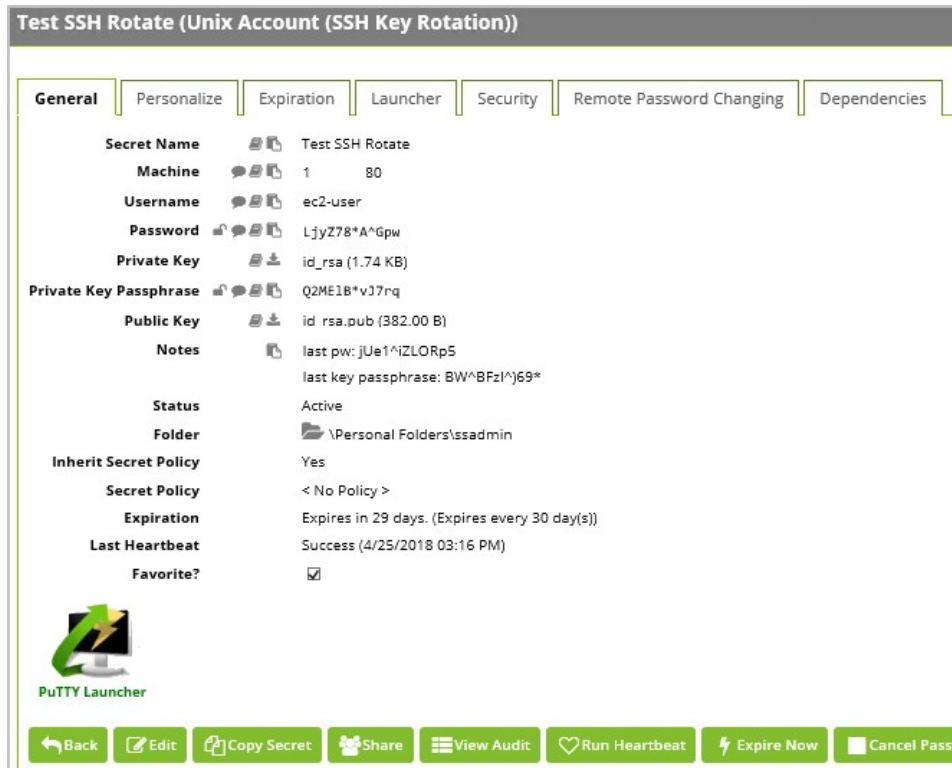
1. Navigate back to the SSH Key Rotation Secret's **View** screen.
2. At the bottom of the screen, click the **Change Password Remotely** button.



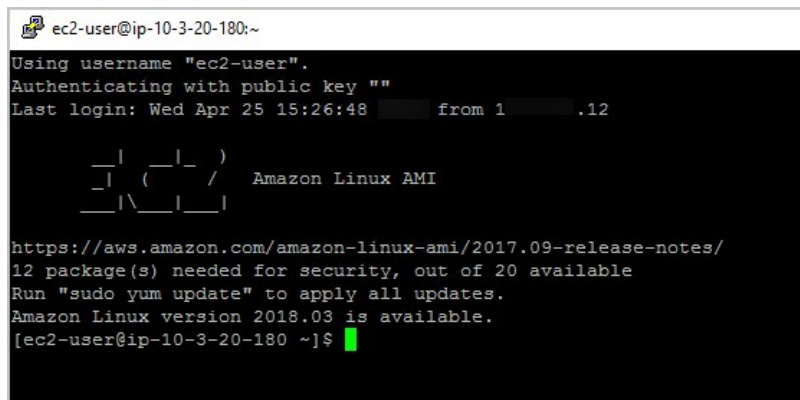
3. From the Change Password Remotely screen, **Generate** a new Password and Passphrase for your new SSH Key.
4. Next to Generate New SSH Key, leave the toggle checked.
5. Click **Change**. You will be directed to a **Password Scheduled for Change** screen.
6. Click **Back** to return to your secret's Remote Password Changing tab.



7. Navigate to the **General** tab.
8. Verify that a new password is listed. In the screenshot below you can see the previous passwords listed in the Notes section, confirming that the rotation was effective.



9. Click the **PuTTY Launcher** icon to confirm that Secret Server can still connect to the unix/linux machine using the newly rotated SSH Key.



Secret Expiration

A core feature of Secret Server is Secret expiration. Any template can be set to expire within a fixed time interval. For a Secret to expire, a field must be selected as the target of the expiration. For example, a Secret template for Active Directory accounts might require a change on the password field every 90 days. If the password remains unchanged past the length of time specified, that Secret is considered expired and will appear in the **Expired Secrets** panel on either the Dashboard's Expired Secrets widget or the Home page.

Secret expiration provides additional security by reminding users when sensitive data requires review. This can assist in meeting compliance requirements that mandate certain passwords be changed on a regular basis. When expiration is combined with Remote Password Changing, Secret Server can completely automate the process of regularly changing entire sets of passwords to meet security needs.

Setting up Secret Expiration for the Secret template

To set up expiration on a Secret, you must first enable expiration on the template from which the Secret is created.

To enable Secret expiration for a Secret template, navigate to **Administration | Secret Templates**. In the Manage Secret templates page, select the template from the dropdown list and click the **Edit** button. In the Secret template Designer page, click on the Change link. On this subsequent page, check the **Expiration Enabled?** box. You can now enter the expiration interval (every x number of days) as well as the field on the Secret you wish to expire and require to be changed. The interval setting can be overridden for each individual Secret.

Enabling expiration for a template will enable expiration for all the Secrets that were created using this template.

Setting up Secret Expiration for the Secret

Now that expiration has been enabled for the template, Secret expiration is enabled for the Secrets that were created using that template as well as Secrets created in the future. The Expiration tab will appear on the Secret View page and requires the user to have Owner permission on the Secret. If you would prefer to set a custom expiration at the Secret level, you can adjust the interval of expiration for the Secret by clicking the **Expiration** tab in the Secret View page. In the Expiration tab, you can set the Secret to expire using the template settings (default), a custom interval, or a specific date in the future.

Forcing Expiration

To force expiration, navigate to the **Secret View** page. From there, click **Expire Now**. This will force the Secret to expire immediately regardless of the interval setting. The expiration date will read "Expiration Forced".

Resetting an Expired Secret

To reset an expired Secret, you will need to change the field that has expired and is required to change. For example, if the field set to expire is the Password field and the current Password is "asdf", then a change to "jklh" will reset the expiration interval and thus remove the expiration text on the Secret View page.

If you do not know which field is set to expire, you will need to go to the Secret template that the Secret was created from. Navigate to **Administration | Secret Template** and select the template. Click the **Edit** button and then on the next page, click the **"Change"** link. In the "Change Required On" textbox you will see the field that is set to expire.

AutoChanging an Expired Secret

Remote Password Changing (RPC) is enabled under the Administration, Remote Password Changing page. Click **Edit** to enable Remote Password Changing, Secret Heartbeat, and Secret Checkout. Once enabled, all Secret templates with RPC configured will be available to use RPC.

The Remote Password Changing tab contains the settings for configuring RPC on an individual Secret. Enabling AutoChange on a Secret will allow Secret Server to Remotely Change the Password when it expires. The user must have Owner permission on the Secret to enable AutoChange. When editing on the RPC tab, the Next Password field can be set. If left blank an auto-generated password will be used.

To auto-change passwords based on secret expiration leave the AutoChange schedule set to **"None."**

If the password change fails, Secret Server will flag the Secret as Out of Sync and continue to retry until it is successful. If the Secret cannot be corrected or brought In Sync, manually disabling AutoChange will stop the Secret from being retried.

Setting the Password Change Retry Interval

Secret Server checks for expired secrets once every minute. However, if a previous password change attempt failed, Secret Server will not immediately try to change that password again the next time expired secrets are checked. Each secret template has a "Retry Interval" that it uses to determine how often to attempt retries of failed password changes. This exists to prevent unavailable machines or network

connection issues from overwhelming the server or network with potentially thousands of password change requests at once. The default retry interval is one hour. To change the default, navigate to **Admin | Secret Templates**, select the template you wish to change from the dropdown menu, and click **Edit**.

Manage Secret Templates

Unix Account (SSH) ▾ Show Inactive

← Back
✎ Edit
+ Create New
↗ Export
📄 View Audit
☑ Active Templates
✳ Password Requirements
Ⓐ Character Sets

⚙️ Configure Launchers
👤 Configure Secret Template Permissions

At the bottom of the **Secret Template Designer** page, click **Configure Password Changing**

Fields

Field Name	Field Description	Field Type	Is Required?	History	Searchable	Edit Requires	Hide On View	Expose For Display	
Machine	The Server or Location of the Unix Machine.	Text	✓	All	✓	Edit	-	-	🗑️ ✎ ⬆️ ⬇️ ☰
Username	The Unix Machine Username.	Text	✓	All	✓	Edit	-	-	🗑️ ✎ ⬆️ ⬇️ ☰
Password	The password of the Unix Machine.	Password	✓	All	-	Edit	-	-	🗑️ ✎ ⬆️ ⬇️ ☰
Notes	Any additional notes.	Notes	-	All	✓	Edit	-	-	🗑️ ✎ ⬆️ ⬇️ ☰
* <input style="width: 100px;" type="text"/>	<input style="width: 150px;" type="text"/>	Text ▾	<input type="checkbox"/>	<input type="checkbox"/> ☑ All	<input type="checkbox"/>	Edit ▾	<input type="checkbox"/>	<input type="checkbox"/>	+

Show Inactive Fields

← Back
✳ Assign Password Requirement
⚙️ Configure Password Changing
👤 Configure Launcher
🔗 Configure Extended Mappings
📄 View Audit

📄 Copy Secret Template

On the **Secret Template Edit Password Changing** page, click **Edit**. Adjust the **Days**, **Hours**, and **Minutes** values of the **Retry Interval**. You can also adjust the **Maximum Attempts** if you want Secret Server to stop attempting to change the password after a specified number of failures. Click **Save** when done.

Secret Template Edit Password Changing

Enable Remote Password Changing

Retry Interval

Days

Hours

Minutes

Maximum Attempts

Unlimited

Enable Heartbeat

Heartbeat Check Interval

Days

Hours

Minutes

Password Type to use

Unix Account Custom (SSH) ▼

Machine Name

Machine ▼ *

Password

Password ▼ *

User Name

Username ▼ *

 Save

 Cancel

Configuring Common Criteria

Verify Settings in Government Edition

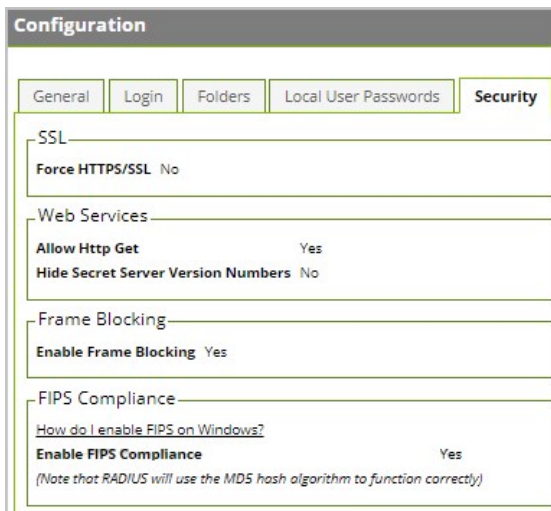
Verify DPAPI Setting is Enabled

The Delinea Government Edition Installer will automatically set encryption in Secret Server to use DPAPI. To verify that DPAPI is in use, navigate to **Admin | Configuration | Security** tab and ensure that DPAPI is enabled by scrolling to the bottom of the page and checking the existence of the **Decrypt Key to not Use DPAPI** button.

If DPAPI is not enabled in your Secret Server installation, this button will say **“Encrypt Key to Use DPAPI.”** To maintain compliance with Common Criteria standards do not decrypt the Secret Server key.

Verify FIPS Mode is Enabled

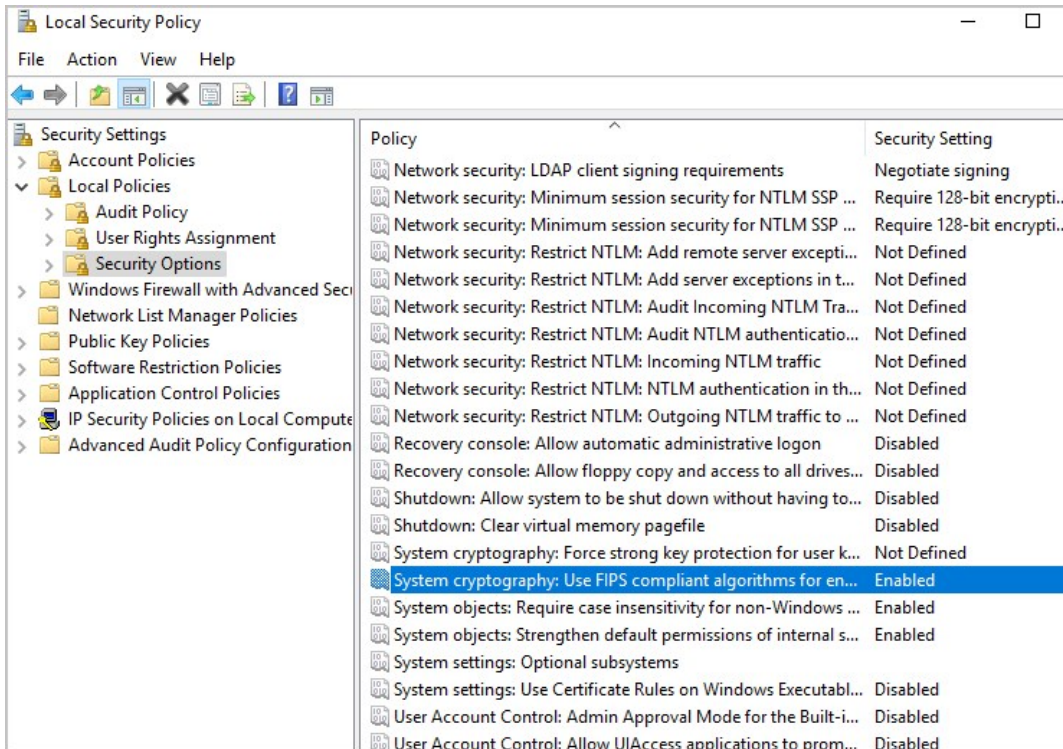
Common Criteria evaluations rely on FIPS 140 validations to provide assurance that cryptographic functionality is implemented properly. The Secret Server Government Edition Installer checks for FIPS encryption security standards on the Windows local server and enables FIPS mode by default in Secret Server. To check this setting, go to **Admin | Configuration | Security** tab and ensure that the **Enable FIPS Compliance** line is set to **Yes**.



Manually Enabling FIPS on Windows Local Server

The **Secret Server Government Edition Installer** ensures that FIPS Compliance is automatically enabled in Windows on your local server. To verify or do this manually:

1. Go to **Windows Local Security Policy editor** (secpol.msc)
2. Navigate on your left pane to **Security Settings | Local Policies | Security Options**
3. Find and go to the property of **System Cryptography: Use FIPS Compliant algorithms for encryption, hashing, and signing...**
4. Choose **Enable** and click **OK**.



1. Restart your Server with **iisreset**

The Microsoft .NET implementations of AES and SHA are not FIPS certified so Secret Server uses the Windows API versions for encryption functionality which *are* FIPS certified. **More information on the FIPS certificate numbers for the Windows operating systems, including the algorithm implementations that we use can be found at:** <http://technet.microsoft.com/en-us/library/cc750357.aspx>

Manually Disabling TLS v1.0

To disable the TLS 1.0 protocol so that IIS does not try to negotiate using the TLS 1.0 protocol, follow these steps:

1. Click **Start**, click **Run**, type regedt32 or type regedit, and then click **OK**.
2. In Registry Editor, locate the following registry key:

HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Server

- *If the subkeys "Server" or "Client" do not exist, create them under the
 - TLS 1.0 key.
1. On the **Edit** menu for each subkey, click **Add Value**.
 2. In the **Data Type** list for each subkey, click **DWORD**.
 3. In the **Value Name** box for each subkey, type "**Enabled**," and then click **OK**.
*Note If this value is present, double-click the value to edit its current value.
 4. Type **00000000** in Binary Editor to set the value of the new key equal to "0".

5. Click **OK**. Restart the Windows Server.

More information can be found at <https://support.microsoft.com/en-us/help/187498/how-to-disable-pct-1-0-ssl-2-0-ssl-3-0-or-tls-1-0-in-internet-informat>

Microsoft Cryptographic Engine

Secret Server relies on Windows Server 2016 Standard Edition to provide protocol and cryptographic functionality. Windows Server 2016 Standard Edition implements a FIPS certified (CMVP #2937) Cryptographic Primitives Library.

More information on the FIPS 140 certificate numbers for the Windows operating systems, including the algorithm implementations that we use can be found at: <http://technet.microsoft.com/en-us/library/cc750357.aspx>

Warning: The use of other cryptographic engines was not evaluated nor supported in the Common Criteria configuration.

Configuring X.509v3 Certificates

In order to maintain compliance with Common Criteria standards, the following installed certificates must follow the standard X.509v3 digital certificate structure.

**Do not use self-signed certificates in production environments.*

Installing Certificate in Trust Store

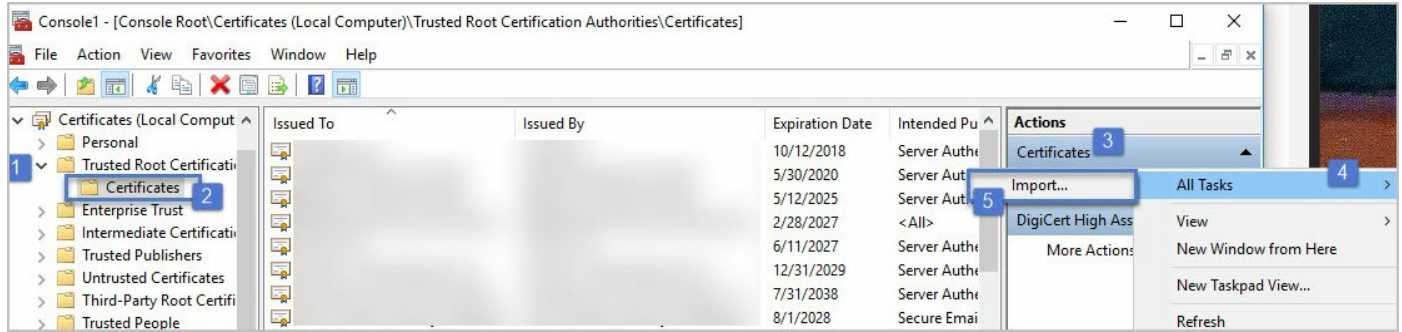
To best protect the users of your Secret Server website, use an SSL certificate from a trusted root authority and install it on your **web server(s)**.

- It is possible to use a self-signed certificate in demo or sandbox environments as long as **all other machines** used to access Secret Server are also configured to trust the certificate. Otherwise, users will receive warnings about it being invalid and unsafe to navigate to your Secret Server URL.
- Local machines either need to:
 - Import your **public certificate** into the **Trusted People** certificate store if you used IIS to create it
- OR
- If you used OpenSSL and set up a **custom Root CA** to sign your web server certificate, the Root CA must be imported into the **Trusted Root Certification Authorities** store.
- If a **custom Intermediary CA** is created, this must also be imported into the **Intermediate Certification Authorities** store.

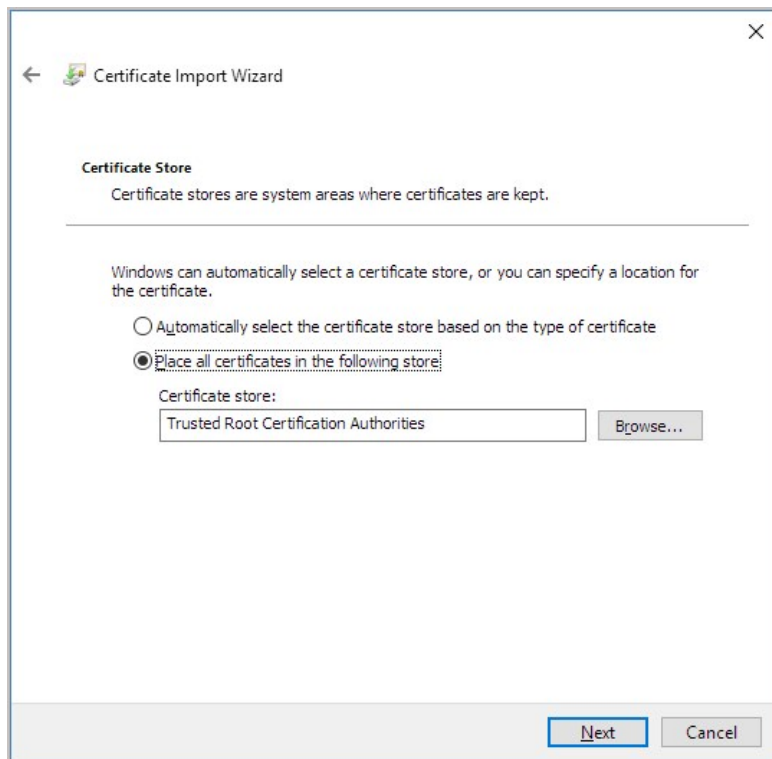
If Secret Server will be communicating with any remote server using TLS, any Root CAs and Intermediate CAs in the trust chains of the certificates presented by those remote servers must be added to the **Trusted Root Certification Authorities** and **Intermediate Certification Authorities** stores respectively. This includes syslog and LDAPS servers.

Follow the steps below to install a certificate in the Trusted Root CA store:

1. On your local machine open **mmc.exe**, select **File**, then **Add/Remove Snap-In...**
2. Next choose **Certificates** and then **Add**. Select **Computer Account**, then **Local Computer**. Click **Finish**.
3. Open **Certificates (Local Computer)**, then expand the **Trusted Root Certification Authorities** store/folder from the left pane and select **Certificates**, then in the right-side **Actions** pane select **More Actions, All Tasks**, and **Import...**



4. Click **Next** then select the certificate to import. This is usually a file with a file extension of .cer, .crt, .pem, or .pfx and click **Next**.
5. If prompted, enter the certificate's password and click **Next**.
6. Leave the selected options at the **Certificate Store** step and click **Next**.



7. Click **Finish** to import the certificate

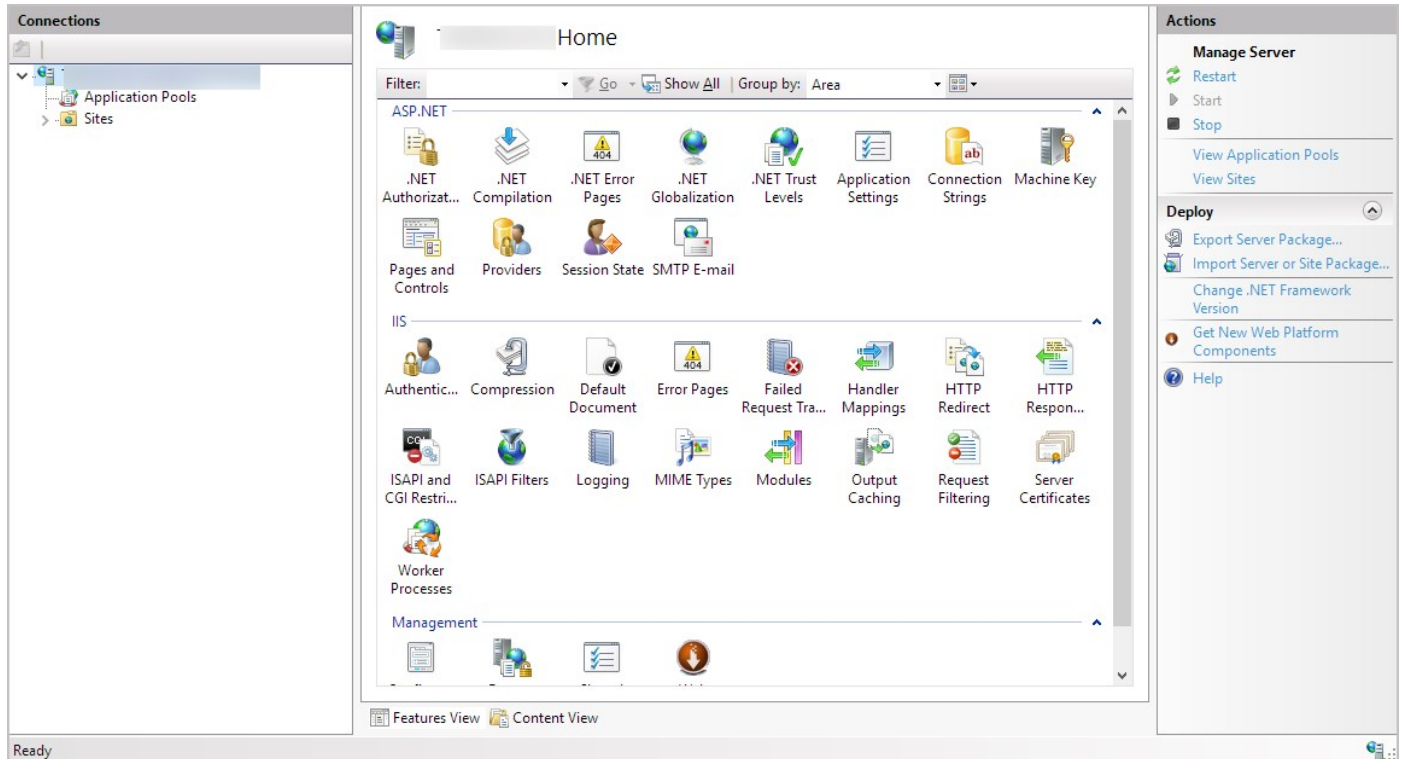
To import and Intermediate Certificate Authority follow the same steps for a Root Certificate Authority but select **Intermediate Certification Authorities** in step 3 instead of **Trusted Root Certification Authorities**. In step 6, the selected certificate store should also say **Intermediate Certification Authorities**.

Configuring Web Certificate

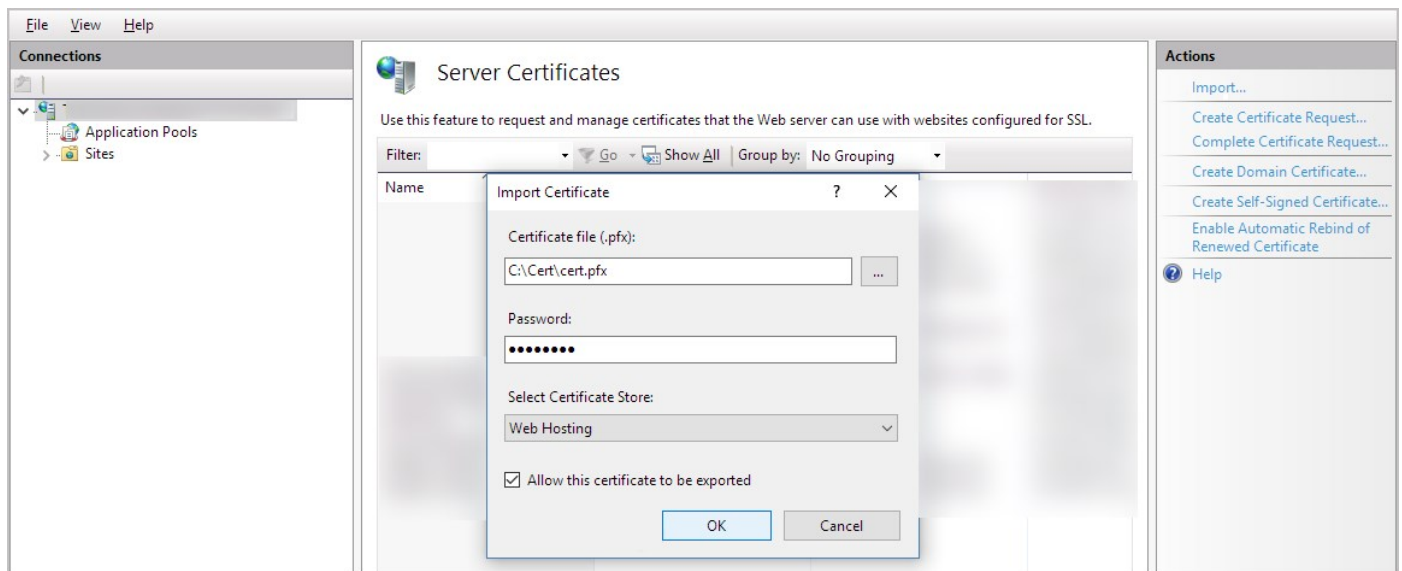
To install a web certificate for use by IIS:

1. Open IIS Manager by clicking **Start**, clicking **Run**, typing in "inetmgr", and clicking **OK**.

- Click on the server node (one of the root nodes) in the left panel, and double click on **Server Certificates**.



- Click **Import...** in the right-hand **Actions** panel



- Select the certificate to use, enter the password, select "Web Hosting" in **Select Certificate Store**, and click **OK**.

Alternatively, to create a self-signed certificate, click **Create Self-Signed Certificate** on the right panel and type in a Friendly Name.

**Note, using Self-Signed certificates is not compliant with Common Criteria standards. If using a self-signed certificate you must also export and install it on every user machine that connects to the Secret Server website.*

Now that your certificate is installed, IIS must be configured to use it.

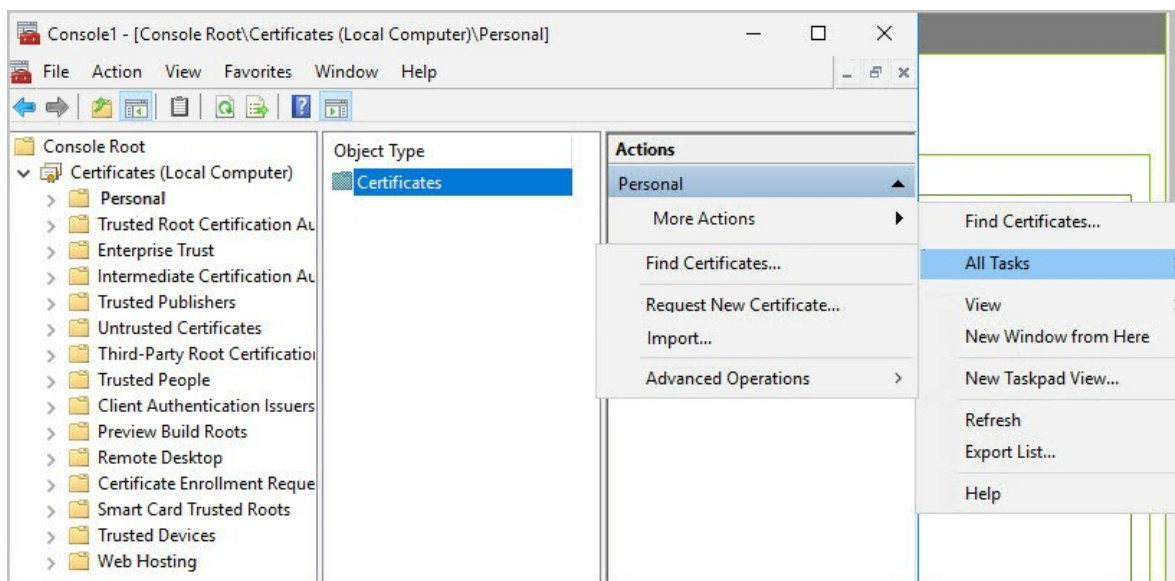
- In IIS Manager, click on your website in the left panel, then click **Bindings** on the right panel.
- If https is not already listed, click **Add**, select **https**, select the correct certificate, and click **OK**.
- If https was already listed, select it, click **Edit**, and select the correct certificate.

Configuring Client Certificates

Client Certificates can be used for **LDAPS** connections and for **Secure TCP** Syslog connections if your LDAPS or Secure TCP Syslog server requires one to be used.

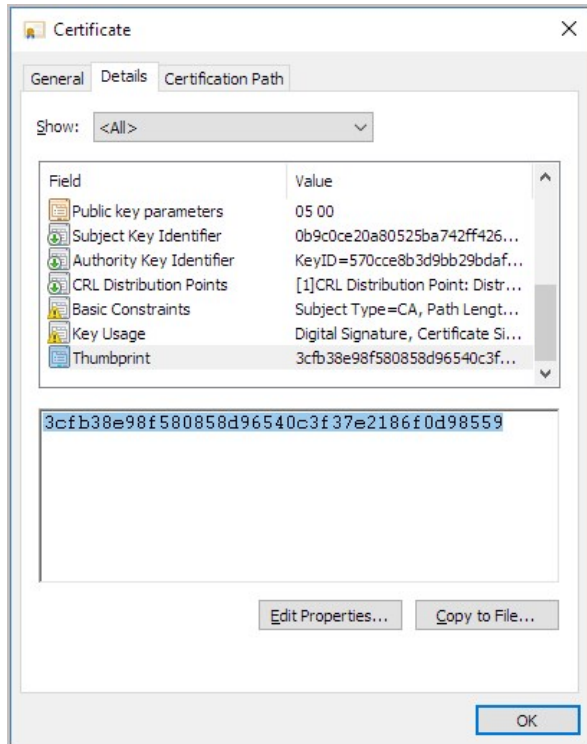
You will need to install a Client Certificate on your **web server(s)** in the Local Computer's **Web Hosting** (used by newer versions of IIS) or **Personal** Stores:

1. On your local machine open **mmc.exe**, select **File**, then **Add/Remove Snap-In...**
2. Next choose **Certificates** and then **Add**. Select **Computer Account**, then **Local Computer**. Click **Finish**.
3. Open **Certificates (Local Computer)**, then import your certificate by selecting the **Personal** or **Web Hosting** store/folder from the left pane, then **All Tasks**, and **Import...**




4. Follow the Import Wizard to complete the installation of the client certificate.
**Note: If you are using an older version of IIS, or IIS is not installed on your machine, the Web Hosting store will not exist. In that case, import into the Personal store.*
 - To enable Secret Server to send the client certificate to remote servers,
 - you must first get the certificate thumbprint. This thumbprint will be added
 - to Secret Server and used to identify which certificate to send when a
 - client certificate is requested.
 - To get the certificate thumbprint:


1. Right-click on the certificate file in Windows Explorer or the certificate listed in MMC and select **Open**.
2. Select the **Details** tab.
3. Scroll down in the list of fields until you find **Thumbprint**. Select it. The value will be displayed in the box below the field list.




4. Select the text in this box and copy it by pressing **Ctrl+V**.
- Now that you have the thumbprint copied to the clipboard, you can add it to
 - Secret Server:
1. In Secret Server, select **ADMIN > Configuration**.
 2. Select the **Security** tab and click **Edit**.
 3. Find the **TLS Auditing** section.

TLS Auditing


Apply TLS Certificate Chain Policy and Error Auditing 


Additional Certificate Chain Policy Options 

[What are X509 Certificate Chain Policy Options?](#)

Enable TLS Debugging and Connection Tracking 

Advanced (not required)

 Secret Server's IIS AppPool must be granted permission to use the Client Certificate, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). Example usage:
`winhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "Certificate Subject" -a "HOSTNAME\IIS APPPOOL\SecretServer"`
[Download WinHttpCertCfg](#) - [Official WinHttpCertCfg documentation](#)

Client Certificate Thumbprint(s)  3cfb38e98f580858d96540c3f37e2186f0d98559 x 706e96862acb755bdf7f1a840d7aca7cdb503684 x

4. Check **Apply TLS Certificate Chain Policy and Error Auditing**.
5. Click **Advanced (not required)** to expand that section.
6. Paste the thumbprint in the **Client Certificate Thumbprint(s)** field. If you need to use more than one client certificate (for example, if you have multiple Secret Server nodes with different client certificates), you can paste multiple thumbprints in this field.
7. Click **Save** when done.

Certificate Identifiers Overview

Certificates used for server authentication are verified by Secret Server using the verification provided by Microsoft Windows' Schannel. A certificate must contain a reference identifier that the client can use to verify that the server is presenting a matching certificate. This identifier can be contained in either the certificate **Subject** or **Subject Alternative Name (SAN)**. In order for the Subject to be verified as valid for a given server, it must contain a CN entry that contains the **Fully-Qualified Domain Name (FQDN)** of the server. If a certificate needs to be valid for more than one reference identifier, the certificate can contain a SAN that contains multiple entries. These entries can include both DNS names and IP addresses. If a certificate contains a SAN, the CN in the subject is ignored and only the entries in the SAN are used to validate the certificate against the server.

Configuring Transport Layer Security (TLS)

Common Criteria users must configure TLS in Secret Server. To enable TLS, follow the steps in the below sections:

TLS Configuration with AD

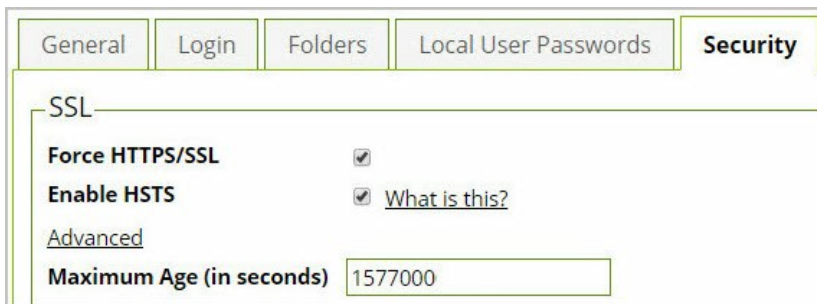
To configure TLS with Active Directory, follow the steps in section **7.1.3 Configuring TLS with Active Directory**.

TLS Configuration with Syslog

To configure TLS with Syslog, follow the steps in section **13.3 Configuring Auditing for TLS Connections**.

TLS Configuration with IIS

According to Common Criteria security standards, you must force Secret Server through HTTPS/SSL for all website connections, ensure you are using an https URL in your Secret Server browser, then navigate to **Admin | Configuration | Security** tab, click **Edit** and then check the **Force HTTPS/SSL** checkbox, followed by the **Enable HSTS**. We recommend adjusting the Maximum Age (in seconds) to a high number, the example below allows for approximately six months, but we recommend up to one year. **Save** changes.



The screenshot shows the IIS configuration interface with the 'Security' tab selected. Under the 'SSL' section, the following settings are visible:

- Force HTTPS/SSL**:
- Enable HSTS**: [What is this?](#)
- Advanced**: [Advanced](#)
- Maximum Age (in seconds)**:

For more information on HSTS, see [Securing Traffic with HTTP Strict Transport Security](#)

Local Auditing

Understanding Local Audit Records and Reports

By design, Secret Server locally audits all actions taken within Secret Server. Secret Server's auditing capacity is not configurable. See **Auditable Events** for details on Secret Server items that are audited and corresponding user permissions required for accessing audit records.

Various User Permissions are tailored for specific kinds of audits, listed below. To adjust user permissions, go to **Admin | Roles** and create a new role or click an existing role to edit the permissions on it. You can assign roles to individual users by going to **Admin | Users** and assigning roles.

- **Add Secret Custom Audit**
Allows a user to make a custom audit entry when accessing a Secret using the web services API.
- **User Audit Expire Secrets**
Allows a user to view the User Audit report, which shows all secrets that have been accessed by a particular user in a specified date range. Also allows the user to force expiration on all these secrets, which would make Secret Server automatically change the password.
- **View Configuration Unlimited Admin**
Formerly "View Unlimited Admin Configuration," allows a user to view the Unlimited Admin Mode configuration. Also allows a user to view the Unlimited Admin Mode audit log.
- **View Secret Audit**
Allows a user to view Secret Audit.
- **View User Audit Report**
Allows a user to view, but not edit, the User Audit Report.

Accessing Local Reports

To view a list of out of the box reports, navigate to the **Reports** tab from an Administrator account.

Reports

General | Security Hardening | User Audit

Secrets

- [What Secrets can all users see?](#)
- [What Secrets can a user see?](#)
- [What Secrets have been accessed?](#)
- [What Secrets have been accessed by a user?](#)
- [What Secret permissions exist?](#)
- [What Secret permissions exist for a user?](#)
- [What Secrets changed passwords in the last 90 days?](#)
- [What Secrets have not changed passwords for over 90 days?](#)
- [What Secret permissions exist for a group?](#)
- [What Secrets don't require approval?](#)
- [What Secrets have failed Heartbeat?](#)
- [What Secrets require approval?](#)
- [Secret Count per Site](#)
- [What Secrets Do Not Have Distributed Engines?](#)
- [What Secrets require Comments?](#)
- [What Secrets have been accessed by an impersonated user?](#)
- [What Secrets are expiring this week?](#)
- [What Secrets Have Distributed Engines?](#)
- [What Secrets have Expiration?](#)
- [Secret Permissions Mismatch.](#)
- [What SSH Command Menus do Secrets have?](#)

User

- [Failed login attempts](#)
- [Who hasn't logged in within the last 90 days?](#)
- [What SSH Command Menus do users have access to?](#)
- [Secret Template Permissions by User](#)
- [What users have had an admin reset their password?](#)

Activity

- [Secret Activity](#)
- [Secret Activity Today](#)
- [Secret Activity Yesterday](#)
- [Folder Activity](#)
- [Users Activity](#)
- [Custom Report Activity](#)
- [Dual Control Audit](#)
- [Internal Communication Changes](#)
- [IP Address Range Audit](#)
- [Unlimited Administrator behavior](#)
- [Distributed Engine Activity](#)
- [Event Subscription Activity](#)
- [Secret Template Activity](#)

Groups

- [Group Membership](#)
- [Group Membership By Group](#)

Roles and Permissions

- [What role permissions does a user have?](#)
- [What role assignments exist?](#)
- [What role permission assignments exist?](#)

Legacy Reports

- [Secret Server Usage](#)
- [Secret Expiration Health](#)
- [Secret Template Distribution](#)
- [Top Ten Viewers](#)

Discovery Scan

- [Discovery Scan Status](#)
- [What computers in Active Directory no longer exist?](#)
- [What computers have been successfully scanned?](#)
- [What computers that exist have not been successfully scanned?](#)

Password Compliance

- [Secret Password Compliance Statuses](#)
- [What Secrets Do Not Meet Password Requirements?](#)

Secret Policy

- [What Folders have Policies assigned?](#)
- [What Secrets have different Policies than their folders?](#)

Folders

- [What folders can all users see?](#)
- [What folder permissions exist?](#)
- [What folder permissions exist for groups?](#)

For customized reports and creation, see the options available to you at the bottom right corner of the Reports page. The **View Audit** button on the bottom left corner will show you a local audit for any reports that have been viewed.

Folders

- [What folders can all users see?](#)
- [What folder permissions exist?](#)
- [What folder permissions exist for groups?](#)
- [What folders can a user see?](#)

Secret Policy

- [What Folders have Policies assigned?](#)
- [What Secrets have different Policies than their folders?](#)
- [What Secrets have policies assigned?](#)

Discovery Scan

- [Discovery Scan Status](#)
- [What computers in Active Directory no longer exist?](#)
- [What computers have been successfully scanned?](#)
- [What computers that exist have not been successfully scanned?](#)
- [What Secrets failed to import by Discovery?](#)
- [What Secrets are pending import by Discovery?](#)

Report Schedules

- [Report Schedules](#)

[Edit](#) [View Audit](#)

Can't find the report you need? [Create it](#), [request a custom report](#) or [search the Reports Gallery](#).

Administrator actions have audit trails inside Secret Server, these are generally found as a **View Audit** button on any landing page under the **Admin** tab. For example, at the bottom of **Admin | Configuration** you can view the audit for every edit that has occurred on the **Configuration** page:

[Back](#) [Edit](#) [View Audit](#) [Change Administration Mode](#)

For User Audits, navigate to **Reports | User Audit** tab, then select a user and time period to view the audit of all actions taken by that user in Secret Server.

Configuring Local Windows Event Log Auditing

Outside of Secret Server, you can also send audit logs as EVT records to your Windows Server's Windows Event Log locally**. **

Note that the Windows Event Log is tied to the syslog implementation, because of this, disabling syslog will disable Windows Event Logs as well. If Syslog/CEF Logging is not yet enabled within Secret Server, go to **Admin I Configuration** and **Edit** the General tab to check **Enable Syslog/CEF Logging** followed by **Write Syslogs As Windows Events**. *Syslog/CEF Logging configuration also detailed in section 13.3 Configuring Auditing for TLS Connections*

The following steps walk you through how to grant required permissions to the Application Pool outside of Secret Server to successfully write audits to the Windows Event Log:

Granting Application Pool Access to Windows Event Log

When the database becomes inaccessible, Secret Server will try to log errors to the Windows Event Log. By default, however, Network Service and standard service accounts will not have permissions to the Event Log. *Permissions must be manually added to the Application Pool for specific Event Log registry keys.*

- **IMPORTANT NOTE:** Changes made to the Windows registry happen immediately, no backup is created automatically. Do not edit the Windows registry unless you are confident about doing so. Using Registry Editor incorrectly can cause serious, system-wide problems that may require you to re-install Windows to correct them. Microsoft cannot guarantee that any problems resulting from the use of Registry Editor can be solved. Use this tool at your own risk.

Required Registry Permissions

The following permissions are required for the Identity configured on the Secret Server Application Pool in IIS (Network Service, IIS APPPOOL\SecretServer, etc.).

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog

Applies to "This key and subkeys".

- Read
 - Query Value
 - Enumerate Subkeys
 - Notify
 - Read Control

- Set Value
- Create Subkey

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security

Applies to "This key and subkeys".

- Read
 - Query Value
 - Enumerate Subkeys
 - Notify
 - Read Control

How to Apply Windows Event Log Permissions

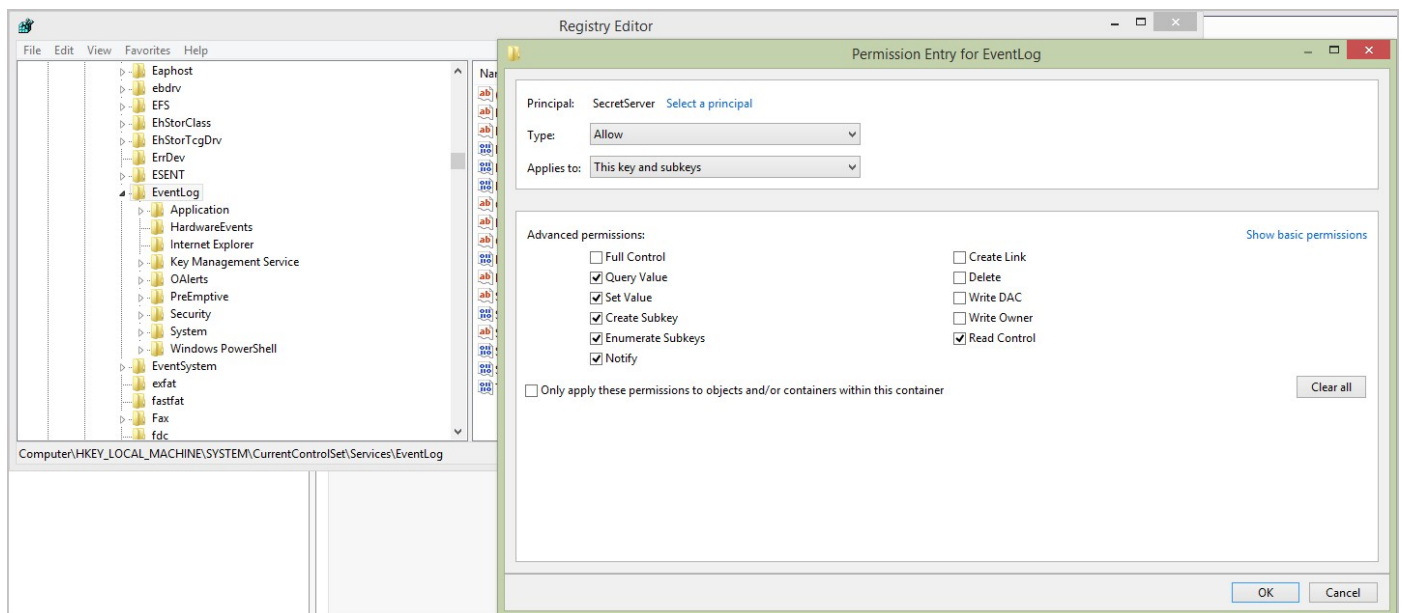
1. Determine the account that is running Secret Server. This can be done by Secret Server. logging in to Secret Server, **Admin I Diagnostics**. Click the button to **Show Background Processes**. Look for any of the "Thread Identity" labels. These will contain the identity of Secret Server (often NT AUTHORITY\NETWORK SERVICE or IIS APPPOOL\SecretServer).

Background Processes	
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

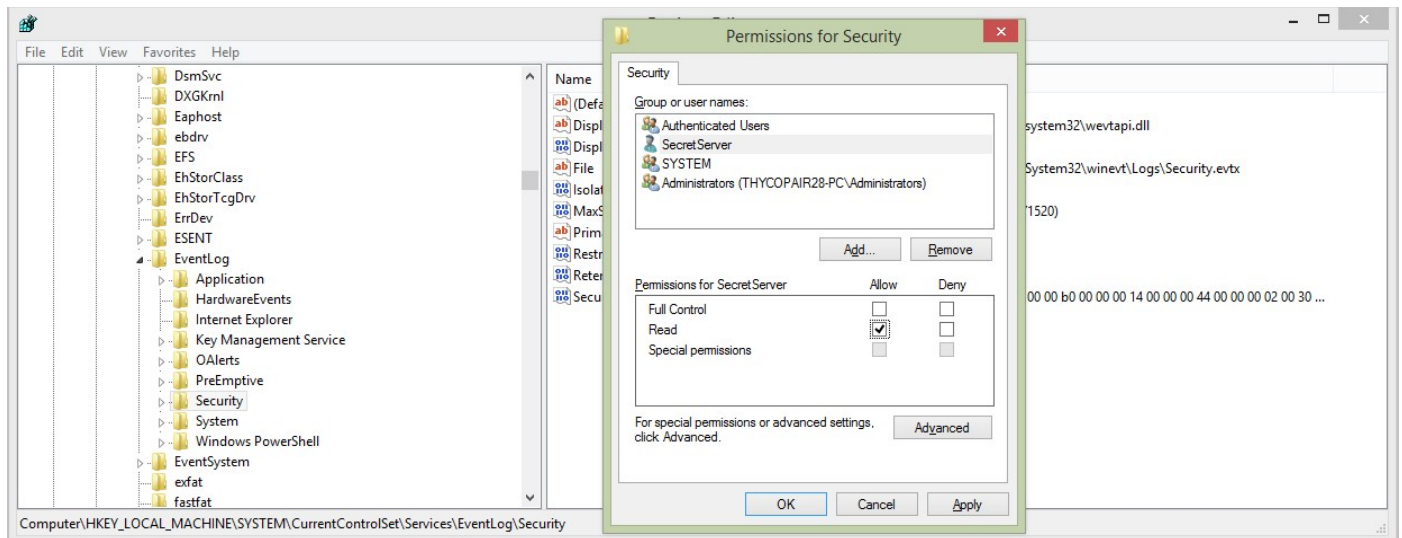
Background Processes	
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

Background Processes	
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[REDACTED]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

2. Open the **Registry Editor** on the machine running Secret Server (**Start | run-regedit**, depending on OS)
3. On the left, navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog**
4. Right click on the **EventLog** folder in your registry editor and select **Permissions**, then click the **Advanced** button
5. On the Permissions tab, Click **Add**
6. Click **Select a principal**.
7. Enter the name of your app pool's account (see step 1, for example: "IIS APPPOOL\SecretServer") in the box listed under **Enter the object name to select (examples)**. Click **Check Names**, Once "SecretServer" (or the name of your Secret Server app pool) is listed in the box and underlined, click **OK**
8. Under **Basic permissions:**, check **Read**.
9. Click **Show advanced permissions**, then check **Set Value** and **Create Subkey** under **Advanced permissions:**.



10. Click **OK** on the remaining dialogs and **Apply** the permissions. You should be back on the main Registry Editor window.
11. Navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security**, right-click and select **Permissions...**
12. Click **Add...**, find the account running Secret Server (like in step 7), then click **OK**.
13. Check **Read** in the **Allow** column, then click **OK** to apply the permission.

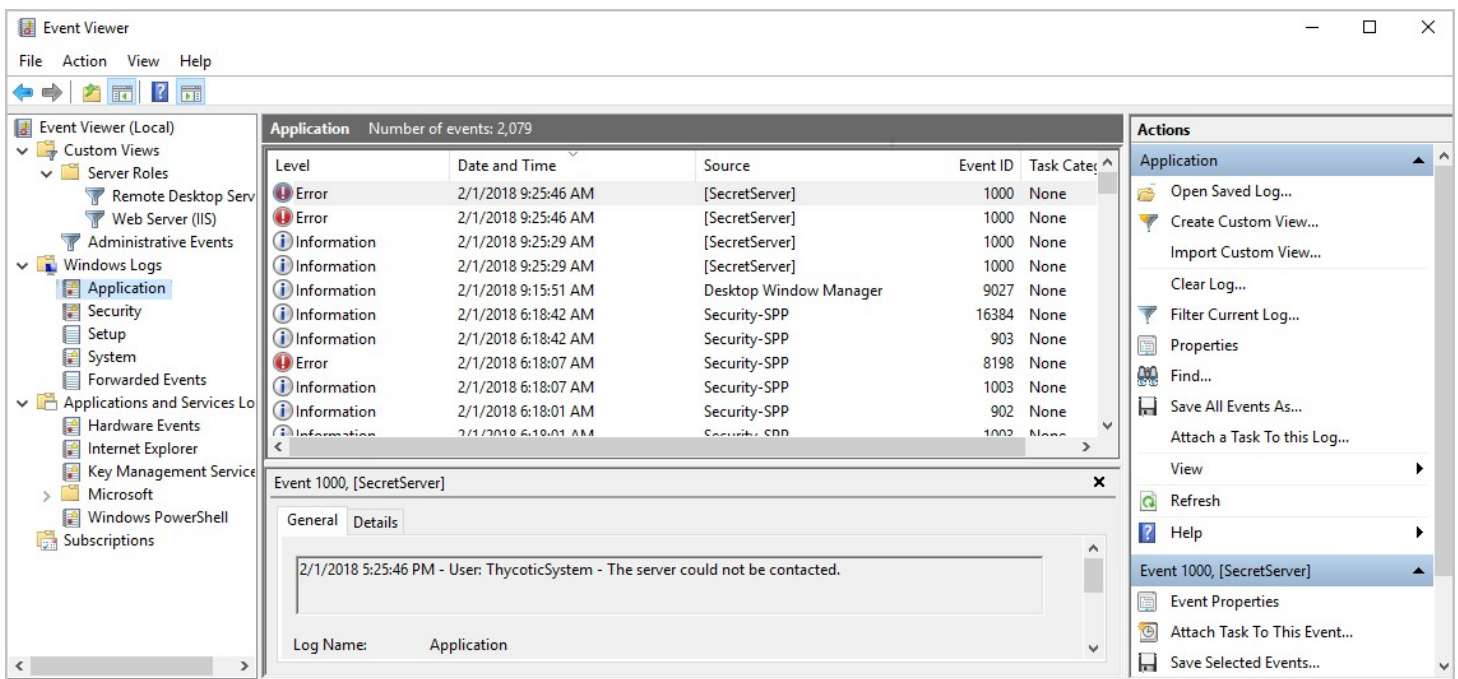


- In Secret Server, navigate to **Admin | Configuration | General** tab to verify that the Application Setting **Write Syslogs as Windows Events** is set to **Yes**.

Accessing Windows Event Logs

After setting up Windows Event Logs to consume Secret Server logs you can access logs for local auditing and troubleshooting purposes through the **Windows Event Log Event Viewer** on Secret Server's local server.

To find the Windows Event Log, open the Windows **Event Viewer** from the local server and navigate to **Windows Logs | Application**. Event Logs for Secret Server will be listed as **[SecretServer]** under the **Source** column:



External Auditing

Security - Connecting to External Audit Server

Using the **Secure TCP** Syslog/CEF Protocol will make Secret Server try to establish a secure connection to your external audit server. Secret Server will only use TLS v1.2 or v1.1 for security reasons.

If your audit server requires Client Certificates, see **Configuring Client Certificates**. If configured, Secret Server will pass the Secret Server certificate to the external audit server when it connects, so the audit server can validate the Secret Server machine.

If the connection between Secret Server and the external audit server is disconnected or cannot be established, Secret Server will log an internal error and automatically try to re-send any missed audit messages later. For more information, see **Determining Status of a Remote Audit Server**.

For a connection to be successfully established using Secure TCP protocol, Secret Server must trust the SSL certificates being used by the audit server. If the audit server is using a self-signed certificate, the Certificate Authorities that created it must be installed on the Secret Server machine. If Secret Server has issues trusting the certificate, details will be logged internally.

More information on creating your own Certificate Authorities can be found at the [OpenSSL Certificate Authority](#).

Configuring Syslog/CEF External Audit Server

Compatible Audit Servers

- syslog-ng
- Any Audit server that accepts TLS encrypted messages using the BSD Syslog Protocol

Configuring Your External Audit Server

1. Navigate to **Admin | Configuration**, then click **Edit**.
2. You will see a configuration area in the **General tab** under **Application Settings** to **Enable Syslog/CEF Logging**. Check this box to enable:

Enable Syslog/CEF Logging	Yes
Syslog/CEF Server	Syslog.Example.com
Syslog/CEF Port	6514
Syslog/CEF Protocol	SECURE TCP
Syslog/CEF Time Zone	Server Time

Note: Syslog/CEF may require an additional license key. To install licenses, navigate to **Admin | Licenses** and select **Install New License**. Once installed, you will need to activate your license. Contact your Delinea Sales Representative if you have questions about your licensing.

- **Syslog/CEF Server.** Configure the Syslog/CEF IP address for the IIS Server hosting the Syslog/CEF web application.
- **Syslog/CEF Port.** Next, configure the port number where the logging information will be passed for the **Syslog/CEF Port**. 6514 is the default port for Secure TCP Syslog. Secret Server requires outbound access to this server and port so communication can pass freely.
- **Syslog/CEF Protocol.** Set the Syslog/CEF Protocol to **Secure TCP**. This setting will accept either TLS v1.2 or v1.1 for added security, because other versions of SSL (i.e. SSL v3 and TLS v 1.0) have known weaknesses.

- **Syslog/CEF Time Zone.** Lastly set Syslog/CEF Time Zone to **UTC Time** or **Server Time** depending on your preference.

Caching Syslog Audits

Once secure Syslogging is enabled in Secret Server, if the connection breaks between the external Syslog server and Secret Server, failed syslog messages will be cached in the Secret Server database and re-sent at regular intervals until the connection between the syslog server and Secret Server is reestablished.

Compatibility Notes Related to Using Client Certificates

If you are using a Client Certificate, Secret Server's IIS AppPool must be granted access to use the certificate using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). Otherwise if Secret Server is configured to use a Client Certificate, and IIS doesn't have permission, you will see errors like this in the logs:

TLS Error Detected (Authentication Error connecting to IP:PORT) - The credentials supplied to the package were not recognized.

Example usage:

```
cd C:\Program Files (x86)\Windows Resource Kits\Tools
winhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "CertificateSubject" -a "HOSTNAME\IIS APPPOOL\SecretServer"
```

Download the [Windows HTTP Services Certificate Configuration Tool](#) (WinHttpCertCfg.exe).

If you are using a Client Certificate and a Syslog-NG logging server, you may occasionally see this message in the main Syslog-NG log file:

SSL error while reading stream; tls_error='SSL routines:ssl_get_prev_session:session id context uninitialized'

On the Secret Server side, it shows up like this:

TLS Error Detected (Authentication Error connecting to IP:PORT) - Authentication failed because the remote party has closed the transport stream.

This error is caused because Windows tries to cache secure connections when client certificates are in use, but because Syslog-NG has not configured their OpenSSL "session id context", OpenSSL gives this error when it tries to resume a previous session.

Secret Server will automatically reconnect and resend any missed messages, so these errors really should cause no impact. However if you would like, you can disable Windows' secure connection caching, by adding the [ClientCacheTime](#) setting set to 0 (zero) in the Registry, and then doing a reboot. This did not cause any significant performance impact in our internal testing, but your mileage may vary.

If Syslog-NG configures their OpenSSL session ID context, this error message correction will no longer be needed.

Configuring Auditing for TLS Connections

To track problems with TLS connections including whenever the connection might fail, enable the TLS Certificate Chain Policy and Error Auditing in Secret Server by navigating to **Admin | Configuration | Security** tab, then scroll down to the TLS Auditing section.

- Ensure that the **Apply TLS Certificate Chain Policy and Error Auditing** is set to **Yes**. If not set to yes, Client Certificates cannot be used.
- Ensure that the **Enable TLS Debugging and Connection Tracking** is set to **Yes**. When set to yes, Secret Server will send information logs to your audit server about when TLS connections are opened or closed. If debug logging is enabled in "web-log4net.config" detailed information about X509 certificate chain validation will also be logged. Note that this setting may create a lot of extra messages in your log files.

TLS Auditing

Apply TLS Certificate Chain Policy and Error Auditing Yes

Additional Certificate Chain Policy Options

[What are X509 Certificate Chain Policy Options?](#)

Enable TLS Debugging and Connection Tracking Yes

To Edit click the **Edit** button and **check** the setting you want to turn on or uncheck the setting you want to turn off.

If the TLS connection breaks, an error message will be logged in the local audit trail and Secret Server will keep trying every 60 seconds to reestablish the TLS connection to the syslog server.

If Secure TCP is used for the Syslog/CEF Protocol, and there are one or more Client Certificate Thumbprints entered, Secret Server will check the Local Computer's Web Hosting and Personal certificate store and use the first one it finds. For more information see **Configuring Client Certificates**.

To add Client Certificate Thumbprints, you can copy and paste a list in bulk after clicking Edit, then Advanced (not required). Separate each SSL Certificate SHA1 Thumbprint (40 characters each) with a semicolon (up to ten total are allowed).

Determining Status of a Remote Audit Server

To view the logs for any TLS-Connection related errors:

- Open **Microsoft SQL Server Management Studio**, navigate to your SecretServer database (**DBMachineName > Databases > SecretServer**) and run a **New Query**.
- Type **select * from tbSecurityAuditLog** to view the events from your TLS Audit:

SecurityAuditLogId	UserId	DateRecorded	EventActionId	LogSourceId	IpAddress	Notes	MachineName	DatabaseName
1	1	2017-12-06 16:19:07.470	10118	4		TLS Error Detected (Authentication Error connecting to 192.168.1.17:7514 - The credentials supplied to the package were not recognized	Secret Server	Secret Server
2	2	2017-12-06 16:24:19.090	10118	4		TLS Error Detected (Authentication Error connecting to 192.168.1.17:7514 - The credentials supplied to the package were not recognized	Secret Server	Secret Server
3	3	2017-12-06 16:32:31.010	10118	4		TLS Error Detected (Authentication Error connecting to 192.168.1.17:6514 - The remote certificate is invalid according to the validation procedure.	Secret Server	Secret Server

For more detailed troubleshooting reporting, refer to the logs in File Explorer on Secret Server's web server (C:\inetpub\wwwroot\SecretServer\log) including ss.log, ss-BSSR.log (Background Scheduler), ss-BSWR.log (Background Worker) for any errors that might crop up.

Management Functions

This section describes management activities and corresponding roles of the evaluated security functionality

Roles and Management Functions

Read-only User	Search and list Secrets
User	Use Secret/Launch session
User	Request access to Secret
Administrator	Create, view, expire, edit, and assign Secrets
Administrator	Perform bulk operations on Secrets
Administrator	Create and manage groups
Administrator	Create and manage roles, assign roles to users
Administrator	Create and manage containers (folders)
Administrator	Create and manage Secret policy
Administrator	Configure TOE SF (see Table 16)
Administrator	Create, manage, and unlock local accounts
Administrator	Configure IIS, SQL, syslog
Administrator	Update TOE

By default each User Role is attached to various Permission Sets. To view the specific permissions that each role is attached to, navigate to **Admin | Roles** and click into the user roles listed to see the list of permissions.

Organizations can tailor these user roles to maintain whatever permissions settings are required for your specific user environment.

The following table specifies the specific user roles required to allow each management activity listed.

ESM_EAU.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_EID.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_ICD.1	Definition of identity and credential data that can be associated with users (activate, suspend, revoke credential, etc.)	Administrator

	Management of credential status	Administrator
	Enrollment of users into repository	Administrator
ESM_ICT.1	Configuration of circumstances in which transmission of identity and credential data (and object attributes, if applicable) is performed	Administrator
FAU_STG_EXT.1	Configuration of external audit storage location	Administrator
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	Administrator
	Management of actions to be taken in the event of an authentication failure	Administrator
FIA_USB.1	Definition of default subject security attributes, modification of subject security attributes	Administrator
FMT_MOF.1	Management of sets of users that can interact with security functions	Administrator
FMT_SMR.1	Management of the users that belong to a particular role	Administrator
FTA_SSL.3	Configuration of the inactivity period for session termination	Administrator
FTA_TAB.1	Maintenance of the banner	Administrator
FTP_ITC.1	Configuration of actions that require trusted channel (if applicable)	Administrator
FTP_TRP.1	Configuration of actions that require trusted path (if applicable)	Administrator

Secret Server can be hosted externally, like any other IIS website. We recommend using all security measures for Secret Server that you would use for any server directly accessible to the internet. We also recommend the measures described below.

Limiting the Attack Surface

- The Secret Server application should reside on a dedicated server in a DMZ.
- Secret Server and its database should reside on separate servers. If a hole for SQL connections can be opened in the DMZ firewall, the database can reside on the other side of the DMZ firewall.

Using Secure Connections

- Use HTTPS to access to the website.
- Use SSL to connect to the Secret Server database.
- Use LDAPS to connect the web server to Active Directory.

Setting Up Remote Password Changing

By default, Secret Server changes passwords on devices and accounts directly from the web server where it is installed, but when Secret Server is installed in a DMZ zone, it does not have direct network connections to these devices and accounts.

However, you can enable Secret Server to change passwords throughout your network over a specified port using distributed engines. See [Distributed Engines](#) for more information on setting up and using Secret Server Distributed Engines.

Note: This document and the information contained in it are confidential and proprietary to Delinea and provided in strict confidence for the sole internal use of Delinea and authorized agents and may not be disclosed to any third party or used for any other purpose without express prior written permission of Delinea.

Note: The PDF version of this online document is automatically generated and thus may have minor formatting anomalies.

Note: This document is not updated with every Secret Server release—some minor releases do not affect the guide's contents and thus do not warrant a document update.

Introduction

Important: This document is closely associated with the Security Hardening Report in Secret Server (Click **Reports > Security Hardening**) and with the [Security Hardening Guide](#), which provides information on security hardening beyond Common Criteria. We recommend having those available while reading this document.

Overview

Secret Server made several security enhancements to achieve Common Criteria (CC) certification. These features are available in all versions of Secret Server 10.4 and later. Due to their stringency and need for additional user configuration, not all these features are enabled by default by our standard installer.

This guide provides the information an administrator needs to configure Secret Server 10.4 and above in compliance with the Common Criteria evaluated configuration. Follow this guide in its entirety to ensure each parameter setting matches those evaluated and certified as secure by Common Criteria standards.

Audience

This document is for administrators who are responsible for installing, configuring, and operating enterprise infrastructure for their organization. To use this guide, you must have knowledge of your organization's network infrastructure and applicable policies. In addition, you must have administrative access to configure your operational environment.

What Is Common Criteria?

The Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), known as "Common Criteria," is an international standard for security certification of computer systems, networks, and application software. The certification ensures that claims about the security attributes of the evaluated product were independently verified in the evaluated configuration in the same specific environment. The certification assumes a specific evaluated configuration and does not validate any security claims when the product is used outside of that configuration.

Procedures

Security Hardening Checklist

After installing Secret Server, navigate to the **Reports > Security Hardening** tab, and follow the checklist to ensure your environment is as secure as possible:

Note: See the [Security Hardening Guide](#) for details.

1563386730221

Configuring TLS

To achieve Common Criteria certification on Secret Server, you must enable Transport Security Layer (TLS).

Manually Disabling TLS Version 1.0

TLS 1.0 is no longer considered secure, so it is important to disable this version of the protocol on Secret Server. To do this, follow the instructions in the "Manually Disabling TLS v1.0" section of the [Common Criteria Hardening Guide](#).

TLS Diffie-Hellman Hardening Overview

For information on configuring your servers with stronger Ephemeral Diffie-Hellman hardening, see the "TLS Diffie-Hellman Hardening Overview" in the [Common Criteria Hardening Guide](#).

Restricting Server Cipher Suites for TLS

Allowed Suites

Common Criteria certification requires restricting the cipher suites configured on your server to only:

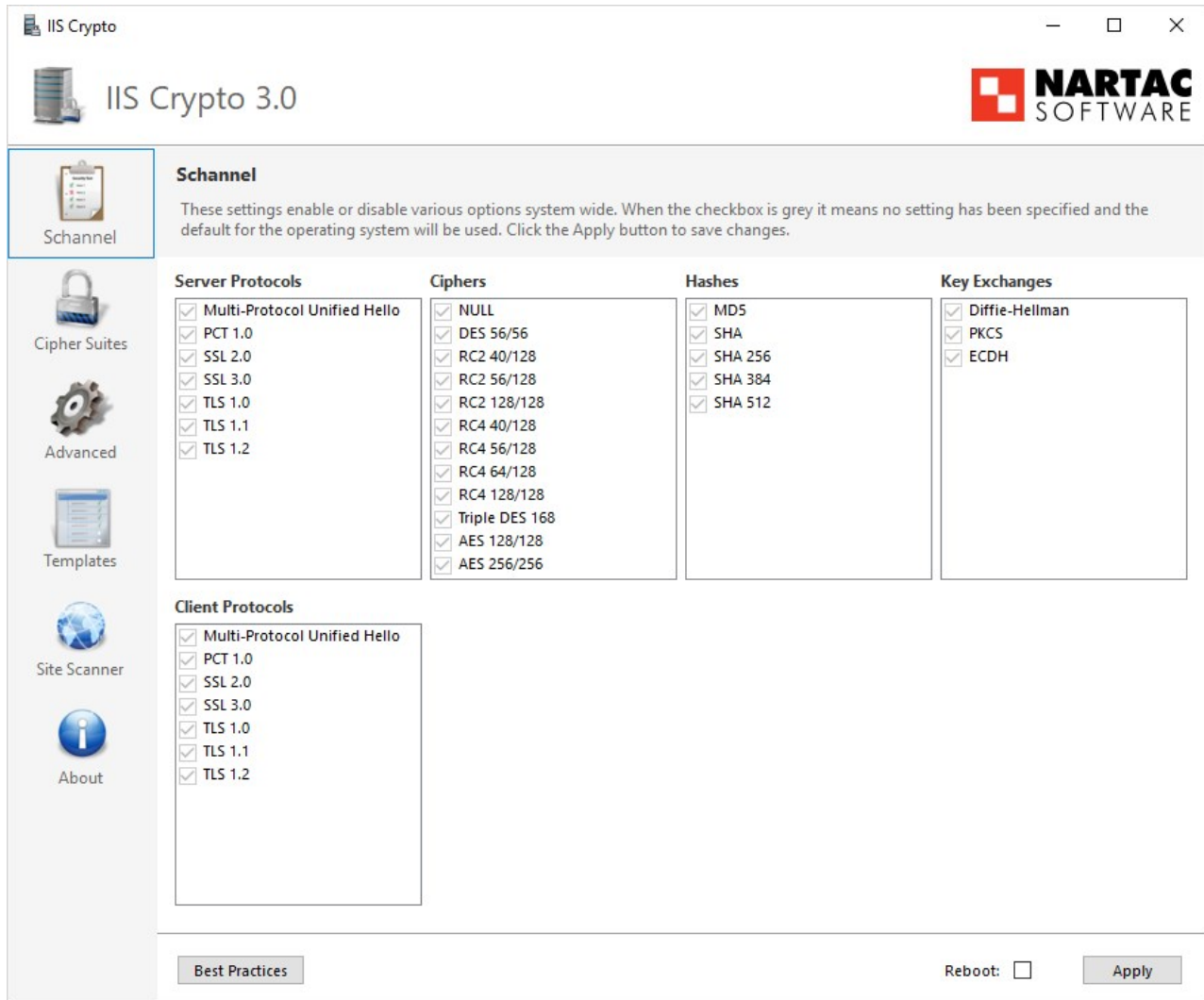
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA256

Restricting them can cause communication issues with other servers if they are not able to communicate using any of the above ciphers. In that case, you need to modify those servers to include these cipher suites to securely communicate according to Common Criteria guidelines.

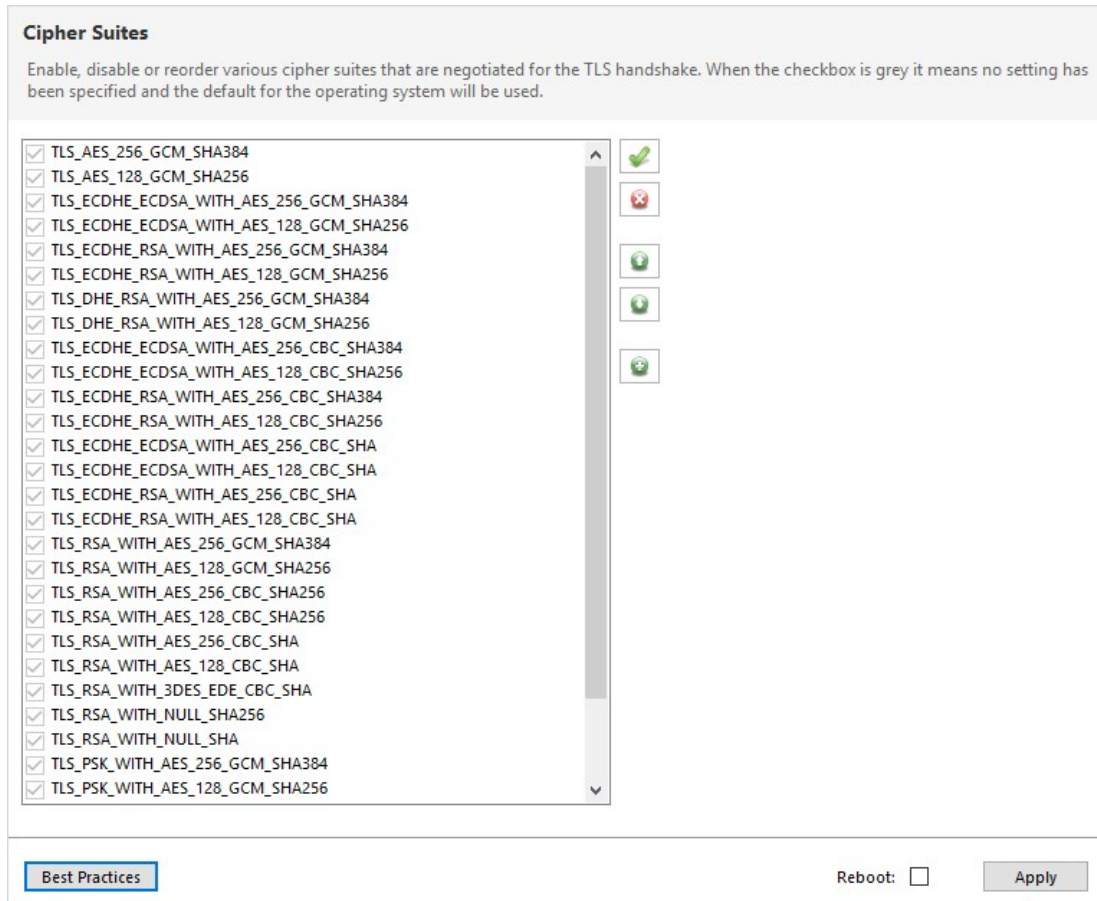
Changing Cipher Suites with the IIS Crypto Tool


One way to change the cipher suites on a computer is to use the free IIS Crypto tool:

1. Download the GUI version of the tool at: <https://www.nartac.com/Products/IISCrypto/Download>
2. Run the tool:



3. Click **Cipher Suites** button on the left. The Cipher Suites window appears:



4. Click the  **Uncheck All** button to uncheck all cipher suites.
5. Find and click to select the suites in the list above.
6. Click the **Apply** button.

