

IBM Security Secret Server
Version 10.6

Security Hardening Guide

Table of Contents

Introduction	2
Overview	3
Things to consider	3
General.....	3
Database	3
Application Server.....	4
Application Settings.....	5
Security Hardening Report.....	6
Configuration	6
Database	8
Environment.....	8
SSL.....	8
Two-factor Authentication	10
Email	10
Soft Tokens	10
RADIUS	10
Duo Security.....	10
Enabling Two-Factor Authentication	10
Roles	12
Controlling Access to Features Using Roles	12
Export.....	12
Unlimited Administration Mode.....	12
Secret Templates.....	12
Event Subscriptions	12
Role Separation	12
Unlimited Administration Mode.....	12
DPAPI Encryption	14
Enabling and Disabling DPAPI.....	14
DPAPI and Clustering	14
SSL / HSTS.....	15
GET Mutation	16
SSH Key Validation.....	17
Mapping SHA1 Digest TO Secrets	17
UNIX Account Discovery.....	17
More Resources.....	18

Last updated: April 30, 2019

Introduction

This document is going to outline some of the best practices for securing your Secret Server instance, whether it be installed on a single server or in a multi-clustered environment.

Throughout this guide, many references refer to “Configuration” settings. Unless otherwise specified, find these settings by selecting **Configuration** from the **ADMIN** menu in Secret Server.

Overview

It is critical to build a secure process around your Secret Server implementation. This needs to include a layered approach to security (defense in depth) starting at the operating system, software updates, access to physical systems, protocols, system settings, backups, and personnel procedures. This section of the guide link to other sections and KB articles containing more detail about certain features.

THINGS TO CONSIDER

General

- ✓ **Keep Windows up-to-date.** Microsoft regularly releases security patches that resolve vulnerabilities in Windows operating systems. We recommend keeping your server up-to-date.
- ✓ **Backup At Least Daily.** Consider your Disaster Recovery plan. Review [Business Continuity and Disaster Recovery Planning](#) (KB) for more information.
- ✓ **Review System Log for Errors.** It is important to periodically check the System Log for any recurring errors. After an upgrade, check for any errors in the System Log (ADMIN > System Log).
- ✓ **Whole Disk Encryption:** Enabling whole disk encryption, such as [BitLocker](#), with a Trusted Platform Module (TPM) can prevent someone with physical access from removing disks to get access to your Secret Server application.

Database

- ✓ **Limit access to your Secret Server database.** When you create your Secret Server database, you must limit access to as few users as possible. We recommend that 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 use by Secret Server, you must ensure that it only has access to the Secret Server database.
- ✓ **Use Windows Authentication for database access.** Windows Authentication is much more secure than SQL authentication. For a detailed explanation of why this is true, please refer to [Choose an Authentication Mode](#) (TechNet article). To use Windows Authentication in Secret Server, you need to create a service account. For details on how to do this, refer to [Using Windows Authentication to access SQL Server](#) (KB).
- ✓ **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 must still limit access to ensure that you are following “defense in depth.” Make sure to limit access to database backups to as few individuals as possible.
- ✓ **Don't store the database on a SQL instance that contains less sensitive databases.** Putting the database on a server with other less secure database instances can open up vulnerabilities. For example, an attacker might potentially use SQL injection on another application to access your private Secret Server database.

- ✓ **Review Microsoft recommendations for SQL security.** For general security best practices and recommendations by Microsoft, the [securing SQL Server resources](#) are a good starting point.

Application Server

- ✓ **Use SSL / HTTPS.** Secure Sockets Layer (SSL) is required to ensure that all communication between the web browser and Secret Server is encrypted and secure (and not cleartext travelling across your network). It is suggested that 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, see [Installing a Self-Signed SSL/HTTPS Certificate](#) (KB).
- ✓ **Force SSL / HTTPS.** Even after you install an SSL certificate, users might still be able to access Secret Server through HTTP. To prevent access through HTTP, enable the Force HTTPS/SSL option in Secret Server under ADMIN > Configuration > Security tab.
- ✓ **Limit access to your Secret Server directory.** It is important to limit access to your Secret Server directory. This contains the Secret Server encryption key and the database connection information (these values are encrypted but remember “defense in depth.” Try to grant access to as few users as possible).
- ✓ **Limit log on rights to the Application Server.** Administrators accessing the Application Server directly might 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 uses machine-specific encryption to encrypt the file. Make sure you back up the original file before you enable 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](#) (KB). 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](#).

