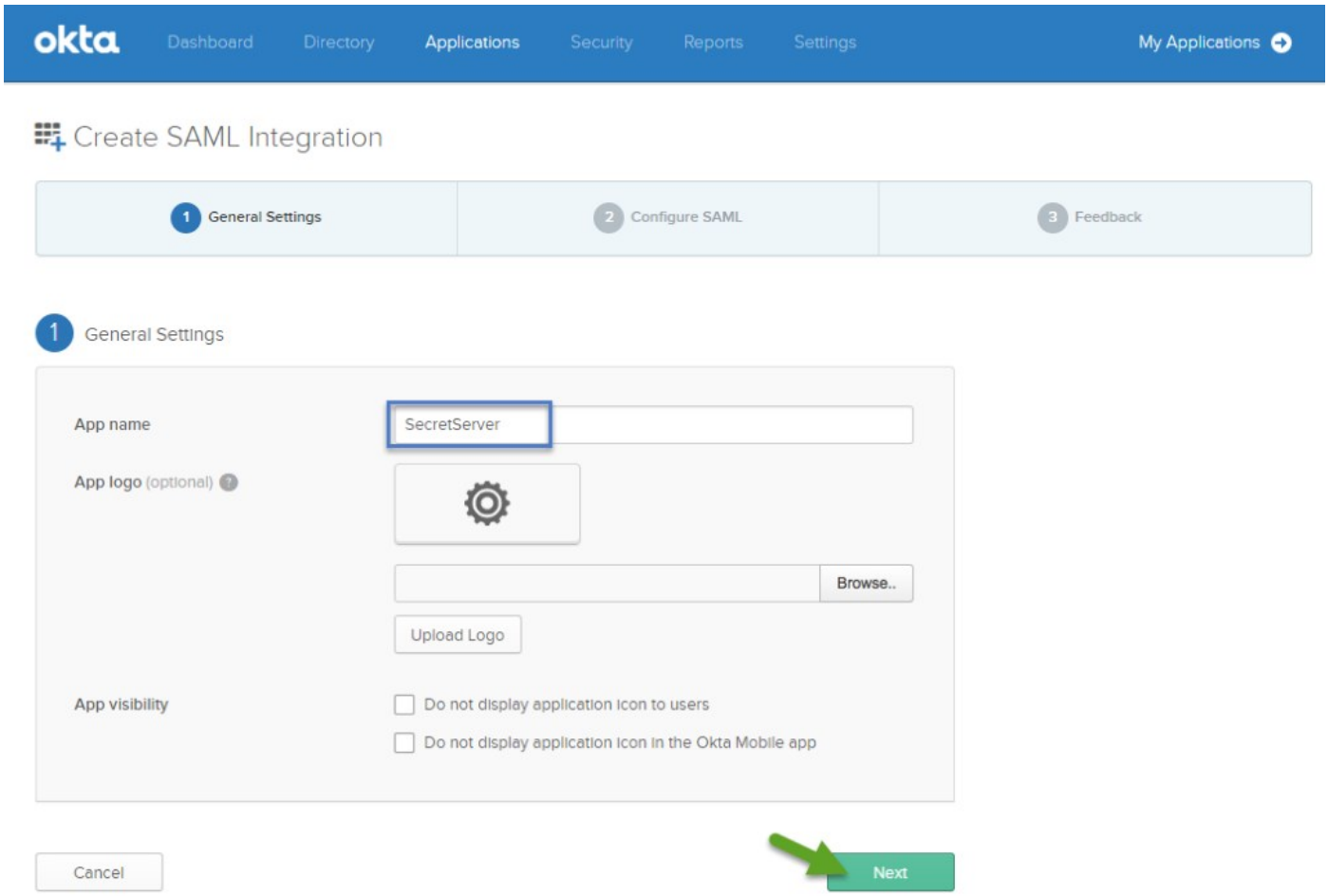
4. On the **General Settings** page, in the **App name** field, enter the **preferred name** (e.g. "SecretServer") and click **Next**. (The App name field is the only required input in this screen.)



Continue with Configure SAML.

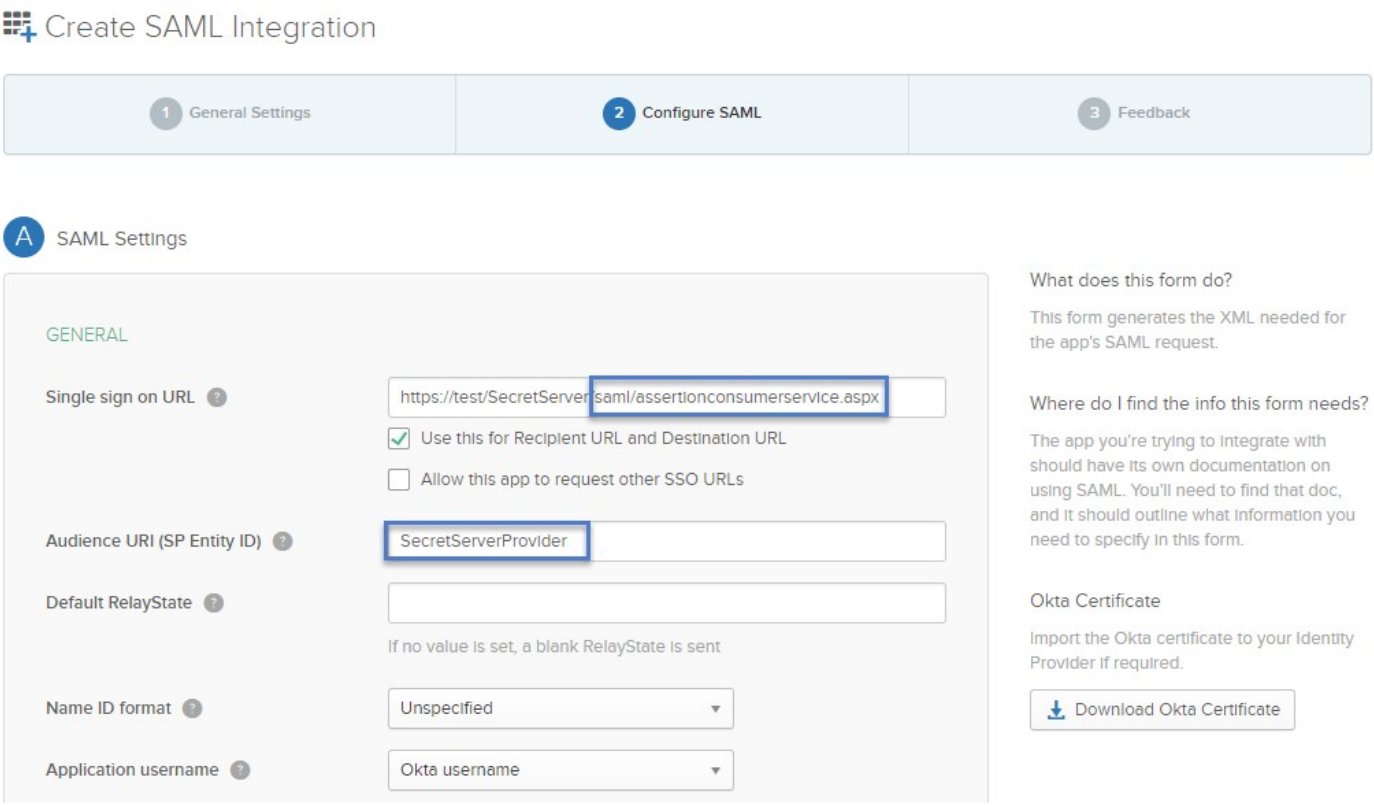
Configuration of Okta for SAML

When configuring Okta for SAML to integrate with Secret Server, the following general steps need to happen:

1. Setting up Secret Server as a new application in Okta.
2. Adding Okta details to the Secret Server SAML settings.
3. Configuring specifics like:
    1. Setting up Single Logout
    2. Adding Users
4. Verifying the integration works.
5. Addressing any advanced settings.

Configure SAML in Okta Application

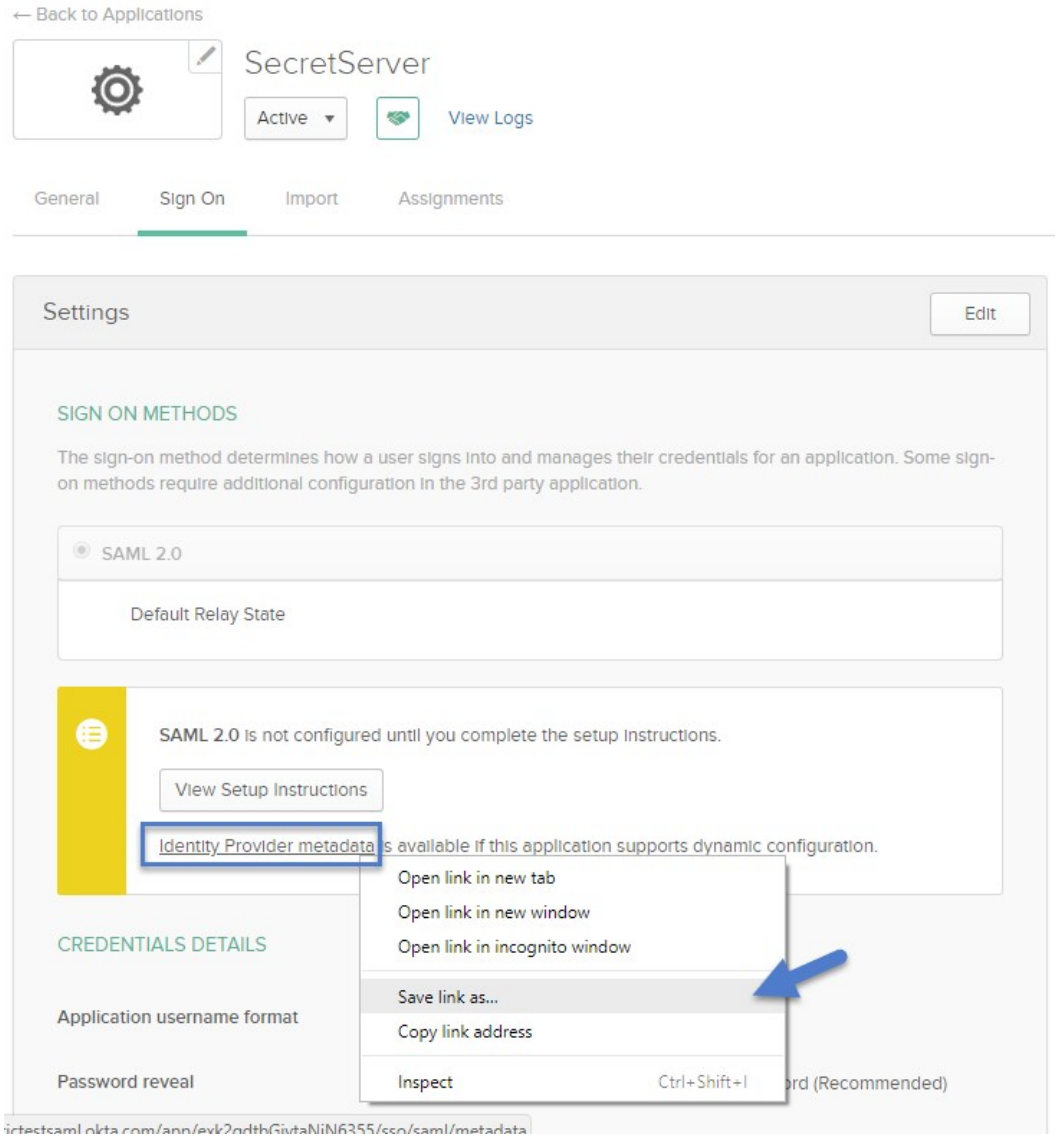
1. On the **Configure SAML** screen, enter the **required information** according to your organizational environment. Required descriptions and examples are:



   1. **Single sign on URL**
      - Enter the **URL** for your Secret Server instance into the following URL:

        https://[YourSecretServerInstance.com]/samlemetadata

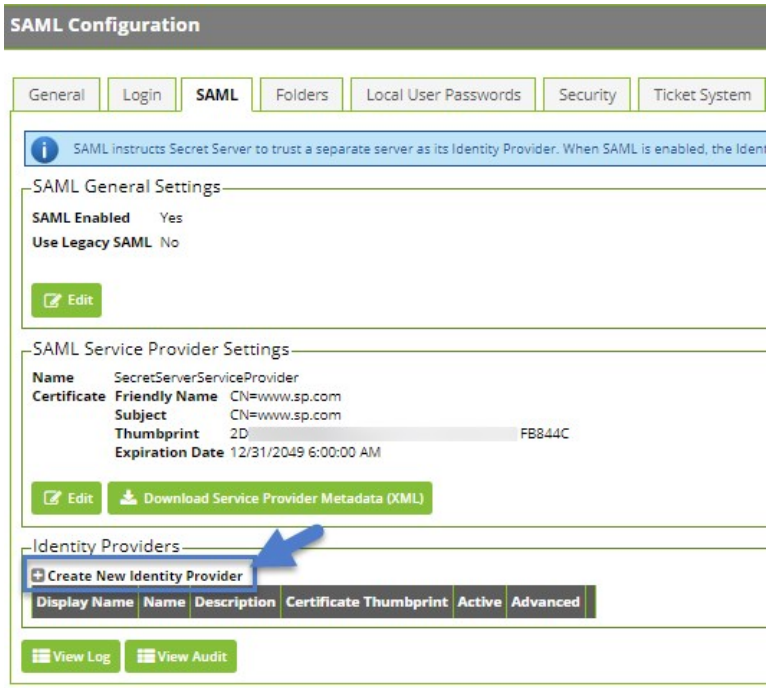
      - Leave the **Use this for Recipient URL and Destination URL** checked
      - Leave the **Allow this app to request other SSO URLs** unchecked if Okta is the only IdP
   2. __Audience URI (SP Entity ID) __
      - Enter the **Service Provider** configured in Secret Server: e.g. "SecretServerServiceProvider"
      - **Default RelayState** can be left blank
      - **Name ID format** can be left unspecified
      - **Application username** can remain "Okta username"
   3. Click **Next**.
2. On the Feedback screen, select__ I'm an Okta customer adding an internal app__.
3. On the Settings' screen, right-click **Identity Provider metadata** and click **Save Link As...** to save the metadata and import into Secret Server.



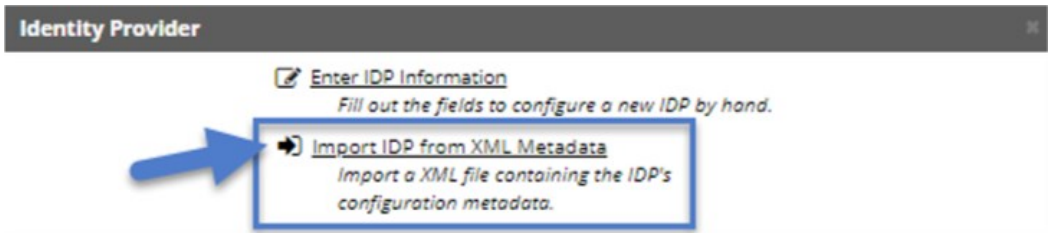
Continue with Configure SAML in Secret Server

Configure SAML in Secret Server

1. In your Secret Server Instance, navigate to the SAML configuration page **Admin | Configuration** and select the **SAML** tab.
2. Click **Create New Identity Provider**.



3. Click **Import IDP from XML Metadata** and select the **Okta metadata** previously saved. (If you do not see the file, you may need to change the metadata filetype to .xml.)

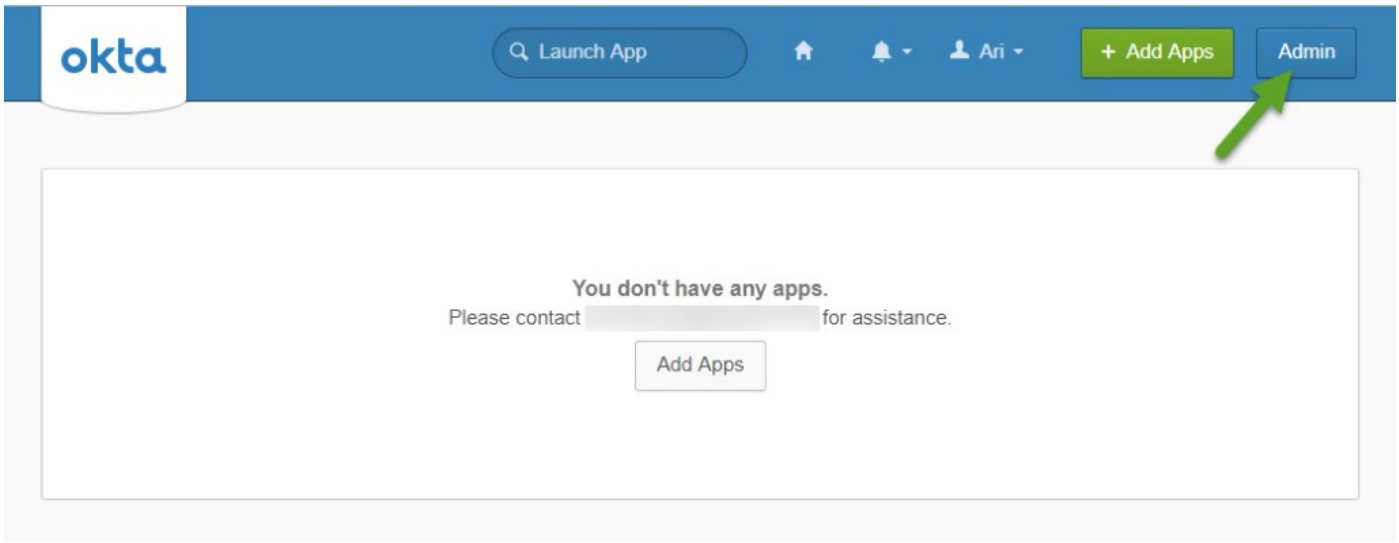


Continue with Configure Single Logout.

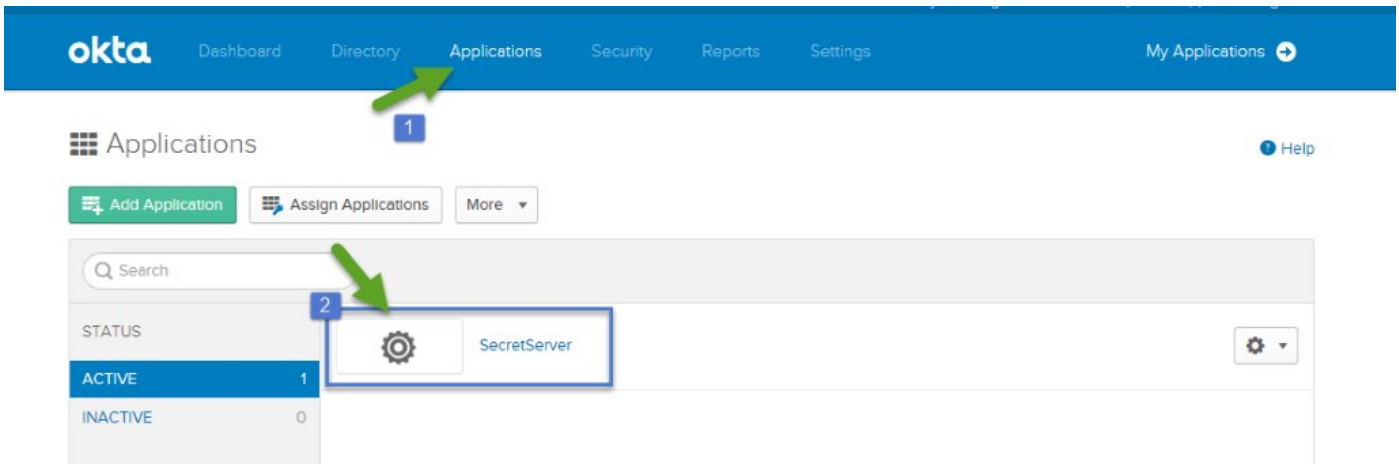
Configure Single Logout

NOTE: Due to browser issues, use the Firefox or Internet Explorer (IE) browsers to complete this section. Do *NOT* use Chrome.

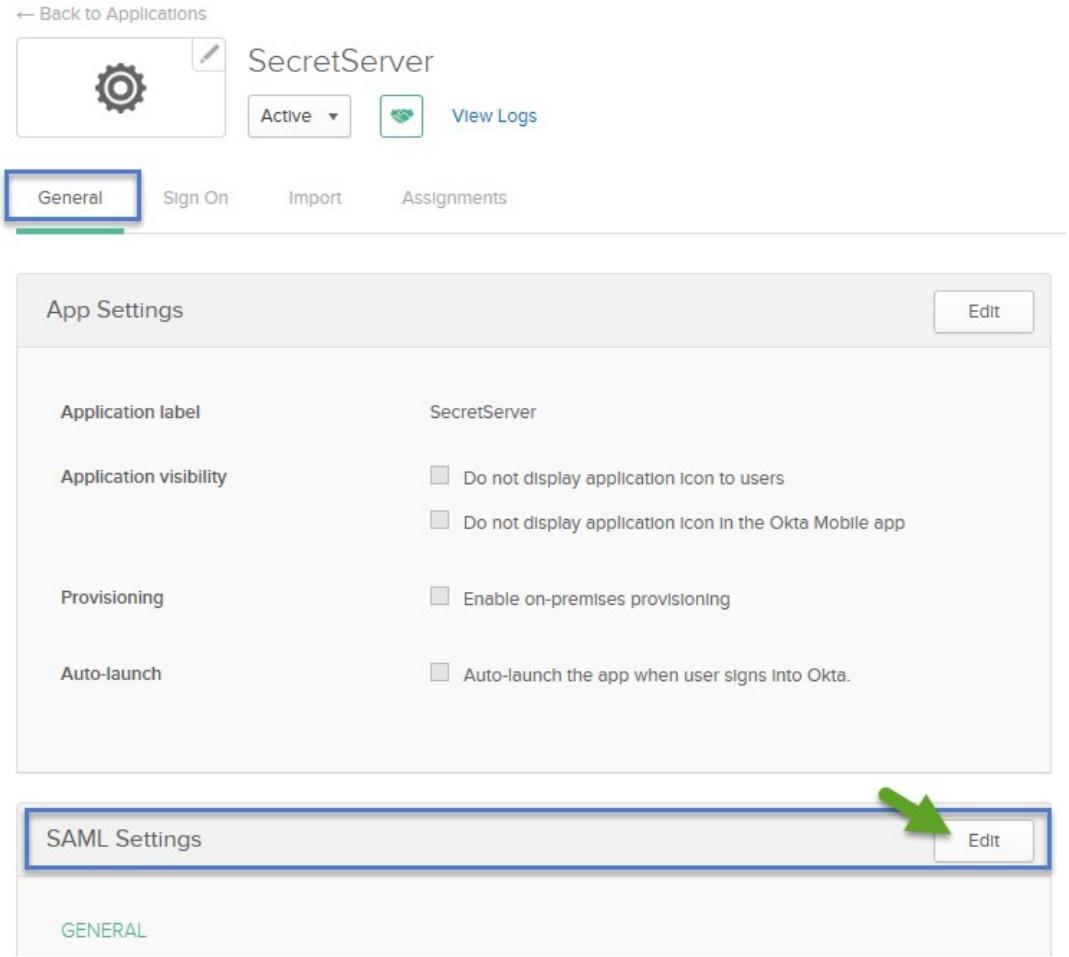
1. Login to Okta and navigate to the **Applications** homepage, then **Admin** page.



2. Click **Applications**, then **Secret Server Service Provider**.

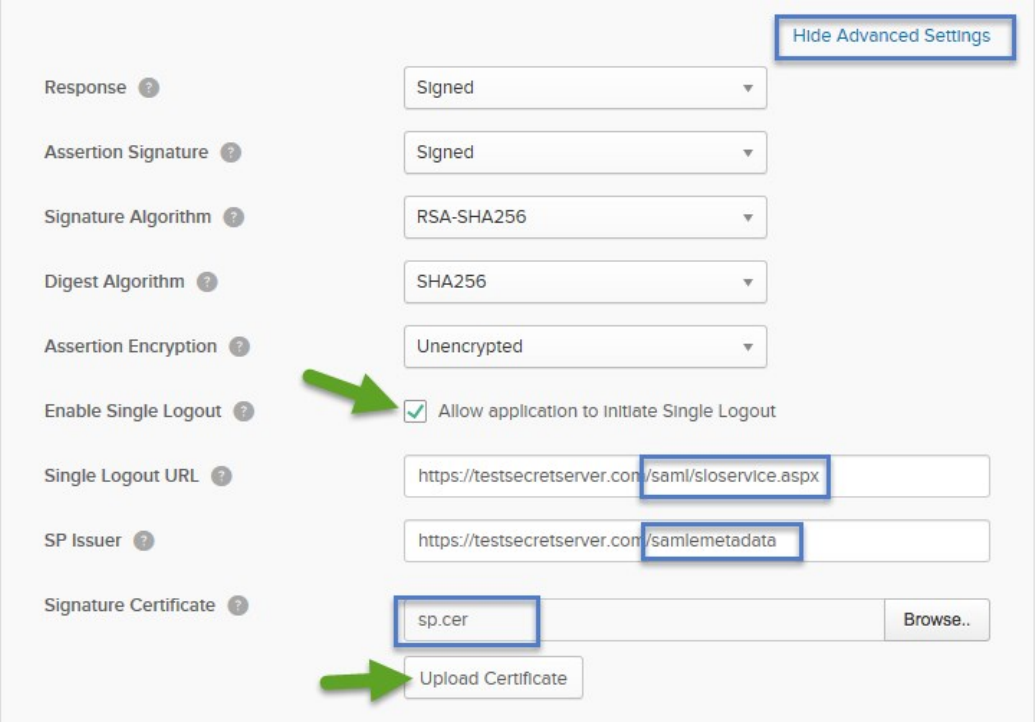


3. Under the **General** tab, click **Edit** next to SAML Settings.



4. On the General Settings screen, click **Next**.
5. On the Configure SAML screen, click **Show Advanced Settings**, then check the **Enable Single Logout** checkbox.
    1. Single Logout (SLO) Settings:
        - **Single Logout URL**: your [SecretServerInstanceName] followed by the URL string: /saml/sloservice.aspx (e.g. https://[YourSecretServerInstance.com]/saml/sloservice.aspx)
        - **SP Issuer**: link to your Secret Server's SAML Metadata, [InstanceName]/samlemetadata (e.g. https://[YourSecretServerInstance.com]/samlmetadata)
        - **Signature Certificate**: the public key (.cer) file that corresponds to the .pfx uploaded into your Secret Server Service Provider
            - Click **Browse** and select your certificate's .cer file
            - Click **Upload Certificate**

    NOTE: The Certificate File Name should change if the upload is successful. If the upload is unsuccessful, ensure you are using the FIREFOX or IE browsers, and not Chrome.