Configuring TLS with IIS

Common Criteria certification requires using HTTPS/SSL for all connections to the Secret Server Web page. To do this, follow the instructions in the "TLS Configuration with IIS" section of the [Common Criteria Hardening Guide](#).

Enabling TLS Auditing

To have Secret Server audit TLS connections and connection failures, follow the instructions in the "Configuring Auditing for TLS Connections" section of the [Common Criteria Hardening Guide](#).

Configuring TLS with Active Directory

To ensure that TLS is configured with Active Directory Follow the instructions in the "Configuring TLS with Active Directory" section of the [Common Criteria Hardening Guide](#).

Note: If you have any existing domains configured in Secret Server, you must edit them and enable LDAPS on each one.

Configuring TLS with Syslog

To configure TLS with Syslog, follow the steps in the "Configuring Syslog/CEF External Audit Server" section of the [Common Criteria Hardening Guide](#).

Additional Common Criteria Configurations

Configuring X.509v3 Certificates

See the "Configuring X.509v3 Certificates" section of the [Common Criteria Hardening Guide](#) for instructions on installing and configuring certificates on the Secret Server Web servers.

Enabling DPAPI

The Windows Data Protection API (DPAPI) is a pair of functions that allow access to operating-system-level data protection services to protect the master encryption key file, encryption.config.

To enable DPAPI, follow the instructions in the "Verify DPAPI Setting Is Enabled" section of the [Common Criteria Hardening Guide](#).

Enabling FIPS Mode

To configure your server and Secret Server to use the Federal Information Processing Standard (FIPS), follow the instructions in the "Verify FIPS Mode Is Enabled" section of the [Common Criteria Hardening Guide](#).

Note: Also see: [Enabling FIPS Compliance](#)

Ensuring Zero Information Disclosure

To comply with Common Criteria requirements, you must configure Secret Server to not display any unnecessary information. This applies to unhandled errors as well as the application version number.

Configuring Custom Error Messages

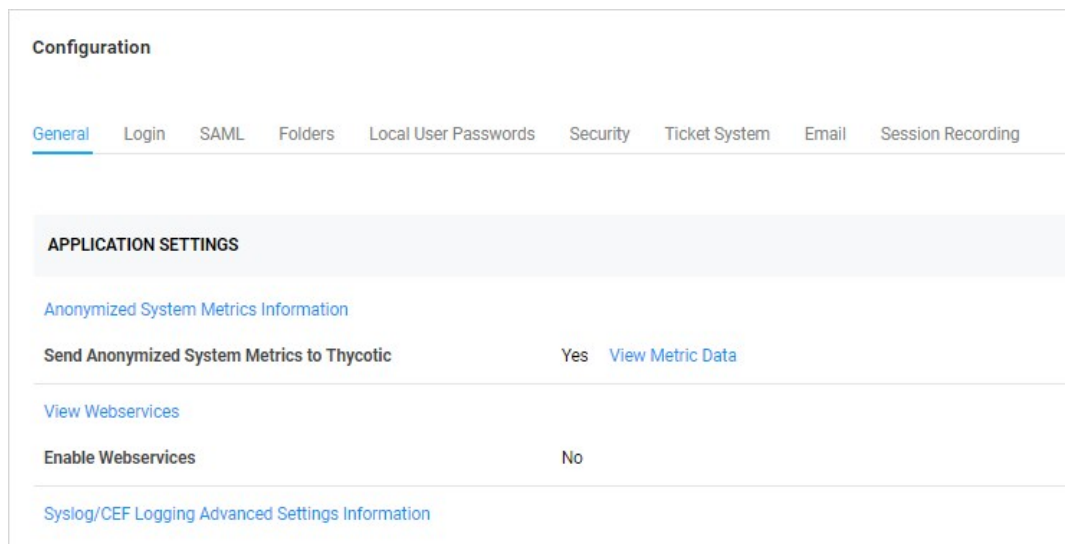
To hide detailed error messages and display a custom message when an unhandled error occurs:

1. Open `https://<your_secret_server_url>/ConfigurationAdvanced.aspx` in your browser.
2. Scroll to the bottom of the page and click the **Edit** button.
3. Type the message you want displayed to users in the **Zero Information Disclosure Message** text box.
4. Click the **Save** button.

Hiding the Application Version Number

To hide the application version number in the application header and footer:

1. Go to **Admin > Configuration > Security**.
2. Click the **Edit** button. The Configuration page appears:



3. Click the **Security** tab.
4. In the **Web Services** section of the page, click to select the **Hide Secret Server Version Numbers** check box.
5. Click the **Save** button.

Note: For diagnostic purposes, the application version number is still displayed on the Diagnostics page. Make sure that permissions to this page is limited to employees that may need to access this page when contacting Delinea technical support.

Configuring the Login Banner

For Common Criteria compliance, when a user first logs in, the login banner must reveal the user policy agreement and force that user to agree to the policy before logging into Secret Server. To configure the Login Banner according to Common Criteria guidelines, follow the instructions in the "Configuring the Login Banner" section of the [Common Criteria Hardening Guide](#).

Configuring Account Lockout

To access Secret Server, users must login with local or domain credentials. To comply with Common Criteria, Secret Server must use "account lockouts" to prevent repeated unsuccessful login attempts. Configurable by an Secret Server admin, an account becomes inaccessible after a defined number of unsuccessful authentication attempts until an admin unlocks the user's account.

To configure settings for account lockouts:

1. Navigate to **Admin > Configuration**.
2. Click the **Login** tab.
3. Click the **Edit** button.
4. Adjust the number in the **Maximum Login Failures** text box. The default is five attempts.

To Unlock a user's account:

1. Navigate to **Admin > Users > Select the User**.
2. Click the **Edit** button.
3. Click to deselect the **Locked Out** check box.
4. Click the **Save** button.

Disabling "Remember Me" Logins

A browser's "remember me" login function stores the user's login name and password so the user does not need to enter it again on that browser, which is both convenient and insecure. To disable "Allow Remember Me" during logins, follow the instructions in the "How to Disable Allow Remember Me during Logins" section of the [Common Criteria Hardening Guide](#).

Configuring SQL Server

To meet Common Criteria requirements, Microsoft SQL Server must be installed on the local machine—the same as the Secret Server Web server. During the install process for MS SQL, ensure that you use Windows authentication mode.

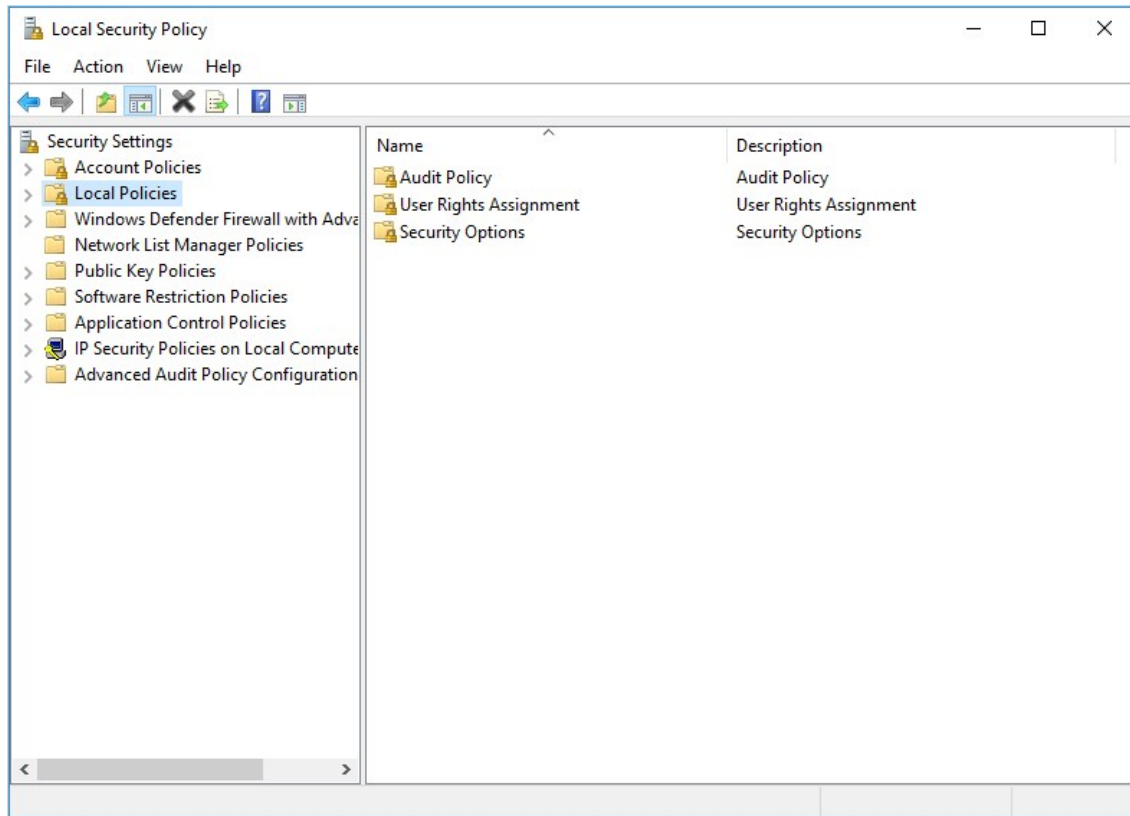
If you have an installed instance of Secret Server that does not meet this requirement, you can migrate the remote database to the server hosting Secret Server. If MS SQL Server is not installed and configured on the Secret Server, you must install it. The server must be configured with enough RAM, storage space, and processors to support running MS SQL Server and the Web site simultaneously. After copying the database, you can go to **Admin > Database** to point Secret Server to the new database location.

Note: Because the database must be installed locally with the Secret Server Web application for Common Criteria compliance, Secret Server is not fully compliant when running multiple nodes.

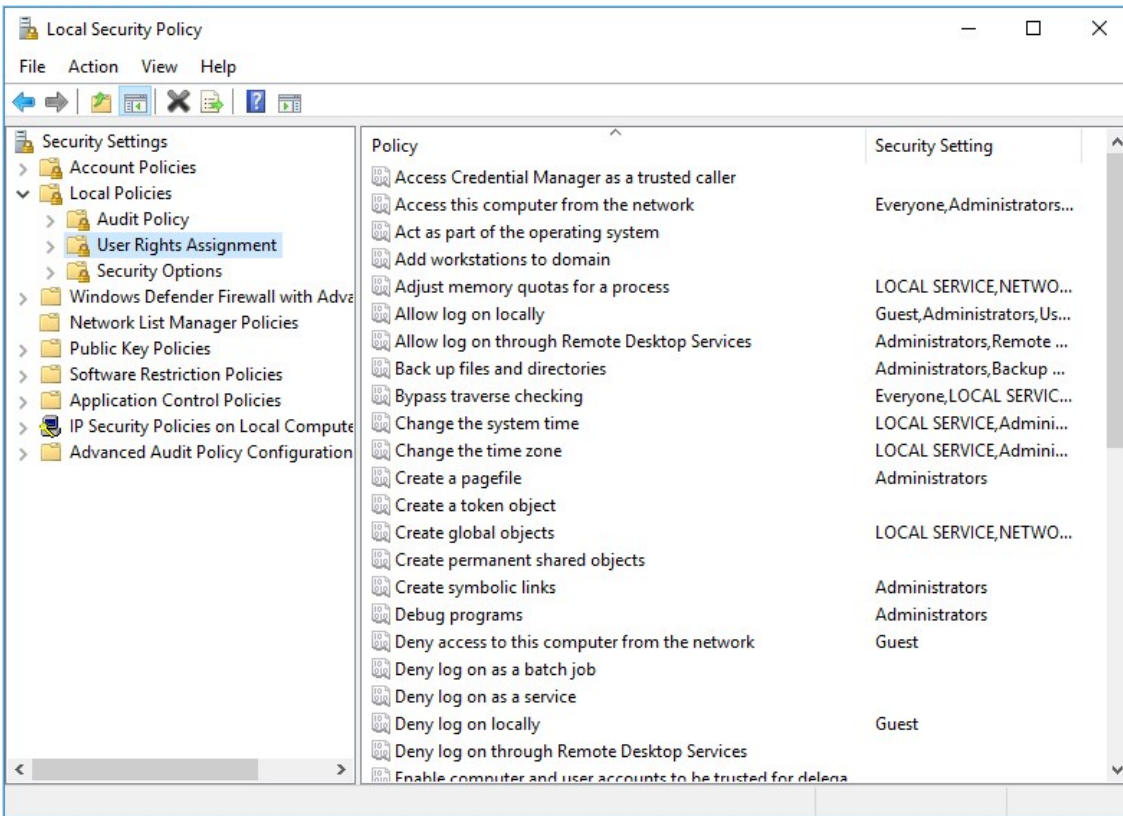
Running the IIS Application Pool with a Service Account

To use Windows authentication to access the SQL database, you should create a service account. To run the Secret Server IIS Application Pool with a service account:

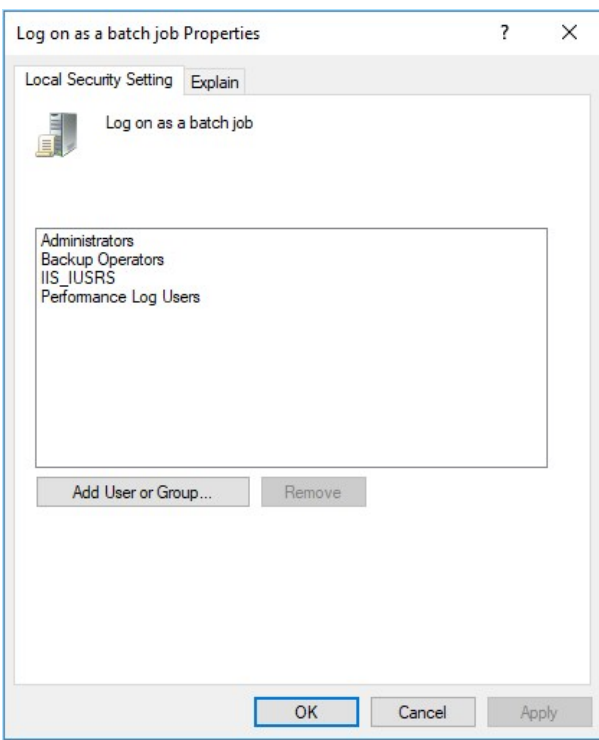
1. Open a command prompt window, change the directory to your .NET framework installation directory (usually C:\Windows\Microsoft.NET\Framework...) using the `cd` command.
2. Type `.\aspnet_regiis -ga <user_name>` and press **<Enter>**. The username is from the MS SQL Server user account.
3. Give your service account "modify" access to C:\Windows\TEMP.
4. Open the Local Security Policy App from your start menu.
5. Grant batch logon permissions to your service account:
 1. Open the Local Security Policy Console (search for and open `secpol.msc`):



1. Expand the **Local Policies** folder (not shown).
2. Click to select the **User Rights Assignment** folder.



3. Right-click **Log on as a batch job** in the right panel and select **Properties**.



4. Click the **Add User or Group** button.
5. Add your service account.
6. Click the **OK** button.

Note: If you use group policy to enforce "Log on as a batch job" and have group-managed service accounts, that will overwrite any local permissions to "Log on as a batch job" on all computers that have the policy applied. Using the local security policy is a safer option if you are not sure about your usage across your domain.

5. Grant "Impersonate a client after authentication" permission to the service account under **User Rights Assignment** the same way "Log on as a batch job" was assigned above.
6. If you now get a "Service Unavailable" error after applying "Log on as a batch job" permissions:
 1. Update your group policy settings (**Start > Run > Cmd** and type `gpupdate /force`) and restart the Windows Process Activation service.

Note: For more information, see [Running the IIS Application Pool As a Service Account] ([setup/installation/running-Secret-Server-iis-app-pool-service-account/index.md](#)).

Assigning Common Criteria Roles and Permissions

See the "Common Criteria Roles and Permissions" section of the [Common Criteria Hardening Guide](#).

Managing User Passwords

See the "Managing User Passwords" section of the [Common Criteria Hardening Guide](#).

Configuring Secret Templates

To enable only the secret templates that are certified Common Criteria compliant and to set Common Criteria-compliant password policies on those templates, see the "Configuring Secret Templates" and "Configuring Password Policy for Secret Templates" sections of the [Common Criteria Hardening Guide](#).

Setting Authentication Strength for Non-Password Credentials

See the "Authentication Strength for Non-Password Credentials" section of the [Common Criteria Hardening Guide](#).

Configuring Remote Password Changing for SSH Key Rotation

See the "Configuring Remote Password Changing for SSH Key Rotation" section of the [Common Criteria Hardening Guide](#).

Configuring External Auditing

Connecting to an External Audit Server

To connect to an external syslog/CEF audit server, see the "Security—Connecting to an External Audit Server" and "Configuring Syslog/CEF External Audit Server" sections of the [Common Criteria Hardening Guide](#).

Configuring Local Windows Event Log Auditing

See the "Configuring Local Windows Event Log Auditing" section of the [Common Criteria Hardening Guide](#).

Introduction

Secret Server integrates with hardware security modules (HSMs). When Secret Server is configured to use an HSM, the Secret Server encryption key is protected by that HSM.

HSMs offer several security features that traditional servers cannot. Depending on the model and design of the HSM, most HSMs are designed to be physically tamper-proof. HSMs may also be independent hardware on a network, which allows physically placing the HSM in a more secure location that might otherwise be too inconvenient for a server.

To provide broad support for HSMs, Secret Server supports any HSM that can be configured with Microsoft's Cryptography Next Generation (CNG) provider. CNG is a layer provided by Windows Server 2008 and later that HSM manufacturers can interface with. If your HSM properly supports CNG and supports the right algorithms, Secret Server can use it.

Note: Turning off HSM (deselecting the check box) in Secret Server may cause a "Server connection unavailable" error. If this happens, a manual reset of the IIS server should take care of it.

Note: CNG provider installation and configuration varies from HSM to HSM; however, documentation is available from each HSM vendor on how to correctly install CNG providers.

HSM Requirements

Each HSM must provide support for these algorithms through CNG:

- **RSA 4096:** Support for RSA with 4096-bit keys is required. The HSM must also support RSA for encryption and decryption, in addition to signing.
- **PKCS#1 v1.5 Padding:** The HSM must support PKCS#1 v1.5 padding for RSA encryption.

Additionally, closely follow the requirements and recommendations of the HSM vendor for things such as minimum latency, redundancy, and operating environment.

Note: Due to limitations of the account, the NETWORK SERVICE account is not supported as an account for the IIS Application Pool. We recommend configuring Secret Server's application pool as a service account. In the advanced settings for the application pool, set "Load User Profile" to true.

Note: Some HSM CNG provider's products interfere with each other. We recommend no more than one HSM CNG provider is configured on a Windows installation at a time.

Silent HSM Operation

Because Secret Server is a Web application with no one physically present at the server at most times, Secret Server interacts with the HSM in "silent" mode. This prevents the HSM from attempting to interact with any users logged onto the server.

Some HSM features require interaction. If the HSM is configured in a way that requires interaction, Secret Server cannot communicate with the HSM and fails during the configuration steps.

For example, Operator Card Sets (OCS) in Thales network HSMs are such a configuration. If the Thales CNG provider is configured to use an OCS for key protection instead of module protection, someone must be physically present at the HSM and the server to insert their operator card when the key is needed. If the OCS quorum is more than a single card, Secret Server cannot interact with the HSM because it requires inserting and removing the OCS cards.

In that case, we recommend that Thales' CNG provider is configured to use module protection instead of an OCS. It is possible to use an OCS with Secret Server if the quorum is exactly one card and the card is left in the HSM at all times.

Consult your HSM vendor and their documentation to ensure that the HSM and their CNG provider are able to operate in silent mode and are configured to do so.

Configuring HSM Integration

To configure the HSM integration, go to the **Admin > Configuration** menu and click **Configuration**, then select the **HSM** tab. This starts the HSM wizard, which guides the process of selecting the HSM's CNG provider.

You can find the list of available CNG Providers by querying for the list of registered CNG providers. Each provider must correctly report that it is a "Hardware" provider, and that it is not a Smart Card reader. If an error occurs while querying the CNG provider for its properties, it will not appear in the list; however the error is reported to Secret Server's system log. If the desired CNG provider does not appear in the list of CNG providers, ensure that the provider is correctly registered and that IIS has been restarted after the CNG registration. Also check that an error is not occurring while querying the HSM by examining the system log.

Once the CNG providers are selected, Secret Server simulates encryption and decryption operations and verifies the results to check that it is functioning properly. Finally, Secret Server verifies the selected providers, and then enables HSM integration. Detailed steps are provided throughout the HSM configuration wizard.

Rotating the HSM Key

If HSM is enabled, Secret Server can now rotate the HSM key to ensure the secret keys are always protected by an HSM key. Rotating the HSM key only decrypts the secret keys and then re-encrypts them with the new HSM key. We recommend performing a secret key rotation after the HSM key has been rotated.

To rotate the HSM key:

1. Navigate to **Admin > Configuration > HSM**.
2. Click the **Rotate HSM Key** button.
3. Make sure to back up the `encryption.config` file before proceeding.
4. Click the **Next** button.
5. Select the **HSM Persistent Provider** and **Key Size**.
6. Click the **Next** button. This performs a test to ensure the HSM can be used.
7. Click the **Next** button.
8. Verify the new HSM configuration.
9. Click the **Save** button.
10. Click the **Finish** button.
11. Do an IISReset or application pool recycle. This starts the rotation.

Securing HSM Integration

The wizard to enable, rotate, and disable HSM integration is protected by the "Administer HSM" role permission in Secret Server. The permission should be carefully assigned—if at all. Additionally, you can create an event subscription that sends alerts when the role permission is assigned or unassigned from a role.

Configuring the HSM also has its own event subscriptions for when the HSM integration is enabled, rotated, or disabled.

Additionally, you can add an application setting to Secret Server to prevent changes to HSM configuration. Enabling, rotating, and

disabling this requires direct access to the file system where Secret Server is installed.

To enable this, edit the `web-appSettings.config` file within Secret Server to contain a key called **LockHsmConfiguration** with a value of **True** as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
  <add key="LockHsmConfiguration" value="True" />
</appSettings>
```

This prevents access to the HSM configuration pages, regardless of role permissions. The only way to gain access is to remove this setting, thus proving you, at a minimum, have access to the server where Secret Server is installed.

HSM Redundancy

HSM redundancy varies from HSM to HSM. Please refer to the vendor's documentation on how to back up the HSM. Backups are typically either made to common file location, another HSM, or onto a smart card with the HSM's built-in smart card reader.

As long as the CNG provider is installed on the server and a key exists on the HSM with the same identifier, Secret Server attempts to use that key.

Testing HSM CNG Configuration

Secret Server does its own testing and verification of the HSM and its CNG provider before the HSM integration can be enabled. To further diagnose any issues with the HSM, the **certutil** command line utility, which is part of Windows, can test the HSM with the **-csptest** option specified. An example output may contain something like this:

```
Provider Name: SafeNet Key Storage Provider
```

```
  Name: SafeNet Key Storage Provider
```

```
.....
```

```
Asymmetric Encryption Algorithms:
```

```
  RSA
```

```
  BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
```

```
  NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION -- 4
```

```
  NCRYPT_SIGNATURE_OPERATION -- 10 (16)
```

```
NCryptCreatePersistedKey(SafeNet Key Storage Provider, RSA)
```

```
  Name: cngtest-6166f8fe-8caf-4e30-8e5c-a-24575
```

```
.....
```

```
Pass
```

Examine the output of the test by looking for your CNG provider's name for your HSM and verifying the result. We recommend running this test using the same account as the application pool Secret Server is using. If the testing tool reports errors, consult your HSM's vendor or documentation for resolution.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Important Support Note: Rotating the master encryption key is a sensitive operation. While we have detailed the steps in this topic, we urge you to contact Support to work with a support representative before attempting any MEK rotation to ensure all pre-requisites have been met. Please provide at least a five-day notice in advance of when you want to conduct the rotation.

Overview

When Secret Server is first installed, a unique random AES256 Master Encryption Key (MEK) is generated and saved in a file, `encryption.config`. The MEK protects anything sensitive in Secret Server that is not associated with a specific secret, as well as each secret's unique AES256 key when an HSM is not used.

For added security, you can rotate the MEK, re-encrypting protected data with the new key.

Note: Geo-Replication, DPAPI, or an active secret key rotation must not be in effect when rotating the MEK.

MEK rotation fully supports using a Hardware Security Module (HSM) for Secret Server On-Premises or key management for Secret Server Cloud. This feature requires Enterprise Plus edition or the advanced encryption license. If you are using an HSM, disable it for your first MEK rotation. You can then re-enabled it, and you can run future MEK rotations without disabling it.

Rotation Procedure

To perform a MEK rotation:

1. **Important:** For Secret Server on Premises, back up the current MEK file `encryption.config` in the application directory **on each node**.

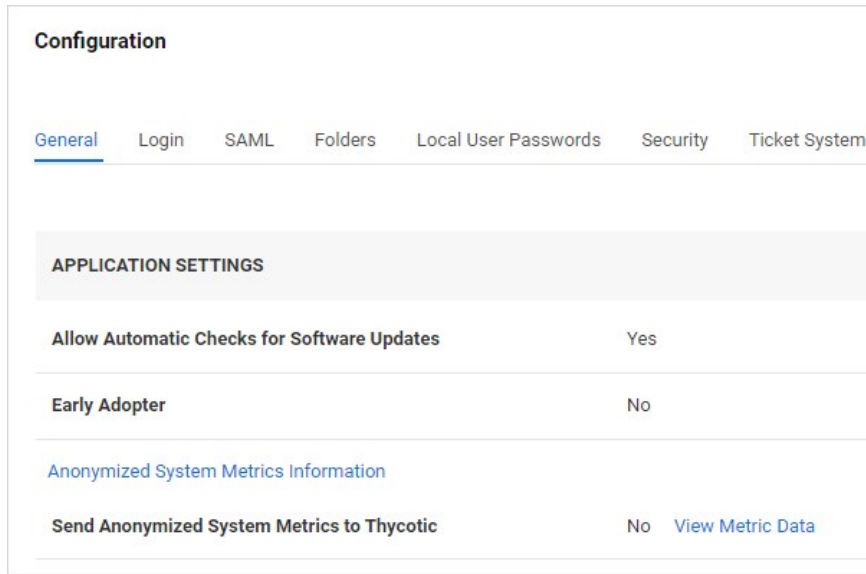
Warning: Do not continue until you back up your encryption key. Failure to do so could cause extensive **permanent data loss**.

Important: For Secret Server on Premises, if this is the first MEK rotation, also backup any encrypted session recording videos saved to disk. They are only updated during the very first MEK rotation. Failure to do so could cause extensive **permanent data loss**.

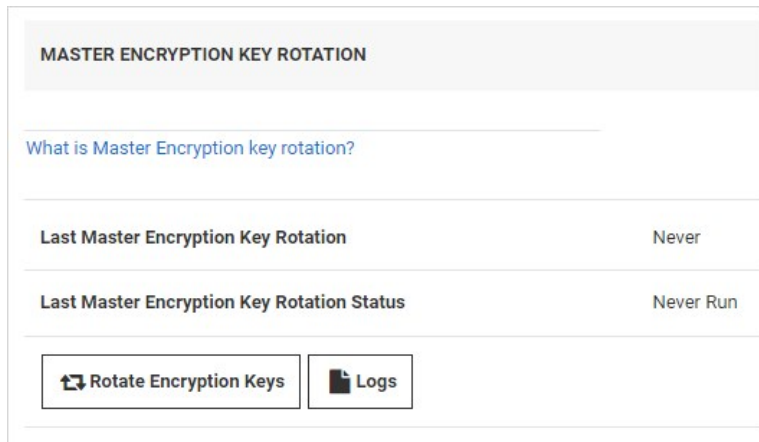
2. **Important:** For Secret Server on Premises, back up your database. See the first task on the [Moving the Microsoft SQL Server Database to Another Machine](#) topic for details.

Warning: Do not continue until you back up your database. Failure to do so could cause extensive **permanent data loss**.

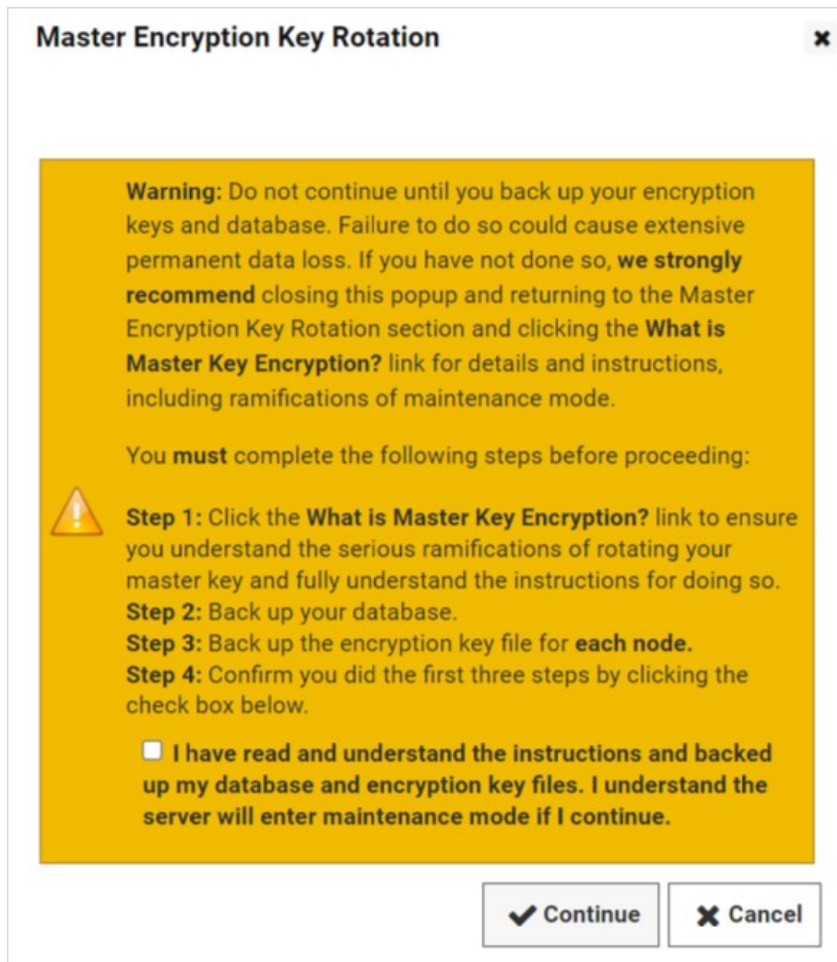
3. Go to **Admin > Configuration > Security**. The Configuration page appears:



4. Click the **Security** tab.
5. Go to the **Master Encryption Key Rotation** section (*not* the Key Rotation section):



6. Click the **Rotate Encryption Keys** button. The Master Encryption Key Rotation popup appears:



7. Carefully read the text, especially for the check box itself.
8. When you are finished, click to select the check box to acknowledge having read it.
9. Click the **Continue** button. The server goes into maintenance mode to perform MEK rotation. Secrets and configuration settings cannot be updated while in maintenance mode. Secrets cannot be updated while in this mode. Processing time will vary, depending on the hardware, number of secrets, and HSM key size.
10. The rotation begins.
11. Secret Server updates the `encryption.config` file on the current server. The file now contains the new MEK, as well as the previous MEK, which is needed to re-encrypt all of the data. The `encryption.config` file is only updated on the node you are connected to when you click the **Rotate Encryption Keys** button, so, depending on whether you have clustered servers, pick one of the following procedures:
 - If you are *not* running a cluster of On-Premises servers:
 1. When prompted to restart the IIS application, run `iisreset` to stop and restart the IIS server.
 2. Open Secret Server to restart it.
 - If you *are* running a cluster of On-Premises servers:
 1. When prompted to restart the IIS server, un `iisreset /stop` on all nodes. This stops but does not restart IIS on the nodes.

2. Copy the `encryption.config` file from the updated node to all other nodes.

Note: If you are not sure which node performed the rotation, check the "modified time" of the `encryption.config` files.

3. Run `iisreset /start` on all nodes. This restarts IIS on the nodes.
4. Open Secret Server to restart it. The first Secret Server node you run continues the rotation process.
12. The new MEK takes effect, and any *new* records encrypted use the new key. Any data still using the old key continues to work until the rotation fully finishes.
13. Secret Server's background worker (`Secret Server-BWSR.log`) enables maintenance mode, preventing changes to secrets and other configuration settings, and continues the rotation process.
14. Secret Server marks the data needing an update in the database and begins rotating everything. A progress bar appears on the **Security** tab.
15. After all data is rotated except for session recording videos, maintenance mode is disabled.
16. Secret Server's session recording role (`Secret Server-SRWSR.log`) starts rotating any existing session recording videos encrypted by the MEK. The **Last Master Encryption Key Rotation Status** text changes to **Pending - Encrypting Session Data**. Depending on the number of videos, this may take some time. This only applies to the first MEK rotation—further rotations do not update the recordings again.
17. Once the process finishes, the **Last Master Encryption Key Rotation Status** text changes to **Completed - Master Encryption Keys Rotated**.

Important: Because we only track the current and previous MEK, the MEK rotation must be fully successful before you can run another one. A Retry button appears if the process times out or needs restarting. Monitor the rotation status and contact support if it fails to complete. Failure to do so could cause extensive **permanent data loss**.

Overview

There are 3 reasons for Delinea products to call home—when:

- Checking for available updates
- Activating licenses
- Reporting anonymized usage metrics

Each of these communications is explained below and can be disabled or avoided.

Checking for and Downloading Updates

Frequency: Once per day

The software checks for available updates and sends the following information to Delinea's update server:

- .NET Framework version
- IP address of the installed instance
- Microsoft SQL Server version
- Microsoft Windows version
- Product version

Checking for updates and sending this information will only occur if both of the following are true:

- The server has outbound network access, which you can block at a firewall.
- The "Allow Automatic Checks for Software Updates" check box is enabled at Admin > Configuration (see below).

No sensitive data is sent during the check. Its only purpose is to alert administrators if a software update is available. The queried website is also used to download new software versions during the upgrade process. If you wish to allowlist the specific servers involved, they are:

- d36zgw9sidnotm.cloudfront.net:443
- updates.thycotic.net:443
- updates.thycotic.net:80
- tmsnuget.thycotic.com/nuget/

License Activation

Frequency: when a new license is activated.

The software also sends contact and license-key information, provided by the administrator, to Delinea during online license activation. The same information is sent via another computer for offline activation.

Reporting Anonymized Usage Metrics

Note: This section only applies to Secret Server and Secret Server Cloud versions 10.6 and above.

Delinea collects anonymized usage data to help guide future research and development plans so that product improvements can provide the greatest benefit to customers.

Frequency: Once per day

Secret Server returns anonymized metrics across several categories:

- A unique identifier number that allows Delinea to correlate metrics from the same server over time but does not contain any information that identifies the customer.
- License information, including edition information and the number of licensed users but not license keys or other identifying data.
- Product configuration and usage, such as number of secrets stored and product feature status, not including any identifying data.
- Product environment, including host operating system and SQL server version, not including any identifying data.

Reporting of anonymized metrics only occurs if:

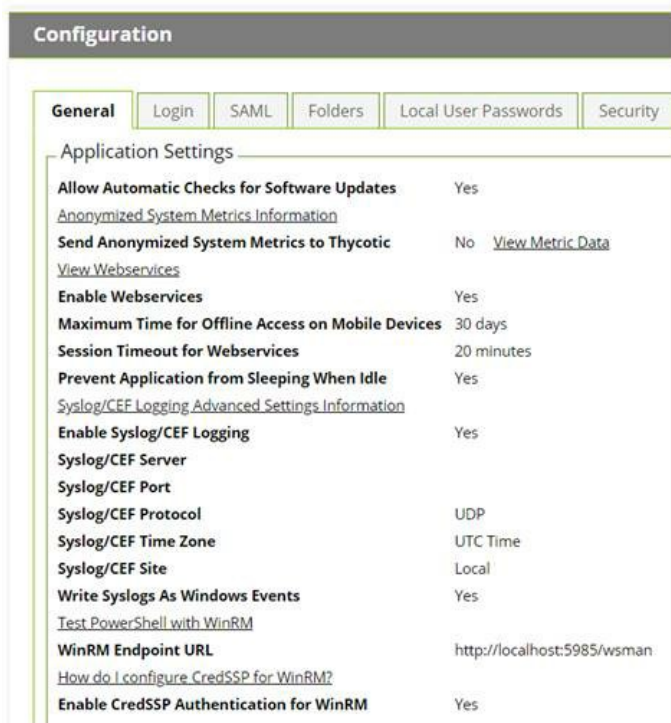
- The server has outbound network access (you can block your server at a firewall if desired)
- The "Send Anonymized System Metrics to Delinea" setting under Admin > Configuration is enabled (see below).

You can allow for the metrics reporting on your firewall by allowlisting: <https://telemetry.thycotic.net:443>.

Setting and Viewing Secret Server Telemetry

To set or view telemetry:

1. Click **Admin > Configuration**. The Configuration page on the General tab appears:



2. (Optional) To view the JSON file for the possible sent metrics, click the **View Metric Data** link. The file appears:

```
{
  "identifier": "cef588896e3e9718d59ccdd0f9e77568",
  "licenses": [],
  "features": [
    {
      "name": "Remote Password Changing",
      "enabled": true,
      "count": 0,
      "countDescription": "Secrets with AutoChange enabled"
    },
    {
      "name": "Heartbeat",
      "enabled": true,
      "count": 614,
      "countDescription": "Secrets with Heartbeat enabled"
    },
    {
      "name": "Checkout",
      "enabled": true,
      "count": 5,
      "countDescription": "Secrets with Checkout enabled"
    },
    {
      "name": "Checkout Change Password",
      "enabled": false,
      "count": 0,
      "countDescription": "Secrets with Checkout change password enabled"
    },
    {
      "name": "DoubleLock",
      "enabled": true,
      "count": 0,
      "countDescription": "Secrets with DoubleLock enabled"
    },
    {
      "name": "Request Access",
      "enabled": true,
      "count": 104,
      "countDescription": "Secrets with Request Access enabled"
    },
    {
      "name": "Request Access Editors need approval",
      "enabled": true,

```

3. Scroll down and click the **Edit** button. The tab changes to edit mode:

General | Login | SAML | Folders | Local User Passwords | Secur

Application Settings

Allow Automatic Checks for Software Updates

[Anonymized System Metrics Information](#)

Send Anonymized System Metrics to Thycotic [View Metric Data](#)

[View Webservices](#)

Enable Webservices

[Maximum Time Offline Explanation](#)

Maximum Time for Offline Access on Mobile Devices Days

Hours

4. Click to select or deselect the **Send Anonymized System Metrics to Delinea** check box.

5. Click the **Save** button.

To secure your ASP session and forms authentication cookies, perform the following steps:

1. Ensure that there is an SSL certificate installed for the instance.
2. Log in to Secret Server using HTTPS.
3. Navigate to the **Admin > Configuration** page
4. Click on the **Security** tab.
5. Click the **Edit** button
6. Check the **Force HTTPS/SSL** check box
7. Click the **Save** button.
8. Open the `web-cookie.config` file in the application installation folder.
9. Set `requireSSL` to `true`.
Save and Close the file.
10. Open the `web-auth.config` file in the application installation folder.
11. Set `requireSSL` to `true` . If the attribute does not exist, add it to the `forms` tag.
Save and Close the file.
12. Recycle the Secret Server's application pool.

Important: If you later migrate Secret Server to a new server, SSL must be configured on the new server before you can log in due to these settings. If you want to log in prior to configuring SSL, reverse steps 8 through 13 and recycle the application pool.

This is a list of items that IIS admin can implement to secure the IIS Web server for additional Secret Server hardening.

Accounts

- Remove unused accounts from the server.
- Disable the Windows Guest account.
- Rename the Administrator account.
- Ensure the Administrator account has a strong password.
- Ensure the IUSR_MACHINE account is disabled if it is not used by the application.
- If your applications require anonymous access, create a custom least-privileged anonymous account. Ensure the anonymous account does not have write access to Web content directories and cannot execute command-line tools.
- Ensure the ASP.NET process account is configured for least privilege. This only applies if you are not using the default ASPNET account, which is a least-privileged account.
- Ensure strong account and password policies are enforced for the server.
- Restrict remote logons—the "Access this computer from the network" user-right is removed from the Everyone group.
- Disable null sessions (anonymous logons).
- Ensure no more than two accounts are in the Administrators group.

Auditing and Logging

- Audit failed logon attempts.
- Relocate and secure IIS log files.
- Configure log files with an appropriate size, depending on the application security requirement.
- Regularly archive and analyze log files.
- Audit access to the Metabase.bin file.
- Configure IIS to use the W3C extended log file format for auditing.

Code Access Security

- Enable code access security on the server.
- Remove all permissions from the local intranet zone.
- Remove all permissions from the Internet zone.

Files and Directories

- Ensure files and directories are contained on NTFS volumes.
- Ensure Web site content is located on a non-system NTFS volume.
- Ensure log files are located on a non-system NTFS volume and not on the same volume where the Web site content resides.
- Ensure the Everyone group is restricted (no access to \Windows\system32 or Web directories).
- Ensure the website root directory has deny write ACE for anonymous Internet accounts.
- Ensure content directories have deny write ACE for anonymous Internet accounts.
- Remove the Remote IIS administration application.
- Remove the Resource Kit tools, utilities, and SDKs.

IIS Metabase

- Use NTFS permissions to restrict access to the metabase (%systemroot%\system32\inetrv\metabase.bin).
- Ensure IIS banner information is restricted (IP address in content location is disabled).

ISAPI Filters

Ensure unnecessary or unused ISAPI filters are removed from the server.

Machine.config

- Ensure protected resources are mapped to HttpForbiddenHandler.
- Remove unused HttpModules.
- Ensure tracing is disabled: `<trace enable="false"/>`.
- Ensure debug compiles are turned off: `<compilation debug="false" explicit="true" defaultLanguage="vb">`

Patches and Updates

- Run Microsoft Baseline Security Analyzer on a regular interval to check for latest operating system and components updates, including Windows, IIS server, and the .NET Framework.
- Test updates on development servers prior to deployment on production servers.
- Check the Microsoft Security Notification Service at docs.microsoft.com on a regular interval for up-to-date Microsoft technical security notifications.

Ports

- Ensure Internet-facing interfaces are restricted to port 80 (and 443 if SSL is used).
- Ensure Intranet traffic is encrypted (for example, with SSL) or restricted.

Protocols

- Disable WebDAV if not used by the application or secure it if it is required.
- Harden the TCP/IP stack.
- Ensure NetBIOS and SMB are disabled if not used (closes ports 137, 138, 139, and 445).

Registry

- Restrict remote registry access.
- Secure SAM (HKLM\System\CurrentControlSet\Control\LSA\NoLMHash).

Script Mappings

- Ensure extensions not used by the application are mapped to 404.dll, including .idq, .htw, .ida, .shtml, .shtm, .stm, .idc, .htr, and .printer.
- Ensure unnecessary ASP.NET file type extensions are mapped to HttpForbiddenHandler in Machine.config.

Server Certificates

- Ensure certificate date ranges are valid.
- Ensure certificates are used for their intended purpose (for example, the server certificate is not used for e-mail).
- Ensure the certificate's public key is valid, all the way to a trusted root authority.
- Ensure the certificate is SHA 256 or better.

Services

- Disable unnecessary Windows services.
- Ensure services are running with least-privileged accounts.
- You can disable FTP, SMTP, and NNTP services if they are not required.
- Ensure the Telnet service is disabled.
- Ensure the ASP.NET state service is disabled and is not used by your applications.

Shares

- Ensure all unnecessary shares are removed (including default administration shares).
- Restrict access to required shares (the Everyone group does not have access).
- Remove administrative shares (C\$ and Admin\$) if they are not required.

Sites and Virtual Directories

- Ensure Web sites are located on a non-system partition.
- Ensure the "Parent paths" setting is disabled.
- Remove potentially dangerous virtual directories, including IISamples, IISAdmin, IISHelp, and Scripts.
- Remove or secure MSADC virtual directory (RDS).
- Ensure include directories do not have the "Read Web" permission.
- Restrict Write and Execute Web permissions for the anonymous account on virtual directories that allow anonymous access.
- Ensure there is script source access only on folders that support content authoring.
- Ensure there is write access only on folders that support content authoring and these folder are configured for authentication (and SSL encryption, if required).
- Remove FrontPage Server Extensions (FPSE) if not used. If they are used, ensure they are updated and access to FPSE is restricted.

Other Considerations

- Ensure server remote administration is secured and configured for encryption, low session time-outs, and account lockouts. Ensure HTTP requests are filtered.
- Use a dedicated machine as a Web server.
- Physically protect the Web server machine in a secure machine room.
- Configure a separate anonymous user account for each application, if you host multiple Web applications.
- Do not install the IIS server on a domain controller.
- Do not connect an IIS Server to the Internet until it is fully hardened.
- Do not allow anyone except the administrator to locally log on to the machine.

Hiding HTTP Header Information

Web applications, such as Secret Server, can leak information useful to attackers via headers, error messages, version numbers, and more. To hide HTTP header information in Secret Server, follow the procedures below.

Hide the IIS Version

To hide the version of IIS used on the server, remove the HTTP header *X-Powered-By* by following the steps below:

1. Open the IIS Manager.
2. In the Connections tree, select the website that Secret Server is running under.
3. Click the **HTTP Response Headers** button on the right. The HTTP Response Headers panel appears.
4. Click to select the **X-Powered-By HTTP** header.
5. Click the **Remove** button in the **Actions** panel.

Hide the ASP.NET Version

To hide the version of ASP.NET used by the Secret Server application pool, remove the HTTP header *X-ASPNET-VERSION* by following the steps below:

1. Open the `web.config` file for Secret Server, which is located in the root directory for the website.
2. Inside the `<system.web>` tag, add the tag `<httpRuntime enableVersionHeader="false"/>`.
3. Save the file.

Hide the Server Type

To hide the server type, remove the line, `Server: Microsoft-HTTPAPI/2.0` (added by the .NET framework) from the HTTP header using the procedure below:

Note: Although there are other methods for hiding the server type, we strongly recommend updating the Windows registry using the procedure below. Do not simply remove the server header variable. Doing so will cause parts of Secret Server to malfunction.

1. Navigate to **Computer > HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > HTTP > Parameters**.
2. Change the **DisableServerHeader (REG_DWORD type)** registry key from 0 to 1.

Introduction

This document outlines security hardening for securing your Secret Server instance, whether it be installed on a single server or in a multi-clustered environment.

Note: Throughout this guide, many references are made to "configuration" settings. Unless otherwise specified, this refers to the settings found by selecting **Configuration** from the **Admin** menu in Secret Server.

Overview

It is critical to secure your Secret Server implementation. That needs to include a layered approach to security (defense in depth), including the operating system, software updates, physical access, protocols, system settings, backups, and personnel procedures. This section of the guide links to other sections and knowledge base articles (KBAs) containing more details.

Best Practices

General

- **Keep Windows up-to-date:** Microsoft regularly releases security patches that resolve vulnerabilities in Windows operating systems.
- **Backup at least daily:** Consider your disaster recovery plan. Review the [Business Continuity and Disaster Recovery Planning](#) KBA for more information.
- **Review system log for errors:** Periodically check the system log (Admin > System Log) for recurring errors. Also do so after any upgrades.
- **Whole-disk encryption:** Use whole disk encryption, such as [BitLocker](#), with a trusted platform module (TPM) to prevent those with physical access from removing disks to gain access to your Secret Server application by circumventing OS and application authentication.
- **Security Hardening Standards:** Consider security hardening standards that apply to either the operating system or applications, such as IIS or Microsoft SQL. Secret Server is compatible with CIS Level 1 and CIS Level 2 hardening and has STIG compatibility.

Note: Attaining full security-hardening standards compatibility is a Delinea priority.

Active Directory

On Active Directory domain controllers, there is a set of unsafe default configurations for LDAP channel binding that allow LDAP clients to communicate with them without ensuring LDAP channel binding and LDAP signing. This can open the controllers to privilege vulnerabilities. See [2020 LDAP channel binding and LDAP signing requirements for Windows](#) for details.

Database

- **Limit access to your Secret Server database:** When you create your Secret Server database, limit access to as few users as possible. We recommend you disable the "sa" account in the SQL instance that contains Secret Server.
- **Limit access to other databases:** When you create a database account for SS, you should ensure it only has access to the Secret Server database.
- **Use Windows Authentication for database access:** Windows authentication is much more secure than SQL authentication. See [Choose an Authentication Mode](#) (TechNet article) for details. To use Windows authentication in SS, you need to create a service account. See the [Using Windows Authentication to access SQL Server](#) for details.
- **Limit access to your database backups:** Database backups are critical for disaster recovery, but they also carry a risk if

someone gains access. The Secret Server database is encrypted, but you should still limit access to ensure maximum security. Limit access to database backups to as few users as possible.

- **Don't share a SQL instance with less secure databases:** Putting the database on a server with less-secure database instances can expose vulnerabilities. For example, an attacker could use SQL injection on another application to access your private Secret Server database. If you intend to put Secret Server on a shared SQL instance, ensure that the other databases are classified internally as sensitive as Secret Server and have similar security controls in place.
- **Review Microsoft's recommendations for SQL security:** See the [Securing SQL Server](#) article in Microsoft's documentation.

Note: Secret Server also supports SQL Server Transparent Data Encryption (TDE) for further protection of the database files. This can have a slight performance impact on the environment and can increase the complexity of the database configuration. Please review this page for more information: [Transparent Data Encryption \(TDE\)](#).

Application Server

- **Use SSL (HTTPS):** We require using Secure Sockets Layer (SSL) encryption to ensure that all communication between the Web browser and Secret Server is secure. We recommend you install a third-party certificate, domain certificate, or self-signed certificate on your website. For information on creating and installing a self-signed certificate, please see the [Installing a Self-Signed SSL/HTTPS Certificate](#) KBA.
- **Force SSL (HTTPS):** Even after you install an SSL certificate, users may still be able to access Secret Server through regular HTTP. To that, enable the "Force HTTPS/SSL" option in Secret Server at Admin > Configuration on the **Security** tab.
- **Limit access to your Secret Server directory.** This contains the Secret Server encryption key, as well as the database connection information (these values are encrypted but remember "defense in depth." Try to grant access to as few users as possible).
- **Limit logon rights to the application server.** Administrators accessing the Application Server directly could attempt to monitor memory in use on the server. Secret Server does several things to protect application memory but the best safeguard is to limit access to the Application Server to as few users as possible.
- **Protect your encryption key.** The encryption key for Secret Server is contained in the encryption.config file, which resides in your Secret Server directory. This file is obfuscated and encrypted, but "defense in depth" would require limiting access to the file. [Using DPAPI to encrypt your encryption.config file](#) is one option. This will use machine-specific encryption to encrypt the file. Make sure you back up the original file before enabling this option. To further protect the file, you can enable EFS encryption. EFS (Encrypting File System) is a Microsoft technology that allows a user or service account to encrypt files with login passwords. For more details, read [Protecting Your Encryption Key Using EFS](#) in this same article. The most secure option is to use a Hardware Security Module (HSM) to protect the Secret Server encryption key. For more information see the [HSM Integration Guide](#).

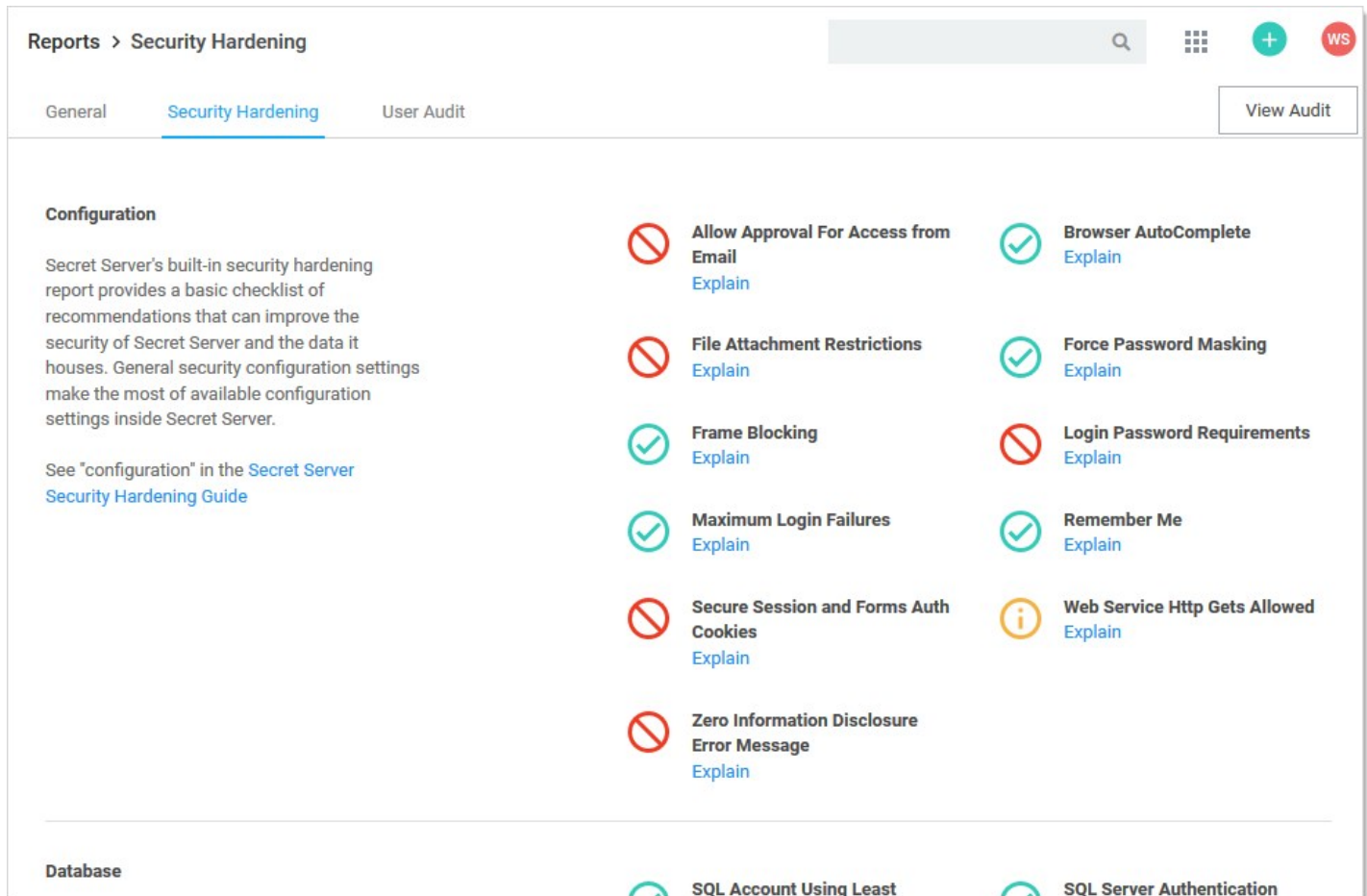
Application Settings

- **Use doublelock for your most sensitive secrets:** DoubleLock is a feature in Secret Server that allows secrets to be protected with additional AES256 encryption keys. Each user gets their own public and private key set when using doublelock. Their private key is protected by an additional password (user-specific, not a shared password) that each user must enter when using doublelock. DoubleLock protects from situations where you accidentally assign someone to the wrong AD group or an attacker gains full access to both your database and Web server - they still will not be able to access doublelocked secrets. For more information, refer to [Using DoubleLock](#).
- **Secure the local admin account:** When you create the first user in SS, it is a privileged admin account that you can use when your domain is down. We recommend that you choose a non-obvious name for this account and protect it with a very strong password. This password should be stored in a physical safe with limited access (there is no need to use this account except in emergencies where other accounts are not working if AD is down or some other reason).
- **Review activity reports:** It is a good practice to regularly review the activity and permissions reports. This can help find anomalies in secret permissions and login failures.
- **Use event subscriptions or SIEM to notify of any security anomalies:** Use event subscriptions to send email alerts on various events in the system, and syslog can send events to a SIEM tool for correlation. For example, this could be used to notify administrators if there are failed login attempts or if certain secrets are viewed.

Security Hardening Report

Secret Server contains a built-in security hardening report to provide a basic checklist of recommendations that can improve the security of Secret Server and the data it houses. The items in this report range from common tasks, such as ensuring SSL is configured, to more advanced options like DPAPI encryption of the encryption key. To find this report, click the **Reports** on the dashboard, and then select the **Security Hardening** tab.

Figure: Security Hardening Report:



An X denotes a failure, and a checkmark denotes a pass. An exclamation point is a warning. Typically, Secret Server could not detect a setting or all aspects of a check were not completed. For example, TLS 1.0 was disabled but TLS 1.1 was not.

You will find the following items in the report:

Note: The individual items below are in alphabetical order, not the order they are in the Hardening Report. The sections are in the same order as the report. This was because the report name does not always match the name of the corresponding label on the configuration UI control. In addition, the controls are not in the same order in the UI as their equivalents in the report.

Configuration Section

Allow Approval for Access from Email

Recommendation: Off

Allow Approval For Access from Email is a convenience option that allows users to approve or deny a secret access request by clicking a link

in the request email sent by Secret Server. Allow Approval From Email does not require a user to authenticate with Secret Server when approving access to a secret. This can be a security concern if the approver's email account becomes compromised, which could allow an attacker to mitigate MFA in some cases to complete an approval. Turn Allow Approval From Email off to get a pass result.

To disable this setting, find the **Permission Options** section of the **Configuration Settings** page and disable **Allow Approval for Access from Email**.

File Attachment Restrictions

Recommendation: On

Note: Labeled **Enable File Restrictions** in the UI.

File attachment restrictions allows administrators to configure what kind of file attachments can be uploaded to secrets. This helps protect users from being tricked into downloading a malicious secret attachment. The file extension and maximum file size can be specified, such as:

*.7z, *.bmp, *.ca-bundle, *.cer, *.config, *.crt, *.csr, *.csv, *.dat, *.doc, *.docx, *.gif, *.gz, *.id-rsa, *.jpeg, *.jpg, *.json, *.key, *.lic, *.p7b, *.pcf, *.pdf, *.pem, *.pfx, *.pkey, *.png, *.ppk, *.pub, *.tar, *.tif, *.tiff, *.tpm, *.txt, *.vdx, *.vsd, *.vsdx, *.xls, *.xlsx, *.xml, *.zip

This security check will fail if the file attachment restrictions is not enabled. This check will return warnings if a potentially dangerous file extension is allowed, maximum file size is not specified, or maximum file size is greater than 30 MB.

Go to **Admin > Configuration > Security tab > File Restrictions section** to change these settings.

Frame Blocking

Recommendation: On

Note: Labeled **Enable Frame Blocking** in the UI.

Do not allow Secret Server to be opened in a <iframe> HTML tag on another, potentially malicious, site. This adds the HTTP header X-Frame-Options: DENY. This deters clickjacking and spreading potential XSS vulnerabilities.

Go to **Admin > Configuration > Frame Blocking** to change this setting.

Force Password Masking

Recommendation: On

Setting: Same

Password masking prevents over the shoulder viewing of your passwords by a casual observer (passwords show as *****). Note the number of asterisks does not relate to the length of the password for added security.

As an administrator, you can force all the secret password fields in the system to be masked when viewed. To do this, enable **Force Password Masking** on the **Configuration Settings** page. Only secret fields marked as a password type field on the secret template will be masked. There is also a user preference setting which will force password masking on all secret password fields viewed by the user.

This **Mask passwords when viewing Secrets** setting is found in the **Tools > Preferences** section for each user.

Note: If the "Force Password Masking" configuration setting discussed above is enabled, this user preference setting will be overridden and cannot be disabled.

Login Password Requirements

Passwords used by local users to log onto Secret Server can be strengthened by requiring a minimum length and using a variety of character

sets. We recommend a minimum password length of eight characters. In addition, all character sets (lowercase, uppercase, numbers, and symbols) are required to get a pass result.

Turn on these login password settings on the **Local User Passwords** tab of the **Configuration Settings** page.

Maximum Login Failures

Recommendation: Reference the lockout policy for your organization. Most often, this setting will mirror the AD GPO lockout policy.

The maximum number of login failures is the number of attempts that can be made to log into Secret Server as a user before that user's account is locked. A user with the administer users role permission will then be required to unlock the user's account. The maximum failures allowed should be set to five or less to get a pass result.

Change the **Maximum Login Failures** setting on the **Login** tab of the **Configuration** settings.

Remember Me

Recommendation: Off

Note: Labeled **Allow Remember Me** in the UI.

"Remember Me" is a convenience option that allows users to remain logged onto Secret Server for up to a specific period. This setting can be a security concern because it does not require re-entry of credentials to gain access to Secret Server.

Disable **Allow Remember Me** on the **Login** tab of the **Configuration** page to get a pass result. It must be set to be valid for 1 day or less to not get a fail result.

Note: Closing a browser completely (all tabs) will log the user out of Secret Server, regardless of this setting.

Secure Session and Forms Auth Cookies

Note: Secure Session and Forms *Authentication* Cookies.

Recommendation: See KBA

Cookies contain potentially sensitive information that can allow users to log onto application. By default, cookies are not marked with the secure attribute. That is, **they are transmitted unencrypted when a user accesses Secret Server through HTTP instead of HTTPS.**

For more information about how to secure your cookies, see [Secure ASP Session and Forms Authentication Cookies](#).

Markdig.Syntax.Inlines.EmphasisInline

Note: Labeled **Allow HTTP Get** in the UI.

Recommendation: Off

Web service HTTP get requests are allowed. Allowing HTTP GET requests allows REST-style calls to many Secret Server Web service methods. This can be a security concern because simply clicking a link to the Web service, created by a malicious user, would cause it to be executed.

Disable **Allow HTTP Get** under the **Security** tab of the **Configuration** settings to pass.

Zero Information Disclosure Error Message

Recommendation: On

Replace all error messages with a custom "contact your admin" message. Error messages can be very helpful when diagnosing installation and configuration issues. However, having errors displayed to a potential attacker can provide him or her with the critical information they need to perform a successful attack.

To hide error messages from the end user, add the `ZeroInformationDisclosureMessage` application setting to the `web-appSettings.config` file. This file is located in directory containing the Secret Server application files. Add the key below to this file in between the `<appSettings>` tags. The contents of that tag is displayed as a message that appears to the user whenever an error occurs in the system. For example:

```
<add key="ZeroInformationDisclosureMessage" value="An error occurred in the application. Please contact your administrator." />
```

Note: This setting is enabled by default in Secret Server 10.7.26+.

Database Section

SQL Account Using Least Permissions

Use the fewest Secret Server permissions as possible in the SQL Account used to access the database. We recommend using a least permission approach where the account only has `dbOwner`. See [Installing and Configuring SQL Server](#).

SQL Server Authentication Password Strength and Username

Note: This section addresses two separate but closely related settings: "SQL Server Authentication Password Strength" and "SQL Server Authentication Username."

Recommendation: Change settings as needed

SQL Server authentication requires a username and a strong password. Strong passwords are eight characters or longer and contain lowercase and uppercase letters, numbers, and symbols. In addition, the SQL Server authentication username should not be obvious.

You can change the credentials of a local SQL account through SQL Server Management Studio, where the Secret Server database is located. The SQL Server authentication credentials used by the application can then be changed by going to the installer `installer.aspx` page and changing them on step three. Using Windows authentication to authenticate to SQL Server is allowed.

For details about creating or modifying a SQL account for Secret Server, see the [Installation Guide](#).

Windows Authentication to Database

Recommendation: Configure

Note: If the SQL instance is *solely* using Windows authentication, this check will pass. If using mixed mode, it will fail—even you are using both Windows authentication plus SQL authentication.

Windows authentication takes advantage of Windows security to provide secure authentication to SQL Server. The SQL Server authentication options can be changed by going to the installer (`installer.aspx`) and changing them.

Note: See the [Installation Guide](#) for instructions on configuring Windows authentication to SQL Server.

Environment Section

Application Pool Identity

Recommendation: Check configuration

The Application Pool identity appears to be a member of the administrators group on the system. This puts the system at risk by giving more access than necessary.

Check the identity of the application pool used by Secret Server in IIS. The Application Pool should be configured to use a service account and not be given unrestricted access to the server or domain.

DPAPI or HSM Encryption of Encryption Key

Recommendation: On

Encrypt your Secret Server encryption key, and limit decryption to that same server. Data Protection API (DPAPI) is an encryption library that is built into Windows operating systems. It allows encryption of data and configuration files based on the machine key. Enabling DPAPI Encryption in Secret Server protects the Secret Server encryption key by using DPAPI, so even getting access to the Secret Server encryption key is not enough to be useful—the machine key is required. If you enable this option, back up your encryption key first, as a DPAPI encrypted file can only be used by the machine it was encrypted on.

To enable DPAPI encryption, go to **Admin > Configuration > Security tab** and click the **Encrypt Key Using DPAPI** button.

Note: This check also passes if Hardware Security Module (HSM) integration is enabled.

SSL Section

Require SMTP SSL

Recommendation: On

Note: Labeled **Use SSL** (on the Email tab) in the UI.

Note: We strongly recommend enabling this setting.

SMTP SSL is required to ensure that all communication between Secret Server and the email server is encrypted. Enable the "Use SSL" option in Secret Server to get a pass result.

Go to **Admin > Configuration > Email tab > Use SSL** to enable the setting.

Require SSL

Note: Labeled **Force HTTPS/SSL** in the UI.

Recommendation: On

Note: We **strongly** recommend using SSL for Secret Server.

Only use SSL (HTTPS) for Secret Server access. Secure Sockets Layer (SSL) is required to ensure that all communication between the Web browser and Secret Server is encrypted. To do so, you need an SSL certificate. You may use an existing wildcard certificate, create your own domain certificate, or purchase a third-party SSL certificate for the Secret Server website. For testing, you can use a self-signed certificate. See [Installing a Self-Signed SSL/HTTPS Certificate](#) (KB) for more information.

Once the SSL certificate is installed, enable **Force HTTPS/SSL** on the **Security** tab of the **Configuration** page to force users to only access Secret Server over HTTPS and to receive a pass in the report.

SSL/TLS Hash

Recommendation: Confirm or remediate

Check the digest algorithm of the certificate. If the algorithm is SHA1, this check returns a warning because SHA1 is being phased out. If the digest algorithm is MD2, MD4, or MD5, the check will fail because they are not secure. SHA256, SHA384, and SHA512 will pass. This check fails if Secret Server cannot be loaded over HTTPS.

Example warning:

"The digest algorithm is sha1RSA, which is considered weak. The algorithm is being phased out and should be replaced with a better algorithm when it comes time to renew the SSL certificate."

Go to the browser's certificate information when logged onto Secret Server. This is usually a button next to the URL text box.

SSL/TLS Key

Recommendation: Confirm or remediate

Check the key size of the HTTPS certificate used. If it is RSA or DSA, the key must be at least 2048-bit to pass. If the signature algorithm of the certificate is ECDSA, the key size must be at least 256-bit to pass. If the algorithm of the certificate is unknown, the result shows "unknown. This check fails if Secret Server cannot be loaded over HTTPS.

Go to the browser's certificate information when logged onto Secret Server. This is usually a button next to the URL text box.

SSL/TLS Protocols

Recommendation: Confirm or remediate

Check for legacy SSL or TLS protocols, which should not be used in a secure environment. If the server accepts SSLv2 or SSLv3 connections, this check will fail. SSLv2 is not considered secure for data transport, and SSLv3 is vulnerable to the POODLE attack. If this server does not support TLSv1.1 or TLSv1.2, this check will give a warning because they are recommended. The SSL certificate used may affect what protocols can be used, even if they are enabled. This check will fail if Secret Server cannot be loaded over HTTPS.

Note: You can check and modify these settings in the Window registry. See [Transport Layer Security \(TLS\) Registry Settings](#) in Microsoft's documentation.

Example warning:

"The server supports the accepts SSLv2 or SSLv3 connections protocol, which are weak. Consider disabling these protocols."

Using HTTP Strict Transport Security

Note: Labeled **Enable HSTS** in the UI.

HTTP Strict Transport Security (HSTS) is an additional security layer for SSL. HSTS allows Secret Server, Password Reset Server, or Group Management Server to inform browsers that it should only be accessible over HTTPS. With this setting enabled, visitors are automatically redirected by their browser to the HTTPS-enabled site.

When the **Force HTTPS/SSL** option is enabled on the **Security** tab of the **Configuration** page, the **Enable HSTS** check box will be displayed. After the option is turned on, you can click **Advanced** to specify the maximum age in seconds for how long the policy should be in effect before re-evaluating. The default value is 25200 seconds (7 hours). We recommend setting this as high as possible, up to a year, if the site, should never be accessed without TLS or SSL.

For details about this, see [Securing with HTTP Strict Transport Security \(HSTS\)](#) (KB).

Security Settings Not in the Hardening Report

Apply TLS Certificate Chain Policy and Error Auditing

Recommendation: Confirm or remediate

Add audits for TLS certificate validation. Auditing will apply to all Active Directory domains using LDAPS and Syslog using TLS. Certificate

policy options, including ignoring certificate revocation failures, apply to syslog using TLS only. The default is the most strict so the certificate chain policy may need to be updated. TLS errors will be logged to Security Audit Log found on the Administration page.

Disable the **Admin > Configuration > Security tab > Apply TLS Certificate Chain Policy and Error Auditing** setting.

TLS errors are logged to the **Security Audit** log at **Admin > See All > Security Audit Log**.

Enable FIPS Compliance

Recommendation: Off

Only allow FIPS-compliant encryption schemes. FIPS (Federal Information Processing Standards) is a set of standards for government entities. It covers many things, including encryption. For businesses, FIPS can be counter productive because it restricts them from using newer or improved existing encryption methods. In addition to enabling this setting, several other tasks are required to meet this standard, including enabling it for Windows itself. For more information, see [Enabling FIPS Compliance](#) for details.

Go to **Admin > Configuration > Security tab > FIPS Compliance** to change this setting.

Key Rotation

Recommendation: Review KBA

Note: Key rotation is not a setting but an activity. It is included here for completeness (the entire Configuration Security tab)

Secret key rotation changes out the encryption key for secret data and re-encrypts that data with a new key. This helps you to meet compliance requirements mandating that encryption keys are changed on a regular basis. See [Secret Key Rotation](#).

Two-Factor Authentication

Users must authenticate to Secret Server at least once using either local Secret Server credentials or their Active Directory credentials. In addition, you can protect Secret Server by enabling two-factor authentication (2FA). 2FA is an additional security layer, such as a text message PIN code sent to your smart phone. The following options are supported by Secret Server for 2FA:

SAML

Secret Server supports the Security Assertions Markup Language (SAML), which provides a more centralized method of adding 2FA to the Secret Server log on. Please see the [Configuring SAML Single Sign-On](#).

Email

Using email for 2FA means that after authenticating with their password, the user receives an email containing a one-time PIN code to enter. For this to work, an SMTP server must be configured in Secret Server and each user must have a valid email address associated with their account. For Active Directory users, the email address will be synced automatically from their domain account.

Check user email addresses at **Admin > Users**.

Soft Tokens

Soft tokens using the Time-based One-time Password (TOTP) algorithm, such as Google Authenticator and Microsoft Authenticator, are supported by Secret Server 2FA. Users are prompted to enter a token displayed on their mobile device each time they log onto Secret Server. The time-based token changes on a regular interval (such as 30 seconds).

RADIUS

One option is to use a Remote Authentication Dial-In User Service (RADIUS)-compliant device, such as an RSA or CryptoCard token, as the

second form of authentication. The user is prompted to enter his or her RADIUS password after initial authentication is done with their Secret Server or AD password.

To set this up, you first need to configure Secret Server to integrate with your RADIUS server, and then you can enable it for individual users or for by domain.

See [Enabling RADIUS Two-Factor Authentication](#) for details.

Duo Security

Using this method requires that you have an active account for Duo Security. Duo Security provides several options for 2FA. The API hostname, integration key, and secret key values are required for Secret Server to authenticate Duo users.

See [Configuring DUO for Two Factor](#) for details.

Enabling Two-Factor Authentication

Enabling for Users

To enable two-factor authentication for a user or several users at once, select **Users** from the **Admin** menu and then select the users in the grid. Use the bulk operation drop-down menu to choose the type of authentication to enable.

Note: If prerequisite settings are not configured, the 2FA option may be disabled or will not appear as an option. See the descriptions above for information about prerequisites for each type of two-factor authentication.

Enabling per Domain

Two-factor authentication can also be enabled per domain if you are syncing users from Active Directory. To do so, select **Active Directory** from the **Admin** menu and then click **Edit Domains**. Click the domain name and then click **Advanced (not required)** to reveal the **Auto-Enable Two Factor for New Users** setting. Select this checkbox and click **Save and Validate**.

Roles

Secret Server uses role-based access control, which allows administrative and user capabilities to be partitioned by role. This can allow for granular control over which areas of the application a user has access to, for example, allowing someone the rights to manage licenses and view reports in Secret Server but nothing else.

Controlling Access to Features Using Roles

Limiting Role Access to the Export Permission

Exporting secrets from your Secret Server as text is very helpful for meeting regulations in certain industries (secrets can then be printed to paper and locked in a physical safe). It can also be used as another disaster recovery option, but access to exporting data from the Secret Server should be tightly controlled. You could create a separate role with just the export permission for anyone needing to export secrets.

Unlimited Administration Mode

Unlimited administration mode allows any role with the "unlimited administrator permission" to see all secrets in the Secret Server. This mode is very helpful for recovering passwords in emergencies or when staff are terminated. You can tightly control access to this feature by splitting out the role permissions for "administer configuration unlimited admin" and "unlimited administrator" into two different roles. This allows you to create the "two-key effect" for access to the mode. See [Using Two Roles for Access to Unlimited Administration Mode\(#Using_Two_Roles_for_Access_to_Unlimited_Administration_Mode\)](#), below, for details.

Limiting Role Access to Secret Templates

Anyone with access to modify your secret templates can change the definitions of the data being stored, and this access should be tightly controlled. Your secret templates are unlikely to need changing once you have defined them, so limiting access to a select number of individuals is typically sufficient.

Monitoring Roles with Event Subscriptions

Another option when protecting roles is to configure event subscriptions to notify appropriate staff in the event that Roles are changed or assigned. Event subscriptions are email alerts that can be sent to users, groups or specific email addresses, based on different events in Secret Server. There are also events available around the "unlimited administrator" role to further protect it from misuse.

Using Two Roles for Access to Unlimited Administration Mode

We recommend determining which role permissions should or should not be combined for users before assigning roles and allowing users access to the application. Part of that is planning access to the "unlimited administration" mode. Users with the "administer configuration unlimited admin" role permission can enable that mode. Once the system is in the mode, users with the "unlimited administrator" role permission can view all secrets in Secret Server and access all configuration settings. So a user with both permissions can enable the "unlimited administration" mode and then view all the secrets or make any configuration change.

To prevent a single person from having that much access, the two role permissions should be given to two different roles and only those roles, and nobody should have access to both of the roles. That enforces accountability and requires the cooperation of two people to enter "unlimited administration" mode.

A solution is to create the two roles, each containing one of the permissions, and then take those two permissions out of the day-to-day administrator role and any other roles besides the two. You can then assign either one of those roles to trusted people with no single person having both roles.

Thus, the access procedure is:

1. User A with the role with the "administer configuration unlimited admin" permission puts the system into "unlimited administration" mode. Not having the correct role, user A cannot make any changes requiring the "unlimited administrator" permission.
2. User B with the role with the "unlimited administrator" permission performs any configuration or accesses secrets only available to that role.
3. When User B is finished, user A takes the system out of "unlimited administration" mode.
4. User B can no longer make any changes requiring the "unlimited administrator" permission because roles with that permission can only be accessed in "unlimited administration" mode. User A cannot make any changes either because User A does not have the role with the "unlimited administrator" permission.

Additional safeguards included:

- Enabling or disabling "unlimited administration" mode is audited, and a comment should be provided each time it is enabled.
- When "unlimited administration" mode is enabled, a banner appears at the top of every Secret Server page notifying users that their secrets can currently be viewed by an unlimited administrator.
- Event subscription notifications should be set up to send an email to a specified user, group of users, or other email address whenever "unlimited administration" mode is enabled or disabled.
- All actions that are normally audited, such as secret views, edits, or permissions changes, are still audited while "unlimited administration" mode is enabled.

Encryption

DPAPI Encryption

Overview

The Data Protection API (DPAPI) is an option that provides an additional layer of security for the Secret Server encryption key. The Secret Server encryption key is contained within a file that is decrypted and used by the application to encrypt or decrypt the sensitive data that is stored in the Secret Server database. Using the DPAPI option in Secret Server ensures that the encryption key file is also encrypted with a key that only Windows knows and is only be usable on same server it was encrypted on. Anybody trying to configure Secret Server on another server using that DPAPI-encrypted key is blocked from doing so.

Important: The encryption key file, `encryption.config`, should be backed up and stored in a secure location before turning on DPAPI encryption. This allows you to restore a backup of the application on another server in a DR scenario. The file is in the Secret Server application directory.

Enabling and Disabling DPAPI

To turn on DPAPI encryption of the file, select **Configuration** from the **Admin** menu. Select the **Security** tab, click **Encrypt Key Using DPAPI**, and then type your password and acknowledge the warning before clicking **Confirm**. To decrypt the key, navigate to the same tab and click **Decrypt Key to not Use DPAPI**.

Using Clustering with DPAPI

You can use DPAPI while clustering is enabled for Secret Server, however there are a few things to take into consideration:

- Backup the encryption key before using this option, otherwise disaster recovery could prove impossible, should the server fail.
- You must initially transfer the un-encrypted key that DPAPI will encrypt to each Secret Server node. Secret Server
- You must enable DPAPI for Secret Server by accessing each server locally (browse to Secret Server while on the server it is installed on, and then enable DPAPI encryption).
- During upgrades, to avoid turning off DPAPI, you can copy all files over to secondary nodes *except* for `database.config` and `encryption.config`.

For more information about clustering Secret Server, see [Setting up Clustering](#) (KBA).

Protecting Your Encryption Key Using EFS

Encrypting File System (EFS) is a Microsoft technology that allows a user to encrypt files with their password. This means that only the user who encrypted the file will be able to access it, even if it is assigned to other users. If an administrator resets the password on this account and the account does not change its own password, then the file is not recoverable.

You can use EFS to protect your Secret Server encryption key. This allows only a single service account to access the file, and no other user can read the key unless they know the service account password. Below are the steps for encrypting your `encryption.config` and `database.config` files with EFS:

1. Backup your `encryption.config` and `database.config` files to a secure location. This is very important for DR recovery purposes.

Important: This step is critical—If you lose access to your service account or the server fails, you will be unable to recover your secrets without these backup files.

2. Create a new service account or select an existing one. The service account should initially have privileges to log on a computer.
3. If you have already installed Secret Server and are using Windows authentication for database access, make sure the service account has access to the database.
4. Run the Secret Server application pool as this service account. See [Running the IIS Application Pool As a Service Account](#) .
5. Give the service account full access to your Secret Server directory through Windows Explorer if it does not have it already.
6. Log on your server as the service account.

7. For both the `encryption.config` and `database.config` files (this instruction uses the former):
 1. Locate the `encryption.config` file in your Secret Server directory (usually `C:\inetpub\wwwroot\SecretServer`).
 2. Right-click the file and select **Properties**.
 3. Click the **General** tab.
 4. Click the **Advanced** button.
 5. Click to select the **Encrypt contents to secure data** check box.
 6. Click the **OK** button.
 7. Click the **Apply** button.
 8. If prompted, select the **Encrypt the file only** option.
 9. Click the **OK** button.
8. Log out of Windows and log back in as an administrator.
9. Confirm that the application still works by performing an IIS Reset (IISReset command at the command prompt) or recycling the application pool.
10. Ensure you you can still log in and view your secrets.

SSL (TLS) and HSTS

We strongly recommend employing SSL (TLS) for Secret Server. Taking SSL a step further, Secret Server also supports HTTP Strict Transport Security (HSTS). HSTS is supported by modern browsers and tells the browser that a site is only accessible by SSL with a valid certificate, period. Even if there is a man-in-the-middle attack with a trusted, but different, SSL certificate, the browser will reject the SSL certificate. Consequently, this setting is very useful for protecting against forged SSL certificates or man-in-the-middle attacks.

For more information about configuring SSL certificates, see the [Installation Guide](#). You can view additional information about HSTS in [Securing with HTTP Strict Transport Security \(HSTS\)](#) (KBA).

SSH Key Validation

Host SSH Key verification is supported for use with heartbeat, proxied launchers, password changers, and discovery. Host SSH key verification can help ensure that the machine you are connecting to is a trusted host. Host SSH key verification will not pass credentials to the target machine unless the public key digest matches the SHA1 digest that Secret Server has on file. This helps prevent man-in-the-middle attacks.

Mapping an SHA1 Digest to Secrets

To configure host SSH key verification:

1. Navigate to **Secret Templates** from the **Admin** menu.
2. And add a field for the host's SSH key digest.
3. Click **Configure Extended Mappings**.
4. Add a **Server SSH Key** mapping to your newly created SSH key digest field.
5. On your secrets, add the SSH key digest of the hosts to your digest field. Verification takes effect the next time you connect to the host.

Validating SHA1 Digests for Unix Account Discovery

To validate SHA1 server digests for Unix account discovery, create a file named `KeyDigests.txt` in the root of the Secret Server website. Each line should contain an IP address or other computer identifier, a comma, and the SHA1 digest, for example:

```
192.168.1.5,7E:24:0D:E7:4F:B1:ED:08:FA:08:D3:80:63:F6:A6:A9:14:62:A8:15 apollo,7A:25:AB:38:3C:DD:32:D1:EA:86:6E:1C:A8:C8:37:8C:A6:48:F9:7B
```

When the file exists and has data, all scanned machines must match one of the SHA1 hashes in the file before scanning. Any computers that do not match will still show up on the "Discovery Network View" page, but authenticated scanning will not take place. That is, no credentials will be passed to the machine, and accounts will not be retrieved from the machine.

Disabling IIS HTTP Headers

Introduction

This section describes plugging some potential, minor but significant, information leaks by the Secret Server Web server. Web applications, such as Secret Server, may unintentionally disclose information about their underlying technologies through headers, error messages, version numbers, or other identifying information. An attacker can use that information to research vulnerabilities in those technologies to attack the application to breach the system.

Procedure

First, **hide the IIS version**. The HTTP header "X-Powered-By" reveals the version of IIS used on the server. To stop this, remove the header:

1. Open the IIS Manager.
2. In the **Connections** tree, select the website that Secret Server is running under.
3. Click the **HTTP Response Headers** button on the right. The HTTP Response Headers panel appears.
4. Click to select the **X-Powered-By** HTTP header.
5. Click the **Remove** button in the **Actions** panel. The header disappears.

Second, **hide the ASP.NET version**. The HTTP header "X-ASPNET-VERSION" reveals the version of ASP.NET being used by the Secret Server application pool. To stop this, remove the header:

1. Open the `web.config` file for Secret Server, which is located in the root directory for the website.
2. Inside the `<system.web>` tag, add the tag `<httpRuntime enableVersionHeader="false"/>`.
3. Save the file.

Third, **hide the server type**. The header line `Server: Microsoft-HTTPAPI/2.0` is added to the header by the .NET framework. To remove that information, you must update the Windows Registry:

Important: Do not simply remove the Server header variable—it will cause parts of Secret Server to malfunction.

1. Open the Windows Registry Editor.
2. Navigate to `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters`.
3. Change the `DisableServerHeader` (REG_DWORD type) registry key from 0 to 1.

Note: There are other ways to hide the server type. We strongly recommend this one.

Adjusting CORS Policy Headers

By default, Secret Server only allows Cross-Origin Resource Sharing (CORS) to unauthenticated resources. CORS allows a restricted resource on a Web page to be requested from a different domain from that resource. To adjust this behavior, either hardening (such as blocking all CORS calls) or relaxing it, follow these instructions:

1. In the Secret Server installation folder, open the `web-appsettings.config` file in a text editor.
2. In the **appSettings** section add: `<add key="UseWebConfigCORS" value="true"></add>`
3. Save the file.
4. In the same folder, also open the `web.config` file in the text editor.
5. In the **system.webServer/httpProtocol/customHeaders** section, add: `<add name="Access-Control-Allow-Origin" value="[customer URL here]" /> <add name="Access-Control-Allow-Headers" value="Content-Type" /> <add name="Access-Control-Allow-Methods" value="GET, POST, PUT, DELETE, OPTIONS" />`
6. Save the file.

Additional Resources

- [Secret Server – Security Hardening webinar](#)
- [Secret Server Best Practices](#)
- [Delinea.com](#)

Session Recording

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Note: macOS Catalina requires additional configuration to use basic session recording. See [macOS Catalina Security](#).

Basic session recording is a licensed feature in Secret Server. It relies on the protocol handler configured on client machines through Secret Server's launcher. Using the launcher, Secret Server captures second-by-second screenshots on the client machine during a user's recorded session. These images of the user's screen are compiled into a video that can be downloaded and played back for auditing and security purposes. Activity recorded in the session is based on screen changes only.

Session monitoring allows administrators with the Session Monitoring permission to view all active launched sessions within Secret Server. If session recording is enabled on the secret, an administrator can watch the user's session in real time.

Admins can search through active and ended sessions. To review and search through sessions go to **Admin > Session Monitoring**.

Searching across sessions can search the following data. To select what data is searched across check the options on the search filters on the left-hand side.

Session Playback Search

Search Filters

Search Across ▾

Secret Name

Secret Items

Username

Proxy Session Client Data

RDP Keystroke Data

Date ▾

Last 30 Days

Status ▾

All

Launcher Type ▾

All

Users ▾

Groups ▾

Secrets 🔍

Folder

< All Folders >

← Back

Search

<p>10.0.0.243\winuser2 - Accessed By ssadmin</p> <p>Remote Desktop · 4/11/2017 05:46 PM · 0:10:03</p> <p>win-h0ko2iq58no · ssadmin</p> <p>View Secret</p>	
<p>10.0.0.243\winuser1 - Accessed By user282</p> <p>Remote Desktop · 4/11/2017 05:41 PM · 0:00:59</p> <p>win-h0ko2iq58no · user282</p> <p>View Secret</p>	
<p>10.0.0.243\winuser1 - Accessed By user282</p> <p>Remote Desktop · 4/11/2017 05:35 PM · 0:01:44</p> <p>win-h0ko2iq58no · user282</p> <p>View Secret</p>	
<p>10.0.0.243\winuser1 - Accessed By user282</p> <p>Remote Desktop · 4/11/2017 05:35 PM · 0:00:11</p> <p>win-h0ko2iq58no</p> <p>View Secret</p>	
<p>10.0.0.243\winuser1 - Accessed By user282</p> <p>Remote Desktop · 4/11/2017 05:33 PM · 0:00:00</p> <p>win-h0ko2iq58no</p> <p>View Secret</p>	
<p>10.0.0.243\winuser1 - Accessed By user282</p> <p>Remote Desktop · 4/11/2017 05:19 PM · 0:00:32</p> <p>win-h0ko2iq58no · user282</p>	

Some search filters require additional components to be installed or configured:

- **Proxy Session Client Data:** Search within keystroke data of proxied SSH sessions. Requires that the SSH proxy is enabled and SSH sessions are using it.
- **RDP Keystroke Data:** Requires the RDP Session Monitoring Agent be installed on the target.
- **RDP Application Name:** Requires the additional RDP Session Monitoring Agent be installed on the target.

To view a recording, click the camera icon on the session. The Watch Session Recording page appears:

Watch Session Recording

Session Summary

Session Secret: 10.0.0.243\winuser2	Session User: Andrew Smithson	Session Start: 3/30/2017 11:10 PM
Machine: win-h0ko2iq58no	Launcher Used: Remote Desktop	Session End: 3/30/2017 11:10 PM

Search Session Activity

Activity Type: All

Elapsed	Type	Activity	Jump To
00:00:00	explorer	explorer	🔍
00:00:00	rdpinput	rdpinput	🔍
00:00:00	TSTheme	TSTheme	🔍
00:00:00	rdpclip	rdpclip	🔍
00:00:00	Thycotic.SessionRecorder	Thycotic.SessionRecorder	🔍
00:00:00	taskhostex	taskhostex	🔍
00:00:00	explorer	explorer	🔍
00:00:04	TSTheme	TSTheme	🔍
00:00:04	powershell	powershell	🔍
00:00:04	conhost	conhost	🔍
00:00:04	powershell	powershell	🔍
00:00:06	hello	hello	🔍
00:00:08	echo	echo	🔍

If there is logged session activity, such as keystroke or application data from the RDP agent or SSH proxy then you can search through session activity and jump to points within the video playback. The playback also displays an activity map to show points of high activity, such as screen changes, keystrokes, and processes started and stopped.

Selecting an activity in the grid also shows additional details below such as the full folder path where the application started and the user that performed the operation.

Note: SSH Keystroke data is shown in one-minute segments. In a short session of less than minute, the "jump to" only goes to the beginning of the video.

For active sessions, there are two actions that can be taken:

- **Watch Live:** When session recording is turned on for the secret and admin can view and replay the user's activity.
- **Terminate:** Sends a message to the end user or terminates their session. The end user sees an alert dialog pop up on their machine with the message. Session recording does not need to be enabled for this to work. For ended sessions admins can watch the recorded video and view the SSH log if session recording was turned on for the secret.

Advanced Session Recording (ASR) is a licensed feature of Secret Server that adds capabilities to those offered by basic session recording. You install the Advanced Session Recording Agent (ASRA), which uses the Remote Desktop Protocol, on any client machine where you want more information from the sessions recorded.

Note: ASR is not available to those using our Mac launcher.

Note: Older ASRAs (earlier than 7.7) only work if a distributed engine configuration is enabled with RabbitMQ or MemoryMQ installed.

ASR enhances the launcher sessions, which typically only include screenshots, keystrokes, and process activity. ASR features include:

- **Screen Capture:** The Secret Server launcher records second-by-second screen images compiled into a playback video of the user's session. This is essentially the same as basic session recording.
- **Logged Processes:** The ASRA logs all processes started and stopped during a user's session.
- **Recorded Key Strokes:** The ASRA records all user keystrokes during the session, which can be disabled.

In addition to those, ASR includes these enhanced video playback features:

- **Searchable Video:** You can search video activity to find locations where specific activities, such as specific keystrokes or ran processes.
- **Enhanced Playback:** Sessions recorded using ASR display additional data on playback, such as the current active window, the used processes, and keystrokes in the session.
- **On-demand video processing**
- **Recording all sessions**
- **Inactivity timeout**
- **Maximum session-length protection**

Note: The Windows protocol handler encodes your session in WEBM format in real time and sends the recording to Secret Server. There is now an "Enable On-Demand Video Processing" option in Secret Server which leaves the recordings in WEBM format, which Chrome and Firefox can playback without any further processing, saving server processing time. If an on-demand recording is viewed with Internet Explorer or Edge (which do not support WEBM playback), you can click a "Request Video Processing" button and the video will be converted to H.264/MP4, which they can then play. If "Enable On-Demand Video Processing" is not checked, then all sessions recorded by the Windows protocol handler will be automatically converted to H.264/MP4.

Note: The Mac protocol handler does not yet support this feature, so any recordings created with it are converted to the chosen legacy video codec format. We recommend H.264/MP4. You can set the advanced session recording agent to "Record All Sessions." If someone logs into a server directly without launching from Secret Server, or even logs in at the console, the full session is recorded, including metadata.

Note: See [Installing the Advanced Session-Recording Agent](#).

The Session Recording tab contains the following configuration options:

- **Enable Deleting:** After the "Days Until Deleting" value, Secret Server deletes the videos from disk. Secret Server
- **Enable Moving to Disk:** After the "Days Until Moved to Disk" value, Secret Server can move videos from the database to an archive path on disk.
- **Enable Session Recording:** Enable session recording for launched sessions.
- **Save Videos To:** By default, videos are stored in the database, Secret Server can also store them directly to a network share. This network share must be accessible from all Web servers that Secret Server is installed on. Secret Server
- **Video Code:** Specify the codec to use to create the videos from the launcher screenshots. This codec must be installed on the Web server (or servers if clustering is enabled) that Secret Server is installed on.

Note: The Microsoft Video 1 codec is for testing only and does not support in browser playback. Sessions encoded with Microsoft Video 1 can still be downloaded for review.

For details on the settings in the Login and "Local User Passwords" tab, see [Configuring Users](#).

General

System requirements apply to both physical and virtual machines.

- Delinea does not support these Web servers:
- Any Client OS
- Domain Controllers
- SharePoint Servers
- Small Business Server (SBS)
- Windows Server Essentials
- For best performance, we recommend using dedicated (clean) servers for hosting Delinea products.
- If .NET and IIS features are not already installed on the Web server, the Delinea Installer adds and configures them automatically.

Database

- Database disk storage depends directly on how many recorded videos are stored to disk. For active users, we recommend you **use a 1 TB shared or local drive for archival or storage space**. For light users, we recommend beginning with 300 GB. Monitor your disk space usage closely, and tailor it for best results.
- **Carefully consider how quickly your allotted storage might be exhausted.** Once again, it is highly variable, but you might expect around 15 hours of recording per GB of storage. Using the example of encoding capacity used in the Session Recording section, if you wanted to record one year of usage by your 60 8-hour users, you would need around 11 TBs of storage (given vacations and holidays). Our recommended 1 TB would last nearly a month in that scenario. A session retention policy using the automatic deletion feature is likely your best option.
- If MS SQL Server is not already installed on your database server, the Delinea Installer can setup SQL Express on the Web server; however, **SQL Express is only for trials and sandbox environments**. Though Delinea supports SQL Express, your users will likely experience performance issues due to memory and product limitations. If experiencing performance issues while using SQL Express, we highly recommend upgrading to MS SQL Server prior to contacting Delinea Support.

Note: Please see Microsoft documentation on SQL Express at: <https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017>

Network Bandwidth and Video

- For Secret Server 10.6 ASR requires around 300 Kbps. Older versions of Session Recording require 1-3 Mbps.

Note: Our Mac launcher uses the older bit rate.

- Session recording bandwidth requirements vary widely based on monitor resolution and image complexity--higher resolutions and more complex images (simpler screen images compress better) use more bandwidth. For example, with a 1024x768 screen resolution, the required network bandwidth is typically between 0.1 Mbps and 1 Mbps.
- If your connection cannot support the needed bandwidth, the session data is still transmitted, but it takes longer to process each session.
- If a user tries to cancel the transmission, this activity appears in the audit record for the Session Recording Secret.
- All sessions are recorded at 1080p.

Note: Before Secret Server 10.6, session recordings 1080p or higher were not supported due to a limitation in Microsoft IIS. The session video would be recorded but may have been corrupted.

- Sessions are recorded using the H.264 MPEG-4 codec.

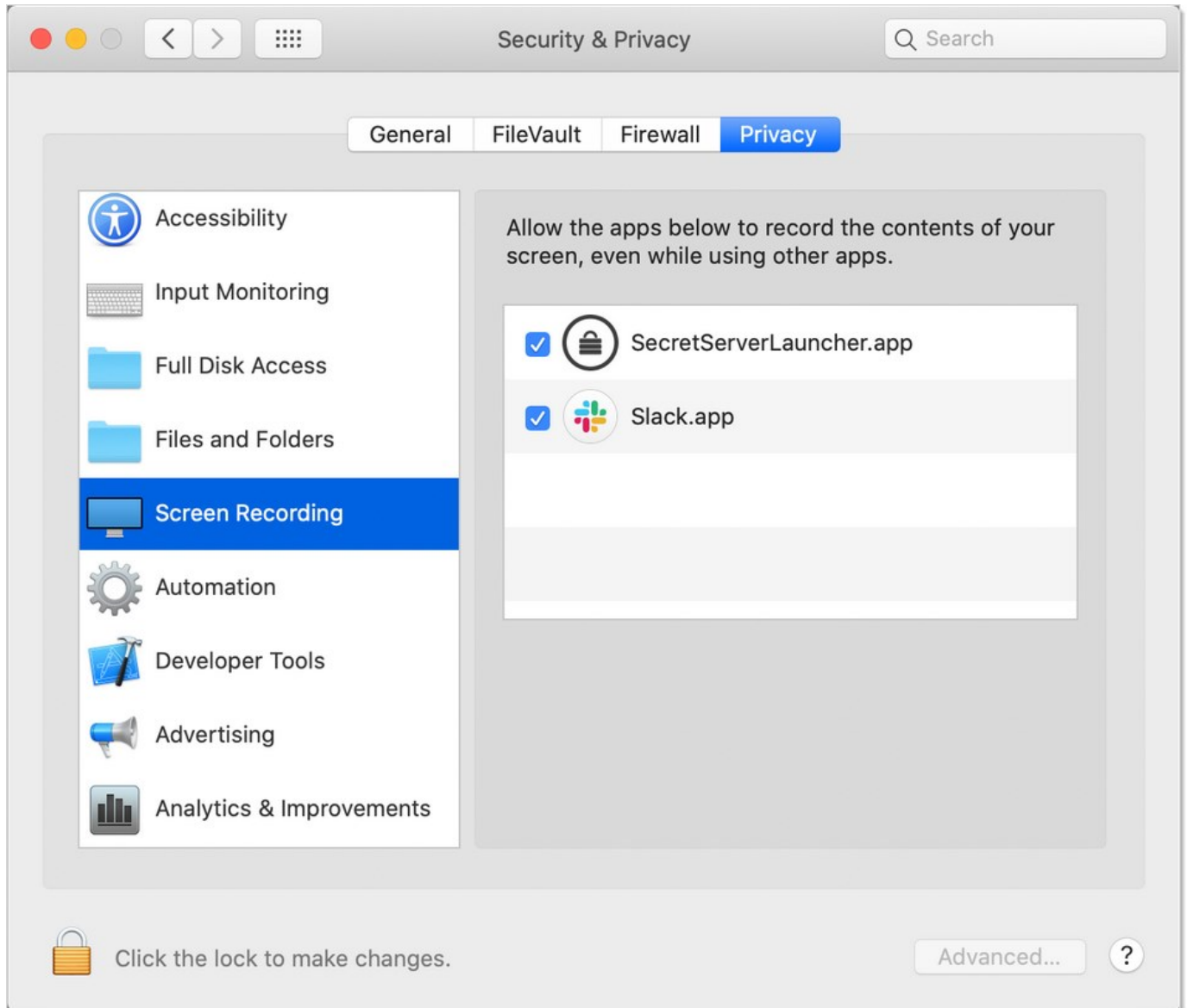
Session Recording

- Server hosting session recording requires fixed RAM and disk space. We strongly recommend that you **do not apply dynamic settings**.
- **Do not record more sessions than you can encode**. If more concurrent sessions are recorded than the system can process, the sessions wait in a queue and are processed when enough server resources become available, which could be in a very long time or perhaps never if your storage is overwhelmed.
- The frame rate we can encode varies dramatically based on many factors, so **testing what encoding rate your session recording configuration can sustain is a must**. From there, you can get an idea of what is possible. For example, let us say you found that we can process 20 FPS on average on your Xeon processors. Given that rate, we could encode around 1 minute of a session recording in 3 seconds, or 1 hour in 3 minutes, or 1 day in 72 minutes--giving you perhaps 480 session hours per day. You could then parse that figure based on your typical usage to arrive at a maximum potential usage, for example, 60 people doing 8-hours of session recording.
- Typically, you can record **up to one hundred sessions at a time per web node**, load balanced, which should handle large use cases.
- CPU usage during video processing varies depending on concurrent users and recording length. We recommend that you **closely monitor CPU percentages on your web server** during video processing, as well on your client machines during recording, to increase CPU count for machines, if needed.
- We recommend that you **set up RabbitMQ as the backbone service bus** in session recording environments. To setup RabbitMQ, see [Installing RabbitMQ](#).

macOS Catalina Security

macOS Catalina enforces security policy around screen recording. To use the session recording feature of the Delinea launcher on MacOS Catalina, you must first:

1. Go to **System Preferences > Security & Privacy > Screen Recording** on your Mac.
2. Allow recording for the SecretServerLauncher.app:



Overview

Session recording allows you to record an RDP or PuTTY session, with optional metadata, and play it back in Secret Server.

The Windows protocol handler encodes your session in WebM format in real time and sends the recording to Secret Server. There is an "Enable On-Demand Video Processing" option in Secret Server which leaves the recordings in WebM format, which Chrome and Firefox can playback without any further processing, saving server processing time. If an on-demand recording is viewed with Internet Explorer or Edge (which do not support WebM playback), you can click the "Request Video Processing" button and the video is converted to H.264/MP4, which they can then play. If "Enable On-Demand Video Processing" is not checked, then all sessions recorded by the Windows protocol handler are automatically converted to H.264/MP4.

Note: The Mac protocol handler does not yet support this feature, so any recordings created with it are converted to the chosen legacy video codec format. We recommend H.264/MP4.

You can set the advanced session recording agent to "Record All Sessions." If someone logs into a server directly without launching from Secret Server, or even logs in at the console, the full session is recorded, including metadata.

Configuration

1. Go to **Admin > Configuration > Session Recording**.
2. On the **Session Recording** tab, click the **Edit** button.
3. Ensure the **Enable Session Recording** check box is selected.

Note: For testing and proof of concept deployments, Secret Server's [Internal Site Connector](#) is sufficient for session recording. For production deployments we strongly recommend [RabbitMQ](#) for a more-robust message queue.

Using Legacy Video Codecs

You can select a legacy video, but it will only apply to sessions recorded by the Mac protocol handler. Delinea recommends the H.264 codec, which was available starting in Secret Server 10.5.000003 because it produces the highest quality videos and requires no additional installation. If you want a different legacy codec, ensure that the codec you select is correctly installed on the same machine as Secret Server. It does not need installation on any client machines, where the session recording is occurring.

Available legacy codecs:

Note: On Windows Server 2008 and above, you can install Window Media Player by adding "Desktop Experience" from the features of Server Manager.

- Microsoft Video 1 (testing only): Microsoft Video 1 is deprecated in favor of Microsoft Video 9 and should not be used for production. Microsoft Video 1 does not support browser-based playback of sessions.
- Microsoft Video 9: High compression level and quality. Requires Windows Media Player. This option produces comparable video sizes to Xvid for moderate activity in an RDP session.
- VP8: High compression level and quality. VP8 is bundled with Secret Server. This option produces comparable sized video to Xvid for moderate activity in an RDP session.
- Xvid: Provides similar quality and compression to DivX and is freely available. This option produces approximately 20 MBs of video for 1 hour of moderate activity in an RDP session. See <https://www.xvid.com/>

Enabling Session Recording on Secrets

You must enable session recording on the Security tab for each secret. Once session recording is enabled, Secret Server records that session when the launcher is used.

To view the recorded session after it is completed, click the **View Audit** button on the secret screen and then the **View Session Recording** link in the **Details** column.

You can also search recordings from the Session Monitoring page under **Admin > Session Monitoring**.

The Session Monitoring page lets users search and filter sessions based on session data, secrets, users, groups, launcher type, date, and folders. This page is also where any recordings appear when using the Record All Sessions option (see below), because such recordings are not tied to a specific secret.

Note: Browser playback is only supported in Secret Server 10.2 and higher. Older versions of Secret Server prompt the user to download the recording.

To view a session, click the camera icon to the right of it. This takes you to the Web playback interface. The video playback shows an activity map to quickly skip to sections of higher usage.

As noted above, if using the "On-Demand Video Processing" option, Chrome and Firefox can play the video. If you try to view an on-demand video using Internet Explorer or Edge, a warning message appears.

If you click the **Request Video Processing** button, the recording is converted from WebM to H.264 as soon as possible, allowing IE/Edge to play it back.

Extending Session Recording with Custom Launchers

You can configure Secret Server with custom launchers to run arbitrary programs, which can then be recorded by session recording. To do so:

1. Define a custom launcher:
 1. Go to **Admin > Secret Templates > Configure Launchers**. The Manage Launcher Types page appears.
 2. Click the **New** button.
 3. Leave the **Launcher Type** dropdown list set to **Process**.
 4. Type a name for the custom launcher in the **Launcher Name** text box.
 5. Type a process name in the Process Name text box.
 6. (optional) Type process arguments in the Process Arguments text box.
 7. Customize other Options as needed.
 8. Click the **Save** button.
2. Associate the launcher with a secret template:
 1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears.
 2. Click the template dropdown list and select the desired template.
 3. Click the **Edit** button.
 4. Click the **Configure Launcher** button. The Secret Template Edit Launcher Configuration page appears.
 5. Click the **Add New Launcher** button.
 6. In the **Launcher Type to use** dropdown list, select your custom launcher.
 7. Customize any other options as needed.

Secret Server 10.8 added two new options to custom launchers:

Record Multiple Windows Option

If this option is not checked, only the main window of the main launcher process will be recorded (this was always the behavior prior to Secret Server 10.8). If it is checked, multiple windows as well as child processes are recorded.

Without this enabled, the main window of the main process sometimes does not show anything useful, depending on the application, resulting in a blank recording. With this enabled, recordings are generally more accurate. This also applies to applications that can open or undock separate windows or those that launch additional processes, such as an application launching PowerShell and then launching other applications from the command prompt.

Record Additional Processes Option

Here you can type an optional comma-separated list of processes to record if found, running under your same user account, that are not started or terminated by the custom launcher. "Record Multiple Windows" must be enabled for this option to be available.

In the example above of launching PowerShell and then opening Notepad, if "Record Multiple Windows" is enabled, both PowerShell and Notepad would be recorded automatically, because the OS can tell that Notepad is a child process of PowerShell. This even works multiple levels deep—for example, launching PowerShell, then the command prompt, and then launching in PowerShell again, finally followed by Notepad.

In some cases, though, you may wish to record an additional process that was already running before the custom launcher was launched or may want to start running one later. To this end, any process names specified in this option are checked for periodically, and recording is attempted on them as well.

Example

If you wanted to run an X11 server such as Xming and then PuTTY with X11 forwarding, you could configure a custom launcher with these values:

```
Process Name: C:\Program Files\PuTTY\putty.exe Process Arguments: -X -ssh $MACHINE -I $USERNAME -pw $PASSWORD Record Additional Processes: Xming.exe
```

In this case, Xming should already be running before the launcher was used and would remain running after the session has ended. It would have no parent/child relationship with PuTTY at all. However, while the launcher session was active, any windows it spawns would still be recorded, allowing the X11-forwarded applications to be recorded, not only the PuTTY window.

Advanced Session Recording

Metadata Recording

By default, session recording creates videos of the launched session. Secret Server supports logging additional metadata, such as keystrokes for RDP and SSH sessions. When these options are enabled, users can search for keystrokes or applications across sessions, and the session playback interface shows additional activity information.

Remote Desktop session metadata requires Secret Server 10.6 and the advanced session recording feature, which in turn requires an installation of an advanced session recording agent (ASRA) on the target servers. See [Installing the Advanced Session Recording Agent](#).

SSH keystroke data relies on the Secret Server SSH Proxy. This can be enabled under **Admin > SSH Proxy**. See [SSH Proxy Configuration](#) for more information. Once proxying is enabled recorded SSH sessions will log SSH traffic which can be searched and is displayed in the session playback interface.

Record All Sessions

As of Secret Server 10.6.26, you can configure the ASRA to record all sessions. This causes it to record video and metadata for anyone logging into the server, even when not using Secret Server, including logging into the console. Since these recordings are not tied to any specific secret, you must go to the **Admin > Session Monitoring** page to view them.

Session Recording Settings

Under **Admin > Configuration > Session Recording** there are several settings for configuring how Secret Server handles session recordings:

Hide Recording Indicator

When viewing a secret, the launcher icon normally indicates if the session will be recorded or not via the recording icon. When launched into, a notification window also informs the user that their session is being recorded. If "Hide Recording Indicator" is checked, users cannot tell which secrets have recording enabled based on the icons, and if they launch a recorded session, they will not be warned that their session is being recorded.

Enable On-Demand Video Processing

The Windows protocol handler encodes the recording on the fly in WebM format and streams the video to Secret Server. Once the session has ended, Secret Server reconstructs the video and leaves it in WebM format, which Chrome and Firefox can natively play back.

Note: WebM is an audiovisual media file format that is a royalty-free alternative to HTML5 audio and video.

Internet Explorer and Edge currently have issues playing back WebM videos, so if you are using those browsers and try to view an on-demand recording, you are presented with a "Request Video Processing" button, which converts the video to H.264/MP4, as soon as possible, which IE or Edge can then play back.

If this option is not checked, all sessions recorded by the Windows protocol handler are converted to H.264/MP4 automatically. If you have many IE or Edge users, Delinea recommends leaving this option unchecked, but this will increase the processing time of videos and increase the load on your Secret Server servers that have the Session Recording role enabled.

This setting has no effect on sessions recorded with the Mac protocol handler, which is always encoded using your legacy video codec choice.

Enable Inactivity Timeout (Minutes)

If enabled, if a session appears idle, users are given a five-minute warning that they will be disconnected. A prompt appears that lets them choose to disconnect immediately or to continue the session. If no response is received, the session is disconnected five minutes later.

Note: This feature was added in Secret Server 10.6.26 and is currently only supported in the Windows protocol handler (not Mac).

Max Session Length (Hours)

This sets a hard limit to how long a recorded session may last. This includes both launched from Secret Server, as well as recorded sessions if using ASRA and the "Record All Sessions" option. This option helps prevent accidental recordings over the weekend, or even longer, if someone forgets to disconnect their session.

Note: This feature was added in Secret Server 10.6.26 and is supported by both the Windows and Mac protocol handlers.

Use Hardware Acceleration

If enabled, when processing H.264/MP4 files, this setting makes Secret Server attempt to use hardware acceleration for video processing if possible (GPU or CPU). Delinea recommends this setting is always enabled because Secret Server will fall back to not using hardware acceleration if necessary.

Note: This feature was added in Secret Server 10.6.0.

Save Videos to

This configuration includes:

- **Database:** Stores the information from a recorded session as encrypted data to your database.
- **Disk:** Stores the recorded session as a video file directly to the specified folder path.

Archive Location Dependent on Site

If you save recordings to disk, enabling this option lets you pick a separate path for each of your sites. This is useful in large environments that need many recordings spread out across multiple devices and locations.

Note: See below for a note about using network shares for storage.

Folder Path

If you save recordings to disk, this is where they are saved. If you use the "Archive Location Dependent on Site" option, this is the default storage location for newly added sites, until you customize their folder path to something else.

Note: See below for a note about using network shares for storage.

Encrypt Archive on Disk

This setting encrypts the session videos when stored on disk. Videos stored on disk are played back through the Secret Server UI but cannot be viewed directly from the file system.

Enable Archiving to Disk

After the specified number of days have passed, all recorded session information in your database is transferred to the specified folder path as video files and cleared from the database.

Enable Deleting

After the specified number of days have passed, all recorded videos in your database will be cleared and video files in your archive path will be deleted.

Setting Notes

- To use "Save Videos to Disk" or "Archive to Disk," the Application Pool service account must have write permission to the specified file path.
- To delete videos from the archive path, the Application Pool service account must have "modify" permissions.
- After saving a change to **Configuration > Session Recording**, the configurations for "Save Videos To Disk" and "Enable Deleting" will immediately be applied to all existing session recordings.

Using Network Share Path

In a clustered environment Secret Server needs to use a network path when saving the files to disk. All nodes need access to the path to read the videos back to the user.

To archive or save to a file path that is a network share, instead of a local folder:

- The SServer IIS application pool must be running as a service account. See [Running the IIS Application Pool as a Service Account](#).
- You must grant access to the network share (using Windows ACLs) to the account running the Secret Server IIS application pool.

Configuring the Maximum Concurrent Recording Sessions per Web Node

To set the maximum number of concurrent recording sessions allowed per web node, follow the procedure below on your Secret Server web server node dedicated to session recording:

1. Navigate to the `web-appSettings_config` file (default location `C:\inetpub\wwwroot\SecretServer\web-appSettings.config`).
2. Right-click the file to open it with Notepad.
3. Before the final `appSettings` line, insert the following string:

```
<add key="PrefetchCount.ConvertVideoMessage" value="7"/>
```

4. At the end of the string, set the value for the maximum number of concurrent sessions you want for this node. In the example above, the maximum number of sessions is set to 7.
5. Save the notepad file.
6. Restart IIS on the server.

Overview

By default, Secret Server session recording creates videos of launched sessions. Secret Server supports logging additional metadata, keystrokes for RDP and SSH sessions, and process activity for Remote Desktop Protocol (RDP) sessions. When these options are enabled, users can search for keystrokes or applications across sessions and the session playback interface displays the additional information. SSH metadata relies on the Secret Server SSH proxy.

As of Secret Server 10.6, recording Remote Desktop session metadata requires the installation of an Advanced Session Recording Agent (ASRA) on the target server. In this scenario Secret Server's protocol handler is still used to launch the session and record the session video, and the ASRA records the metadata only.

As of 10.6.24, ASRA can optionally record any session on the target server. If enabled and the session was not launched from Secret Server, the ASRA will record both video and metadata for the session and upload both to Secret Server once the session is disconnected. This works even if someone logs into the console directly or Remote Desktops into the server without using Secret Server at all. Live viewing of this type of session is not supported.

If you are licensed for session recording, you can install unlimited numbers of ASRAs.

How Advanced Session Recording Agents Work

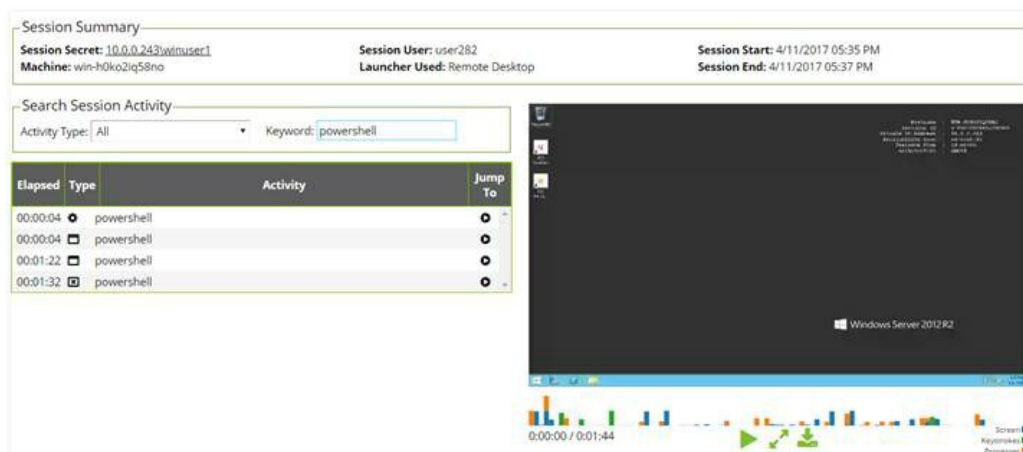
1. Once the ASRA is installed, it contacts the ASRA callback URL to determine if it should record metadata or video any time someone logs on to the computer.
2. A user logs on.
3. The ASRA sends Secret Server the computer's hostname, the username of the user who logged on, any domain name if available, and a list of the computer's IP addresses.

Note: This data is not logged by Secret Server unless you enable DEBUG logging, only for troubleshooting purposes. Secret Server

4. Secret Server checks for any recently-launched protocol handler sessions with matching details, and tells the ASRA if it should record the session.
 - If there is a match, the ASRA starts recording metadata and sends it back to Secret Server over the chosen response bus for the duration of the session.
 - If there is no match and "Record All Sessions" is enabled for the ASRA, the ASRA records both the video and metadata for the session.
 - If there is no match and "Record All Sessions" is not enabled, the ASRA records no video or metadata and waits for the next person to log in.
5. Once a recording session has been closed:
 - If the session was launched from Secret Server using the protocol handler, the video from the protocol handler is matched up to the metadata provided by the ASRA and combined. Secret Server
 - If the session was not launched from Secret Server and "Record All Sessions" is enabled, the ASRA uploads both the video recording and the metadata to Secret Server.
6. On the Session Monitoring page, additional icons are presented based on what extra metadata is present for that session, such as keystroke data for both RDP and SSH, and process data for RDP.
7. Once the session recording has been processed, on the Session Playback page the additional metadata is visible:



And on the Session Monitoring page:



In this example, we searched for activity where the user typed in the word "powershell."

Record All Sessions

As of Secret Server 10.6.24, recording video and metadata for all sessions on a server including console access requires Distributed Engine and ASR to be enabled as described above. ASRA must be installed on the target server, and it must be set to "Record All Sessions." If the server is set to "Only Record Secret Sessions," the ASRA will only provide metadata when people launch into the server from Secret Server. Secret Server When recording all sessions, they appear on the Session Monitoring page. They are not tied to any specific secret because the sessions are not started by Secret Server.

Secret Server Configuration

First, session recording must be enabled (**Admin > Configuration > Session Recording**). As that page warns, Delinea highly recommends using RabbitMQ when using session recording in any production environments. See [Configuring Session Recording](#) for more information.

SSH Metadata

To record SSH keystroke data, enable the SSH proxy (**Admin > SSH Proxy**). Individual secrets then require configuration of the Enable Proxy setting and the Enable Session Recording setting. Then when the SSH session is launched and recorded, keystroke data is recorded, which can be searched and is displayed in the session playback interface. See [SSH Proxy configuration](#) for more information.

Remote Desktop Metadata

To record RDP session metadata, first distributed engine needs to be enabled (**Admin > Distributed Engine**) with an appropriate response bus site connector, which should be a RabbitMQ site connector in production environments. The ASRAs will communicate with the chosen site connector to return any recorded metadata. Next, the Advanced Session Recording feature must be enabled (**Admin > Configuration > Session Recording > Configure Advanced Session Recording**), and an ASRA callback URL entered. HTTPS should always be used in production environments. Individual secrets then just need the Enable Session Recording setting enabled, and the computer you launch into must have the ASRA installed. The secrets do not have to use SSH proxy since the ASRA is what records the metadata.

Session Recording Worker Role

As of Secret Server 10.6, there is a dedicated Session Recording Worker role. If you have a clustered Secret Server environment, you can pick which nodes process recordings on the **Admin > Server Nodes** page. In a large environment with many recordings, you can configure nodes to be dedicated just to session recording, letting other nodes run the Background Worker and other roles. Session recording processes multiple videos at once, which can be controlled with the PrefetchCount.ConvertVideoMessage AppSetting (default: 2). We recommend setting this AppSetting to half the number of CPU cores on the server as a starting point. This setting applies only when using a RabbitMQ site connector, which is another reason Delinea highly recommends using Rabbit if you are using session recording.

Advanced Session Recording Agent

First, create one or more collections to group the ASRAs together, for example, for different domains or environments. Each collection has a unique installer that you can download from their page – the installer is customized to know which collection it is associated with. On the collection, you can specify if you want new agents to **Record All Sessions**, or to **Only Record Secret Sessions**. New agents adhere to this setting, and you can toggle it for individual agents, once they have registered.

Agent Manual Installation

The downloaded installer can be manually installed on a computer by running the setup.exe inside the zip file. It can also be deployed using group policy software installation or other MSI management software. The ASRA installs itself in C:\Program Files\Thycotic Software Ltd\Session Recording Agent and adds a Windows service, Delinea Session Recording Agent.

Note: Only 64-bit Windows operating systems are currently supported. .NET Framework 4.5.1 or greater is also required.

Agent Updates

The ASRA does not automatically update, so new versions must be manually installed or re-deployed using the Group Policy MMC.

Agent Uninstallation

You can deactivate specific ASRAs or an entire collection in Secret Server, and the next time the ASRA reaches out to the ASRA callback URL, it will uninstall itself. Since it only reaches out when someone logs on to the computer, it will remain uninstalled until someone logs on again. The ASRA can also be manually uninstalled directly on the computer like any normal Windows application, but then Secret Server will still show it in the list of active ASRAs under its collection. It should also be deactivated in Secret Server to keep the agent list accurate. If "Record All Sessions" is enabled for this server, you will get a prompt to stop the Delinea Session Recorder application when you attempt the uninstall because it is running and recording your session, as expected. To avoid these issues, we recommend using the Deactivate feature in Secret Server and then logging into the machine, and it will uninstall itself on its own.

Agent Group Policy Installation

Task 1: Review the Prerequisites

The ASRA requires a 64-bit operating system with .NET Framework 4.5.1 or greater installed on the client machine. This is the version that ships with Windows 8.1 and Windows 2012 R2.

Task 2: Download the Advanced Session Recording Agent Installer

1. Log on to Secret Server.
2. Go to **Admin > Configuration > Session Recording > Configure Advanced Session Recording**.
3. Click on an existing collection, or create a new one, as appropriate.

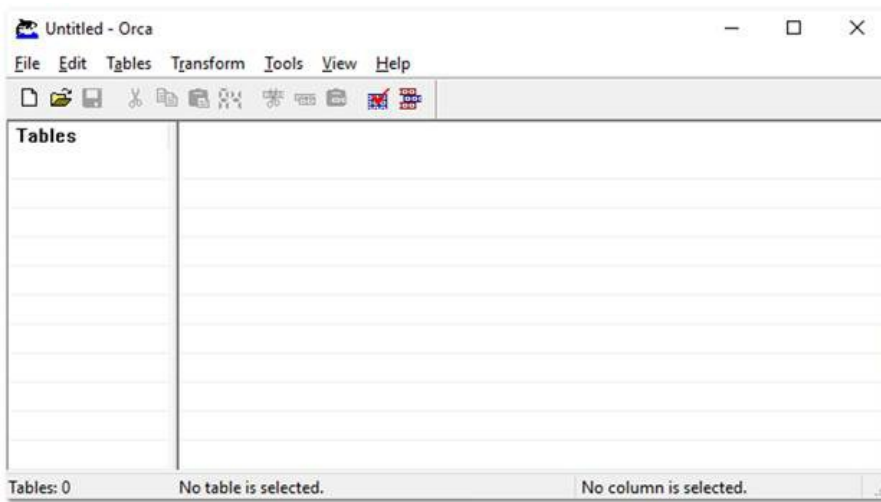
4. Click the **Download Session Recording Installer** (64-bit) button. The installer is downloaded to your computer.

Note: The zip file is customized for each collection. Be sure to download the installer from the collection you want your new ASRAs associated with.

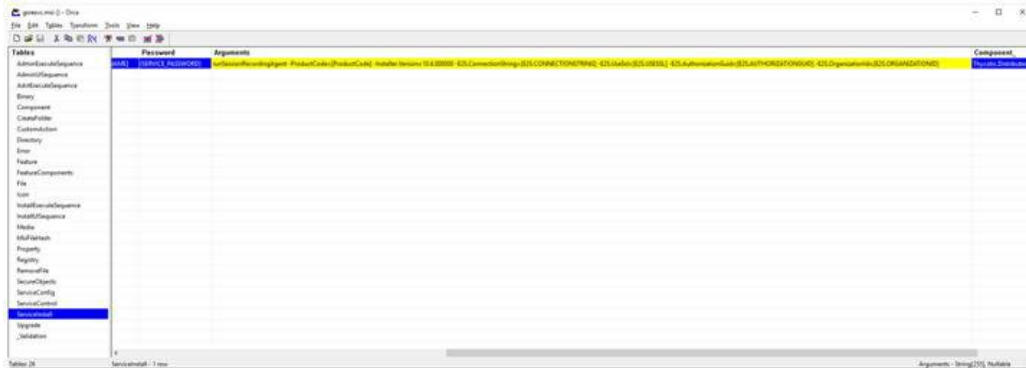
Task 3: Customize the Installer

For a normal manual installation, you extract the zip file, and run setup.exe. There are settings saved in setup.exe.config that customize the installation of the MSI file contained in the zip (gsresvc.msi). When you deploy the ASRA using Group Policy software installation instead of a manual one, the only other files in the zip you need is the MSI, which is deployed from a network share, and an MST (Master Software Tools?) "Transform" file which configures the custom settings.

1. Install Microsoft's free MSI editing tool, Orca, if you do not already have it installed.
2. Extract the ASRA zip file into its own folder.
3. Right click on the MSI file (gsresvc.msi) in the folder where you extracted the zip and select **Properties** to verify that there is a **Digital Signatures** tab indicating that the MSI was signed by Delinea Software.
4. Launch Orca.



5. Open the extracted MSI file (gsresvc.msi). The Tables list appears.
6. Click the **Transform** menu at the top
7. Select **New Transform**.
8. In the **Tables** list, click **ServiceInstall**. Only one row should be listed on the right.



Note: This screen shot shows the Arguments column dragged wider to see its contents. When you initially see it, it will be very narrow, barely showing the contents.

9. Scroll to the **Arguments** column and copy and paste its contents into a text editor. It should look like this example. Be sure to select the entire column. You might need to adjust the column width.

Note: The entire string of text is essentially a CLI command with parameters that begin with a hyphen. For illustration purposes we put each parameter on its own line below.

```
runSessionRecordingAgent
```

```
-ProductCode=[ProductCode]
-Installer.Version=10.6.000000
-E2S.ConnectionString=[E2S.CONNECTIONSTRING]
-E2S.UseSsl=[E2S.USESecret Server]
-E2S.AuthorizationGuid=[E2S.AUTHORIZATIONGUID]
-E2S.OrganizationId=[E2S.ORGANIZATIONID]
```

Everything highlighted is what we will customize. The fields in brackets are what setup.exe would normally customize.

1. In your text editor, open the setup.exe.config XML file from the zip. You will get the Globally Unique Identifier (GUID) from it.
2. In your text editor, replace each of these with the correct values as listed below. The GUID will require looking in setup.exe.config in the XML block. The values are as follows:
 - o **ProductCode** should always be: "" (not in setup.exe.config).
 - o **Installer.Version** should match your Secret Server version (visible in Secret Server in the bottom right corner).
 - o **E2S.ConnectionString** is the callback URL configured on the Advanced Session Recording page.
 - o **E2S.UseSsl** is True or False, based on if you are using HTTP:// or HTTPS:// for the callback URL (Secret ServerL should always be used in production).
 - o **E2S.AuthorizationGuid** is a unique GUID specific to the ASRA collection that you downloaded the installer file from. You can find it in the setup.exe.config file (in setup.exe.config). This is unique for each ASRA Collection.
 - o **E2S.OrganizationId** should always be: 1.

For example:

```
runSessionRecordingAgent¶
-ProductCode={A7FA0ADA-BEED-4841-9D3E-9D700B36F653}¶
-Installer.Version=10.6.000000¶
-E2S.ConnectionString=https://example.com/SecretServer¶
-E2S.UseSSL=True ¶
-E2S.AuthorizationGuid=728ba6b2-1be1-48e5-8d28-¶
984914ca783e¶
-E2S.OrganizationId=1¶
```

1. Back in Orca, delete everything in the ServiceInstall Arguments column.
2. Copy and paste the customized version you just created from your text editor into the Arguments column.
3. Click the **Transform** menu.
4. Click **Generate Transform**.
5. Save the file as `gsresvc.mst` in the column you extracted the installer into. This transform file now contains your customizations for the ServiceInstall Arguments.
6. Close Orca.
7. Check the MSI file's digital signature again to ensure that it was not edited: If you right-click the MSI file and select **Properties** again, the Digital Signatures tab should still show that the MSI is signed by Delinea Software. You created your own custom MST transform file, but the MSI itself should be unchanged. Orca can edit the MSI file itself, but that will invalidate Delinea's digital signature and it is unnecessary.

Task 4: Set up a Network Share

1. Place the `gsresvc.msi` and `gsresvc.mst` files on a network share on your domain controller.
2. Give "Authenticated Users" read access to this share.

Note: Computers in the domain will access this network share to get the installer files before any users log into the machine. It will be the machine account authenticating to the network share, before any users have logged in.

Task 5: Create a Group Policy with Software Installation to install the MSI

1. Open the Group Policy Management Console (**Start\ > Administrative Tools > Group Policy Management**).
2. Expand the **Forest** and **Domain** nodes until you locate the domain on which you are installing the ASRA.
3. Right click on **Group Policy Objects** and click **New**.
4. Enter a descriptive name for your GPO, such as "Delinea Session Recording Agent Installation, and click **OK**.
5. Right click on the newly created **GPO** node and click **Edit**.
6. Select **Computer Configuration > Policies > Software Settings > Software Installation**.
7. Right click on the **Software Installation** node and select **New > Package**.
8. Browse to the MSI on your network share using the share's UNC path, not its folder path. For example: `\\ServerMachineName\Shared` and not `C:\Shared`.
9. Click **Open**.
10. Click to select the **Advanced** option button.

11. Click **OK**. The name is automatically be set to "Delinea Session Recording Agent", since that is the product name in the MSI file.
Note: You can customize the name here, but if you use something else, that is what you will want to check for in the Verify Configuration section, instead of "Delinea Session Recording Agent."
12. On the **Modifications** tab, click **Add** and select your MST transform file. Be sure to again use a UNC path like \\ServerMachineName\Shared, not C:\Shared.
Note: If you wish to have the ASRA uninstalled when it falls out of management, click on the Deployment tab and click to select the box next to "Uninstall this application when it falls out of the scope of management".
13. Click **OK**.
14. In the group policy object editor, expand **Computer Configuration > Administrative Templates > System**.
15. Click the **Logon** node.
16. Right-click **Always wait for the network at computer start-up and logon** and select **Properties**.
17. Click **Enabled**.
18. Click **OK**. This helps reduce the number of reboots required for this policy to take effect as noted in the description of this option.

Task 6: Link your Group Policy Object to an OU

1. Open the Group Policy Management Console (**Start > Administrative Tools > Group Policy Management**)
2. Expand the **Forest** and **Domain** nodes until you locate the domain on which you are installing the Secret Server protocol handler.
3. Right-click the Organizational Unit (OU) for which you want Secret Server protocol handler to be installed and select **Link an Existing GPO**.
4. Select the GPO you created earlier.
5. Click **OK**. The GPO is now linked the entire OU.

Note: To immediately force the group policy change and install the software on a client machine, open a command console on the client machine (start > run > cmd), type `gpupdate /force`, and restart the client machine. You can also wait for the group policy to go into effect, which usually takes one to two hours, but a reboot will still be required due to the mechanics of group policy software installations.

Task 7: Verify Configuration at the Domain Level

1. Go to **Start > Administrative Tools > Active Directory Users and Computers**.
2. Right-click the OU for which Secret Server Protocol Handler is now configured and select **All Tasks > Resultant Set of Policy**.
3. Check to select the box next to **Skip to the final page of this wizard without collecting additional information**.
4. Click **Next** twice.
5. Click **Finish**.
6. In the new **Resultant Set of Policy** window, expand **Software Settings** under **Computer Configuration**.
7. Click to select **Software installation**.
8. **Delinea Session Recording Agent** should be visible in the **Installed Applications** column.

Task 8: Verify the Configuration of a Domain Member

1. From a command prompt, run `gpresult /h report.html` to output a report for just that one computer to the specified HTML file, which you can then view in a browser.

2. The Delinea session recording agent should be visible in the Installed Applications section. Delinea
3. Once the computer has rebooted and completed the installation, the software shows up in Apps and Features (Add Remove Programs).
As usual, the Delinea Session Recording Agent Windows Service is installed in C:\Program Files\Thycotic Software Ltd\Session Recording Agent.

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Advanced Session Recording

Note: This applies to ASRA and Secret Server. See below for additional details.

Table: Advanced Session Recording Requirements

8 CPU Cores	8 CPU Cores	2 CPU Cores
32 GB RAM	32 GB RAM	16 GB RAM (4 GB for the agent itself)
50 GB Disk Space	100+ GB Disk Space	25 GB Disk Space
Windows Server 2012 or newer	Windows Server 2012 or newer	Windows XP (>5.1) or newer MacOS 10.11 (El Capitan) or newer
IIS 7 or newer	SQL Server 2012 or newer	
.NET 4.6.1 or newer		

Basic Session Recording

Note: See below for additional details.

Table: Basic Session Recording Requirements

8 CPU Cores	8 CPU Cores
16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2012 or newer	Windows Server 2012 or newer
IIS 7 or newer	SQL Server 2012 or newer
	.NET 4.6.1 or newer

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

The latest ASRAs use more-reliable durable message exchanges, which are not compatible with earlier (already deployed) ASRAs. Version 7.7+ of the ASRA only requires HTTP connectivity to Secret Server—the distributed engine response-bus site connector is no longer required.

To prevent this from breaking older ASRS, exchanges remain permanently transient. Newer ASRAs use HTTP uploads, which do not use the message queue. Thus, older versions of ASRAs continue to function as they have, and newer ASRA versions do not use the message queue and will have "durable" behavior over HTTP. We recommend updating your ASRA to version 7.7 or later as soon as feasible.

Enabling Inactivity Timeout

If enabled, if a session appears idle, users are given a five-minute warning that they will be disconnected. A prompt appears that lets them choose to disconnect immediately or to continue the session. If no response is received, the session is disconnected five minutes later.

Note: This feature was added in Secret Server SP2 and is currently only supported in the Windows protocol handler (not Mac).

Enabling On-Demand Video Processing

As described above, this feature was added in Secret Server 10.6.24 to greatly improve session recording performance.

The Windows protocol handler now encodes the recording on the fly in WEBM format and streams the video to Secret Server. Once the session has ended, Secret Server reconstructs the video and leaves it in WEBM format, which Chrome and Firefox can natively play back.

Internet Explorer and Edge currently have issues playing back WEBM videos, so if you are using those browsers and try to view an on-demand recording, you are presented with a "Request Video Processing" button, which converts the video to H.264/MP4, as soon as possible, which IE or Edge can then play back.

If this option is not checked, all sessions recorded by the Windows protocol handler are converted to H.264/MP4 automatically. If you have many IE or Edge users, Delinea recommends leaving this option unchecked, but this will increase the processing time of videos and increase the load on your Secret Server servers that have the Session Recording role enabled.

This setting has no effect on sessions recorded with the Mac protocol handler, which is always encoded using your legacy video codec choice.

Record All Sessions

As of Secret Server SP2, you can configure the ASRA to record all sessions. This causes it to record video and metadata for anyone logging into the server, even when not using Secret Server, including logging into the console. Since these recordings are not tied to any specific secret, you must go to the **Admin > Session Monitoring** page to view them.

Recording Metadata

By default, session recording creates videos of the launched session. Secret Server supports logging additional metadata, such as keystrokes for RDP and SSH sessions. When these options are enabled, users can search for keystrokes or applications across sessions, and the session playback interface shows additional activity information. Remote Desktop session metadata requires Secret Server 10.6 and the advanced session recording feature, which in turn requires an installation of an advanced session recording agent (ASRA) on the target servers. See [Installing the Advanced Session-Recording Agent](#). SSH keystroke data relies on the Secret Server SSH Proxy. This can be enabled under Admin > SSH Proxy. See the SSH Proxy configuration KB article for more information. Once proxying is enabled recorded SSH sessions will log SSH traffic which can be searched and is displayed in the session playback interface.

Note: This applies to both ASR and basic session recording. See below for details.

Table: Session Recording Capacities

Dedicated for session recording	4	2 hours	10 minutes
Shared for front-end processing and session recording	2	2 hours	20 minutes

Note: See [Configuring the Maximum Concurrent Recording Sessions per Web Node](#).

Secret Server Setup

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This section contains information about installation and upgrading Secret Server and its components.

- Components Installation
- Protocol Handler
- ASRA
- Launcher plugins
- Distributed Engines
- RabbitMQ
- SDK Client

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Please [review our prerequisites](#) and then select either our [basic \(automatic\)](#) or [advanced \(manual\)](#) installation.

Advanced (Manual) Installation

Procedure

Step 1: Downloading the Secret Server Application Files

Important: Ensure you have the IIS, .NET Framework, and SQL Server prerequisites installed before following the steps below.

Go to the [download page](#) to get a .zip file that contains both Secret Server and Privilege Manager files in the manual installation section. Use this .zip file for the instructions below.

Step 2: Creating Folders and Extracting Contents

1. Extract the contents of the .zip file downloaded above (Right-click, **Extract All...**). The original file is named with the latest version number for Secret Server.
2. Extracting this file reveals a `nugetCache` folder, as well as another zipped folder named `ss_update`. For a Secret Server-only install, you will not need the contents of the `nugetCache` folder.
3. Create a folder called `SecretServer` in the location `C:\inetpub\wwwroot\`.
4. Extract the contents of the `ss_update.zip` file (Right-click, **Extract All...**) to `C:\inetpub\wwwroot\SecretServer`.

Step 3: Configuring IIS

Open Internet Information Services (IIS) Manager* and create a new application pool:

Note: Our IIS installation sets the .NET trust level to "Full (internal)", which may affect other applications on the server.

1. Right-click **Application Pools** and select **Add Application Pool...**
2. Type a name (for example, `SecretServerAppPool`).
3. Ensure that the highest .NET CLR version is selected.
4. Ensure the Managed pipeline mode is set to **Integrated**.
5. Click the **OK** button.

Note: The Secret Server installer sets the application pool to default to the system Network Service account. If you selected Windows Authentication Mode during the SQL Installation process, see [Running the IIS Application Pool As a Service Account](#). To use Windows Authentication you must use an Active Directory service account to run the application pool in IIS. We recommend this as a security best practice.

6. See [Changing IIS to Not Stop Worker Process in IIS 7.0 and Later](#) to set the Idle Timeout and Regular Timeout settings to 0 for the application pool in IIS.
7. Install Secret Server as either a virtual directory (4a) or as a website (4b):

Step 4a: Installing Secret Server as a Virtual Directory

1. Right-click **Default Web Site** and select **Add Virtual Directory...**
2. Select an alias for your Secret Server. The alias is appended to the website, and it is best to name it the name of your earlier unzipped folder. For example, `SecretServer` becomes `https://myserver/SecretServer`.

3. Select the physical directory for where you unzipped Secret Server, for example, C:\inetpub\wwwroot\SecretServer.
4. Click the **OK** button.
5. In the tree, right-click the new virtual directory and select **Convert to Application**.
6. Set the **Application Pool** to the same one you created in the Manual Installation section, for instance, SecretServerAppPool. Secret Server is now ready for installation. Skip to Step 5.

Step 4b: Installing Secret Server as a Website

1. In IIS, right-click **Sites** and select **Add Website...**
2. Type a site name.
3. Click **Select...** and choose the application pool you created in the Manual Installation section.
4. Click the **OK** button.
5. Click the ... button beside the **Physical path** field and select the directory containing the unzipped Secret Server files, for example C:\inetpub\wwwroot\SecretServer.
6. Click the **OK** button.
7. Click the **OK** button at the bottom of the **Add Website** window to save your settings. Secret Server is now ready for installation.

Step 5: Completing Secret Server Installation from the Website

Your Secret Server advanced installation is now ready to complete:

1. [Install your SQL Server](#).
2. Open a browser and navigate to where your Secret Server is located, such as <http://localhost/secretserver>. You should arrive at a page that says "Secret Server (Not Installed or Unable to Access the Database)."
3. Click the **Install Secret Server** button.
4. On the **SQL Server Location** page, specify the server name of your SQL Database Server, <DatabaseMachineName>\InstanceName and then the database name that you created in SQL for Secret Server.
5. If you are using Windows authentication mode to access SQL (recommended), ensure the correct service account is listed.
6. If you selected mixed mode during the SQL install, select **SQL Server Authentication** and enter the SQL username and password you created for the SQL account. For information about adding a SQL Server user, see the [Adding a SQL Server User](#).
7. Click the **Install Secret Server** button. Secret Server verifies it is able to successfully create the Secret Server database. If an error occurs no database changes will be made.

Note: Secret Server attempts to download and install the latest version from the Internet. If you do not have an active Internet connection on your Web server, Secret Server will continue to install the version from your downloaded application files.
8. The install may take a few minutes to complete. Once successful, click the **Return to Home** button.
9. Create a username and password for the administrator account for Secret Server and store these credentials in a safe location.
10. Click the **Create User** button and log on after entering the username and password.

11. Once logged on Secret Server, you are prompted with the Getting Started wizard. The wizard guides you through adding your Licenses, setting up an email server, and creating your first group.

Note: If you skipped the wizard and would like to return, go to **HELP > Getting Started** from the top menu.

Secret Server is now installed. See our [Getting Started Tutorial](#) or contact Delinea Support about training.

Troubleshooting Notes

- If the database name you provide does not yet exist in the specified instance of SQL Server, Secret Server attempts to create the database using the SQL or Windows account you have specified. For that account to create a database, it needs to have the dbcreator server role in SQL Server. Secret Server
- If using Windows authentication mode (recommended) you need to use a service account to run SS's application pools with appropriate permissions. See [Running the IIS Application Pool As a Service Account](#).

Basic (Automatic) Installation

Introduction

This is the installation guide for Windows Server 2016 and Windows 10. For other operating system installation guides, [contact Delinea Support](#).

Secret Server Is an ASP.NET Website

Secret Server is installed as an ASP.NET website. The setup.exe file sets up the website with the correct permissions and creates the settings in IIS.

SQL Server Is Usually Required

Secret Server requires an instance of SQL Server for the database backend and is installed by the setup.exe file, if missing. The SQL Server database will require a SQL account with *db_owner* permission to complete the installation.

Administrative Access

Throughout the installation, you will be required to be an administrator to perform most of these actions. Please ensure that you are logged onto your system with a Windows account that has administrative rights.

Review the Prerequisites

Important: Except for the operating system, the following prerequisites are installed automatically by our installer. If you already have some of them installed or wish to install them yourself, the installer will skip over them.

If this is the first time you are installing Secret Server, please take the time to review the [full list of system requirements and recommendations](#).

System Requirements Overview

- Windows Server 2016 operating system
- Microsoft SQL Server 2008 or greater (any edition)
- Microsoft Internet Information Services (IIS)
- Microsoft .NET Framework 4.6

Note: Windows Server 2016 and Windows 10 come with the .NET Framework 4.6 already installed.

Additional Recommendations

We suggest you:

- Use an SSL certificate for Secret Server.
- Run [Microsoft Update](#) on your server to make sure all components are up to date.

Procedure

Step 1: Downloading the Latest Version of Secret Server

The latest version of Secret Server is available for [download](#). A setup.exe file is downloaded to your machine. We recommend running setup.exe as an administrator.

Step 2: Running the Installer

Welcome Page

The first installer page you are presented is the Welcome Page. The installer should detect whether the machine has Secret Server or Privilege Manager for Windows and will declare which of those products it will install.

Database Page

The Database page allows you to choose to install SQL Express or connect to an existing SQL Server. If you select SQL Express, the installer requires Internet access to download the installation for SQL Server Express.

If Internet access is not available, a link to download SQL Server Express is presented. You are expected to install SQL Server Express and then restart the installer.

If Internet access is available, SQL Server Express is installed.

Pre-Requisites Page

The Pre-Requisites page ensures everything that is required to install Secret Server is setup correctly. Everything on this page *can* be installed outside of the installer. If not, the installer installs and configures them for you. This page is primarily for third party server configuration. If there are issues, please refer to support for the specific non-Delinea vendors.

Database Connection Page

The Database Connection page contains the connection information that Secret Server (and Privilege Manager) uses. You must click the **Test Connection** button and have a successful result before installation can continue.

Create User Page

The Create User page is where you enter the information for the initial Secret Server user.

Email Server Page

Enter connection information for the email server on this page. This is also optional and you can skip it and set it up later in Secret Server. This page will configure email for both Secret Server and Privilege Manager for Windows.

Review Page

Review the, mostly default, settings on the Review page, and change them if needed. Some of the settings are validated before the install can begin.

Install Page

The Install page shows the status from log files as both Secret Server and Privilege Manager are installed.

Step 3: Reviewing the Log Files (Optional)

After the applications are installed, the installer opens a Web browser to the Secret Server log on page. At this point, everything is installed to start using both Secret and Privilege Manager. If the installation failed or you wish you view the logs from the installation, click the **View Log Files** button.

Step 4: Opening Secret Server

If the setup.exe did not automatically open a browser, navigate to where Secret Server is located, for example: <http://localhost/secretserver>.

Step 5: Learning Secret Server

See our [Getting Started Tutorial](#) or contact Delinea Support about training.

Choosing a SQL Server Edition to Use with Secret Server

Choose the Microsoft SQL Server edition to work with Secret Server that best supports the functionality you wish to achieve.

The brief guide below should help you decide which licensing model best suits the needs of your organization.

SQL Server Express Edition

SQL Server Express is a free version of SQL that is sufficient to run most of the functionality within the Secret Server application itself.

However, advanced functionality like mirroring and clustering is not available. SQL Server Express is not recommended for use in production environments due to the database size limitation.

SQL Server Standard Edition

SQL Server Standard provides most of the functionality administrators typically want, including the most common type of mirroring, and clustering up to two cluster nodes.

SQL Server Enterprise Edition

SQL Server Enterprise provides all of the functionality found in the Standard Edition, plus the ability to cluster up to eight nodes and to perform asynchronous mirroring.

For more information on the different editions of SQL Server, see the [Microsoft SQL Server 2016 Licensing Guide](#).

Creating and Using a SQL Server Privileged Account

Overview

This document enables a user to password change SQL accounts using a privileged account. Enabling the takeover of those accounts without knowing their password.

Procedure

Task 1: Creating an Account

1. Open SQL Server Management Studio and connect to your database server.
2. Expand the root level security folder.
3. Right click on the **Logins** folder and select **New Login**.
4. Type the account's login name in the **Login Name** text box.
5. Click to select the **SQL Authentication** selection button.
6. Go to Secret Server.
7. Create a secret using the **SQL Server Account** template.
8. Give it the same username as the login name you just created.
9. For best security, click the **Generate** button on the secret password field and copy that password to the account creation wizard in SQL Server Management Studio.
10. Click **OK** button to save your secret.

Task 2: Assigning Permissions

1. Return to SQL Server Management Studio and connect to your database server.
2. Right click the SQL login and click **Properties**.
3. Select **Securables** in the left column.
4. In the **Permissions** table on the **Explicit** tab, click to select the **Grant** check box for the **Alter any login** row.
5. Click the **OK** button.

Step 3: Using the Account

1. In Secret Server, select the SQL account secret for your new privileged account.
2. Select the **Remote Password Changing** tab.
3. Click the **Edit** button.
4. Click to select **Privileged Account Credentials** on the **Change Password Using** selection button.
5. Click the **No Selected Secret** link. The Select a Secret popup appears.
6. Locate and select the secret you created earlier in the folder tree.

7. Click the **Save** button. The popup disappears.
8. Click the **Change Password Remotely** button.
9. Provide or generate a new password.
10. Click the **Change** button. You have now successfully changed a SQL account password using a privileged account.

Note: You can also assign an account for multiple secrets by creating a secret policy and applying that policy to a folder.

Enabling SQL Server Encryption

Administrators can enable end-to-end encryption with the SQL database by using an Encrypted connection. This is a feature that is built into Microsoft SQL Server and Secret Server supports. To enable encryption:

1. Go to **Admin > See All**. The admin panel appears.
2. Type Database in the **Search** text box and select **Database**. The Database Configuration page appears:

Help

Secret Server supports Microsoft SQL Server versions 2012, 2014, 2016, 2017, and Express.

View [Collation Requirements](#). Need help installing or configuring SQL Server? View [Installing and Configuring SQL KB Article](#).

Database Configuration

SQL SERVER LOCATION

Server Name	QA-CUST-SQL-01
Database	SS_Playground

SQL AUTHENTICATION

Windows Authentication using Application Identity (GAMMA\ss_iis_svc) - **Recommended**
(Advanced Setting. Windows Authentication requires a Service Account and advanced IIS setup. Detailed in [Windows Authentication KB](#).)


SQL Server Authentication *(SQL Authentication requires Mixed Mode. See setup in [SQL Authentication KB](#).)*

[+] ADVANCED (NOT REQUIRED)


Edit View Audit

3. Click the **Edit** button.
4. Click the **Advanced (Not Required)** link. A new section appears:

[+] ADVANCED (NOT REQUIRED)



SSL Encryption  Enable

Trust Server Certificate Enable

Failover Partner 
(Requires SQL Server Configuration change)

Multi-Subnet Failover Enable
(Enabling Multi-Subnet Failover for AlwaysOn Availability Groups requires SQL Server 2012 and higher with AlwaysOn enabled)

Connection Timeout (in seconds)

 Save Database Connection Settings  Cancel

5. Click to select the **SSL Encryption** check box.
6. Click the **Save Database Connection Settings** button.

Note: SQL Server must be pre-configured to support encryption. This [Microsoft TechNet article](#) explains how to configure the SQL Server environment for encryption. The SSL encryption used for communicating with SQL Server is either 40 or 128 bit, depending on the Windows operating system used.

Note: Using this setting can adversely affect [performance](#) (KBA). See this [TechNet article](#) for additional information.

Manual IIS Installation

IIS is an internal part of the Windows operating system, and only needs to be enabled. If IIS is not found, the Delinea Installer will install it for you. If you would prefer to install IIS manually, please refer to the instructions listed below for example steps in the Windows Server 2016 Operating System. For the most up-to-date setup instructions, see [Microsoft's Technical Documentation](#). Navigate to **Docs > Internet Information Services > Install**.

Roles and Features

Delinea products recommend the following roles and features to be installed on the Secret Server IIS Server for maximum security and functionality options:

Roles

- Web Server (IIS)
- Web Server (IIS)\Web Server
- Web Server (IIS)\Web Server\Common HTTP Features
 - Default Document
 - Directory Browsing
 - HTTP Errors
 - Static Content
 - HTTP Redirection
- Web Server (IIS)\Web Server\Health and Diagnostics
 - HTTP Logging
- Web Server (IIS)\Web Server\Performance
 - Static Content Compression
 - Dynamic Content Compression
- Web Server (IIS)\Web Server\Security
 - Request Filtering
 - Windows Authentication
- Web Server (IIS)\Web Server\Application Development
 - .NET Extensibility 4.6
 - ASP.NET 4.6
 - ISAPI Extensions
 - ISAPI Filters
- Web Server (IIS)\Web Server\Management Tools

- IIS Management Console

Features

- .NET Framework 4.x Features
 - .Net Framework 4.x
 - ASP.NET 4.x
- WCF Services
 - HTTP Activation
 - TCP Activation
 - TCP Port Sharing
- PowerShell
 - Windows PowerShell 5.1

Step One: Windows Server 2012–2019 IIS Installation

To install Internet Information Services (IIS) Manager on Windows Server 2016, you will need to give your server the Web Server (IIS) role using the following procedure:

Note: If this is *not* the first time you have run the wizard (that is, when first installing IIS), the Web Server Role (IIS) and Role Services windows will not appear, and the wizard order changes a bit. Instead, role services are selectable in the Server Roles window.

1. Click the **Server Manager** button on your server. The Server Manager Dashboard appears.
2. Click the **Add Roles and Features** button. The Add Roles and Features Wizard on the Before You Begin window appears.
3. Click the **Next** button. The Select Installation Type window appears.
4. Click to select **Role-based or feature-based installation** selection button.
5. Click the **Next** button. The Select Destination Server window appears.
6. Ensure the **Select a Server from the Server Pool** selection button is selected.
7. In the **Server Pool** section, click to select your server.
8. Click the **Next** button. The Select Server Roles window appears.
9. Click to select the **Web Server (IIS)** check box.
10. Click the **Next** button. The Select Features window appears.
11. In the **Features** list, Click to select the following checkboxes (If necessary, click the **Add Features** button when prompted):
 - .NET Framework 4.x Features > WCF Services > **HTTP Activation**
 - .NET Framework 4.x Features > WCF Services > **TCP Activation**
12. Click the **Next** button. The Web Server Role (IIS) window appears.

13. Click the **Next** button. The Select Role Services Window appears.
14. In the **Roles** list, click to select the following check boxes:
 - Note:** Leave all the auto-selected check boxes as is.
 - Web Server (IIS) > Web Server > Common HTTP Features > **HTTP Redirection**
 - Web Server (IIS) > Web Server > Performance > **Dynamic Content Compression**
 - Web Server (IIS) > Web Server > Security > **Windows Authentication**
15. Click the **Next** button. The Confirmation window appears
16. Confirm your installation details.
17. Click the **Install** button. Wait for the installation to complete. The Results window appears.
18. Click the **Close** button. An IIS tile should now appear on your server.

Note: We recommend you run [Windows Update](#) to install the latest security patches for IIS once you have IIS installed.

Step Two: Configure the IIS Website

Follow these steps to configure a website in IIS for Secret Server:

1. Extract the Secret Server files into C:\inetpub\wwwroot\SecretServer or your location of choice.
2. Open Internet Information Server (IIS) Manager: On the taskbar, click **Server Manager > Tools > Internet Information Services (IIS) Manager**.
3. In the Connections pane, expand the server name.
4. Click on the **Application Pools** node. The Application Pools window appears.
5. Click the **Add Application Pool** link. The Add Application Pool dialog box appears.
6. Type SecretServer in the **Name** text box.
7. Click to select **4.x** in the **.NET Framework Version** dropdown list.
8. Click to select **Integrated** in the **Managed Pipeline Mode** dropdown list.
9. Click the **OK** button to save the new application pool. The dialog box closes.
10. (optional) Customize the Windows account Secret Server runs as:
 1. Right click the new application pool and select **Advance Settings...**
 2. Click the **Identity** setting in the **Process Model** section to select the desired account. Using this, you can, for example, set Secret Server to use IWA to connect to SQL.
11. Expand the **Sites** node on the **Connections** tree.
12. Click on the Default Web Site node.
13. In the **Actions** pane, click **Bindings** to set your desired website. The Edit Bindings dialog box appears.
14. Edit or add bindings as desired. We recommend using HTTPS with a real SSL certificate.

15. Click the **Close** button.
16. In the **Connections** tree, expand the **Default Website** node.
17. **Either**, If you see the default folder, **SecretServer**, which you created earlier:
 1. Right click the **SecretServer** folder and select **Convert to Application**. The Add Application dialog box appears.
 2. Click the **Select...** button to choose the pool you created earlier for Secret Server.**Or**, If you used a custom location instead:
 1. right click the Default Website. The Add Application dialog box appears.
 2. Type SecretServer in the **Alias** text box.
 3. Click **Select...** and pick the app pool created for Secret Server.
 4. Type the path where you extracted the Secret Server files in the **Physical Path** text box.
18. Click the **OK** button.

Step Three: Ensure IIS Does Not Stop the Worker Process

When using IIS version 7.0 and above, by default, the worker process terminates after an inactive period. If Secret Server is in its own application pool, that application pool will stop after a period of no requests. To ensure this does not happen, perform the following procedure. Additionally, by default, IIS launches a worker process when the first request for the Web application is received, so if the Secret Server application takes a long time to start, issues can result. Thus, we recommend launching the Secret Server application pool worker process as soon as IIS starts by setting the start mode to "AlwaysRunning."

Procedure:

1. Open **Internet Information Server (IIS) Manager**:
 - o If you are using Windows Server 2012 or Windows Server 2012 R2: On the taskbar, click **Server Manager > Tools > Internet Information Services (IIS) Manager**.
 - o If you are using Windows Server 2008 or Windows Server 2008 R2: On the taskbar, click **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, expand the server name.
3. Click **Application Pools**.
4. Determine which application pool Secret Server is running as:
 1. Expand **Sites** at the left.
 2. Find the website Secret Server is running on.
 3. Click on the Secret Server website or virtual directory (if it is running on one).
 4. Click **Basic Settings** on the right panel. This indicates Secret Server's application pool.
5. Right-click the application pool and select **Advanced Settings...** The Advance Settings dialog appears.
6. In the **General** section, set **Start Mode** to **AlwaysRunning**.
7. In the **Process Model** section, set **Idle Time-out (minutes)** to **0**.

8. In the **Recycling** section, set **Regular Time Interval (minutes)** to **0**.
9. In the **Recycling** section, click the > next to **Specific Times** to ensure there are no times set. If there are, click the ... to clear them.
10. Leave IIS Manager open—we will return to it below.

Step Four: Ensure the User Profile Always Loads

As of version 10.2, Secret Server requires its application pool "Load User Profile" setting enabled. Otherwise, Secret Server reports a critical alert to system admins.

Note: Even without the setting enabled, Secret Server loads to give access to secrets but many internal operations may malfunction, so we recommend resolving this issue as soon as possible.

Procedure:

1. Right-click the Secret Server application pool in IIS Manager and select **Advanced Settings...** The Advance Settings dialog appears.
2. Go to the **Process Model** section in the **Advanced Settings** dialog.
3. Set **Load User Profile** to **True**.
4. Perform an `iisreset` on the server (in an administrator command prompt).

Installing and Configuring SQL Server

For step-by-step instructions on how to install SQL 2016, see [SQL Server 2016 Standard Edition Installation](#).

Secret Server requires Microsoft SQL Server as the back-end database. All editions including the Express version of 2012–2017 are supported.

Setting up SQL Server requires:

- Installing SQL Server
- Creating a SQL Account
- Configuring database access in Secret Server
- Installing SQL Server

Note: If you are using SQL Express make sure to get the edition with tools that will include SQL Management Studio. Follow the link in the KB article [Download SQL Express with Tools](#).

Creating a SQL Account

SQL Authentication

The fastest method to get started with Secret Server is to create a SQL Authentication account. Follow the instructions in the Database section of the [Installation Guide](#).

For troubleshooting and configuring SQL installation on a different server than the application server see [SQL Authentication Configuration](#) article.

Windows Authentication

A more advanced way to have Secret Server access the SQL server would be through a service account and using Windows Authentication. Because of the requirement of a service account and added IIS settings, we only recommend this for non-evaluation setups. See instructions in [Configuring Integrated Windows Authentication](#).

Configuring Database Access in Secret Server

Once the account has been created and SQL server installed with the MSI. The third step of the Web installer will ask for database access information.

SQL Location

- **Server Name or IP:** If it is a local machine the server name will be (local) or localhost for the default instance, or if a named instance such as SQL Express it would be localhost\SQLEXPRESS. If you are unsure, copy the value from the "Server name" text box when connecting through SQL Management Studio.
- **Database Name:** If you have created a database, enter the name. If you have given the SQL account dbCreator permission, enter a database name for Secret Server to create.

SQL Authentication

- **SQL Server Authentication:** Implies a SQL account has been created that exists only with SQL Server. The account will need to be dbOwner on the database or need dbOwner permission to create the database. This is recommended for quickest setup. For more detailed information and troubleshooting see [SQL Authentication Configuration](#) article.
- **Windows Authentication:** The identity of the application pool will access the database. This requires a domain Service account that has been granted access to run ASP.Net and the database. This is an advanced setting that is not recommended for evaluations. Follow the instructions on using a service account in [Configuring Integrated Windows Authentication](#).

Installing RabbitMQ

Overview

What is RabbitMQ?

RabbitMQ is a robust message queuing software package that Secret Server uses to communicate with its distributed engines. For detailed information about RabbitMQ go to the [RabbitMQ website](#)

Why Install It?

RabbitMQ is an enterprise-ready alternative to MemoryMQ. While MemoryMQ is sufficient for basic and prototyping installations, RabbitMQ is the preferred messaging framework when the need for greater reliability and clustering arises.

RabbitMQ and Encryption

All data sent from or read by Secret Server from RabbitMQ is encrypted. If you would like to add SSL despite the data already being encrypted, please follow the "Advanced installation of RabbitMQ with TLS" use case. Please note that Delinea Support can help with non-SSL installations. For SSL installation, configuration, troubleshooting, and RabbitMQ clustering, please contact [Delinea Professional Services](#) to learn more about our Professional Services rates.

Downloading Delinea's RabbitMQ Helper

Please go to [Delinea RabbitMQ Helper](#) to download the most recent version.

Prerequisites

Important: Secret Server only supports RabbitMQ on Windows operating systems.

RabbitMQ requires:

General

- Windows Server 2008 or higher with PowerShell v3 support
- Nodes hosting RabbitMQ need a minimum of 2 GB RAM
- Nodes hosting RabbitMQ should have at least 128 MB of memory available at all times
- Disk space is not an issue, but it should not go below 50 MB (default value), especially if you host RabbitMQ on the same server as Secret Server
- Minimum two vCPUs. This is an **absolute minimum** otherwise installation fails without much useful feedback to troubleshoot. We strongly recommend four vCPUs or more.
- Ports 5672 (non-SSL) or 5671 (SSL) opened on the machine and firewall

SSL Certificate

- A server certificate PFX type and a root certificate authority certificate CER type.
- The PFX certificate should have:
 - A name that matches the RabbitMQ Fully qualified machine name
 - If you plan on making a RabbitMQ cluster, add DNS names (SANs) to your certificate
 - Your certificate must be an RSA certificate. CNG is not supported and will cause the installation to fail.
- If you do not have an internal PKI and prefer not to use a public certificate, you can use a self-signed certificate.

Note: Delinea will not assist with creating or troubleshooting self-signed certificates.

Installation

Task 1: Secret Server

In Secret Server UI

1. Navigate to **Admin > Distributed Engine**.
2. Click the **Manage Site Connectors** button. The Manage Site Connectors page appears:

SITE CONNECTOR	ACTIVE	VALIDATED	QUEUE TYPE	HOST	VERSION
Default MemoryMq Service	Yes	No	MemoryMq	QA-CUST-01	5.0.0.40
RMQ_for_BUG169089	Yes	No	RabbitMq	EARTH.solar.local	Unknown

Show Inactive

[← Back](#) [+ New Site Connector](#)

3. Click the **+ New Site Connector** button. The Site Connector Details page appears:

Site Connector Details

Queue Type: Memory MQ

Name: *

Active:

Use SSL:

Host Name: *

Port: 8672 *

[Save](#) [Cancel](#)


4. Click to select **Rabbit MQ** in the **Queue Type** dropdown list.
5. Type a name for your new site connector in the **Name** text box.
6. Click to select the **Active** check box.
7. Type the host name of the machine where you plan to install RabbitMQ in the **Host Name** text box.


Note: The Engines need to be able to resolve this host name or the connection will fail. Also, inbound firewall rules must be

created on the machine that is hosting the connector.

8. Type either port 5672 (non-SSL) or 5671 (SSL) in the **Port** text box.
9. Click the **Save** button.
10. After the site connector is created, click the site connector's link. The Site Connector Details page appears:

Site Connector Details

 Connectivity has not been validated.

 To be included in the mailing list about updates to MemoryMq, please send an email to MemoryMQ@thycotic.com.

Queue Type MemoryMq

Name Default MemoryMq Service

Active Yes

Use SSL No

Host Name QA-CUST-01


Port 8672


← Back
✎ Edit
👁 View Credentials
📄 Download Site Connector Installer


✔ Validate Connectivity

11. Click the **View Credentials** button to retrieve the automatically generated credentials. The Site Connector Credentials popup appears. You can ignore the informational message that the connectivity has not been validated for now as you will be doing so after you install RabbitMQ on the host you have selected.

Site Connector Credentials ✕

 These credentials are automatically used by the Site Connector and by Engines that use this Site Connector.

User Name  eFAQUJzHtIHUtDnD05YME0ZOLxF8qggv09OqAVfJudDpxJVIBEr0gAzGG-7nz5F

Password  FHkdCq6gSRlit30L4Vcb1x9EZDyggqJe8rho5R1hdjtxaitnaTYoya4HOQu4M3U0

✔ OK

12. Click the copy icons to copy both the **User Name** and **Password**, and store them for use in the next section.
13. Click the **OK** button.

Task 2: RabbitMQ Host

1. Download the [Delinea RabbitMQ Helper](#).
2. Install the Delinea RabbitMQ helper by running the downloaded MSI.

3. Review the supported [installation scenarios](#).
4. Navigate to the installation folder in %PROGRAMFILES%\Thycotic Software Ltd\RabbitMq Helper
5. Launch the Delinea.RabbitMq.Helper.exe, which opens the Windows PowerShell.
6. Then, issue a cmdlet command from the scenario that applies to your need.
7. After installation completes, the helper opens a Web browser to the RabbitMQ management console. There is no need to interact with the site at this time, so you can minimize or close the page for now.
8. Return to Secret Server, and go to the site connector you created in the previous section.
9. Click the site connector's link. The Site Connector Details page appears.
10. Click the **Validate Connectivity** button.
11. If everything is set up correctly, you will see "Validation Succeeded."
12. If you see "Validation Failed," do the following:
 1. Ensure the RabbitMQ Windows service is running.
 2. Check the logs found under C:\Program Files\Thycotic Software Ltd\RabbitMq Site Connector\log.
 3. Check the Secret Server system log for a full error report.

Troubleshooting

Please refer to [RabbitMQ Helper](#).

Clearing RabbitMQ Message Queues

Some users note that older RabbitMQ message queues in Ready state are not clearing as expected, so messages accumulate. To clear the message queues, use the procedure below.

1. On the machine where RabbitMQ is installed, download the [utility](#) for removing old RabbitMQ queues.

SHA1(RMQ_QueueRemoval.zip)= B8EE3CD2488AF2D7A42421B870EB8041434245C8

SHA256(RMQ_QueueRemoval.zip)= B9AF3BF51B0E1E6E937830A6CF0974D3546183B78E1E86F6C8563E5E7243146A

2. Extract the zip file.
3. Open Windows PowerShell.
4. Navigate to the directory where you extracted the zip file.
5. Load the file by typing the following command:

```
.\RMQ_QueueRemoval.ps1
```

6. Run the commands below in the order shown:

1. ShowAllQueues
2. ShowQueuesNoConsumer
3. DeleteQueuesNoConsumer

Installing Secret Server via the Command Line

Overview

ThycoticSetup.exe accepts command line arguments for a silent or automated installation. This topic discusses how to do that.

Basic usage:

```
ThycoticSetup.exe -q -s PARAMETER=<value> PARAMETER2=<value> (/nodetect) (/l <log file path>)
```

Important considerations:

- Always pass `-q -s` to `ThycoticSetup.exe` and then pass in your parameters or switches.
- There are two stages to the installer. The first (optional) stage is to install the prerequisites such as IIS and .NET 4.8. Then in a second stage, once all required pre-reqs are present, you can install Secret Server or Privilege Manager (PM).
- The installer UI performs additional validation steps, such as testing the database connection information, that a silent CLI one does not. Thus, this install can fail if you provide incorrect settings.
- The installer checks to see if Secret Server and PM are already installed by default. It will install them if either is not found. If you would like to specify which applications to install, you must use the `/nodetect` switch to avoid the automatic detection, so the `InstallSecretServer` and `InstallPrivilegeManager` settings are respected.
- Due to how MSI installers work, if you need to pass in parameters that contain spaces, use the special `CMDLINE` parameter, using extra double quotes to delineate each parameter. For instance:

```
ThycoticSetup.exe -q -s PARAM1=some_string PARAM2=1234 CMDLINE=" PARAM3=""Something with a space"" PARAM 4=""Another value with spaces"" "
```

Note:

- You can mix and match regular parameters (numerical values, or strings without spaces) with `CMDLINE`.
- Any parameters sent inside of `CMDLINE` are treated as strings. Numerical parameters inside of `CMDLINE` are ignored.
- Be aware of how you call `ThycoticSetup.exe` and what special characters need escaping in your shell. This includes passwords containing symbols. What is required is shell dependent. For example, running the installer from PowerShell (or a .ps1 script) rather than an older command prompt (or a .bat file) would require escaping a different set of special symbols.
- We recommend using the `/l <logfile>` option to create a log file, which you can use to verify your parameters are correctly passed to the installer. This is especially useful when using `CMDLINE` for parameters with spaces, which is prone to mistakes.

Note: For security, parameters involving passwords are not logged.

Install Prerequisites

As of Secret Server 10.11, you can silently install all required prerequisites. These are the same prerequisites the "Fix Issues" button in the installer UI fixes. The important difference is missing prerequisites are not auto detected—you must tell the installer which ones you want installed. Older versions will not do a silent command line installation unless these necessary prerequisites are already installed.

Parameters

Table: ThycoticSetup.exe Parameters

Parameter	Value	Required (if not present)	Purpose	InstallPreReqs	Boolean	Yes	Triggers the prerequisites installation.
			Enable this to install prerequisites with no application. This overrides <code>InstallSecretServer=1</code> and <code>InstallPrivilegeManager=1</code> .				
			Comma separated list (see below)				Specifies which prerequisites to install. Unexpected Link Text

Prerequisites

Table: Prerequisite (PRE_REQS_TO_INSTALL) Values

Prerequisite	Required (if not present)	Purpose
CONFIGURE_FIPS	No	Ensures the AES 256 and 128 ciphers are enabled in Windows.
ENABLE_FIPS	No	Enables FIPS mode in Windows. Generally, not needed, unless required by your environment.
INSTALL_HTTPS_BINDING	No	Enables HTTPS binding in IIS for the default website. Tries to pick an existing valid SSL certificate and creates a self-signed certificate if necessary for temporary use. Always use HTTPS with a valid certificate in production environments.
INSTALL_IIS	Yes	Installs the Web Server (IIS) Windows Role.
INSTALL_IIS_COMPS	Yes	Installs various required IIS features.
INSTALL_NET_WCF	Yes	Installs the WCF HTTP and TCP activation features.
INSTALL_NetFx48	Yes	Installs .NET 4.8. This requires a reboot. Unexpected Link Text

Single-Line Example

```
ThycoticSetup.exe -q -s InstallPreReqs=1
PRE_REQS_TO_INSTALL=INSTALL_IIS,INSTALL_IIS_COMPS,INSTALL_NET_WCF,INSTALL_HTTPS_BINDING,INSTALL_NetFx48 /I C:\temp\install-prereqs.log
```

Installing Applications

Secret Server or Privilege Manager can be installed and pre-configured using these parameters. If the required prerequisites are not already present, the installer exits. They can both be installed at the same time but will then share the same database and email settings.

If you need more control over configuring the website, you can create a site and configure it in advance (that is, using IIS's AppCmd.exe), and then pass the preconfigured website name as SecretServerSiteName Or PrivilegeManagerSiteName.

Secret Server Parameters

Table: Secret Server Parameters

Parameter	Value	Default	Notes
CreateWebSite	Boolean	0	Required if the SecretServerSiteName website does not exist.
InstallSecretServer	Boolean	1	Whether or not to install Secret Server. Must also use the /nodetect switch to avoid this being set to 1 if not already installed.
SecretServerApplicationName	String	SecretServer	Used for the application pool name as well as the website application or subfolder.
SecretServerAppPassword	String	Optional	Only required if you are configuring SecretServerAppUserName.
SecretServerAppUserName	String	Optional	Only required if you are configuring SecretServerAppPassword.
ApplicationPoolIdentity	String	Optional	What identity to run the app pool as. The user must already exist.
SecretServerConfigLogFile	String	Optional	The base <logname> is specified with the /I option.
SecretServerDestinationFolderPath	String	Optional	If you would like to use a directory containing spaces, see note above about how to use the CMDLINE parameter to pass it in.
SecretServerSiteHttpsPort	Integer	Optional	HTTPS port. Always use HTTPS in production. If using the default website, this port is an additional HTTP binding, along with the default. If you use this option to bind HTTPS on another port, configure the HTTPS binding yourself and choose a certificate after the installer completes.
INSTALL_HTTPS_BINDING	Boolean	0	only configures the certificate on the normal 443 binding.
SecretServerSiteName	String	Optional	Used by both Secret Server and PM. Must already exist, unless you also use CreateWebSite=1 (see the Website Parameters section below).
SecretServerSitePort	Integer	Optional	HTTP port. If using the default website, this port HTTP binding is in addition to any defaults.
SecretServerUserDisplayName	String	Optional	The display name for SecretServerUserName.
SecretServerUserEmail	String	Optional	The email address for SecretServerUserName.
SecretServerUserName	String	Optional	The initial Secret Server administrator user. If not set, once Secret Server is installed, the first person to visit the website will be able to pick the details on the "Create Initial Administrator" page.
SecretServerUserPassword	String	Optional	The password for SecretServerUserName. Unexpected Link Text

Privilege Manager Parameters

Table: Privilege Manager Parameters

Parameter	Value	Default	Notes
CreateWebSite	Boolean	0	Required if the PrivilegeManagerSiteName website does not exist.
InstallPrivilegeManager	Boolean	1	Whether or not to install PM. Must also use the /nodetect switch to avoid this being set to 1 if not already installed.
PrivilegeManagerApplicationName	String	TMS, TMSAgent, TMSWorker	Used as the base name of the PM application pools (Regular, Agent, and Worker), plus the website application or subfolder.
PrivilegeManagerAppPassword	String	None	Only required if you are configuring

PrivilegeManagerAppUserName. | PrivilegeManagerAppUserName | String | None | What identity to run the app pool as. User must already exist. | PrivilegeManagerDestinationFolderPath | String | C:\inetpub\wwwroot\TMS | If you would like to use a directory containing spaces, see above on using the CMDLINE parameter. | PrivilegeManagerLogFile | String | None | Optional | PrivilegeManagerSiteName | String | Default website | CreateWebsite=1 must also be set to customize this. | [Unexpected Link Text](#)

Required Database Parameters

Table: Required Database Parameters

Parameter	Value	Default	Notes
DatabaseConnectionTimeout	Integer	15	Database connection timeout in seconds.
DatabaseEnableMultiSubnetFailover	Boolean	0	If your application connects to an AlwaysOn availability group (AG) on different subnets, setting MultiSubnetFailover=true provides faster detection of and connection to the (currently) active server. For more information about SqlClient support for AlwaysOn availability groups, see SqlClient support for High Availability, Disaster Recovery .
DatabaseEnableSslEncryption	Boolean	0	Whether or not to use SSL/TLS for the database connection.
DatabaseFailoverPartner	String	None	The name or address of the partner server to connect to if the primary server is down. Only used if DatabaseEnableMultiSubnetFailover=1.
DatabasesUsingWindowsAuthentication	Boolean	1	Whether or not to use integrated Windows authentication for MSSQL access. If enabled, before running the install, you must configure your IIS application pool to run a Windows account permission to access the DatabaseServer, and the DatabaseUserName and DatabasePassword will not be used.
DatabaseName	String	None	Database name. Is created if it does not exist. Defaults to SecretServer if installing SQL Express.
DatabasePassword	String	None	Database SQL login password. Ignored if using Windows authentication.
DatabaseServer	String	None	Database server hostname or IP.
DatabaseTrustServerCertificate	Boolean	0	Only used if DatabaseEnableSslEncryption=1 is set. Do not enable in production if you are using SSL encryption; certificate trust validation is critical to security. Certutil.exe can be used to diagnose untrusted certificates. When TrustServerCertificate is set to true, the transport layer uses SSL to encrypt the channel and bypass walking the certificate chain to validate trust. If TrustServerCertificate is set to true and encryption is turned on, the encryption level specified on the server is used even if Encrypt is set to false. The connection fails otherwise.
DatabaseUserName	String	None	Database SQL login username. Ignored if using Windows authentication.
InstallSqlExpress	Boolean	0	Whether or not to install the free SQL Express to use as a database server. If enabled, none of the other database parameters are used. We only recommended this for testing. , do not use in production due to performance limits. Unexpected Link Text

Email Parameters

You can set email parameter either in the UI after installation or by using these parameters at install time.

Table: Optional Email Parameters

Parameter	Value	Default	Notes
EmailDomain	String (optional)	None	Domain for SMTP credentials. Used if EmailUseCredentials=1.
EmailFromAddress	String (required)	None	The "from" address to use when sending emails.
EmailPassword	String (optional)	None	Password for SMTP credentials, used if EmailUseCredentials=1.
EmailPort	Integer	25	The TCP port used to connect to the SMTP server.
EmailServerName	String (required)	None	Hostname or IP of a SMTP server.
EmailUseCredentials	Boolean	0	Whether or not SMTP credentials should be sent when connecting.
EmailUseCustomPort	Boolean	0	Whether or not to use a custom port connecting to the email server.
EmailUserName	String (optional)	None	Username for SMTP credentials. Used if EmailUseCredentials=1.
EmailUseSSL	Boolean	0	Whether or not to use SSL/TLS when connecting to the email server Unexpected Link Text

Single-Line Example

This example installs Secret Server and not PM, leaving variable for the database parameters:

```
ThycoticSetup.exe -q -s InstallSecretServer=1 InstallPrivilegeManager=0 DatabaseServer=<hostname> DatabaseName=<SecretServer> DatabaseUserName=<username> DatabasePassword=<password> /I C:\temp\ss-install.log /nodetect
```

Moving Secret Server to Another Machine

If you are moving/migrating Secret Server to a new machine and have installed IIS and .NET Framework as described in the Installation Guide on the new machine, you do not need to run the installer; you just need to follow the steps below:

1. If you use the "Force HTTPS/SSL" option, disable it by clicking **Configuration** from the **Administration** menu, and then click the **Security** tab, and **Edit**. You can re-enable the "Force HTTPS/SSL" option after you set up and install an SSL certificate on the new machine.

Note: If you are also moving the SQL Server database, be sure to create a new backup of the database, as this setting is written to it. To move the database, follow the steps in [Moving the Microsoft SQL Server Database to Another Machine](#).

2. If you have configured encryption of your key using DPAPI, you will also need to turn this off before continuing with Step 3. To do so, click **Configuration** from the **Administration** menu, then click the **Security** tab. Click **Decrypt Key** to *not use* DPAPI and enter your Secret Server account password.
3. Copy the folder that holds your Secret Server instance to the new computer.
4. Shut down the old web site and recycle its application pool as it is running background threads that are accessing the database.
5. Set up the new folder in Internet Information Server (IIS) as a virtual directory/application under the Default Web Site or as a separate Website. For detailed instructions, refer to [Advanced Installation](#) in the Installation Guide.
6. If your database server and credentials have not changed, skip this step. If they have changed, follow the steps below:
 1. Delete the database.config file from the secretserver folder (on the ASP.NET/IIS machine).
 2. Restart your new Secret Server website, so it is running.
 3. Browse to your Secret Server URL \ dbconnectionreset.aspx http:\\secretserverurl\dbconnectionreset.aspx and you will be prompted to enter your new database connection details.
 4. Enter your new SQL Server and the account information.
 5. Click **Next** and the site will connect to the new database. Your site is now pointing the new database.
7. Activate the licenses for the new server by going to the **Licenses** page.
8. If you are using certs, remember to set them on your new IIS, and then browse to Secret Server over HTTPS and re-enable force HTTPS if this was set on the original machine.
9. Re-enable DPAPI if this was disabled in the earlier step.

Note: If you are moving Secret Server web application from Windows Server 2008 to 2012 AND your Secret Server is below version 8.5, make sure that:

- .Net extensions 3.5 and ASP.Net 3.5 when adding the IIS role on the new server.
- Change the Secret Server application pool to 2.0 and recycle the application pool after running the installer.

Moving the Microsoft SQL Server Database to Another Machine

Important: This article only applies if your MS SQL Server database is only for Secret Server.

Follow the steps below for moving MS SQL Server database for Secret Server.

Task 1: Backing up and Restoring the Database

To back up your Secret Server installation:

Note: Your Secret Server instance may be running during this procedure.

1. Stop the Secret Server site in Internet Information Server (IIS) to prevent any changes to the database.
2. Navigate to the directory where Secret Server is installed.
3. Copy the folder (holding the application) to your back up location.
4. Open your SQL Server Management Studio.
5. Right click the database your Secret Server is running on, and select **Tasks > Backup**.
6. Click the **Add** button. You are prompted to enter a file path for the .bak file. This can be the final destination (not recommended) or a temporary one (for later moving to a back up location).
7. Make sure SQL Server has permissions for this location. That is, create (if needed) and or grant access to the account that will access the database (see the [Installation Guide](#) for account creation instructions). See [Running the IIS Application Pool As a Service Account](#) (Task 2) for details.
8. Copy the resulting database backup file (.bak) to your backup location.

Note: You can also automate steps 2-4 using the command: `osql -S myserver\SQLEXPRESS -E - Q "BACKUP DATABASE SECRETSERVER TO DISK = 'c:\backup\ss.bak' .`

Note: We recommend taking the old database offline after all steps are complete.

Task 2: Connecting Secret Server to the New Database

1. Restart your Secret Server website in IIS.
2. Log on Secret Server as a local admin.
3. Navigate to `https://<your_ss_url>/Setup/Database`. The Database Configuration page appears:

Help

Secret Server supports Microsoft SQL Server versions 2012, 2014, 2016, 2017, 2019, and Express.

View [Collation Requirements](#). Need help installing or configuring SQL Server? View [Installing and Configuring SQL KB Article](#).

Database Configuration

SQL SERVER LOCATION

Server Name	QA-CUST-SQL-01
Database	SS_Playground

SQL AUTHENTICATION

Windows Authentication using Application Identity (GAMMA\ss_iis_svc) - **Recommended**
(Advanced Setting. Windows Authentication requires a Service Account and advanced IIS setup. Detailed in [Windows Authentication KB](#).)

SQL Server Authentication *(SQL Authentication requires Mixed Mode. See setup in [SQL Authentication KB](#).)*

[+] ADVANCED (NOT REQUIRED)

Edit View Audit

Note: The settings here are stored in `C:\inetpub\wwwroot\Playground\database.config`. You can back that file up to revert or simply return to this page to reset the connection again. See the [Privilege Manager documentation](#) if you need to change its configuration too.

4. Click the **Edit** button. The page becomes editable.
5. Type your new SQL Server location (server name) and database.
6. Click the **Save Database Connection Settings** button, and the site will connect to the new database.
7. Your site is now pointing to the new database.

Note: To roll back changes and restore the original database, complete both tasks again to move the database back to the original database server.

Note: If you are also moving the Secret Server application to another server, see [Moving Secret Server to Another Machine](#) for more information.

Running the IIS Application Pool As a Service Account

Overview

We recommend setting up a domain service account that can both access the Delinea product's SQL database and run the IIS Application Pool(s) dedicated to your Delinea product.

Note: The service account created in this KB should **not** be the same account that is created during the installation of SQL and used to manage SQL as a whole.

To set up this service account correctly you will need to:

1. Create a service account in Active Directory that will be dedicated to your Delinea product (domain).
2. Granting the service account access to the SQL Server database.
3. Assign the service account as the identity of the application pool or pools in IIS.
4. Grant folder permissions for the service account on two folders.
5. Configure User Rights Assignment to the service account.

Procedure

Note: You must have IIS installed on your Web server before completing these steps.

Task 1: Creating a Domain Service Account

1. Create a local or domain user account (or identify one to use).
2. Open IIS (**Search > inetmgr**) on your Web server.
3. Open the **Active Directory Users and Computers** link from **Administrative Tools**.
4. Click to open the directory where you want to assign this account, such as testlab.com.
5. Select **Service Accounts**.
6. Right click and and select **New > User**. The "New Object - User" wizard dialog box appears.
7. Type a name and logon name for the service account.
8. Click the **Next** button. The wizard advances to the next dialog box (same name).
9. Type a password in the Password and Confirm Password text boxes.
10. If necessary, click to deselect the **User must change password at next login** check box.
11. Click to select the **Password never expires** check box. Failing to do this could lock the account out of Secret Server.
12. " Check **Password never expires** or the account could lock you out of Secret Server.
13. Click **Next** button.
14. Click the **Finish** button. You can now give the account access to the database server and the application server.

Task 2: Granting Access to the SQL Database

Note: You must have SQL installed on your database server before completing these steps.

Grant access:

1. Open the SQL Management Studio on your database server.
2. Connect to your Delinea product's SQL database using an administrator account.
3. Click to select the Security folder in the Object Explorer.
4. Right-click the same folder and select **New > Login...** A log on dialog box appears.
5. Ensure the **Windows Authentication** radio button is selected.
6. Click the **Search...** button. The "Select User, Service Account, or Group" dialog box appears.
7. Ensure that your domain or AD server appears in the **From this location** text box. If not, click the **Locations...** button and select it.
8. Type the login name you created for your Delinea service account, such as svc_thycotic, in the **Enter the object name to select** text box.
9. Click the **Check Names** button.
10. Click to select the correct account.
11. Click the **OK** button. The dialog box closes, returning you to the Login - New dialog box.
12. **Either**, if you have already created the database for your Delinea product:
 1. Click **User Mapping** in the **Select a page** list box.
 2. Click to select the check box for the database in the **Users mapped to this Login** list.
 3. Click to select the **db_owner** check box in the **Database role membership...** list.
13. **Or**, if you have not yet created the database:
 1. Click **Server Roles** in the **Select a page** list box.
 2. Click to select the **db_creator** check box.
14. Click the **OK** button.

Task 3: Assigning the Identity of Application Pools

1. Click the **Applications** node under the server name in the **Connections** tree.
2. Right-click the node and select **Advanced Settings...** The Advance Settings dialog box appears.
3. Click the ... button for the **Identity** entry in the **Process Model** section. The Application Pool Identity dialog box appears.
4. Click to select the **Custom Account** selection button.
5. Click the **Set...** button. The Set Credentials dialog box appears.
6. Type your service account's name, such as test and password.
7. Click the **OK** button. The dialog box closes.
8. Open the command console as an Admin.
9. Change the directory to your .NET framework installation directory using the "cd" command, for example,

C:\Windows\Microsoft.NET\Framework\v4.0.30319.

10. Type `.\aspnet_regiis -ga <domain name>\<username>`, replacing `<domain name>` and `<username>` with your information. For local accounts omit the domain name parameter.

Task 4: Granting Folder Permissions

Note: You must have the Delinea product application files installed (on your Web server) before completing this section.

Following the steps below, you give the service account "Modify" access to **two** folders:

- C:\Windows\TEMP
- The folder where your Delinea product's application files are located, such as C:\inetpub\wwwroot\SecretServer

Procedure (for each folder):

1. In a file manager, navigate to the Secret Server application folder.
2. Right-click the folder and select **Properties**. The Properties dialog box appears.
3. Click the **Security** tab.
4. Click the **Advanced** button.
5. Click the **Add** button. A permissions panel appears.
6. Click the **Select a Principal** link. The "Select User, Computer, Service Account, or Group" dialog box appears.
7. Ensure that your domain or AD server appears in the **From this location** text box. If not, click the **Locations...** button and select it.
8. Type the login name you created for your Delinea service account, such as `svc_thycotic`, in the **Enter the object name to select** text box.
9. Click the **Check Names** button.
10. Click to select the correct account.
11. Click the **OK** button. The dialog box closes, returning you to the permissions panel.
12. Click to select the **Modify** check box in the **Basic Permissions** section. Your service account should have the **Modify, Read & Execute, List folder contents, Read, and Write** permissions selected for this folder
13. Click the **OK** button.
14. Click the **Apply** button.

Note: If a Windows Security pop-up appears, click the **Yes** button. The service account will now be able to access this folder.

Note: The application folder only needs "Write" and "Modify" permissions during the installation or during an upgrade. You can remove these once the installation process is complete.

Task 5: Configuring User Rights

The following settings are required for Delinea Secret Server to function:

- "Log on as a batch job"
- "Impersonate a client after authentication"

You can adjust these settings either at the **Domain** level using group policy or locally on your IIS Web server using the Local Security Policy Console. See [User Rights Assignment](#) to learn more.

Option 1: Setting User Rights Assignment on the Domain

Note: This is an example of how to create a Group Policy Object (GPO), we recommend consulting with your organizational group policy administrator to create this policy.

Note: This overwrites any configuration in the local security policy. The local security policy is a safer option if you are not sure about usage across your domain.

1. Open the Group Policy Management Console.
2. Right-click the desired GPO folder (under the domain node) in the **Group Policy Management** Tree, and select **New**. The New GPO dialog box appears.
3. Type the name, such as "Delinea User Rights Assignment," in the **Name** text box.
4. Click the **OK** button. The dialog box closes.
5. Right-click the GPO you just created and select **Edit**. The Group Policy Object Editor appears.
6. On the **Computer Configuration** node, click to expand **Policies > Windows Settings > Security Settings > Local Policies**.
7. Click to select the ****User Rights Assignment**** folder.
8. Repeat the following procedure for the "Log on as a batch job" and "Impersonate a client after authentication" permissions (for this instruction we show the former):
 1. In the list on the right, right-click **Log on as a batch job** and select **Properties**. The "Log on as a batch job Properties" dialog box appears.
 2. Ensure that the **Define these policy settings** check box is checked.
 3. Click the **Add User or Group** button. A dialog box appears.
 4. Add your Delinea service account.
 5. Click the **OK** button. The dialog box closes. The new policy appears in the list.
 6. Click the **Apply** button.
9. Link your new GPO to the OU where your Delinea product machine accounts exist, that is, the Web and database servers.

Option 2: Setting User Rights Assignment Locally

1. On the Web server hosting IIS and your Delinea Application files, open the "Local Security Policy Console" as an administrator (Run as administrator).
2. On the Local Policies node, click to expand **Local Policies > User Rights Assignment**.
3. Click to select the **User Rights Assignment** folder.
4. Repeat the following procedure for the "Log on as a batch job" and "Impersonate a client after authentication" permissions (for this instruction we show the former):
 1. Right-click on **Log on as a batch job** in the list on the right and select **Properties > Add User or Group**.
 2. Click to select your Delinea service account.

3. Click the **OK** button.

Note: If you get a "Service Unavailable" error after applying "Log on as a batch job" permissions, try updating your group policy settings: Open the **Command Console**, type in `gpupdate /force**`, and restart the **Windows Process Activation Service**.

SQL Server 2016 Standard Edition Installation

Overview

The following steps walk you through setup and configuration for SQL Server 2016 Standard Edition as an example. For the most up to date resources on installing SQL see [Microsoft SQL Technical Documentation](#) for more information.

At the completion of this article you will have:

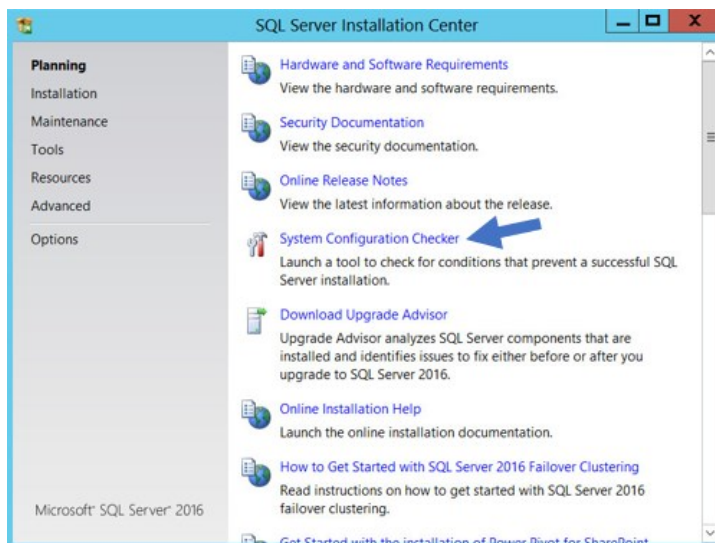
- Installed a basic stand-alone instance of SQL Server 2016 Standard with the minimum features necessary for SQL Server.
- Installed SQL Server Management Studio for managing the local database.
- Created a database in SQL for your Delinea product
- Created a new SQL Server user login for your SQL database

Note: This document uses Delinea's Secret Server product as example in the instructions, but the same steps apply for Privilege Manager advanced installs.

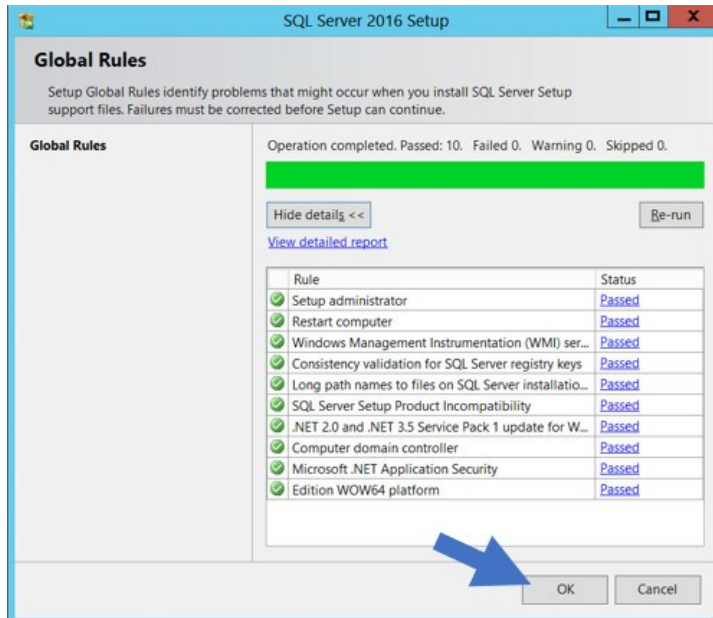
Procedures

Installing SQL Server 2016

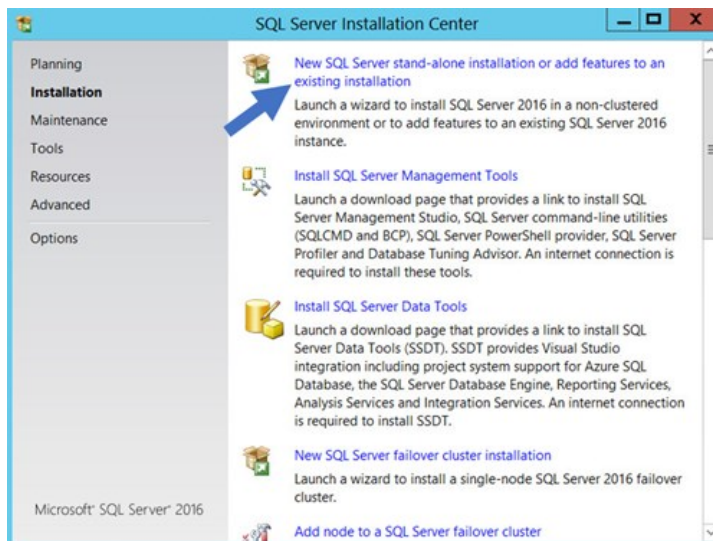
1. Launch the SQL Server installer from CD or file download. The SQL Server Installation Center opens to the Planning window:



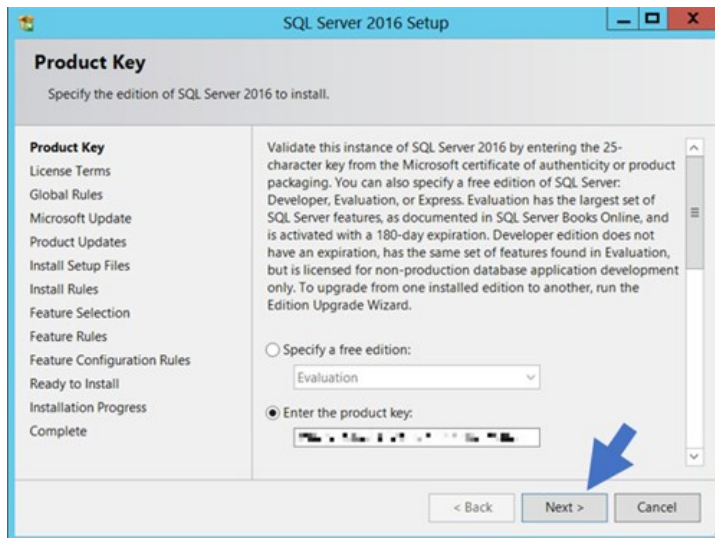
2. Click the **System Configuration Checker** link. This runs a tool that checks for conditions on your server that could prevent SQL Server from installing.
3. When the tool launches, click the **Show details** button. A successful scan should look like the one shown below. If you encounter any issues, look at the detailed report, resolve the reported issues, and rerun the scan.



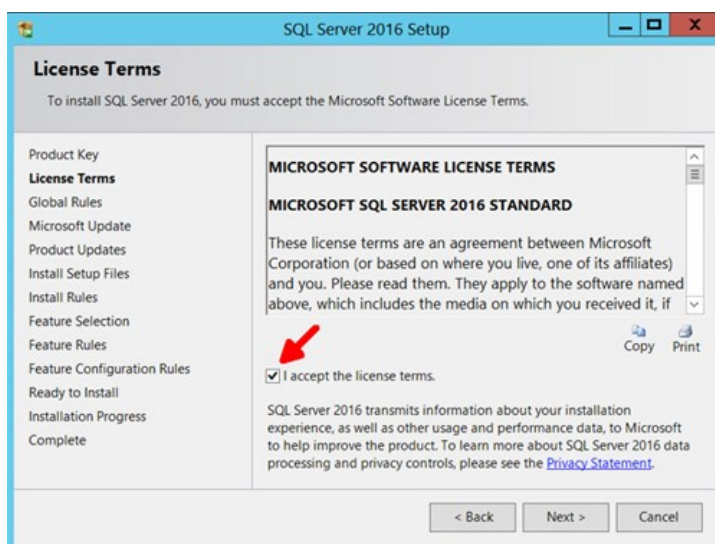
4. Click the **OK** button when done to return to the "SQL Server Installation Center" window.
5. In the SQL Server Installation Center window, click the **Installation** link. The Installation Window appears:



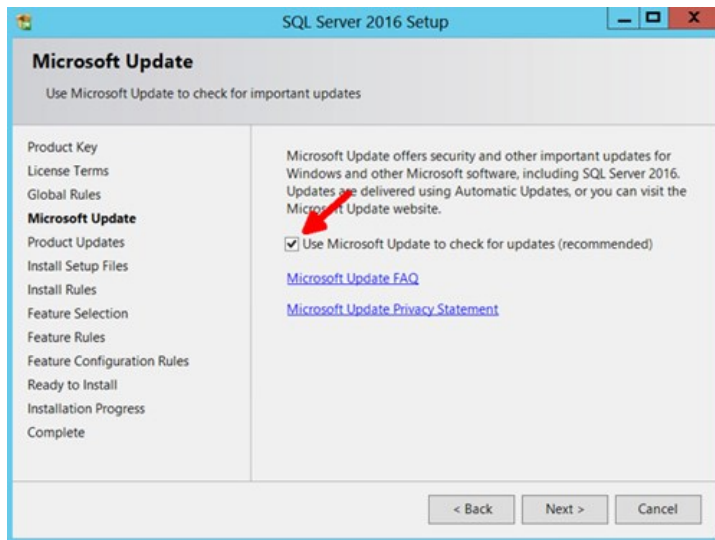
6. Click **New SQL Server stand-alone installation...** link. The Product Key page appears:



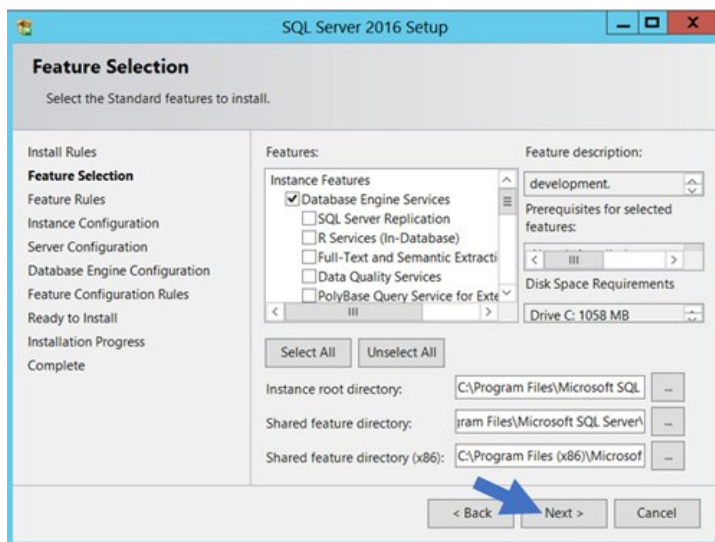
7. Click to select the **Enter the Product Key** selection button.
8. Type your product key in the the **Enter the Product Key** text box.
9. Click the **Next >** button. The License Terms page appears:



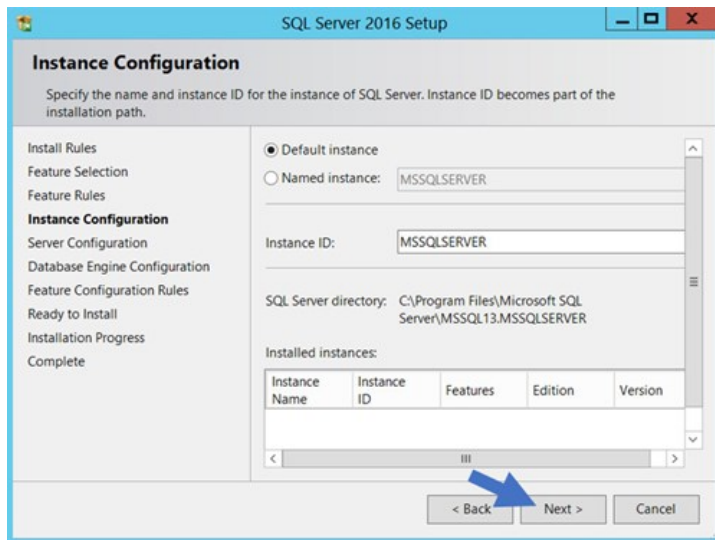
10. Click to select the **I accept the license terms.** check box.
11. Click the **Next >** button. The Global Rules page appears (not shown) after the rule check runs.
12. Click the **Next >** button. The Microsoft Update page appears:



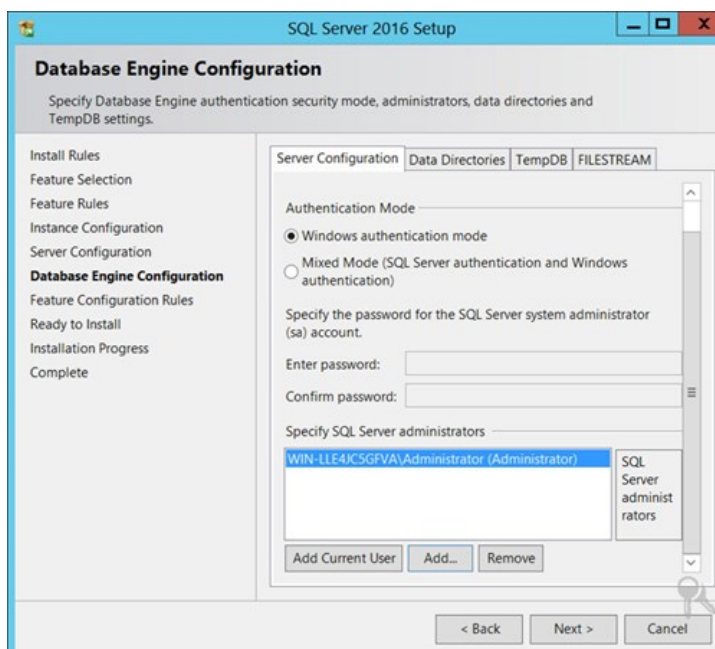
13. Click to select the **Use Microsoft Update...** check box to check for updates (recommended), unless your software update process does not use automatic updates from Microsoft
14. Click the **Next >** button twice to bypass the Product Updates page. The Install Setup Files page appears.
15. Wait for the installation to complete.
16. Ensure that all operations pass.
17. Click the **Next >** button twice to bypass the Install Rules page. The Feature Selection page appears:



18. Ensure the **Database Engine Services** check box is selected. This is the only feature necessary for Secret Server. Unless you are using Geo-Replication, you can leave everything else unchecked. Leave the directory locations unchanged.
19. Click the **Next >** button twice to bypass the Feature Rules page. The Instance Configuration page appears:



20. Ensure the **Default Instance** selection button is selected.
21. Type a name for your SQL Instance in the **Instance ID** text box.
22. Click the **Next >** button twice to bypass the Server Configuration page. The Database Engine Configuration page appears:

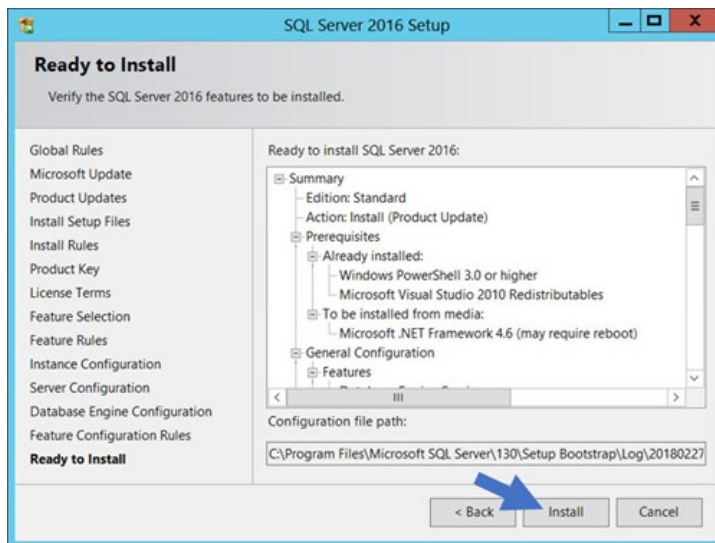


23. You have the choice to select either **Windows Authentication Mode** or **Mixed Mode**. Select the option that will work best for your environment:
 - **Mixed Mode (for easiest configuration)**: This mode is required if you intend on using a SQL Server account to authenticate Secret Server to your SQL Server instance. We recommend using mixed mode if you are setting up a test or demo environment. Selecting this option will also require you to set a password for the SQL Server system administrator (sa) account. See **Adding a SQL Server User** below for instructions on adding more users.
 - **Windows Mode (recommended for best security)**: This mode prevents SQL Server account authentication. We

recommend using Windows mode for production environments. Whatever user or group assigned will have administrative access to your SQL instance. According to best security practices, limit this number to as few users as possible.

Note: If choosing **Windows Mode** you will also need to [run the IIS application pool as a service account](#) later in the installation process.

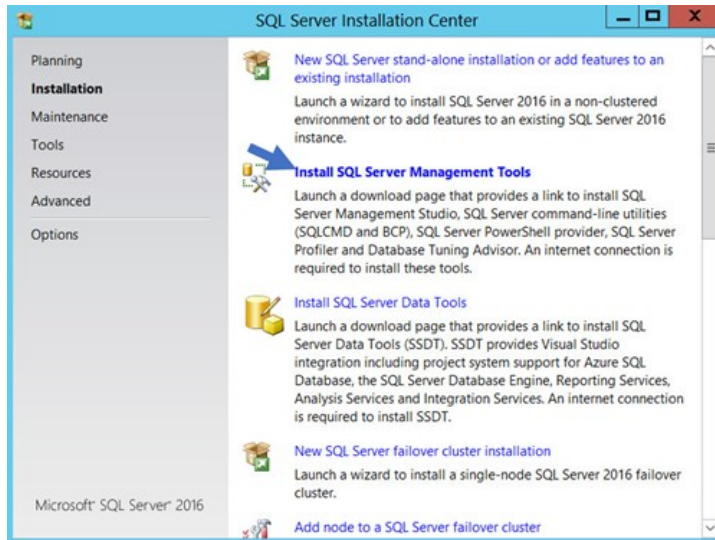
24. You can leave the options in the remaining tabs at their default values or change the file locations in the **Data Directories** and **TempDB** tabs if you wish to store the database and log data in a different drive or directory.
25. Click the **Next > button** twice to bypass the Feature Configuration Rules page. The Ready to Install page appears:



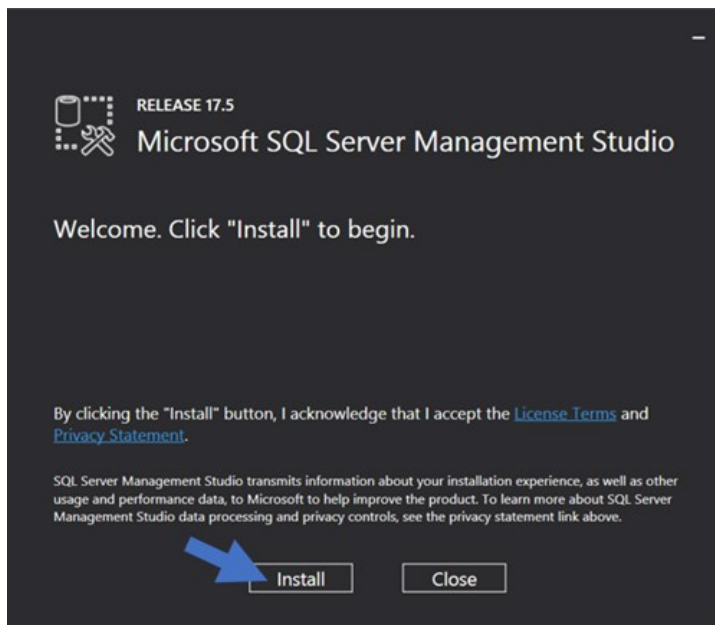
26. Click the **Install** button.
27. Wait for installation to complete. This may take several minutes.
28. Click the **Close** button.

Installing SQL Server Management Studio

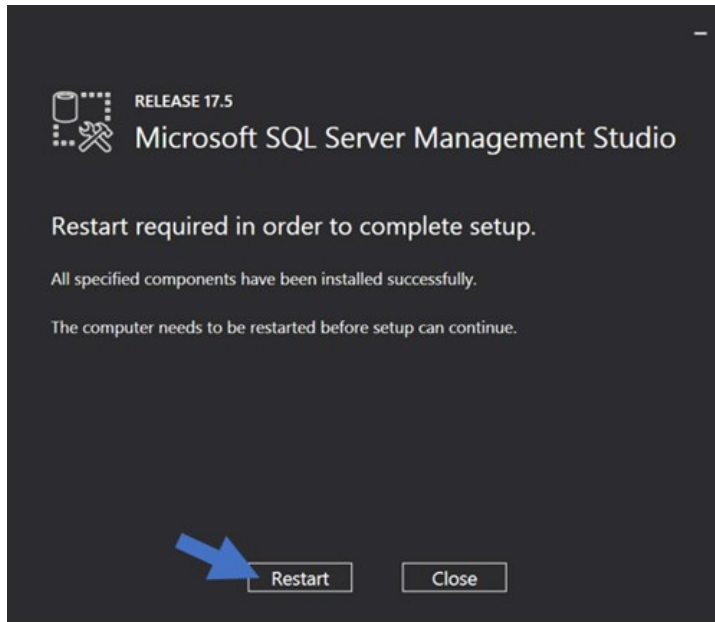
1. In the "SQL Server Installation Center" window, click the **Installation** menu item. The Installation page appears:



2. Click the **Install SQL Server Management Tools** link.
3. Wait for the Web page to load then click the **Download SQL Server Management Studio...** link. A file downloads.
4. Run the downloaded file (varies by browser). The SQL Server Management Studio installer starts.



5. Click the **Install** button.
6. Wait for the installer to complete. This may take several minutes.



7. Click the **Restart** button if prompted. Otherwise, click the **Close** button.
8. Close "SQL Server Installation Center."

Creating the SQL Server Database

To install Secret Server, the Delinea installer creates the SQL database for you if it does not exist and if the user account has permission to create a new database, which requires the dbcreator server role.

If not using the Delinea Installer, use the following steps to create a database manually through SQL Server Management Studio:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server instance.
3. Right click the **Databases** folder and select **New Database...** The New Database page appears.
4. Type a name for your database in the **Database Name** text box.
5. Click the **OK** button.

Adding a SQL Server User

According to security best practices, limit the number of users with access to your SQL database as much as possible. Use the following instructions to add a SQL Server account for Secret Server to use to access the SQL database:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server Database.
3. Expand the **Security** folder.
4. Right-click the **Logins** folder and select **New Login...**
5. Select a method of authentication:

- **SQL Server Authentication:** Use this option to create a new SQL Server account (this requires mixed mode to be enabled). To create the account, enter a new username and password and then deselect the **Enforce Password Policy** check box to prevent the account from expiring.
 - **Windows Authentication:** Use this option to add access to SQL Server for an existing Windows account. To add the account, enter the login name or click **Search** to find the account. It is recommended to use a domain account rather than a local Windows account.
6. Click **User Mapping** in the left menu.
 7. Click to select the check box next to your Secret Server database.
 8. In the **Database Role Membership** window, click to select the **db_owner** check box.
 9. Click the **OK** button.

SQL Server Authentication Configuration

SQL Authentication requires:

- Creating a new SQL account
- Enabling mixed mode
- Enabling named pipes and SQL Browser a non-local SQL Server

Note: For instructions on Creating the SQL account or Installing SQL Server see [Installing and Configuring SQL Server](#) article.

Enabling Mixed Mode

1. Connect to SQL Server in SQL Management Studio.
2. Right click on the instance node and select **Properties**.
3. Go to the **Security** tab.
4. In the **Server Authentication** section, select **SQL Server and Windows Authentication Mode**.
5. Click the **Ok** button.
6. Restart the SQL Server, by right clicking on the instance node and selecting **Restart**.

Note: If your SQL server is running on a separate machine, you need to turn on named pipes and SQL browser to ensure the SQL server can be accessed from an external machine.

Enabling Named Pipes and SQL Browser

1. Open SQL Server Configuration Manager.
2. Click the **SQL Server Network Configuration** node.
3. Select **Protocols for MSSQLSERVER**.
4. Enable the following:
 - Shared memory
 - Named pipes
 - TCP/IP
5. Enable **SQL Browser**.
6. Click to select the **SQL Server Services** node.
7. Right click **SQL Server Browser** and select **Start**.

SQL Server 2014 Express Edition Installation

Overview

Important: Delinea recommends using SQL Express in sandbox or trial environments **only** due to size and performance limitations.

SQL Express is a free edition of SQL and is available for use with Delinea products. The following steps walk you through setup and configuration for SQL Server 2014 Express Edition as an example. For the most up to date resources on installing SQL see [Microsoft SQL Technical Documentation](#) for more information.

At the completion of this article you will have:

- Installed a basic stand-alone instance of SQL Server 2014 Express with the minimum features necessary for SQL Server. This includes SQL Server Management Studio and other tools.
- Created a database in SQL for your Delinea product
- Created a new SQL Server user login for your SQL database

Note: This document uses Delinea's Secret Server product as example in the instructions, but the same steps apply for Privilege Manager advanced installs.

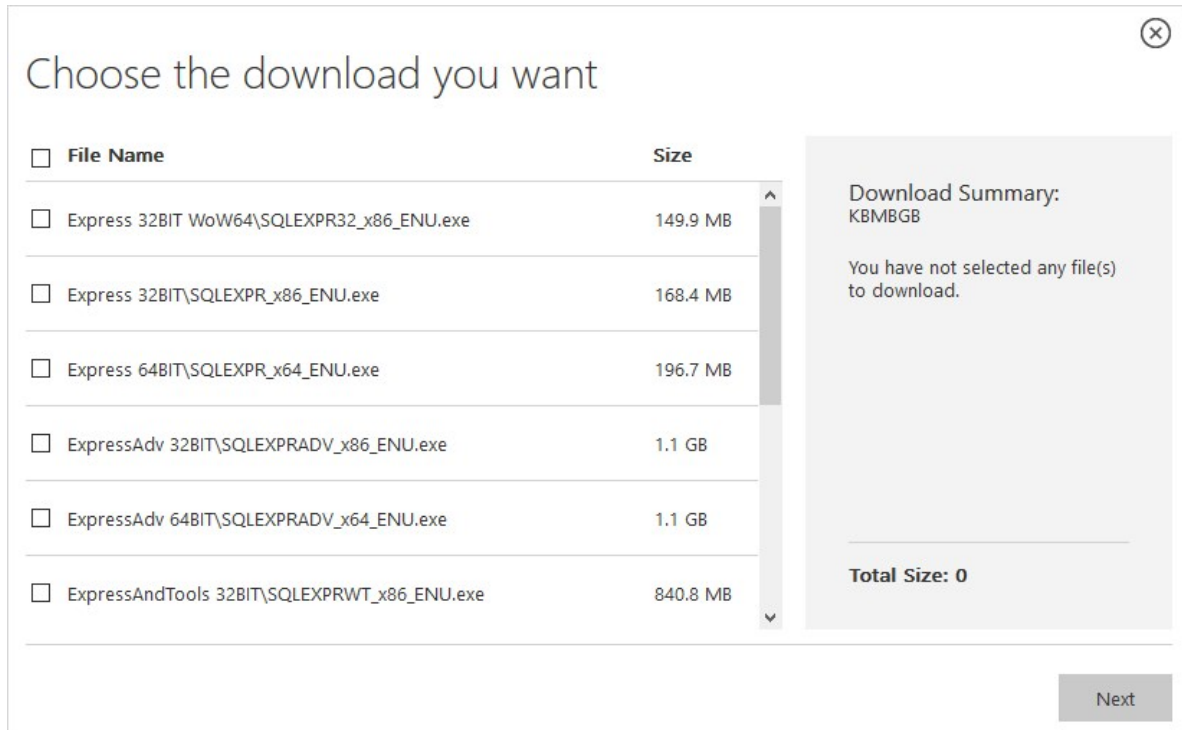
Procedures

Downloading SQL Server Express with Tools

If you plan to use SQL Server Express, we strongly recommend downloading the package that includes **Tools**. This also installs SQL Server Management Studio that allows you to connect to the database directly and gives access to server settings.

Procedure:

1. Go to the [SQL Server 2014 Express download page](#).
2. Click the **Select Language** list box and select **English**.
3. Click the **Download** button. A popup page appears:



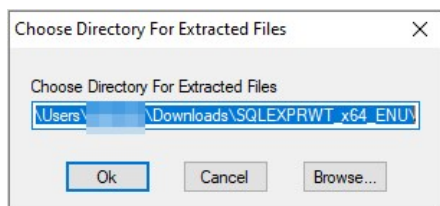
4. Click to select the following check boxes (you may need to scroll down):

- **ExpressAndTools 64BIT\SQLEXPRWT_x64_ENU.exe**
- **MgmtStudio 64BIT\SQLManagementStudio_x64_ENU.exe**

5. Click the **Next** button. SQLEXPRWT_x64_ENU.exe and SQLManagementStudio_x64_ENU.exe* download to your computer.

Installing SQL Server Express 2014

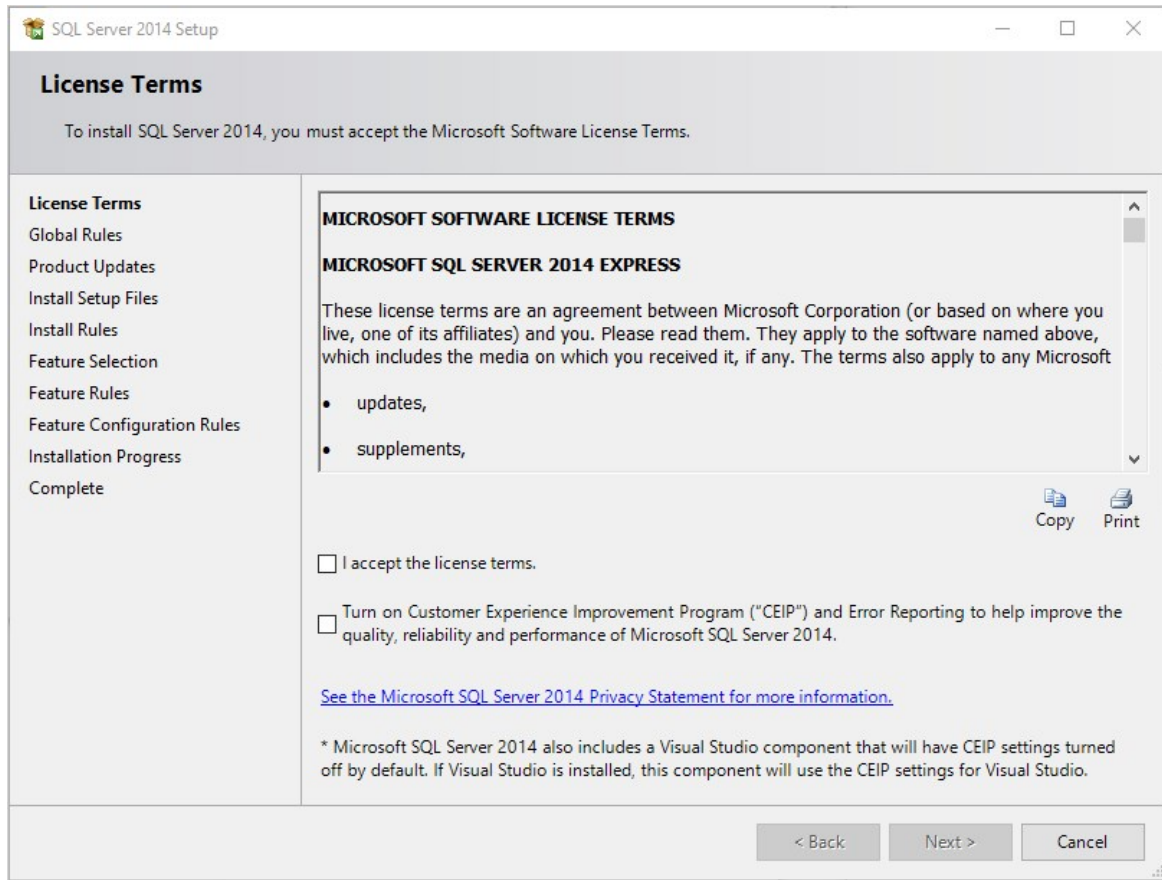
1. If necessary, download and install the latest version of .NET Framework. See [Microsoft .NET Framework 4.8 offline Installer for Windows](#) for the latest version as of when this topic was written. If you have already installed Secret Server, you have already done this.
2. Double click the SQLEXPRWT_x64_ENU.exe you downloaded to run it. The User Account Control appears.
3. Click the **Yes** button. The Choose Directory... dialog box appears:



4. Click the **OK** button. The files are extracted to that location, and the SQL Server Installation Center appears:

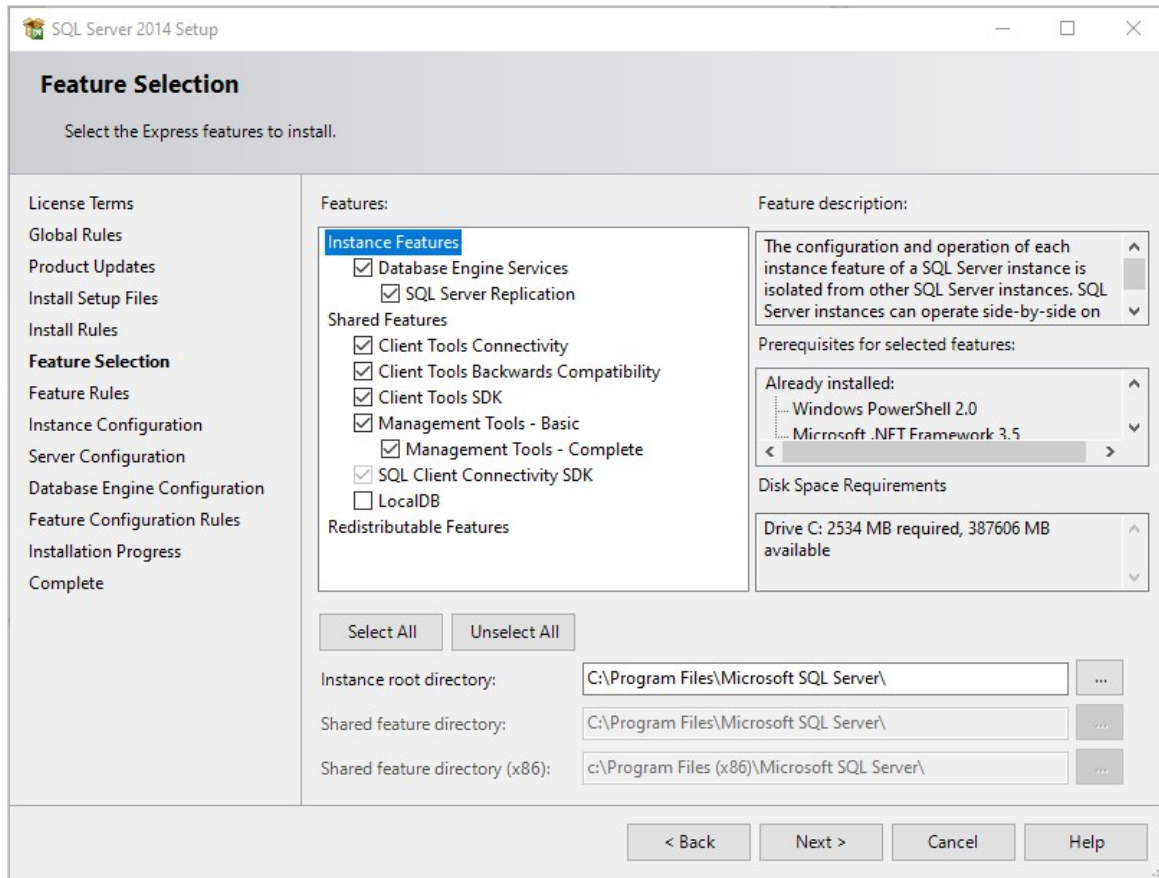


5. Click the **New SQL Server stand-alone...** link. The License Terms wizard page appears:



6. Click to select the **I accept the license terms** check box.

7. Click the **Next >** button. The installation processes four pages with no input from you and stops on the Feature Selection page:

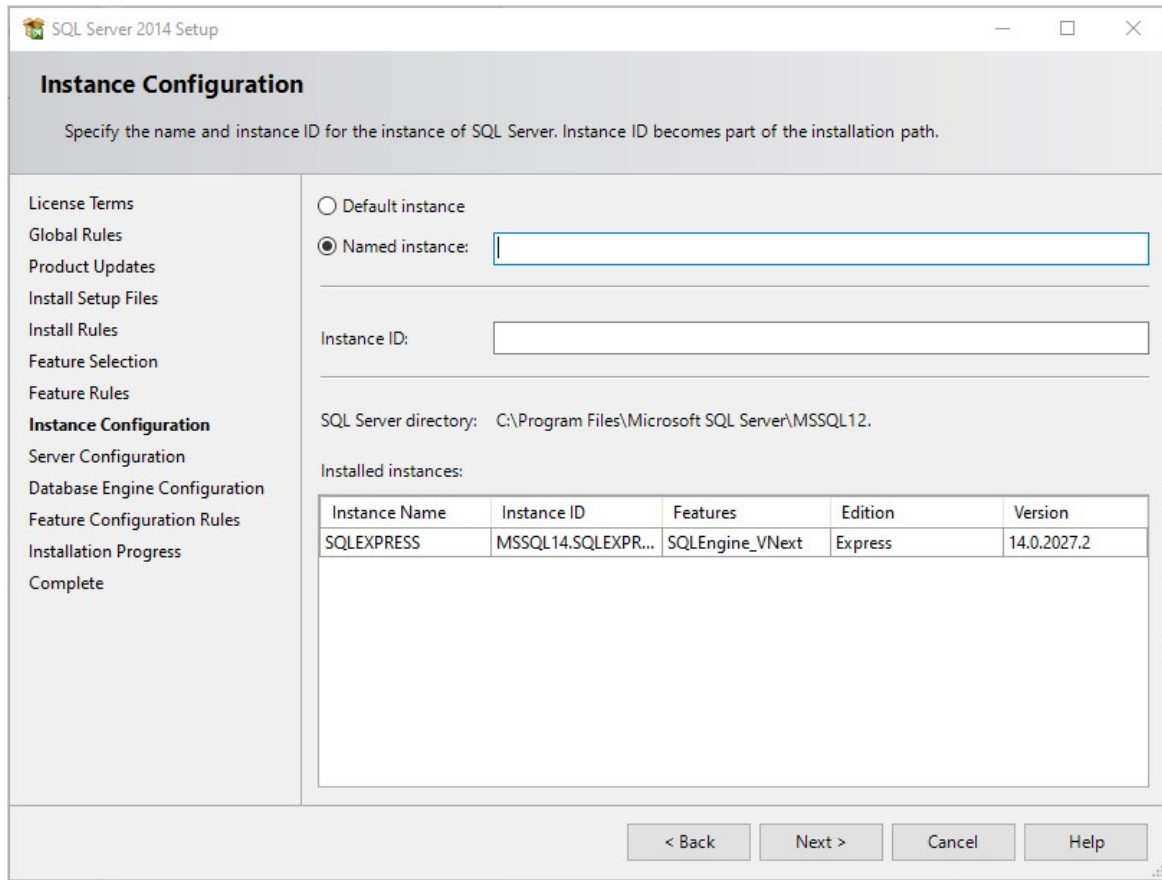


8. Ensure that the **Database Engine Services** and **Management Tools – Basic** check boxes are selected. Leave the others as is.

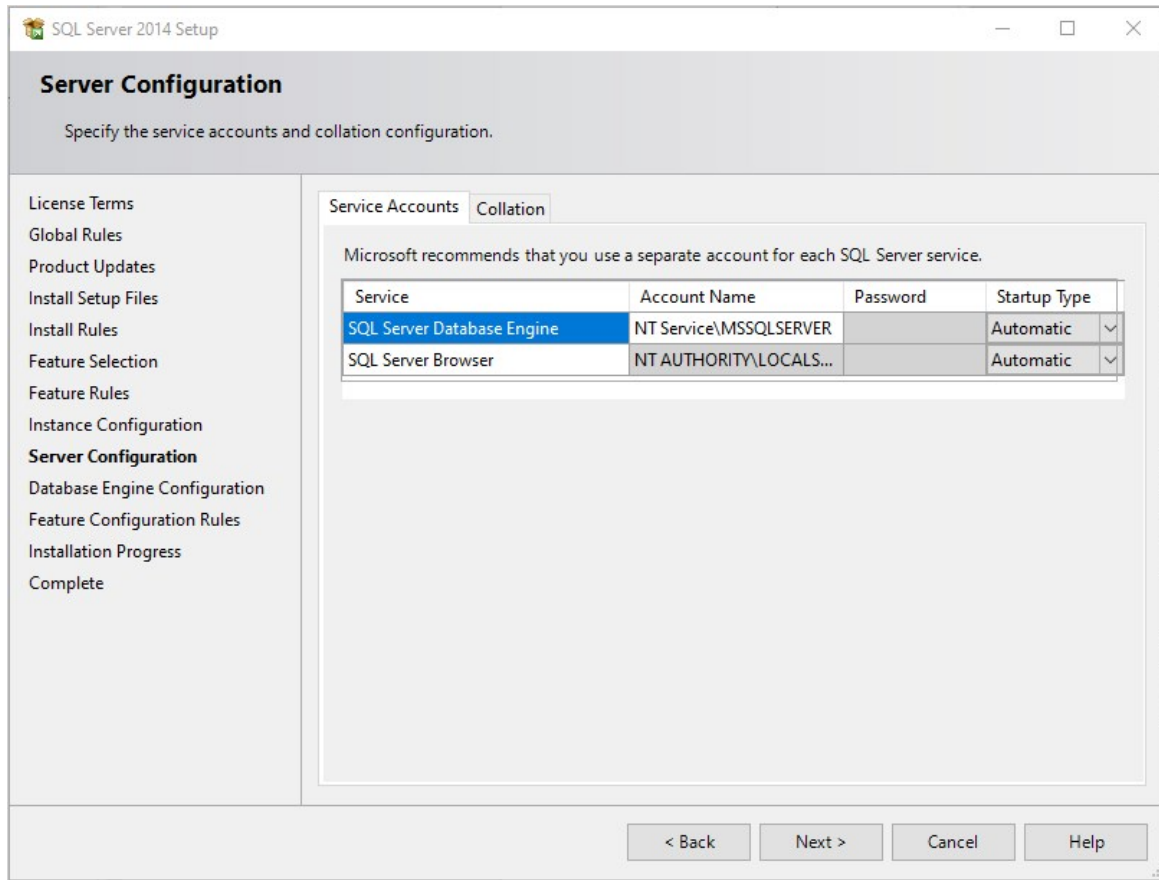
Note: A SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

Note: Management tools include Management Studio support for the database engine and SQL Server Express, SQL Server CLI (SQLCMD), SQL Server PowerShell provider, and the distributed replay administration tool.

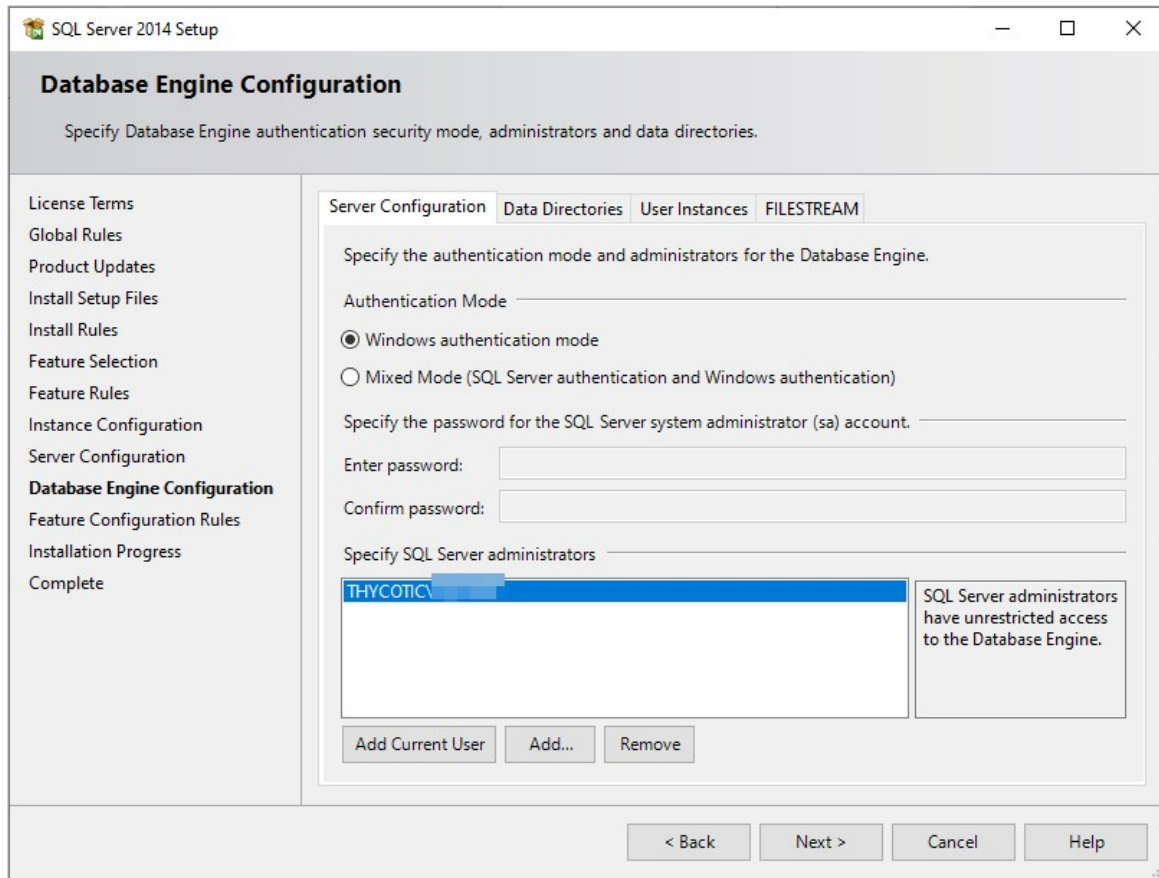
9. Click the **Next >** button. The installation processes one page with no input from you and stops on the Instance Configuration page:



10. **Ether** click to select the **Default instance** selection button, which uses an already present instance called SQLEXPRESS.
11. **Or** type your desired name in the **Named instance** text box.
12. Type your instance ID in the **Instance ID** text box. We chose MySQLInstance. The instance ID will become part of the installation path.
13. Click the **Next >** button. The Server Configuration page appears:

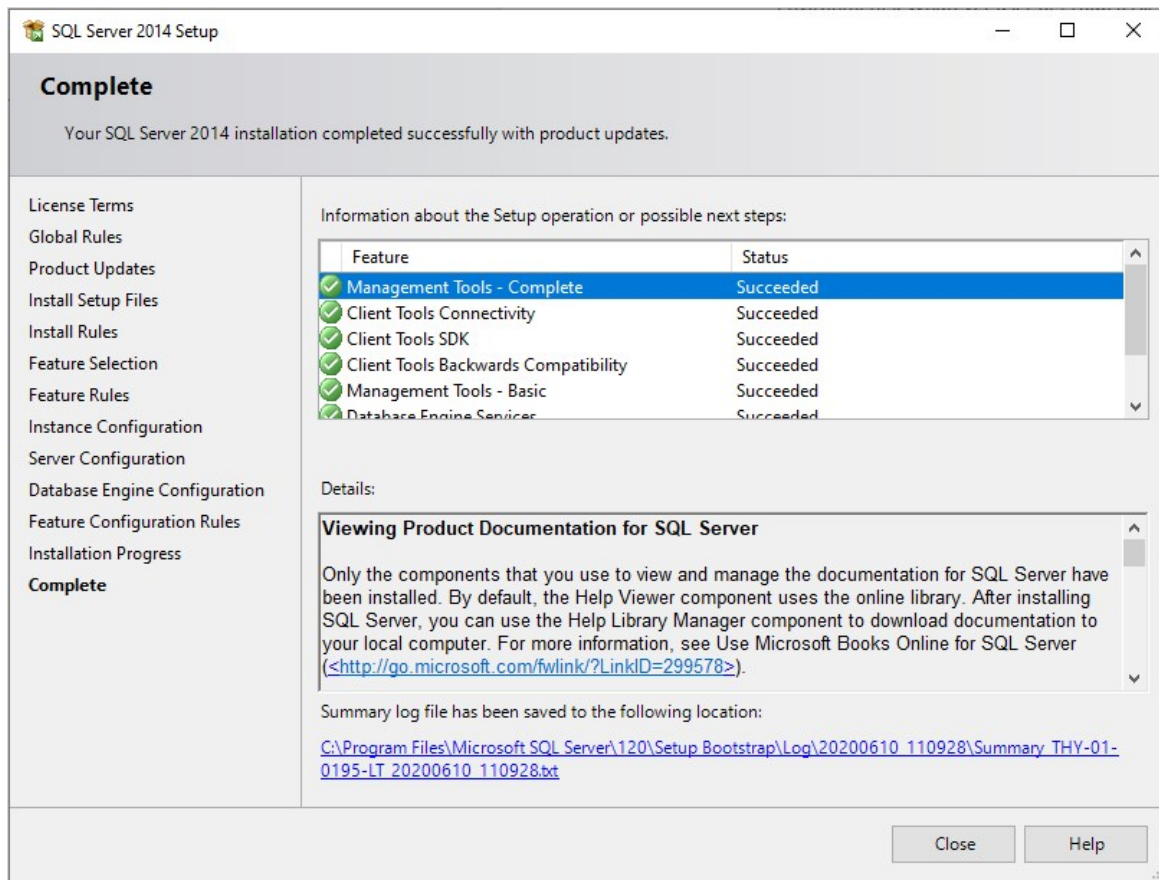


14. Leave the page as is, and click the **Next >** button. The Database Engine Configuration page appears:



15. You have the choice to select either **Windows Authentication Mode** or **Mixed Mode**. Click to select the option that works best for your environment:
 - **Mixed Mode (for easiest configuration)**: This mode is required if you intend on using a SQL Server account to authenticate Secret Server to your SQL Server instance. **We recommend using mixed mode if you are setting up a test or demo environment.** Selecting this option will also require you to set a password for the SQL Server system administrator (sa) account. See **Adding a SQL Server User** below for instructions on adding more users.
 - **Windows Mode (recommended for best security)**: This mode prevents SQL Server account authentication. We recommend using Windows mode for production environments. Whatever user or group assigned will have administrative access to your SQL instance. According to best security practices, limit this number to as few users as possible. Only choose this if you have experience and require this for a specific issue—we do **not** recommend SQL Server Express for production accounts.

Note: If choosing **Windows Mode** you will also need to [run the IIS application pool as a service account](#) later in the installation process.
16. If you selected mixed mode, which you almost certainly did, type your SQL Server system administrator (sa) account password in the **Enter password** and **Confirm password** text boxes. The password must meet Microsoft's definition of a strong password. Click the **Help** button and search for "Database Engine Configuration - Account Provisioning" if you what to find out what that is. A 16 character mixture of lower and uppercase letters and numerals works fine.
17. Your user account should already be shown in the **Specify SQL Server administrators** text box. If not, click the **Add Current User** button.
18. Click the **Next >** button. The Installation Progress page appears and SQL Server Express is installed. This can take awhile. Eventually, the Complete page appears:



19. Click the **Close** button.

Creating the SQL Server Database

To install Secret Server, the Delinea installer creates the SQL database for you if it does not exist and if the user account has permission to create a new database, which requires the dbcreator server role.

If not using the Delinea Installer, use the following steps to create a database manually through SQL Server Management Studio:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server instance.
3. Right click the **Databases** folder and select **New Database...** The New Database page appears.
4. Type a name for your database in the **Database Name** text box.
5. Click the **OK** button.

Adding a SQL Server User

According to security best practices, limit the number of users with access to your SQL database as much as possible. Use the following instructions to add a SQL Server account for Secret Server to use to access the SQL database:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server Database.
3. Expand the **Security** folder.
4. Right-click the **Logins** folder and select **New Login...**
5. Select a method of authentication:
 - o **SQL Server Authentication**: Use this option to create a new SQL Server account (this requires mixed mode to be enabled). To create the account, enter a new username and password and then deselect the **Enforce Password Policy** check box to prevent the account from expiring.
 - o **Windows Authentication**: Use this option to add access to SQL Server for an existing Windows account. To add the account, enter the login name or click **Search** to find the account. It is recommended to use a domain account rather than a local Windows account.
6. Click **User Mapping** in the left menu.
7. Click to select the check box next to your Secret Server database.
8. In the **Database Role Membership** window, click to select the **db_owner** check box.
9. Click the **OK** button.

Understanding Licenses

The Secret Server licensing model allows for scalability and enhanced core functionality in the form of edition enhancements (Professional, Premium Edition) and user packs. Licenses can be purchased for these items as follows:

- **Users:** Secret Server ships with one free license for a single user. Additional user licenses can be purchased through <https://thycotic.com>.
- **Support:** Support licenses allow you to receive technical assistance from the Secret Server support team, and software updates for installed versions of Secret Server. To be eligible for updates, the number of support licenses and user licenses must match.

After installation is complete, you need to enter your licenses using the Getting Started Wizard or from the Licenses Administration page. Entering your licenses allows you to add more users and enables additional features in Secret Server.

Note: For more information, see [Adding, Activating, Converting, and Deleting Licenses](#).

For the Express edition, you have one license to enter. For Professional edition and higher, you have, at minimum:

- An edition license
- A user license
- A support license

If you purchased additional licenses for sites or distributed engines, you may have more licenses to add.

Note: If you have purchased the installed edition of Secret Server but did not purchase support or upgrade protection, you will not have a support license.

Activating Licenses

After entering your license information, you must activate your (non-evaluation) licenses. For more information, see [Adding, Activating, Converting, and Deleting Licenses](#) and the [License Activation FAQ](#).

Licensing Limited Mode

If you fail to activate your licenses, your system is placed in limited mode, which prevents the following actions:

- AD sync
- Creating and editing secrets
- Importing secrets
- Creating and editing secrets
- Web services (mobile applications)

Adding, Activating, Converting, and Deleting Licenses

This section explains how to add and activate Secret Server licenses (both online and offline) how to delete licenses, and how to convert from a trial license.

Note: For more information on understanding Secret Server licensing, see the [Licensing](#) page.

Adding and Activating Secret Server Licenses Online or Offline

1. Log on to Secret Server as an administrator.
2. Go to **Admin > Licenses**. The Licenses page appears:

Admin > Licenses

Licenses Server Activation Audit

- You are currently licensed for 5001 user(s). You currently have 104 enabled user(s).
- Support licenses allow you to get free upgrades for new releases of Secret Server. You must purchase support for as many users as you are licensed.

[Install New License](#)

5 Items

LICENSE NAME	LICENSE KEY	DESCRIPTION	⚙️ ↓
FOR DEVELOPMEN...	822UJ420L7H2	Enterprise Plus Edit...	Delete
For Development U...	417KJ1N0V8S144	Connection Manager	Delete
Not for Resale	WV15B1UC2AF17X	Support (5000 user...	Delete
FOR DEVELOPMEN...	424SE1C24817707	Secret Server (500...	Delete
FOR DEVELOPMEN...	1706L187H151404	Distributed Engine ...	Delete

3. Click the **Install New License** button. The Install New License popup appears:

Install New License

Licenses will have to be manually activated after install.

Entry Type Single Entry Bulk Entry

License Name *

License Key *

- Click to select **Single Entry** in the **Entry Type** selection button.

Note: If you have numerous licenses, you can click the license **Bulk Entry** selection instead. It allows you to paste an entire licensing email or a formatted list of licenses, adding all licenses in a few clicks. For a small number of licenses, especially if you are new to the process, we recommend using single entry, which provides better feedback on what you are doing.

- Type (or paste) the **License Name** and **License Key** for the license that you received from your account manager.
- Click the **Install** button. The License Installed Successfully popup appears:

License Installed Successfully - Activation Required

Install is complete. Activation is required or the system will be in a limited functionality mode until all licenses are activated. In order for the licenses to be activated you must provide additional information.

- If you have another license, click the **Install Another License** button to repeat the process.
- Click the **Continue with Activation** button. The License Activation page appears:

Admin > Licenses > License Activation

Activation Details

The following information is only used for activation purposes.

[What is Activation?](#)

Name *

Email *

Phone Number *

Activation Type

If your server has outbound network access, then select Online, to activate licenses online. Otherwise, select Offline, to manually activate licenses offline.

Activation Type *

[Need to activate on an AirGap Network?](#)

9. Ensure your name, email address, and phone number are present and correct.
10. If you have an internet connection and want to activate **online**:
 1. Click the **Activation Type** dropdown list and select **Online**.
 2. Click the **Activate** button. An Activation Successful popup briefly appears and then disappears, and you are returned to the Licenses page where your new license now appears. The procedure is complete. **Do not do the remaining steps.**
11. If you do not have an internet connection and want to activate **offline**:
 1. Click the **Activation Type** dropdown list and select **Offline**. The Offline Activation section appears:

Offline Activation

If Secret Server is unable to connect to the Activation Server, you will have to manually activate the licenses. Go to the Activation Center and paste the Request value there. Once you get a Response, copy it and paste here.

Request *