+ Note

When setting up clustering, it is necessary to copy a version of your encryption.config file that is NOT encrypted to the additional server. Once Secret Server is up and running on that server, you can then DPAPI-encrypt the encryption.config file by logging in to Secret Server locally and turning on DPAPI.

Application Settings

- ✓ **Use DoubleLock for your most sensitive Secrets.** DoubleLock is a feature in Secret Server that allows Secrets to be protected with more AES256 encryption keys. Each user gets their own public/private key when you use DoubleLock. Their private key is protected by an extra password (user-specific, not a shared password) that each user must enter when you use 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 do not have access to DoubleLocked secrets. For more information, refer to [Using DoubleLock](#) (KB).
- ✓ **Secure the Local Admin Account.** When you create the first user in Secret Server, 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.** Event subscriptions can be used to send email alerts on various events in the system, and syslog can send events to a SIEM tool for correlation. This might be used to notify administrators if there are failed login attempts or if certain Secrets are viewed, and so on.

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 **REPORTS** from the top menu and then select the **Security Hardening** tab. You find the following items in the report:

CONFIGURATION

Allow Approval for Access from Email

Requires Approval for Access is a security setting that can be applied to Secrets to require users that must access the Secret to request approval first. The access request is emailed to those users who are listed as “approvers” for the Secret, and they then can approve access for a set period or to deny access. With this setting enabled, the approval emails that contain links the approvers can click to quickly approve or deny access. This setting can provide a more convenient way to handle approval requests. However it might also pose a security risk if someone other than the approver were to gain access to the approver’s email account.

To disable this setting, find the **Permission Options** section of the Configuration settings page and disable **Allow Approval For Access from Email**.

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, navigate to the Configuration settings and disable the **Allow AutoComplete** option on the **Login** tab.

Force Password Masking

Password Masking prevents over the shoulder viewing of your passwords by a casual observer (passwords show as *****). 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** in the Configuration settings. Only Secret fields marked as a password type field on the Secret template is masked. There is also a user preference setting that forces password masking on all Secret Password fields that are viewed by the user. This **Mask passwords when viewing Secrets** setting is found in the **Tools > Preferences** section for each user. If the Force Password Masking configuration setting discussed is enabled, this user preference setting is overridden and cannot be disabled.

Login Password Requirements

Passwords that are used by local users to log in to Secret Server 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 **Local User Passwords** tab of the Configuration page.

Maximum Login Failures

The maximum number of login failures is the number of attempts that can be made to log in to Secret Server as a particular user before that user's account is locked. A user with the Administer Users role permission is required to unlock the user's account. The maximum failures that are allowed must be set to 5 or less to get a pass result. Change the **Maximum Login Failures** setting on the **Login** tab of the Configuration settings.

Remember Me

Remember Me is a convenience option that allows users to remain logged in to Secret Server for up to a specific period. This setting can be a security concern, as it does not require reentry 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.

Closing a browser completely (all windows/tabs) will log the user out of Secret Server regardless of this setting.

Secure Session and Forms Auth Cookies

Cookies contain potentially sensitive information that can allow users to authenticate to an application. By default, cookies are not marked with the secure attribute. This means that cookies are transmitted in clear text 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](#) (KB).

Web Service Http Gets Allowed

Allowing HTTP GET requests allows REST-style calls to many Secret Server web service methods. This can be a security concern because clicking a link to the web service that is formed by a malicious user would cause it to be run. Disable **Allow Http Get** under the **Security** tab of the Configuration settings to get a pass result.