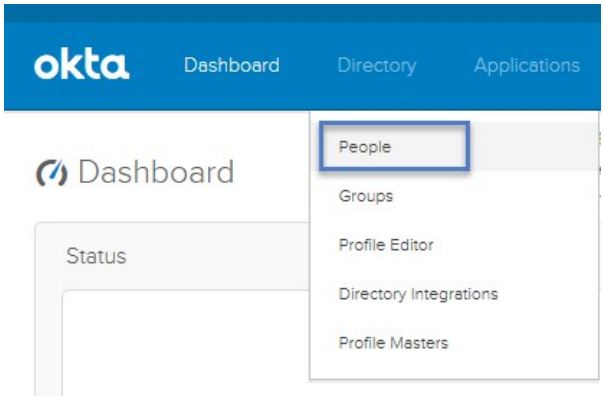


6. Click **Next**, then **Finish**.

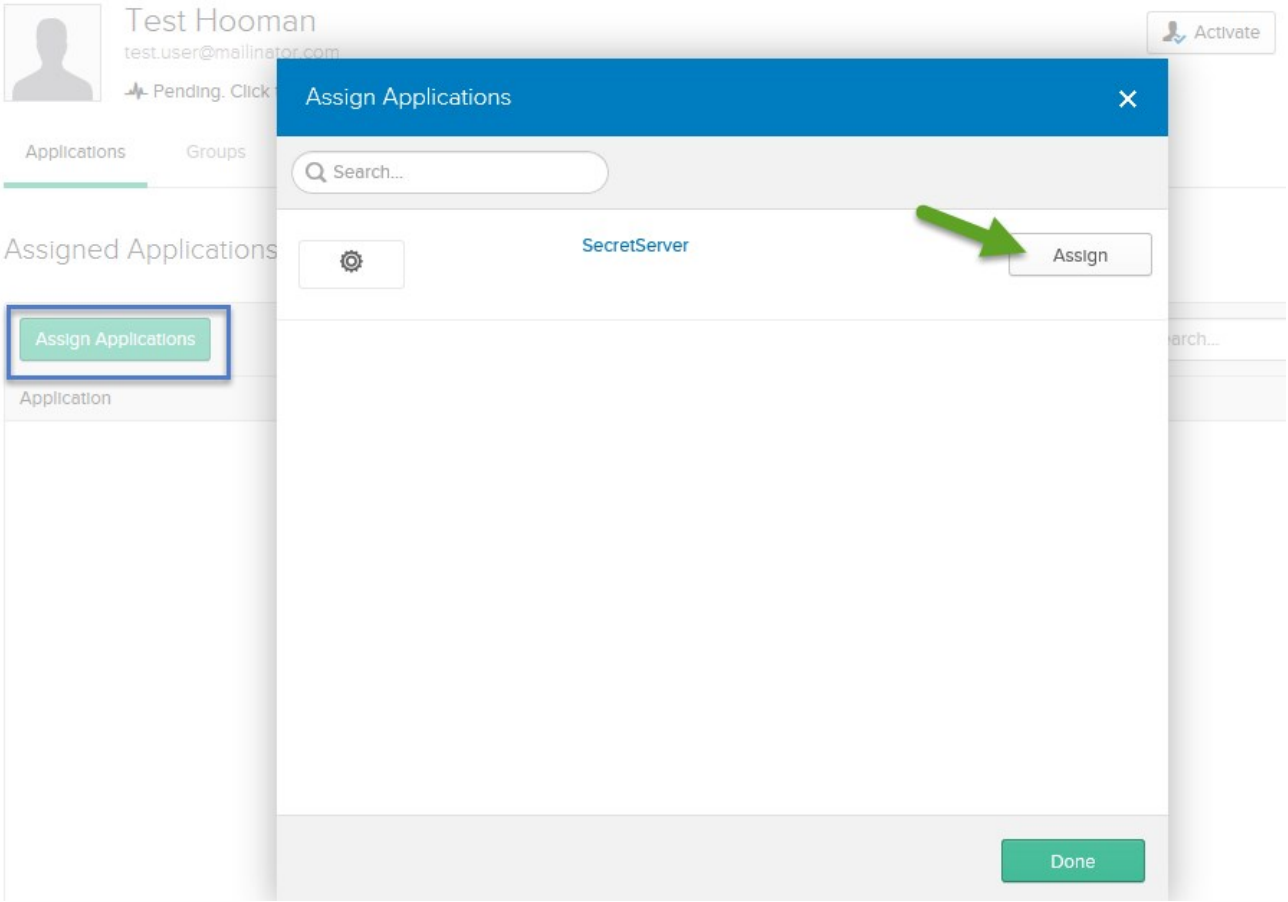
Continue with Add Users to Okta.

Add Users to Okta

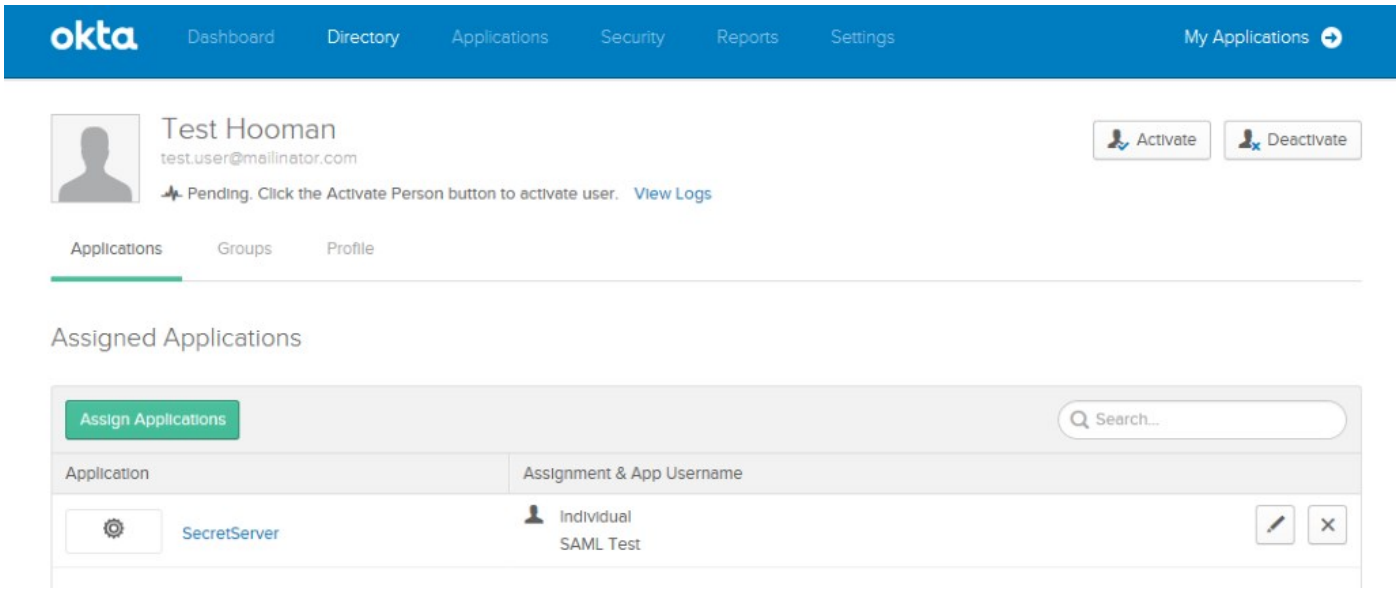
1. From Okta's Application Home page (InstanceName/app/UserHome) click **Admin**.
2. Hover over the **Directory** tab and click **People**.



3. Click **Add Person**.
4. Fill out the information as required and once created, click the **username**.
5. Click **Assign Applications**, then click **Assign** next to the Secret Server Service Provider.



6. Enter the **Username** of the corresponding user from Secret Server.



Click **Save and Go Back** and click **Done**. The Secret Server username should now be listed under Assigned Applications.

This user should now be able to use the SAML workflow.

Continue with Verifying the Integration.

Verifying the Integration

To test this, login to Okta as the user, then browse to your Secret Server instance. The user should be logged in automatically to Secret Server without prompting for login credentials.

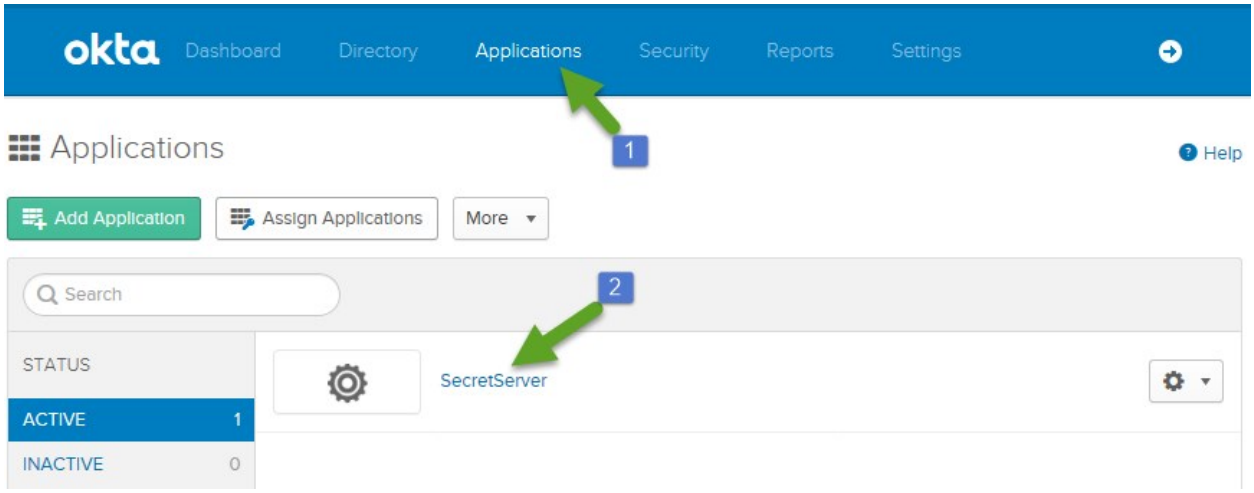
If SLO is configured, logging out of Secret Server as the user will also log them out of Okta.

Also refer to Advanced Settings for further configuration options.

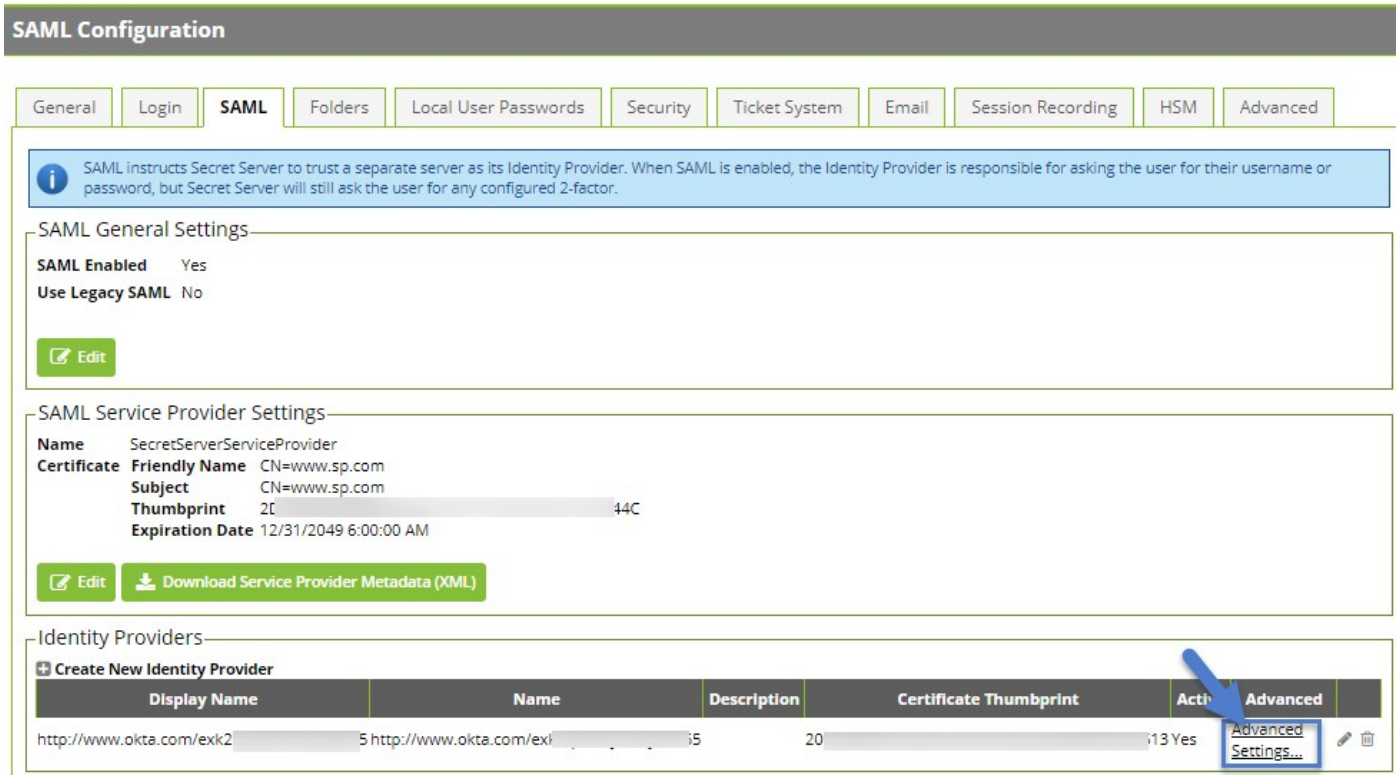
Advanced Settings

The following Secret Server Identity Provider Advanced Settings can be configured in Okta.

1. From the Okta Admin Dashboard, navigate to **Applications** then click on your **Secret Server**.
2. Click **Next**.
3. On the Configure SAML page click **Show Advanced Settings**.



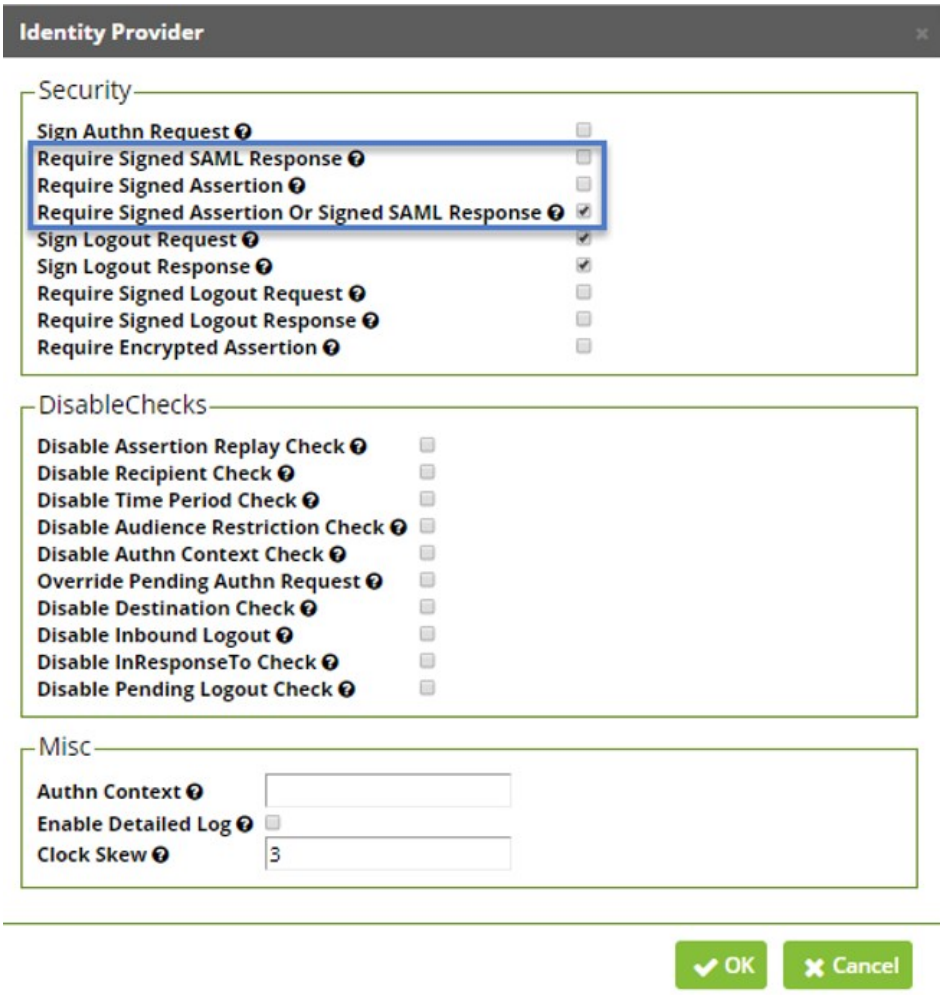
Note: These settings correspond to the following Advanced Settings in Secret Server: **Admin I Configuration** on the **SAML** tab under **Advanced Settings** next to your OKTA Identity Provider.

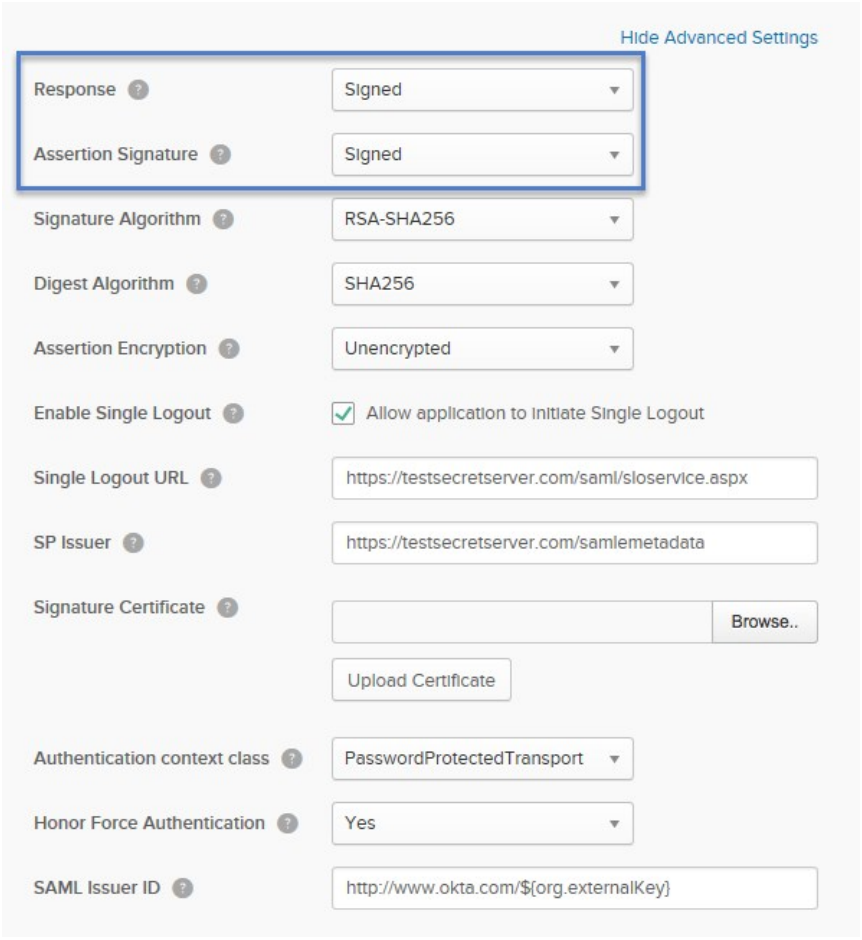


Below are screenshots for the corresponding settings between Secret Server and OKTA.

SAML Response

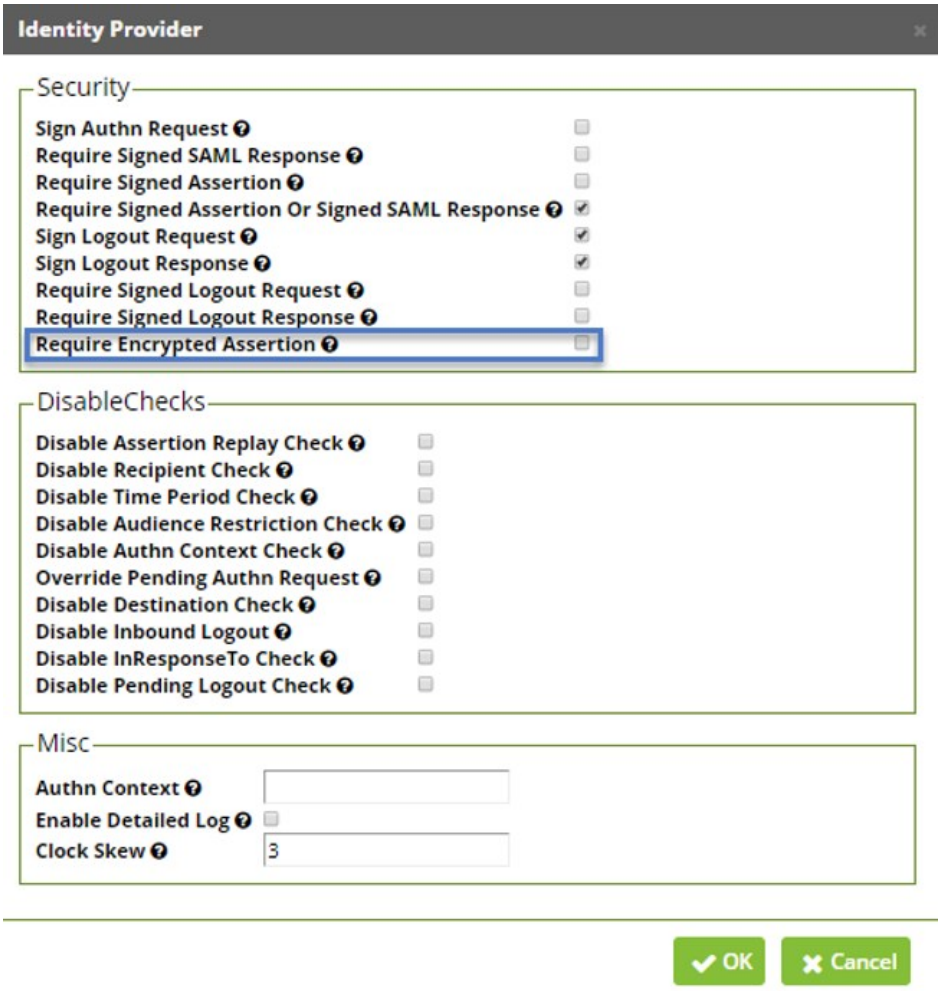
Require Signed SAML Response

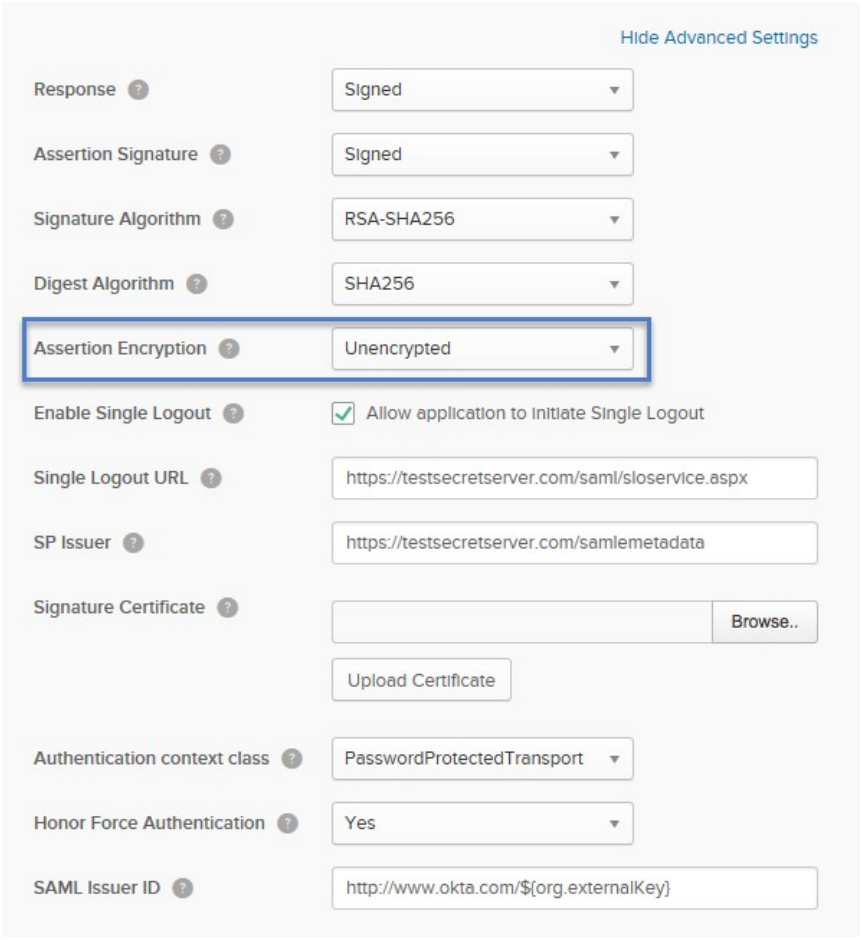




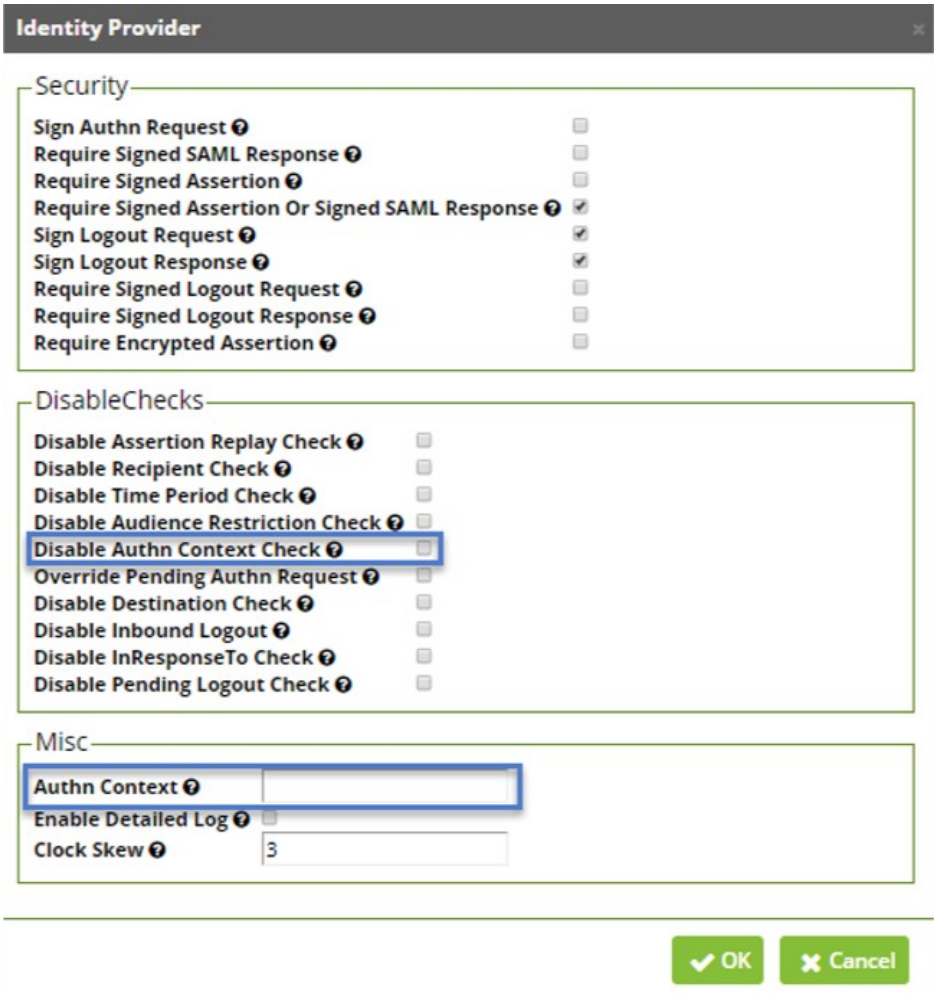
Require Encrypted Assertion

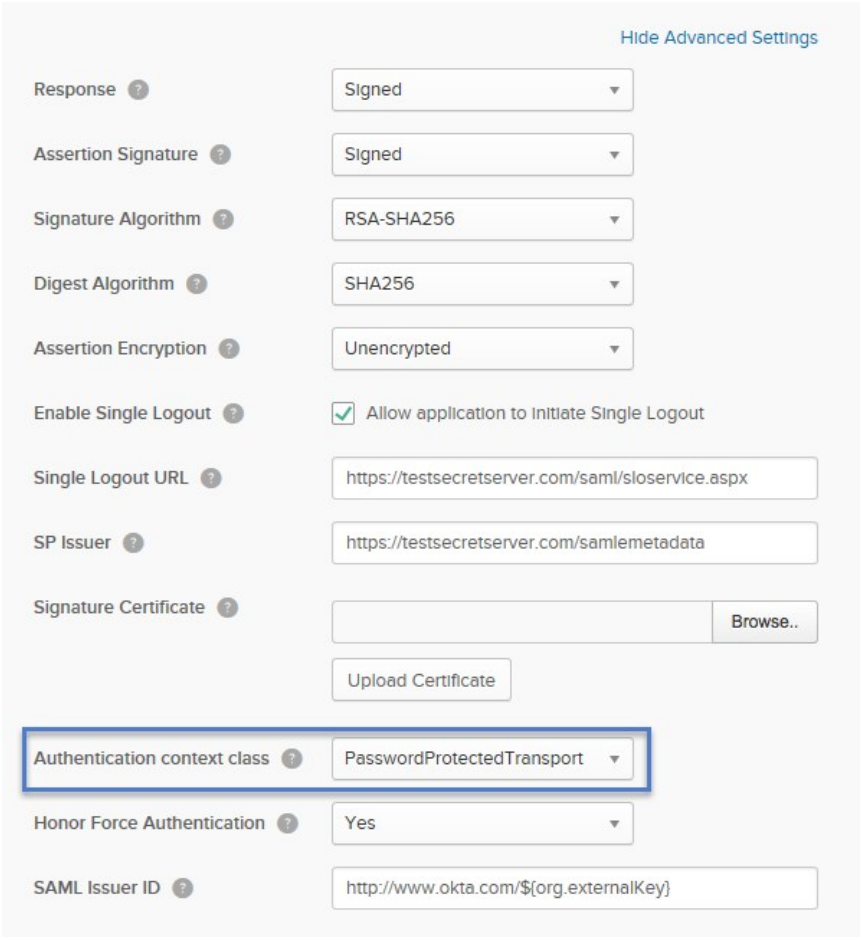
Require Signed Assertion


Identity Provider
Security
Sign Authn Request
Require Signed SAML Response
Require Signed Assertion
Require Signed Assertion Or Signed SAML Response
Sign Logout Request
Sign Logout Response
Require Signed Logout Request
Require Signed Logout Response
Require Encrypted Assertion
DisableChecks
Disable Assertion Replay Check
Disable Recipient Check
Disable Time Period Check
Disable Audience Restriction Check
Disable Authn Context Check
Override Pending Authn Request
Disable Destination Check
Disable Inbound Logout
Disable InResponseTo Check
Disable Pending Logout Check
Misc
Authn Context
Enable Detailed Log
Clock Skew
3
OK
Cancel