```
27daf9b50738fab1ac1
468306255f87c171727
066517bfac443fa0ad8
d34a73d1efb286e0553
230574586fd88398974
6bf2cced5860963a10b
e2a3d61960d19c23a13
f328daa5f90a9e624b5
b7da70c2aad3ac16d4c
```

[Copy to Clipboard](#)

Obtain Response * [Activation Center](#)

Response *

[Paste from Clipboard](#)

[Need to activate on an AirGap Network?](#)

2. Click the **Copy to Clipboard** link to copy the text in the **Request** text box.
3. Click on the **Activation Center** link in the **Obtain Response** section. The License Activation Center page appears. Do not close the Secret Server browser tab.
4. Paste the copied text into the text box.
5. Click the **Activate** button. Activation Successful! appears at the top of the page and the text box now contains the activation confirmation.
6. Copy the entire text box contents.
7. Return to Secret Server License Activation page and paste the response into the **Response** text area.
8. Click the **Activate** button. An Activation Successful popup briefly appears and then disappears, and you are returned to the Licenses page where your new license now appears.

Note: For more information on activating Secret Server licenses, see the [License Activation FAQ](#). Secret Server may be activated on an air gap network for both trials and licensed products. Please let your Account Manager know you will be using Secret Server on an air gap network for more information.

If you receive an error message, please take note of the error code and call the phone number contained in the message.

If an error message persists after successful activation, remove expired and invalid licenses from Secret Server by following the steps

below, under **Deleting Secret Server Licenses**.

If you need help and your Secret Server has a current support license for each user license, please contact our [technical support team](#).

Note: For more information on Secret Server licensing and license activation, see [Licensing](#) and the [License Activation FAQ](#).

Converting Evaluation Licenses

If you had evaluation licenses initially and you recently purchased Secret Server, you need to remove all evaluation licenses before you install your purchased licenses. Follow the steps below, under **Deleting Secret Server Licenses**.

Deleting Secret Server Licenses

1. Log on to Secret Server as an administrator.
2. Go to **Admin > Licenses**.
3. In the **Licenses** dialog, click the **License Name** of the license you want to remove.
4. Click **Delete**. The license information will remain available to you from your account at [Delinea.force.com](https://delinea.force.com).
5. Click **OK**.
6. Verify that the selected license key has been removed from the list.

License Activation FAQ

What happens if we find that we had more named users than licenses after activation? Will the account lock us out?

The user licenses are per named individual. You can simply disable any excess users so you are within your license count—these users can be re-enabled later and all audit log information is kept.

Why is license activation required?

Activation of license keys is standard practice in the software industry. We try to focus exclusively on implementing customer requests but occasionally we must spend time on licensing especially as Secret Server goes into new geographical markets.

Is there a grace period before we must activate?

Existing customers have 30 days to activate their licenses after upgrading. New licenses must be activated immediately on adding them to Secret Server. Evaluation licenses do not require activation.

How is license activation implemented in Secret Server? Activation is per license and Web server (the combination of the two). Therefore, even if a Web server was already activated, if you bring up a new Web server, it also needs activation. The activation process gathers the name, email, and phone number of the individual activating, for internal purposes only. No other personal information is sent to Delinea.

What will happen if we don't activate our licenses?

Secret Server will go into limited mode if you do not activate your licenses. Limited mode allows you to view passwords, but many other features are disabled such as creating secrets, editing secrets, changing permissions, and using Web services. Activate your licenses to get out of limited mode.

We have several license keys. Do we need to activate each license key individually?

No, the license activation process will activate all license keys that are currently added to your Secret Server. However, additional license key for distributed engine may need to be activated individually if you receive the key after the other licenses.

What if we have been using our license keys on more than one instance of Secret Server? Secret Server software licenses (user, professional, enterprise, or enterprise plus edition licenses) may only be used on a **single production instance** of Secret Server. You may use your same licenses for a single testing (non-production) environment. If you have used your licenses on multiple production instances of Secret Server, please [contact us](#).

What information is collected and sent during license activation?

License Activation is required for each web server that will be running Secret Server. The request and the response to or from [thycotic.com](#) are encrypted for added security.

The following information is sent to [thycotic.com](#) when you activate:

- Name (user entered)
- Phone number (user entered)
- Email (user entered)
- All licenses (license name, license key)
- Hardware hash of each Web server

The following information is one-way hashed or omitted before it is sent to ensure it does not reveal any identifiable hardware information.

- Secret Server version
- An encrypted value to identify the instance
- Secret data or the `encryption.config` file (both omitted).
- The data that is gathered for is for contacting you if there is a licensing issue. Delinea will not sell or distribute the information provided

during activation. The only information available to Delinea staff is the contact information for technical support and customer service.

Our Secret Server does not have outbound access to the Delinea.com Web site. Can activation be done while offline?

Yes, there is an offline option for activating licenses. See [Adding, Activating, and Deleting Licenses](#).

If we have trouble activating our licenses, what should we do?

1. If your Secret Server is currently supported, with an equal number of current support licenses and user licenses, our technical support team can help you. Please [contact us](#).
2. If an error message persists after successful activation, remove expired or invalid licenses from Secret Server by clicking the license name and then deleting it (the license information will remain available to you from your account at [Delinea.force.com](#)).

My Server is a VM that moves to different hardware often. Will this cause me to need to reactivate over and over?

You do not need to reactivate over and over. When you activate, you can use Secret Server for a year without needing to reactivate, regardless VM hardware changes. However, if your machine name changes as well as your hardware, you will need to reactivate. If you are using a version older than 7.8.000000, you must reactivate when the VM moves.

11.2.000003 (GA Version)

Note: Calculated on 2022-09-27 14:50:40-04:00

DelineaSetup.exe:

- SHA1 = 0a3f0ee7a735b4257b0b9bf793a28087b398050b
- SHA256 = d69e546d445e44833d9dc87f3805d2273dd7f616ed132d2ff410f130d5872a3a

Version_11_2_000003.zip:

- SHA1 = 849867ce2447c88ca81df90428ed3ac1878dfd4e
- SHA256 = 0f17f998c8e5c935081f5c0430bcddbe538fff6e77824ee46d9ea01b5bb2de3

Earlier Versions

11.2.000002 (GA Version)

Note: Calculated on 2022-07-06 17:25:22-04:00

DelineaSetup.exe:

- SHA1 = 3d1ea696aeb08f4c15339ab15e3cbb7f841f8866
- SHA256 = 0e91cd6f34a8667cf876445ea968540dfd65262fa7250b0f7ed36cb510f45c33

Version_11_2_000002.zip:

- SHA1 = 65e0bd2e1d99ff1350dc35114c4b3c76522af41a
- SHA256 = 05a4b735ecaa5329e110fb773f29199a582d3a43f0b63a779c6dfdf7e889556f

11.2.000000 (EA Version)

Note: Calculated on 2022-05-04 22:11:30-04:00

Version_11_2_000000.zip:

- SHA1 = bcb9f7a5dd0effe825d6cc1054bf63be9ee1a3f5
- SHA256 = 8246742e7b5d0c8b8cd601cbe970be8ce1f2927daca1ac4eefc84aff83898b3c

11.1.000012 (Security Release)

Note: Calculated on 2022-05-05 10:20:39-04:00

ThycoticSetup.exe:

- SHA1 = a5d9b6f6b99234d91f70be4c23025583a4dcb32e
- SHA256 = 4d2a8fe7420ef27645971df3391e1a6638f71448bcdcb0589ef81b6c79e70d1e

Version_11_1_000012.zip:

- SHA1 = eb2c75a1c616fea2df14ea32b06268107b4bec16
- SHA256 = 4952a541814d2f2aba439c620e550e4c7df28b6623a63edb7f70482fc30eb600

11.1.000007 (GA Version)

Note: Calculated on 2022-01-24 16:28:53-05:00

ThycoticSetup.exe:

- SHA1 = 947d48da08bf6ab63381132fd3719dbcc35dffc2
- SHA256 = d1664e7ff2e4961f251dc58ca12918335658ed7f27f7752b9f465af0deb83d7a

Version_11_1_000007.zip:

- SHA1 = 36038fd831ac038ea8aa0082c0032f98df21e9f4
- SHA256 = b3cc1b1884f67c0bf382b8b03b2e57ce9b9b2d5792ee607aa37b44770534b84c

11.1.000006

Note: Calculated on 2022-01-14 11:52:53-05:00

ThycoticSetup.exe:

- SHA1 = a23e086c85c0b526c8b824221485b6906b07d290
- SHA256 = 89deea52f03f9854a88b909974eb625db3815175a0c84110853f23d9db9f26b8

Version_11_1_000006.zip:

- SHA1 = 8e8bb603dc6441c0808bef8cd5a435f8aa95610d
- SHA256 = f063c0a57513f08332d40457996c9cf0bb203a0425f344fc2d322b6002d07c8a

11.1.000000 (EA Version)

Note: Calculated on 2021-11-29 14:15:45-05:00.

ThycoticSetup.exe:

- SHA1 = e719e40d932c9acec3b5f35c3243074c4ee8aaba
- SHA256 = 90e9abe510f9cd30e5e3804e88875b66e05934606bdcbae960019cd20813c542

Version_11_1_000000.zip:

- SHA1 = 87c84f7141285ecbaf8d68655feea6a31b8c2247
- SHA256 = 2bbbebec5fe8c66125a9baf046141542368ef40edaf79ad28efe5667d6fbc0b5

11.0.000008

Note: Calculated on 2021-12-13 16:06:20-05:00.

ThycoticSetup.exe:

- SHA1 = 1be211f3a870b6d53f1a9ac2dc069dac469f609
- SHA256 = 5aae7c1e697f3ba1f01d167b33ea93e5300e51b1d3b6ee1e27303c1fc8cbee8c

Version_11_0_000008.zip:

- SHA1 = 661c39feadb0bb98cf80d9c0fee4d25f41062f1f
- SHA256 = 9ee6ea74d841109c34ef14815dca77586d58e6ef64375295d82974c211bbbc38

11.0.000007

Note: Calculated on 2021-09-13 13:23:59-04:00.

ThycoticSetup.exe:

- SHA1 = 8cac67d4e41116249de860059835d9c26732b2d0
- SHA256 = 2f151eb6689e782043a03ecc0e0e4930ef64e14c9155f4ee81d1365815cb403b

Version_11_0_000007.zip:

- SHA1 = 71a339f2f74565091d2e2f50056dda2128771fbc
- SHA256 = 69e1bdc20a6493060fc3e94243973e40b467f9e12dd4b2bc83dcd78455da5cc4

11.0.000006

Note: Calculated on 2021-08-21 16:08:39-04:00.

ThycoticSetup.exe:

- SHA1 = 216814721746f98a0887fd15f573397a5d6e58af
- SHA256 = fd16aaa367248eb79339bfefdc7705ff6a4c60574cb9bba161ed6c0e1c659729

Version_11_0_000006.zip:

- SHA1 = b2a46956a9c5a56923b380893b3922230ba26356
- SHA256 = c9bc00a82475911d9d2d38de5007fb60102e0ef1461e443e3a41b76455e4d93b

11.0.000005

Note: Calculated on 2021-08-16 13:39:22-04:00.

ThycoticSetup.exe:

- SHA1 = 1d4cd40964ec54281f75bea7fc49cea26449899c
- SHA256 = 3eb79053aed18462d2579e73726b902108e6bb30749b7a48fa7582781b148817

Version_11_0_000000.zip:

- SHA1 = 982f5cfd74c5f231fa620cbaed60de3a7b8362ae
- SHA256 = fcb694990f5649e89a99d25290829efa9f65d95aae13065a6f21ea66d1d4616e

10.9.000064

Note: Calculated on 2021-06-02 19:32:10-04:00.

ThycoticSetup.exe:

- SHA1 = 340eae40a405e446dfde698c469316bba1a331
- SHA256 = cee3168cfe5c745e79c6d0509385c4d06547acd640e5a5eaa889dc31cc6f73ce

Version_10_9_000064.zip:

- SHA1 = aec38df961ccc88650188ee4158c15b6a7a4a107
- SHA256 = b896ef24904beb2414edcfc259ef14c9eadb2fdd442289a2781d3e1ac8fa411b

Below are the hashes for the step-upgrade version. When upgrading from a version prior to 8.4.000004, the upgrade process will upgrade Secret Server to 8.4.000004.

If Secret Server is version 8.4.000004 to 9.1.000000 the upgrade process will upgrade to 9.1.000001 and from there Secret Server can be upgraded to the latest release.

10.9.000063

Note: Calculated on 2021-03-22 16:43:23-04:00.

ThycoticSetup.exe:

- SHA1 = 326f59abdc61976b9e7d589247fd94952657458b
- SHA256 = b6abecd7be6016d7524a4a57b5b5e0cf488305df085eb273aeefa931a697d3d5

Version_10_9_000063.zip:

- SHA1 = a6da7bdee1d5018f050b10317299c8993a6dd14a
- SHA256 = 853de55ba231e37d2a8787e253cfd1037c9c8d3a89fa697cdd425a87c258481d

10.9.000033

Note: Calculated on 2021-02-22 14:14:09-05:00.

Downloaded February 24th 2021 or Later

The following are the download hashes for the current version installation files of Secret Server. This download hash is for the combined Secret Server and Privilege Manager version release on February 24, 2021.

Note: Calculated on 2021-02-22 14:14:09-05:00.

ThycoticSetup.exe:

- SHA1 = 3df5f61ceb50a5f1e84d1f76a5e0d5db05f99bc0
- SHA256 = e67f06e0052a826196438c9c09178aa4d2391f731d8307ff21f68722fbbe868c

Version_10_9_000033.zip:

- SHA1 = e9dda8a53e301a950f697e628d22d1aea485768b
- SHA256 = ba85391275bbcc1e1e9ce2a53727de660b358a6fad35b61861cff1ab1d68e948

Downloaded Before February 24th 2021

For ThycoticSetup.exe downloaded before February 24th 2021, use the below hash for the installer:

Note: Calculated on 2020-12-12 16:07:09-05:00.

ThycoticSetup.exe:

- SHA1 = d64257d3489d31e7af8554e2dd69d1b2dd8ca181
- SHA256 = bd58ac1a0946ed4b3b8b7d5bf91eb5fe701b67a21ffd5532315dafbd6cce09e6

Version_10_9_000033.zip:

- SHA1 = e9dda8a53e301a950f697e628d22d1aea485768b
- SHA256 = ba85391275bbcc1e1e9ce2a53727de660b358a6fad35b61861cff1ab1d68e948

10.9.000005

The following are the download hashes for the first half of the combined "step upgrade" installation files of Secret Server.

ThycoticSetup.exe:

- SHA1 = 47a1e5eafa8797f9c6012a2b54efaa09e8ff0649
- SHA256 = 6ccb2e6a9b695a432a006a143ad7fa672a8191c5aba9556eb4cff59180d1f82c

Version_10_9_000005.zip:

- SHA1 = e1f8f6dd8e8e43f81d4b30d9a6cabafe08c46023
- SHA256 = 0e31766c54af67944e0ef16f8ad6512672640d0a5988ff7e03b99a40a1525de5

10.9.000032

ThycoticSetup.exe:

- SHA1 = 332006c38327bf5f48e4d687239fad3f7798849b
- SHA256 = 173f3a796412684808c256341995c50713e6b61ddef4510f9471748d9fa29b98

Version_10_9_000032.zip:

- SHA1 = d03cad17517e5b5054b93086d992d34909182c39
- SHA256 = 4d5365b93670d33f3c9b9f02573c0b53e971943a1e35adf1950750e69821ee41

10.9.000002

The following are the download hashes for the combined installation files of Secret Server and Privilege Manager as of Secret Server version 10.9.000002.

ThycoticSetup.exe:

- SHA1 = fdd1667fa445cdf2b90a74fb61e3846b76e53b2
- SHA256 = 3697f03c44a9d0439ed368ee95e4a5b5ce96c092ca9ed76f1d9ef0f970b63d09

Version_10_9_000002.zip:

- SHA1 = 61f19f05aa32597aa93c9590c7fa254129ba2577
- SHA256 = f8d737cf3e60e81c89b6edee0f8d14236e90c831eda3814a353f0744a5e61c0c

10.9.000000

The following are the download hashes for the combined installation files of Secret Server and Privilege Manager as of Secret Server version 10.9.000000.

ThycoticSetup.exe:

- SHA1 = e507431a5312bc732501227726b35d3979c8036c
- SHA256 = 6947cdda490735bb7e876bf7ae49da417b083788da4711aefaeffc56e551798

Version_10_9_000000.zip:

- SHA1 = f260f2ca5974d83770087f4f4c16744fe37b5c10
- SHA256 = aaf98a57062c8e11600e5767ee08dd5af2aae34dc37aa574b7a44d0be2011f6d

10.8.000004

The following are the download hashes for the combined installation files of Secret Server and Privilege Manager as of Secret Server version 10.8.000004.

ThycoticSetup.exe::

- SHA1 = 4e1afea043e7c78005b39cf9eb1c97afcd7239de
- SHA256 = 6507c104d7224200552406c7ced0ce65dfcda9f76e564b1c9c257fe22c0162bc

Version_10_8_000004.zip:

- SHA1 = ac498f2f8de667e860c3eecf8ad324c5b7dd2019
- SHA256 = c65aec870e6b67df1faf8f20cf892eb8f43d9d7e2f936c2bf817545b0055022f

10.8.000000

The following are the download hashes for the combined installation files of Secret Server and Privilege Manager as of Secret Server version 10.7.000059.

ThycoticSetup.exe:

- SHA1 = eda25d63b9538ec0cffaf62b7ed27c594ba3b606
- SHA256 = 74b897a5ce011a4d0e32d714dcb1d4f6aa86c4d8453be84798c13f64a89a04ec

Version_10_8_000000.zip:

- SHA1 = 53eb1a460db1476ac96a1df4b762f0525afbc3de
- SHA256 = 6560528ac2270faf721c52671f6252d8e7e162ee79a0077480a320e63552544e

10.7.000059

The following are the download hashes for the combined installation files of Secret Server and Privilege Manager as of Secret Server version 10.7.000059.

ThycoticSetup.exe:

- SHA1 = a43267c43a5f5752830a705cb473b92ff298b5e9
- SHA256 = a09954b86c04ce1dfe2ef94e0e5dcacf350cc2fac6cbe98e65ddec8f20c65968

Version_10_7_000059.zip:

- SHA1 = cc94fc953504d9b9ec854af5b94ae3d58e69cd33
- SHA256 = 0d99abd35ea1e6bda96018613071b1ab8ad96e2bdf10d645e72587a631f4669e

9.1.000001

Version_9_1_000001.zip:

- SHA1: F9E3B771A71AAED1CD2D31B465690A200471B34F
- SHA256: CC1F977FE748E5925AA694240E4C66814393DCC64F37CE920966C45F5E999D2B

8.4.000004

Version_8_4_000004.zip:

- SHA1: f67f2331cf84ac1716a7673a0c8293add8d9270f
- SHA256: 17515f5ab81f9f0537fd74ff4edaa89e8af6486439219d2cc6f8538082f15a37

Important: Please read the notes at the bottom of this article.

Minimum Requirements for Basic Deployments

2 CPU Cores	2 CPU Cores
4 GB RAM	4 GB RAM
25 GB Disk Space	50 GB Disk Space
Windows Server 2012	Windows Server 2012
IIS 7 or newer (64-bit applications only)	SQL Server 2012-2019
.NET 4.8 or newer	Collation SQL_Latin1_General_CP1_CI_AS

Recommended Requirements for Basic Deployments

Note: Environments budgeting for over 10,000 secrets require a scoping call with a Delinea engineer

4 CPU Cores	4 CPU Cores
16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2012-2022	Windows Server 2012-2022
IIS 7 or newer (64-bit applications only)	SQL Server 2012-2019
.NET 4.8 or newer	Collation SQL_Latin1_General_CP1_CI_AS

Note: Windows Server 2022 is supported by Secret Server 11.0 or later. It may work with earlier versions, but that has not been officially confirmed.

Minimum Requirements for Advanced Deployments

Recommended for organizations deploying discovery, session recording, or increased numbers of distributed engines:

Note: Also see feature-specific guides listed below.

8 CPU Cores	8 CPU Cores
-------------	-------------

16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2012-2022	Windows Server 2012-2022
IIS 8 or newer (64-bit applications only)	SQL Server 2012-2019
.NET 4.8 or newer	Collation SQL_Latin1_General_CP1_CI_AS

Note: Windows Server 2022 is supported by Secret Server 11.0 or later. It may work with earlier versions, but that has not been officially confirmed.

4 CPU Cores	4 CPU Cores
4 GB RAM	4 GB RAM
25 GB Disk Space	40 GB Disk Space

[Unexpected Link Text](#)

Note: Further adjustments to system requirements for both RabbitMQ and distributed engines are at the discretion of Delinea Professional Services engineers.

Recommended Requirements for Specific Features

[Session Recording Requirements: Basic and Advanced](#)

[Ports Used By Secret Server](#)

Notes

Important: This section contains caveats potentially having a significant effect on any installation.

- The use of Server Core for Secret Server installations is not recommended; All QA and testing is based non-core versions of Windows Server.
- To comply with Microsoft licensing requirements, there is an additional constraint on which Microsoft Windows Server version you can use as the RDS server for session connector.

If you use Microsoft User Client Access Licenses (CALs), you cannot use Windows Server 2019. You must use Windows Server 2012 or 2016. If you use Microsoft Device CALs, you can use any supported version of Windows Server.

- Secret Server requires that Microsoft SQL Server and its database be set to the collation SQL_Latin1_General_CP1_CI_AS. See [Microsoft SQL collation requirements](#) and check your server collation settings before upgrading.
- System Requirements apply to both physical and virtual machines.
- For best performance, we recommend using dedicated (clean) servers for hosting Delinea products.

- If .NET or IIS features are not already installed on the web server, the Delinea Installer will add and configure them automatically.
- If SQL is not already installed on a database server, the Delinea installer can setup SQL Express on the web server; however, SQL Express is intended for trials and sandbox environments **only**. Though Delinea will support SQL Express, users will likely experience performance issues due to the memory and product limitations. If experiencing performance issues while using SQL Express, we highly recommend upgrading to SQL Server prior to contacting Delinea Support.
- A link to Microsoft documentation on the use and limitations of SQL Express can be found at: <https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017>.
- Installing Secret Server with Azure SQL: Currently, we do not recommend using Secret Server with Azure SQL when the Web host and the Azure SQL instance are in different datacenters. According to Microsoft, applications, such as Secret Server, that use frequent, high-volume, ad hoc queries use substantial response time on network communication between the application and Azure SQL database tiers. Thus, network latency with many data access operations across datacenters can become an issue.
- Unsupported Web Servers: Small Business Server (SBS), The Essentials Edition, Any client OS, domain controllers, SharePoint servers.
- Secret Server Cloud requires an on-premise machine to use a distributed engine.
- SQL launchers do not support SSMS 18.0 or higher.
- Discovery scanning for Windows Server 2016 scheduled tasks requires that either the Secret Server node or the distributed engine that is executing the scan must run on Windows Server 2016 or later. This is due to changes in Windows Server 2016 API used for scheduled task dependency scans.
- AWS RDS: Currently, we do not recommend using Secret Server with AWS Relational Database Service when the Web host and the SQL instance are in different datacenters. Applications, such as Secret Server, that use frequent, high-volume, ad hoc queries depend on fast network communication response time between the application and SQL database. Thus, network latency with many data access operations across datacenters can become an issue.
- Secret Server requires the application pool to have the "load user profile" setting enabled. Secret Server will report a critical alert to notify admins if this setting is not enabled.
- Supported Web browsers:
 - Google Chrome
 - Mozilla Firefox
 - Microsoft Edge. Edge Chromium only. Legacy Microsoft Edge is not supported.
 - Safari

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

To upgrade Secret Server, you need valid support licenses. To renew your support, please use our [online Web form](#) or contact sales. Once you have valid support licenses, see [Upgrading with Web Clustering](#) to upgrade.

Ensuring the Upgrade Is Secure

We take the following measure to ensure the upgrade is secure:

Secret Server on-Premises upgrades are packaged in a wrapper zip file for delivery. These upgrade zip files are named with the release version, such as `Version_11_1_000006.zip`. Each wrapper zip file contains two files, a security catalog file named `hashes.crt` and yet another zip file named `ss_update.zip`.

The `ss_update.zip` file holds the changes for the new Secret Server version (database and file changes). The `hashes.crt` file is a signed security catalog containing the hash of the `ss_update.zip` file, which is used to ensure the authenticity and integrity of the `ss_update.zip` file.

A hash is a long string of characters that represents a unique digital "fingerprint" of any file the hash function is run on. Any change to a hashed file causes future hashes of the altered file to differ from the original hash, proving the file is different.

When the upgrader processes `ss_update.zip`, the following occurs:

1. The `hashes.crt` file is validated to ensure that its digital signature is the original produced by Delinea.
2. The same hash function that created the original hash is run on the `ss_update.zip` file, creating a new hash.
3. The original hash is compared to the new hash, ensuring the two match and `ss_update.zip` has not been tampered with.
4. The original wrapper zip file is inspected to ensure it does not contain any unexpected, likely malicious, files.

As a result of this inspection process, the upgrader is positive both `ss_update.zip` and `hashes.crt` are genuine and there are no foreign files present.

Minimizing Upgrade Downtime

Introduction

Large enterprise Secret Server customers with clustered environments often have a strong interest in minimizing downtime during their upgrade process. This document details our recommendations for accomplishing that.

Note: This strategy may require close coordination between networking, server administration, and SQL DBA teams. We recommend they are available at the same time during the upgrade to minimize downtime. This procedure works best for those upgrading from the prior most recent version of Secret Server.

We recommend that you have a QA or test environment mirroring your production environment and that you first run this procedure through that environment to ensure the desired results occur, prior to attempting this in a production environment. As with any third-party application upgrade, we strongly recommend taking snapshots or virtual machine backups of your web, Secret Server distributed engine, and database servers so you can easily revert to a pre-upgrade state if issue arise during the upgrade.

Note: Distributed engines auto-upgrade as part of the upgrade process to the latest release. It is possible that you will have up to 10 minutes of downtime while the engines upgrade, regardless of this procedure.

Note: Customers using IWA and upgrading from 10.6 or lower to 10.6 or greater need to follow the steps in [Configuring Integrated Windows Authentication](#) to configure their distributed engines.

Procedures

Load Balanced Configuration Upgrade

Prerequisites

The upgrade procedure requires that you do these steps outside of the Secret Server install.

- Download the upgrade package (step 1)
- Backup your database (step 2)
- Obtain the database upgrade script from support (step 4)
- Backup any customized web.config or web-appsettings.config files (step 10)

Procedure

1. Download the latest version of Secret Server from the [Delinea Support Website](#).
2. Perform a full backup of your Secret Server database using the preferred backup method used by your company. For a quicker recovery procedure in case of disaster, we recommend creating a local SQL backup, engaging your SQL DBA team as needed. If you use an AlwaysOn configuration, perform the backup from your primary node. If desired, you can choose the option to do a "copy only" backup to avoid interrupting any log truncation performed by your enterprise backup tool.
3. Restore the database onto a separate SQL server or separate instance within your environment. This restored backup is used to test the upgrade process. For this instruction, we call this server the "Test SQL Server."

Note: Depending on your circumstances, you might want to provision a standalone SQL server to accommodate this need in the future.

4. Request a database upgrade script from the Delinea Support team. You must provide them the current **exact** version of software you are on. The script can **only** be provided by Delinea. You can be request it prior to your upgrade.
5. Run the upgrade script on the Test SQL Server to verify the upgrade script runs without errors. If the upgrade script completes without errors, you can proceed.

6. Remove the database from the Test SQL Server. This may require your restarting SQL Server Engine services prior to removal.
7. Choose one of the web servers in a load balancer pool. For this instruction, we call this the "Target Web Server A." The pool has three servers total (Target Web Server A, B, and C).
8. Stop IIS on Target Web Server A. That server will appear offline in the pool list on your load balancer configuration application. For now, leave the other target web servers as is.
9. Temporarily remove Target Web Server A from the load balancer pool using your load balancer configuration application. Leave the server itself running. For example, on a F5 Big-IP load balancer:
 1. Click Local Traffic.
 2. Click Pools.
 3. Click to select the desired pool.
 4. Click Advanced in the Configuration section.
 5. Ensure Reject is selected for Action on Service Down.
10. Copy the Secret Server application files you downloaded to Target Web Server A. Copy all files in the SS_update.zip file into Secret Server directory on Target Web Server A. This typically takes less than five minutes, depending on VM resources.

Important: If you have any customized settings that are per-node-specific in the existing web.config or web-appsettings.config files on the target servers, consider whether you want to protect those changes from being overwritten during the upgrade. Because the default contents of those files might change with the upgrade, we strongly recommend copying any customizations line-by-line to the new files, rather than simply replacing the new files with your customized ones. We suggest running a diff or comparison operation on the file pairs to see what, if anything, was customized in the existing files or changed in the new files.

11. When you are prompted, instruct the system copy dialog to overwrite all files.
12. Modify the script you received earlier for use on your production database. You can probably do this while the Secret Server files are copying. For this instruction, we call this the "Production SQL Server."
13. (Optional) When the Secret Server files are finished copying, enable maintenance mode on Target Web Servers B and C to eliminate the possibility of password changes occurring during your database upgrade. The Secret Server read-only vaulting function is still possible from these nodes.

Note: See [Maintenance Mode FAQ](#) for more on maintenance mode.

14. Start IIS on Target Web Server A. Wait until the site fully loads. This may take some time.

Note: Once the site loads, when you locally access that web server, you will see a Secret Server error message saying your database does not match your Secret Server. You can ignore that and click the Continue button.

15. Run the script you modified on the Production SQL Server.
16. Once the script completes, disable Target Web Servers B and C on your load balancer pool.
17. Enable Target Web Server A on your load balancer pool.
18. Access the web server through the load balancer URL. The Secret Server Login page should appear, and you should be able to log on immediately.

Note: If the warning message about the database not matching Secret Server appears, ignore it and click the Continue button.

19. Make Target Web Server A the primary load balancer node in the load balancer pool until the other target web servers are upgraded and online.
20. Disable IIS on Target Web Servers B and C.

21. Manually upgrade the Secret Server application files as previously discussed.
22. Enable IIS on Target Web Servers B and C.
23. Enable Target Web Servers B and C in the load balancer pool.
24. If you earlier put any of the target web servers in maintenance mode as a precaution, return them to normal function.

Manual Rolling Upgrade

Introduction

The manual rolling upgrade provides a way to upgrade Secret Server with little to no downtime. That is, users will continue to have secret access during the upgrade.

Note: This procedure only applies to clustered (multiple Web node) Secret Server environments environment.

Prerequisites

The administrator role needs the following permissions:

- Administer Configuration
- Administer Nodes
- Administer Backup

In addition, the role:

- Needs a database login with permission to change the database
- Requires access with permission to update files on web servers
- Must go through the current upgrade process
- Must not turn on maintenance mode until needed

Procedure

Task One: Uploading the Upgrade

1. Download the latest version of Secret Server from the [Delinea Support Website](#).
2. Navigate to **Admin > See All > Upgrade Secret Server:**

Upgrade Secret Server



How do I upgrade Secret Server with little to no downtime? [Click here](#) to learn more.

MAINTENANCE MODE

A user should enable Maintenance Mode before upgrading Secret Server to ensure limited downtime during the upgrade process. Be aware that a user cannot make changes to the database while in Maintenance Mode, this includes changing Secrets or Secret-related data. Want to learn more about Maintenance Mode? [Click here](#).

Maintenance Mode has not been enabled on all nodes. Please enable to make Secret Server read-only during the upgrade.

Enable Maintenance Mode

Do not put Secret Server in Maintenance Mode during the upgrade process.

BACKUP

Last Successful Backup - 08/30/2019 14:19:36

1) Backup your Secret Server application folder.

IMPORTANT: All your data is encrypted using a file named encryption.config in your Secret Server application folder and cannot be decrypted without it. Please be sure to make a backup of the application folder and its contents to avoid any complications.

Secret Server application folder location: C:\inetpub\wwwroot\SecretServer\encryption.config

2) Backup the database SecretServer on SQLSERVER

The Secret Server database and application folder have been backed up.

3. **Important:** Click to select the **Do not put Secret Server in Maintenance Mode during the upgrade process** check box.
4. Backup the Secret Server application folder.
Important: Ensure the encryption.config file is backed up. It is located at c:\inetpub\wwwroot\SecretServer\encryption.config.
5. Click the **Backup** button to back up the Secret Server database.
6. Click the **Continue** button. The Upgrade Secret Server page appears:

Help

This page is used to apply patches to Secret Server that have been delivered from support or to apply upgrades of Secret Server. Before continuing ensure that the Secret Server application and database have been backed up successfully.

Upgrade Secret Server

Current Version	10.7.000000	
Latest Version	10.6.000027	The latest version is already installed.

[Advanced \(not required\)](#)

7. Click the **Advanced (not required)** link. The Advanced section appears:

Advanced (not required)

WARNING!

This option is for advanced users only. Use this option only if you are unable to update Secret Server from our servers. Providing an invalid upgrade file may result in permanent loss of data.



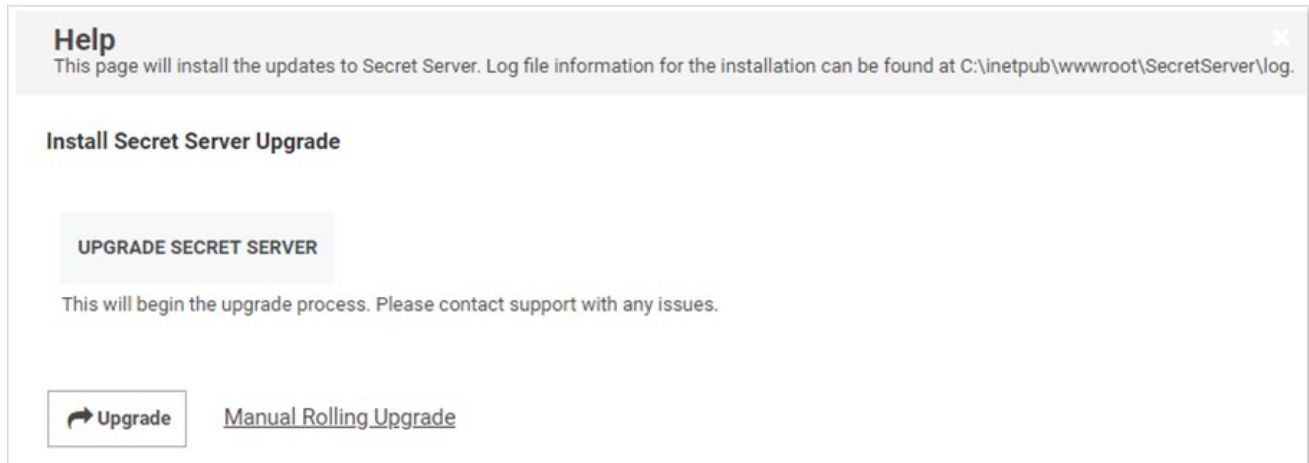
Make sure you have backed up your installation before continuing. For more information, please see our [knowledge base article](#).

If you are currently connected to the Internet the latest version can be downloaded from here: [Download Latest Version](#). Once downloaded choose to upload that file from the option below. After upload is complete an option will appear to install that version.

Choose File No file chosen

 Upload Upgrade File

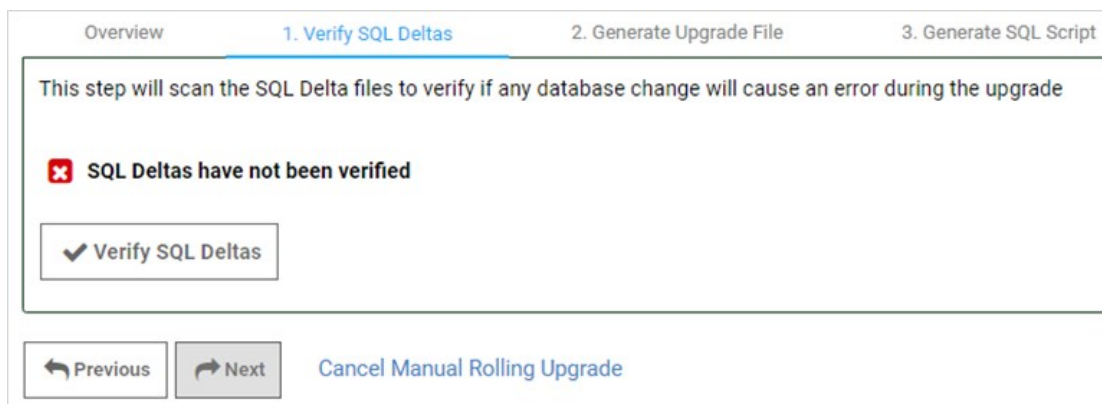
8. Click the **Choose File** button, and select the zip file you downloaded earlier to upgrade to.
9. Click the **Upload Upgrade File** button. The new version appears as available for installation:



10. Click the **Manual Rolling Upgrade** link. The Manual Rolling Upgrade wizard appears.

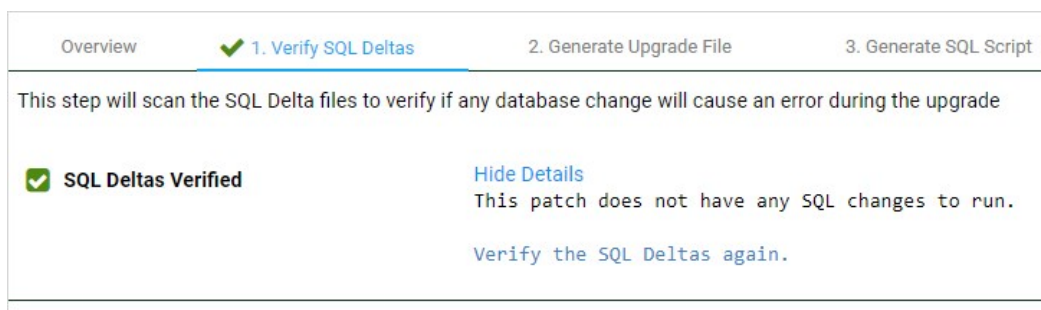
Task Two: Verifying SQL Changes (Wizard Step One)

1. Click the **Next** Button. The Verify SQL Deltas tab appears:



Note: Clicking the "Cancel Manual Rolling Upgrade" link, at any time, will take you to the Install Secret Server Upgrade page.

2. Click the **Verify SQL Deltas** button. This tests the prospective changes to see if errors result. If errors result, please contact Delinea Technical Support. If the verification succeeds:



Task Three: Generating the Upgrade File (Wizard Step Two)

1. Click the **Next** button. The Generate Upgrade File tab appears:

Overview ✓ 1. Verify SQL Deltas **2. Generate Upgrade File** 3. Generate SQL Script

This step will scan the application files, remove any unchanged files and any files that are not needed. Then it will create a zip file with only those files that are needed for the upgrade.

✗ Application Zip File has not been generated

[Download Application Zip File](#)

[Previous](#) [Next](#) [Cancel Manual Rolling Upgrade](#)

2. Click the **Download Application Zip File** button. This generates a zip file with only the changed files needed to upgrade the application files on the Web server nodes.

Note: This may take a few minutes to generate and download.

Overview ✓ 1. Verify SQL Deltas **✓ 2. Generate Upgrade File** 3. Generate SQL Script 4. Backup & Staging

This step will scan the application files, remove any unchanged files and any files that are not needed. Then it will create a zip file with only those files that are needed for the upgrade.

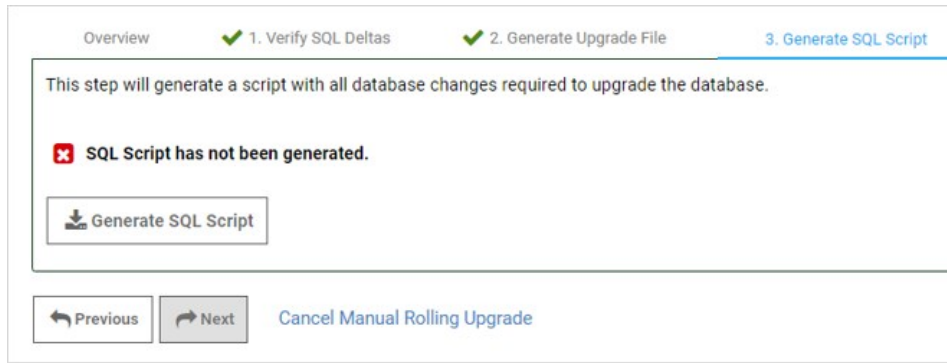
✓ Application Zip File Downloaded [Hide Details](#)

Validating temporary update contents.
Began building Manual Upgrade package at 10:57 AM
Begin Removing Unchanged Files at 10:57 AM
Scanning for unchanged application files from:
C:\Windows\TEMP\SecretServer\ManualUpdate\ManualUpgradeFileStaging
Ignore Pattern: (\.config|\.sql)\$
Cleaning any additional unnecessary files and folders.
Total folders checked: 168
Total unchanged files removed: 2055
Total changed files detected: 1
Completed Removal of Unchanged Files.
Building Manual Upgrade Zip package.
Finished building Manual Upgrade Zip package at 10:57 AM
Ran in 17.1013256s.

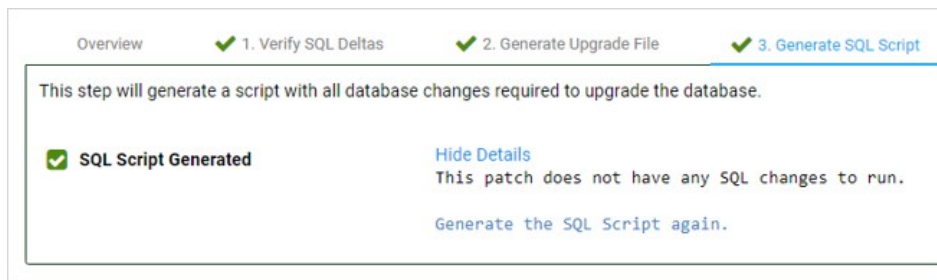
[Download Application Zip File again.](#)

Task Four: Generating the SQL Script (Wizard Step Three)

1. Click the **Next** button. The Generate SQL Script tab appears:



2. Click the **Generate SQL Script** button. This generates script file with all the database changes needed to upgrade the database. When finished:



The wizard proceeds to step four:


Task Five: Backing up and Staging (Wizard Step Four)

Overview ✔ 1. Verify SQL Deltas ✔ 2. Generate Upgrade File ✔ 3. Generate SQL Script 4. Backup & Staging

✘ Maintenance Mode has not been enabled on all nodes. Please enable to make Secret Server read-only during the upgrade.

✔ Enable Maintenance Mode

The purpose of this step is to run through upgrading Secret Server in a staging environment, using a current back up of Secret Server. This will verify that the upgrade will be successful, before performing the upgrade in your production environment. Follow these manual steps to ensure a successful upgrade before proceeding to the next tab:

1. Enable maintenance mode. This will ensure that the staging system will start in maintenance mode.
2. [Backup Secret Server](#). 
3. Disable maintenance mode.
4. Restore Secret Server files to staging location.
5. Restore Secret Server database to a staging database.
6. Go to ADMIN > Server Nodes to confirm that the staging system is in Maintenance Mode.
7. Copy the contents of the generated application ZIP file to the staging location's web application folder.
8. Run the generated SQL script on the staging database.
9. Log into the upgraded staging Secret Server and verify the upgrade was successful.
10. (Optional) Delete the restored staging location and database.

i Important: Keep your backup files until you have finished this entire upgrade process in case complications arise.

Check to signify that your staging upgrade was successful before proceeding to the production upgrade:

Staging Test Successful

[← Previous](#) [Next →](#) [Cancel Manual Rolling Upgrade](#)

1. Click the **Enable Maintenance Mode** button.
2. Back up Secret Server: Type "backup" in the Admin search text box, and click the item that appears in the dropdown list to access the Backup Configuration page. Click the **Backup Now** button.
3. Click the **Disable Maintenance Mode** button.
4. Restore Secret Server files to the staging location:
 1. Copy the backup zip file to the staging location.
 2. Unzip the backup file.
 3. Copy the files to the web application folder.
5. Restore the Secret Server database to a staging database:
 1. In SQL Server Management Studio, right click on **Databases**.
 2. Click **Restore Database**.
 3. In **Source**, select **Device**.
 4. Select and add the backup database file location.
 5. Click **Ok**.

- Go to **Admin > Secret Nodes** to confirm the staging system is in maintenance mode.
- Copy the contents of the generated application Zip file to the staging location's web application folder. Typically, this is `C:\inetpub\wwwroot\SecretServer`.
- Run the generated SQL script on the staging database.
- Log on the upgraded staging Secret Server to verify the upgrade was successful.
- (Optional) Delete the restored staging location and database.

Important: Keep the backup files till you verify the upgrade was successful. You may need them if an issue develops.

- Click to select the **Staging Test Successful** check box to confirm your staging upgrade was successful. This is your confirmation that there were no errors before performing the actual upgrade in your production environment. The confirmation is recorded.

Task Six: Starting Upgrade Mode (Wizard Step Five)

- Click the **Next** button. The Enter Upgrade Mode tab appears:

- Click the **Enable Maintenance Mode** button. This mode limits the activities of users on secrets, secret templates, password requirements, and others and can take several minutes to start. A confirmation popup appears.
- Click the **Enable** button to confirm the mode change. The popup disappears.
- Click the **Enable Ignore Version Mismatch** button. This prevents users from being redirected to the Version Mismatch page. A confirmation popup appears.
- Click the **Enable** button to confirm the setting change. The popup disappears.
- Click the **Next** button. The Manual Steps tab appears:

Overview
✓ 1. Verify SQL Deltas
✓ 2. Generate Upgrade File
✓ 3. Generate SQL Script
✓ 4. Backup & Staging
✓ 5. Enter Upgrade Mode
6. Manual Steps
Finish

SUMMARY

- ✓ Verify SQL Deltas
- ✓ Generate Upgrade File
- ✓ Generate SQL Script
- ✓ Backup & Staging
- ✓ Enter Upgrade Mode

MANUAL STEPS TO UPGRADE WEB NODES

Once you are ready to perform the upgrade in your production environment, these steps will guide you through completing the manual rolling upgrade.

For more details about the Manual Rolling Upgrade process [click here](#).

Divide your nodes so that traffic can be routed to one set of nodes (group A) while you upgrade the other nodes (group B).

To upgrade the nodes in group B, follow these steps:

1. Make sure Maintenance Mode is enabled on all nodes and that Ignore Version Mismatch is enabled. Go to the Enter Upgrade Mode tab to change these settings.
2. On the load balancer, force all traffic to group A nodes by disabling traffic to group B nodes. To prevent end user interruption wait until all connections are drained from group B before proceeding with the next step.
3. Unzip the Application zip file and copy all the files to the web application folder on all the nodes in group B.
4. Log into a node in group B and confirm that the site loads and logs in correctly.
5. On the load balancer, force all traffic to group B nodes by enabling them back into the pool and disabling group A nodes. To prevent end user interruptions you will want to wait until all connections are drained from group A nodes prior to proceeding with the next step.
6. Run the SQL script on the database and confirm that there are no errors. If you encounter any errors, follow the rollback instructions below.
7. Log into a node in group B again and confirm that the site loads and logs in correctly.

Next, complete upgrade on the other web nodes:

1. Unzip the Application zip file and copy all the files to the web application folder on all the nodes in group A.
2. Log into a node in group A and confirm that the site loads and logs in correctly.
3. On the load balancer, restore the original configuration, sending traffic to all nodes.

If you encounter any errors at any step of the upgrade, rollback to the previous version by doing the following:

1. Restore your database from backup.
2. Restore your application files from backup to the web application folder on all nodes.
3. On the load balancer, restore the original configuration, sending traffic to all nodes.
4. [Contact support](#) for assistance.

[Show Server Nodes](#)

Check to signify that your manual rolling upgrade was successful before proceeding:

Upgrade Successful

Need Help? [Contact Support](#)

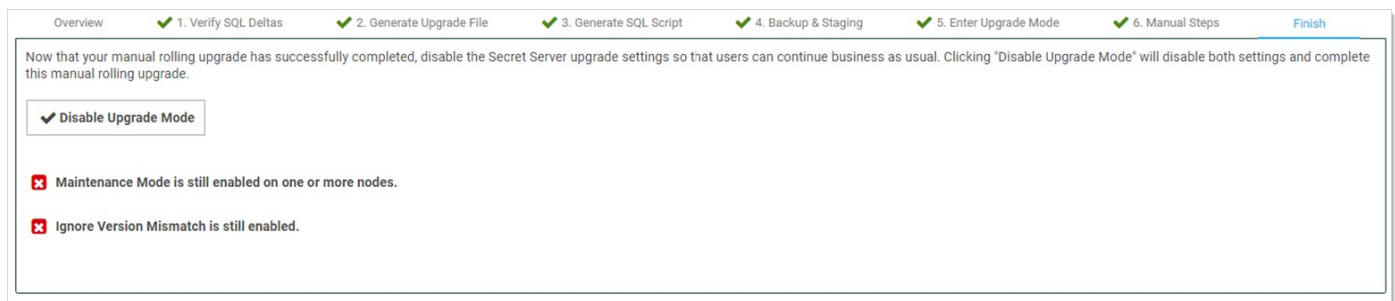
← Previous
Next →
Cancel Manual Rolling Upgrade

Task Seven: Upgrading Web Nodes (Wizard Step Six)

To upgrade Web nodes:

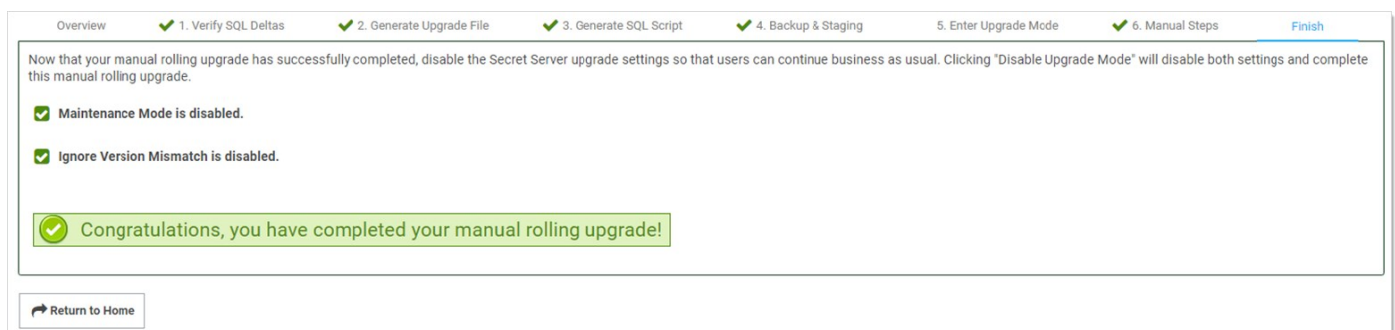
1. Split your nodes into two approximately even groups (A and B) so that one group can service traffic while the other is upgrading.
2. Ensure "maintenance mode" and "ignore version mismatch" are enabled on each node. You can change them from the Enter Upgrade Mode tab.
3. On the load balancer, disable traffic to group B. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group A, alone, now handles the traffic. For example, on a F5 Big-IP load balancer you:
 1. Select the Members tab on the pool page.
 2. Select the node to disable.
 3. Click Force Offline.
4. For each node in group B:
 1. Navigate to the Downloads folder.
 2. Extract all the files from the application zip file downloaded earlier.
 3. Copy the extracted files to the Web application folder.
 4. Log onto the node to ensure the site correctly loads and logs on.

5. On the load balancer, enable the group B nodes to return them to the pool.
6. Disable traffic to group A. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group B, alone, now handles the traffic.
7. Execute the script you created on the database, confirming there are no errors. If there are errors, follow the [rollback instructions](#).
8. Log onto each group B node again to ensure the site correctly loads and logs on.
9. For each node in group A:
 1. Navigate to the Downloads folder.
 2. Extract all the files from the application zip file downloaded earlier.
 3. Copy the extracted files to the Web application folder.
 4. Log onto the node to ensure the site correctly loads and logs on.
10. On the load balancer, enable the group A nodes to return them to the pool, restoring the original configuration and returning traffic to all nodes.
11. Click to select the **Upgrade Successful** check box.
12. Click the **Next** button. The Finish tab appears:



Task Eight: Finishing up (Wizard Step Seven)

1. Click the **Disable Upgrade Mode** button. A Finish Manual Rolling Upgrade popup appears. This popup both disables maintenance mode and disables the ignore version mismatch setting.
2. Click the **Disable** button. The popup disappears, and a completion message appears:



Troubleshooting and Notes

Rolling Back to the Previous Version

If you encounter errors at any step of the upgrade, rollback to the previous Secret Server version:

1. Restore the database from the backup.
2. Restore the application files from the backup files to the Web application folder on all nodes.
3. On the load balancer, restore the original configuration, sending traffic to all nodes.
4. For assistance, contact us at the [Delinea Support Website](#).

Version Guard

If an uploaded upgrade file cannot be used to upgrade the current version of Secret Server, then "Version Guard" will block the upgrade and provide instructions on how to continue:

Help

Welcome to the Manual Rolling Upgrade Process for Secret Server. The process below is designed to apply a Secret Server upgrade with little to no downtime for your organization. This process is only available for organizations with a clustered (multiple web node) environment.

Completing all required steps will take varying amounts of time to complete, depending on your environment. Please be prepared to complete all steps of the process at once.

Before continuing, ensure that the Secret Server application and database have been backed up successfully.

For more details about the Manual Rolling Upgrade process [click here](#).

Secret Server Manual Rolling Upgrade

VERSION GUARD

Unable to upgrade Secret Server from 10.7.000000 to version Version.xml file does not exist.

[Return To Upgrade](#)

This usually occurs when not completing the prerequisite steps in order. Click the **Return To Upgrade** button to return you to the first upgrade page to remedy the situation.

Secret Server Manual Rolling Upgrade

VERSION GUARD

Unable to upgrade Secret Server from 10.7.000002 to version 10.8.000001.

Secret Server must be upgraded to the following versions, in order, before upgrading to version 10.8.000001:

1. Version **10.7.000050**

[Return To Upgrade](#)

This page also list the blocking versions that you must upgrade to prior to running the manual rolling upgrade.

New Advanced Configuration Setting

There is a new setting called "Manual Upgrade: Allow version mismatch while in Maintenance Mode." This setting, which only applies in maintenance mode, prevents Secret Server from redirecting users to the version mismatch message page.

New Audit Type

To support the manual rolling upgrade, there is a new audit type—ManualUpgrade. Its audits are stored in the tbAudit table and record the following actions:

- CANCEL
- COMPLETED
- GENERATE DB SCRIPT
- GENERATE UPGRADE ZIP
- STAGING TEST
- STARTED
- VERIFY DELTAS

Secret Server and Secret Server Cloud .NET Framework 4.8 Mandatory Upgrade

Important: This is a major part of the 10.9.000005/33 step upgrade. Please see [Upgrading to 10.9.000005/33](#).

Introduction

This topic explains the steps to prepare your environment for the December 2020 upgrade to Secret Server and Secret Server Cloud (SSC). These changes must be made before the December release is deployed. Secret Server Online is not affected by these changes.

The Microsoft .NET Framework is a core component of the Secret Server architecture. Secret Server versions up to and including 10.9.000002 require version 4.5.1 or higher of the .NET Framework.

The December 2020 releases of Secret Server and SSC will change this requirement. From December 2020 onwards, version 4.8 or higher of the .NET Framework will be required.

This change improves the security of communication between Secret Server and distributed engines. It also allows Delinea to maintain compatibility with other third party libraries required by Secret Server and SSC. Secret Server online is not impacted by these changes.

The December release is scheduled for Secret Server installed (on-premise) customers on 8 December 2020. The December release is scheduled for SSC customers on 12 December 2020.

Preparing Secret Server for the December Release

To accommodate this change in software requirements, Secret Server and SSC customers may need to install .NET Framework 4.8 runtime on the computer systems hosting components of Secret Server in their environment. These changes can be made in advance without impacting the operation of your current version of Secret Server or SSC. The required actions, workarounds, and the impact of not taking action are listed here for each component of Secret Server infrastructure:

Secret Server Web Nodes

Products

Secret Server

Required Action

Installing .NET Framework 4.8 on servers running web nodes.

Impact of Inaction

Upgrade to the December 2020 release will not proceed until .NET Framework 4.8 has been installed.

Notes

The web node components of SSC are managed directly by Delinea.

Distributed Engines

Products

-\$1Secret Server\$2\$1Secret Server\$2

- Secret Server Cloud

Required Action

Installing .NET Framework 4.8 on servers running web nodes.

Impact of Inaction

- Secret Server upgrade will request confirmation before upgrading if any connected distributed engines do not have .NET Framework 4.8 installed. Any distributed engines which do not have .NET framework 4.8 installed will fail to upgrade.
- SSC will be upgraded whether or not distributed engines have .NET Framework 4.8 installed. Any distributed engines which do not have .NET framework 4.8 installed will fail to upgrade.

Notes

Installing .NET Framework 4.8 after the upgrade will allow a distributed engine to start and reconnect to Secret Server.

Protocol Handler

Products

-Secret Server-Secret Server

- Secret Server Cloud

Required Action

- New installations of the protocol handler on Windows systems after the December release will require .NET Framework 4.8 installed.
- Fully patched Windows 10 systems should already have the framework in place.
- Customers with existing deployments of protocol handler have two options:
 - Ensure that .NET Framework 4.8 is installed on all endpoints using protocol handler before the Secret Server or SSC update.
 - Disable automatic updating of protocol handler:
 1. Go to **Admin > Configuration**.
 2. Set **Enable Protocol Handler Auto-Update** to **No**.
- Secret Server is compatible with older versions of protocol handler. Older versions will continue to function when used with the Secret Server December release.

Impact of Inaction

Any updated installs of protocol handler will fail on systems that do not have .NET Framework 4.8 installed.

Notes

None.

Advanced Session Recording Agent (ASRA)

Products

-Secret Server-Secret Server

- Secret Server Cloud

Required Action

Existing installations of ASRA will continue to operate. New installations of ASRA after the December release will require .NET Framework 4.8 installed on the servers where ASRA is deployed.

Impact of Inaction

Existing installations of ASRA will continue to operate.

Notes

None.

Session Connector

Products

-1Secret Server21Secret Server2

- Secret Server Cloud

Required Action

None. This release will not require .NET Framework 4.8 for session connector.

Impact of Inaction

None. Session connector will continue to function.

Notes

Although the December release will not require .NET Framework 4.8, future updates to session connector will require it. Therefore, we recommend installing .NET Framework 4.8 on any Microsoft Remote Desktop Services (RDS) servers used as part of the session connector infrastructure.

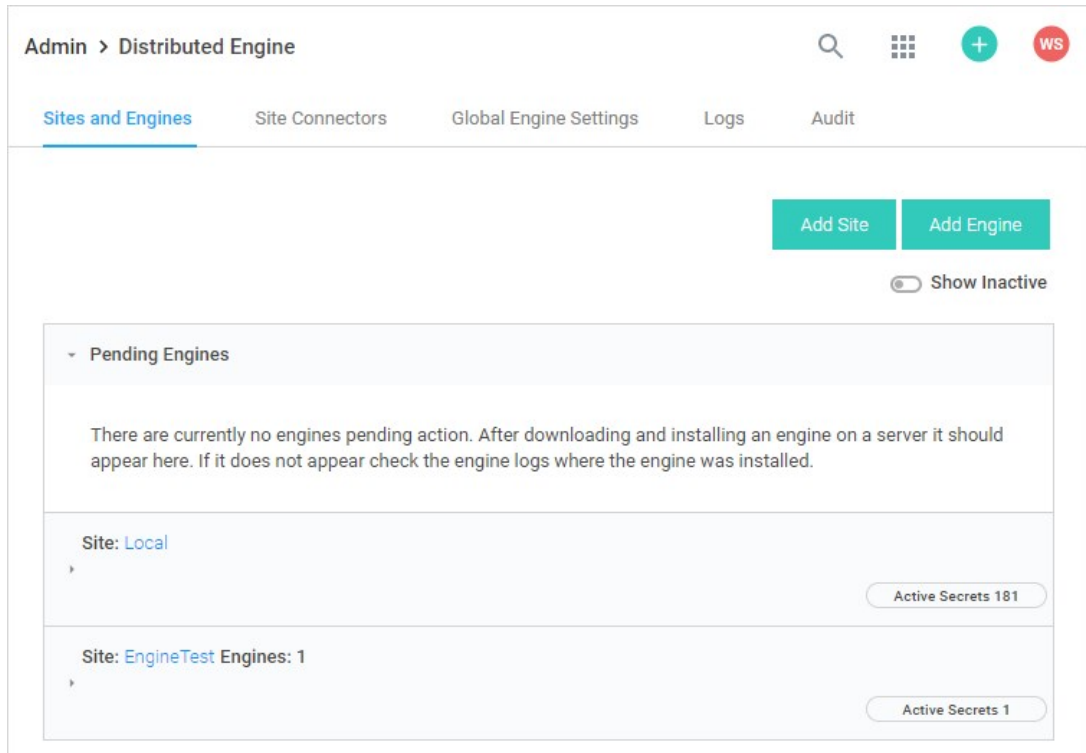
Effects on Connection Manager

None: Connection Manager will be updated to require .NET Framework 4.8 at a later date.

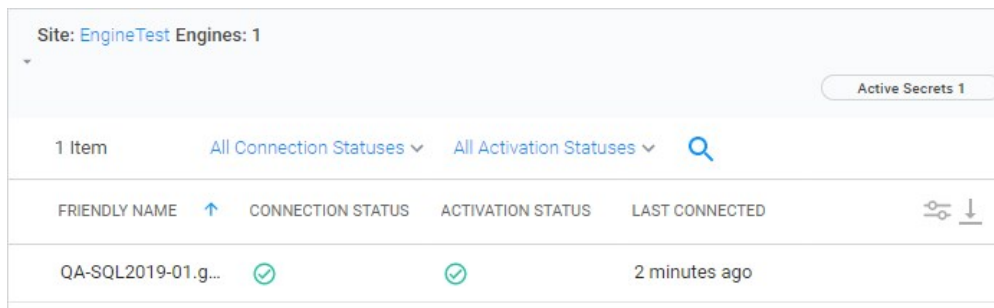
Identifying Distributed Engine Servers

To identify which servers in your environment are running Distributed Engines and may require a .NET Framework 4.8 update:

1. In Secret Server, go to **Admin > Distributed Engine**. The Distributed Engine Configuration page appears:



2. Click the site panel button for the desired DE. The panel expands, displaying the DEs for that site:



3. Note the servers in your environment that currently have distributed engines installed. These are the machines where you need to install the .NET 4.8 runtime.

Unaffected Secret Server Components

The following components of Secret Server and SSC are not affected by this change:

- SQL Server. Database system requirements are not affected by this change.
- Web password filler
- Secret Server Desktop Application Secret Server
- Secret Server Mobile

Determining Your .NET Framework Version

- Microsoft provides several methods to determine the installed versions of .NET Framework. Please see [How to: Determine which .NET](#)

[Framework versions are installed.](#)

- See [.NET Framework versions and dependencies](#) for additional Microsoft .NET Framework version information.
- You can run the following command at a Windows PowerShell prompt to view currently installed .NET Framework version number: (Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full").version
- Secret Server on-premises (not SSC) displays the .NET framework version on the Admin > Diagnostics page. The version displayed is for the Web server being accessed. It does not include information about the .NET Framework version installed on any other server.

Important: The version displayed on the Diagnostics page incorrectly identifies .NET Framework 4.8 as version 4.5.1 followed by a release number, such as 4.5.1.528040. If the release number is 528040 or higher, the framework version number is 4.8. If the release number is lower than 528040 or the release number is not displayed at all, the framework version is lower than 4.8 and an the mandatory update is required.

Installing .NET Framework 4.8

- Microsoft .NET Framework 4.8 is available as a recommended update for customers using Windows Update, Windows Server Update Services (WSUS), and Microsoft Update (MU) Catalog. Please see [.NET Framework 4.8 is available on Windows Update, WSUS and MU Catalog](#).
- Microsoft provides a Web installer for .NET Framework 4.8. Please see [Download .NET Framework 4.8](#).
- Microsoft also provides an offline installer package for .NET Framework 4.8. Please see [Microsoft .NET Framework 4.8 offline installer for Windows](#).

Note: If there is an exception while monitoring ASP .NET applications, see [Troubleshooting steps when there is an exception while monitoring ASP .NET applications](#) for more information

Secret Server Cloud IP Address Change for March to May 2021

Overview

Delinea is making infrastructure upgrades to Secret Server Cloud (SSC) over the next few weeks. As part of this work, the IP addresses for SSC will change. Your SSC subdomain will not change.

You may need to modify your inbound or outbound firewall configurations to maintain uninterrupted service.

FAQ

Who Is Affected?

Customers using inbound or outbound firewall rules to control traffic to and from SSC are affected.

Inbound firewall rules are necessary for customers using Remote Authentication Dial-In User Service (RADIUS) authentication.

What Other Delinea Products Are Affected?

None—these products are **not** affected:

- All other Delinea cloud products
- Secret Server private cloud customers Secret Server
- Secret Server on-premises customers

What Delinea Domains Are Affected?

If you use IP filtering, the following domains are affected:

Table: Affected Domains by Region

America	secretservercloud.com
Australia	secretservercloud.com.au
Canada	secretservercloud.ca
Europe	secretservercloud.eu
Singapore	secretservercloud.com.sg

[Unexpected Link Text](#)

How Do I Verify My Domain Is Affected?

Review the table above to see if any of those domains appear in your SSC URL. For example, if your domain were `https://acmewidgets.secretservercloud.eu` you would be affected.

What Do I Need to Change?

If you confirmed your domain is affected, you need to change your inbound IP address and outbound hostnames for your firewall rules prior to

the dates in the **New IP Addresses and Hostnames** table below.

When Are The IP Addresses and Hostnames Changing?

The IP addresses and host names for the listed regions are **not** changing at the same time. Please see the **New IP Addresses and Hostnames** table below for the date or dates applying to you.

Will I Lose Connectivity During the Change?

It is unlikely you will lose connectivity with SSC unless you use inbound or outbound IP filtering and have not updated to the new IP addresses or hostnames found the table below.

How Will I Know When the Change Is Complete?

Delinea will update the SSC banner as we complete the IP address change.

Who Do I Contact If I Have Issues After the Change?

Contact Delinea technical support at any of the phone numbers below or by opening a case using the [Support Portal](#).

Delinea Support

Support Portal

[Support Portal](#)

Telephone

Americas

+1 202 991 0540 (US)

EMEA

- +44 20 3880 0017 (UK)
- +49 69 6677 37597 (Germany)

APAC

- +61 3 8595 5827 (Australia)
- +63 2 231 3885 (Philippines)
- +64 9-887 4015 (New Zealand)
- +65 3157 0602 (Singapore)

IP Addresses and Hostnames

New IP Addresses and Hostnames

Table: New IP Addresses and Hostnames

--

secretservercloud.com	52.224.253.7 52.224.253.4	<ul style="list-style-type: none"> • thycotic-ssc-us-er-sb-01-prod-b.servicebus.windows.net (primary) • thycotic-ssc-us-er-sb-01-prod-g.servicebus.windows.net (primary) • thycotic-ssc-us-er-sb-02-prod-b.servicebus.windows.net (dr) • thycotic-ssc-us-er-sb-02-prod-g.servicebus.windows.net (dr) 	24 April 2021 8 May 2021
secretservercloud.com.au	20.37.251.37 20.37.251.120	<ul style="list-style-type: none"> • thycotic-ssc-au-er-sb-01-prod-b.servicebus.windows.net (primary) • thycotic-ssc-au-er-sb-01-prod-g.servicebus.windows.net (primary) • thycotic-ssc-au-er-sb-02-prod-b.servicebus.windows.net (dr) • thycotic-ssc-au-er-sb-02-prod-g.servicebus.windows.net (dr) 	10 April 2021
secretservercloud.ca	52.228.117.246 52.228.113.119	<ul style="list-style-type: none"> • thycotic-ssc-ca-er-sb-01-prod-b.servicebus.windows.net (primary) • thycotic-ssc-ca-er-sb-01-prod-g.servicebus.windows.net (primary) • thycotic-ssc-ca-er-sb-02-prod-b.servicebus.windows.net (dr) • thycotic-ssc-ca-er-sb-02-prod-g.servicebus.windows.net (dr) 	3 April 2021
secretservercloud.eu	20.79.64.213 20.79.65.3	<ul style="list-style-type: none"> • thycotic-ssc-eu-er-sb-01-prod-b.servicebus.windows.net (primary) • thycotic-ssc-eu-er-sb-01-prod-g.servicebus.windows.net (primary) • thycotic-ssc-eu-er-sb-02-prod-b.servicebus.windows.net (dr) • thycotic-ssc-eu-er-sb-02-prod-g.servicebus.windows.net (dr) 	1 May 2021 15 May 2021
secretservercloud.com.sg	20.195.97.220 20.195.98.154	<ul style="list-style-type: none"> • thycotic-ssc-sea-er-sb-01-prod-b.servicebus.windows.net (primary) • thycotic-ssc-sea-er-sb-01-prod-g.servicebus.windows.net (primary) • thycotic-ssc-sea-er-sb-02-prod-b.servicebus.windows.net (dr) • #thycotic-ssc-sea-er-sb-02-prod-g.servicebus.windows.net (dr) 	3 April 2021

[Unexpected Link Text](#)

Note: The US and EU regions have two dates listed for changes because the upgrade work will be performed over two days. Please make any necessary changes before the earlier date.

Old IP Addresses and Hostnames

For reference, the original IP addresses and hostnames were as follows:

Table: Old IP Addresses and Hostnames

secretservercloud.ca	13.88.237.67 52.228.62.157	<ul style="list-style-type: none"> • thycotic-ssc-02-prod-ca-bus-er.servicebus.windows.net • thycotic-ssc-02-prod-ca-bus-er-g.servicebus.windows.net
secretservercloud.com	40.76.197.147 40.121.181.52	<ul style="list-style-type: none"> • thycotic-ssc-02-prod-use1-bus-er.servicebus.windows.net • thycotic-ssc-02-prod-use1-bus-er-g.servicebus.windows.net
secretservercloud.com.au	20.36.47.199 20.36.45.106	<ul style="list-style-type: none"> • thycotic-ssc-02-prod-auce-bus-er.servicebus.windows.net • thycotic-ssc-02-prod-auce-bus-er-g.servicebus.windows.net
secretservercloud.com.sg	137.116.141.200 137.116.143.17	<ul style="list-style-type: none"> • thycotic-ssc-02-prod-sea-bus-er.servicebus.windows.net • thycotic-ssc-02-prod-sea-bus-er-g.servicebus.windows.net
secretservercloud.eu	51.116.228.208 51.116.228.152	<ul style="list-style-type: none"> • thycotic-ssc-02-prod-dewc-bus-er.servicebus.windows.net • thycotic-ssc-02-prod-dewc-bus-er-g.servicebus.windows.net

Upgrading Secret Server

Note: See [Upgrading to Secret Server 10.9.000005/10.9.000032](#) for instructions specific to that upgrade.

Important: If upgrading to 10.7.000000 or later, using SQL Server 2008 R2 as the Secret Server database is no longer supported. In addition, the *existing* MS SQL database must be version 2012 or later for this upgrade. Otherwise, if the upgrade fails and you attempt to roll it back, the previous installation will not work. For more information, see the [release notes](#).

Important: Customers upgrading to 10.6.000000 or later and are using RabbitMQ, please see [Clearing RabbitMQ Message Queues](#).

Important: Customers using the Integrated Windows Authentication (IWA) feature need to perform a workaround when upgrading to Secret Server 10.6 with a Distributed Engine. IWA is the Windows feature where users log on their Windows domain only once—once logged on, any additional domain logons are done automatically without having to reenter a user name and password.

Important: Upgrading to Secret Server version 8.9.000000 and above will require **Windows Server 2008 R2 or greater**.

Important: If you are upgrading to Secret Server version 8.5.000000 and above, there are changes in the .NET Framework version you will need to be aware of along with some additional steps in the upgrade process. For more information, see [Secret Server and Secret Server Cloud .NET Framework 4.8 Mandatory Upgrade](#).

Important: Upgrading to Secret Server version 10.0.000000 and above will require configuring integrated pipeline mode on the Secret Server Application Pool. Please see [Manual IIS Installation](#) for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication you will also need to update IIS authentication settings as detailed in [Configuring Integrated Windows Authentication](#). If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000 and above.

Important: If you are doing an incremental upgrade from Version 9.1.000000 to a higher version, the system may require additional time to process the changes before proceeding. A typical symptom of this behavior will be the software will redirect you to the home page. If this happens, please allow up to 24 hours before retrying the upgrade. If the issue persists, please contact technical support. You should lose no other functionality of the software whilst this occurs.

Important: If you have Privilege Manager installed, the Secret Server upgrade process will begin an upgrade for Privilege Manager as well.

How Upgrades Work

Secret Server periodically polls the update server to detect new updates. If the "Allow Automatic Checks for Software Updates" option is enabled in the Admin > Configuration menu, you will see the "An update is available (xx.x.xxxxx)" link after logging in with an administrator account. The steps below can be used to perform an upgrade for versions 7.1.000015 and higher. If you have an older version of Secret Server, please contact Delinea technical support for assistance.

Before You Begin

1. Ensure you will have access to account credentials for the server hosting Secret Server AND the SQL Server instance hosting your Secret Server database.
2. Ensure you have a recent backup of the application files and database available.
3. If you use clustering, stop the application pools on all of the servers except the one that is currently the "primary."

How to Upgrade

1. From a computer that has outbound network access, click on the upgrade link to go to: <http://<yourinstance>/Setup/Home>. If you are upgrading from a version lower than 10.2.000000, the URL will be <http://<yourinstance>/installer.aspx>. The Secret Server Setup Home page appears:

Note: If your computer does not have outbound network access, please see [Upgrading Secret Server Without Outbound Access](#).

Secret Server Setup Home

Secret Server Version 10.2.000000

Please note the following configuration settings that will need to be configured in order to use certain features.

[Secret Server Upgrade Available](#) ⓘ
[No RabbitMq Internal Communication Site Connector](#) ⓘ

Online Setup Resources

[Getting Started with Secret Server](#)

[User Guide](#)

Privilege Manager for Windows


[TMS Setup](#)
[Privilege Manager](#)

Privileged Behavior Analytics

[Configure PBA](#) ⓘ
[Access PBA](#) ⓘ

2. Click the **Secret Server Upgrade Available** link to continue. The Upgrade Secret Server page appears:

Upgrade Secret Server

 Please be sure to backup the Secret Server application folder and database before continuing the upgrade.

1) Backup your Secret Server application folder.

IMPORTANT: All your data is encrypted using a file named encryption.config in your Secret Server application folder and cannot be decrypted without it. Please be sure to make a backup of the application folder and its contents to avoid any complications.

Secret Server application folder location: C:\inetpub\wwwroot\SecretServer\encryption.config

2) Backup the database SecretServer on 10.12.30.3

The Secret Server database and application folder have been backed up.

[Continue](#)

3. Backup your Secret Server application folder.

Important: All your data is encrypted using the encryption.config file in your Secret Server application folder. **Your data cannot be decrypted without it.** Thus, it is critical that you backup the application folder and its contents before

proceeding.

4. Backup the database named SecretServer at the IP address listed.
5. When finished backing up both, click to select the **The Secret Server database and application folder have been backed up** check box.
6. Click the **Continue** button. Another Upgrade Secret Server page appears:

Upgrade Secret Server

Current Version	10.2.000000
Latest Version	10.2.000001

[Download Latest Version](#)

[Advanced \(not required\)](#)

7. Click the **Download Latest Version** button to download Secret Server. Wait for the download to finish. The Install Secret Server Upgrade page appears:

Install Secret Server Upgrade

Upgrade Secret Server


This will begin the upgrade process. Please contact support with any issues.

[Upgrade](#)

8. Click the **Upgrade** button. The upgrade starts. When it is finished, the Secret Server Upgrade Installation Status page appears:

Secret Server Upgrade Installation Status

[?](#)

 **CONGRATULATIONS! Installation is complete.**

[Show Details](#)

[Return to Home](#)

9. Click the **Return to Home** button to return to the dashboard. The upgrade is complete.

10. If you intend to use Web clustering, proceed to [Upgrading Secret Server with Web Clustering](#).

Upgrading Secret Server with Web Clustering

Introduction

Secret Server has a built-in Web installer. The Web installer is a series of pages inside Secret Server that allow you to download and run updates. Secret Server is accessible by users for most of the upgrade process. You can bring down outside access to the site if you want to prevent users from making changes during the upgrade. Preventing user access makes restoring the database and site backups simpler if you decide to roll back the upgrade immediately afterward.

Note: You do not need to download the installer or `setup.exe`.

Important: Never overwrite or delete your `encryption.config` file.

Important: Back up your Secret Server folder and database before performing the upgrade.

Important: Upgrading to Secret Server version 10.7.000000 and above, requires SQL Server 2012 or later as the database for Secret Server. For more information, see the [Release Notes](#).

Important: Upgrading to Secret Server version 10.0.000000 and requires configuring integrated pipeline mode on the Secret Server Application Pool. Please see [Configuring IIS for installing or upgrading to Secret Server 10](#) (KBA) for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication, you will also need to update IIS authentication settings as detailed in [Configuring Integrated Windows Authentication](#). If you are at version 9.1.000000 and below, you need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000 and above.

Important: Upgrading to Secret Server version 8.9.000000 and above requires Windows Server 2008 R2 or later.

Important: Upgrading to Secret Server version 8.5.000000 and above, there are changes in the .NET Framework version you will need to be aware of along with some additional steps in the upgrade process. For more information, see [Secret Server Moving to .NET Framework 4.5.1](#).

Before Beginning

1. Ensure that you have account credentials information and access for the server hosting Secret Server *and* the SQL Server instance hosting your Secret Server database.
2. Have a recent backup of the application files and database available.
3. If you use clustering, stop the application pools on all of the servers.

Upgrading a Clustered Environment

1. Follow the instructions in [Upgrading Secret Server](#) or [Upgrading Secret Server Without Outbound Access](#) as applicable to upgrade one server.
2. Once upgraded and working, copy the Web application folder (without the `database.config` or the `encryption.config` files) to all secondary servers, and replace the content of the existing Web application folder with the new.
3. If Delinea Management Server (TMS) is installed and clustered, you need to copy the TMS directory to the secondary servers as well. The TMS directory is included by default for new installs of Secret Server 10.2 and above. TMS is used by advanced session recording and Privilege Manager. If the TMS folder and site does not exist in IIS, then no additional actions are needed beyond copying the Secret Server directory.
4. Start secondary servers and confirm they still work.

EFS and DPAPI Encryption

When upgrading, after the initial cluster configuration, you do not need to copy the `database.config` or `encryption.config` files to the other servers. If you need to copy those files because the database configuration changed and are using DPAPI, disable DPAPI encryption in Secret Server by going to **Admin > Configuration** and click **Decrypt Key to not use DPAPI** on the **Security** tab before copying those files to secondary servers.

Note: EFS encryption is tied to the user account running the Secret Server application pool, so it is not machine specific. Copying EFS encrypted files between Secret Server instances will not result in errors, but is not needed.

Upgrading Database Mirroring

1. If there is more than one Web server running Secret Server, ensure all instances are pointing to the same database.
2. Stop all but one of the web servers.
3. Perform the upgrade on that single instance.
4. Once upgraded and working, copy the Web application folder to all secondary servers.
5. Start the secondary servers, and confirm they work.
6. Ensure all instances are properly activated.
7. Ensure that the database changes have been replicated to the mirror database.
8. If the secondary Web server was pointing originally to the secondary database, adjust it to point back to the secondary database.

Upgrading Remote DR Instances

1. Perform the upgrade on one instance.
2. Backup that instance.
3. Copy the database backup to the remote DR instance.
4. Restore the database.
5. Once the instance is upgraded and working, copy the Web application folder (but not the `database.config` or `encryption.config` files) to the remote DR instance (overwriting the existing files).
6. Restart IIS or recycle the application pool running Secret Server on the remote DR instance.
7. Confirm that the remote DR instance is working correctly.

Error Conditions

Error that may arise:

- **Version does not match:** If a node is not properly updated from the source node after an upgrade, that node will not run because the application version does not match the database. The solution is to copy the application folder (minus the `database.config` or `encryption.config` files) to replace the files on the secondary server.

Upgrading Secret Server Without Outbound Access

Important: Upgrading to Secret Server version 8.9.000000 and above will require **Windows Server 2008 R2 or greater**.

Important: Upgrading to Secret Server version 8.5.000000 and above, there are changes in the .NET Framework version you will need to be aware of along with some additional steps in the upgrade process. For more information, see [Secret Server and Secret Server Cloud .NET Framework 4.8 Mandatory Upgrade](#).

Important: Upgrading to Secret Server version 10.0.000000 and above will require configuring integrated pipeline mode on the Secret Server Application Pool. Please see [this KB](#) for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication you will also need to update IIS authentication settings as detailed in [this KB](#). If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000 and above.

How Upgrades Work

Secret Server periodically polls our update server to detect updates. If your Secret Server is on an internal network that has no outbound access or goes through a proxy, Secret Server will not be able to perform updates automatically, therefore, outbound access to the below connections on your firewall is needed if you want to perform updates automatically:

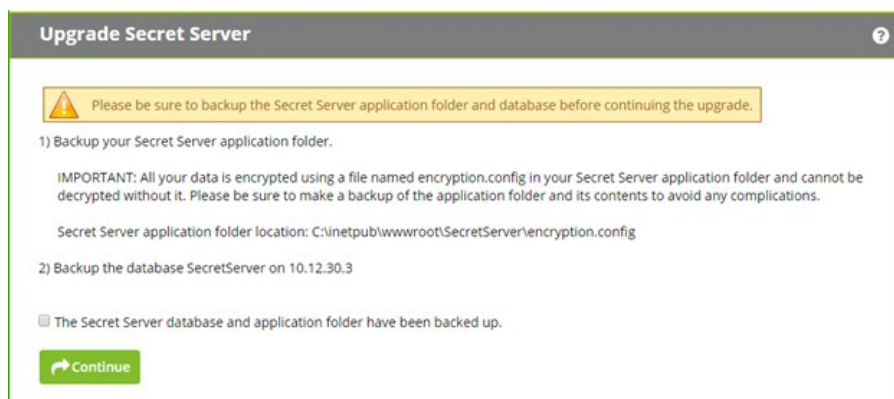
- d36zgw9sidnotm.cloudfront.net:443
- updates.thycotic.net:443
- updates.thycotic.net:80

The steps below can be used to perform an upgrade for versions 7.1.000015 and higher. If you have an older version of Secret Server, please contact Delinea technical support for assistance.

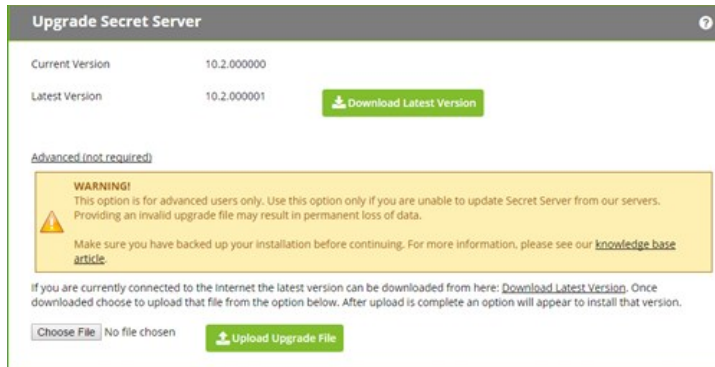
Procedure

Step 1: Open the Upgrade Secret Server Wizard

1. From a computer that does have outbound network access and Secret Server access, go to the Secret Server Upgrade page by browsing to: `http://<yourinstance>/Installer.aspx?patch=true` (filling in your Secret Server URL for <yourinstance>). The wizard appears:



2. Backup your Secret Server application folder and your Secret Server database.
 3. Click to select the **The Secret Server database...** check box on the page.
 4. Click the **Continue** button. The next page appears:
-



Step 2: Get and Upload the Latest .zip File

1. Download the latest version .zip file by clicking the **Download Latest Version** button on the installer page. The file name will appear something like Version_10_2_000000.zip. Note where you save it.

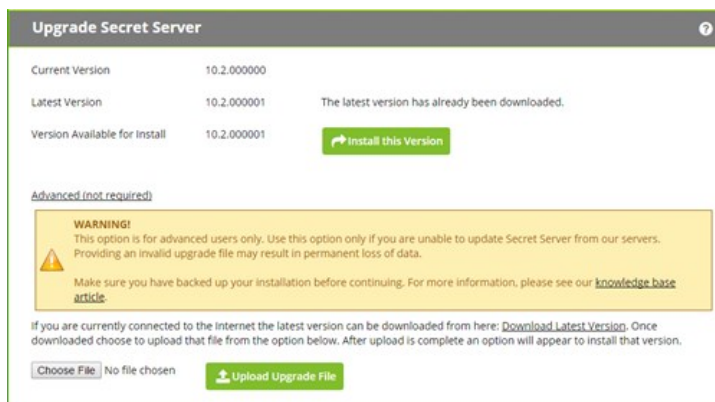
Note: You also can find the downloadable update files [below](#).

2. Click the **Choose File** button to select the Secret Server .zip file you just downloaded.

Note: To verify the file hashes for the latest version using the posted hash values, refer to [Secret Server Download Hashes](#).

Note: You should **not** use the fresh install SecretServer.zip or setup.exe that is first downloaded from [Delinea.com](#). Only use the Get Latest Version link—there is a difference between the upgrade file and fresh install zip.

3. Click the **Upload Upgrade File** button. You see a message confirming the file was successfully uploaded, and the Install This Version button appears.



4. Click the **Install this Version** button. The Upgrade Secret Server page appears (not shown).

Step 3: Upgrade Secret Server

1. Click the **Upgrade** button. The upgrade automatically processes and once it has finished you will see a confirmation page.
2. Click **Return to Home** to return to the dashboard.

Offline Installation Download Files

If you do not have access to another installation of Secret Server or you are upgrading from an earlier version, click one of the following links,

depending on your *current* installed version:

- [8.4.000003 or earlier](#)
- [8.4.000004 to 9.1.000000](#)
- [9.1.000001 to 10.9.000003](#)
- [10.9.000005 or later](#)

Upgrading to 10.9.000005/33

This upgrade of Secret Server is a two-step process where you first upgrade to version 10.9.000005 and then to 10.9.000033. The reasoning behind this is to provide a safeguard that warns you if the new .NET system requirement is not met *prior* to the installation making irreversible changes, potentially resulting in a non-functioning Secret Server installation. The second step to 10.9.000033 depends on the .NET update in the first step and deploys the new features for this release.

Important: Customers upgrading to 10.6.000000 or later and are using RabbitMQ, please see [Clearing RabbitMQ Message Queues](#).

Important: Upgrading to Secret Server version 10.0.000000 and above requires configuring integrated pipeline mode on the Secret Server Application Pool. Please see [Manual IIS Installation](#) for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication you will also need to update IIS authentication settings as detailed in [Configuring Integrated Windows Authentication](#). If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000 and above.

Important: If you have Privilege Manager installed, the Secret Server upgrade process will begin an upgrade for Privilege Manager as well.

Important: The **existing MS SQL database must be version 2012 or later for this upgrade**. Otherwise, if the upgrade fails and you attempt to roll it back, the previous installation will not work.

How Upgrades Work

Secret Server periodically polls the update server to detect new updates. If the "Allow Automatic Checks for Software Updates" option is enabled in the Admin > Configuration menu, you will see the "An update is available (xx.x.xxxxx)" link after logging in with an administrator account. The steps below can be used to perform an upgrade for versions 7.1.000015 and higher. If you have an older version of Secret Server, please contact Delinea technical support for assistance.

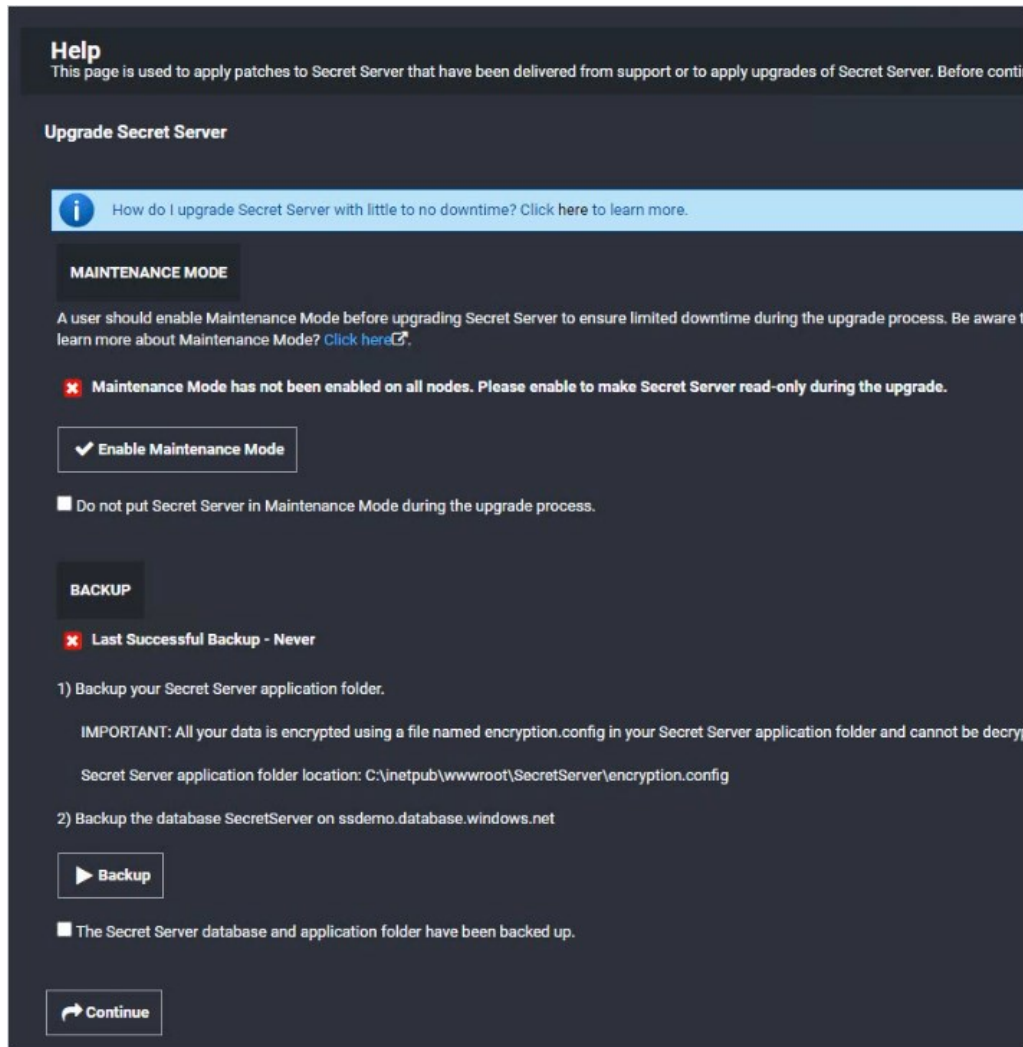
Before You Begin

1. Ensure you will have access to account credentials for the server hosting Secret Server AND the SQL Server instance hosting your Secret Server database.
2. Ensure you have a recent backup of the application files and database available.
3. If you use clustering, stop the application pools on all of the servers except the one being upgraded.
4. Conduct the [Secret Server and SSC .NET Framework 4.8 Upgrade](#).

How to Upgrade

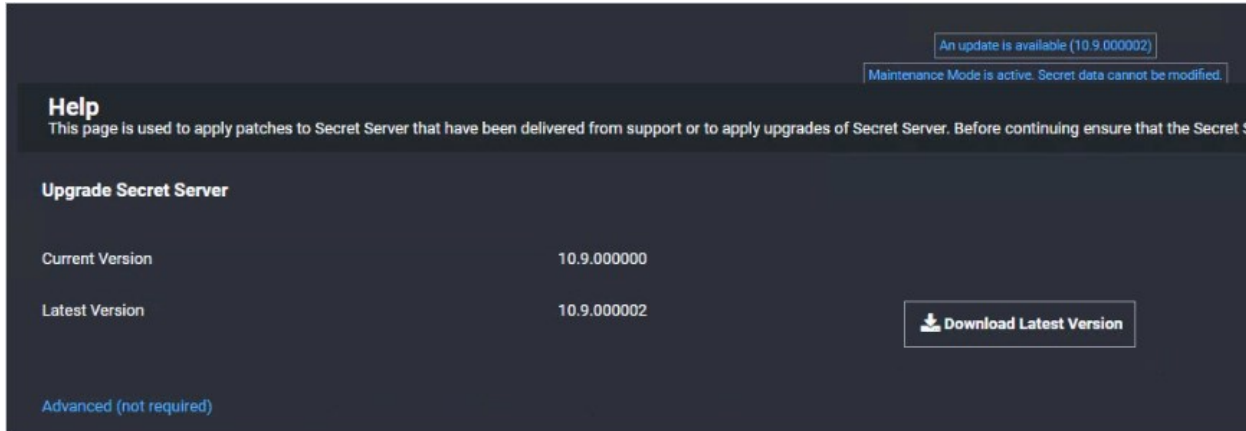
1. From a computer that has outbound network access, click on the upgrade link to go to: `http://<yourinstance>/SecretServer/Setup/Upgrade`. The Secret Server Setup Home page appears:

Note: If your computer does not have outbound network access, please see [Upgrading Secret Server Without Outbound Access](#).

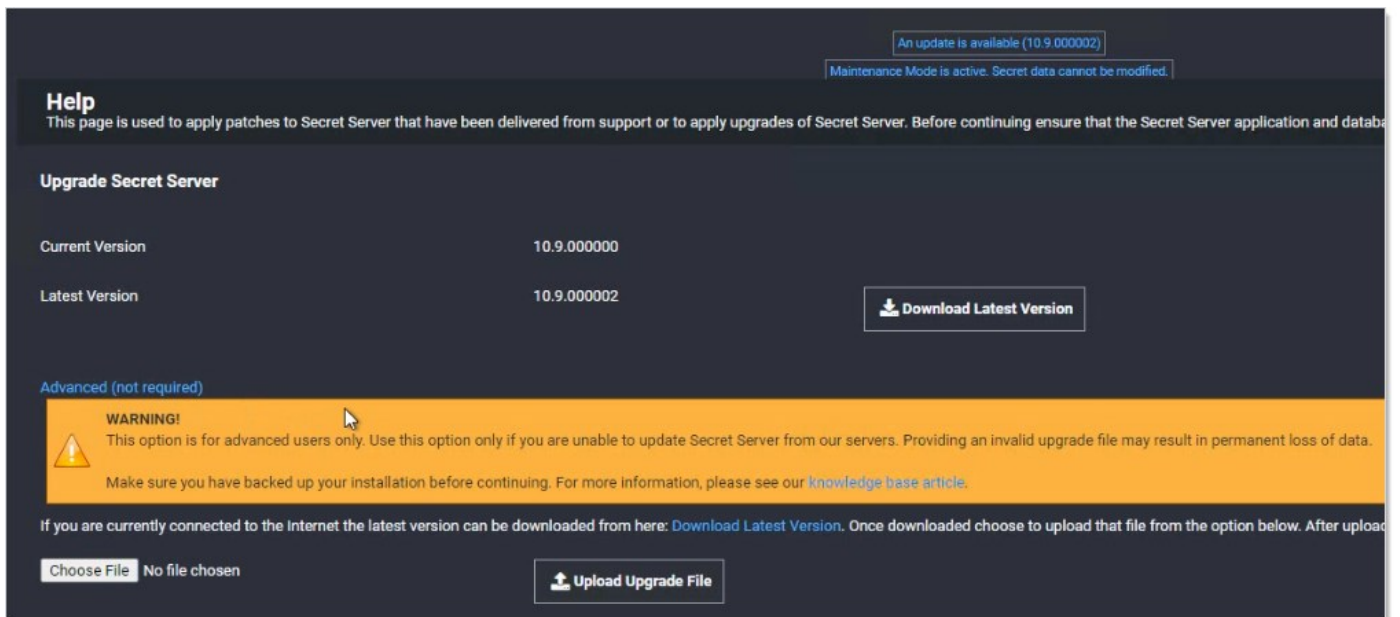


2. Download the two zip files (Version_10_9_000005.zip and Version_10_9_000033.zip), which contain the installs, using the link at the top of the page.
3. Click the **Enable Maintenance Mode** button to enter maintenance mode.
4. Backup your Secret Server application folder.

Important: All your data is encrypted the encryption.config file in your Secret Server application folder. **Your data cannot be decrypted without it.** Thus, it is critical that you make a backup of the application folder and its contents before proceeding.
5. Backup the database named SecretServer at the IP address listed.
6. When finished backing up both, click to select the **The Secret Server database and application folder have been backed up** check box.
7. Click the **Continue** button. The download page appears:



8. Click the **Advanced (not required)** link. The section appears:



9. Click the **Choose File** button and locate the Version_10_9_000005.zip file you downloaded.

10. Click the **Upload Upgrade** button. The file uploads, which can take several minutes. When the upload is complete the 10.9.000005 version appears:

An update is available (10.9.000002)
Maintenance Mode is active. Secret data cannot be modified.

Help

This page is used to apply patches to Secret Server that have been delivered from support or to apply upgrades of Secret Server. Before continuing ensure that the Secret Server

Upgrade Secret Server

Current Version	10.9.000000
Latest Version	10.9.000002
Version Available for Install	10.9.000005 (DEV_10_9_3_StepUpgrade-buildFix-Nov_24-21:20)

[Download Latest Version](#)
[Install this Version](#)

[Advanced \(not required\)](#)

11. Click the **Install this Version** button. The Install Secret Server Upgrade page appears:

An update is available (10.9.000002)
Maintenance Mode is active. Secret data cannot be modified.

Help

This page will install the updates to Secret Server. Log file information for the installation can be found at C:\inetpub\wwwroot\SecretServer\log.

Install Secret Server Upgrade

UPGRADE SECRET SERVER

This will begin the upgrade process. Please contact support with any issues.

[Upgrade](#) [Manual Rolling Upgrade](#)

12. Click the **Upgrade** button. An upgrade installation status page appears (not shown), and the upgrade begins. This will take several minutes. When done, an "Installation Complete" message appears.
13. Click the **Show Details** link to view the entire upgrade process:

An update is available (10.9.000002)

Maintenance Mode is active. Secret data cannot be modified.

Help

This page will report the progress of the Secret Server Upgrade process. During this process Secret Server may restart and this page may become unresponsive. This web page will be unavailable during the upgrade process.

More detailed logging information can be found at C:\inetpub\wwwroot\SecretServer\log\ss.log.

Secret Server Upgrade Installation Status

Process Started 11/25/2020 5:04 PM

✓ Setup	Show Details
✓ Remove Unchanged Files	Show Details
✓ Verify SQL Deltas	Show Details
✓ Upgrade Database	Show Details
✓ Update Script Resources	Show Details
✓ Update Web Resources	Show Details
✓ Create New Web Resources	Show Details
✓ Upgrade Themes	Show Details
✓ Update Application Files	Show Details
✓ Update Config Files	Show Details
✓ Clean Up	Show Details
✓ Turn Off Maintenance Mode	Show Details
✓ Upgrade Success	

Process Completed 11/25/2020 5:06 PM

[Return to Home](#)

14. Click the **Return to Home** button. You return to the previous page. The first installation of the two is complete. The new installation of Secret Server starts, which may take a bit of time. Secret Server opens to the All Secrets page:

All Secrets

28 Items Active ▾ All Templates ▾ 🔍

NAME	SECRET TEMPLATE	FOLDER	HEARTBEAT	OUT OF SYNC
AWS discovery	Amazon IAM Key	Personal Folders/SSAdmin/AWS Demo	Success	No
Azure	Web Password Extended	Personal Folders/SSAdmin/WPF demo		No
BBC login	Web Password Extended	Personal Folders/SSAdmin/WPF demo		No
Citi Test 1000	Web Password 1000 URL	URL testing		No
Citi Test 1000 with URLs	Web Password 1000 URL	URL testing		No
GCP Account Manager	Google IAM Service Account ...	Personal Folders/SSAdmin/GCP Demo	Success	No
GCP discovery	Google IAM Service Account ...	Personal Folders/SSAdmin/GCP Demo	Success	No
Launchers	Launcher demo	Personal Folders/SSAdmin/Launchers		No

15. If you intend to use Web clustering, proceed to [Upgrading Secret Server with Web Clustering](#).
16. Repeat the install procedure using the Version_10_9_000033.zip file you downloaded.
17. Once again, if you intend to use Web clustering, proceed to [Upgrading Secret Server with Web Clustering](#).

System Requirements

Please review the detailed [System and Memory Requirements for Secret Server](#). The *Minimum Requirements* are for trial, sandbox, and POC environments. The *Recommended Requirements* are for production deployments.

Hardware Requirements

Secret Server can be installed on a physical server or virtual machine.

If you would like to set up front-end (application) clustering, you need to have two or more servers available.

For testing of high availability for the SQL Server, you can use either existing Microsoft AlwaysOn infrastructure or database mirroring. If you choose to test this, this is something your database team needs to prepare in advance.

Software Requirements

Checklist

- Windows Server 2012 or newer (recommended) (one server, minimum)
- SQL Server (one instance, minimum)
- Application server prerequisites
- SSL certificate

SQL Server

You can create the SQL database in an existing SQL instance, or a new installation of SQL Server. For high availability, this needs to be a paid edition of SQL Server (not SQL Express). If you are using a new installation of SQL Server, please have this installed beforehand.

Detailed instructions for installation and configuration of SQL Server are included in one of the installation guides below (choose the guide matching the OS that SQL server will be installed on).

Application Server

We recommend installing Secret Server on Windows Server 2012 or greater. Include IIS, ASP.NET and .NET Framework. Refer to the System Requirements KB above to view prerequisite details.

Secret Server Major Browser Support Secret Server

Secret Server can accommodate most major browsers available today. This article covers each major browser and version supported by Secret Server, as well as support for the copy-to-clipboard feature.

For the best security, always keep your browser updated to the latest version.

Chrome	v25.0 and later	v25.0 and later; requires extension (all platforms)
Edge	v20.10240 and later	Not supported
Firefox	v57.0 and later	v57.0 and later; requires add-on (all platforms)
Internet Explorer	Not supported	Not supported
Safari	Not supported	v5.0 - v11; requires add-on for MacOS X Snow Leopard and later. Safari for Windows does not support copy-to-clipboard.
Opera	Not supported	Not supported

Using Chrome to Access Secret Server

The Chrome extension prompts users for "tabs" permissions, which Chrome uses to detect your browsing history, including what tabs you open and the URLs they open to. Secret Server uses the tabs permission function only to clear passwords on exit. No history is recorded.

Following these instructions ensures there is absolutely no residue or trace of Secret Server on the server.

Uninstalling Secret Server is a quick, three-step process:

1. Delete the database.
2. Delete the virtual directory.
3. Delete Secret Server files.

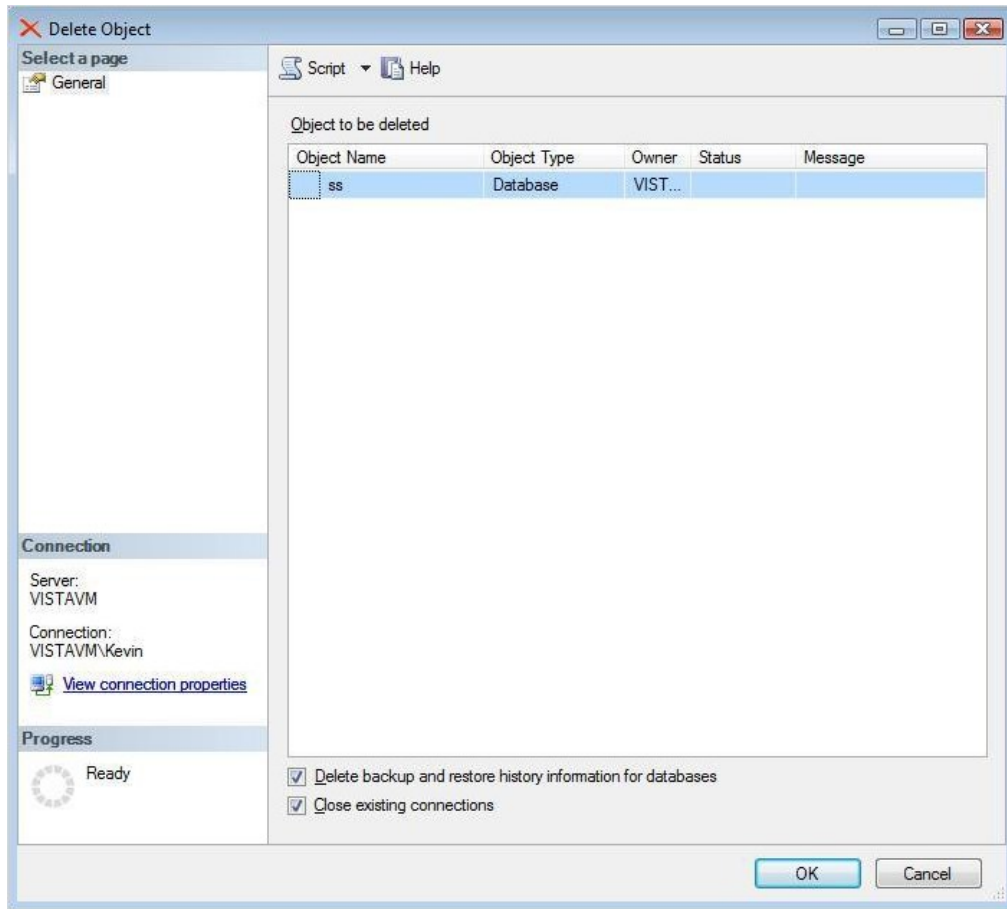
Task 1: Deleting the Database

Dropping the database deletes all of your data.

Warning: You cannot undo this procedure once you are done. We strongly suggest backing up the installation first in case you need to restore it.

Procedure:

1. Open the Microsoft Management Studio.
2. Connect to the database.
3. Locate your Secret Server database in the object explorer, which is normally in the **Databases** folder. If necessary press F8 to show the object explorer.
4. Right click the database and select **Delete**. The Delete Object dialog appears:

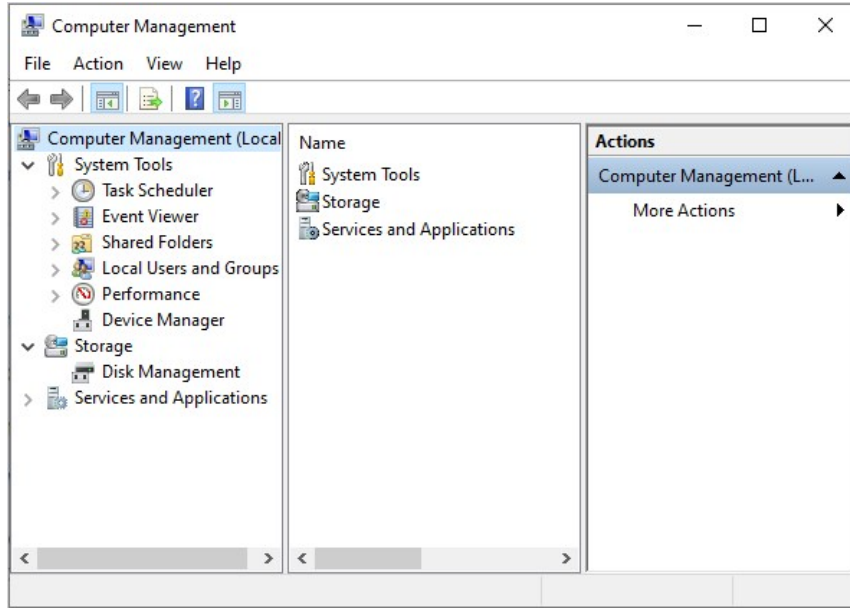


5. Ensure the **Drop Existing Connections** check box is selected. This disconnects all connections to the Secret Server database.
6. Ensure the **Delete backup and restore history information for databases** check box is selected.
7. Click the **OK** button. The database is permanently deleted.

Task 2: Deleting the Virtual Directory

If you installed Secret Server as a virtual directory, the virtual directory must be deleted first. If Secret Server is not configured as a virtual directory, skip this task.

1. In the search text box in your **Start Menu**, type Computer Management.
2. Click the **Computer Management** result. The Computer Management Console appears:



3. Click to expand the **Services and Applications** node.
4. Click the **Internet Information Services (IIS) Manager** node.
5. Click to expand the **Web Sites** subfolder.
6. Click to expand the Secret Server Web site.
7. Right click the virtual directory and select **Delete** or **Remove**. The directory is deleted.
8. (Optional) Delete ASP.NET's cached version of Secret Server:
 1. Open the directory C:\Windows\Microsoft.NET\Framework\<version number>\Temporary ASP.NET Files, substituting your ASP.NET version number.
 2. Delete the subfolder with the same name as your virtual directory.

Note: These files are not a security risk, but removing them eliminates any evidence that Secret Server was installed.

Task 3: Deleting Secret Server Files

Warning: The encryption.config file is crucial to restoring any backup. Ensure this file is backed up if you may want to restore Secret Server.

1. Locate the directory where Secret Server is was installed.
2. Click to select it.
3. Press **<Shift> + <Delete>** to **permanently** delete the files. Holding shift bypasses the recycle bin. Secret Server is now permanently removed from the system.

Note: Even "permanently" deleted files can sometimes be recovered with special tools. If that is a concern, we suggest using a file shredding application to delete the folder.

Decommissioning a Secret Server Node

To remove a Secret Server node from the list of active nodes, for example after you migrate an instance of Secret Server to another server, use the procedure below to decommission the node.


1. Click **Admin > Licenses**.
2. Click **View Server Activation (Advanced)** to open a list of Secret Servers designated as activated.

Licenses

You are currently licensed for 100 user(s). You currently have 17 enabled user(s).




Save To File < 1 to 5 of 5 >

LICENSE NAME	LICENSE KEY	DESCRIPTION
FOR DEVELOPMENT PURPOSES ONLY		Secret Server (100 users)

 Support licenses allow you to get free upgrades for new releases of Secret Server. You must purchase support for as many users as you are licensed.

[View Server Activation \(Advanced\)](#)

MACHINE NAME (ID)	JOINED	STATUS
	5/15 09:39 AM	Activated Decommission

 Back Install New License View Audit

3. In the row of the server you wish to decommission, click **Decommission**.

Ticketing System Integration

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server can allow users to enter a ticket number when viewing a secret. This number can be validated through a regular expression, and can also be marked as required, if needed. Secret Server can integrate with third party ticket systems. See below for more information.

You can add multiple ticket systems from the **Ticket System** tab. To add a new system, click **New Ticket System**.

You can make a select ticket system be Secret Server's default ticketing system by clicking on the link of the desired system, then clicking **Set as Default**.

Overview

Secret Server can require users to enter a ticket number when viewing a secret. Admins can track access to secrets based on an external ticket system. On the **Ticket System** tab of the **Configuration** page, an administrator can enter the settings to match the ticket system.

After the ticket system is enabled in Secret Server, a user can enter a ticket number on the Comment screen or the Request Access screen.

The secret needs to have Require Comment or Requires Approval for Access enabled to allow the user to enter a ticket number. When a ticket number is required, this secret setting is displayed as "Require Comment/Ticket Number" on the Security tab.

Configurable Settings

- **Auditing:** The ticket number appears in the audit log and can be queried in reports. If the **View Ticket URL** has been set, the log shows the ticket number as a hyperlink linking to the external ticket system. See the next bullet for more.
- **View Ticket URL Template:** The format of the URL to be used for viewing the ticket. This is placed in the audit log so you can easily view the corresponding ticket from Secret Server. The Ticket URL Format field can be edited on the "Ticketing System Integration" tab of the configuration page. In this field, the \$TICKETID parameter will be replaced by the ticket number that is entered by the user. For example, if you specify the template as `http://myticketingsystem/ticket.aspx?ticketid=$TICKETID`, and a user enters 5125-242 as the ticket number, a link will appear in the audit log to `http://myticketingsystem/ticket.aspx?ticketid=5125-242`.
- **Ticket Number and Reason Options:** This option allows fine-grained control of what the user must enter when Require Comment is enabled and ticket system integration is turned on.
 - **Reason Only Required:** Ticket number is optional, reason is required.
 - **Both Required:** Ticket number and reason are required.
 - **Ticket Number or Reason Required:** Either ticket number or reason must be entered.
 - **Ticket Number Only Required:** Ticket number is required, reason is optional.
- **Ticket Number Format Pattern (Regex):** A regular expression to use for validating the ticket number entered. This can help prevent typos in the number. For details on creating this expression, see the [Setting a Ticket Pattern Regex](#).
- **Ticket Number Label:** The text that displays next to the Ticket Number box on the Comment or Request Access page.

- **Ticket Number Validation Error Message:** The error message to display to the user when their entered ticket number fails the validation pattern regex.

Setting a Ticket Pattern Regex

If you are using ticketing system integration, you can set a ticket pattern on the Ticket System Integration tab of the Configuration page. If you do not want to restrict what ticket numbers a user can enter, you can leave the Ticket Number Validation Pattern (Regex) text box empty. If you do want to restrict it, you can enter a regular expression in the text box. The ticket number entered must match the regular expression.

For more information on regular expressions, please refer to [Regular Expressions](#).

If you are supported and need assistance setting up a validation pattern, feel free to email support@thycotic.com.

Here is an example for a ticket pattern that must be a valid number: `^[0-9]+$`

Secret Server can integrate into third-party ticket systems as well. Those supported are listed below. You can add multiple ticket systems to Secret Server by clicking New Ticket System. You can make a specific ticket system the default system used by Secret Server by clicking the System link and then clicking the Set as Default button.

The third-party integrations:

- [Atlassian JIRA](#)
- [BMC Remedy](#)
- [ManageEngine](#)
- [PowerShell](#)
- [ServiceNow](#)

Secret Server can integrate with Atlassian JIRA via PowerShell. This integration includes validating ticket numbers and their status, and adding comments.

For more information about integrating ticket systems with PowerShell, see [PowerShell Ticketing Integration](#).

Requirements

- PowerShell, see [Creating and Using PowerShell Scripts](#).
- [Configuring CredSSP for WinRM with PowerShell](#).

Note: Atlassian has deprecated TLS 1.0 and 1.1, and will support only TLS 1.2 and 1.3 going forward. See [Deprecating TLSv1 and TLSv1.1 for Atlassian Cloud Products](#).

Ticket Number Validation Pattern (Regex)

Before making a call to the ticket validation script, you can have Secret Server validate that the number matches a pattern. For example, you will probably have multiple JIRA projects and will want your users to specify these correctly. One regex would be to simply allow any project name followed by a dash and numbers:

```
^w{3}-\d+$
```

Or perhaps you want to specifically match projects that you know are real followed by a dash and numbers:

```
^((PRO1)|(PRO2))-d+$
```

For more information, see **Setting a Ticket Pattern Regex** on the [Ticketing System Integration](#) page.

Validating Ticket Status

You need to create a PowerShell script to retrieve and validate tickets. This integration assumes that the user will pass in the full ticket name, including the project name. For example: (PROJ-123). This could easily be extended so that multiple JIRA instances could be made for each project. In that case, you could have the user provide only the ticket number and pass in an argument to the script that specifies the project. This implementation also assumes that any ticket not in "Closed" status is invalid.

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$closedStatus = "Closed"
$fields = "status"
```

```
$p = $password | ConvertTo-SecureString -AsPlainText -Force
```

```
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$statusMethod = "$url/rest/api/latest/issue/$ticket"
```

```
function ConvertTo-UnsecureString([System.Security.SecureString][parameter(mandatory=$true)]$SecurePassword)
{
    $unmanagedString = [System.IntPtr]::Zero;
    try
    {
        $unmanagedString = [Runtime.InteropServices.Marshal]::SecureStringToGlobalAllocUnicode($SecurePassword)
        return [Runtime.InteropServices.Marshal]::PtrToStringUni($unmanagedString)
    }
    finally
    {
        [Runtime.InteropServices.Marshal]::ZeroFreeGlobalAllocUnicode($unmanagedString)
    }
}
```

```

function ConvertTo-Base64($string)
{
    $bytes = [System.Text.Encoding]::UTF8.GetBytes($string);
    $encoded = [System.Convert]::ToBase64String($bytes);
    return $encoded;
}

function ConvertFrom-Base64($string)
{
    $bytes = [System.Convert]::FromBase64String($string);
    $decoded = [System.Text.Encoding]::UTF8.GetString($bytes);
    return $decoded;
}

function Get-HttpBasicHeader($Credentials, $Headers = @{})
{
    $b64 = ConvertTo-Base64 "$($Credentials.UserName);$($ConvertTo-UnsecureString $Credentials.Password)"
    $Headers["Authorization"] = "Basic $b64"
    return $Headers
}

try
{
    $headers = Get-HttpBasicHeader $credentials
    $response = Invoke-RestMethod -Method Get -uri $getStatusMethod -Headers $headers -ContentType 'application/json'

    if($response.fields.status.name -eq $closedStatus)
    {
        throw "JIRA ticket ($ticket) is closed."
    }
}
catch
{
    $exception = $_.Exception
    if ($exception.Response.StatusCode.value__ -eq 404)
    {
        throw "JIRA ticket ($ticket) does not exist."
    }
}
}

```

Adding Comments to Tickets

To add comments to tickets, you will need to create the script below.

```

$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]

$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($user,$p)

function ConvertTo-UnsecureString([System.Security.SecureString][parameter(mandatory=$true)]$SecurePassword)
{
    $unmanagedString = [System.IntPtr]::Zero;
    try
    {
        $unmanagedString = [Runtime.InteropServices.Marshal]::SecureStringToGlobalAllocUnicode($SecurePassword)
        return [Runtime.InteropServices.Marshal]::PtrToStringUni($unmanagedString)
    }
    finally
    {
        [Runtime.InteropServices.Marshal]::ZeroFreeGlobalAllocUnicode($unmanagedString)
    }
}

function ConvertTo-Base64($string)
{
    $bytes = [System.Text.Encoding]::UTF8.GetBytes($string);
    $encoded = [System.Convert]::ToBase64String($bytes);
    return $encoded;
}

```



```
function ConvertFrom-Base64($string)
{
    $bytes = [System.Convert]::FromBase64String($string);
    $decoded = [System.Text.Encoding]::UTF8.GetString($bytes);
    return $decoded;
}

function Get-HttpBasicHeader($Credentials, $Headers = @{})
{
    $b64 = ConvertTo-Base64 "$($Credentials.UserName):$(ConvertTo-UnsecureString $Credentials.Password)"
    $Headers["Authorization"] = "Basic $b64"
    return $Headers
}

try
{
    $updateObject = @{'body'=$comment}
    $body = $updateObject | ConvertTo-Json
    $addComment = "$url/rest/api/latest/issue/$ticket/comment"
    $headers = Get-HttpBasicHeader $credentials
    $response = Invoke-RestMethod -uri $addComment -Headers $headers -Method Post -ContentType "application/json" -Body $body

    if ($response.body -ne $comment)
    {
        throw "There was an issue adding a comment to the ticket ($ticket)."
    }
}
catch
{
    $exception = $_.Exception
    if ($exception.Response.StatusCode.value__ -eq 404)
    {
        throw "The ticket ($ticket) does not exist.";
    }
    if ($exception.Response.StatusCode.value__ -eq 400)
    {
        throw "There was an issue adding a comment to the ticket ($ticket)."
    }
}

throw "There was an unhandled issue with adding a comment: " + $exception.ToString()
```

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Overview

Secret Server can integrate with BMC Remedy's Incident and Change Management. This integration includes validating ticket numbers, their status, and adding work detail items to the request.

The integration with BMC Remedy leverages the out-of-the-box, SOAP-based Web services that are installed with the ITSM product installation. These services must be installed on your mid-tier BMC Remedy server to allow for this integration if they are not already installed and configured.

Requirements

- BMC Remedy SOAP Web Services enabled
- A username and password that has access to execute the Web services. This can be set up in the developer studio by accessing the application in the navigator and viewing Permissions for the CHG_ChangeInterface_WS or HPD_IncidentInterface_WS. This user should also have access to query requests and add work items to requests for the appropriate module.
- Secret Server environment needs to be able to connect to the BMC Remedy Web services via port 80 or 443. SSL is highly recommended because the SOAP messages contain a username and password.

Configurable Settings

Validating Ticket Status

When a BMC Remedy request number is entered into Secret Server, the status of that request is retrieved to ensure that it is an open state. For example, if an incident number is entered that is in the "Closed" state, the user is informed that the ticket is closed.

Incident Management: Service Incident request cannot be closed or canceled. Change Management: Change management requests cannot be complete, closed, or canceled.

View Ticket URL Template

The format of the URL to be used for viewing the ticket. This is placed in the audit log so you can easily view the corresponding ticket from Secret Server. Depending on your version of BMC Remedy, the URL to link directly to a request may be slightly different.

Incident management:

```
https://<midtier_server>/arsys/forms/<servername>/SHR%3ALandingConsole/Default+AdmSearchTicketWithQual&F304255610='Incident Number'%3D%22$TICKETID%22
```

Change management:

```
https://<midtier_server>/arsys/forms/<servername>/SHR%3ALandingConsole/Default+AdmSearchTicketWithQual&F304255610='Change Number'%3D%22$TICKETID%22
```

Ticket Number Format Pattern (Regex)

Before even making a call to the BMC Remedy Web service, you can have Secret Server validate that the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure users enter the prefix. Some sample expressions to validate the ticket number are listed below:

Incident management: ^INC_CAL_[d]{7}\$

Change management: ^CRQ_CAL_[d]{7}\$

Ticket Number Validation Error Message

The error message to display to the user when their entered ticket number fails the validation pattern regex.

Service Endpoint URL

This is the URL for the SOAP-based Web services. Below are some samples for what is expected. You can find the actual endpoint using BMC Remedy Developer Studio and accessing the correct application from the AR System Navigator and viewing the Web services section of the application.

Incident management: HPD_IncidentInterface_WS

Change management: CHG_ChangeInterface_WS

System Credentials

Select or create a secret that contains the username and password for a user that has access to execute the SOAP Web services. The username and password are added to the authentication header for the SOAP request.

Authentication

If your installation of BMC Remedy uses an authentication server, enter it in this text-entry field. Most installations allow this text-entry field to be blank.

Add Comments to Ticket

Check this box if you want the comment that a user enters to be added to the request in BMC Remedy. This adds information such as the secret for which access is requested, who requested access, and the requester's comments.

Comment Work Type

When a comment is added to a request as a work item, the Work Item type is required. "General Information" is selected by default, but all default Work Type options are supported.

Testing Your Integration Setup

After configuring the ticket system (see configurable settings below), use the **Test Validation** button to verify that Secret Server can successfully access BMC Remedy. This button opens a dialog in which you can enter a ticket number from BMC Remedy. This validation process returns success or an error code. BMC Remedy may not return much detail in the error message so you need to look at the BMC Remedy API log to see a detailed error message, see the next section.

BMC Remedy Error Messages

When Secret Server calls the BMC Remedy SOAP-based web services, there are times that BMC will only return a 500 error without any details of the exception. You can see the details of this exception from the BMC Remedy server logs as described below:

1. Log on BMC Remedy as a user with access to the administrative console.
2. Navigate to **AR System Administration** from the main menu.
3. Navigate to **System > General > Service Information**.
4. Click the **Log Files** tab.
5. Click to enable the **API Log** check box.

6. Click the **Apply** button.
7. Once enabled, you can click **View** from this window to see the log or navigate to the mid-tier server's file system at the location specified. Details of the SOAP web service exception are written to the log file including a stack trace.

Secret Server can integrate with ManageEngine ServiceDesk Plus via PowerShell. This integration includes validating ticket numbers and their status, and adding comments (referred to as notes in ServiceDesk Plus).

For more information about integrating ticket systems with PowerShell, see [PowerShell Ticketing Integration](#).

Requirements

- PowerShell, see [Creating and Using PowerShell Scripts](#).
- Access to the REST API for your ManageEngine ServiceDesk Plus instance.
- [Configure CredSSP for use with WinRM/PowerShell](#).

Ticket Number Validation Pattern (Regex)

Before making a call to the ticket validation script, you can have Secret Server validate that the number matches a pattern. For more information, see **Setting a Ticket Pattern Regex** on the [Ticketing System Integration](#) page.

Validating Ticket Status

You need to create a PowerShell script to retrieve and validate tickets. This integration assumes that the administrator will set the technician key for accessing ServiceDesk Plus in the script. This could easily be extended to pass in the key as an argument so that it can be managed from the ticket system interface.

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$validStatus = "Open"
$fields = "status"
$technicianKey = "<YOUR API GUID>"

$p = $password | ConvertTo-SecureString -AsPlainText -Force

$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$getStatusMethod = "$url/sdpapi/request/$ticket"

try
{
    $postParams = @{OPERATION_NAME="GET_REQUEST";TECHNICIAN_KEY=$technicianKey}
    [xml]$response = Invoke-WebRequest -Uri $getStatusMethod -Method POST -Body $postParams -Credential $credentials

    if ($response.API.response.operation.result.status -ne "Success")
    {
        throw "Response not successful." + $response.API.response.operation.result.message
    }

    $statusNode = $response.API.response.operation.Details.ChildNodes | Where-Object { ($_.name -eq "status") }
    if($statusNode.value -ne $validStatus)
    {
        throw "Manage Engine Service Desk Plus ticket ($ticket) is not in Open status."
    }
}
catch
{
    if ($response.operation.result.message -eq "Invalid requestID in given URL")
    {
        throw "Manage Engine Service Desk Plus ticket ($ticket) does not exist."
    }
    throw "Manage Engine Service Desk Plus encountered an error: " + $response.operation.result.message
}
```

Adding Comments (Notes) to Tickets

To add comments (notes) to tickets you will need to create the script below. Other considerations are to pass in whether or not the note should be public.

```
$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]
$technicianKey = "<YOUR API GUID>"
$isPublic = 'true'

$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($user,$p)

#Clean comment input for use in XML
$comment = $comment.replace('&', '&amp;').replace("'", '&apos;').replace('"', '&quot;').replace('<', '&lt;').replace('>', '&gt;')

$inputData = @"
<Operation>
  <Details>
    <Notes>
      <Note>
        <isPublic>$isPublic</isPublic>
        <notesText>$comment</notesText>
      </Note>
    </Notes>
  </Details>
</Operation>
"@

$URI = "$url/sdpapi/request/$ticket/notes"
$postParams = @"{OPERATION_NAME='ADD_NOTE';TECHNICIAN_KEY=$technicianKey;INPUT_DATA=$inputData}"
$response = Invoke-WebRequest -Uri $URI -Method POST -Body $postParams
$xml]$responseContent = $response.Content

if ($responseContent.API.response.operation.result.status -ne "Success")
{
  $message = $responseContent.API.response.operation.result.message;
  throw "Unable to add comment to ticket ($ticket). Message: " + $message
}
```

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server can integrate with your ticketing system via PowerShell. This integration includes validating ticket numbers, their status, and adding comments. In our example we are connecting to a ServiceNow instance.

Note: See [Creating and Using PowerShell Scripts](#).

Configurable Settings

View Ticket URL Template

You can configure the view ticket URL if you have a web based ticketing system to allow easy access to link to your ticketing system from Secret Server.

Ticket Number Validation Pattern (Regex)

Before making a call to the PowerShell script you can have Secret Server validate the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure they enter this prefix. See [Setting a Ticket Pattern Regex](#).

Ticket Number Validation Error Message

The error message to display to the user when their entered ticket number fails the validation pattern Regex.

The PowerShell RunAs Credentials

In Secret Sever a domain credential is required to execute the PowerShell script. This is a required field.

System Credentials

The system credentials are specific to your ticketing system. Any secret using the Username and Password extending mapping can be used as your system credential. Additional arguments can be populated from field on this secret and reference in your script.

Validating Ticket Status

Overview

To validate tickets you will need to create a PowerShell script to retrieve and validate the ticket. The integration will use arguments to pass custom values to your script. By default we will map certain fields to the first set of arguments. The ticket number will be collected by user input and assigned to the first parameter. When you have your ticketing system credentials mapped to a secret and assigned to the "System Credentials" field in the ticketing system setup, Secret Server inserts UserName and Password as the second and third parameters.

Therefore, for the sample script below, the Ticket Status Script Arguments text box should be only contain \$url (which is also retrieved from the System Credentials secret), as \$ticket, \$user and \$password are supplied automatically by the system.

Sample Script

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$validStatus = "2"
$fields = "state"
$sp = $password | ConvertTo-SecureString -AsPlainText -Force
```

```
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$getstatusMethod = "$url/api/now/table/incident?sysparm_limit=10&sysparm_query=number=$ticket&sysparm_display_value=&sysparm_fields=$fields"
$response = Invoke-RestMethod $getstatusMethod -Method Get -ContentType 'application/json' -Credential $credentials
if($response.result.state -ne $validStatus)
{
    throw "Invalid State"
}
```

Adding Comments to Tickets

To add a comment to tickets, create another script to do so. Example:

```
$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$restEndpoint = "$url/api/now/table/incident?sysparm_limit=10&sysparm_query=number=$ticket&sysparm_display_value=&sysparm_fields=sys_id"
$response = Invoke-RestMethod $restEndpoint -Method Get -ContentType 'application/json' -Credential $credentials
$id = $response.result.sys_id
$updateObject = @{'work_notes'=$comment}
$body = $updateObject | ConvertTo-Json
$addComment = "$url/api/now/table/incident/$id"
$response = Invoke-RestMethod $addComment -Method Put -ContentType 'application/json' -Credential $credentials -Body $body
```

Adding Comments to a General Audit Log

In addition to adding comments to specific tickets, you may want general audit entries made in your ticket system. The arguments are passed in the following order.

```
$comment = $args[1]
$user = $args[2]
$password = $args[3]

## custom script here
```


Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Introduction

Secret Server can integrate with ServiceNow's Incident and Change Management service. This integration includes validating ticket numbers, their status, and adding Work Detail items to the request. The integration with ServiceNow leverages the out-of-the-box REST-based Web services.