Zero Information Disclosure Error Message

Error messages can be helpful when you diagnose installation and configuration issues. However, having errors that are displayed to a potential attacker can provide them with the critical information they need to perform a successful attack. To hide specific error messages from the end user, add the ZeroInformationDisclosureMessage application setting to the web-appSettings.config file. This file is located in the Secret Server application files. Add the key below to this file in between the <appSettings> tags. The message that you put in that setting is displayed to the user when an error occurs in the system. For example:
`<add key="ZeroInformationDisclosureMessage" value="An error occurred in the Application, please contact your administrator." />`

DATABASE

SQL Server Authentication Password Strength and Username

SQL Server authentication requires a user name 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 user name must not be obvious. The use of "sa", "ss" or "secretserver" will give a fail result.

Changing the credentials of a local SQL account can be done through SQL Server Management Studio where the Secret Server database is located. For details about creating or modifying a SQL account for Secret Server, see the [Installation Guide](#). 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 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 page 19 of the [Installation Guide](#) for instructions on configuring Windows authentication to SQL Server.

ENVIRONMENT

Application Pool Identity

Check the identity of the application pool that is used by Secret Server in IIS. The Application Pool must be configured to use a service account and not given unrestricted access to the server or domain.

SSL

Require SSL

As best practice, it is suggested to set up SSL (or https) for access to Secret Server. Secure Sockets Layer (SSL) is required to ensure that all communication between the web browser and Secret Server is encrypted and secure. To do so, you need an SSL certificate. You might use an existing wildcard certificate, create your own domain certificate or purchase a third-party SSL certificate for the Secret Server website. Another option that is not as practical for a production environment but can be used for testing is to use a self-signed certificate. See [Installing a Self-Signed SSL/HTTPS Certificate](#) (KB) for more information.

After 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 result.

 *Use of SSL is highly suggested for Secret Server.*

SSL/TLS Hash

This checks the digest algorithm of the certificate. If the algorithm is SHA1, this check returns a warning since use of SHA1 is being phased out. If the digest algorithm is MD2, MD4, or MD5, the check fails because they are not secure. SHA256, SHA384, and SHA512 passes. This check fails if Secret Server cannot load over HTTPS.

+ **Example warning**

The digest algorithm is sha1RSA, which is considered weak. The use of this algorithm is being phased out and must be replaced with a better algorithm when it comes time to renew the SSL certificate.

SSL/TLS Key

Checks the key size of the HTTPS certificate that being used. If the signature algorithm of the certificate 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 will fail if Secret Server cannot be loaded over HTTPS.

SSL/TLS Protocols

This checks for legacy SSL/TLS protocols that must not be used in a secure environment. If the server accepts SSLv2 or SSLv3 connections, this check fails. 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 gives a warning since they are recommended. The SSL certificate that is used might affect what protocols can be used, even if they are enabled. This check will fail if Secret Server cannot be loaded over HTTPS.

+ **Example failure message**

The server supports the protocols SSL 3, which are weak. Consider disabling these protocols.

Using HTTP Strict Transport Security

An extra layer of security for HTTPS / SSL is Strict Transport Security (HSTS). HSTS allows Secret Server, Password Reset Server, or Group Management Server to inform browsers that it must 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 is displayed. After the option is turned on, you can click **Advanced** to specify the Maximum Age in seconds for how long the policy must be in affect before reevaluating. The default value is 25200 seconds, or 7 hours. It is suggested that you set this value as high as possible, up to a year, if the site, must never be accessed without SSL. For more details, see [Securing with HTTP Strict Transport Security \(HSTS\)](#) (KB).

Two-factor Authentication

Users must authenticate to Secret Server at least once by using either local Secret Server credentials or their Active Directory credentials. However, when a password gets compromised, you can protect yourself by enabling two factor authentication (TFA) in Secret Server. When you use multiple factors of authentication, each factor must be a different type of information – that is, either a piece of information a user **knows**, **possesses**, or **is** (for example, when a fingerprint is used as a biometric identifier). The following options are supported by Secret Server for TFA:

EMAIL

After authenticating with their password, the user receives an email that contains 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 that is associated with their account. For Active Directory users, the email address is synced automatically from their domain account. User email addresses can be checked by selecting **Users** from the **ADMIN** menu.