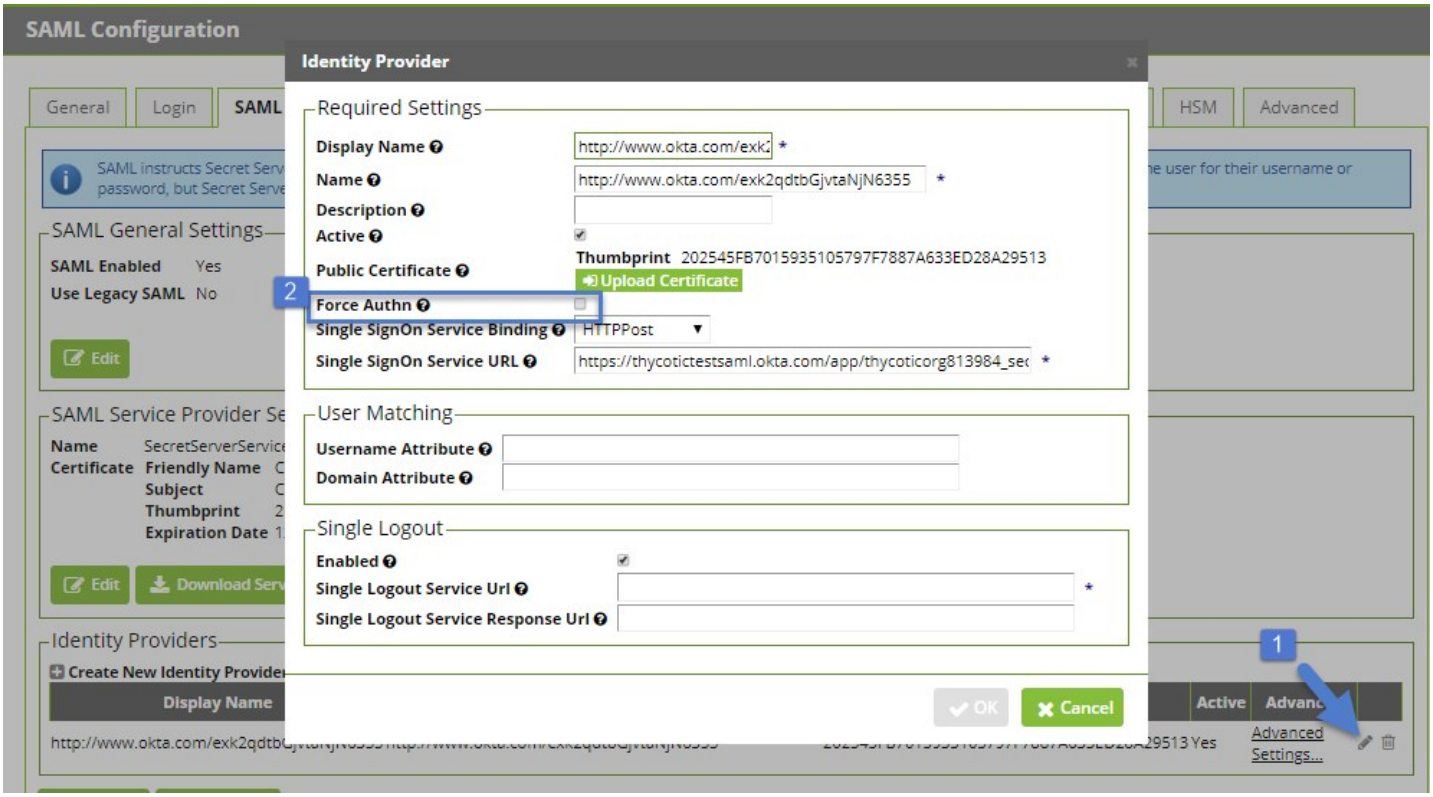
Authn Context / Disable Authn Context Check

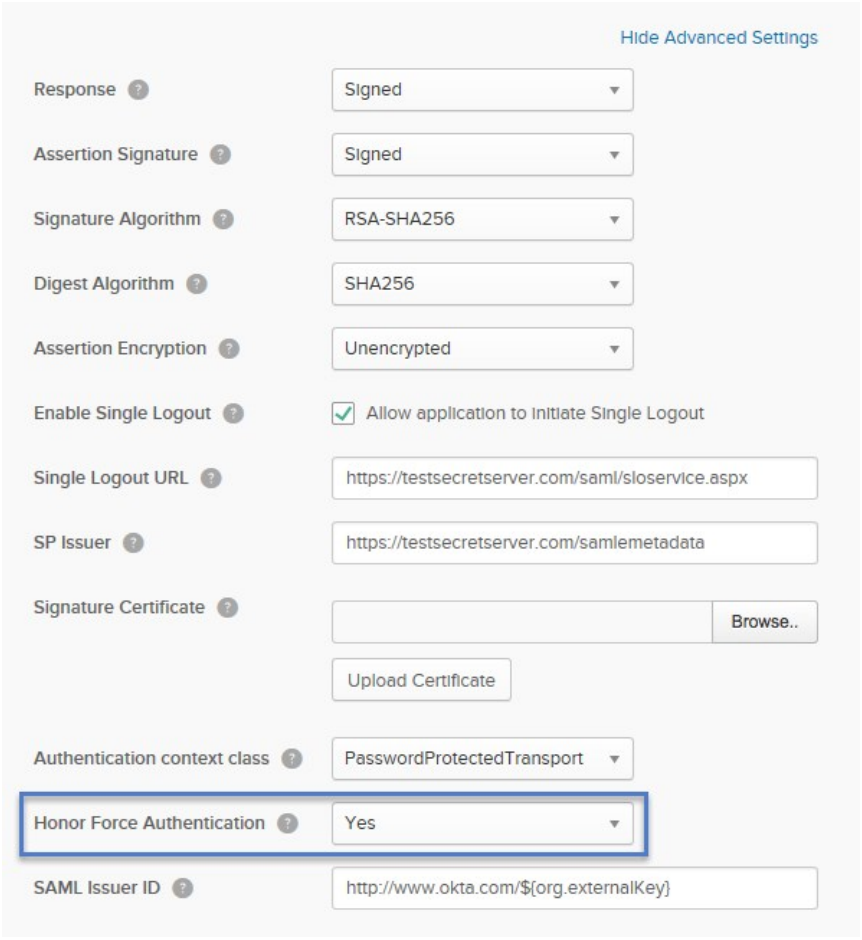

Identity Provider
Security
Sign Authn Request
Require Signed SAML Response
Require Signed Assertion
Require Signed Assertion Or Signed SAML Response
Sign Logout Request
Sign Logout Response
Require Signed Logout Request
Require Signed Logout Response
Require Encrypted Assertion
DisableChecks
Disable Assertion Replay Check
Disable Recipient Check
Disable Time Period Check
Disable Audience Restriction Check
Disable Authn Context Check
Override Pending Authn Request
Disable Destination Check
Disable Inbound Logout
Disable InResponseTo Check
Disable Pending Logout Check
Misc
Authn Context
Enable Detailed Log
Clock Skew
3
OK
Cancel



Force Authn

Require Signed Assertion Or Signed

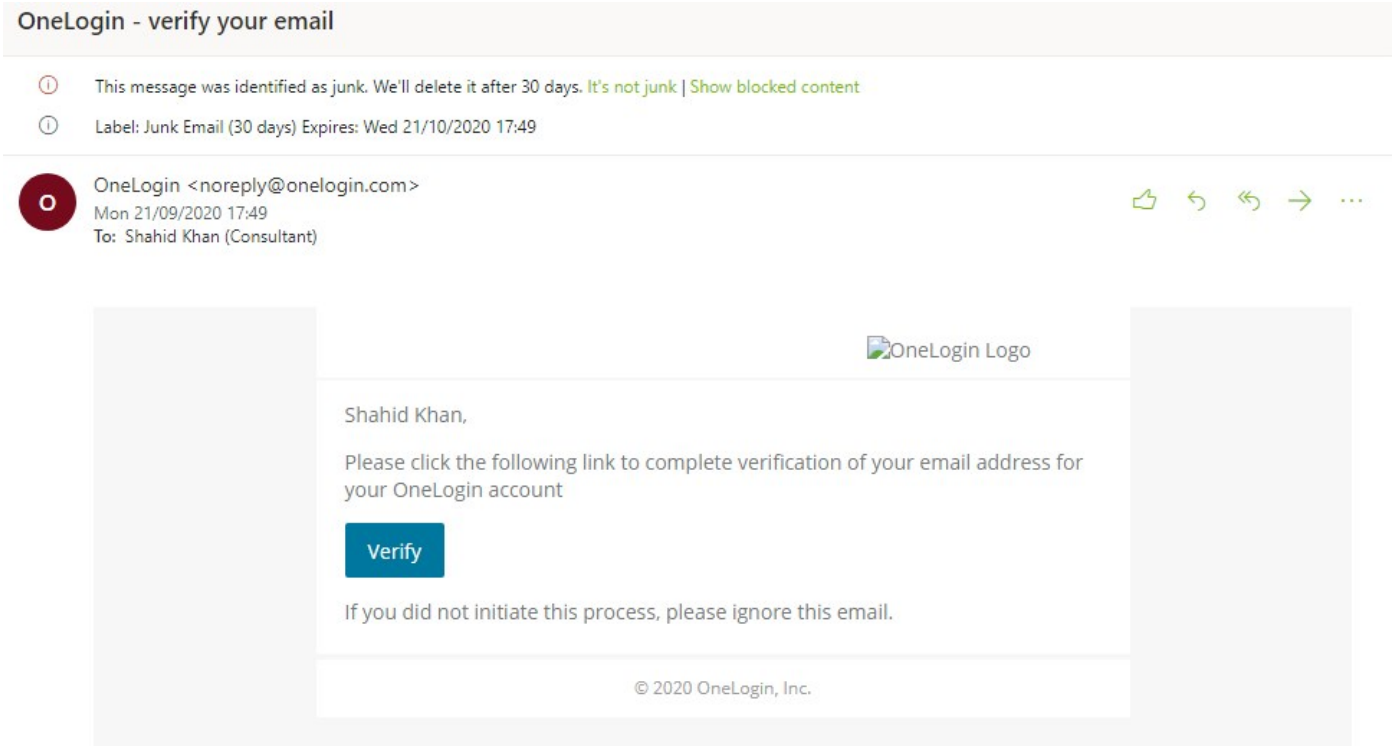

SAML Configuration
Identity Provider
General
Login
SAML
HSM
Advanced
SAML General Settings
SAML Enabled Yes
Use Legacy SAML No
Edit
SAML Service Provider Se
Name SecretServerServic
Certificate Friendly Name
Subject
Thumbprint
Expiration Date
Edit
Download Serv
Identity Providers
Create New Identity Provide
Display Name
Active
Advanc
http://www.okta.com/exk2qdtbGjvtaNjN6355
Yes
Advanced Settings...
Required Settings
Display Name
http://www.okta.com/exk2
Name
http://www.okta.com/exk2qdtbGjvtaNjN6355
Description
Active
Public Certificate
Thumbprint 202545FB7015935105797F7887A633ED28A29513
Upload Certificate
Force Authn
Single SignOn Service Binding
HTTPPost
Single SignOn Service URL
https://thycoticctestsaml.okta.com/app/thycoticorg813984_sec
User Matching
Username Attribute
Domain Attribute
Single Logout
Enabled
Single Logout Service Url
Single Logout Service Response Url
OK
Cancel
1
2


Hide Advanced Settings
Response
Signed
Assertion Signature
Signed
Signature Algorithm
RSA-SHA256
Digest Algorithm
SHA256
Assertion Encryption
Unencrypted
Enable Single Logout
Allow application to initiate Single Logout
Single Logout URL
https://testsecretserver.com/saml/sloservice.aspx
SP Issuer
https://testsecretserver.com/samlemetadata
Signature Certificate
Browse..
Upload Certificate
Authentication context class
PasswordProtectedTransport
Honor Force Authentication
Yes
SAML Issuer ID
http://www.okta.com/${org.externalKey}

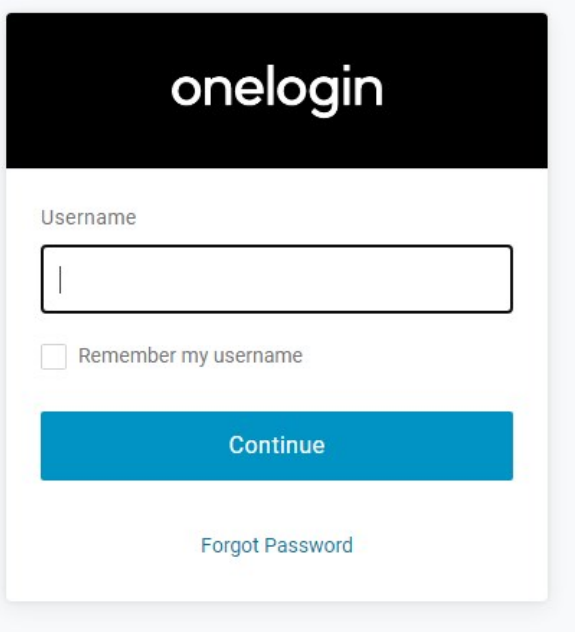
OneLogin Provisioning

Connect your SCIM service with a OneLogin integration

1. Begin by signing up for a developer account using URL: https://www.onelogin.com/developer-signup
2. After creating the account, you will receive an email with a link to verify your developer account.

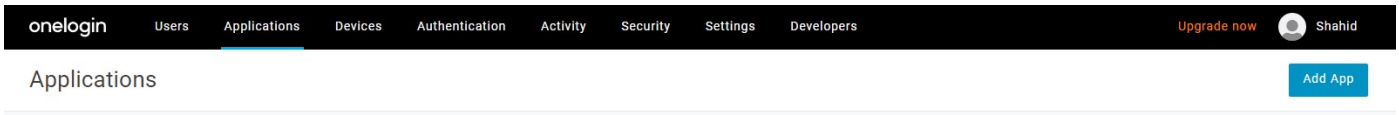


3. Enter the **username** and **password**.

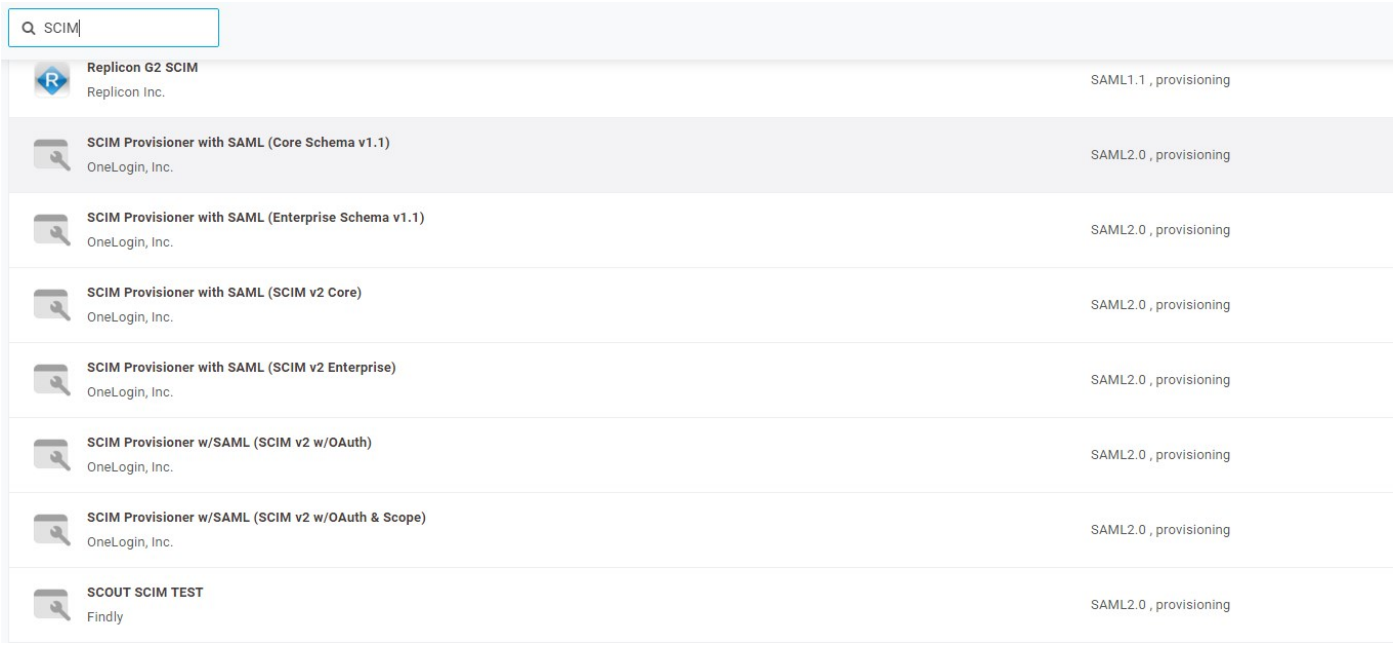


4. Click **Applications**.

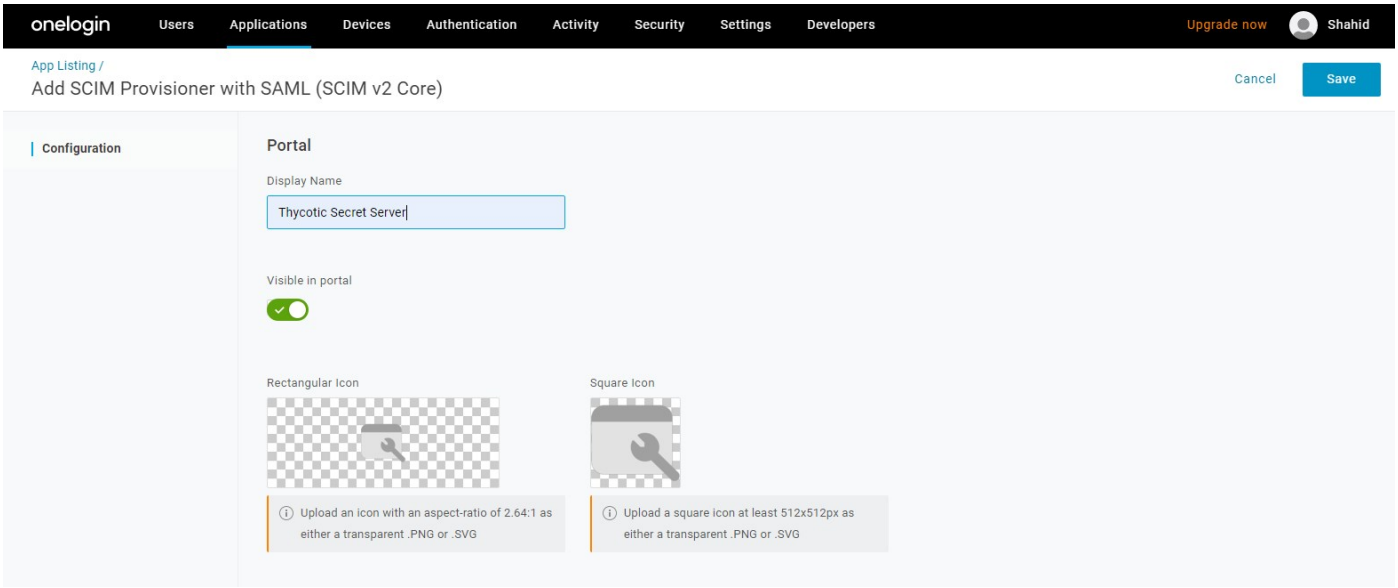
5. Click on **Add App**.



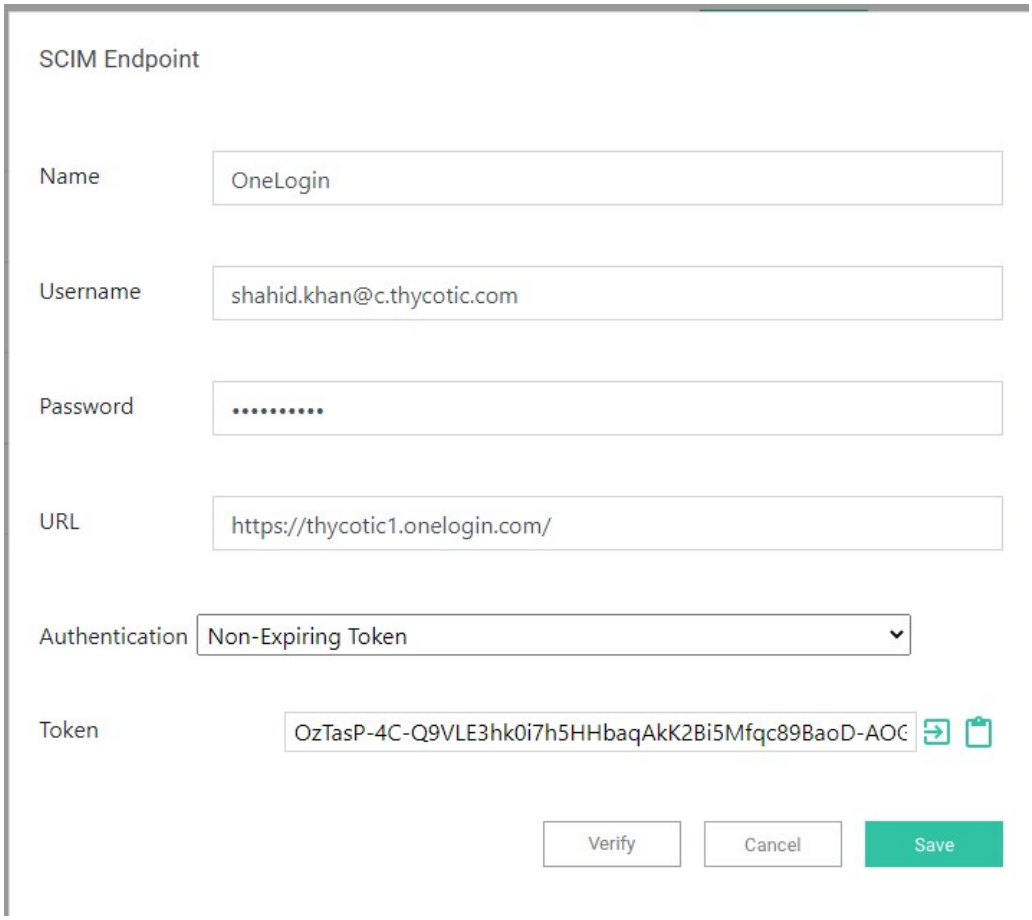
6. Search for **SCIM**.



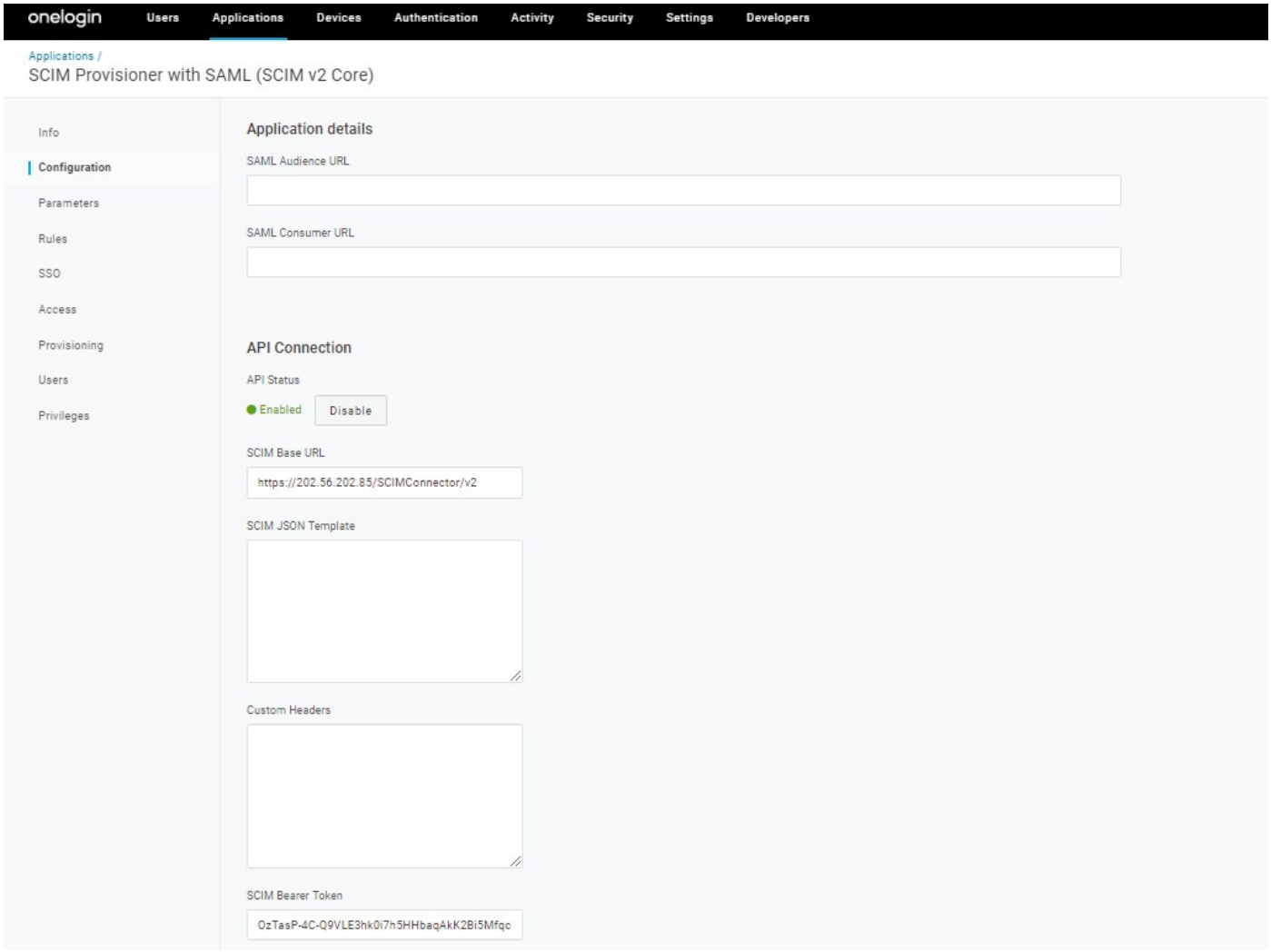
7. Select SCIM Provisioner with SAML(SCIM v2 core) and enter **display name** and click **Save**.



8. Create a new endpoint in “Thycotic SCIM Connector” for OneLOgin. To authenticate using SCIM Bearer Token, provide a bearer token to access your SCIM implementation.

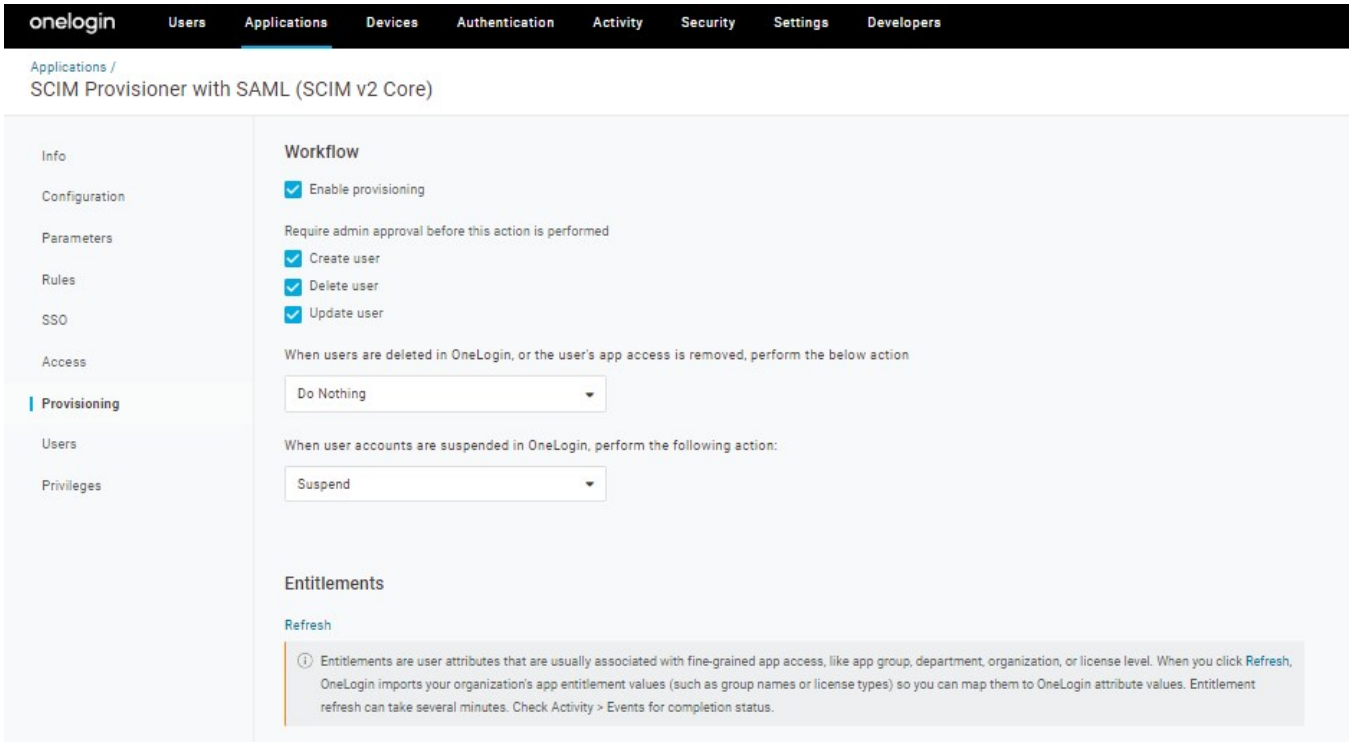


9. Click on **Configuration**, give your **SCIM Base URL** and **SCIM Bearer Token**.
10. Click on **Enable API Connection**.