Requirements

- ServiceNow instance running the Eureka version or later with REST services enabled.
- A username and password that has access to execute the REST services, specifically GET and MODIFY on the following tables: Change Request and Incident.
- The Secret Server environment needs to be able to connect to the ServiceNow Web services via port 80 or 443. SSL is highly recommended because the REST messages authenticate with a username and password.

Configurable Settings

View Ticket URL Template

The format of the URL to be used for viewing the ticket. This appears in the audit log so you can easily view the corresponding ticket from Secret Server.

Incident management: `https://<instance name>.service-now.com/nav_to.do?uri=incident.do?sysparm_query=number=$TICKETID`

Change management: `https://<instance name>.service-now.com/nav_to.do?uri=change_request.do?sysparm_query=number=$TICKETID`

This field specifies the URL that will be used when displaying a link to a ticket in the audit log. In this field, the \$TICKETID parameter will be replaced by the ticket number that is entered by the user.

For example, if you specify the **View Ticket URL Template** as `http://myticketingsystem/ticket.aspx?ticketid=$TICKETID`, and Bob enters 5125-242 as the ticket number, a link will appear in the audit log to `http://myticketingsystem/ticket.aspx?ticketid=5125-242`.

Ticket Number Format Pattern (Regex)

Before even making a call to the ServiceNow Web service you can have Secret Server validate the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure they enter this prefix. Some sample expressions to validate the ticket number are listed below:

Incident management: `^INC[d]{7}$`

Change management: `^CHG[d]{7}$`

Ticket Number Validation Error Message

The error message to display to the user when their entered ticket number fails the validation pattern regex.

Domain Name

This is the domain name of your instance in the format `<instance name>.service-now.com`, for example `dev5859.service-now.com`.

System Credentials

Select or create a secret that contains the username and password for a user that has access to execute the REST Web services. Secret Server uses these credentials to authenticate to ServiceNow.

Add Comments to Ticket

Check this box if you want the comment that a user enters to be added to the request in ServiceNow. This adds information such as the Secret to which access is requested, who requested access, and their comments. The comment is added as a work note in the activity section of the request.

Testing your Integration Setup

After configuring the ticket system, use the **Test Validation** button to verify Secret Server can successfully access ServiceNow. This button opens a dialog in which you can enter a ticket number from ServiceNow. This validation process either succeeds or shows an error code.

Troubleshooting and Notices

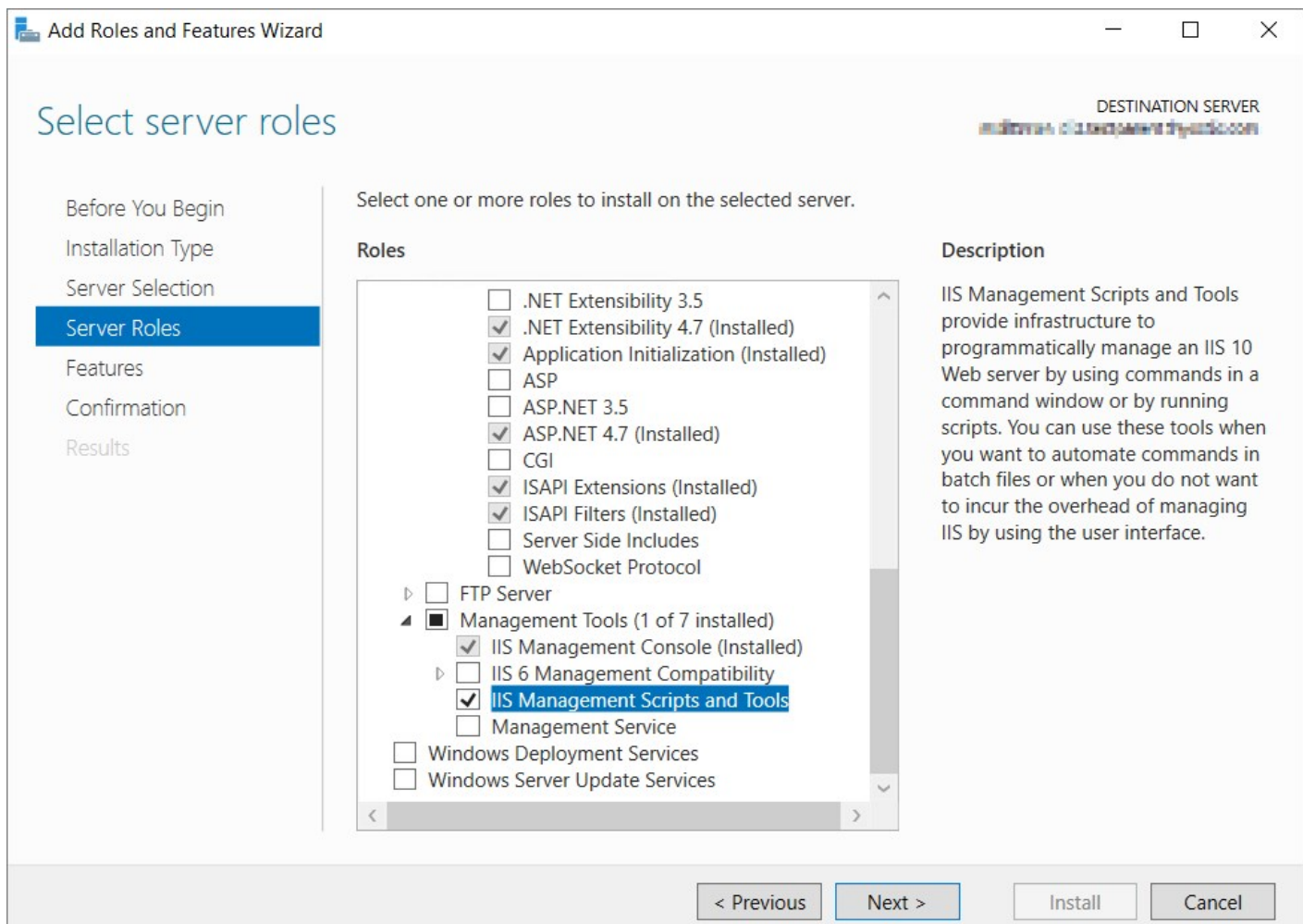
Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

This section contains common troubleshooting issues, workarounds, and technical notices.

Note: This section is a work in progress. It does **not** contain a complete set of Secret Server troubleshooting and workaround articles.

As of Secret Server version 8.9.000000, you can perform discovery over distributed engines. However, application pool scanning cannot use the discovery source's identity for impersonation. The distributed engine service must be run as an account with sufficient privileges to scan the application pools on its own.

Important: For discovery to work on application pool service accounts on IIS 7+, the "IIS Management Scripts and Tools" feature must be installed on the DE or Web server running the code:



Check that the following privileges are set:

- Enable the account to log on as a service.
- Grant the account read, write, and execute privileges to the entire distributed engine installation directory and sub-folders.
- Add the account to the administrators group on each computer that will be scanned.

80070005 Error

An 80070005 error similar to the one below indicates that the distributed engine service account does not have sufficient privileges on the machine being scanned.

Exception: Retrieving the COM class factory for remote component with CLSID {2B72133B-3F5B-4602-8952-803546CE3344} from machine <MACHINE> failed due to the following error: 80070005

To resolve the error, add the account to the administrators group on that machine. When changing the user of the distributed engine service, Secret Server will interpret the service as a new engine. You will need to delete the old engine and activate the new engine in Secret Server, if applicable.

The typical workflow for changing the account logon for the distributed engine service is as follows:

1. Go to **Admin > Distributed Engine > Manage Sites** and deactivate distributed engine in **Site**.
2. Delete the engine by selecting the trash can icon next to the engine.
3. Go to the distributed engine server and stop the distributed engine service, or end the process from the Task Manager.
4. Change the Log On option – this must be the same account used for your discovery.
5. Start the distributed engine service.
6. In the Secret Server application, go to **Admin > Distributed Engine > Manage Sites** and select your site (or /SiteView.aspx? SiteId=X page) then refresh the page.
7. Activate distributed engine by clicking the check mark icon next to it (or from the /ManageEngines.aspx page).
8. Wait 3-5 minutes and click Validate Connection. You should see a successful connection.

Secret Server requires the application pool to have the "load user profile" setting enabled. Secret Server will report a critical alert to notify admins if this setting is not enabled.

Note: The site will load to give access to secrets but many internal operations will not function correctly so we recommend fixing the issue as soon as possible.

Note: This applies to version 10.2 and later.

Steps to enable the "load user profile" setting:

1. On each Web server that is running Secret Server, open IIS Manager.
2. Under the **Application Pool** node on the left, select **Secret Server**.
3. On the right-hand panel, select **Advanced Settings** to get to the full properties.
4. Scroll to the **Load User Profile** setting in the **Process Model** section.
5. Set **Load User Profile** to **True**.
6. Click the **OK** button.
7. Perform an iisreset on the server:
 1. Open a Windows command prompt as an administrator.
 2. Type iisreset.
 3. Press the **<Enter>** key.

Overview

When using IIS version 7.0 and above, by default, the worker process terminates after a period of inactivity. If Secret Server is in its own application pool, the application pool will stop after a period of no requests. To make sure that the application pool associated with Secret Server does not stop when idle:

- Set the idle time-out to 0 minutes.
- Set the regular time interval to 0.
- Ensure there are no specific times scheduled for recycling.

Additionally, by default, IIS launches a worker process when the first request for the Web application is received. So if the Secret Server application takes a long time to start, we recommend launching the worker process as soon as IIS is started by setting the start mode to AlwaysRunning to launch the worker process for the Secret Server application pool as soon as IIS is started.

Procedure

To change IIS advanced settings:

1. Open **Internet Information Server (IIS) Manager**: On the taskbar, click **Server Manager**, click **Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, expand the server name.
3. Click **Application Pools**.
4. Locate the application pool Secret Server is running as. To determine this:
 1. Expand **Sites** at the left, then find the website Secret Server is running on.
 2. Click on the Secret Server website or virtual directory (if it is running on one).
 3. Click **Basic Settings** on the right panel. This indicates Secret Server's application pool.
5. Right-click the application pool, and select **Advanced Settings**. The Advanced Settings panel appears.
6. Go to the **(General)** section.
7. Set **Start Mode** to **AlwaysRunning**.
8. Set **Maximum Worker Processes** to **1**.
9. Go to the **Process Model** section.
10. Set **Idle Time-out (minutes)** to **0**.
11. Go to the **Recycling** section.
12. Set the **Regular Time Interval (minutes)** to **0**.
13. Select **Specific Times**.
14. **Either** click the > expander arrow to see if there is time specified below. **Or** click the ... to see if there are any values in the **TimeSpan Collection Editor** dialog box. If so, clear it out.
15. Click the **OK** button. The dialog closes.

16. Click the **OK** button.

There are several ways to find the version number of your Secret Server release. The procedures in this section apply to the Professional and Platinum editions of the Secret Server on-premises application.

Through Secret Server

You can find the version number of your Secret Server release from any Secret Server page. Just click your user icon in the top right corner, and a box opens displaying information including the full version number.

Through Windows File Explorer

1. Log into the web server where the Secret Server application is installed.
2. Open Windows File Explorer.
3. Navigate to `C:/inetpub/wwwroot/SecretServer/bin`
4. Right-click the `Thycotic.ihawu.Business.dll` file and select **Properties**.
5. In the Properties dialog, click the **Details** tab, which displays information including your product version.

Through SQL

1. Log into your Secret Server database server.
2. Open and log into your Microsoft SQL Server Management Studio application.
3. Click **Connect**.
4. At the top level of the **Object Explorer** pane, click the **+** symbol next to `servername\instance`.
5. Click the **+** symbol next to **Databases**.
6. Right-click your Secret Server database and select **New Query** from the dialog.
7. In the SQL Query pane that opens, paste the following query:

```
`SELECT * from tbVersion`  
`ORDER BY CAST(REPLACE(VersionNumber, '.', '' ) AS INT) DESC;`
```
8. Click **Execute**.
9. On the **Results** tab below the SQL Query pane, confirm that your query executed successfully and find the version number of your current Secret Server release in the top row.

Through the API

You can also use the REST API to check the version of Secret Server using the endpoint `GET /v1/version`.

An HTTP 404.2 error code is received when ISAPI/CGI Restrictions are preventing the .NET Framework 4.5.1 from running.

Resolution

1. Open Internet Information Services.
2. Select the Server in the left tree view.
3. In the **IIS** section, open ISAPI and CGI Restrictions.
4. For all items beginning with **ASP.NET v4.0**, right-click the item and select **Allow**.

Error

After upgrading Secret Server and changing the CLR version, when attempting to load Secret Server, you receive the following error in Internet Explorer:

HTTP Error 404.17 - Not Found The requested content appears to be script and will not be served by the static file handler

Resolution

This error can be caused by ASP.NET 4.5 not being correctly registered on the server. To correct this:

Windows Server 2012 or 2012 R2

Use the Server Manager to install ASP.NET 4.5.

1. Open the Server Manager.
2. Select **Manage > Add Roles and Features**. The Add Roles and Features wizard appears.
3. Click the **Next** button. The Select Installation Type page appears.
4. Click to select the **Role-based or feature-based installation for your server** selection button.
5. Click the **Next** button twice. The Select Server Roles page appears.
6. Click to select the **Web Server (IIS)** check box in the **Roles** list.
7. Click the **Next** button until you arrive at **Role Services** under **Web Server (IIS)**.
8. Drill down to **Web Server > Application Development** in the **Role Services** list.
9. Click to select the **ASP.NET 4.5** check box.
10. Click the Next button until you arrive at the final page.
11. Click the **Install** button.
12. Once installed, follow the resolution instructions in [HTTP Error 404.2 - ISAPI and CGI Restrictions](#) (KBA) to ensure ASP.NET 4.0 is allowed to execute in IIS.

Error

Error Text

Oracle error: Invalid length for connection option 'Data Source', maximum length is 128.

Scenario

You run an SQL script that attempts to connect to an Oracle DB using the `System.Data.OracleClient.OracleConnectionStringBuilder.set_DataSource(String value)` call. The script runs from the local site as a database account, and an error occurs.

The error stack trace from the logs looks similar to this:

```
2020-05-20 11:24:16,149 [CID:] [C:] [TID:PriorityScheduler Thread @ Normal] ERROR Delinea.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner - Invalid length for connection option 'Data Source', maximum length is 128.
```

```
System.ArgumentException: Invalid length for connection option 'Data Source', maximum length is 128.
```

```
at System.Data.OracleClient.OracleConnectionStringBuilder.set_DataSource(String value)
```

```
at System.Data.OracleClient.OracleConnectionStringBuilder.set_Item(String keyword, Object value)
```

```
at System.Data.Common.DbConnectionStringBuilder.set_ConnectionString(String value)
```

```
at Delinea.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner.BuildConnection(DbProviderFactory dbProviderFactory, String connectionString, IDictionary`2 additionalConnectionItems)
```

```
at Delinea.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner.ExecuteScript(String providerName, String connectionString, IDictionary`2 additionalConnectionItems, String commandText, IEnumerable`1 parameters)
```

Resolution

This error is an Oracle issue. Oracle has built-in string length restrictions on data source strings. One possible workaround is to use IP addresses for the Oracle server instead of FQDNs. Failing that, search the Internet for "Invalid length for connection option 'Data Source', maximum length is 128" to discover the latest information and workarounds.

Relevance

This Delinea **technical issue** knowledge base article is relevant to:

- Product(s): Secret Server using jQuery 3.2.1
- Version(s): 10.7
- Edition(s): All

Technical Issue

Secret Server 10.7 uses jQuery 3.2.1, which is listed as vulnerable to the jQuery CVE-2019-11358 security issue on the [Common Vulnerabilities and Exposures \(CVE\) list](#).

Resolution

Delinea removed the jQuery vulnerability from Secret Server's copy of jQuery v3.2.1 by applying a patch (see [Related Articles and Resources](#)).

To verify the fix:

1. Navigate to `https://<your_secret_server_URL>/assets/libs/jquery-3.2.1.js`
2. Open the file in a text editor.
3. Search for the string `proto` in the code: ...

```
// Prevent Object.prototype pollution
// Prevent never-ending loop
if ( name === "__proto__" || target === copy ) {
  continue;
}
```

4. If the string appears, the patch has been applied.

Related Articles and Resources

- [NIST website for CVE-2019-11358](#)
- [GitHub commits on the fix](#)

Note: The commit shows two files, the top file is the security fix, and the bottom file is a unit test for the fix. Secret Server does not ship with any jQuery unit tests as found in that second file.

- [Common Vulnerabilities and Exposures \(CVE\) list](#)

Relevance

This Delinea **technical issue** knowledge base article is relevant to:

- Product(s): Secret Server using jQuery 3.2.1
- Version(s): 10.8.000004
- Edition(s): All

Technical Issue

Secret Server 10.8.000004 uses jQuery 3.2.1, which is listed as vulnerable to the jQuery CVE-2020-11022 security issue on the [Common Vulnerabilities and Exposures \(CVE\) list](#).

Resolution

Delinea removed the jQuery vulnerability from Secret Server's copy of jQuery v3.2.1 by applying a patch (see [Related Articles and Resources](#)).

To verify the fix:

1. Navigate to `https://<your_secret_server_URL>/assets/libs/jquery-3.2.1.js`
2. Open the file in a text editor.
3. Search for the string `htmlPrefilter` in the code (line 5919):

```
jQuery.extend( {  
  htmlPrefilter: function(html) {  
    return html;  
  }  
}
```

4. If the string appears, the patch has been applied.

Related Articles and Resources

- [NIST website for CVE-2020-11022](#)
- [GitHub commits on the fix](#)

Note: The commit shows multiple files, the top file is the security fix, and the bottom files are unit tests for the fix. Secret Server does not ship with any jQuery unit tests.

- [Common Vulnerabilities and Exposures \(CVE\) list](#)

Detection

During a security review of Secret Server on June 4, 2019, an internal security team found the security issue described below. The issue was also detected later by an internal team in Password Reset Server.

The Security Issue

An attacker with administrator permissions could modify the input field data in one specific location to execute a SQL injection attack against Secret Server or Password Reset Server. This means that the attacker could append, modify or delete data in the Secret Server or Password Reset Server SQL databases, and upgrade their access to code execution on the SQL server.

Common Vulnerability Scoring System Version 3.0

The CVSS score for this issue is 9.1. The CVSS vector string is CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

Products Affected

- Secret Server On-Premises version 10.6.000026 and earlier.
- Secret Server Cloud, which was updated June 15, 2019 to permanently remove this issue for all customers.
- Password Reset Server earlier than version 5.1.000005.

Recommended Actions

- All Secret Server On-Premises customers should upgrade to version 10.6.000027 or later.
- All Password Reset Server customers should upgrade to version 5.1.000005 or later.
- Secret Server Cloud users do not need to take any action.

Note: This topic is for upgrades of Secret Server from a version earlier than 10.2.

Initial Troubleshooting

Changes to the `saml.config` were introduced in Secret Server 10.2. Secret Server should automatically convert the existing `saml.config` to the latest format. If it does not:

1. Ensure that the application pool has write access to the `saml.config` file.
2. Restart the applicationpool in IIS and try to log in again.
3. If Secret Server is running in a clustered environment:
 1. Copy the `saml.config` from the Web node that was upgraded to the remaining web nodes.
 2. Restart their application pools in IIS.

If that does not resolve the issue or Secret Server is not running in a clustered environment, there may be some other issue that prevented the `saml.config` from converting successfully during the upgrade. Please contact [Technical Support](#) for assistance.

Note: See the [Configuring SAML in Secret Server](#) article for more information on configuring your `saml.config` in 10.2

Additional Troubleshooting

If the `saml.config` is not loading properly, there are a few possibilities:

- The `saml.config` file is invalid. Ensure that it contains valid XML. Element and attribute names are case sensitive. Ensure that the elements and attributes names and value are valid for SAML configuration.

Note: See the `saml.config.template` file in Secret Server's root folder for guidance on which elements and attributes can be used.

- Secret Server is running in a clustered environment and some nodes are not yet configured. Copy the `saml.config` from the functioning Web node to all of the remaining Web nodes and restart their Application Pools in IIS.

Restart the Application Pool in IIS any time changes are applied to the `saml.config` file. If issues remain after following these steps, please contact [Technical Support](#) for assistance.

Google Authenticator relies on time to create tokens. If Secret Server's clock is inaccurate or is not synchronized with devices running Google Authenticator, user token validation may fail when enrolling or logging in.

Solution A (preferred)

Ensure that Secret Server's clock is accurate and synchronized with the device running Google Authenticator. Set the web servers to synchronize their clocks with an accurate domain controller clock or with an NTP server.

Solution B

By default the leniency value for token time accuracy is zero, which means the token supplied must be accurate. Follow the steps below to configure Secret Server with higher leniency to accept tokens with times that are slightly behind or slightly ahead.

1. Open the `web-appSettings.config` file and add the following key between the `appSettings` tags.

```
<add key="TOTPLeniency" value="0 or greater value here" />
```

2. Change the leniency value. We recommend setting this value to no higher than 2.
3. Recycle your IIS application pool. You must recycle your IIS application pool for the setting to take effect.

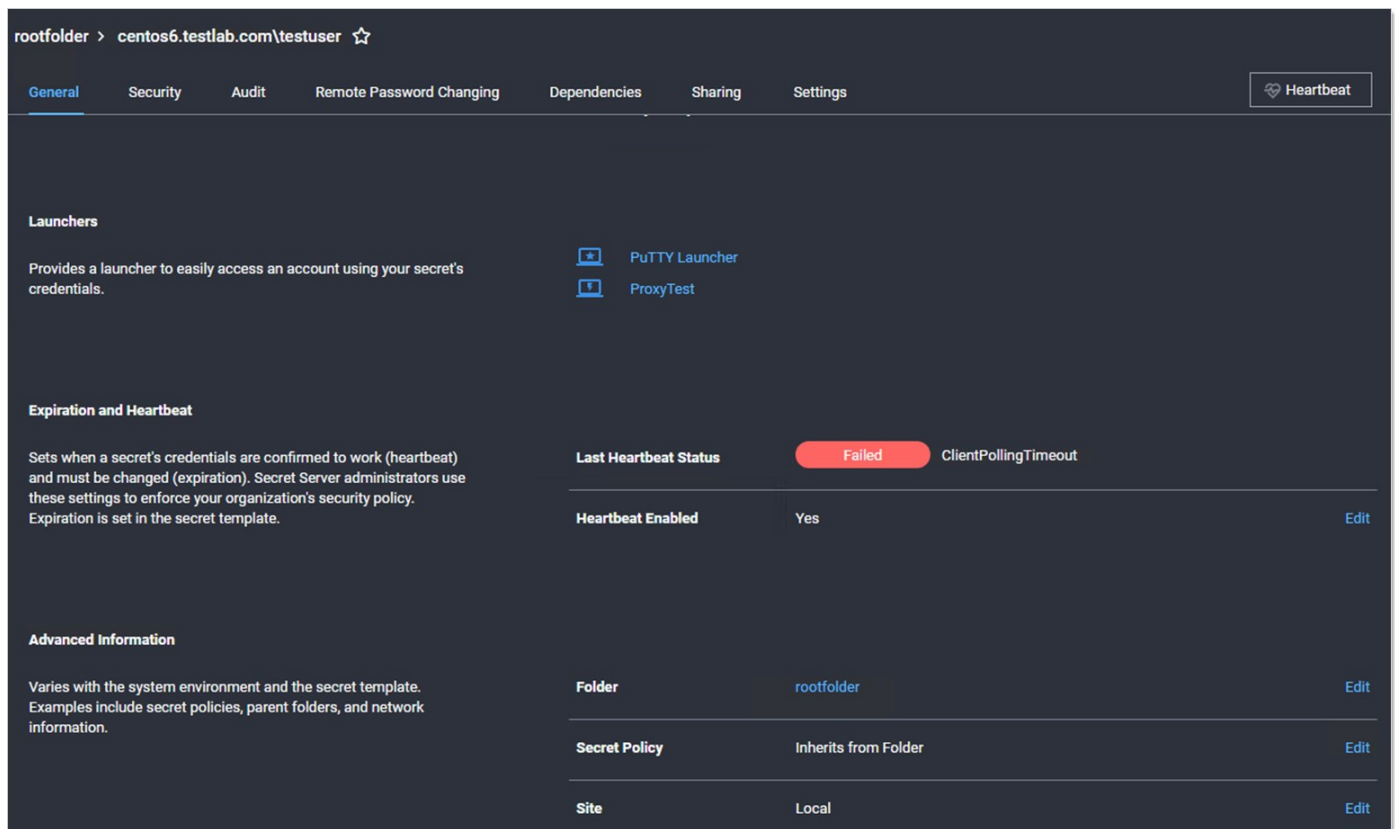
When using Secret Server for SSH password rotation, you may encounter errors when changing a secret. This article helps the reader troubleshoot the configuration of Remote Password Changing (RPC) in Secret Server to avoid errors.

Note: See [Troubleshooting SSH Issues](#) for other SSH issues.

Step 1: Verifying Ports and Connectivity

To determine if the heartbeat issue is outside of Secret Server:

1. Open the secret which is failing Remote Password Changing in Secret Server.

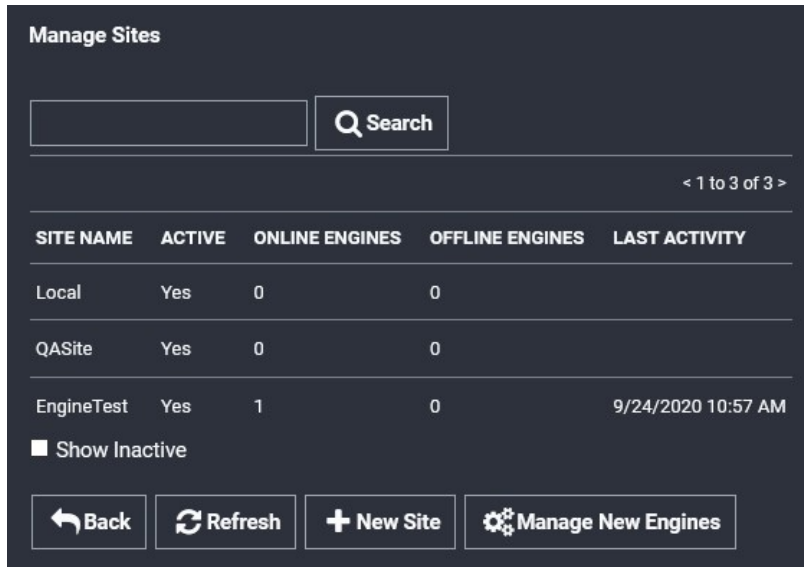


The screenshot shows the configuration page for a secret in Secret Server. The breadcrumb is 'rootfolder > centos6.testlab.com\testuser'. The 'General' tab is selected. A 'Heartbeat' button is visible in the top right. The 'Launchers' section lists 'PuTTY Launcher' and 'ProxyTest'. The 'Expiration and Heartbeat' section shows 'Last Heartbeat Status' as 'Failed' with a 'ClientPollingTimeout' value. 'Heartbeat Enabled' is set to 'Yes'. The 'Advanced Information' section shows 'Folder' as 'rootfolder', 'Secret Policy' as 'Inherits from Folder', and 'Site' as 'Local'.

Expiration and Heartbeat	
Last Heartbeat Status	Failed ClientPollingTimeout
Heartbeat Enabled	Yes Edit

Advanced Information	
Folder	rootfolder Edit
Secret Policy	Inherits from Folder Edit
Site	Local Edit

2. Scroll down to the **Advanced Information** section. You may have to click the **Advanced** link.
3. Note the **Site** parameter.
4. Go to **Admin > Distributed Engine**.
5. Click the **Manage Sites** button. The Manage Sites page appears:



- Click the **Site Name** link for the site. The Site View page appears.
- Note the **Processing Location** parameter for the site.
- If the processing location is **Local** and website processing is enabled, do your testing on the Secret Server application server. If it is **Distributed Engine**, do your testing on the distributed engine machine.
- In PowerShell run one of the following command for the machine you are trying to connect to from the secret:

```
Test-NetConnection -ComputerName <computer_name> -Port 22
```

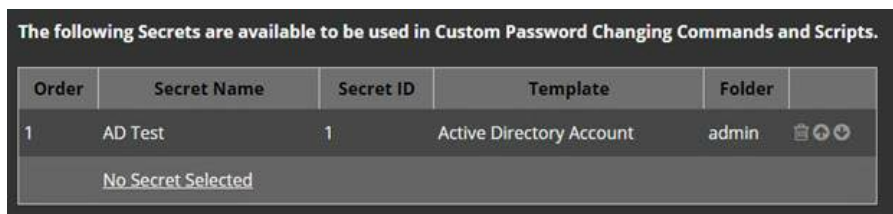
Note: If you chose a custom port, note it—that port will need to be changed on the RPC too.

- If the test was successful, proceed to the next step. If it was not successful, contact your networking team to open the port and test the connectivity. They can refer to [Ports Used by Secret Server](#).

Step 2: Testing Heartbeat and RPC in Secret Server

Procedure:

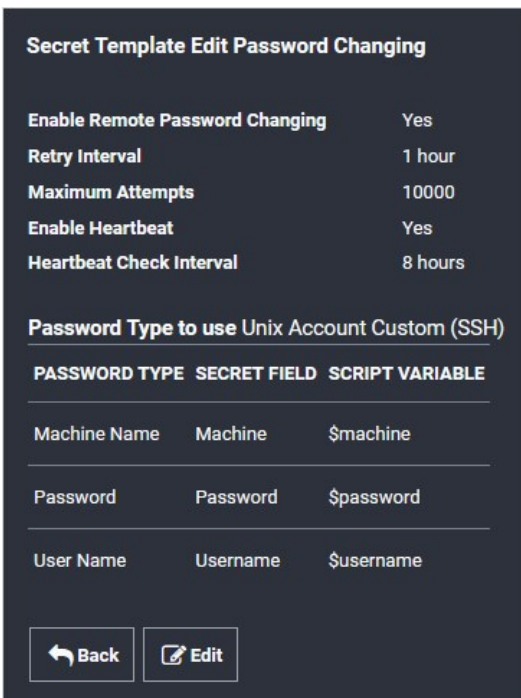
- Return to the secret on Secret Server.
- Click the **Remote Password Changing** tab of the secret (not shown).
- Check the **Associated Secret** section to see if there is an associated account set on the secret for use with RPC:



- Return to the **General** tab for the secret:

		Edit All
Secret Name *	centos6.testlab.com\testuser	Edit
Secret Template	Unix Account (SSH)	Edit
Machine *	centos6.testlab.com	Edit
Username *	testuser	Edit
Password *	***** Show	Edit
Notes		Edit
Private Key		Edit
Private Key Passphrase	***** Show	Edit

5. Note the **Secret Template** type.
6. Determine the password type for the template:
 1. Go to **Admin > Secret Templates**.
 2. Click to select the desired template in the dropdown list.
 3. Click the **Edit** button. The Secret Template Designer page appears (not shown).
 4. Click the **Configure Password Changing** button at the bottom of the page. The Secret Template Edit Password Changing page appears:



- Note the password types used, the applicable secret field, and the equivalent script variable. These indicate reserved variables that reference fields in the secret, in this case, \$USERNAME, \$MACHINE and \$PASSWORD. You will need to test your script using known-good values for these.
- Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears (not shown).
- Click the **Configure Password Changers** button. The Password Changes Configuration page appears:

Password Changers Configuration

PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (Telnet)	SSH Local Account	Yes
Cisco Enable Secret Custom (SSH)	SSH Local Account	Yes
Cisco Enable Secret Custom (Telnet)	SSH Local Account	Yes

9. Click the name link for the same password changer. The password changer page for that changer appears:

Unix Root Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$(CURRENTPASSWORD)	Privileged password	2000
3	whoami	Get name of account	2000
4	\$(CHECKFOR \$USERNAME)	Check the privileged login worked	2000

Password Change Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$(CURRENTPASSWORD)	Privileged password	2000
3	passwd	Change password	2000
4	\$(NEWPASSWORD)	New password	2000
5	\$(NEWPASSWORD)	New password	2000

The **Verify Password Changed Commands** section defines the secret fields and commands to use to confirm that a password has rotated (changed) successfully on the target machine. The **Password Change Commands** section defines the secret fields and commands to use to change the password on the target machine.

10. Click the **Edit** button at the bottom of the page. The Edit Password Changer page appears:

Edit Password Changer

Name *

Line Ending New Line (\n) ▼

Custom Port (e.g. override the default value of 22 for SSH or 23 for Telnet with another value)

Runner Type Standard ▼

Request Terminal (If checked, the standard out and standard error data streams combine for \$\$CHECK* commands, else \$\$CHECK* will only check standard out and standard error will cause an error)

Exit Command (A custom command can be used to exit or logout of the session if only one connection per user is allowed on the device. Or if the SSH connections are not closing.)

Use SSH Password Authentication

Active

Valid for Discovery Import

Save
Cancel

- If a port for the RPC is listed in the **Custom Port** text box, it must match the port that Secret Server connects to when running the commands seen on the previous page.
- Click the **Cancel** button to return to the previous page:

Unix Root Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username `${1}$USERNAME`
 Password `${1}$PASSWORD`
 Key `< None >`
 Passphrase `< None >`

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	<code> su \$USERNAME</code>	Turn on privileged commands	2000
2	<code> \$CURRENTPASSWORD</code>	Privileged password	2000
3	<code> whoami</code>	Get name of account	2000
4	<code> \$\$CHECKFOR \$USERNAME</code>	Check the privileged login worked	2000

Password Change Commands Test Action

AUTHENTICATE AS

Username `${1}$USERNAME`
 Password `${1}$PASSWORD`
 Key `< None >`
 Passphrase `< None >`

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	<code> su \$USERNAME</code>	Turn on privileged commands	2000
2	<code> \$CURRENTPASSWORD</code>	Privileged password	2000
3	<code> passwd</code>	Change password	2000
4	<code> \$NEWPASSWORD</code>	New password	2000
5	<code> \$NEWPASSWORD</code>	New password	2000

- The **Verify Password Changed Commands Test Action** button tests the defined password-changed verification listed under it. When clicked, it uses the "Authenticate As" parameters to connect to the accounts and run the commands to test for a heartbeat and check that the account and password is valid.

Note: This authenticates with a non-privileged associated secret and then uses that account to connect to the Linux machine. This is needed because root accounts are often unable to directly authenticate. Thus, several commands are run to test if the active account can be set to root. If that fails, heartbeat fails.

- In the example command set for the section, when the heartbeat runs, the associated account (`${1}$USERNAME`) authenticates, logs into the remote SSH device, and runs:

- `su $USERNAME` (The username from the secret)
- `$CURRENTPASSWORD` (The password which the password should have been changed to)
- `whoami` (Returns the name of the active user, which indicates the su command and the provided parameters worked). This test checks that the returned username is the same as the username field in the secret. If it is not, the heartbeat fails.
- `CHECKFOR $USERNAME` (Checks if the 'whoami' returns the username field from the secret. If it does not, an error is thrown and the heartbeat fails)

Note: Some of the command sets run by the "Verify Passwords Changed Test Action" button are empty. In that case, the test authenticates with the provided username and password, and if that is successful, so is the heartbeat. That is, the heartbeat uses the secret's own account (`$USERNAME`) and value to connect, rather than those of an associated secret.

Note: If the RPC is set up to use an associated secret but the secret does not have one, the secret fails to rotate and throws

an error.

Note: For more on how Secret Server interprets what values to supply your custom script from the secrets involved, see [Editing Custom Commands](#) and [Remote Password Changing](#).

Note: The heartbeat above is designed to authenticate with a non-privileged associated secret and use that account to connect to the Unix machine because root accounts are often unable to authenticate directly. Then, several commands are run to check if root can be successfully switched to. If this fails, the heartbeat fails.

- When you click the Verify Password Changed Commands **Test Action** button, the commands cannot read the fields from a secret or associated secret because when setting up the password changer no specific secret is calling it. Instead, for the test only, you manually provide the input parameters from the secret and associated secrets involved with the RPC. For example, the \$USERNAME field refers to the user on the secret that you are trying to change. Whereas [1][1]USERNAME refers to the first associated secret linked to that secret.
- When you click the button a popup appears for you to do just that:

Test Action

Please enter the information to test the action. Note this test may take a while to complete since it will run against the actual host.

Authenticate As

Username

Password

Key No file chosen

Passphrase

Fields

MACHINE

Site

- Type or select your parameters.
- Click the **OK** button. The password-changed command set is tested with a simulated heartbeat, using what you entered. If any errors occur, record them for troubleshooting later. The console outputs something similar to this:


```

su root
Password:

[root@CentOS user1]#
whoami
root
[root@CentOS user1]#
$$CHECKFOR passed.
    
```

You then return to the previous page:

Unix Root Account Custom (SSH)

Verify Password Changed Commands ↗ Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME

Password \${1}\$PASSWORD

Key < None >

Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$(CURRENTPASSWORD)	Privileged password	2000
3	whoami	Get name of account	2000
4	\$(CHECKFOR \$USERNAME)	Check the privileged login worked	2000

Password Change Commands ↗ Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME

Password \${1}\$PASSWORD

Key < None >

Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$(CURRENTPASSWORD)	Privileged password	2000
3	passwd	Change password	2000
4	\$(NEWPASSWORD)	New password	2000
5	\$(NEWPASSWORD)	New password	2000

19. The **Password Change Commands** section defines the secret fields and commands to use to rotate (change) a password on the target machine. We now run a similar test on it.

20. Click the Password Change Commands **Test Action** button. Another Test Action popup page appears:

Test Action

Please enter the information to test the action. Note this test may take a while to complete since it will run against the actual host.

Warning: This will change the password and/or rotate the SSH Keys on the target account if successful.

Authenticate As

Username

Password

Key No file chosen

Passphrase

Fields

MACHINE

\$CURRENTPASSWORD

\$NEWPASSWORD

Site

Important: Clicking the OK button in the following instructions **really changes the password or rotates the SSH keys on the target account** (the \$NEWPASSWORD parameter gets changed), so record what you change it to, and update the secret with the new password (assuming the RPC is successful).

21. Similar to the last test, manually provide the input parameters. See [Step 2: Testing Heartbeat and RPC in Secret Server](#) for a description of how to fill in the parameters.
22. Click the **OK** button. The test connects with the "Authenticate As" accounts and runs the commands to change the password. A password rotation occurs, and more console output appears. Record any errors and output.
23. If the rotation did not occur, check the information that was presented to the changer from your secret. It is possible that the secret's data is involved in the issue.

Step 3: Troubleshooting Heartbeat or RPC Outside of Secret Server

This section troubleshoots the commands used by Secret Server to heartbeat and RPC outside of Secret Server. The intent is to confirm a successful authentication and password change on the endpoint when the same commands are issued outside of Secret Server. If they do not, the commands must be revised to work within Secret Server.

1. Use the procedure from [Step 1](#) to determine which machine (Secret Server application or DE) to perform this step on.
2. If you did not already, [Download PuTTY](#) on the application or any of the DE servers.
3. Open a browser tab on the secret which is failing to Heartbeat or RPC.
4. Do the same for each associated secret of that secret.

Note: Instead of the following three steps, you can instead go to ...\\SecretServer\\CustomCommandsEdit.aspx.

5. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears (not shown).
6. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:

Password Changers Configuration		
PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (Telnet)	SSH Local Account	Yes
Cisco Enable Secret Custom (SSH)	SSH Local Account	Yes
Cisco Enable Secret Custom (Telnet)	SSH Local Account	Yes

7. Click the name link for the subject password changer. The password changer configuration page for that changer appears:

Unix Root Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$CURRENTPASSWORD	Privileged password	2000
3	whoami	Get name of account	2000
4	\$\$CHECKFOR \$USERNAME	Check the privileged login worked	2000

Password Change Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$CURRENTPASSWORD	Privileged password	2000
3	passwd	Change password	2000
4	\$NEWPASSWORD	New password	2000
5	\$NEWPASSWORD	New password	2000

8. Determine if you are troubleshooting heartbeat or RPC: The **Verify Password Change Commands** section applies to heartbeat, and the **Password Change Commands** section applies to RPC. Which you use (or both) depends on what failed in Step

9. Return the Secret Server or DE server you are testing.

10. Launch PuTTY.

11. Type the host name or IP address of the subject Linux machine (generally, the Machine field in the secret).

12. Log on with the username and password for the main or associated secret.

Note: If you are successful with connecting with PuTTY but not Secret Server, launch PuTTY in in debug mode and collect a log file. Determine what cipher was used to connect. If you have a machine that works with Secret Server, compare the ciphers. Also check if the endpoint handles interactive logins differently. Secret Server's logins for RPC are non-interactive. See [Troubleshooting SSH Issues](#) for more information about troubleshooting connection issues in Putty.

13. Use the commands listed in the "Authenticate As" section you are troubleshooting directly in PuTTY to determine if they work outside of Secret Server. For example, given these heartbeat (Verify Password Changed) commands:

Unix Root Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$CURRENTPASSWORD	Privileged password	2000
3	whoami	Get name of account	2000
4	\$\$CHECKFOR \$USERNAME	Check the privileged login worked	2000

Password Change Commands Test Action

AUTHENTICATE AS

Username \${1}\$USERNAME
 Password \${1}\$PASSWORD
 Key < None >
 Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su \$USERNAME	Turn on privileged commands	2000
2	\$CURRENTPASSWORD	Privileged password	2000
3	passwd	Change password	2000
4	\$NEWPASSWORD	New password	2000
5	\$NEWPASSWORD	New password	2000

Your console would look like this:

```

testuser@centos6:/home/testuser
login as: testuser
testuser@192.168.1.140's password:
[testuser@centos6 ~]$ su root
Password:
[root@centos6 testuser]# whoami
root
[root@centos6 testuser]#
    
```

In this example, we assumed the secret contained a value of "root" for the Username field, and the associated account in the first position was "testuser." This example was successful because the \$\$CHECKFOR \$USERNAME found "root" on the previous line.

If the `su root` command were to fail above and reports the message "Username is not in the sudo users file. This incident will be reported." then the `$$CHECKFOR` would fail and the heartbeat would fail to verify. This type of issue needs to be remediated on the endpoint.

14. If the issue is clearly an endpoint issue, remediate it and repeat the commands in PuTTY.
15. Once the commands work properly in PuTTY, if the RPC or heartbeat command set needs adjustment to match the working PuTTY equivalent, return to Secret Server and make the changes to the command set (see the next step).

Important: Before you change the RPC commands, ensure that the device that you are working on belongs to the secret template you are using. Secret templates dynamically update all the secrets based on them, so **all secrets with this template are affected by your changes**. We strongly recommend that if this device is unique or you are storing an independent root account in the associated secret template, you should:

1. Copy the template you are using, giving the copy a descriptive name.
2. Create a new password changer based on the current one that you are using.
3. Assign it to the secret template you just created.

This ensures that you do not change how ALL devices related to a secret template when you only intend to change a single device. Accounts that are the same type on the same device should share the same template.


16. Click the **Edit Commands** button at the bottom of the subject password changer page. The commands for RPC and heartbeat appear:
-

Unix Account Custom (SSH)

Verify Password Changed Commands

AUTHENTICATE AS

Username	<input type="text" value="\$Username"/>
Password	<input type="text" value="\$CURRENTPASSWORD"/>
Key	<input type="text"/>
Passphrase	<input type="text"/>


 Save

ORDER	COMMAND	COMMENT	PAUSE(MS)
	<input type="text"/>	<input type="text"/>	2000 

Password Change Commands

AUTHENTICATE AS

Username	<input type="text" value="\$Username"/>
Password	<input type="text" value="\$CURRENTPASSWORD"/>
Key	<input type="text"/>
Passphrase	<input type="text"/>

 Save

17. Scroll down to the command list in the **Password Change Commands** section:

Password Change Commands

AUTHENTICATE AS

Username

Password

Key

Passphrase

ORDER	COMMAND	COMMENT	PAUSE(MS)	
1	passwd	Password Command	2000	
2	\$CURRENTPASSWORD	Current Password	2000	
3	\$NEWPASSWORD	New Password	2000	
4	\$NEWPASSWORD	Confirmed Password	2000	
	<input type="text"/>	<input type="text"/>	<input type="text" value="2000"/>	

18. Click the blue edit icon to the right of any commands you want to change. The command becomes editable.
19. Edit the command to make it match your known-good revision.
20. Click the blue save icon next to the amended command.
21. Click the **Back** button to return to the password changer page:

Unix Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username \$Username

Password \$CURRENTPASSWORD

Key < None >

Passphrase < None >

Password Change Commands Test Action

AUTHENTICATE AS

Username \$Username

Password \$CURRENTPASSWORD

Key < None >

Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	passwd	Password Command	2000
2	\$CURRENTPASSWORD	Current Password	2000
3	\$NEWPASSWORD	New Password	2000
4	\$NEWPASSWORD	Confirmed Password	2000

22. One of the Test Action popups appears.

23. Type the known-good values from the secret in the text boxes.

24. Click the **OK** button to test heartbeat or RPC. The result should look something like this:

```

su root
Password:
[root@CentOS user1]#
whoami
root
[root@CentOS user1]#
$$CHECKFOR passed.
    
```

This topic discusses resolving the "The specified domain is not a valid domain" error.

Troubleshooting Procedure

1. Verify that you are entering the fully qualified domain name in the domain field and that the domain username and password fields are correct.
2. Ensure that the ports used for LDAP (389) or LDAPS (389 and 636) are open. For more information about the ports used by Secret Server, see [Ports Used by Secret Server](#).
3. Ensure that your server is connecting to the correct DNS server:
 1. Open the command console as an administrator (**Start > Run > cmd**).
 2. Type `ipconfig /all`.
 3. Press **<Enter>**.
 4. Find your primary ethernet adapter and look in the **DNS Servers** section. Verify that the DNS server is correct.
4. If the DNS server is incorrect, then follow these steps to configure the DNS server:
 1. Open up your control panel (**Start > Control Panel**).
 2. Click on **Network and Sharing Center**.
 3. Click **Manage Network Connections** on the left.
 4. Right click on your primary network adapter and select **Properties**.
 5. Click **Internet Protocol Version 4 (TCP/IPv4)**.
 6. Click **Properties**.
 7. Click to select the **Use the following DNS server addresses selection** button.
 8. Type your primary DNS server in the first row.
 9. If you have a secondary DNS server, put it in the second row.

Important: Both DNS servers must contain the SRV record for your domain controller.

5. Check that your server is retrieving domain controller DC records correctly:
 1. Open up your control panel (**Start > Control Panel**).
 2. Type `nslookup`.
 3. Press **<Enter>**.
 4. Type `set q=srv`
 5. Press **<Enter>**.
 6. Type `_ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name>`.
 7. Press **<Enter>**.

8. If you get a result that looks like:

```
_ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name> SRV service location: priority = 0 weight = 100 port = 389 svr hostname = *Domain_Controller_Host_Name*
```

Then you are retrieving the DNS record correctly. Otherwise, your DNS records are not correctly configured.

Configuring the DNS Record on Your Server

1. If you are **not** using a Windows DNS server, contact your vendor to ask how to add SRV records. You will need to add a SRV record pointing `_ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name>` to your primary DNS server.
2. Connect to your Windows DNS server.
3. Open the DNS control panel (**Start > Administrative Tools > DNS**).
4. Expand the node corresponding to your server.
5. Expand the **Forward Lookup Zones** node.
6. Expand the node corresponding to your domain.
7. Delete the **_msdcs** node if it exists.
8. Right click on the domain node and select **New Domain...**
9. Type `_msdcs` as the name.
10. Right click on the new **_msdcs** node, and select **New Domain...**
11. Type `dc` as the name.
12. Right click on the new **dc** node and select **Other New Records...**
13. Select **Service Location (SRV)** as the record type.
14. Click the **Create Record** button.
15. Select **_ldap** as the service.
16. Select **_tcp** as the protocol.
17. Type `389` as the port.
18. Type the fully qualified host name of your DC or the IP address in the **Host offering this service:** text box.
19. Click the **OK** button.
20. Click the **Done** button.
21. Open up the services console (**Start > Run > services.msc**)
22. Right click on the **DNS Server** service and select **Restart**. Your domain DNS record should now be set up.

Resolving Other DNS Issues

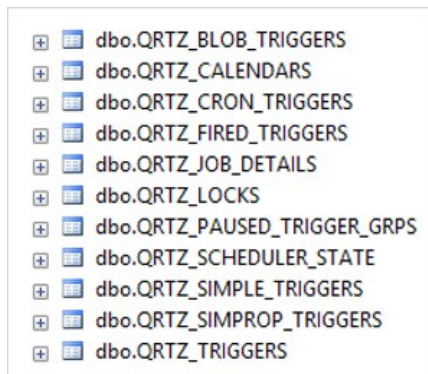
Secret Server requires that the DNS is correctly configured to add a domain. For additional tips on tracking down DNS Issues, see this [Troubleshooting Active Directory Installation Wizard Problems](#).

Also ensure the domain controller is using the appropriate DNS. The `ipconfig /registerdns` command (as per the link above) is frequently helpful for entering the correct DNS entries in for a given domain.

In Secret Server 10.7 and newer, all background operations have moved to Quartz. All scheduled background operations have become jobs registered in tables inside the Secret Server database.

Quartz-Related Tables

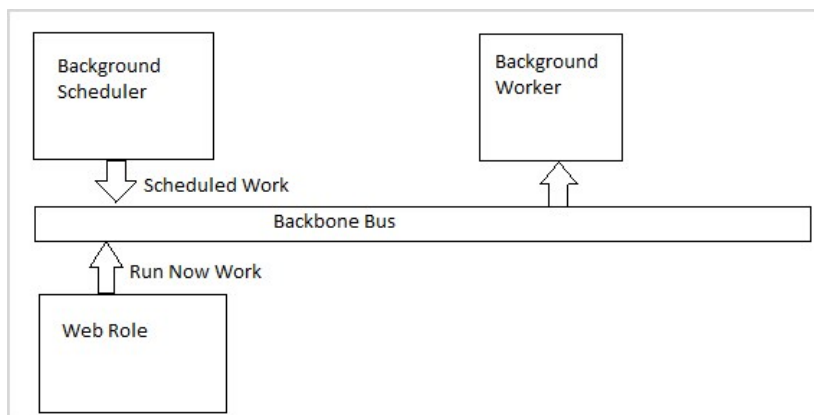
All Quartz-related table names start with "QRTZ_"



+	dbo.QRTZ_BLOB_TRIGGERS
+	dbo.QRTZ_CALENDARS
+	dbo.QRTZ_CRON_TRIGGERS
+	dbo.QRTZ_FIRED_TRIGGERS
+	dbo.QRTZ_JOB_DETAILS
+	dbo.QRTZ_LOCKS
+	dbo.QRTZ_PAUSED_TRIGGER_GRP
+	dbo.QRTZ_SCHEDULER_STATE
+	dbo.QRTZ_SIMPLE_TRIGGERS
+	dbo.QRTZ_SIMPROP_TRIGGERS
+	dbo.QRTZ_TRIGGERS

Trigger Job Components

Each background job runs as a consumer, similar to how a distributed engine (DE) works, in the background worker. Each job has a corresponding trigger job registered in Quartz. Quartz executes the trigger job at the configured time and interval. The trigger job puts a message on the backbone bus, which is picked up and executed by the background worker:



Simple Trigger Job

There are two kinds of registered trigger jobs, *simple triggers* and *cron triggers*. A simple trigger fires on an interval. For example, on premises the heartbeat job runs every 30 seconds:

```
1 | select * from QRTZ_SIMPLE_TRIGGERS where trigger_name like '%heartbeat%'
```

	SCHED_NAME	TRIGGER_NAME	TRIGGER_GROUP	REPEAT_COUNT	REPEAT_INTERVAL	TIMES_TRIGGERED
1	BackgroundScheduler	ProcessHeartbeat TriggerJob-trigger	group1	-1	30000	3
2	BackgroundScheduler	ProcessLocalHeartbeat TriggerJob-trigger	group1	-1	30000	3

In this case, you can ignore the TRIGGER_GROUP and REPEAT_COUNT columns. The latter column limits the number repetitions, which we do not do. The REPEAT_INTERVAL is in milliseconds, so 30000 is 30 seconds. TIMES_TRIGGERED is a counter of trigger firings. There are two trigger jobs for heartbeat. The local job handles secrets that do not run through a distributed engine while the other heartbeat handles secrets that do run through a distributed engine.

Cron Trigger Job

Cron jobs run at a specific time using a variation of the well-known cron syntax. For example:

```
1 | select * from QRTZ_CRON_TRIGGERS where trigger_name like '%license%'
```

	SCHED_NAME	TRIGGER_NAME	TRIGGER_GROUP	CRON_EXPRESSION	TIME_ZONE_ID
	BackgroundScheduler	ExpiringLicenseTask TriggerJob-trigger	group1	0 0 12 * * ?	UTC

In this example, the trigger ExpiringLicenseTaskTriggerJob runs every day at 12:00 PM UTC.

The Quartz jobs are run within the background scheduler role, which runs on every node that has the background-worker-enabled bit set to true. Quartz coordinates trigger-job execution through the database to ensure that the same trigger-job is run just once, even if the Quartz scheduler is running on multiple nodes simultaneously.

Procedures

Viewing Trigger Job State (QRTZ_TRIGGERS Table)

The current state of each trigger-job can be seen in the QRTZ_TRIGGERS table:

```
1 | select Trigger name,NEXT FIRE TIME,prev fire time,trigger state from QRTZ TRIGGERS
```

	Trigger name	NEXT_FIRE_TIME	prev_fire_time	trigger_state
1	BackgroundWorkerTask TriggerJob-trigger	637019930838743141	637019930738743141	WAITING
2	Backup TriggerJob-trigger	637019930838863056	637019930238863056	WAITING
3	CheckinExpiredCheckedoutSecret TriggerJob-trigger	637019930838143479	637019930238143479	WAITING
4	ComputerScan TriggerJob-trigger	637019932036884199	637019928436884199	WAITING

Viewing Trigger Firing Times

In the previous figure, the NEXT_FIRE_TIME and PREV_FIRE_TIME columns are not human readable. To convert them to something you can read, run the following SQL query:

```
SELECT TOP 1000
  [TRIGGER_NAME]
  ,[TRIGGER_STATE]
```

```

,[TRIGGER_TYPE]
,NEXT_FIRE_IN = DATEDIFF(second, GETUTCDATE(), CAST(NEXT_FIRE_TIME/864000000000.0 - 693595.0 AS DATETIME))
,UTCNOW = GETUTCDATE()
,NEXT_FIRE_TIME2 = CAST(NEXT_FIRE_TIME/864000000000.0 - 693595.0 AS DATETIME)
,[PREV_FIRE_TIME2] = CAST(PREV_FIRE_TIME/864000000000.0 - 693595.0 AS DATETIME)
,[START_TIME2] = CAST(START_TIME/864000000000.0 - 693595.0 AS DATETIME)
,[END_TIME2] = CAST(END_TIME/864000000000.0 - 693595.0 AS DATETIME)
,[PRIORITY]
FROM [dbo].[QRTZ_TRIGGERS]

```

This query shows the NEXT_FIRE_TIME in seconds and the other fields as DateTimes, along with the current UTC date to make it easier to compare to the current time.

The TRIGGER_STATE is normally WAITING. If the trigger job is currently running, the state will be ACQUIRED. If there was an error running the trigger job, the state will be ERROR. Once a trigger job enters an ERROR state it will not run again. To address this, Secret Server automatically looks for any trigger jobs in the ERROR state every 10 minutes and changes their state to WAITING. If the NEXT_FIRE_TIME is a large negative number, it indicates that the scheduler role is not running—take a look at the Secret Server-BSSR.log file to see its status.

Adjusting Trigger Job Frequency

Secret Server populates the Quartz schedules when it starts up and creates them if they do not exist. Secret Server does not update pre-existing schedules, so it is possible to change the frequency of a job in the table and have the system run with that frequency from that point on.

Important: Do not change the job frequency unless it is absolutely necessary.

Delinea controls those schedules and in future releases may not recognize changes you have made to those schedules. Furthermore, we do not test with a variety of schedules and have no plans to do so. Therefore, adjusting the schedules is a risky undertaking that may cause issues within Secret Server. If you must change the tables, you must bounce IIS on all the nodes where the scheduler runs, after you change the table values.

To trigger an infrequent task just once, without altering its schedule, follow these steps:

1. Update the NEXT_RUN_TIME value in the QRTZ_TRIGGERS table to match the NEXT_RUN_TIME value from a trigger that fires frequently, such as one of the heartbeat triggers.
2. Recycle the scheduler role by running Recycle Background Processes on the Diagnostics page.
3. Reset the background worker (inside the application pool) in one of the following ways:
 - Click Recycle Background Processes on the Diagnostics page. This shuts down and restarts the scheduler.
 - Recycle the the application pool. This restarts Secret Server in its entirety.
 - Reset IIS. From SS's perspective this is nearly the same as the second option.

Note: The PDF version of this online document is automatically generated and thus may have minor formatting anomalies.

Note: This document is not updated with every release—many releases do not affect the guide's contents and thus do not warrant a document update.

When troubleshooting performance or connectivity issues with SSH with or without proxy it is useful to enable SSH debug logging on your remote host. There are several things that could go wrong during the connection process and the SSH debug log tells you how far the connection gets before failing. To enable debug logging on a host the SSH service should be started with the debug flag.

Important: A UNIX administrator should be tasked with these operations because if the box is a remote host with no local access then an incorrect action could leave you locked out of the machine if SSH is the only remote connection possible.

Local Servers with Direct Access

The following example works best with a local connection. You can start the SSH service in verbose debug mode where the debug output is sent to a file on the local system and the service terminates after the remote connection ends with the following:

```
/usr/sbin/sshd -ddd 2> sshd-debug.log
```

Remote Servers

Another way to configure SSH to log debugging information is to have syslog set up. You will need to add a syslog entry for the SSH service in `/etc/syslog.conf`:

```
*.* /var/log/sshd/sshd.log
```

And then configure the SSH service to have a log level of `DEBUG3`, which can be modified in `/etc/ssh/sshd_conf`:

```
LogLevel DEBUG3
```

Then restart the SSH service:

```
service sshd restart
```

Note: On some systems, the log may already be configured to output to `/var/log/auth.log`.

Logging from the Client Perspective

You can also do logging from the perspective of the client connection to the remote host. This sort of logging helps you to understand what a normal successful connection should look like. To obtain logging from the client connection, you can run SSH in verbose mode.

```
ssh -vvv user@host
```

The debug information will be sent to the console. Or if you are using PuTTY, then you can right click the PuTTY window title after connecting to the remote host and selecting "Event Log".

Understanding SSH Logging

Example

The following is an example of standard debug output from PuTTY looks like the following:

1. The client begins by lookup up the hostname and see if the host is valid:

```
2016-01-07 12:23:57 Looking up host "192.168.1.60"
```

2. The client proceeds to make a TCP connection to the host:

```
2016-01-07 12:23:57 Connecting to 192.168.1.60 port 22
```

3. The client sends a message to the server saying what version of SSH we are using. In this example we are using SSH 2 over PuTTY v0.65:

2016-01-07 12:23:57 We claim version: SSH-2.0-PuTTY_Release_0.65

- The server sends back a message saying what version of SSH they are using. In this example we are connecting to an Ubuntu Server running an SSH 2 over OpenSSH v6.6.1p1:

2016-01-07 12:23:57 Server version: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2

2016-01-07 12:23:57 We believe remote version has SSH-2 channel request bug

- The client confirms the type of connection that can be used with the server. In this example the communications will be done using SSH 2:

2016-01-07 12:23:57 Using SSH protocol version 2

- The client begins the key exchange process and agrees on using Diffie-Hellman Group Exchange 256:

2016-01-07 12:23:57 Doing Diffie-Hellman group exchange

2016-01-07 12:23:57 Doing Diffie-Hellman key exchange with hash SHA-256

- The server replies with their host key fingerprint information to identify its identity and aide in the prevention of Man in the Middle attacks. In this example the server is presenting an RSA 2048-bit key:

2016-01-07 12:23:57 Host key fingerprint is:

2016-01-07 12:26:53 ssh-rsa 2048 e0:d4:94:36:e9:20:fd:e3:58:ad:8d:4c:4a:1f:27:e8

- The client initializes the transport layer encryption using AES-256 with SDCTR enabled and uses SHA-256 for message authentication:

2016-01-07 12:26:53 Initialized AES-256 SDCTR client->server encryption

2016-01-07 12:26:53 Initialized HMAC-SHA-256 client->server MAC algorithm

- The server initializes the transport layer encryption using AES-256 with SDCTR enabled and uses SHA-256 for message authentication:

2016-01-07 12:26:53 Initialized AES-256 SDCTR server->client encryption

2016-01-07 12:26:53 Initialized HMAC-SHA-256 server->client MAC algorithm

- The client sends a password.

2016-01-07 12:27:12 Sent password

- The server validates the password and granted access to the user:

2016-01-07 12:27:12 Access granted

- The session opens a shell for user interaction:

2016-01-07 12:27:12 Opening session as main channel

2016-01-07 12:27:12 Opened main channel

2016-01-07 12:27:12 Allocated pty (ospeed 38400bps, ispeed 38400bps)

2016-01-07 12:27:12 Started a shell/command

Confirming Proper Operation

Things to look for in an SSH log:

- Verifies the host IP address that you are connecting to.
- Verifies the port is correct for the address that you are connecting to.
- Verifies that you are not using an outdated SSH client.

- Verifies the SSH protocol you are using.
- Verifies what group exchange algorithm is being used.
- Verifies the server identity using the presented fingerprint. If the fingerprint is not expected then there may be malicious server between you and the remote host you want to connect to. Alert the administrator to verify if the host key has changed or if there is another issue.
- Verifies the transport layer and HMAC ciphers being used.
- Verifies that the password or key being sent is accepted by the server.
- If a connection does not open, it notes what the last successful step was and then what the next failed step is to find what the issue is.
- If using Secret Server proxy, it is useful to collect the client-to-proxy SSH log and then the proxy-to-remote-host log from the remote server.

Secret Server supports operating systems in a VMware virtual machine environment in an identical manner as it runs on any other major x86-based systems without initially requiring reproduction of issues on native hardware. Should Delinea Support suspect that the virtualization layer is the root cause of an incident, you must contact VMware support provider to resolve the VMware issue. While Delinea products are expected to function properly in a VMware virtual environment, there may be performance implications that can invalidate Secret Server sizing and recommendations.

Note: Migrating between CPU families or versions will require re-activation.

Overview

Beginning with Windows 10 version 1607 (Creator's Update) and Windows Server 2016, the default GPO security descriptor denies users [remote access to Security Account Manager \(SAM\)](#) with non-domain credentials, and therefore prevents remote heartbeat and password changes made by otherwise-authenticated local user accounts. Affected Windows local account secrets return "Access Denied" on a heartbeat or remote password change.

This article provides a script and instructions to address these "access denied" errors. The script modifies the default local group policy remote SAM access security descriptor to allow all local users on a specified machine remote SAM access after authentication. This script requires elevated PowerShell permissions.

Note: Adding an account to the local computer's Administrators group does not solve the problem.

On most systems, the Administrators group on the local machine is part of the "Network Access: Restrict clients allowed to make remote calls to SAM" security policy setting. Through testing, we determined that Windows currently treats this group as only the built-in administrator account for this configuration. Therefore, if you add another user to the Administrators group on the machine, that user will be unable to heartbeat since it is not the built-in administrator account. In addition, the built-in object, "Local account and a member of Administrators" does not allow a local account that is a member of Administrators to heartbeat for any account other than the built-in administrator account.

Additional Requirements

For heartbeat to work correctly, make sure that the local or authenticated users are:

- *Not* in the "Deny access to this computer from the network" security policy
- *In* the "Access this computer from the network" security policy

Remediation Options

Option 1: Creating a custom group and adding users to it (this is what the script does for users on the endpoint) then adding that group to the security setting to allow the user to heartbeat successfully. New local users need to be added to the custom group if they are created in the future.

Option 2: Adding a user individually to the security setting to allow the user to heartbeat successfully.

Option 3: Modifying the Default GPO: Adding "allow authenticated or local users" to the security setting. This allows all local users or all users who are authenticated to the machine to bypass this setting. This does require the PowerShell Script below. The drawback is that this allows all users to remotely access SAM, so long as they are authenticated.

Option 4: Create a heartbeat workaround for GPO "Network Access: Restrict Clients Allowed to Make Remote Calls to SAM." This is addressed in the last section. This is for situations where the GPO needs to be completely bypassed.

Option 3: Modifying the Default GPO

PowerShell Script Description

This script adds a local non-privileged user group to the machine (a custom group name can be specified with the `-GroupName` parameter), adds all local users to the group, and then adds this group to the "Network Access: Restrict clients allowed to make remote calls to SAM" local group policy. This allows all local users within the group remote access to SAM after authentication, which is required for Secret Server heartbeat and password changing.

Download

Extract the .ps1 script found here: https://updates.thycotic.net/secretserver/support/PowerShell_Win10-HB-RPC-Fix/Win10-HbFix.zip Run in an elevated PowerShell ISE session.

Script Argument Help

Command Prompt Help

For full help text, run:

```
> Get-Help C:\Script\Win10-HbFix.ps1 -Examples
```

Parameters

-ComputerNames (string)

Specifies the computers on which the script runs (comma separated). If unspecified, the default is the local computer.

-Username (string)

Specifies a username of an account that has administrative permissions on the computer to add a local user group and modify the local group policy. You will be prompted for a password. Examples: Administrator Or TestDomain\AdminUser.

-GroupName (string)

Specifies a name for the SAM access local user group. If unspecified, the default group name is "Secret Server Remote SAM Access"

-ForceGPUdate

Specifies whether a group policy update should be forced for immediate effect following the script. (Otherwise Group Policy changes may take up to 120 minutes to take effect by default).

Examples

```
> C:\Script\Win10-HbFix.ps1 This example gives remote SAM access to all local users on the current machine. The current PowerShell credentials would be used for authentication.
```

```
> C:\Script\Win10-HbFix.ps1 -LogDir "D:\Win10-HbFix\log" This example changes the default output log path to D:\Win10-HbFix\log (default is [user temp directory]\log).
```

```
> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER","TestDomain\SOMEMACHINE" -Username "TestDomain\Administrator" This example gives remote SAM access to all local users on the WINSERVER and TestDomain\SOMEMACHINE remote computers. The domain user "TestDomain\Administrator" credentials will be used. You would be prompted for a password.
```

```
> C:\Script\Win10-HbFix.ps1 -ComputerNames "D:\Win10MachineList.txt" -Username "TestDomain\Administrator"
```

This example gives remote SAM access to all local users on the remote computers listed in D:\Win10MachineList.txt (one machine per line). The domain user "TestDomain\Administrator" credentials will be used. You would be prompted for a password.

```
> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER" -GroupName "Secret Server Group"
```

This example gives remote SAM access to all local users on the WINSERVER remote computer. The local group created will be named "Secret Server Group". Current PowerShell credentials would be used for authentication.

```
> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER" -ForceGPUdate -Verbose This example gives remote SAM access to all local users on the WINSERVER remote computer, with verbose output. The current PowerShell credentials will be used for authentication. Group policy update
```

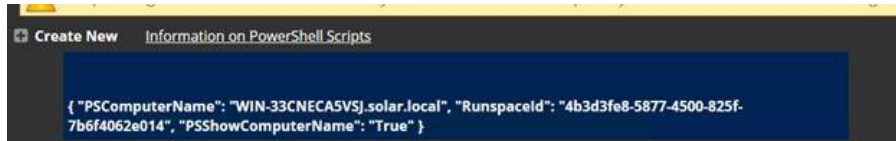
will be forced on WINSERVER for immediate effect.

Related Articles and Resources

[Network access: Restrict clients allowed to make remote calls to SAM](#)

Option 4: Creating a Heartbeat GPO Workaround

1. Make sure that **Admin > Scripts** is functional. Once you have it working, download, unzip, and run this script [HBWorkAroundScripts.zip](#).
2. Go to **Admin > Scripts**.
3. Add the HBWorkAroundScript and the HBWorkAroundPasswordChange scripts.
4. Test the first script. Add the appropriate `args[]` as needed. Add arguments 0-4 with no quotes or commas. Spaces are the argument separator and are required.
5. You should get a return of "True," such as this:



```
{ "PSComputerName": "WIN-33CNECA5VSJ.solar.local", "RunspaceId": "4b3d3fe8-5877-4500-825f-7b6f4062e014", "PSShowComputerName": "True" }
```

6. Navigate to **Admin > Remote Password Changing**.
 7. Click the **Configure Password Changers** button. The Password Changers Configuration page appears.
 8. Click the **New** button at the bottom.
 9. Click the **Base Password Changer** dropdown list to select **PowerShell Script** as your password changer.
 10. Type a name in the **Name** text box.
 11. Click the **Save** button. The password change command page appears:
-

Test PC

Verify Password Changed Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

PowerShell Script: <select> *

Script Args:

Password Change Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

PowerShell Script: <select> *

Script Args:

[Hide Advanced Settings](#)

SETTING	VALUE
Heartbeat Unknown Error to Unable to Connect Translation (regex)	<input type="text"/>
Attempt Password Change with new password when error contains (regex)	<input type="text"/>

12. Click the **PowerShell Script** dropdown list in the **Password Change Commands** section to select the script you ran earlier.

13. Add the appropriate tokens in the **Script Args** text box.

Note: See [Dependency Tokens](#) for a complete list.

12. Click the **Save** button. Your configuration should look like this:

HBWorkAroundChangerSAM

Verify Password Changed Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

PowerShell Script: HBWorkAroundScript

Script Args: \${1}\$Username \${1}\$Password \$Machine \$Username \$Password

Password Change Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

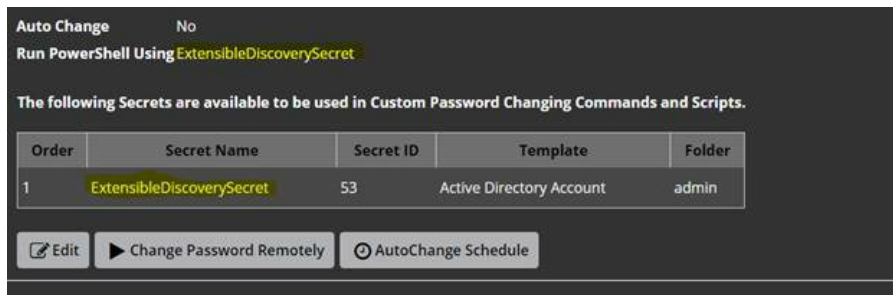
PowerShell Script: HBWorkAroundPasswordChange

Script Args: \${1}\$Username \${1}\$Password \$Machine \$Username \$NewPassword

13. Go to **Admin > Secret Templates**.

14. Select **Windows Account**.

15. Click the **Edit** button.
16. Click the **Copy Secret Template** button.
17. Click the **Configure Password Changing** button.
18. Click the **Edit** button.
19. Click the **Password Type to Use** dropdown list to select the password change you created earlier.
20. Create your windows secret using the custom template.
21. Once it is created, add your privileged and associated secret to the RPC tab as seen below. In that example we use the same one for the privileged and associated secret.



22. Run a heartbeat to confirm it works as desired.

User Groups

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server allows administrators to manage users through *user groups*. Users can belong to different groups and receive the sharing permissions, as well as roles, attributed to those groups. This setup simplifies the management of the permissions and roles that can be assigned to a user. Additionally, groups can be synchronized with Active Directory to further simplify management.

Group Administrators can also set another group or user as the group owners for a Secret Server local group. Group owners can manage membership just for that group. To assign the group owner:

1. Navigate to the **Groups** page:

The screenshot shows the 'Admin > Groups' page. At the top, there is a search bar, a grid icon, a plus sign, and a 'WS' indicator. Below the navigation, there are tabs for 'Groups' and 'Audit'. A toolbar contains three buttons: 'Manage Active Directory Groups', 'View Group Assignment', and 'Create Group'. The main content area shows 'Groups: 65' and a dropdown for 'All Domains'. A table lists various groups with columns for 'GROUP NAME', 'ENABLED', and 'CREATED'. The 'Include Disabled' checkbox is checked. A table with 10 rows is visible, showing group names and their creation dates.

GROUP NAME	ENABLED	CREATED
Access Control Assistance Operat...	Yes	8/9/2019 02:08 pm
Account Operators	Yes	8/9/2019 02:08 pm
Administrators	Yes	8/9/2019 02:08 pm
Allowed RODC Password Replicati...	Yes	8/9/2019 02:08 pm
Backup Operators	Yes	8/9/2019 02:08 pm
Cert Publishers	Yes	8/9/2019 02:08 pm
Certificate Service DCOM Access	Yes	8/9/2019 02:08 pm
Cloneable Domain Controllers	Yes	8/9/2019 02:08 pm
Cryptographic Operators	Yes	8/9/2019 02:08 pm

2. Click the desired group in the list. The Group's page appears:

Group

Group Name	Test Group
Enabled	Yes
Created	8/12/2019
IP Address Restrictions	None

Group Owners

NAME
gamma.thycotic.com\Developers
admin
gamma.thycotic.com\...
gamma.thycotic.com\...
gamma.thycotic.com\...
...
...
...
...
...

Members

There are no members.