SOFT TOKENS

Soft tokens by using the TOTP algorithm, such as Google Authenticator and Microsoft Authenticator, are supported for use with Secret Server TFA. Users are prompted to enter a token that is displayed on their mobile device each time they log in to Secret Server. The time-based token changes on a regular interval (such as 30 seconds).

RADIUS

One option is to use a RADIUS-compliant device, such as an RSA or CryptoCard token, as an extra form of authentication. The user will be prompted to enter their RADIUS password after initial authentication is done with their Secret Server or AD password. To set this up, you will first need to configure Secret Server to integrate with your RADIUS server. You can then enable it for individual users or for by domain. See [Enabling RADIUS Two-Factor Authentication](#) for configuration details.

DUO SECURITY

Using this method requires that you have an active account for Duo Security. Duo Security provides several options for two-factor authentication. The API Hostname, Integration Key, and Secret Key values are required for Secret Server to authenticate the users that where this two-factor method is enabled. See [Configuring DUO for Two Factor](#) for configuration details.

ENABLING TWO-FACTOR AUTHENTICATION

To enable two-factor authentication for a user or several users immediately, 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. If prerequisite settings are not yet configured, the two-factor option might be greyed

out or appear as an option. See the descriptions above for information about prerequisites for each type of two-factor authentication.

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 check box 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 no other administrative permissions otherwise.

CONTROLLING ACCESS TO FEATURES USING ROLES

Export

Exporting Secrets from your Secret Server as cleartext is 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 must be tightly controlled. You might create a separate Role with just that permission for anyone that needs to perform exports.

Unlimited Administration Mode

Unlimited Administration Mode allows anyone with the Unlimited Administrator role permission to see all Secrets in the Secret Server. This mode is 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 lets you create the “two key effect” for access to use this feature. See [Roles Separation](#) section for more information on ways to lock down this important feature.

Secret Templates

Anyone with access to modify your Secret templates can change the definitions of the data that is being stored, and this access must be tightly controlled. Your Secret templates are unlikely to need changing after you define them, so limiting access to a select number of individuals is typically sufficient.

Event Subscriptions

Another option when you protect roles is to configure Event Subscriptions to notify appropriate staff when 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 Unlimited Administrator to further protect its use.

ROLE SEPARATION

Unlimited Administration Mode

It is suggested that you determine which role permissions must or must not be combined for users before you assign roles and allow users access to the application. As best practice, a strategy must be planned for the use

of Secret Server's Unlimited Administration (break-the-glass) mode. Unlimited Administration mode only can be enabled by a user with the **Administer Configuration Unlimited Admin** role permission, but when enabled, allows users with the **Unlimited Administrator** role permission to view all Secrets in Secret Server and access all configuration settings. This means that:

- ✓ A user with both the **Administer Configuration Unlimited Admin** and **Unlimited Administrator** role permissions lets you enable Unlimited Administration mode and then can also view all Secrets in the system or make any configuration changes.
- ✓ The role permissions can be split into two different roles to enforce accountability and require the cooperation of two individuals to use Unlimited Administration mode.

One solution for this is to create two roles – each containing one of the Unlimited Admin role permissions – and take those permissions out of the Administrator role. You can then assign those roles to users that must be responsible for either enabling/disabling Unlimited Administration mode or retrieving Secrets while Unlimited Administration mode is enabled, but not both. When a user needs a Secret retrieved, one person must enable the mode. Another person can retrieve the Secrets or make any changes necessary before you disable Unlimited Administration mode once more.

Some additional checks in place to ensure that use of Unlimited Administration mode can be monitored include:

- ✓ Enabling/disabling Unlimited Administration mode is audited, and a comment can be provided each time that it is enabled.
- ✓ When Unlimited Administration mode is enabled, a banner is displayed at the top of every Secret Server page that is notifying users that their Secrets can currently be viewed by an Unlimited Administrator.
- ✓ Event subscription notifications contain the option 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 also audited while Unlimited Administration mode is enabled.