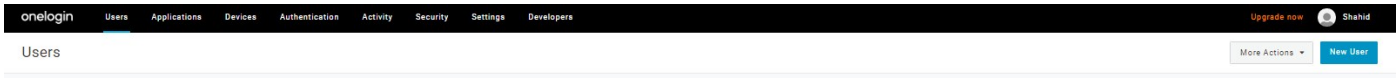


Configure your OneLogin integration

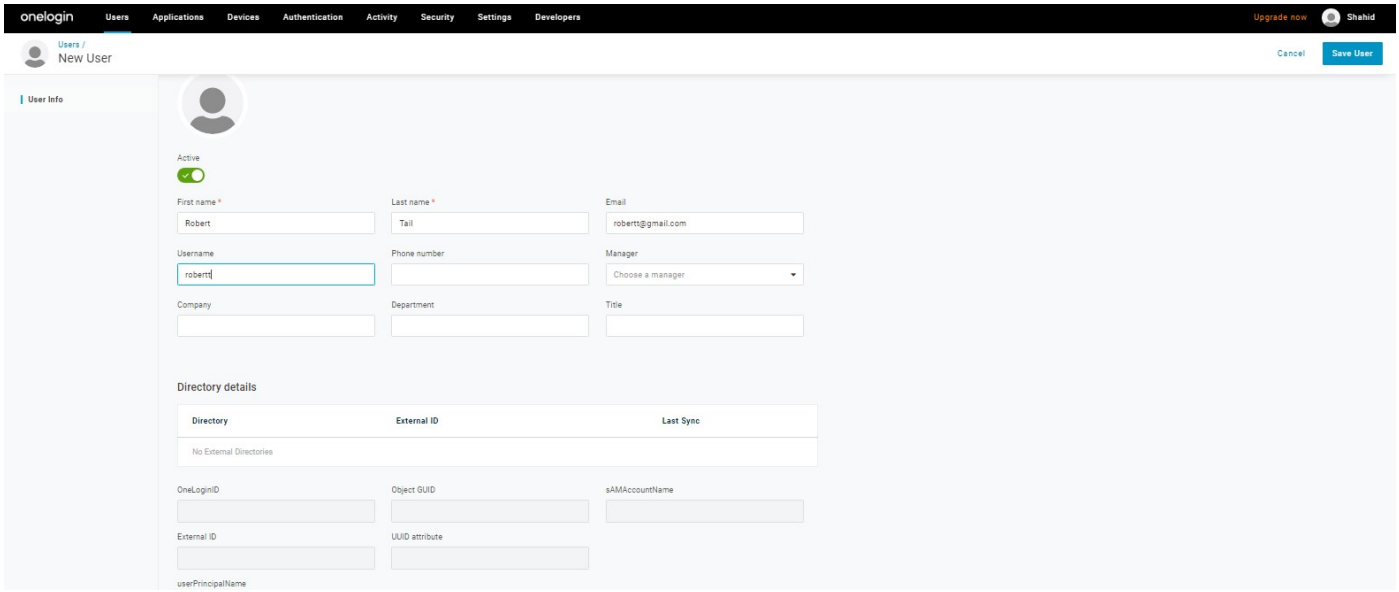
1. Click on **Provisioning** and select enable **Provisioning**.



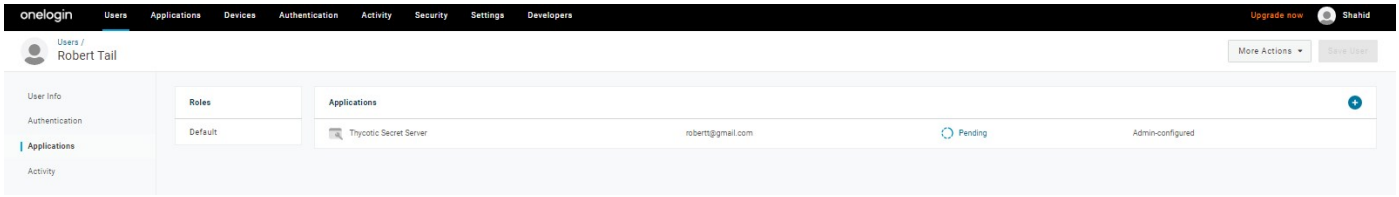
2. Click the **Users** tab, and click **New User**.



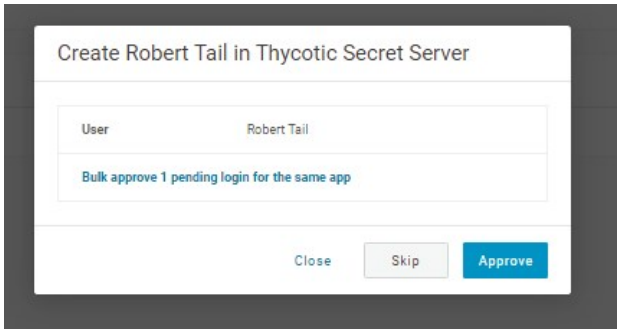
3. Enter the details of the user and click on the **Save User** button.



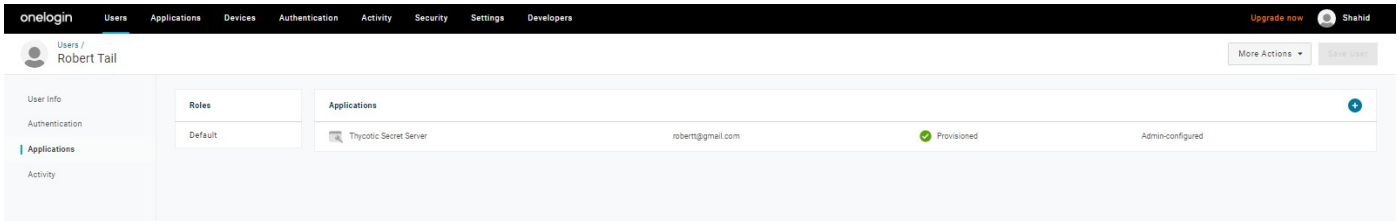
4. Click **Application** then click on **+** button to Add **Thycotic Secret Server** application.
5. Click on **Save**.



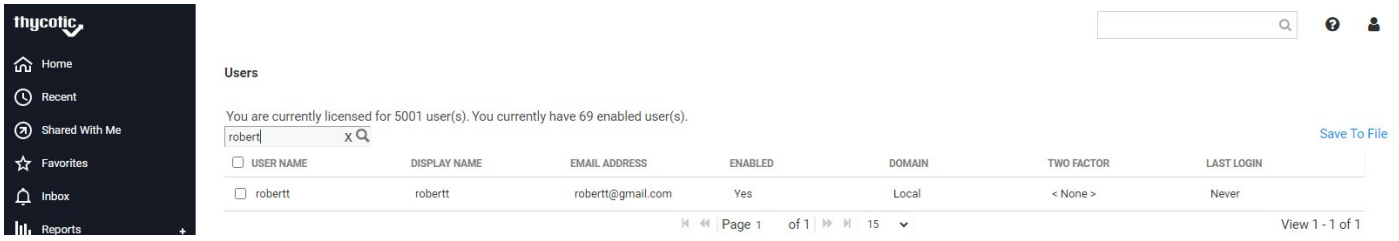
6. Click **Pending**.
7. Click **Approve**.



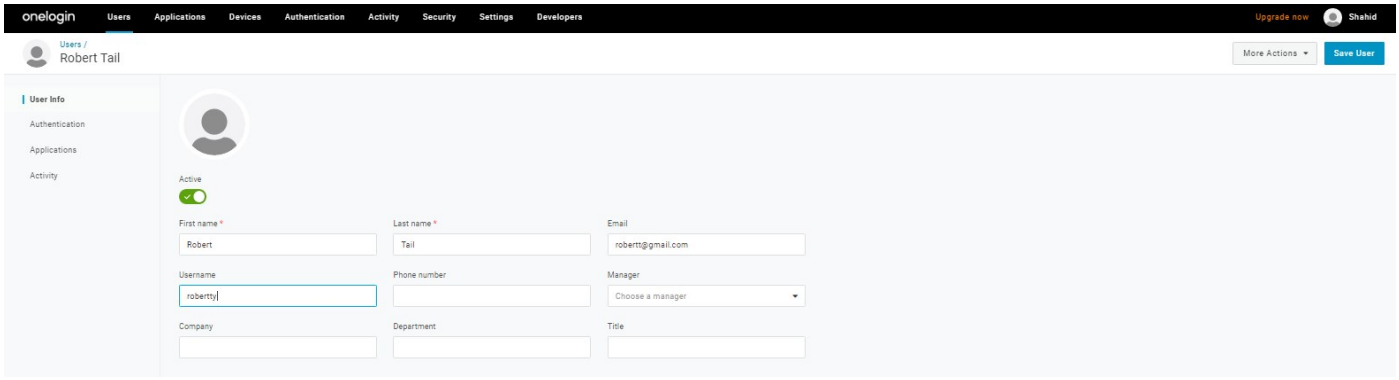
8. After clicking approve the user will be **Provisioned**.



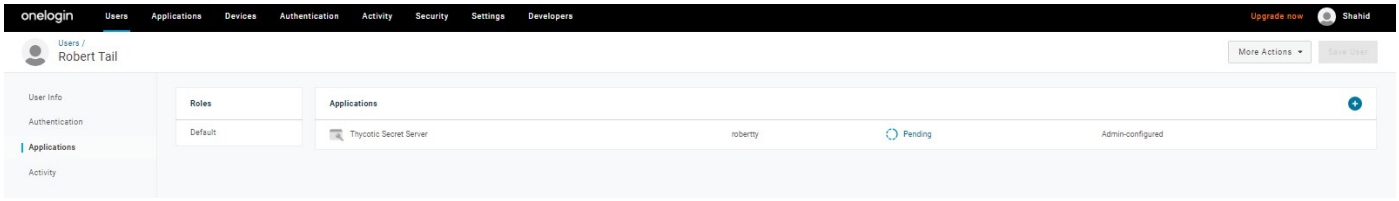
9. Navigate to **Secret Server | Users** and search for **robertt**. the user Robertt has been created in secret server.



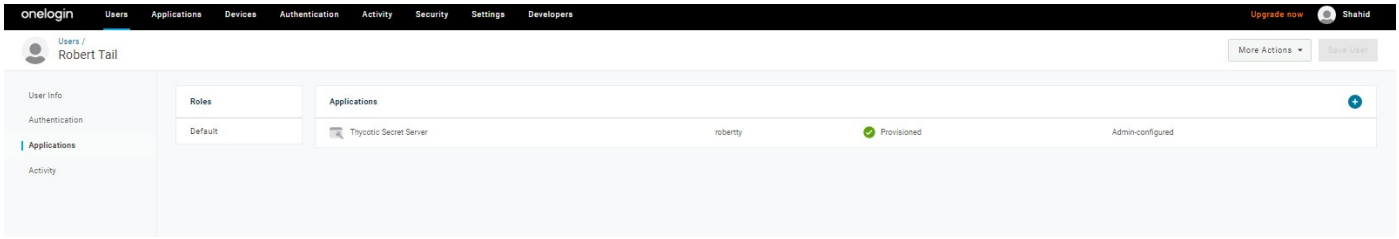
10. Click on "**User Info**" in OneLogin and update the username to **robertty**.
11. Click on **Save User**.



12. Now move to Application and click on **pending**.
13. Click on the **Approve** button.

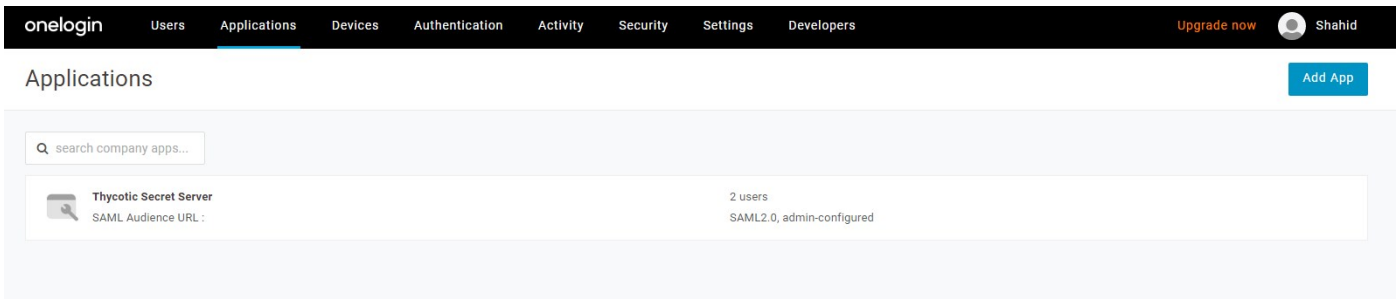


14. The User will be **modified** and **Provisioned**.

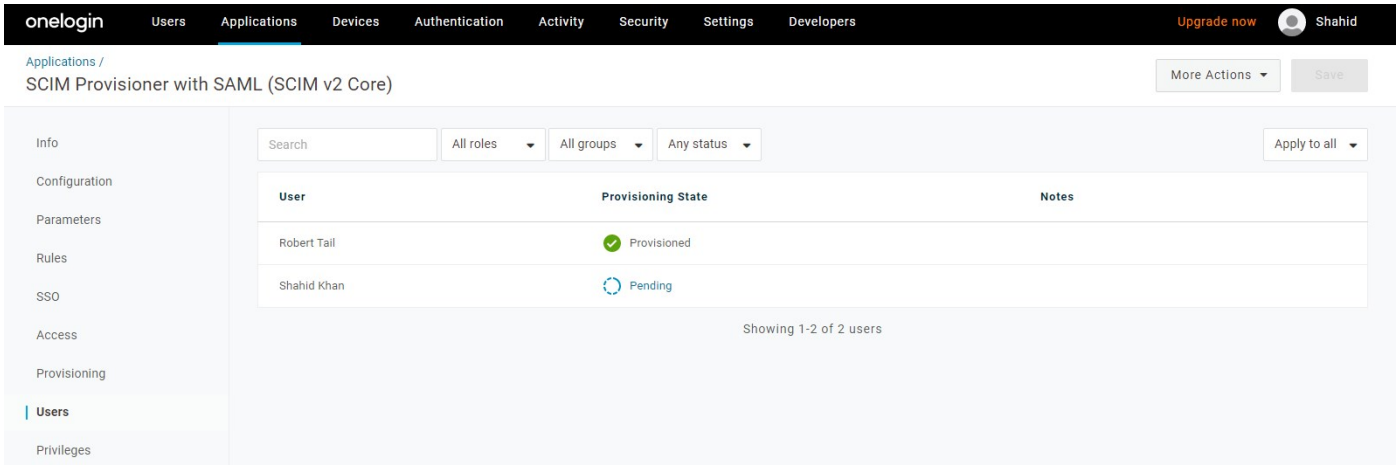


15. Navigate back to **Secret server** and you should see the user **displayName** is updated with **robertty** .

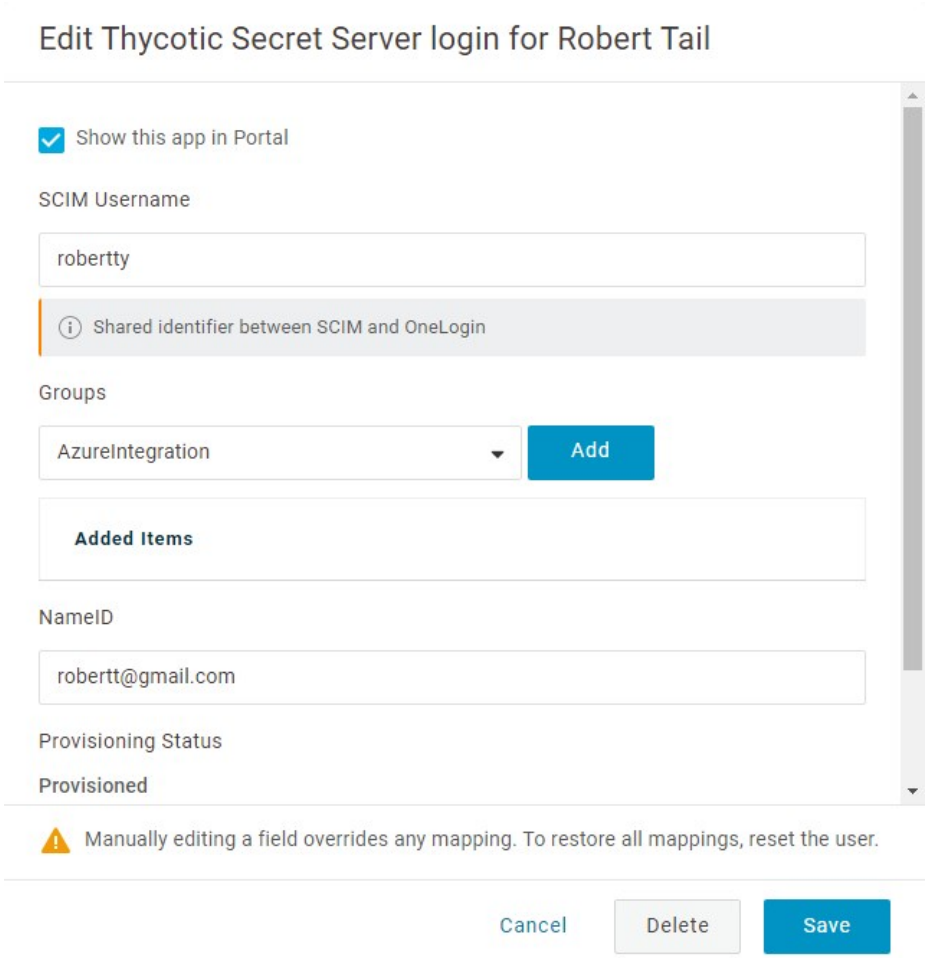
16. Click on **Application** I **Applications** and select **Thycotic Secret Server** application.



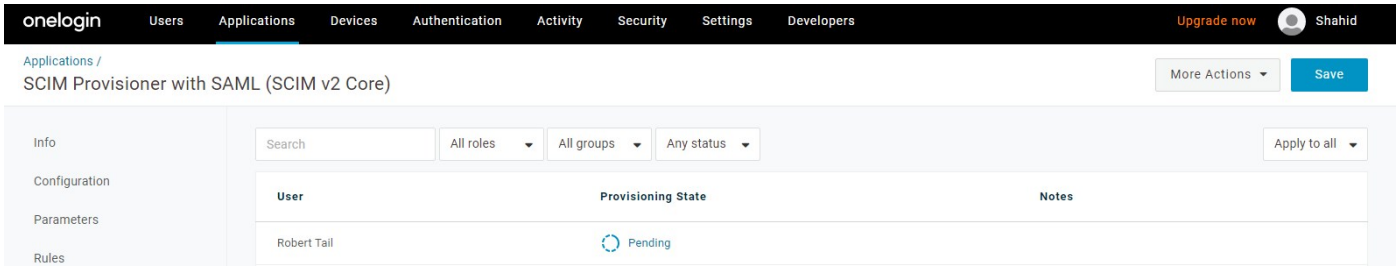
17. Click on **Users** and select Robert tail user.



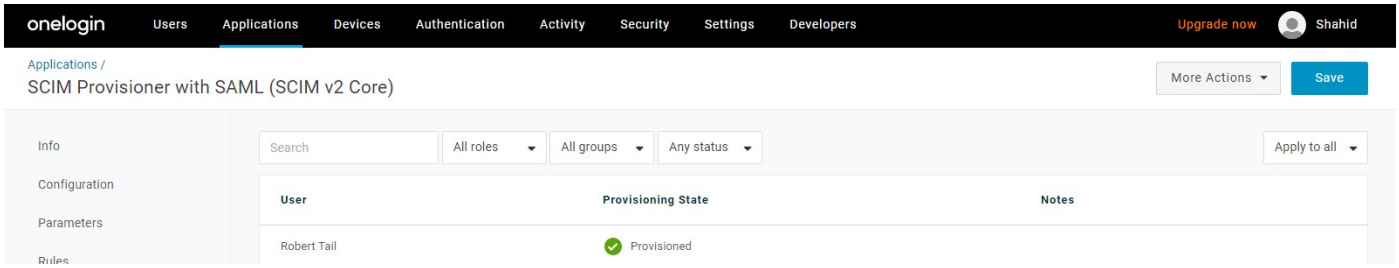
18. Select the group that has come from Secert Server for the user and click on **Add**.
19. Click **Save**.



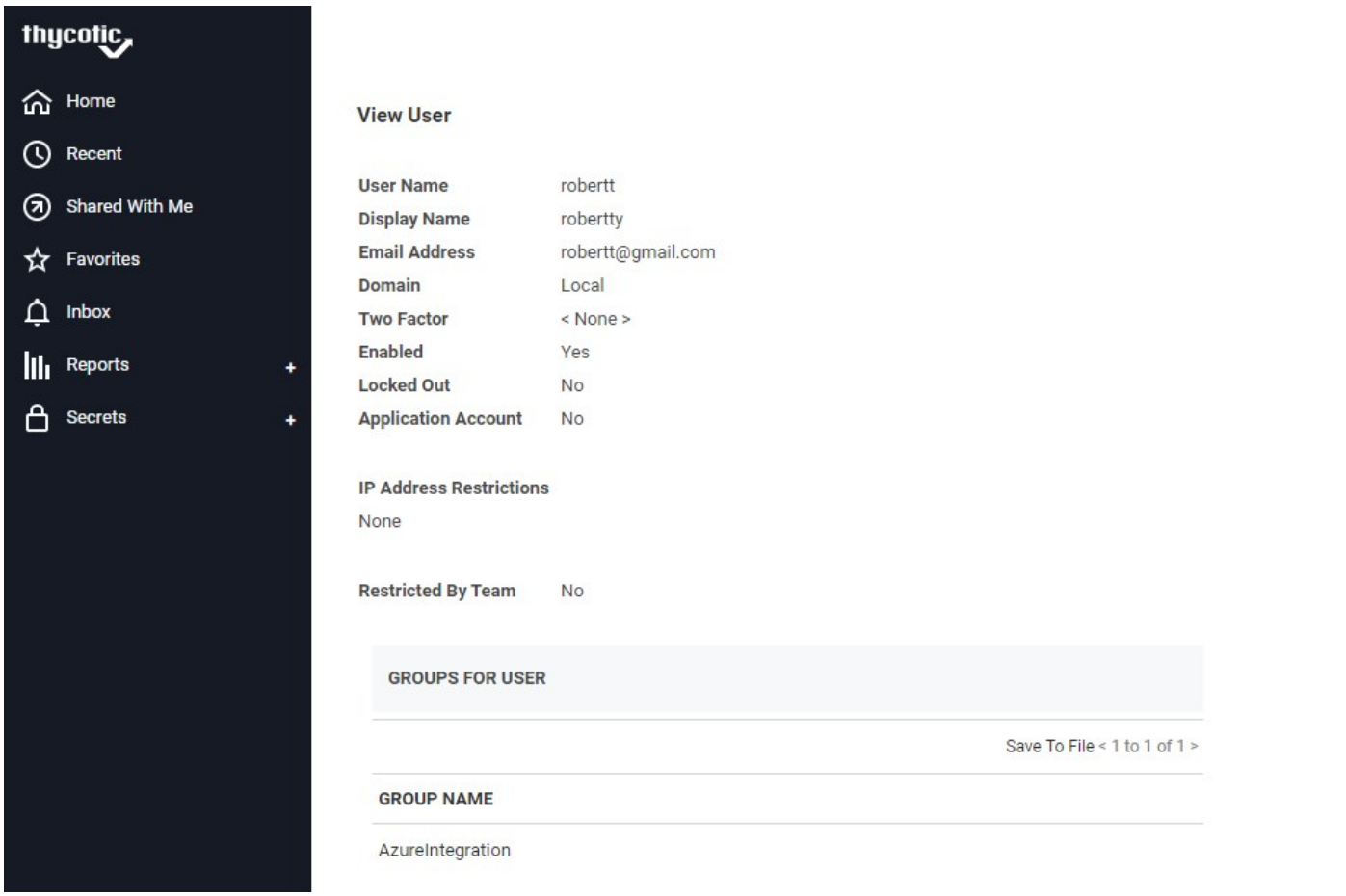
20. Click on **Pending** for the user and approve it.



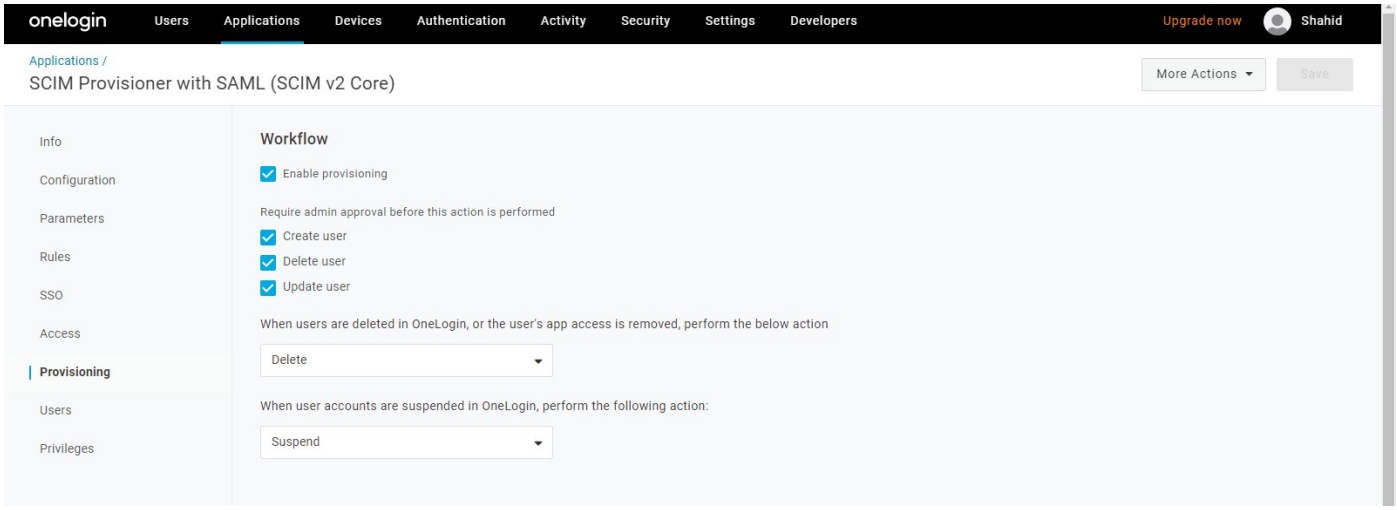
21. The user is **Provisioned**.