[Back](#) [Edit](#) [Change IP Restrictions](#) [Configure Secret Template Permissions](#)
[View Audit](#) [View Group Assignment Audit](#) [View IP Address Audit](#)

3. Click the **Edit** button. The Group Edit page appears:

Group

Group Name gamma.thycotic.com\Administrators
Enabled Yes
Created 8/9/2019
IP Address Restrictions
None

Group Owners
Managed by Active Directory

Members

| Show All < 1 to 15 of 17 >

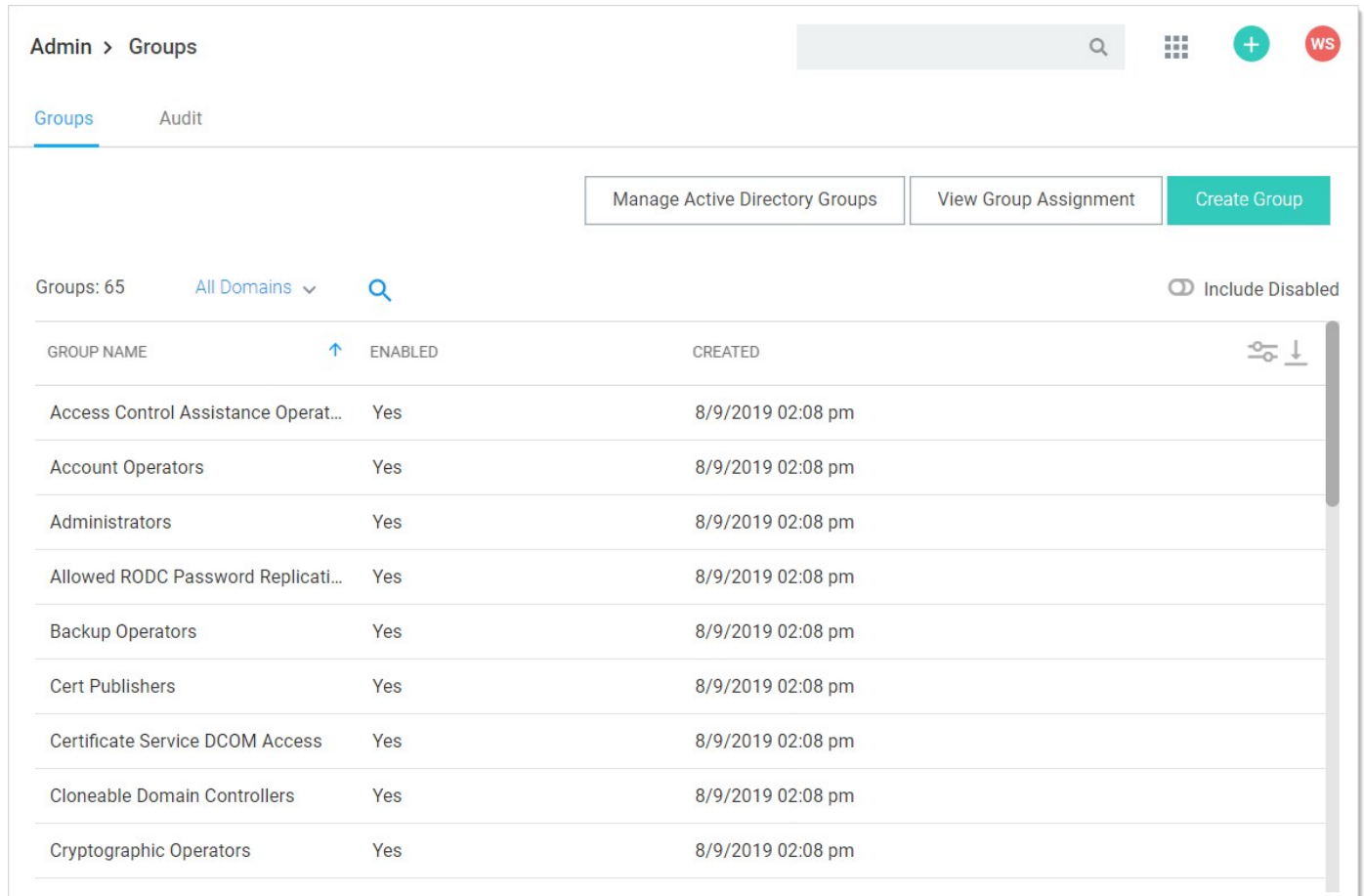
NAME

gamma.thycotic.com\ [blurred]
gamma.thycotic.com\ [blurred]
gamma.thycotic.com\ [blurred]
gamma.thycotic.com\ [blurred]
gamma.thycotic.com\ [blurred]

- Back
- Change IP Restrictions
- Configure Secret Template Permissions
- View Audit
- View Group Assignment Audit
- View IP Address Audit

On the Group Assignment page, users can be added and removed from the group.

1. Navigate to the **Groups** page:



The screenshot shows the 'Admin > Groups' page in the Delinea interface. At the top, there is a search bar, a grid icon, a plus sign, and a 'WS' indicator. Below the navigation bar, there are two tabs: 'Groups' (selected) and 'Audit'. A toolbar contains three buttons: 'Manage Active Directory Groups', 'View Group Assignment' (highlighted), and 'Create Group'. Below the toolbar, the page displays 'Groups: 65', a dropdown menu for 'All Domains', a search icon, and an 'Include Disabled' toggle. The main content is a table with the following data:

GROUP NAME	ENABLED	CREATED
Access Control Assistance Operat...	Yes	8/9/2019 02:08 pm
Account Operators	Yes	8/9/2019 02:08 pm
Administrators	Yes	8/9/2019 02:08 pm
Allowed RODC Password Replicati...	Yes	8/9/2019 02:08 pm
Backup Operators	Yes	8/9/2019 02:08 pm
Cert Publishers	Yes	8/9/2019 02:08 pm
Certificate Service DCOM Access	Yes	8/9/2019 02:08 pm
Cloneable Domain Controllers	Yes	8/9/2019 02:08 pm
Cryptographic Operators	Yes	8/9/2019 02:08 pm

2. Click the **View Group Assignment** button. The Group Assignment page appears:

Group Assignment

[By Group](#) [By User](#)

Group

Assigned

[Empty list box]

Unassigned

[List of users]

«

<

>

»

[← Back](#)

3. Use the arrow buttons to move users into and out of the current group. When you have finished with your changes, click the **Save Changes** button and your new group members are added.

Alternatively, you can click the By User tab and manage the groups for a single user:

Group Assignment

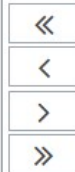
By Group By User

User

Assigned

Unassigned

Duo Approvers
Test Group
TJWForWorkflow



 Back

Note: If the group was created using Active Directory synchronization, this group is not be editable. See [Active Directory Synchronization](#).

You can create and edit groups from the Groups page. You can get to the Groups page by navigating to **Admin > Groups**

Admin > Groups

Groups Audit

Manage Active Directory Groups View Group Assignment Create Group

Groups: 65 All Domains Include Disabled

GROUP NAME	ENABLED	CREATED
Access Control Assistance Operat...	Yes	8/9/2019 02:08 pm
Account Operators	Yes	8/9/2019 02:08 pm
Administrators	Yes	8/9/2019 02:08 pm
Allowed RODC Password Replicati...	Yes	8/9/2019 02:08 pm
Backup Operators	Yes	8/9/2019 02:08 pm
Cert Publishers	Yes	8/9/2019 02:08 pm
Certificate Service DCOM Access	Yes	8/9/2019 02:08 pm
Cloneable Domain Controllers	Yes	8/9/2019 02:08 pm
Cryptographic Operators	Yes	8/9/2019 02:08 pm

By either selecting an already existing group from the list, or clicking **Create Group**, you can modify or add the group.

Note: To add groups and the users inside them from your Active Directory setup, you can use Active Directory synchronization (see [Active Directory Synchronization](#)).

User Teams

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

With Secret Server teams, administrators can create special groups called *teams* to restrict what users can see. A team bundles users and groups to assign them the same rules as to what other users and sites are visible to them. For example, a managed service provider could isolate their customers from seeing other customer's user accounts or a large company could "firewall" their users by department. Site visibility can also be restricted by teams.

Note: Teams are designed for shared secrets and do not apply to Secret Server administration as a whole.

Note: Users *without* any team-related permissions are subject to team restrictions. The Unrestricted by Teams permission must be present to remove them. That is why the User role comes with that permission by default. See [Team-Related Permissions](#).

Note: Team restrictions are designed for regular users so granting additional administrative permissions can override the restriction. This applies to group owners, so if a user is assigned as a group owner, that user will be able to see all users when assigning members.

Team visibility and management are controlled by user roles. Those roles, and by extension users, are governed by the following team-related role permissions:

- **Administer Teams:** Users can create, edit, and view all teams.
- **No Teams-related Permissions:** Users can only view other users within their team.
- **Unrestricted by Teams:** Users can view all users, groups, and sites, regardless of Team affiliation. Essentially, teams do not exist for the users with this permission, and the Teams page is not available to them. The default user role has this permission.
- **View Teams:** Users can view all teams. This is essentially a read-only Administer Teams.

To set up Secret Server to use the team management feature:

1. Create a new role called *Team Limited User*.
2. Assign all permissions of the standard user role except *Unrestricted by Teams*.
3. Assign users you want restricted by teams to this role.
4. Remove the User role from their account.

Note: If you want all new users restricted by team, you can configure Secret Server to assign the Team Limited User role as the default upon creation of a new user.

1. Navigate to **Administration**. The Admin Side Panel appears:

2. Click **Users, Roles, Access** and select **Teams** in the list. The Teams page appears:

Teams

Teams allow you to restrict users from seeing other Users, Groups or Sites that are not specifically included in the same team. In order to get started with Teams, you will need to apply a Role modification to your users.

[Learn More](#)

12 Items Include Inactive

TEAM NAME ↑	ACTIVE
Admin Team	Yes
Marrio's QA Team	Yes

3. Click the **Create Team** button. The Create Team popup page appears:

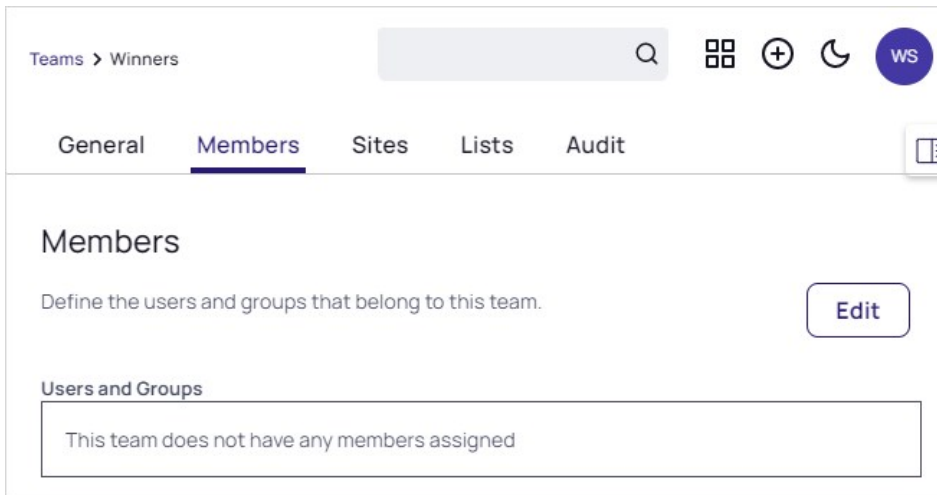
Create Team

Team Name *

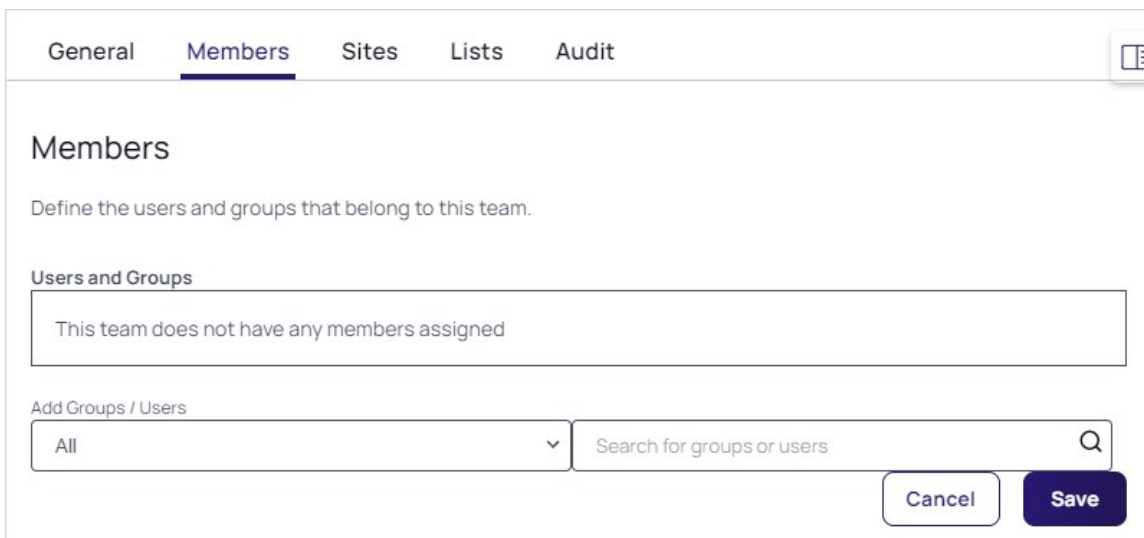
Team Description

4. Type the name for the new team in the **Team Name** text box.

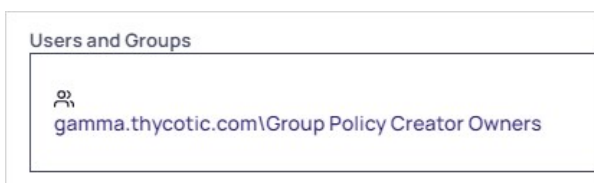
- (Optional) Type a description in the **Team Description** text box.
- Click the **Create Team** button. The new team's Members tab appears:



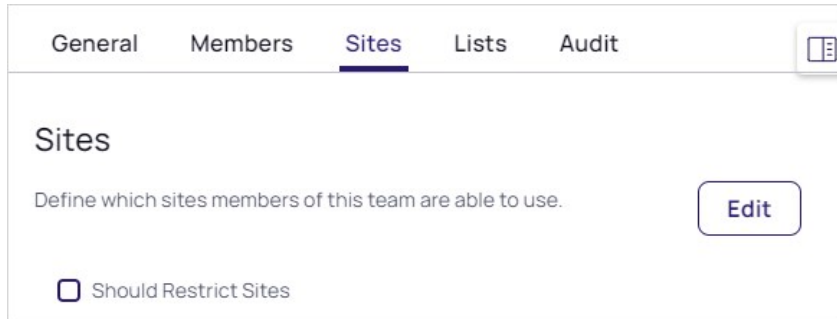
- Click the **Edit** button. An Add Groups / Users section appears:



- Click the first **Add Groups / Users** dropdown list to select a domain.
- Click the search box. A dropdown listing the users and groups in that domain appears. You can also type the name of desired users or groups in the search box.
- Click the desired user or group. The user or group appears in the Users and Groups box:

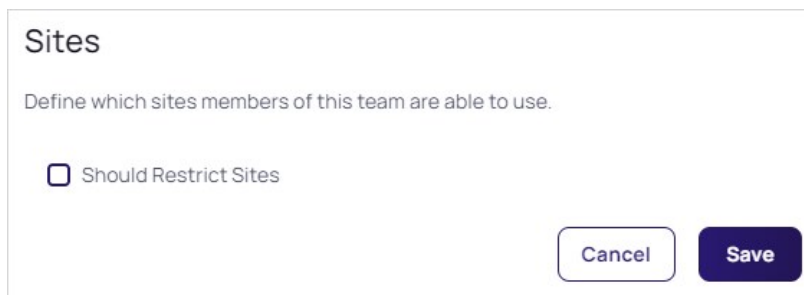


11. Repeat the process for additional users and groups.
12. Click the **Save** button.
13. Click the **Sites** tab:



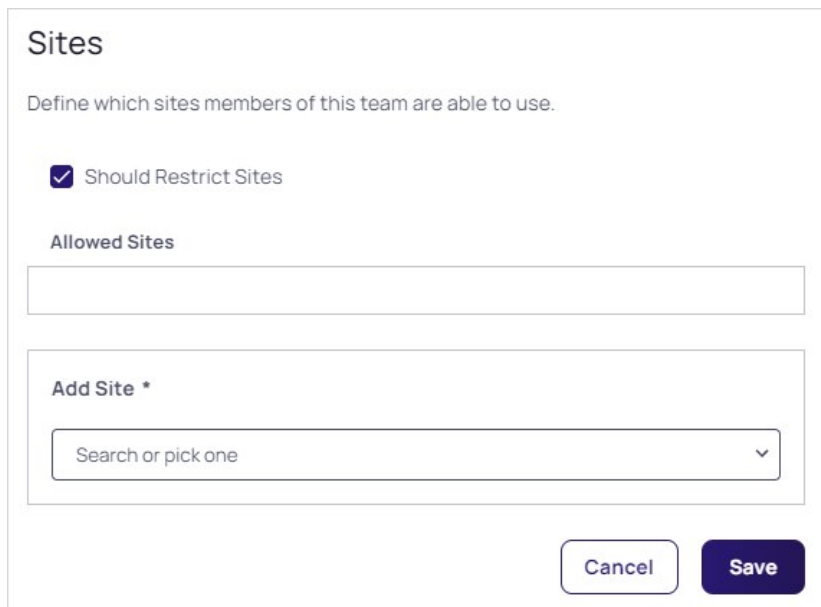
The screenshot shows a configuration page with tabs for General, Members, Sites, Lists, and Audit. The 'Sites' tab is selected. Below the tabs, the title 'Sites' is followed by the instruction 'Define which sites members of this team are able to use.' and an 'Edit' button. At the bottom, there is a checkbox labeled 'Should Restrict Sites' which is currently unchecked.

14. Click the **Edit** button. The page becomes editable:



The screenshot shows the 'Sites' configuration page in an editable state. The 'Edit' button from the previous step is now a 'Save' button, and a 'Cancel' button has appeared. The 'Should Restrict Sites' checkbox remains unchecked.

15. Click to select the **Should Restrict Sites** check box. A Site dropdown list appears:



The screenshot shows the 'Sites' configuration page with the 'Should Restrict Sites' checkbox now checked. Below this, there is a section titled 'Allowed Sites' with an empty text box. Underneath, there is an 'Add Site *' section containing a dropdown menu with the placeholder text 'Search or pick one' and a downward arrow.

16. Click the **Add Site** list to select a site to restrict the team to. The selected site appears in the Allowed Sites box:

Sites

Define which sites members of this team are able to use.

Should Restrict Sites

Allowed Sites

Gamma-Engines	Remove
---------------	--------

Add Site *

[Cancel](#) [Save](#)

17. Click the **Save** button.

18. Click the **Lists** tab:

General Members Sites **Lists** Audit

Lists

Define which lists members of this team are able to use. [Edit](#)

Should Restrict Lists

19. Click the **Edit** link. The page becomes editable:

Lists

Define which lists members of this team are able to use.

Should Restrict Lists

[Cancel](#) [Save](#)

20. Click the **Should Restrict Lists** check box.

Lists

Define which lists members of this team are able to use.

Should Restrict Lists

Allowed Lists

Add List *

21. Click the **Add List** dropdown list to select what lists the team members have access to. The chosen list appears in the Allowed Lists box:

General Members Sites Lists Audit

Lists

Define which lists members of this team are able to use.

Should Restrict Lists

Allowed Lists

TestRemove

Add List *

22. Repeat the process for any additional lists.

23. Click the **Save** button.

Note: You cannot delete teams because of auditing restrictions.

1. In Secret Server, click the **Admin** menu item. The Administration page appears.
2. Click the **Teams** button in the list. The Teams page appears:

TEAM NAME	ACTIVE
Team_Unrestrict_01	Yes
team3	Yes
TeamDup	Yes
Teams_01_Un	Yes
The A Team	Yes

3. Click the table row for the desired team. That team's page appears:

TEAM		EDIT
Team Name *	Accounting	
Description	Accounting leads	
Active	Yes	

4. On the **General** page, click the **Edit** button. The tab becomes editable:

TEAM

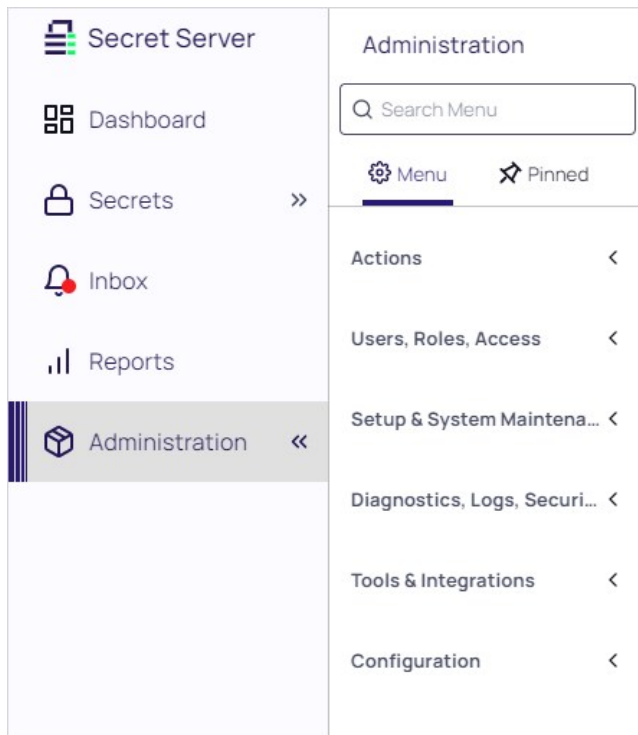
Team Name *

Description

Active

5. Click the **Active** check box to deselect it.
6. Click the **Save** button. The team is deactivated.

1. Navigate to **Administration**. The Admin Side Panel appears:



2. Click **Users, Roles, Access** and select **Teams** in the list. The Teams page appears:

Teams allow you to restrict users from seeing other Users, Groups or Sites that are not specifically included in the same team. In order to get started with Teams, you will need to apply a Role modification to your users.

[Learn More](#)

12 Items Include Inactive

TEAM NAME ↑	ACTIVE
Admin Team	Yes
Marrio's QA Team	Yes

3. Click the table row for the desired team. That team's page appears on the General tab:

Team

Teams allow you to restrict users from seeing other Users, Groups or Sites that are not specifically included in the same team. In order to get started with Teams, you will need to apply a Role modification to your users.

[Learn More](#)

Team Name * Winners

Description Folks That Win

Active * Yes

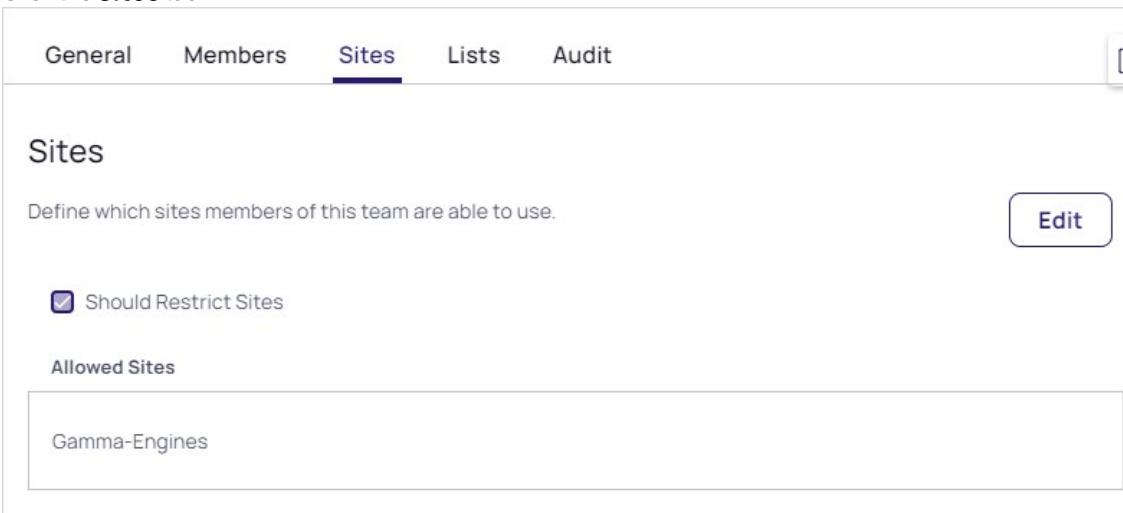
[Edit](#)

4. Click the **Edit** button to change:

- The team name
- The team's description
- The team's status

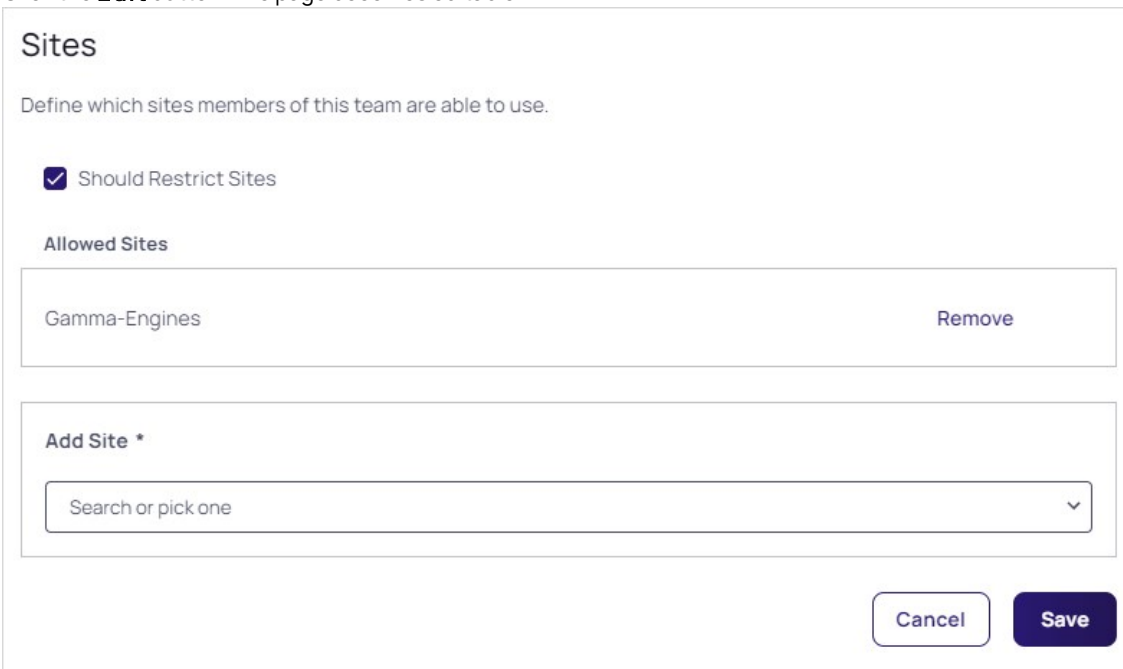
5. To restrict the visible sites:

1. Click the **Sites** tab:



The screenshot shows a web application interface with a navigation bar at the top containing tabs: General, Members, Sites (which is selected and underlined), Lists, and Audit. Below the navigation bar, the main content area is titled "Sites". Underneath the title, there is a descriptive text: "Define which sites members of this team are able to use." To the right of this text is an "Edit" button. Below the text is a checked checkbox labeled "Should Restrict Sites". Underneath the checkbox is the heading "Allowed Sites" followed by a table with one row containing the text "Gamma-Engines".

2. Click the **Edit** button. The page becomes editable:



The screenshot shows the same "Sites" page as in the previous image, but now it is in an editable state. The "Edit" button is no longer present. The "Should Restrict Sites" checkbox remains checked. The "Allowed Sites" table now has a "Remove" button next to the "Gamma-Engines" entry. Below the table is a section titled "Add Site *" which contains a search input field with the placeholder text "Search or pick one" and a dropdown arrow. At the bottom right of the page, there are two buttons: "Cancel" and "Save".

3. If necessary, click to select the **Should Restrict Sites** check box.
4. Click the **Remove** button next to any sites you want to remove.
5. Click the Add Site dropdown list to select sites you desire to add. The selected site appears in the Allowed Sites table:

The screenshot shows a section titled "Allowed Sites" with a list of two entries: "Gamma-Engines" and "Omega-Engines". Each entry has a "Remove" button to its right.

Site Name	Action
Gamma-Engines	Remove
Omega-Engines	Remove

6. Click the **Save** button.

6. To edit the team's member users or groups:

1. Click the **Members** tab:

The screenshot shows the "Members" tab selected in a navigation bar with options: General, Members, Sites, Lists, Audit. Below the navigation bar, the "Members" section is titled "Members" with the instruction "Define the users and groups that belong to this team." and an "Edit" button. Underneath, the "Users and Groups" section contains a list with one entry: "gamma.thycotic.com\Group Policy Creator Owners".

2. Click the **Edit** button. An Add Groups / Users section appears:

The screenshot shows the "Members" tab with the "Edit" button clicked. The "Users and Groups" section now includes a "Remove" button next to the entry "gamma.thycotic.com\Group Policy Creator Owners". Below this, a new "Add Groups / Users" section is visible, featuring a dropdown menu currently set to "All", a search box labeled "Search for groups or users", and "Cancel" and "Save" buttons.

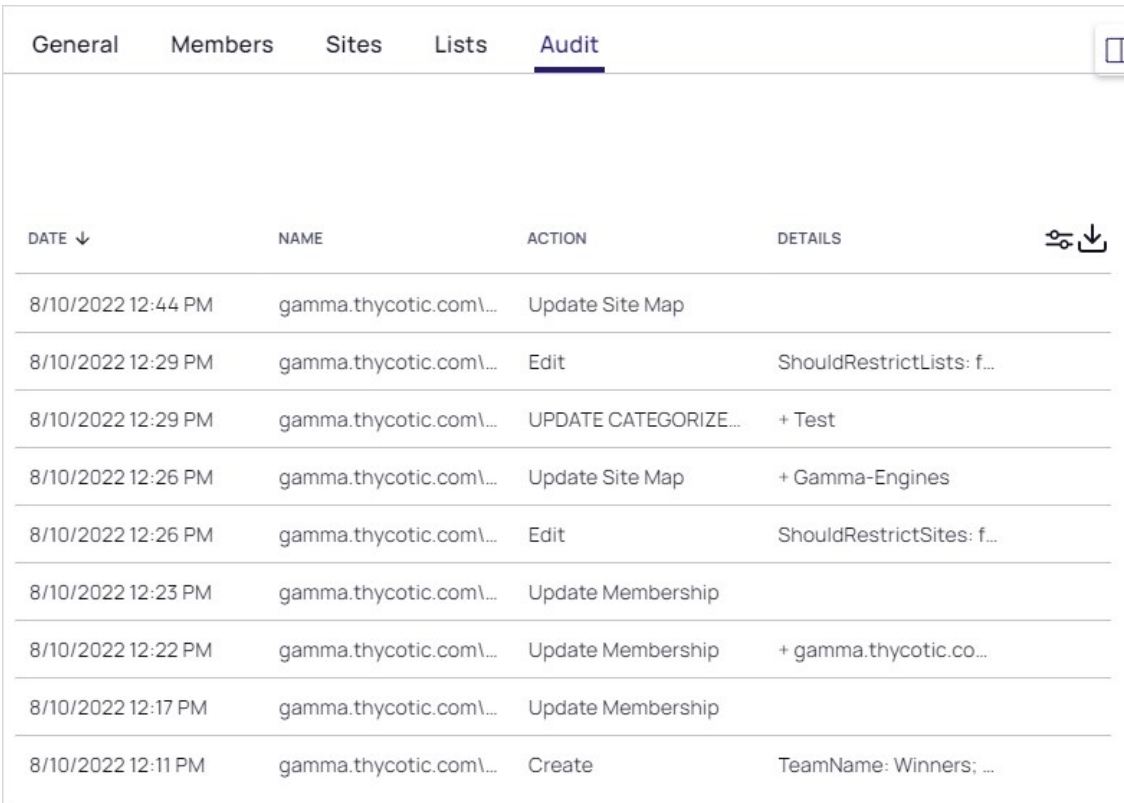
3. Click the first **Add Groups / Users** dropdown list to select a domain.

4. Click the search box. A dropdown listing the users and groups in that domain appears. You can also type the name of desired users or groups in the search box.

5. Click the desired user or group. The user or group appears in the Users and Groups box:

![[image-20220810122124607](../../../../Thycotic Files/Repos/Thycotic.SecretServer.Docs/user-teams/creating-teams/images/image-20220810122124607.png)]

6. Repeat the process for additional users and groups to add.
 7. Click the **Remove** button next to a user or group to delete it.
 8. Click the **Save** button.
7. View events for the team using its audit trail:
1. Click the **Audit** tab:



DATE ↓	NAME	ACTION	DETAILS	⚙️ ↓
8/10/2022 12:44 PM	gamma.thycotic.com\...	Update Site Map		
8/10/2022 12:29 PM	gamma.thycotic.com\...	Edit	ShouldRestrictLists: f...	
8/10/2022 12:29 PM	gamma.thycotic.com\...	UPDATE CATEGORIZE...	+ Test	
8/10/2022 12:26 PM	gamma.thycotic.com\...	Update Site Map	+ Gamma-Engines	
8/10/2022 12:26 PM	gamma.thycotic.com\...	Edit	ShouldRestrictSites: f...	
8/10/2022 12:23 PM	gamma.thycotic.com\...	Update Membership		
8/10/2022 12:22 PM	gamma.thycotic.com\...	Update Membership	+ gamma.thycotic.co...	
8/10/2022 12:17 PM	gamma.thycotic.com\...	Update Membership		
8/10/2022 12:11 PM	gamma.thycotic.com\...	Create	TeamName: Winners; ...	

2. Audit events occur when:
 - The team is created
 - General tab: name, description, or active status is changed
 - Sites tab: restrictions are added, removed, or changed
 - Members: users or groups are added or removed

Users can view other users not in their teams if that user already had a connection, such as a shared secret, with the other user prior to setting up the team restrictions.

The API does not restrict who can be assigned if they use the known group ID of a user or group not in their team. This is designed so secret permissions can be saved across teams without removing the permissions of another team.

1. Navigate to **Admin > See All**. The Administration page appears:

What are you looking for?

Search for an admin option



Simplified View ▾



Actions

Secret Server features that perform important jobs



Setup & System Maintenance

Setup your Secret Server system and keep it running with Licensing, Backups, Imports, Networking options, and more



Users, Roles, Access

These features help you organize users & permission settings within Secret Server



Diagnostics, Logs, Security

Reference options for diagnostics, logs, and security features



Tools & Integrations

Find Secret Server tools and other product integrations here

2. Type and then click **Users** in the search text box. The View User page appears:

View User

User Name	[Redacted]
Display Name	[Redacted]
Email Address	[Redacted]@thycotic.com
Domain	Local
Two Factor	< None >
Enabled	Yes
Locked Out	No
Application Account	No

IP Address Restrictions

None

Restricted By Team No

You can see if the user belongs to a team, and if so, what teams the user belongs to. If the Restricted by Team line says *No*, it means the user has been granted the Unrestricted by Teams permission, which means the user can view all users, groups, and sites.

Users

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Bulk operations on users can also be performed from the **Users** page. Select one or more users using the check boxes beside the **User Name** column, or select all or none by toggling the check box in the header row. Once the appropriate users have been selected, use the Bulk Operation list at the bottom of the grid to select an action. Bulk operations on users currently include enabling or disabling user access, as well as configuring users for email or RADIUS two-factor authentication.

User settings can be modified by clicking the username in the **User Name** column on the **Users** page. Search for users using the search bar at the top of the grid. To show users that are marked inactive, check the **Show Inactive Users** box below the grid.

To manually create a single user, navigate to **Administration > Users** and click the **Create New** button. On the subsequent page, you can enter the relevant information for a user.

Note: To add many users from your Active Directory setup, you can use Active Directory synchronization (see [Active Directory Synchronization](#)).

You cannot delete users per se because of auditing requirements; however, deactivating the user from the [User Settings page](#) accomplishes the same thing. See [Removing Deactivated User PII](#) for eliminating all traces of a deactivated user.

The following settings are found in the **Administration > Configuration** page, inside the **Local User Passwords** tab. These settings apply to users that were created manually, not users brought into Secret Server through Active Directory synchronization:

- **Allow Users to Reset Forgotten Passwords:** If enabled, the "Forgot your password?" link appears on all users' login screens. Clicking on this link prompts the user to enter the email address that is associated with the user's Secret Server account. If the email address is found, then an email containing a link for password reset is sent. Note that this only works for local user accounts and not for Active Directory accounts. Secret Server
- **Enable Local User Password Expiration:** When enabled, Secret Server forces a password change for a user after a set interval. After the interval time has elapsed, the next time the user attempts to log in, they are prompted for the old password, a new password, and a confirmation of the new password. The new password is validated against all the password requirements. Newly created local users are also be prompted to change their password upon logging into Secret Server for the first time when this setting is enabled.
- **Enable Local User Password History:** If enabled, this prevents a user from reusing a password. For example, if set to "20 Passwords", this would prevent the user from using a password they have used the previous 20 times. This in conjunction with "Enable Minimum Local Password Age" helps ensure that users are using a new and unique password frequently rather than recycling old passwords.
- **Enable Minimum Local User Password Age:** If enabled, the value for this setting reflects the minimum amount of time that needs to elapse before a password can be changed. This prevents a user from changing their password too frequently, which allows them to quickly re-use old passwords.
- **Local User Password is valid for:** If enabled, this is the interval that a local user password is valid before it must be changed (see "Enable Local User Password Expiration" setting for details). If this setting is disabled, the entered value displays as "Unlimited".
- **Lowercase Letters Required for Passwords:** Force all user Secret Server login passwords to contain at least one lowercase letter.
- **Minimum Password Length:** Force all user Secret Server login passwords to contain a set minimum number of characters.
Note: The maximum number of characters is 1024.
- **Numbers Required for Passwords:** Force all user Secret Server login passwords to contain at least one number.
- **Symbols Required for Passwords:** Force all user Secret Server login passwords to contain at least one symbol, such as !@#%&*.
- **Uppercase Letters Required for Passwords:** Force all user Secret Server login passwords to contain at least one uppercase letter.

Overview

General Data Protection Regulation (GDPR) adherence raises the possibility that Secret Server users may make a data removal claim against a Secret Server administrator. This requires removing any personally identifiable information (PII) in Secret Server for that individual.

To address this, Secret Server has a button that automatically removes most PII for any deactivated user.

Removing the PII

1. Remove the user from Active Directory (AD). See [Active Directory Considerations](#) below.
2. In Secret Server, go to **Admin > Users**. The Users page appears.
3. Click the user name link for the desired user. The View User Page appears.
4. Click the **Remove Personally Identifiable Information** button. A confirmation dialog box appears.

Important: Once you confirm, the user cannot log on to Secret Server. Click the Cancel button if you are not positive this is what you want to do.

Clicking the **OK** button will change these to random values or set them to null:

- Username
- Display name
- Password
- Personal folder name
- Personal group name
- RADIUS username

In addition:

- The user's AD GUID is cleared
- The user's email address is removed from their record
- The user's name is replaced with "< redacted >" in event audits where it can be clearly identified.
- The PII removal is recorded in the user's audit

5. Click the **OK** button. The removal begins. Once complete, the Remove PII button disappears for that user.
6. (Optional) Run a query that scans the entire Secret Server database for the removed strings. You may want to do this because the process cannot find *all* potential instances of USER PII throughout Secret Server, such as that in secret names or notes.

Note: You can create an Event Subscription to "remove user PII" events.

Active Directory Considerations

We recommend removing the user from AD before removing the PII. If you remove the PII without first removing the user from AD, the user is reintroduced into Secret Server on subsequent AD synchs. This creates a new user account in Secret Server, which might require you to to disable this new user account and remove its PII too (after removing the AD user).

If a user fails their login too many times (specified in the **Local User Passwords** section of the configuration page), their account is locked out and they are not be able to log in.

To unlock the account:

1. Log on as an administrator.
2. Click on **Admin > Users**.
3. Click on the user who is locked out.
4. Click **Edit**.
5. Click to deselect the **Locked out** check box.
6. Click **Save**.

Secret Server users can be set up with many different login requirements. For example, you can force strong Login passwords by requiring a minimum length and the use of various character sets.

The following settings are available under the **Administration > Configuration** page, inside the **Login** tab:

Note: "Allow Autocomplete" is now permanently on.

- **Allow Remember Me:** This option enables the Remember Me checkbox on the Login screen. When a user chooses to use remember me, an encrypted cookie is set in their browser. This enables the user to revisit Secret Server without the need to log in. This cookie is no longer be valid when the remember me period has expired. They then have to enter their login information again. This option allows users to remain logged in for up to a specific period (specified in the "Remember Me Is Valid for" setting mentioned below). This option can be a security concern as it does not require re-entry of credentials to gain access to Secret Server.

Note: "Remember me" is only visible if the "Allow Remember Me" setting is enabled. This is the period that the remember me cookie mentioned above is valid. For example: if set to one day, then users taking advantage of "remember me" have to log in at least once a day. To set a time value (minutes, hours, or days), uncheck the Unlimited checkbox.

- **Enable RADIUS Integration:** Allow for RADIUS server integration with your user login authentication. Other RADIUS settings appear upon enabling this option. These settings are discussed in [RADIUS Authentication](#).
- **Maximum Concurrent Logins Per User:** This setting allows a user to log into Secret Server from multiple locations at once without logging out their sessions at other locations.
- **Maximum Login Failures:** Set the number of login attempts allowed before a user is locked out of their account. Once locked out, they need a Secret Server administrator to reset their password and enable their account. For details on how to reset a locked account, see [Creating Users](#).
- **Require Two Factor for these Login Types:** This setting specifies which types of login require two-factor authentication:
 - Website and Web service Log on
 - Website log on only
 - Web service log on only
- **Visual Encrypted Keyboard Enabled:** This setting enables a visual keyboard for logins.
- **Visual Encrypted Keyboard Required:** This setting requires a visual keyboard for logins.

User Administrators can also set another group or user as the *user owner* for a Secret Server local user. User owners can manage and edit just that user. For example, a developer might need to unlock or reset the password for an application account but should not have access to all users. Set **Managed by to User Owners** on a user and then select **Groups** or **Users**. Note that Unlimited Administrator mode can still be used to manage groups with user owners assigned.

Note: Users can set their preferences by hovering on their profile icon in the top right and selecting preferences.

General Tab

The following configuration settings are available for users under the General tab:

- **Date Format and Time Format:** Date and time format displayed for a user in Secret Server.
- **Language and My Theme:** Customize the look of Secret Server on a per user basis. For details, see [Themes](#).
- **Mask passwords when viewing Secrets:** When enabled, this masks the Password text box for a secret. There is a configuration setting that forces this to be enabled for all users. For details on password masking, see [Setting Up Password Masking](#).
- **Send email alerts when dependencies fail to update:** Enables emails to be sent when dependencies fail to update.
- **Send email alerts when Heartbeat fails for Secrets:** When enabled, the user is emailed when a heartbeat fails for any secret the user has view permission to.
- **Send email alerts when Secrets are changed:** Enables emails to be sent on all changes of any secret that the user has view permission. There is a limit of one mail per five minutes per edit of the same user. For example, if user "User1" edits the secret twice within this grace period, only one email is sent.
- **Send email alerts when Secrets are viewed:** Enables emails to be sent on all views of any secret that the user has view permission. There is a limit of one email per five minutes per view of the same user. For example, if user "User1" views the secret twice within this grace period, only one email is sent.
- **Show the full folder path on search results:** Enables the full path to be displayed in the Folder column on the Home page.
- **Use the TreeView control for search on the home screen:** Enables the TreeView control for the Search tab on the Legacy Home screen. This option does not apply to the Dashboard.

Launcher Tab

The following configuration settings are available to users on the Launcher tab:

- **Allow Access to Printers, Allow Access to Drives, Allow Access to Clipboard:** Allow access to various items when using the launcher.
- **Connect to Console:** Allows you to connect to remote machines using the Remote Desktop launcher and connects as an administrator. This is the equivalent of using the `/admin` or `/console` switch when launching Remote Desktop.
- **Use Custom Window Size:** Checking this box displays Width and Height text boxes for the user to specify a custom window size for an RDP launcher.

The following restriction settings are available:

- **Enable Login Policy:** If enabled, this simply displays the policy. To force the acceptance of the policy.
- **Force Inactivity Timeout:** This setting is the time limit on idle Secret Server sessions. Once a session expires, the user must login again with their username and password.
- **Force Login Policy:** This setting forces the checking of the "I accept these terms" checkbox before allowing the user to login to SS.
- **IP Restrictions:** This setting can be entered by going to **Administration > IP Addresses**. In there, you can enter the IP ranges you wish your users to use. To configure a user to use the ranges, navigate to the **User View** page and click the **Change IP Restrictions** button. In the subsequent page, you can add all the ranges you want your user to use.
- **Login Policy Agreement:** The Login Policy Agreement is displayed on the login screen. You can change the contents of the Login Policy Statement by editing the file `policy.txt`. By default, this is not enabled. The settings to enable this are accessed by first navigating to **Administration > Configuration** and going into the **Login** tab. Then click the **Login Policy Agreement** button.

Below is a brief explanation of each text-entry field or control:

- **Display Name:** Text that is used throughout the user interface, such as in audits.
- **Domain:** If a drop-down list is visible, selecting a domain from the list is one way to set the expected domain of the user. However, a more dynamic way to have this text-entry field (and all the other text-entry fields) set is through Active Directory synchronization.
- **Email Address:** Email address used for Request Access, email two-factor authentication, and the like.
- **Email Two-Factor Authentication:** On a login attempt, the user has an email sent to the email address entered above. This email contains a pin code that the user needs to log into the account. See [Email Two-Factor Authentication](#) for details.
- **Enabled:** Disabling this control removes the user from the system. Effectively, this is the way to delete a user. Secret Server does not allow complete deletion of users due to auditing requirements. To re-enable a user, navigate to the **Administration > Users page**, check the **Show Inactive Users** checkbox just under the **Users** grid, and edit the user to mark them enabled (see [Configuring Users](#)).
- **Locked Out:** If checked, then this user has been locked out of the system due to too many login failures. To remove the lock, uncheck the check box. For more details on locking out users, see Maximum Login Failures setting described in the Login Settings section.
- **Password:** Login password for the user. For the various login settings, see Login Settings section.
- **RADIUS Two-Factor Authentication:** This text-entry field only appears if RADIUS authentication is enabled in the configuration. On a login attempt, the user must enter the RADIUS token sent from the RADIUS server. See [RADIUS Authentication](#).
- **RADIUS User Name:** This text-entry field only appears if the above RADIUS Two Factor Authentication setting is enabled. This is the username the RADIUS server is expecting. See [RADIUS Authentication](#).
- **User Name:** Login name for the user.

Note: A new user is assigned the User role by default. For more information on roles, see "Roles."

Webservices

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Secret Server provides a suite of webservices which can be used to retrieve and update secrets, and folders. The webservices allow Secret Server to be accessed using the mobile apps as well as custom built integrations. The webservices are secure and require authentication in the same manner as regular access to Secret Server. All actions that involve data are also logged, such as secret views, updates, and adds.

Webservices can be enabled at the **Administration > Configuration** general tab. Enabling webservices simply makes the ASP.NET webservices built into Secret Server available. They are found under `/webservices/sswebservice.asmx` in your Secret Server directory. They run on the same port as the Web application. You can view them with a browser to see the functionality that is offered. Specific webservice functionality is documented in the Secret Server Webservice API guide.

Secret Server also provides a webservice that uses integrated Windows authentication instead of a username and password. This webservice can be used in an application or script to access Secret Server and retrieve secrets without storing the login credentials in the application or configuration file.

Note: See the [Windows Integrated Authentication Webservice](#) (KBA) article for more advanced technical information on using this webservice.

Secret Workflows

Note: Please click the table of contents on the left to see any sub-pages to this one. Click the table of contents on the right to see headings on this page.

Starting in 10.6, Secret Server introduced *access-request workflows*. These allow users to build more complex interactions based on events within Secret Server than currently possible. The first release of workflows offers access requests. Workflow templates define the series of steps and reviewers required for an access request. You can assign workflows to secrets or secret policies.

With Access-Request Workflow Templates, you can:

- Require that multiple people approve a request before access is granted
- Require multiple workflow steps, each with different reviewers and number of required approvers, if desired.
- Select "Owners" as a review group

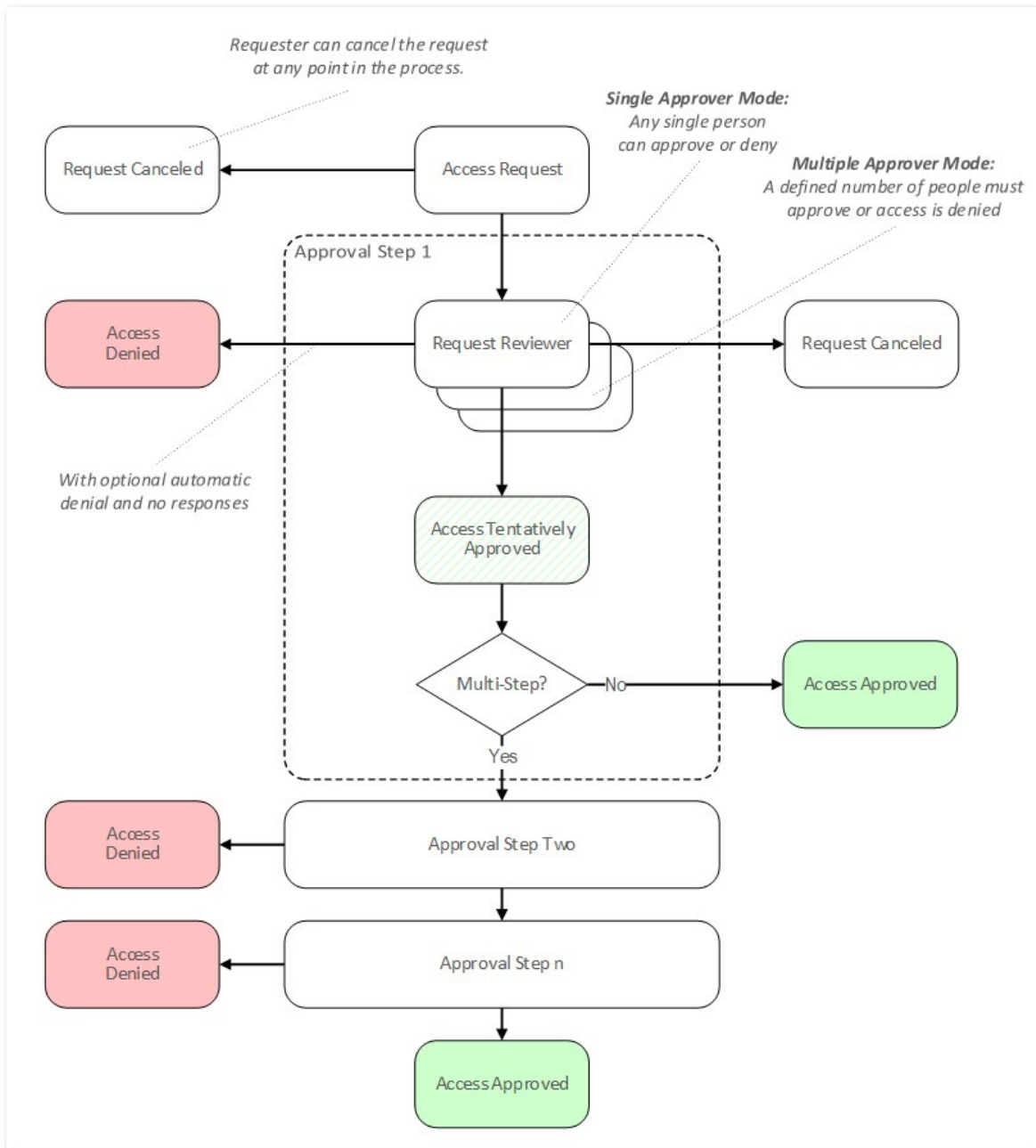
Note: Access Requests already existed in Secret Server, but with 10.6 they become much more powerful. Previously, if access requests enabled on a secret, requests were granted after a single reviewer approved the request. Now, approval workflows can require multiple approvers, and multiple approval levels.

The original access requests are one level or step—anyone approving approves the request—no other input is required. Workflows allow up to 15 approval steps where approval by reviews in step 1 moves the request to step 2, approval at step 2 moves it to step 3 and so forth. Denial at any step denies the request.

The new workflow feature can be configured where one approver at a given step is not enough. In effect, approvers in each step can "vote" for approval—you stipulate how many approvers at a step must approve for the approval to move on to the next step.

The following diagram is the entire process summarized:

Figure: The Approval Process Workflow



In general, "simple access requests," the only type available to older versions of Secret Server, are the same as a one-step stepped approval. The major exception is that with stepped requests, once a workflow access request has been approved, denied, or canceled, its status cannot be changed. In contrast, simple, non-workflow, access requests retain the original behavior of allowing a request to be approved after it has been denied or denied after it has been approved.

Release notes

You can configure workflow steps to time out after a specified number of minutes. Workflow administrators can define approval workflows that notify a different set of users if a step in the workflow is not responded to within a specified time period.

Applications include:

- Improving responsiveness to access requests on time-sensitive secrets by moving the request to another step if not responded to quickly.
- Overflowing to a different region so a secret can be accessed by users in different time zones with different sets of support personnel responding.

Timed out access requests automatically advance to the next step in the workflow. The last step of the workflow cannot time out and must be approved to access the secret.

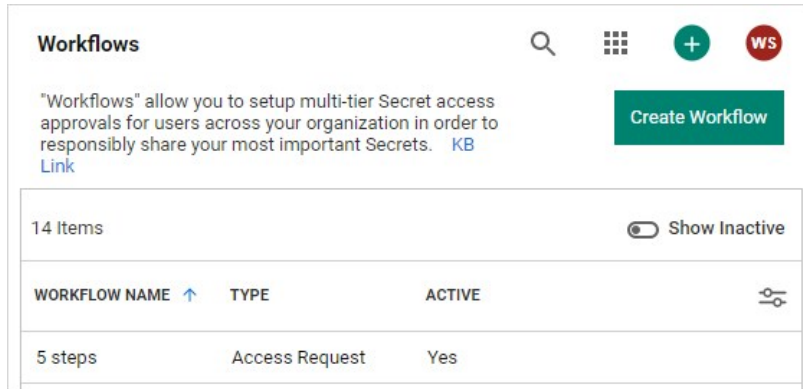
You can apply multiple timeouts to create workflows that can cascade through multiple steps if the previous steps do not receive the required approvals.

Denying or canceling a request in any step stops the workflow and stops any time out to the next step.

Important: Once a workflow step times out, only the reviewers in the next step can approve the request. If you want reviewers from the initial step to respond after the initial request has timed out, add them to the reviewers of the next step.

To access workflows:

1. Go to **Admin > Workflows**. The Workflows page appears:



Workflows

"Workflows" allow you to setup multi-tier Secret access approvals for users across your organization in order to responsibly share your most important Secrets. [KB Link](#)

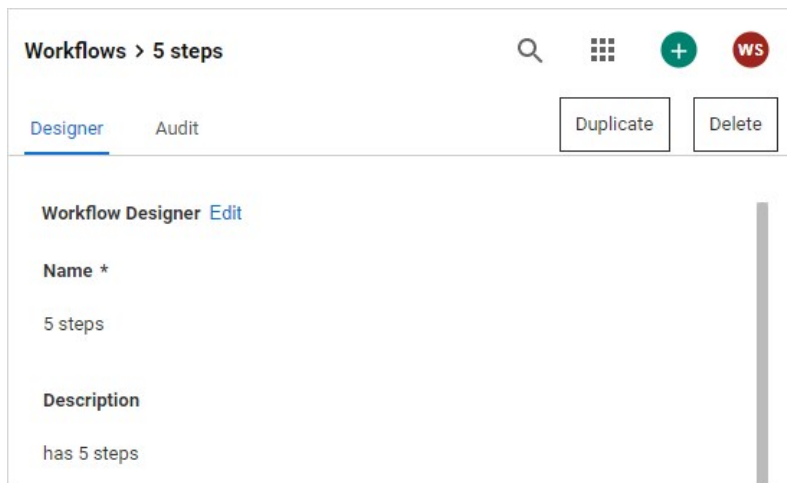
[Create Workflow](#)

14 Items Show Inactive

WORKFLOW NAME ↑	TYPE	ACTIVE	
5 steps	Access Request	Yes	

The page lists all active workflows.

2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click any workflow in the list to go to the designer page for that workflow:



Workflows > 5 steps

[Designer](#) [Audit](#) [Duplicate](#) [Delete](#)

Workflow Designer [Edit](#)

Name *

5 steps

Description

has 5 steps

1. Click **Admin > Secret Policy**. The Secret Policy page appears:

Secret Policy

[Explain](#)

Use Secret Policies to establish consistent sets of security requirements assigned at the folder or Secret level.

< 1 to 5 of 5 >

SECRET POLICY NAME	DESCRIPTION	ACTIVE
Disabling_policy	test	Yes
EPP Testing		Yes
tjwSEcretPolicy2	make Thomas approver	Yes
tjwSecretPolicyForcedApproval	Forces Legacy Approval	Yes
Web Password Policy	rpc auto, rpc priv account, rpc daily	Yes

Show Inactive

[← Back](#) [+ Create New](#)

2. For this instruction, we are going to create a new policy.
3. Click the **+ Create New** button. Another Secret Policy page appears:

Secret Policy

[Explain](#)

- Any items selected as 'Default' will be applied on the creation of any Secret that has this Secret Policy applied to it.
- Any items selected as 'Enforced' will be applied to all Secrets that have this Secret Policy applied to it.
- 'Enforced' settings cannot be changed on the Secret.
- Certain settings will only be applied to a Secret if they are valid settings for the Secret.

Secret Policy Name *

Description

Active

SECTION	SECRET POLICY ITEM NAME	SETTING	VALUE
General	Site		< Not Set > ▼
Remote Password Changing	Auto Change		< Not Set > ▼

- Type the new policy name in the **Secret Policy Name** text box.
- Scroll down the page to the **Security Settings** section of the unlabeled table.
- Click the **Enable Requires Approval for Access** list and select **Enforced**.
- Click to select the check box next to the list. The Assign Approvers popup page appears:

Assign Approvers ✕

Select the users or groups to be approvers.

NAME

User/Group

8. Click the **Cancel** button. The Request Access Approvers setting become enabled:

Note: You cannot set approvers and use a workflow at the same time. The intent of the next few instructions is avoid attempting to do so, which causes an error.



Security Settings Request Access Approvers
(Dependent on: Enable Requires Approval for Access) Enforced < None >

9. Click the **Request Access Approvers** list and select **Not Set**.

10. Click the **Request Access Workflow** list and select **Enforced**. A new list appears alongside:



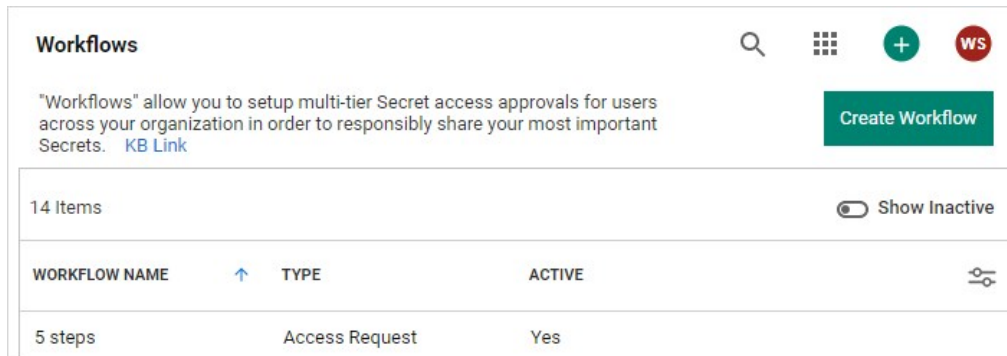
Security Settings Request Access Workflow
(Dependent on: Enable Requires Approval for Access) Enforced < None >

11. Click the new unlabeled list and select the access template workflow to associate with the policy.

12. Click the **Save** button at the bottom of the page. The policy is now available for assignment to secrets and folders, just like any other policy.

Markdig.Syntax.Inlines.EmphasisInline

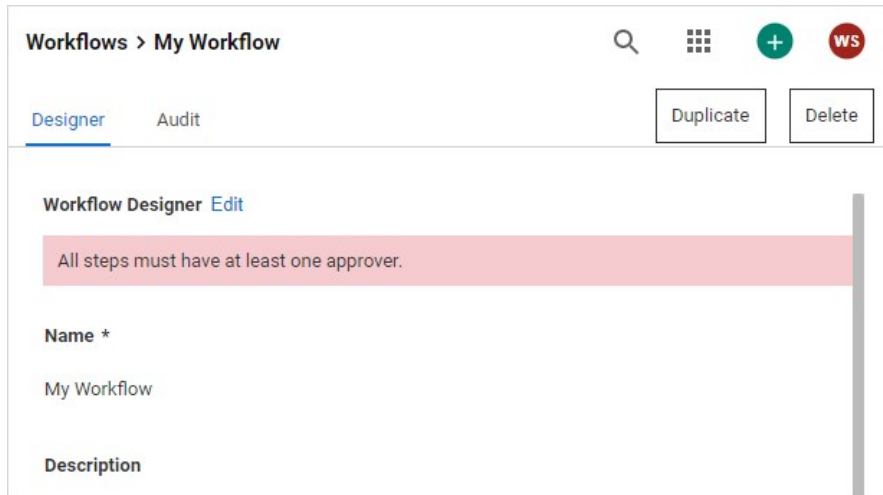
1. Go to **Admin > Workflows**. The Workflows page appears:



2. Click the **Create Workflow** button. The Create Workflow popup appears:

The screenshot shows the 'Create Workflow' popup form. It has a title 'Create Workflow'. Below the title are two text input fields: 'Workflow Name' and 'Description'. At the bottom of the form are two buttons: 'Cancel' and 'Create Workflow'.

3. Type the workflow's name and description (optional) in their text boxes. Once you type the name, the Create Workflow button becomes enabled.
4. Click the **Create Workflow** button. The Edit page for the new workflow appears on the Designer tab.



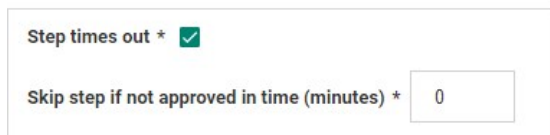
A new workflow has only one empty step by default.

Markdig.Syntax.Inlines.EmphasisInline

1. Click the **Edit** link next to the **Workflow Designer** heading. The page becomes editable.
2. (Optional) Type a name for the first step in the **Step 1 Name** text box, such as "Line Managers."
3. Click the **Add Groups / Users** dropdown list to select the domain where you want to get your approvers from.
4. Type the name of the user or group you desire as an approver in the unlabeled search box to the right. Options appear in the dropdown.
5. Click the desired user or group. It appears in the Approvers table:



6. Repeat as desired.
7. (Optional) To automatically include the owner of the secret the template is assigned to, click to select the **Include owners as reviewers** check box.
8. If you wish to have multiple approvers required on the step, type the minimum required in the **Number of approvers required** text box. Otherwise, leave it set to 1.
9. If you want the step to time out, click to select the **Step Times out** check box. Another text box appears:

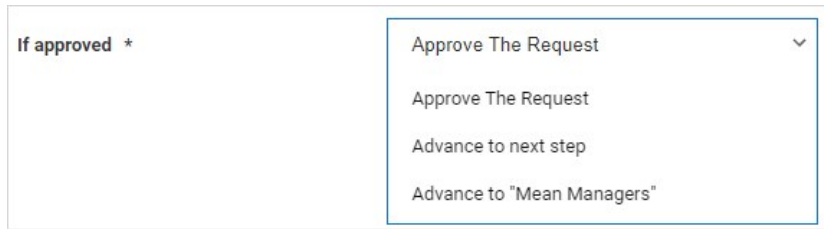


Important: This feature is part of the early release of Secret Server 10.11. The general release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version.

Note:

- Timeout minutes must be a positive integer set to 1 or greater.
- If a step is set to time out and then all following steps are deleted, the step will no longer time out because the last step in a workflow is not allowed to time out.
- Multiple steps can time out, so cascading timeouts are possible. That is, step one times out to step two and step two times out to step three.

1. Replace the 0 in the **Skip step if not approved...** text box with the number of minutes for the desired timeout.
2. Click the **If approved** dropdown list to select what to do next:



The image shows a dropdown menu for the 'If approved' field. The menu is open, showing four options: 'Approve The Request' (selected), 'Approve The Request', 'Advance to next step', and 'Advance to "Mean Managers"'. The selected option is highlighted with a blue background and a small downward arrow on the right.

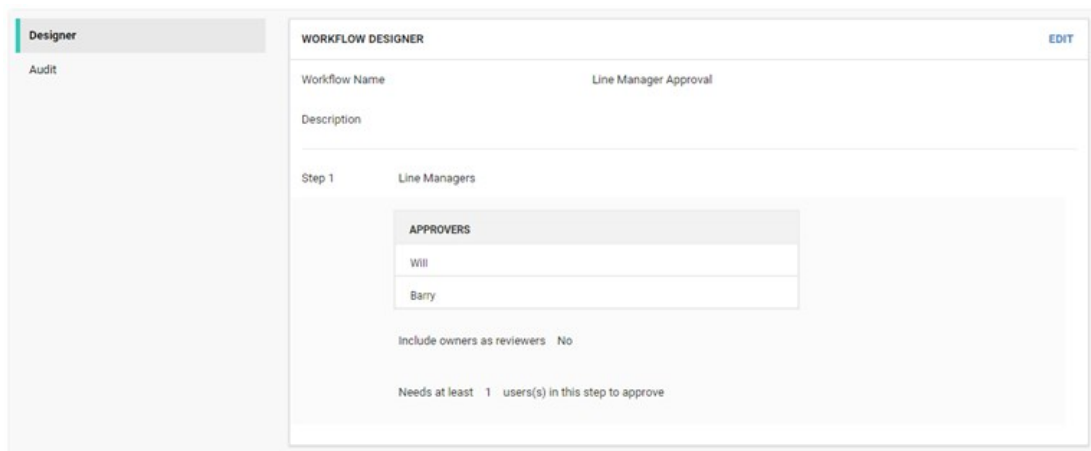
You can:

- Approve the request
- Advance to the next step in a linear fashion
- Jump to another already defined step that is presented as an option in the list box.

Markdig.Syntax.Inlines.EmphasisInLine

Note: There are situations where you might want to have only one workflow step, seemingly doing the same thing as a simple access request. Workflows provide options to require multiple approvers or have owners as approvers, which are not available to simple access requests.

1. Click the **Insert a Step** button. A new step appears below the first two.
2. Repeat the process as for earlier steps. Keep adding steps as needed.
3. Click the **Save** button to create the access-request workflow. The template exits editable mode:

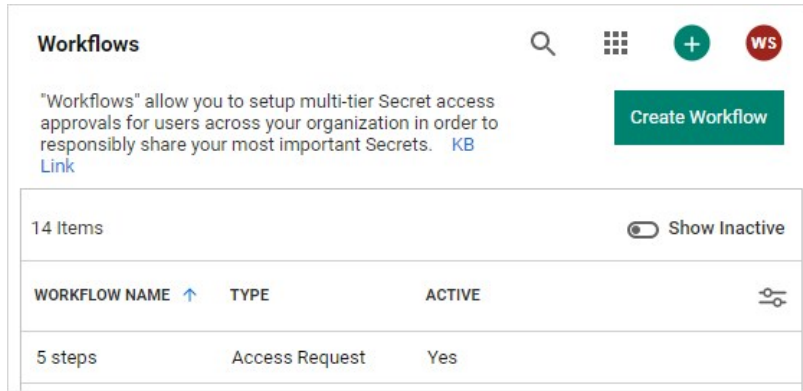


The image shows the 'Workflow Designer' interface. The 'Workflow Name' is 'Line Manager Approval'. The 'Description' field is empty. The 'Step 1' is 'Line Managers'. Below this, there is an 'APPROVERS' section with a list of names: 'Will' and 'Barry'. There is a checkbox for 'Include owners as reviewers' which is currently unchecked. At the bottom, it says 'Needs at least 1 users(s) in this step to approve'.

4. Click the **Workflows** bread crumb link on the top of the page to return to the table.

To delete a workflow:

1. Go to **Admin > Workflows**. The Workflows page appears:



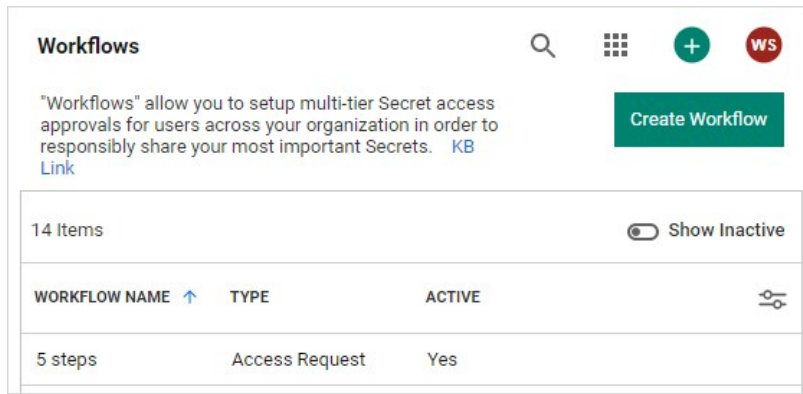
The page lists all active workflows.

2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow to delete in the list to go to the designer page for that workflow (not shown).
4. Click the **Delete** button. A confirmation popup page appears.
5. Click the **Yes, Delete** button.

Note: Because workflows based on the template may still be in play, the template is not completely deleted. Instead, it is inactivated. You can reactivate the template later. See [Accessing the Workflow Designer](#).

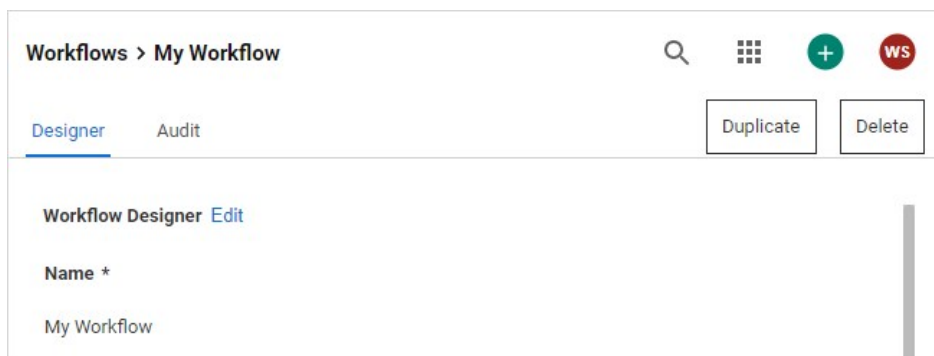
If you need to create a new workflow that is like one you already have, you can save time by copying the similar template and then making the any changes:

1. Go to **Admin > Workflows**. The Workflows page appears:

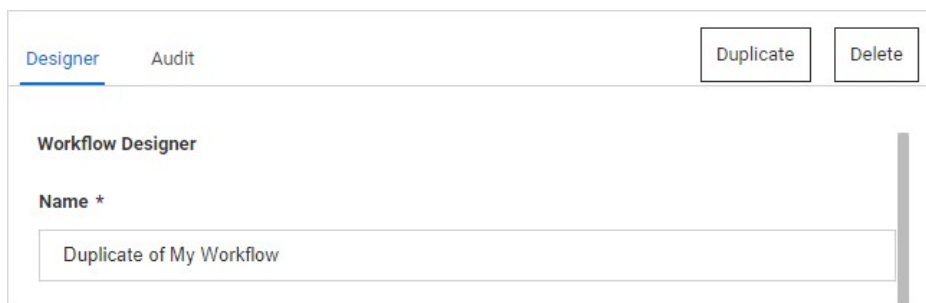


The page lists all active workflows.

2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow you want to copy in the **Workflow Templates** table. That template appears:



4. Click the **Duplicate** button. The new template appears, filled in the same as the original but with a "Duplicate of" name:

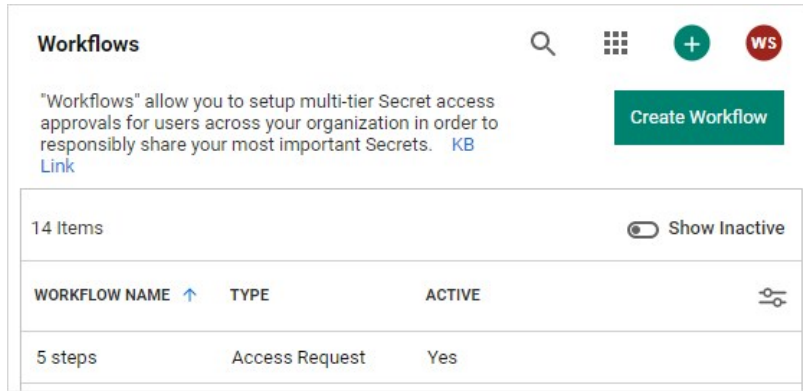


5. Change the name and edit as desired.

6. Click the **Save** button when finished.

To edit the template:

1. Go to **Admin > Workflows**. The Workflows page appears:



The page lists all active workflows.

2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow to edit in the list to go to the designer page for that workflow (not shown).
4. At this stage the process is nearly identical to creating a new workflow. The only difference is many of the parameters and additional steps are already completed. Change them as desired. If you want to eliminate an entire step, click the **Delete This Step** link for that step.

Note: You cannot make any changes to the behavior of a workflow if there are active requests using that template without cancelling those requests. An active request is any unexpired request that has not been approved, denied, or canceled by the user. If you do make an alteration, any requests are canceled and those affected are notified by email so they can resubmit their requests. Any user editing the template is notified when he or she tries to save changes on the canceled request.

Consider the following when setting up an access-request workflow:

- Use multiple-step approval workflows when you need to have different people (such as different departments) sign off on an approval request.
- We do not recommend assigning equally important approvers or groups to multiple steps. Having a single step with multiple approvers works better. Remember, steps are best used for hierarchical approval--an approval chain.
- A reviewer can only respond to a request once. If you have the same user as a reviewer in multiple steps, that approver cannot respond if he or she already responded on an earlier step. In addition, the reviewer's earlier approval does **not** count towards the number of approvals required in later steps. Thus, if you want to assign the same user as a reviewer in multiple steps, make sure that you have enough reviewers in each step to approve without that user.
- A well-crafted workflow design ensures there are enough approvers in a group to satisfy the multiple approver (x of n reviewers must approve) requirement, but group membership can change after the workflow is created. Thus, if you remove members from groups used by workflows, ensure there are still enough members in those groups to approve requests.
- Once a workflow step times out, only the reviewers in the next step can approve the request. If you want reviewers from the initial step to respond after the initial request has timed out, add them to the reviewers of the next step.

Secret Server Release Notes

Note: As of Secret Server 10.8, Secret Server Cloud release notes are included in the main release notes. Scroll down for legacy Secret Server Cloud.

[Secret Server 11.2.000003](#)

Release dates:

- General Availability:
 - June 7, 2022 (On-Premises)
 - June 18, 2022 (Cloud)

Note: The system requirements last changed with version 10.7.000000. See [that version's release notes](#) for details.

- [Secret Server 11.2.000002](#)
- [Secret Server 11.2.000000](#)
- [Secret Server 11.1.000012](#)
- [Secret Server 11.1.000007](#)
- [Secret Server 11.1.000006](#)
- [Secret Server 11.0.000008](#)
- [Secret Server 11.0.000007](#)
- [Secret Server 11.0.000006](#)
- [Secret Server 11.0.000005](#)
- [Secret Server 10.9.000064](#)
- [Secret Server 10.9.000033](#)
- [Secret Server 10.9.000032](#)
- [Secret Server 10.9.000002](#)
- [Secret Server 10.9.000000](#)
- [Secret Server 10.8.000004](#)
- [Secret Server 10.8.000000](#)
- [Secret Server 10.7.000059](#)
- [Secret Server 10.7.000002](#)
- [Secret Server 10.7.000000](#)
- [Secret Server 10.6.000027](#)
- [Secret Server 10.6.000026](#)
- [Secret Server 10.6.000001](#)
- [Secret Server 10.6.000000](#)
- [Secret Server 10.5.000003](#)
- [Secret Server 10.5.000001](#)
- [Secret Server 10.5.000000](#)
- [Secret Server 10.4.000000](#)
- [Secret Server 10.3.x](#)
- [Secret Server 10.2.x](#)
- [Secret Server 10.1.x](#)
- [Secret Server 10.0.x](#)
- [Secret Server 9.x](#)

- [Secret Server 8.x](#)
- [Secret Server 7.x](#)
- [Secret Server 6.x](#)
- [Secret Server 5.x](#)
- [Secret Server 4.x](#)

- [Cloud Release December 21, 2019](#)
- [Cloud Release September 21, 2019](#)
- [Cloud Releases prior to September 21, 2019](#)

[Documentation Changelog](#)