DPAPI Encryption

DPAPI is an option that can provide an extra 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/decrypt the sensitive data that is stored in the Secret Server database. Using the DPAPI option in Secret Server means that the encryption key file is encrypted with a key that only Windows knows and is only usable on that same server it was encrypted on. If someone attempted to configure Secret Server on another server by using that DPAPI-encrypted key, the application is not able to use it, protecting against theft of the data stored in the Secret Server database.

It's important to note that a backup of the encryption key file must be taken and stored in a secure location before you turn on DPAPI encryption. This will let you restore a backup of the application on another server in a DR scenario. The encryption.config key file is located in the Secret Server application file 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 you click **Confirm**. To decrypt the key, navigate to the same tab and click **Decrypt Key to not Use DPAPI**.

DPAPI AND CLUSTERING

DPAPI can be used while clustering is enabled for Secret Server, however you must consider a few things:

- ✓ A backup of the encryption key must be made before you use this option, otherwise disaster recovery might prove impossible if the server fails.
- ✓ The backup of this un-DPAPI-encrypted key needs to be transferred initially to each front-end Secret Server node.
- ✓ DPAPI must be enabled when you access each server locally (browse to Secret Server while on the server it is installed on, and then enable DPAPI encryption).

For more information about clustering Secret Server, see [Setting up Clustering](#) (KB).

SSL / HSTS

SSL is a basic best practice that is suggested for Secret Server. Taking SSL a step further, Secret Server also supports HTTP Strict Transport Security, or HSTS. Strict Transport Security is supported by modern browsers and tells the browser that a site is only ever accessible by SSL with a valid certificate. Even if a man-in-the-middle attack occurs with a trusted, but different, SSL certificate, the browser rejects the SSL certificate. So, this setting is useful for protecting against forged SSL certificates or man-in-the-middle attacks.

For more information about configuring SSL certificates, see [Creating 2048-bit Domain SSL Certificate](#) (KB) and the [Installation Guide](#). Additional information about HSTS can be found in [Securing with HTTP Strict Transport Security \(HSTS\)](#) (KB).

GET Mutation

Another security option is to disable GET requests for the Secret Server web services. A GET request is a simple means of accessing a web server by API. However, browsers also use GET requests when you load pages. This allows an attacker to potentially send someone a misnamed link to the web service in an email that does a GET request. If you aren't using web services or GET requests, this option must be disabled as best practice.

To disable GET requests, disable **Allow Http Get** on the **Security** tab of the Configuration page.

SSH Key Validation

Host SSH Key verification is supported as of Secret Server 8.8.000004 for use with Heartbeat, proxied Launchers, password changers, and Discovery. Host SSH key verification can be used to help ensure that the computer you are connecting to is a trusted host. Host SSH key verification does not pass credentials to the target computer unless the public key digest matches the SHA1 digest that Secret Server has on file. This helps prevent man-in-the-middle attacks.

MAPPING SHA1 DIGEST TO SECRETS

To configure host SSH key verification, navigate to **Secret Templates** from the **ADMIN** and add a field for the host's SSH key digest. Next, click "Configure Extended Mappings" and add a "Server SSH Key" mapping to your newly created SSH key digest field. On your Secrets, add the SSH key digest of the hosts to your digest field; verification will take effect the next time you connect to the host.

If the mapped Digest field is blank, the digest is not checked for Heartbeat, password change, and Launchers. If it is available, it is checked.

UNIX ACCOUNT DISCOVERY

To validate SHA1 server digests for UNIX Account Discovery, create a file that is named KeyDigests.txt in the root of the Secret Server website. Each line must 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 computers to be scanned must match one of the SHA1 hashes in the file. Any computers that do not match still shows up on the Discovery Network View page. However, authenticated scanning does not take place. For example, no credentials are passed to the computer, and accounts are not retrieved from the computer.

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
```

More Resources

- [Knowledge Base](#)
- [Secret Server Best Practices Guide](#)