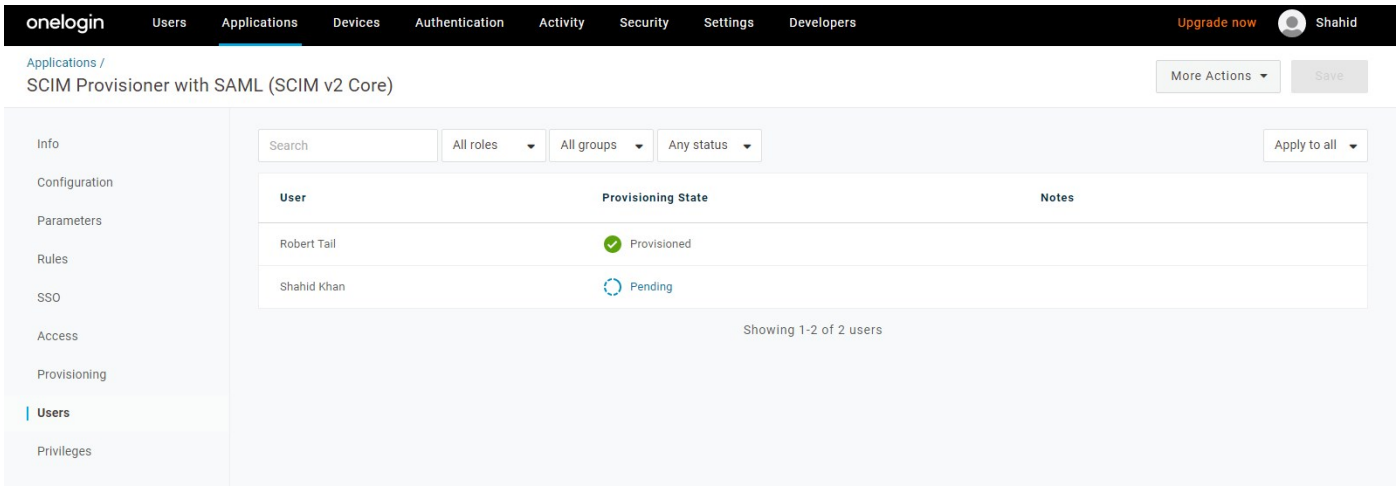
22. Navigate back to **Secret Server** and search for user “robertt” and click on the user. You should now see that the group is getting assigned to the user.



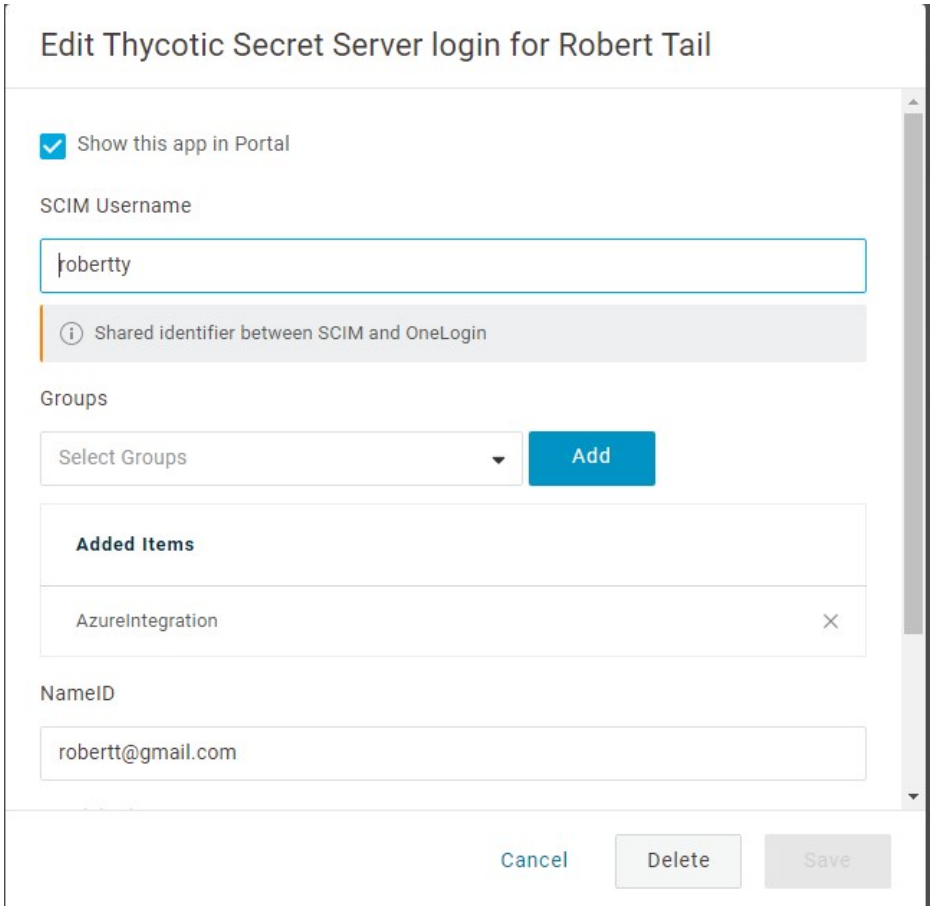
23. Click on **Application** I **Applications** and select **Thycotic Secret Server** application.
24. Click on **Provisioning** Select “When users are deleted in OneLogin, or the user's app access is removed, perform the below action” to **Delete**.
25. Click **Save**.



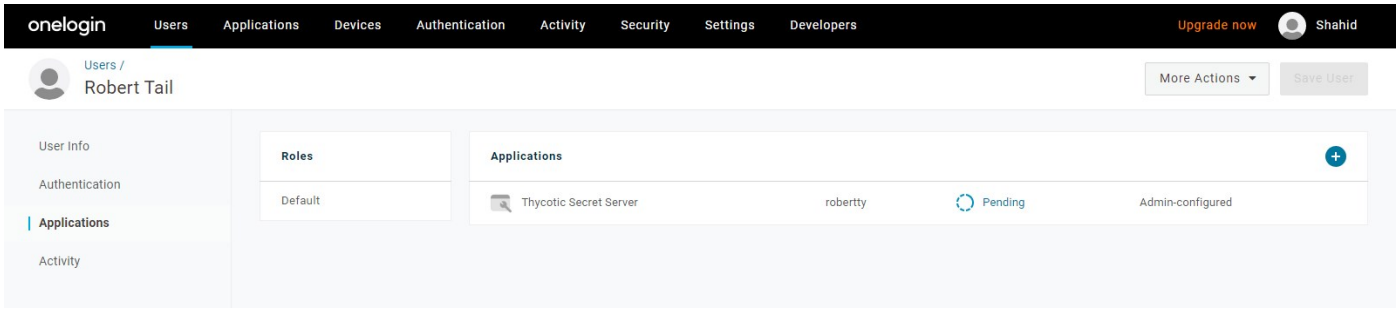
26. Click on **Users** Robert Tail.



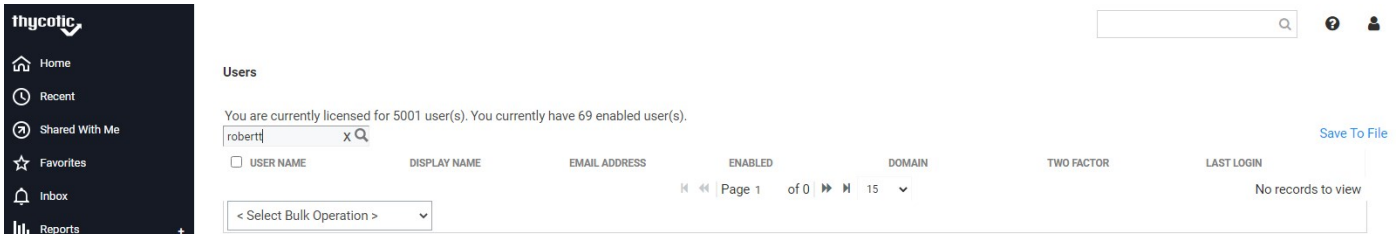
27. Click on the “**Delete**” button.



28. Navigate to **Users | Applications**.
29. Click on **Pending** and approve it.



30. Navigate to **Secret Server| Users** and search for the robertt user, you should see the robertt user is deleted



Introduction

The integration between Thycotic Secret Server and Qualys is created and maintained by Qualys. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Using Secret Server as a Credential Vault for Authenticated Scans

Secret Server is an on-premise, web-based password vault used to help organizations properly manage privileged account passwords. Secret Server allows users to control access and automate password changes for a variety of enterprise resources. Organizations can easily deploy Secret Server to be more secure, reduce labor costs, adopt password best practices, and satisfy audit requirements.

QualysGuard can use the Secret Server as a Credential Vault for the accounts used for authenticated scanning. Instead of adding individual credentials for trusted scans, the Administrator can use named records stored in Secret Server. There are several benefits to this approach:

- Using Secret Server means that all the credentials used for authenticated scans will be stored securely on-premise and will not leave the network.
- Password rotation can happen frequently and automatically as Secret Server performs the password changes and QualysGuard retrieves the passwords as needed during scans.
- Credentials can still be securely controlled in Secret Server with full auditing over their usage.

Configuration

Please review the following steps to setup and configure Qualys for Secret Server:

- Configure the Vault.
- Creating Authentication Records.
- Scan for Vulnerabilities.

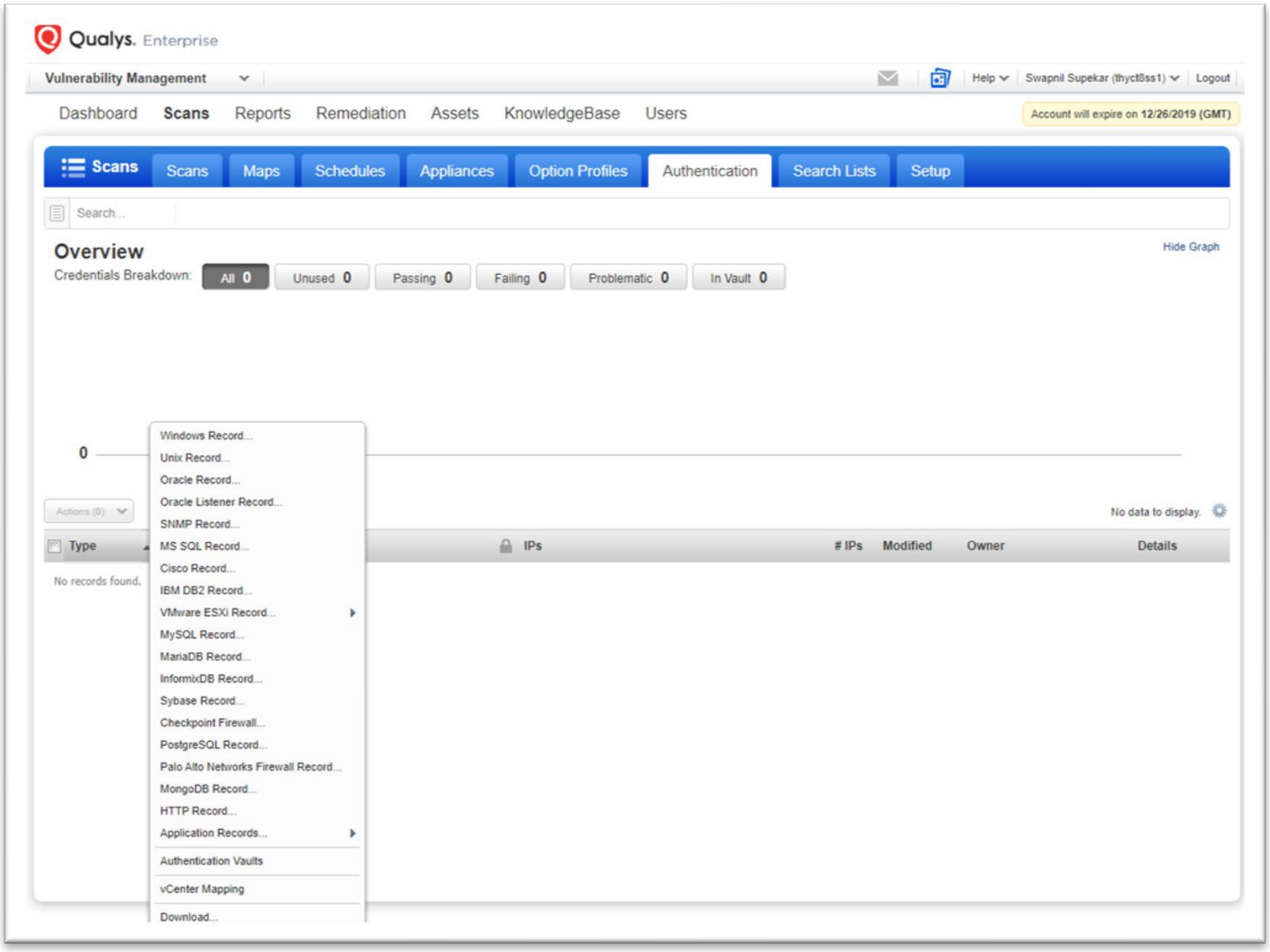
Configure the Vault

To use Secret Server, an administrator must configure it as a Vault within Qualys by specifying a URL and credentials to access the on-premises Secret Server instance.

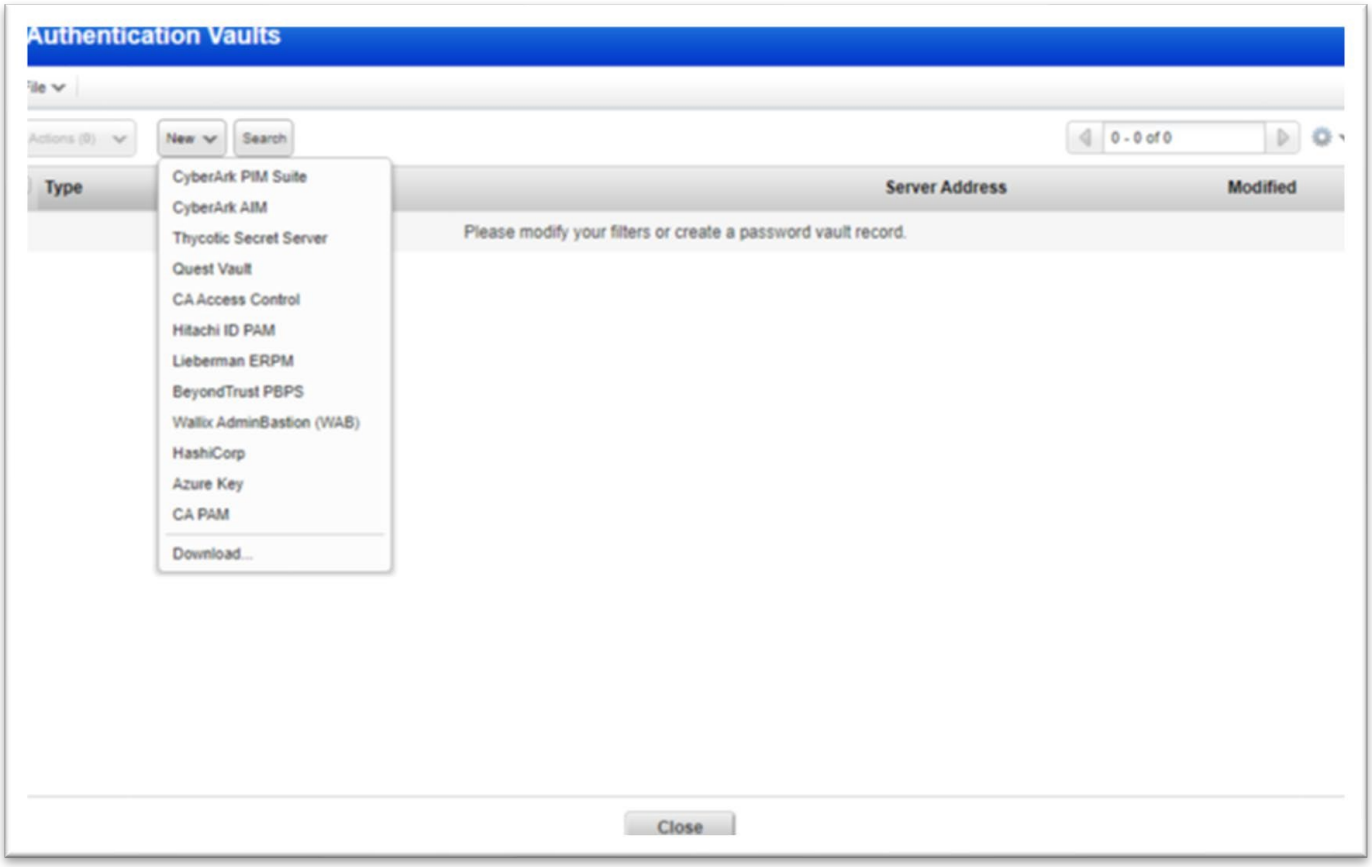
Instead of adding username/password credentials for use in trusted scans, the administrator can point to named records stored in Secret Server. Qualys will retrieve the credentials from Secret Server at scan time for trusted scans.

Add New Authentication Vault in Qualys

1. Navigate to the Scan and click the **Authentication** tab.

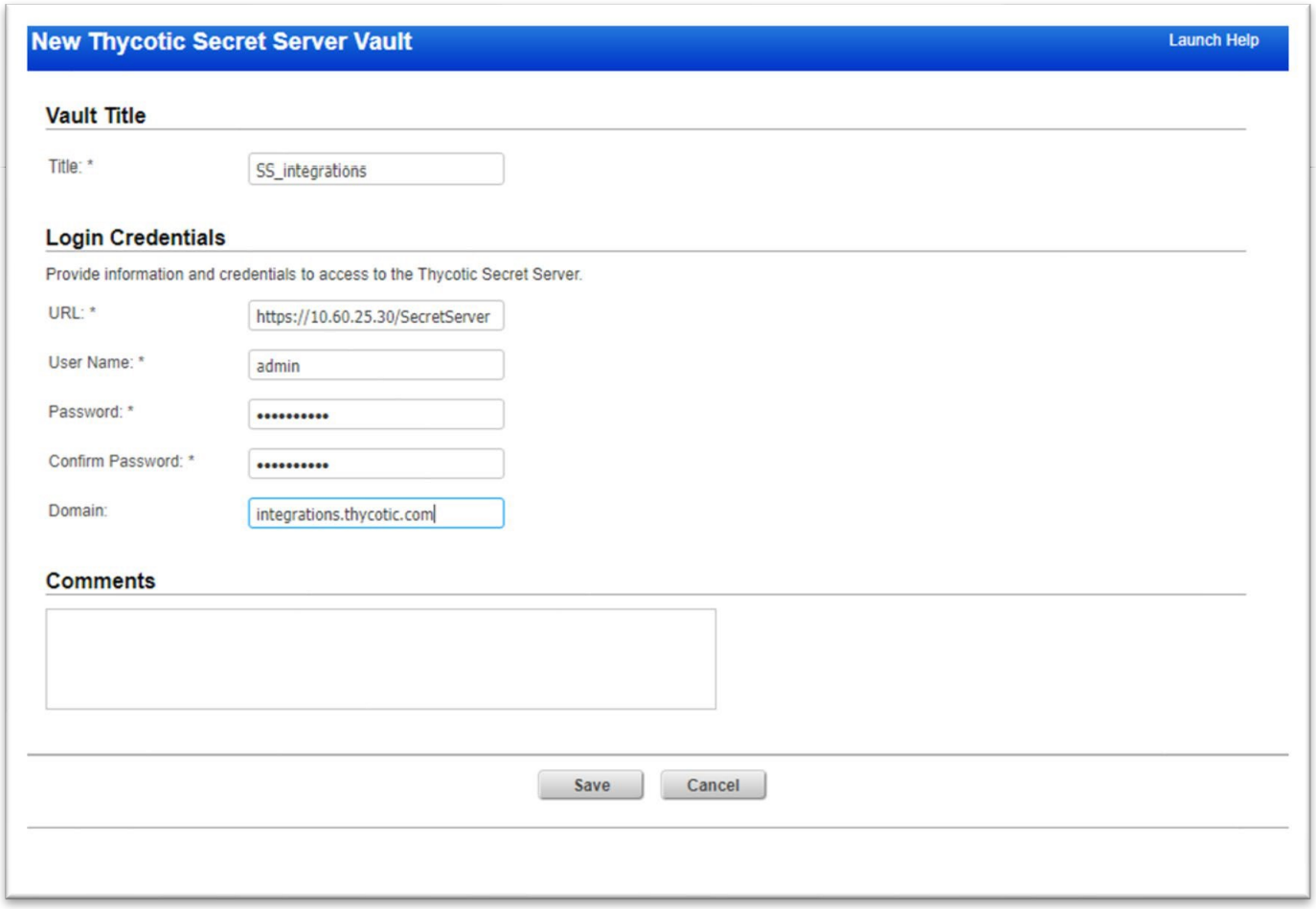


2. Select **Authentication Vaults** from the new drop-down list.



3. Select **Thycotic Secret Server** from the list.
4. Enter the following **access information** for your Secret Server site:
    - **URL**: This is the URL for Secret Server web services. Ensure web services are enabled in your Secret Server instance by clicking **Configuration** from the Administration menu and enabling web services.
    - Add /sswebservices/sswebservice.asmx to your Secret Server URL to obtain the URL for the web services:

      https://yoursecretserver/webservices/sswebservice.asmx.



Note: If you do not have SSL enabled, web services can still be accessed via http but it is not advisable for production systems. The vault is accessed from the scan agent, so the Secret Server website must be reachable from the Qualys scanner appliance – not the Qualys cloud instance.

- **User Name**: The user account for accessing Secret Server. This can either be a local Secret Server account or an Active Directory account. User accounts can be created in Secret Server from the Users section of the Administration menu. This user account should be an application account.

5. Click **Advanced**, click the checkbox of **Application Account**.
6. Click **Save**.


Edit User
User Name
qualys_scanner
Display Name
qualys_scanner
Email Address
Domain
Local
Password
Confirm
Two Factor
< None >
Enabled
Locked Out
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
User Administrators
Save
Cancel

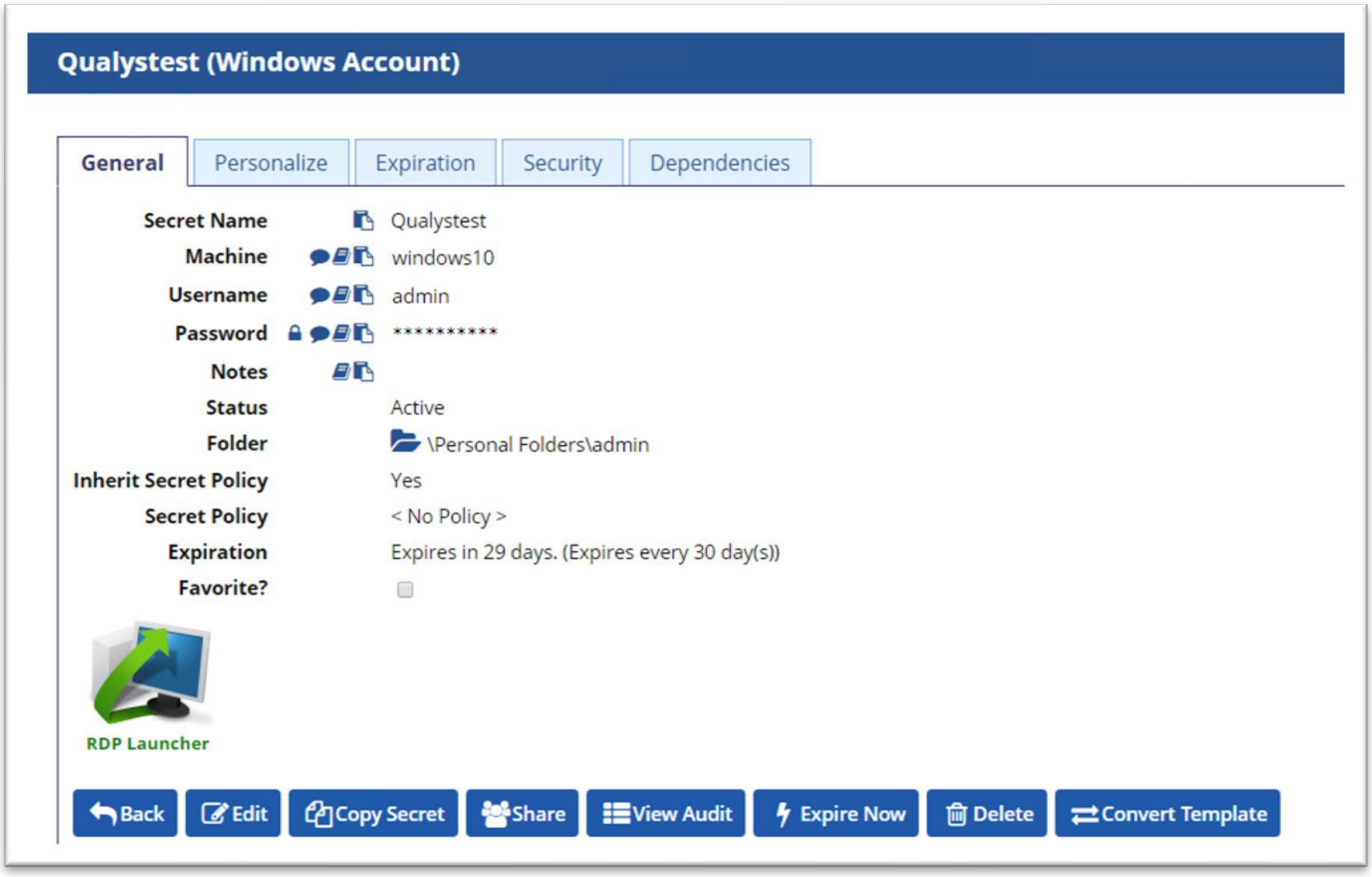
Creating Authentication Records

Once the authentication vault has been configured, individual authentication credentials can be configured to retrieve their passwords from Secret Server.

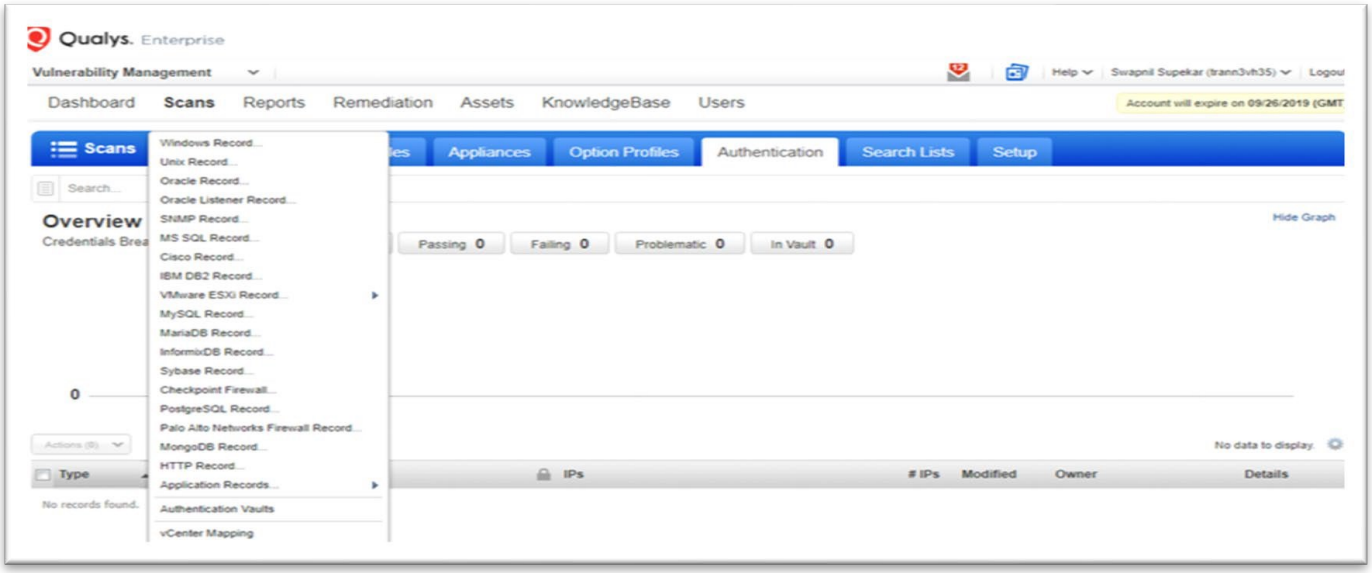
1. **Authentication l New l record type**.

   Authentication vault configuration requires three additional details to retrieve the password:

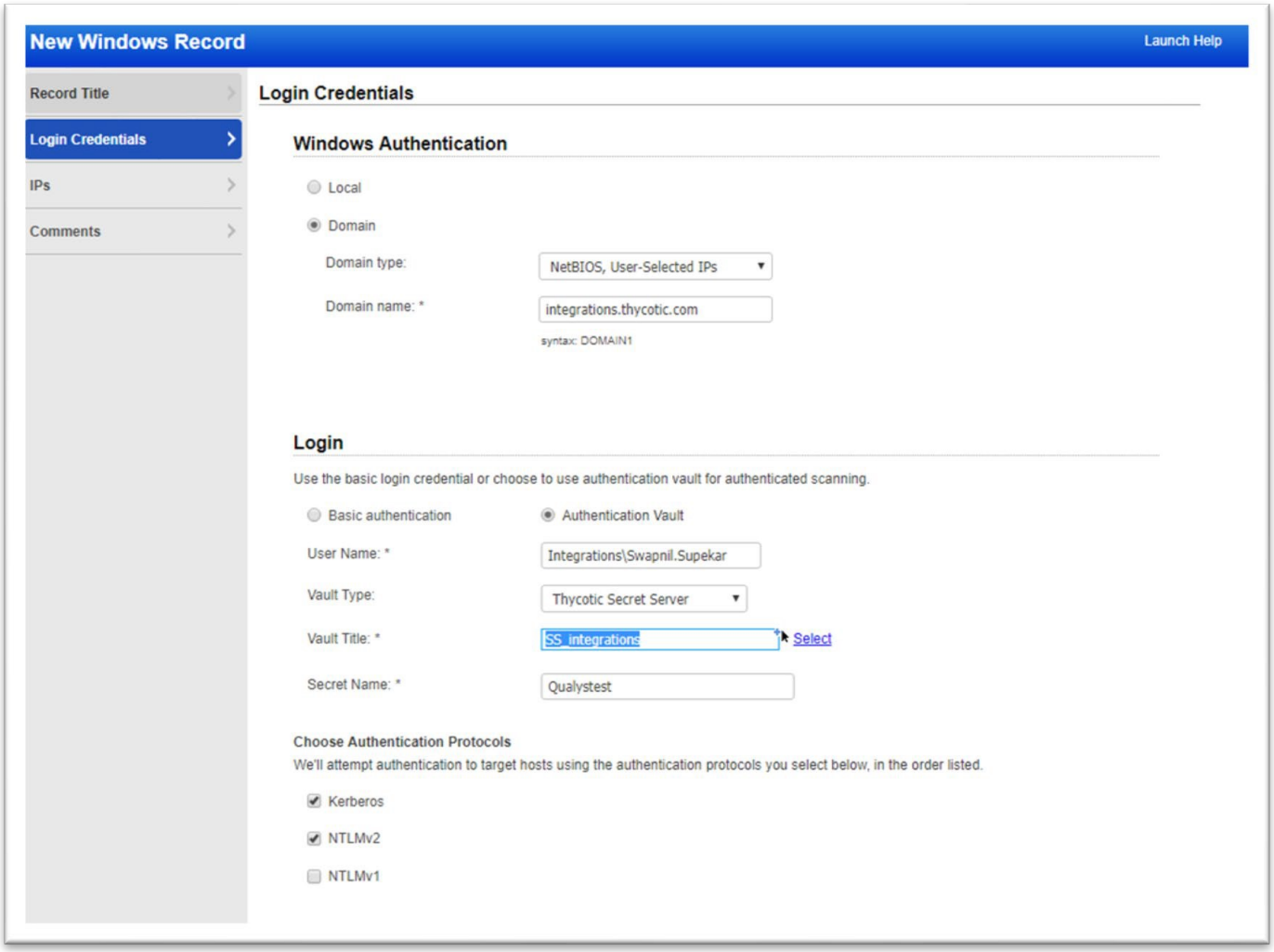
   - **Vault Type**: Set to Thycotic Secret Server.
   - **Vault Title**: The previously created Authentication Vault record in Qualys.
   - **Secret Name**: The Secret record in Secret Server containing the accounts password. In this case, the Secret name for the Windows account is Qualystest.



2. Navigate to **Scan** > **Authentication**.
3. Click on **New** dropdown and select **Windows Record**.

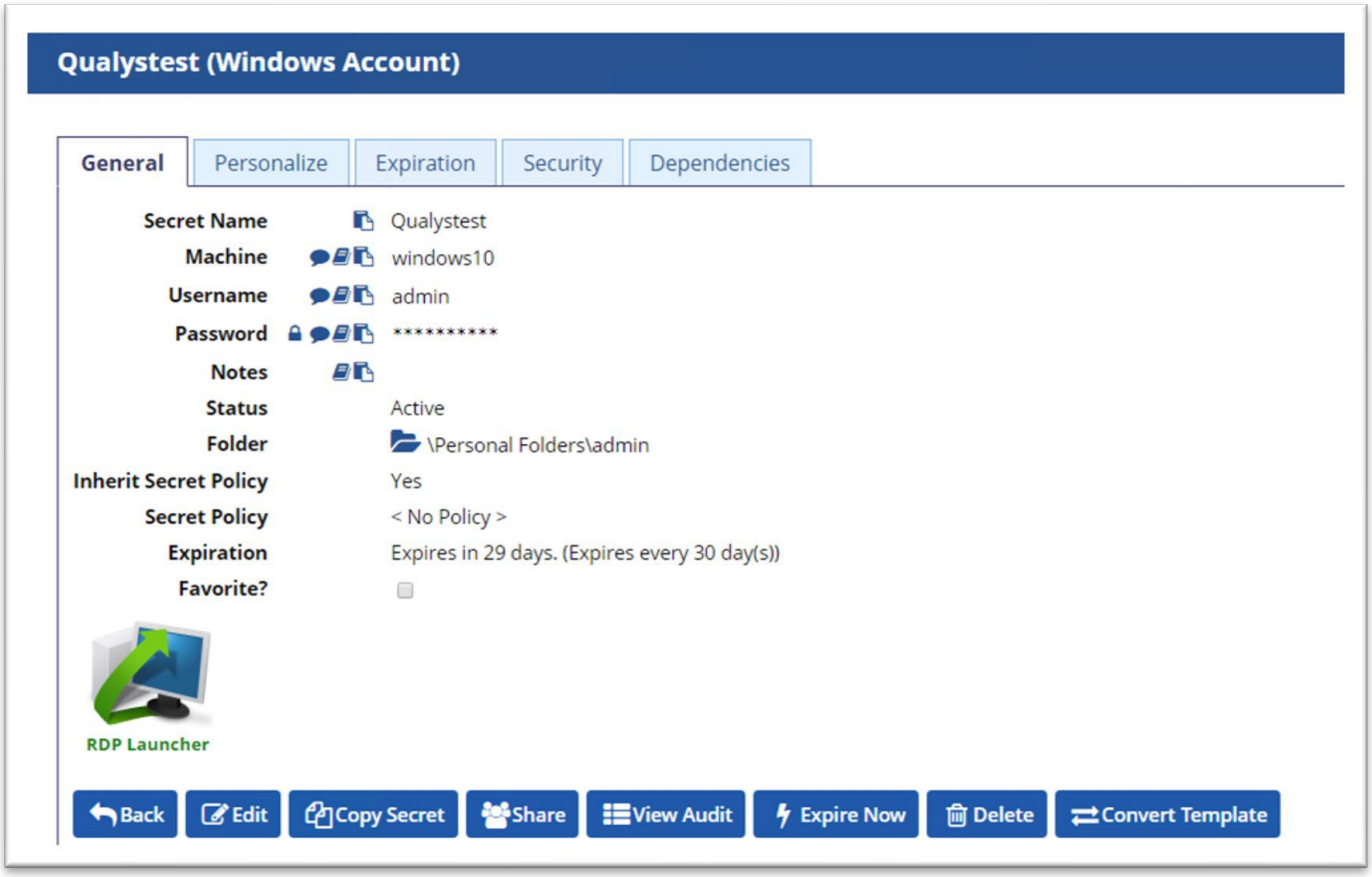


Note: The Secret name must match the corresponding Secret name in Secret Server.



Note: The **UserName** in the above screenshot should be the same as the Secret's username. In addition to creating a Secret with the correct password for the credentials used for authenticated scanning, the Application account (set in the previous steps for Configuring the Vault, under step 4) must have at least a View access to the Secret.

4. Click **Share** to view the permissions on the Secret.



A Secret inherits permissions from the folder settings. View the folder level permissions by editing the folder in which the Secret is stored.

Once the Secret is configured with the proper permission, Qualys can use it in scans. Run a scan that uses that authentication record to verify that everything is working end-to-end.

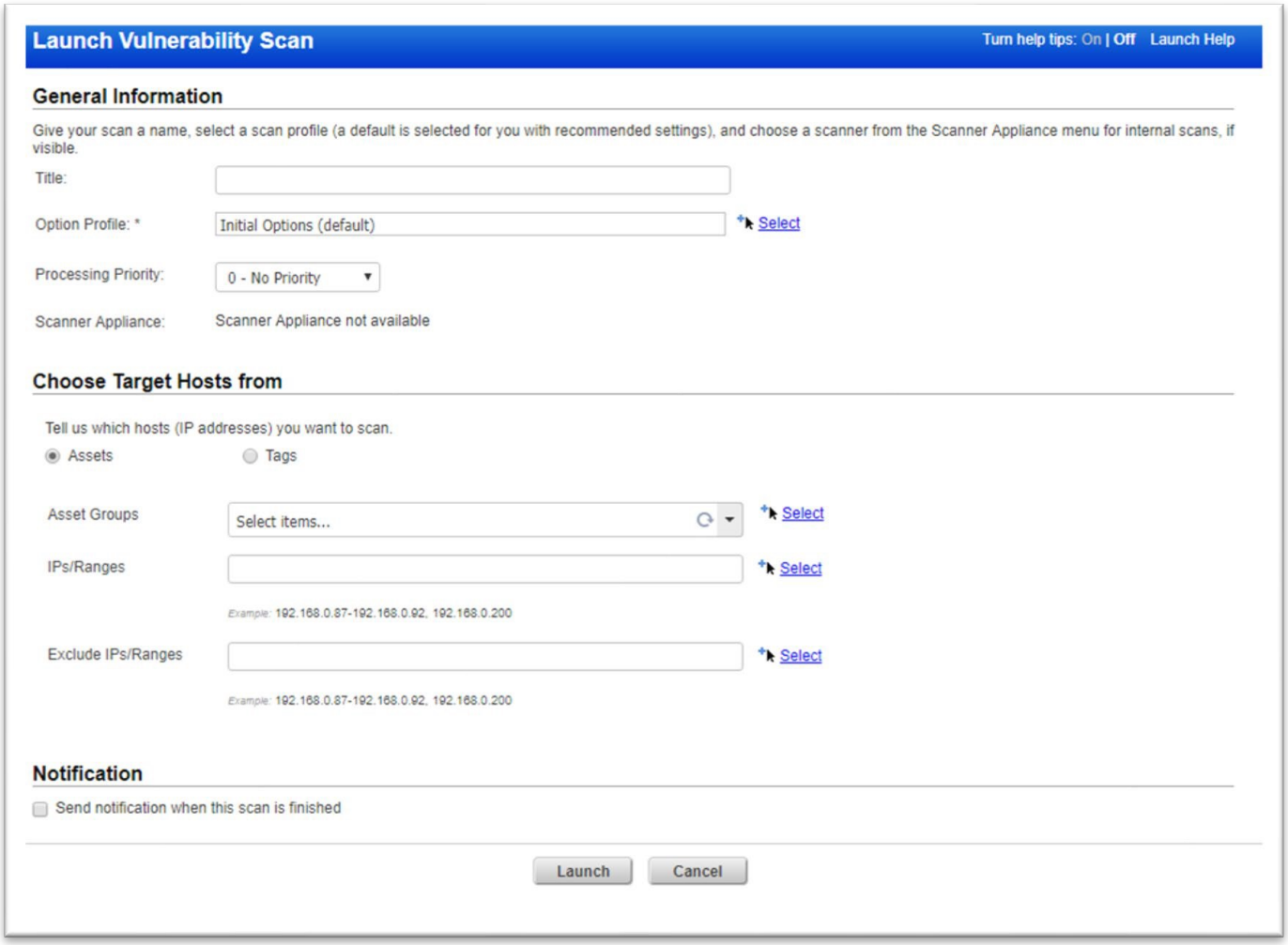
Scan for Vulnerabilities

It is important to scan your systems for known vulnerabilities to understand security risk. By automating scans, you'll get up to date security intelligence in real time.

Launch a Vulnerability Scan

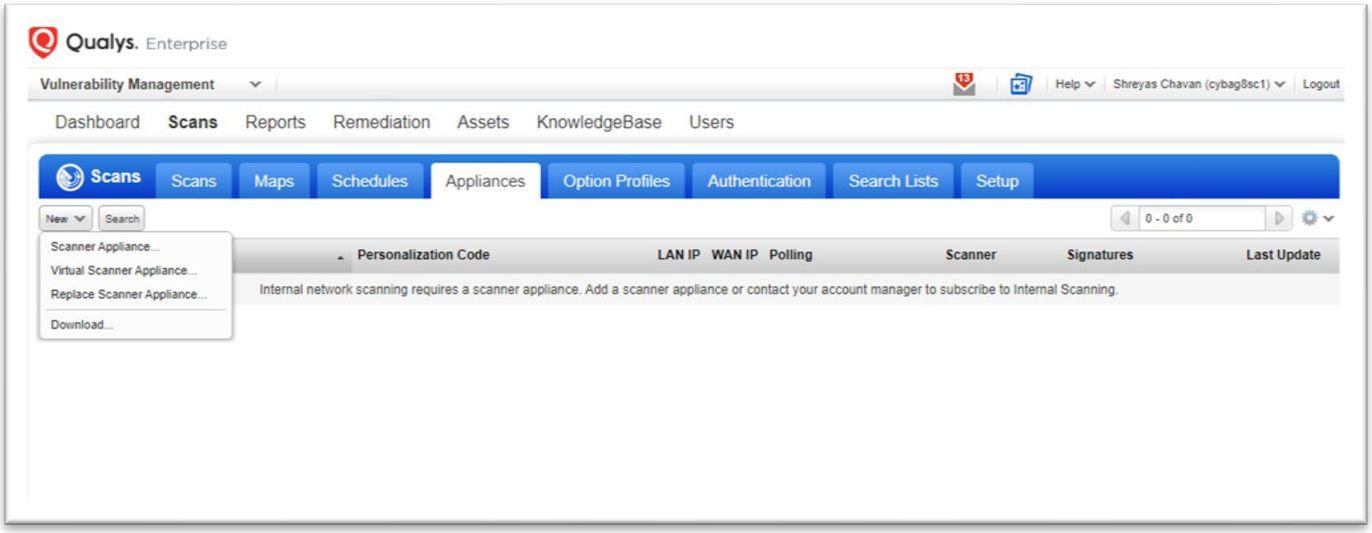
1. Navigate to **Scans I Scans I New I Scan** (or Schedule Scan). The Launch Vulnerability Scan window will open.

   Note: If Scanner Appliance: field reads *Scanner Appliance not available,* it must be set up.

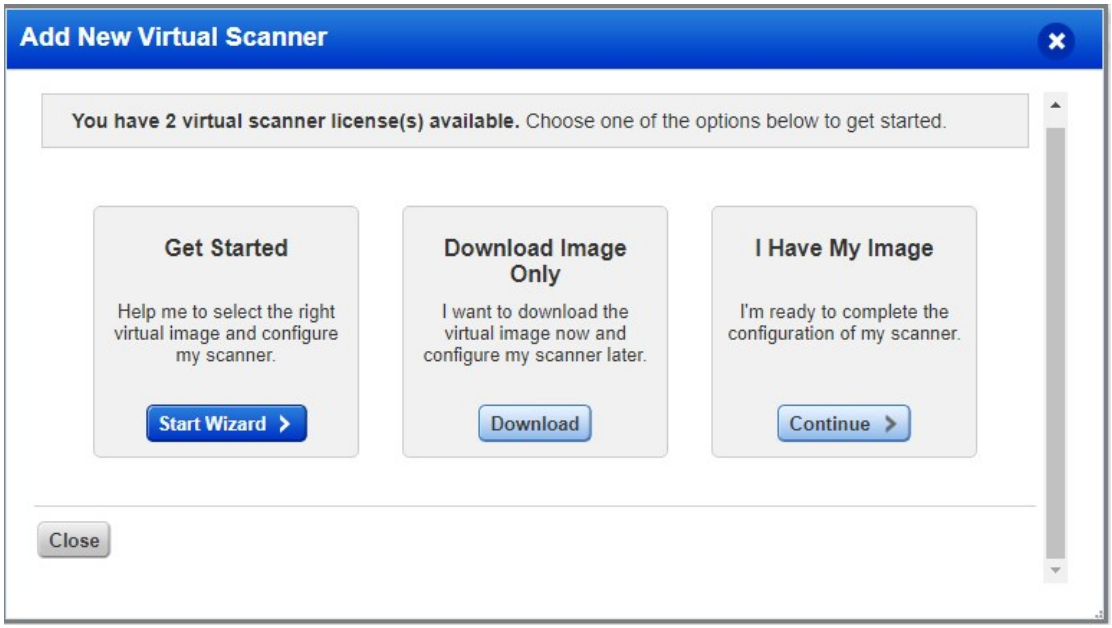


Set up Scanner Appliance

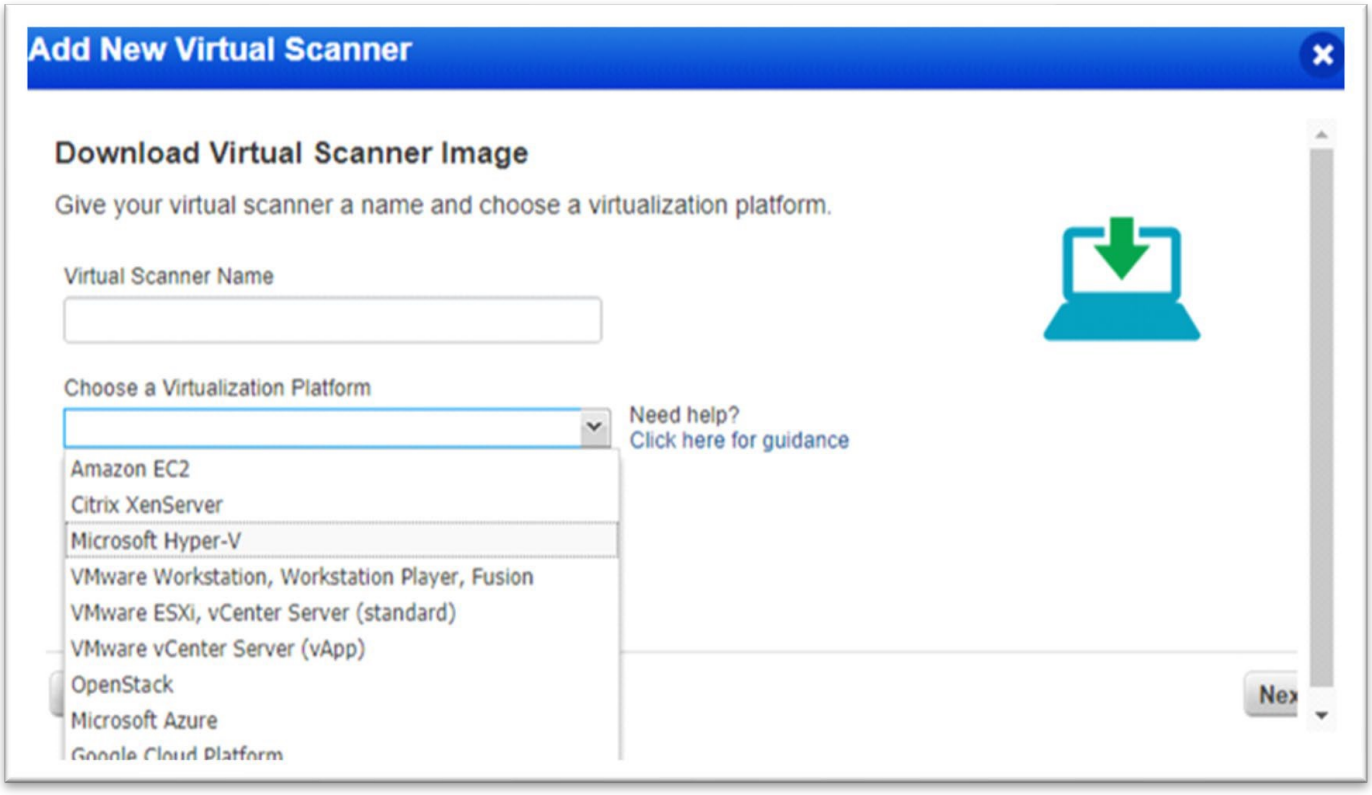
1. Navigate to **Scans I Appliances I Select** option.



2. Navigate to **Scans I Appliances I New I Virtual Scanner Appliance**.



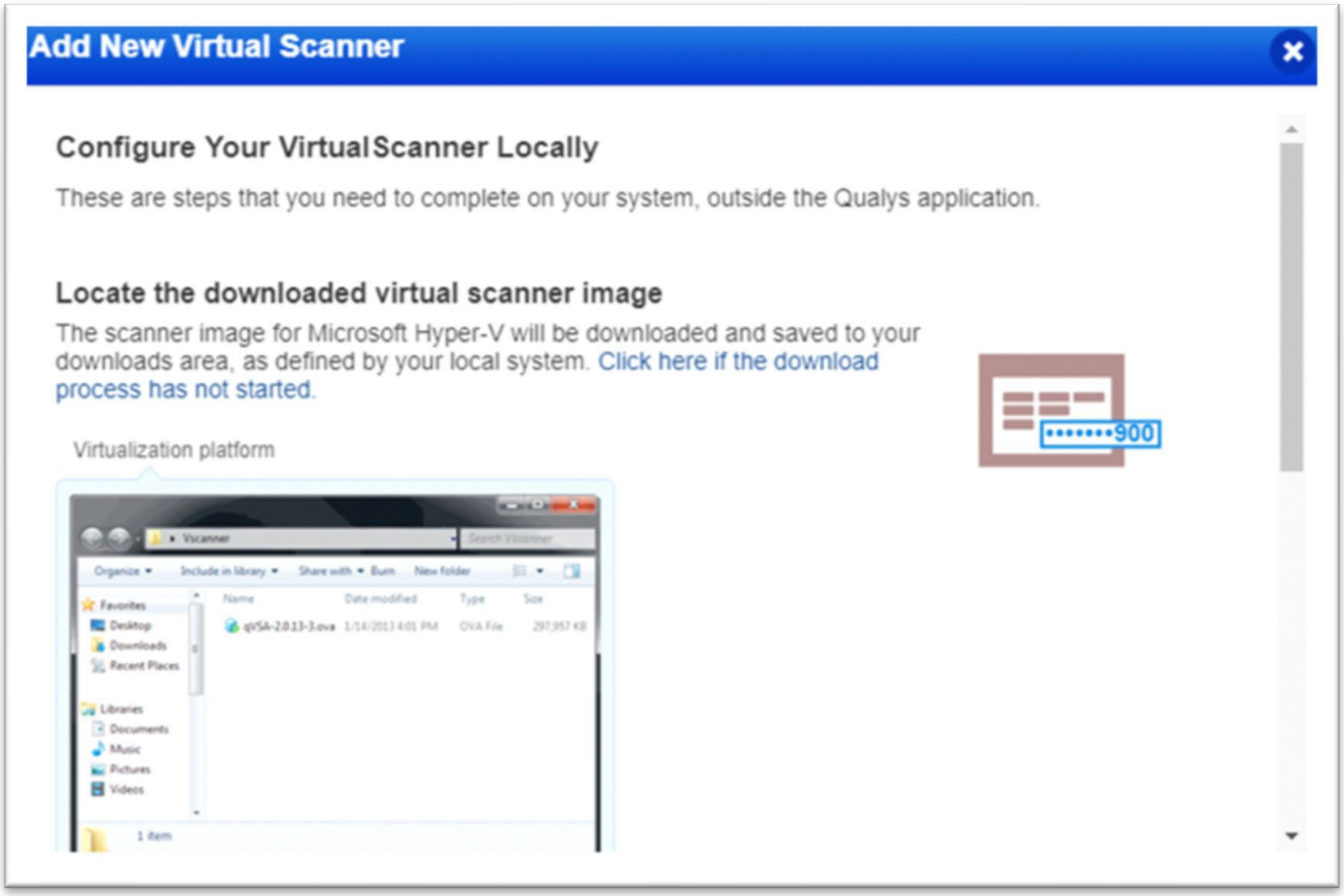
3. Click **Start Wizard**.
4. Enter the scanner name and select **the preferred virtualization platform**.



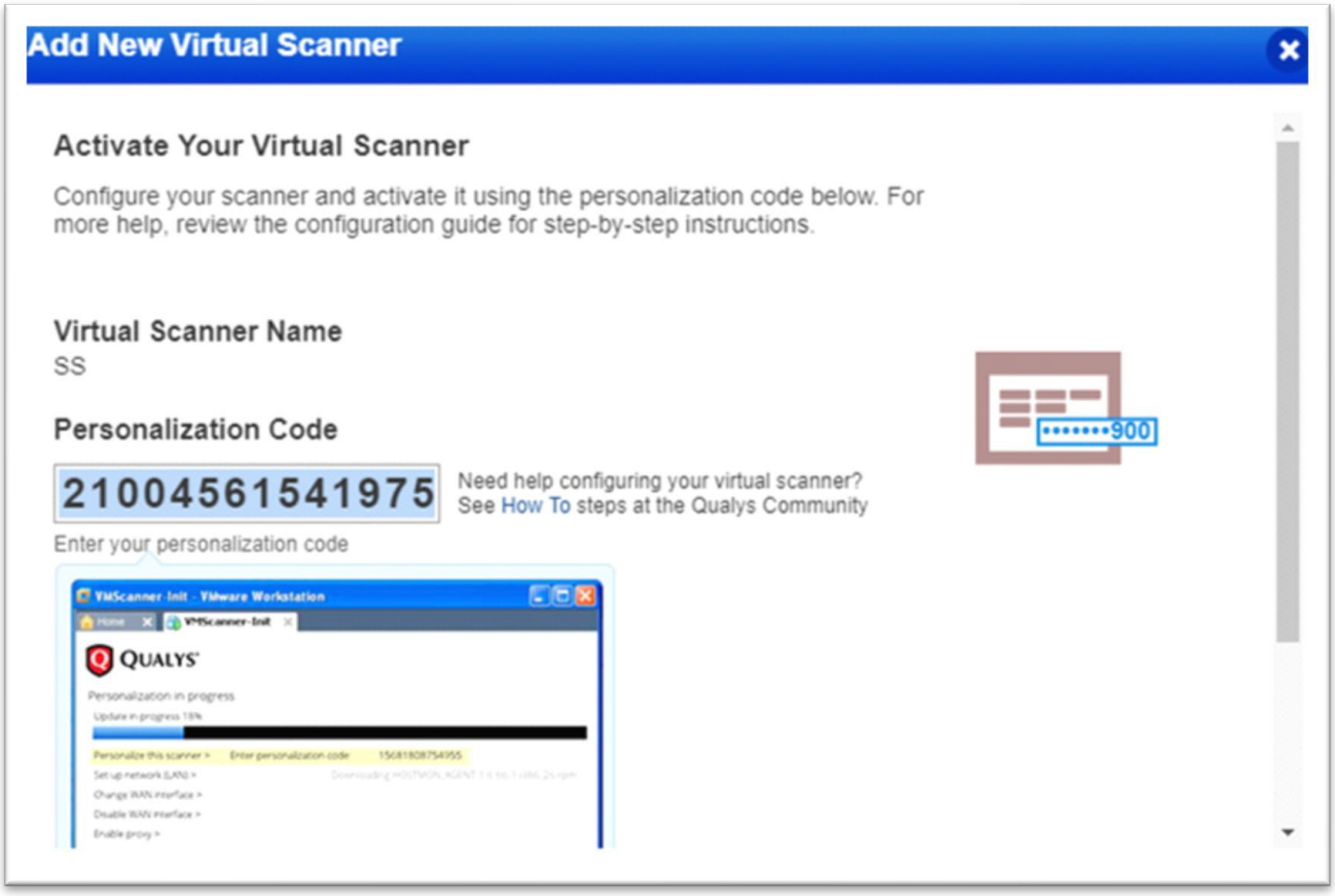
5. Download the **Virtual Scanner Image**.

   Note: This step applies to virtualization platforms with a scanner appliance image download (e.g. for VMware, Citrix XenServer, etc.)

6. Locate the Virtual Scanner Image on your local system.



7. Copy the **Personalization code** to a safe place as it will be required later.



8. Personalize Your Scanner Local system or server.

These steps apply when you have downloaded a scanner appliance image (i.e. for VMware, Citrix XenServer, etc.). You'll use Virtual Scanner Console running on your virtualization software to complete these steps.

Configure a Virtual Scanner using Microsoft Hyper-V

Note: The following steps assume you have downloaded the virtual scanner image (qVSA-2.0.13-1- vhd.zip or later) and obtained a personalization code as noted in the above sections.

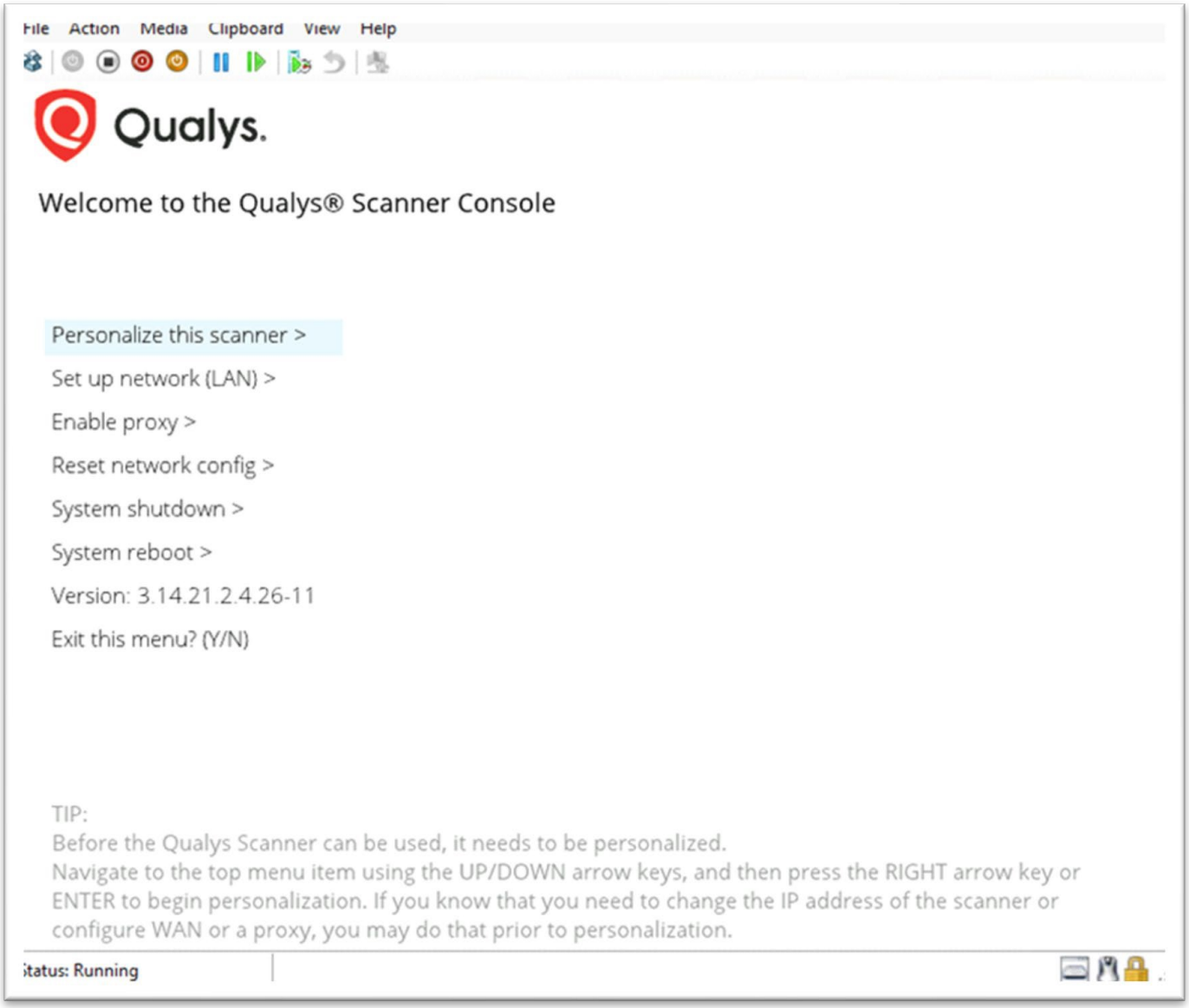
1. Start the virtual scanner machine.
2. Unzip the downloaded file: qVSA.i386-2.4.26-11.vhd.zip to obtain the virtual hard disk file: qVSA.i386-2.4.26-11.vhd.zip.
3. Log in to the Hyper-V server.
4. Navigate to **Manager** | **Hyper-V Manager** | **add a new Virtual Machine**.
5. Provide a name for the scanner.
6. Configure the memory. (Recommended is 2048 MB or more.)
7. Configure the networking as appropriate so the network adapter on the scanner can use a virtual network for communication.
8. For the virtual hard disk configuration, select **Use an existing virtual hard disk** and provide the location of the .vhd file obtained from the download .zip file.

9. Click **Next** and then **Finish**.

Personalize the Virtual Scanner

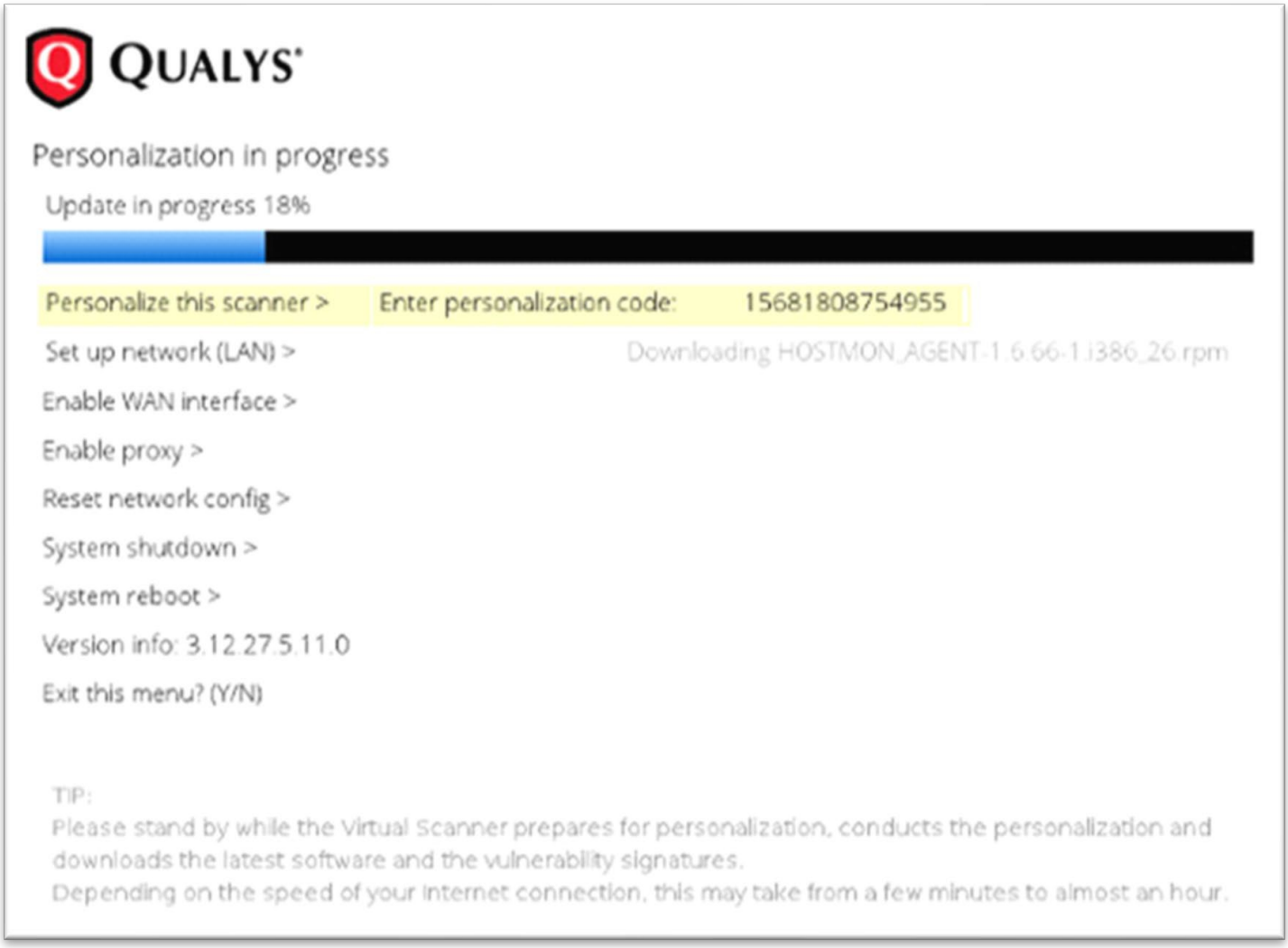
The virtual scanner will use Dynamic Host Configuration Protocol (DHCP) without proxy configuration, unless custom settings are required. (For custom configuration: **Set up network (LAN)**.)

1. Click **Personalize this scanner**.



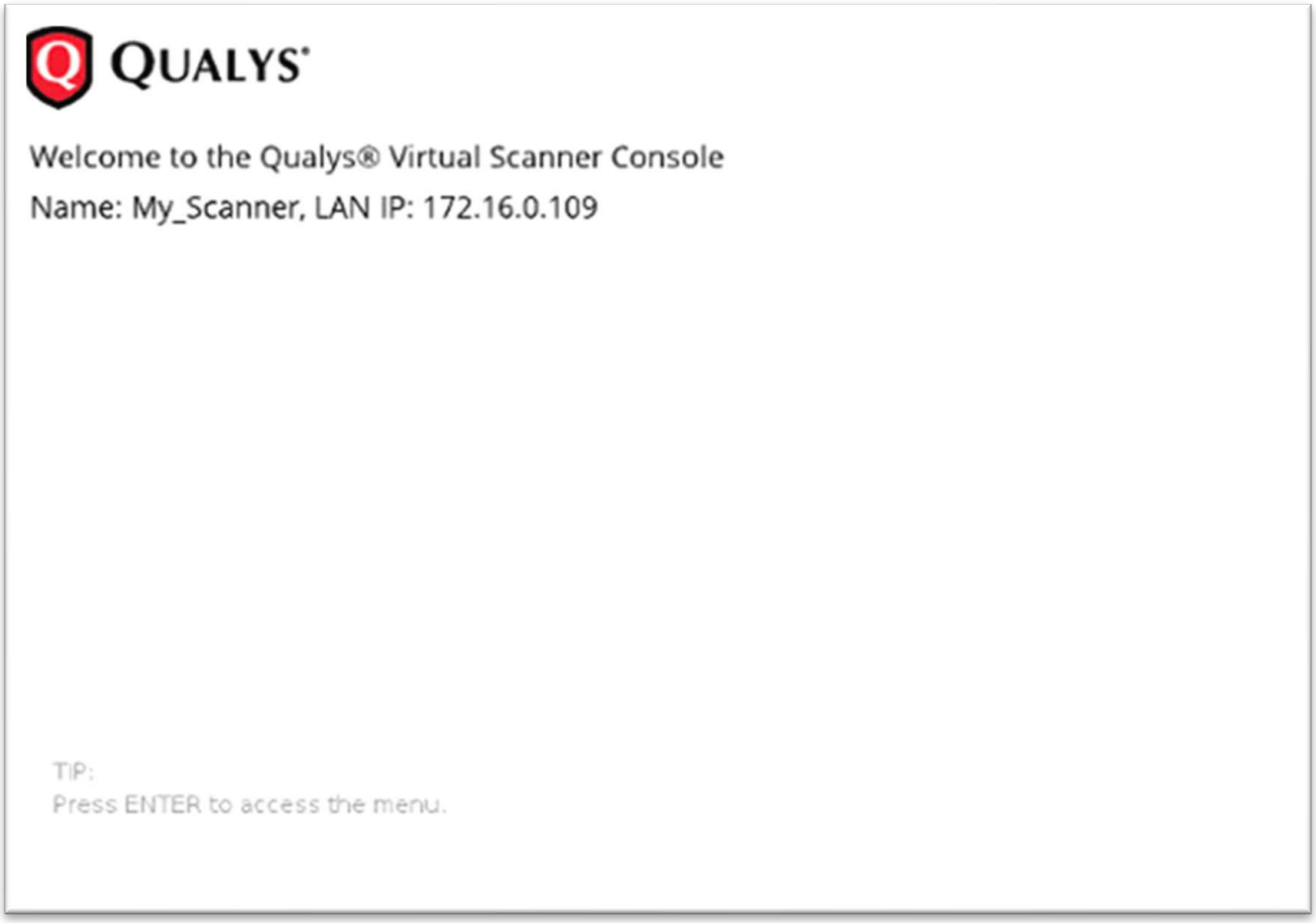
2. Enter the **personalization code**.

One activation code is used to activate one virtual scanner. After enteringthe code, the activation process begins, and the service reports the progress. Activation may take a few minutes to complete.



Upon success, the scanner's friendly name and IP address appear, and the scanner is ready to be used for scanning.

1. Press **Enter** to go to the main menu.



Auto Change Passwords After Each Scan

You can leverage Secret Server's Check Out feature to ensure passwords used for authenticated scanning are changed after use. This is an important measure in protecting privileged accounts from "Pass-the- Hash" attacks, because it means that the password hash stored in each machine from use during scanning will no longer be valid once the password has been changed in Active Directory. Check Out with Change Password on Check In means the password for your domain account will be changed automatically after scanning is complete.

> **Note:** Check Out requires Enterprise or Enterprise Plus edition.

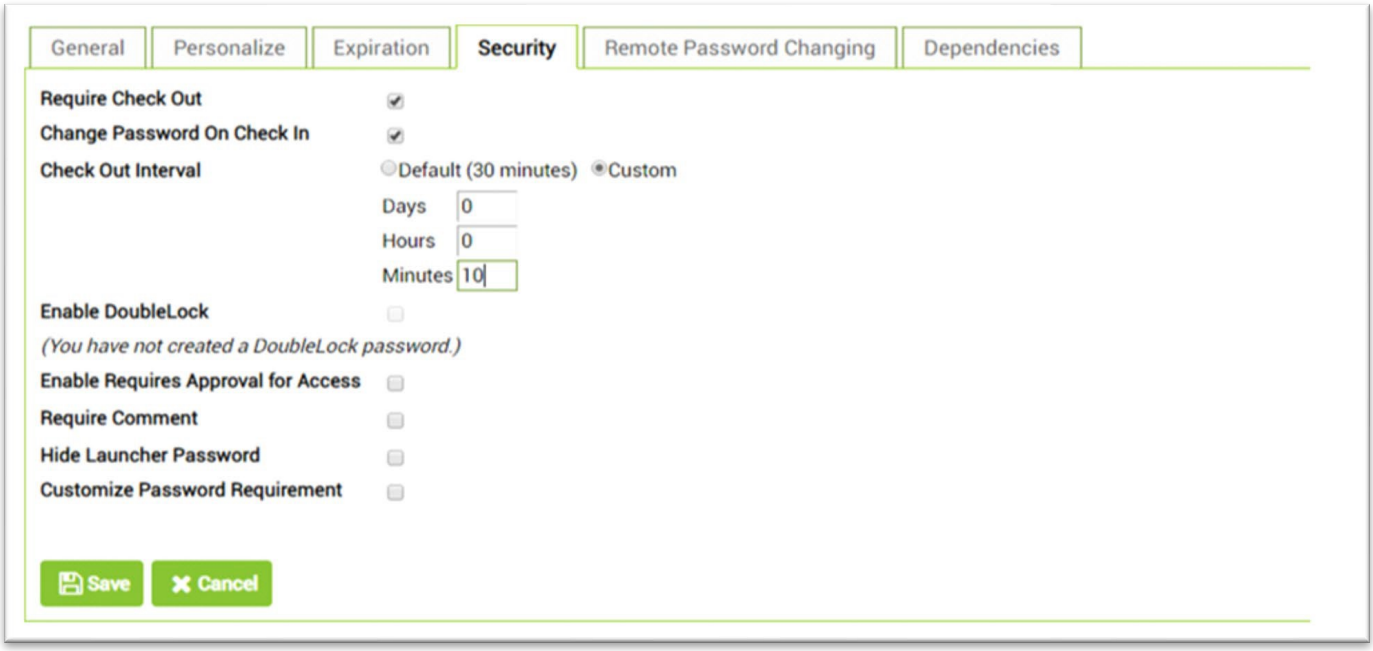
To verify Password Changing on Check In is enabled, perform the following:

1. From the Admin menu, navigate to **Remote Password Changing**.
2. If Enable Password Changing on Check In is set to **No** or doesn't appear, click **Edit** to ensure the Enable Remote Password Changing and Enable Password Changing on Check In check boxes are both selected.
3. Click **Save**.

Configure a Secret for Check Out

Enable Check Out with Change Password on Check In to ensure the password for privileged accounts is changed after each use.

1. Navigate to your privileged account Secret and click **View**.
2. Click the **Security | Edit**.
3. Select the **Require Check Out** check box, and then select the **Change Password On Check In**.
4. Click **Custom** and enter the estimated time period for authenticated scanning to complete.
5. Click **Save**.



Troubleshooting

If the authenticated scan is failing, or isn't finding the Secret there are a few settings to check to make sure the integration is configured correctly:

Are web services enabled?

Web services in Secret Server must be enabled for this integration to function. Check the web services status from the **Configuration** section of the **Administration** menu.

Is Secret Server accessible?

The URL defined when setting up the authentication vault must be accessible by the Qualys appliance. Ensure proper DNS settings and firewall rules are in place to allow access from the appliance to Secret Server web services.

Are permissions and security settings configured correctly?

The Qualys user needs at least read access to the Secret.

> **Note:** If certain security settings are in place, such as Approval for Access, and Hide Launcher Password the user will need elevated access. Other settings such as DoubleLock will prevent the Qualys account from accessing the Secret. If a Secret is configured for Check Out, Secret Server will automatically check out the Secret to the Qualys user.

Check audit trails

To help determine whether the Qualys account is correctly accessing Secret Server and the stored credential, check the audit on the Qualys user to see if it is logging in successfully from the appliance and view the audit on the Secret to see if there are view records which will show that the Secret was actually accessed.

BEST PRACTICES

Folders

Permissions in Secret Server are similar to NTFS, and are assigned at the Secret or folder level. To make maintenance simple going forward, a standard model is to place all the credentials needed for scanning in a separate folder that the security team can manage. Give the Qualys Secret Server account read-only access to the folder and it will automatically gain access to new credentials added to the folder for scanning. In the example screenshots in this guide, the scanadmin Secret is placed in the Scan Credentials folder, which an admin and the Qualys user account have access to.

Groups

If there are multiple authentication vaults set up in Qualys with separate user accounts for accessing Secret Server, it is recommended to assign all of them to a local Secret Server group to simplify permission management moving forward.

SSL

Ensure that Secret Server is configured to force connections over HTTPS. This can be verified by clicking the **Security Hardening** tab from the **Reports** section of Secret Server.

IP Address Restrictions

Secret Server allows whitelisting of IP address ranges to restrict where accounts can log in from. For the Qualys Secret Server account, it is recommended to only allow it access from the Qualys agent appliance to help prevent misuse of the account.

Event Subscriptions / SIEM

Configure email or SIEM alerts to get notified if the Qualys account gets locked out of fails to login to the vault. This can indicate the password was updated in Active Directory or Secret Server without updating the Qualys Authentication Vault record. For more information on alerts refer to the user guide and the syslog integration guide.

Introduction

The integration between Thycotic Secret Server and Rapid7 is created and maintained by Rapid7. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Nexpose and InsightVM

Nexpose and InsightVM are assessment solutions for your physical, virtual, mobile, and cloud environments. Their dynamic discovery integrates with your existing infrastructure to find any security gaps. Thycotic's industry-leading integration with Rapid7 Nexpose provides a revolutionary way to scan and mitigate risk in virtual networks.

Thycotic Secret Server (SS) is an on-premises Web-based password vault used throughout the world to help organizations properly manage privileged account passwords. SS allows users to control access and automate password changes for a variety of enterprise resources, such as servers, databases, network devices, and applications. SS features auditing throughout the application and role-based access control (RBAC) on all its information and features. Organizations can easily deploy SS to ensure security, reduce labor costs, adopt password best practices, and satisfy audit requirements.

Thycotic

Thycotic IT security and password management solutions empower companies to remove the complexities of proper access control and management of privileged accounts. An Inc. 5000 company, Thycotic is trusted by more than 3,000 organizations worldwide—including Fortune 500 members, enterprises, government agencies, technology firms, universities, non-profits and managed service providers. To learn more, please visit thycotic.com.

Rapid7

Rapid7 cyber security analytics software and services reduce threat exposure and detect compromise for 3,900 organizations across 90 countries, including 30% of the Fortune 1000. They help manage risk, stop attacks faster, and systematically advance your security program with comprehensive real-time data collection, advanced correlation, and insight into attacker techniques. With their innovative technology and trusted expertise, they prepare you for everything from drive-by download attacks to advanced threats, from the endpoint to the cloud.

Integration Requirements

Pre-requisites

Ensure you have an API user account that has, at minimum, view permissions on the Secrets with which you are going to integrate. The API account has been associated to the built-in user role and has explicitly only been assigned view access to the Secrets with which we integrate. For Rapid7, the minimum permissions required for the Nexpose related API account has not yet been isolated. Currently, customers are providing "admin" site specific Rapid7 permissions.

Known Constraints and Limitations

1. The integration only works with IP address matching and short name (NetBIOS) hostnames. You cannot currently supply FQDN values with the integration, which implies the scanner nodes are required to resolve short names appropriately.
2. The integration cannot handle specifying CIDR ranges in the site configuration within Nexpose/Insight VM. You must have 1:1 mapping.
3. If a Secret is checked out, an error message appears while running the script stating it is unable to access that particular Secret/asset for integration. You will receive an indication of which user currently has the Secret checked out. This asset will not be added within the authentication section of the site configuration in Rapid7.
4. If utilizing Ruby on Windows, the highest version known to be working with integration is 2.5.7-1 with both hostnames and IP addresses. The latest version of Ruby for Windows works with IP address matching only.

Configuration

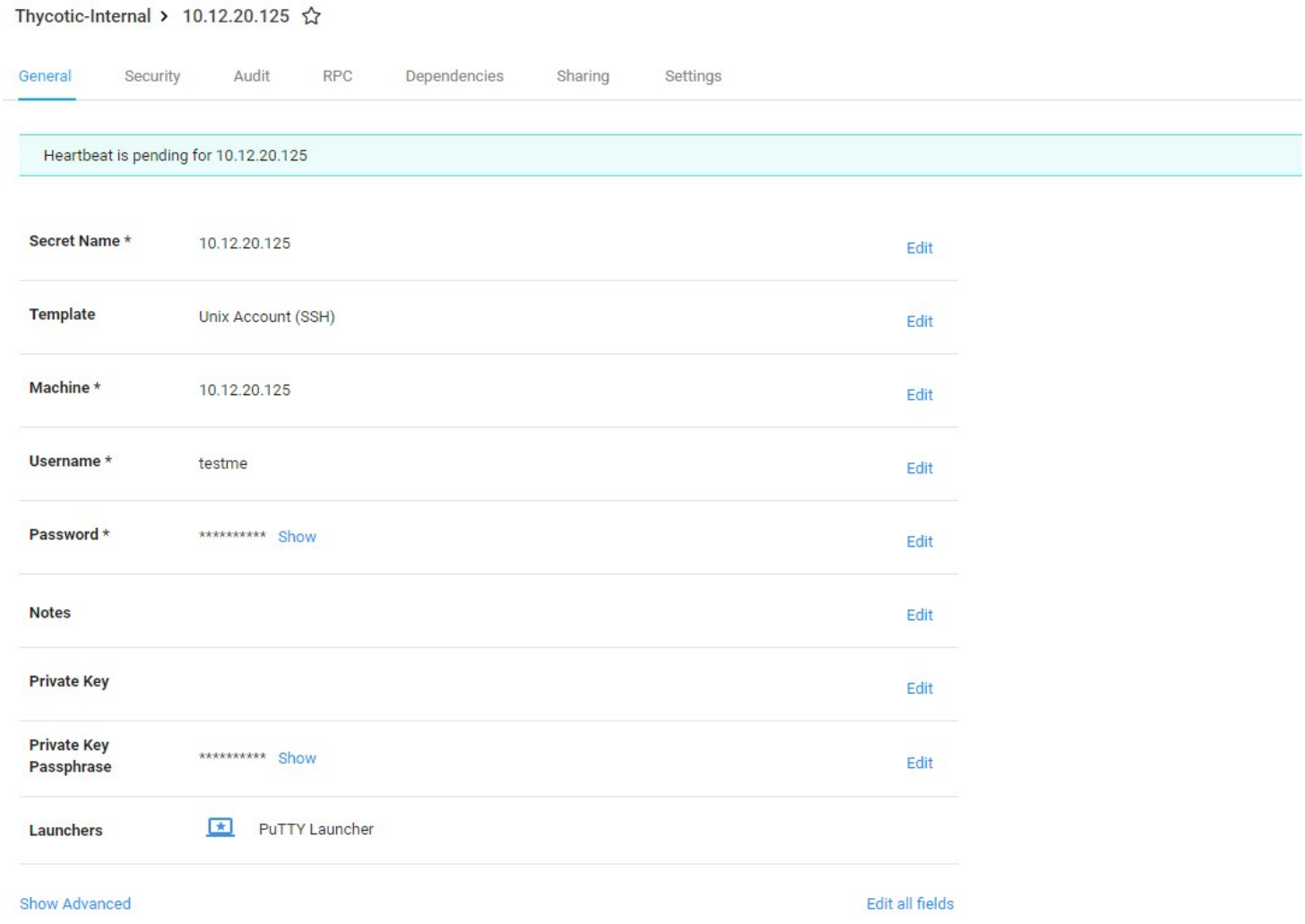
Please see the steps below to properly configure Rapid7 with Secret Server:

- Configure the Environment.
- Configure the Ruby Interpreter.
- Configure and run the Script.
- Automate.

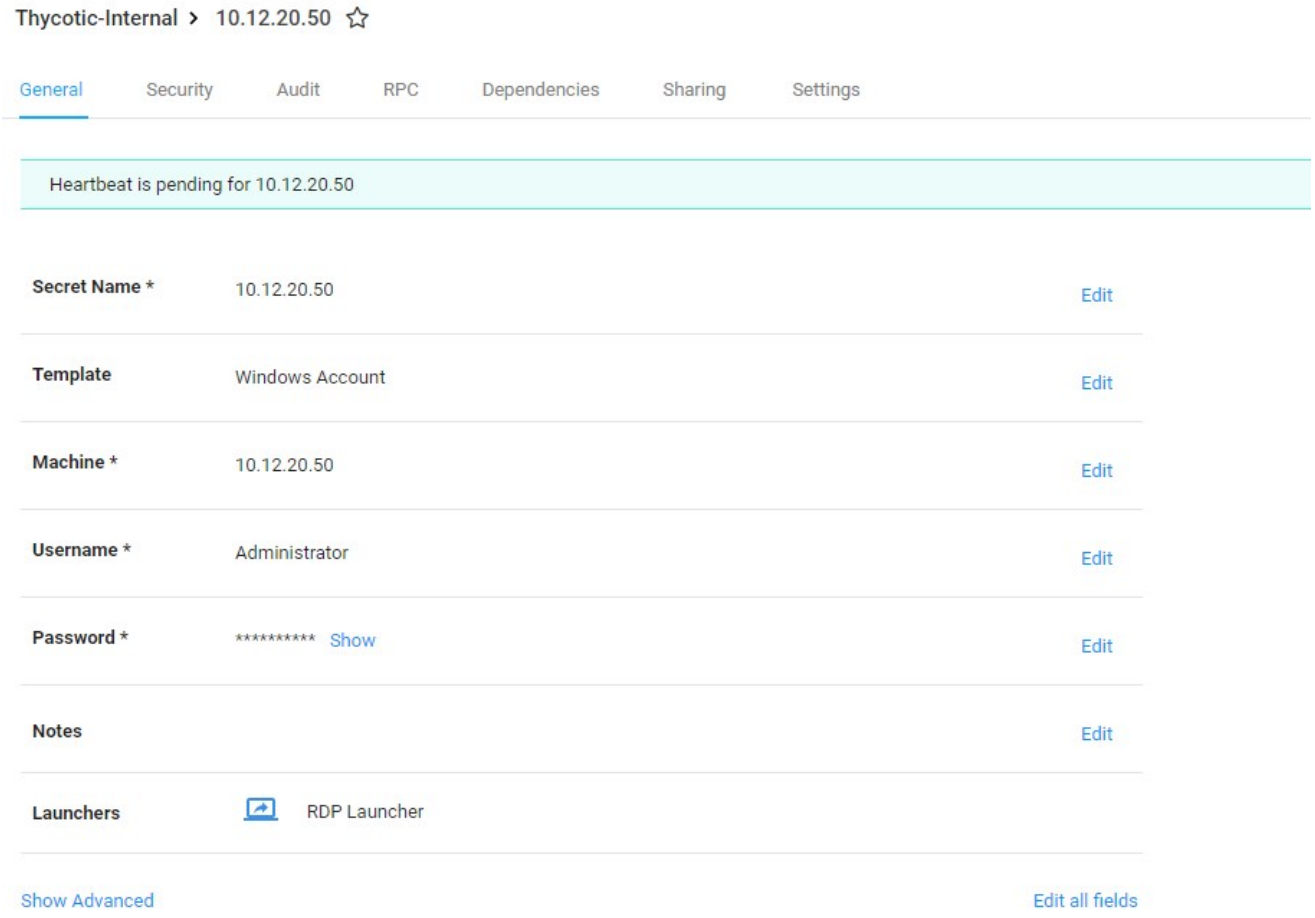
Configure the Environment

1. Enable the SS Web Service API in the user interface:
2. Navigate to **Admin I Configuration I General tab**.
3. Set the Enable Webservices option to **Yes**.
4. Ensure the assets to be managed have the following characteristics:
    - The Secret name must be the same as the IP address or host name within the Secret itself. It is required that the Machine field is explicitly an IP address or short hostname value. To keep the examples simple, the secrets are named the same as the machine values.

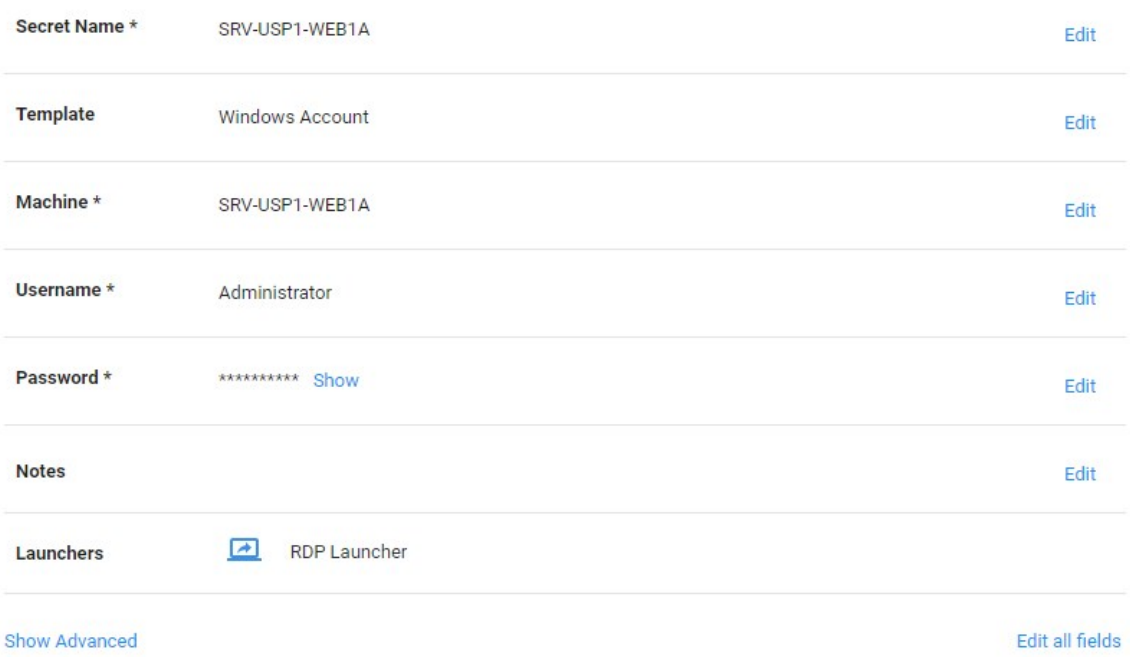
Linux Template example using IP address:



Windows Template example using IP address:



Windows Template example using Hostname:



Note: The Secret template must be one of the defaults: Unix Account (SSH) for Unix/Linux systems and Windows Account for CIFS/Windows systems.

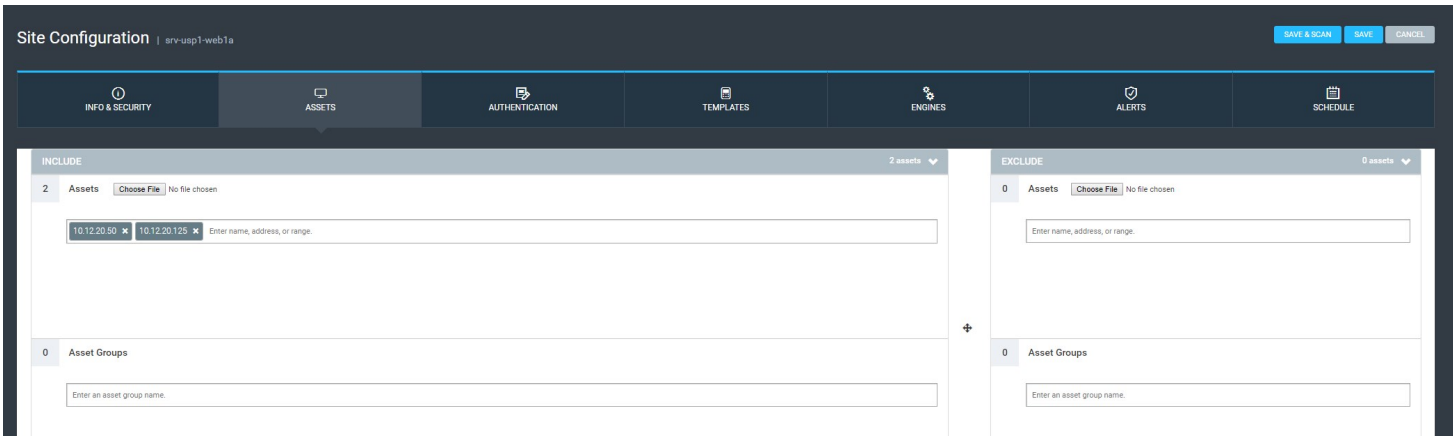
1. Create sites in InsightVM or Nexpose with assets to work with Secret Server. You can manually create the assets in InsightVM or Nexpose or import them from a list.

   Note: Please be aware that you cannot enter a CIDR range. You must have individual entries for each asset to be included in the Rapid7 integration.

Configure the Ruby Interpreter

Configure the Thycotic Gem to import Thycotic credentials into the Nexpose vulnerability management system. The script you will use requires a Ruby interpreter and must be installed on the system where it is going to run.

1. View the options for installing Ruby on various platforms and choose the most appropriate version.
2. Once installed, this Ruby Gem must also be installed: nexpose_thycotic 0.2.0.
3. This can be either manually downloaded or through the GEM application repositories: gem install nexpose_thycotic.



Configure and Run the Script

Once all dependencies have been installed, configure the script:

1. Open the Nexpose_Thycotic.config file located in the Ruby installation folder. For example:C:\\Ruby25-x64\\lib\\ruby\\gems\\2.5.0\\gems\\Nexpose_Thycotic-02.0\\lib\\Nexpose_Thycotic\\config.

Configure Secret Server

Set up the following options for the user who is running the Gem:

- **Thycotic url:** The IP address for the Secret Server instance with WSDL. For example: https://127.0.0.1/SecretServer/webservices/sswebservice.asmx?wsdl.
- **Thycotic username:** The username configured in Secret Server.
- **Thycotic passsword:** The password for that Secret Server user.
- **Comment:** The comment used when retrieving each password.

Configure Nexpose settings

- **Nexpose url:** The URL/IP For the Nexpose instance. For example: https://192.168.0.1/.
- **Nexpose username:** The username in Nexpose.
- **Nexpose password:** The password for the user in Nexpose.
- **Nexpose port:** Set if Nexpose was installed with a different port than the default 3780.

Configure additional settings

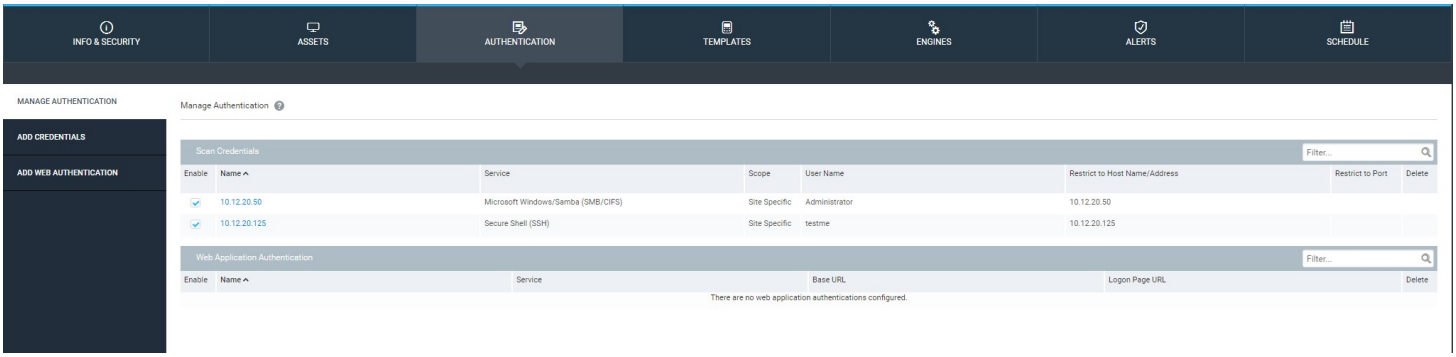
- **Sites:** Sites to manage. There can be more than one site.
- **Clear creds:** Set to True to delete all pre-existing credentials when importing from SS.
- **Logging enabled:** Enables logging to the log directory.
- **Log level:** Default is info but can be set to debug, warn, or error.
- **Log console:** Determines if log messages are output to the console.

Step Four: Run the Script

1. Run the script from the command line: ruby Nexpose_Thycotic.
2. The scriptruns and performs the queries.
3. On windows systems, you can also run the Nexpose_tycotic command from the C:\\Ruby25-x64\\bin directory.

The credentials will then be available for each asset in your newly existing site(s). This can be verified by the following lines in the script output and in the Authentication Tab of the site for each asset.

```
Logging enabled at level <info>
Starting integration.
Logging into Thycotic at https://ps01.thycotic.blue/SecretServer/webservices/sswebservice.asmx?wsdl
Processing sites
Processing site 1
Discovered 0 hosts and 1 IPs.
Clearing existing credentials.
Getting credentials for 10.12.20.50
Finished processing 1
```


INFO & SECURITY
ASSETS
AUTHENTICATION
TEMPLATES
ENGINES
ALERTS
SCHEDULE
MANAGE AUTHENTICATION
ADD CREDENTIALS
ADD WEB AUTHENTICATION
Manage Authentication
Scan Credentials
Filter...
Enable
Name
Service
Scope
User Name
Restrict to Host Name/Address
Restrict to Port
Delete
10.12.20.50
Microsoft Windows/Samba (SMB/CIFS)
Site Specific
Administrator
10.12.20.50
10.12.20.125
Secure Shell (SSH)
Site Specific
testme
10.12.20.125
Web Application Authentication
Filter...
Enable
Name
Service
Base URL
Logon Page URL
Delete
There are no web application authentications configured.

Automate

There is a batch file in C:\\Ruby25-x64\\bin\\nexpose_thycotic.bat.

You may configure as a scheduled task to run the script at specific times.

> **Note:** Be mindful of secrets with password changing configured and aligning these types of secrets with scan times appropriately.

Troubleshooting

The most common script errors are configuration based. Such as, users without permission to update sites or query credentials from Secret Server.

To begin Troubleshooting

1. Ensure secrets have the same name in Nexpose and in SS. (IP address or short name hostname only for secret name and machine name.)
2. Ensure the cited Nexpose user can save sites and start scans.

```
---
# This configuration file defines all the particular options
necessary to run the service.
# Fields marked (M) are mandatory.
#
# Service options:
:options:
  # (M) Enables logging to the log directory.
  :logging_enabled: true
  # (M) Sets the log level threshold for output.
  :log_level: info
  # (M) Enables logging output to console
  :log_console: true
  # Filters to specific sites one per line, leave empty to
generate for all sites.
  :sites:
  - '1'
  # Delete existing credentials for the site before importing
  :clear_creds: true
:nexpose_options:
  # (M) Nexpose IP address
  :nexpose_url: 10.12.20.175
  # (M) Nexpose username
  :nexpose_username: <%= SymmetricEncryption.try_decrypt
"QEVuQwEAgLHvB6NzjmNDRsARydwzcA==" %>
  # (M) Nexpose password
  :nexpose_password: <%= SymmetricEncryption.try_decrypt
"QEVuQwEAUJLGUd4aYLB3X7gX3FSqNg==" %>
  # (M) The port Nexpose listens on. Default is 3780
  :nexpose_port: 3780
:thycotic_options:
  # (M) Thycotic Instance IP address with WSDL
  :thycotic_url:
https://ps01.thycotic.blue/SecretServer/webservices/sswebservice
.asmx?wsdl
  # (M) The Thycotic username
  :thycotic_username: <%= SymmetricEncryption.try_decrypt
"QEVuQwEAF2zy9UXpGR0qprgoScHcVQ==" %>
  # (M) The password for the above user
  :thycotic_password: <%= SymmetricEncryption.try_decrypt
"QEVuQwEANOCUAzdnTmXXphnsUtRkgek/+gIEo1RcmSXTC1UYl84=" %>
  # (M) The comment used when retrieving each password
  :comment: 'Retrieved via Thycotic gem.'
:encryption_options:
# (M) Path to the encryption.config file
  :directory: ../../config/encryption.config
```

1. The encryption.config file shown above does not reference Thycotic's encryption.config file, nor does it need copied to the location specified for this integration to work.
2. Ensure the nexpose_url reflects the actual IP address assigned to the network adapter attached to the system running the script and not a localhost value.
3. If issues still exist, email support@rapid7.com.

Introduction

The integration between Thycotic Secret Server and Royal TS is created and maintained by Royal TS. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Thycotic

Thycotic IT security and password management solutions empower companies to remove the complexities of proper access control and management of privileged accounts. An Inc. 5000 company, Thycotic is trusted by more than 3,000 organizations worldwide—including Fortune 500 members, enterprises, government agencies, technology firms, universities, non-profits and managed service providers. To learn more, please visit thycotic.com.

Secret Server

Thycotic Secret Server (SS) is an on-premises Web-based password vault used throughout the world to help organizations properly manage privileged account passwords. SS allows users to control access and automate password changes for a variety of enterprise resources, such as servers, databases, network devices, and applications. SS features auditing throughout the application and role-based access control (RBAC) on all of its information and features. Organizations can easily deploy SS to ensure security, reduce labor costs, adopt password best practices, and satisfy audit requirements.

Royal TS

Royal TS (Terminal Services), called RTS in this document, provides powerful, easy and secure access to remote systems for server admins, system engineers, developers and others who need to access remote systems using different protocols. RTS supports RDP, VNC, SSH, HTTP/S, and more. It is available for Windows, OS X, iOS and Android, and RTS documents open on all those platforms. See http://www.royalts.com or more information.

Royal TS and Secret Server Integration

SS randomizes and stores passwords for accounts on target systems on a regular recurring basis. Because these passwords are stored and managed in a vault, they can be retrieved via a SOAP or REST Web service. RTS integrates with SS via a PowerShell script, provided by RTS. Specifically, SS integrates with the RTS dynamic folder feature.

Connection Management

With RTS, you can organize and manage all your connections to your remote systems. The following connection types are currently available and supported:

- External Application
- File Transfer
- Hyper-V Management
- Performance View
- PowerShell Connection
- Remote Desktop
- TeamViewer
- Terminal
- Terminal Services Management
- VMware
- VNC
- Web Page
- Windows Events View
- Windows Processes
- Windows Services

With RTS, you can check your event logs, restart services, or manage Hyper-V and VMware virtual machines right from your mobile device without connecting to remote desktops.

Getting Started with Royal TS

Application Features

User Interface

- Open RTS/X documents using any cloud provider or other application which can open files
- Copy credentials or any other property value to the clipboard via long press

Credential Management

- Assign a single credential to multiple connections or folder with inheritance support
- Use autofill for login forms and web page connections

Team Sharing

- Share safe and secure connections, keeping your personal credentials private
- Assign credentials by name and allow users to specify their own credentials for shared connections

Documents and Management

- Use multiple document types
- Auto start documents
- Apply encryption and password protection

Royal Server

- Manage connections
- Manage Windows Server from iOS as if it were native to iOS

Configuration

Please review the following steps to setup and configure Royal TS for Secret Server:

- Prerequisites for Integration.
- Configure RTS for Integration.
- Configure Secret Server for Integration.
- Create an RDP Credential in Secret Server.

Royal TS Dynamic Folders

A RTS dynamic folder allows you to import data from external sources. All imported objects are read-only, but you can assign credential objects to other objects outside of dynamic folders.

Dynamic folders have several properties:

- Display Name: Required. A human-readable display name for an object.
- Color. Click the color picker button in the Display Name text box to select a color. In the User Interface settings you can configure RTS to show the color in the navigation tree, the connection tab, or as a connection border.
- Icon: Click the icon picker button next in the Display Name text box to select and assign a custom icon to the object.
- Description: An optional description for the object.

RoyalJSON and Dynamic Folders

RoyalJSON (rJSON) is a unidirectional, human-friendly data format for importing data from external sources into RTS/X. It provides users an easy, yet powerful, way to access data stored outside of RTS/X into the application.

You can get dynamic folder samples, as well as documentation, for creating RoyalJson at our Dynamic Folder Toolbox repository on GitHub.

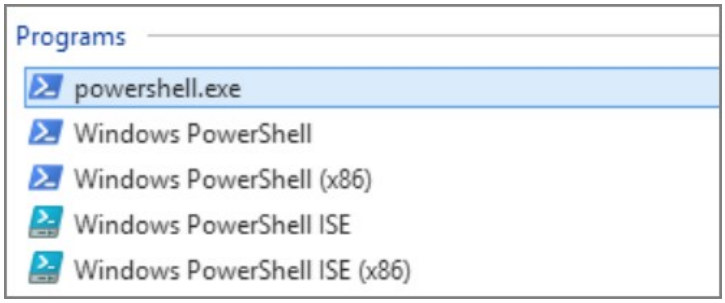
Prerequisites for Integration

1. Navigate to https://www.royalapps.com/server/main/download and **download** RTS for Windows 5.0.
2. Install **RTS** for Windows.

   Note: You can download, install, and use RTS products free without any time limit, license key, or registration. This allows you to get started quickly. If have a small environment, you can continue using our products free of charge in "Shareware Mode."

   Note: We recommend that you *do not* associate RTS with the Remote Desktop connection file (.rdp) extension.

   Note: It can take RTS for Windows several minutes to install and optimize itself.
3. Ensure Windows PowerShell ISE (Integrated Scripting Environment) is on your Windows installation:
4. Click your Windows **Start** button.
5. Type PowerShell in the search text box:



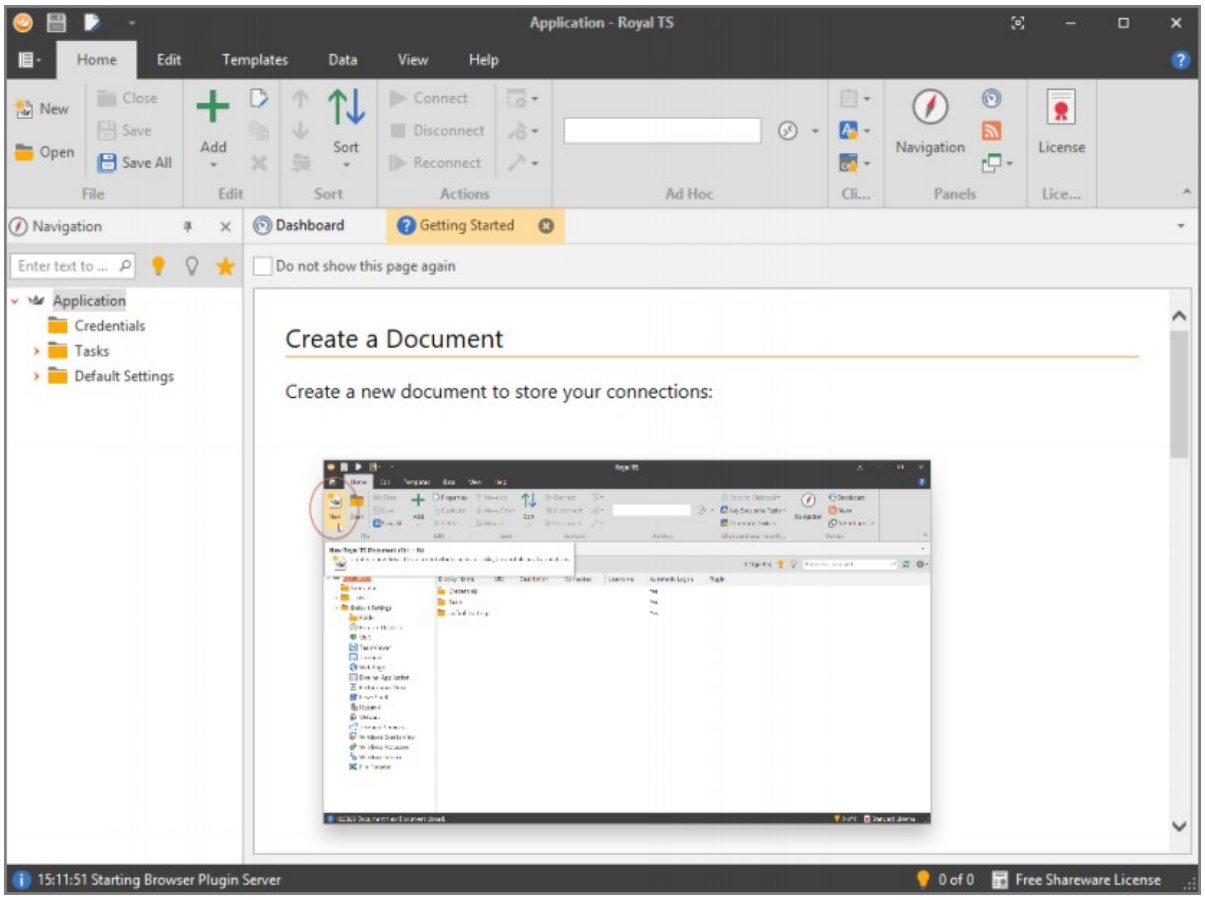
6. Ensure the Windows PowerShell ISE entry appears as a choice.

   Important: The script that you will run is designed for PowerShell 6, so please ensure you have at least this version or newer.

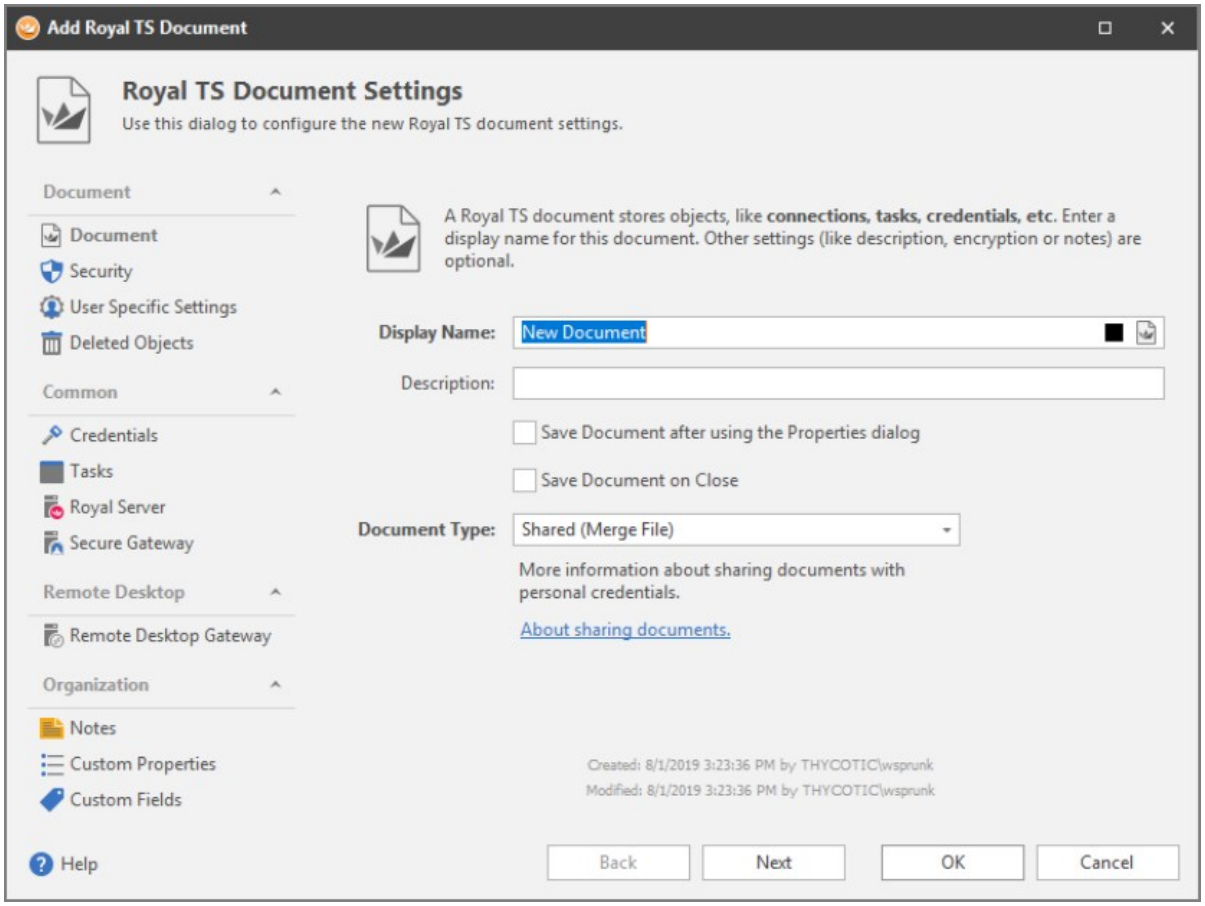
   Note: Each version of Windows and Windows Server includes a version of Windows PowerShell and the ISE. If you suspect yours is out of date, you can upgrade to the latest available by installing the latest Windows Management Framework (WMF).

Configure RTS for Integration

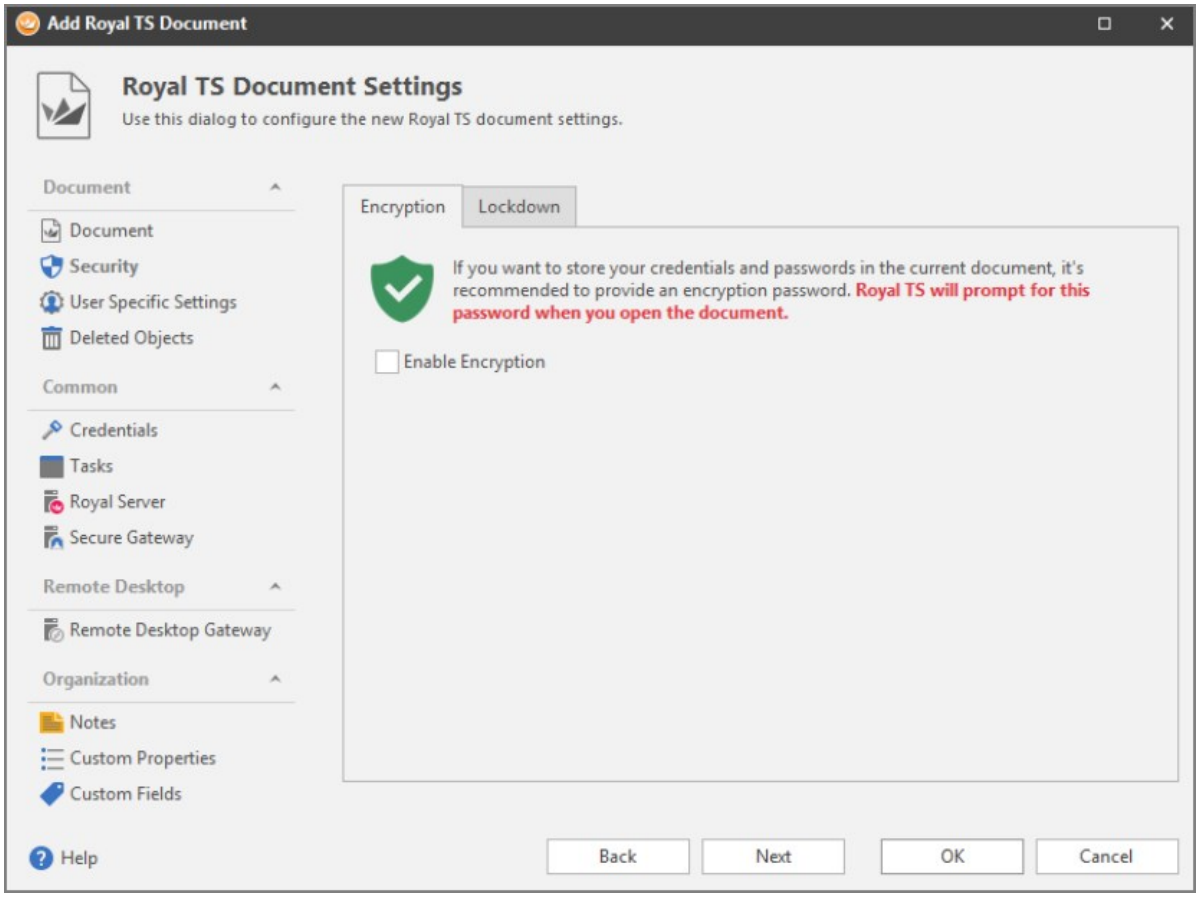
1. Start **RTS**.



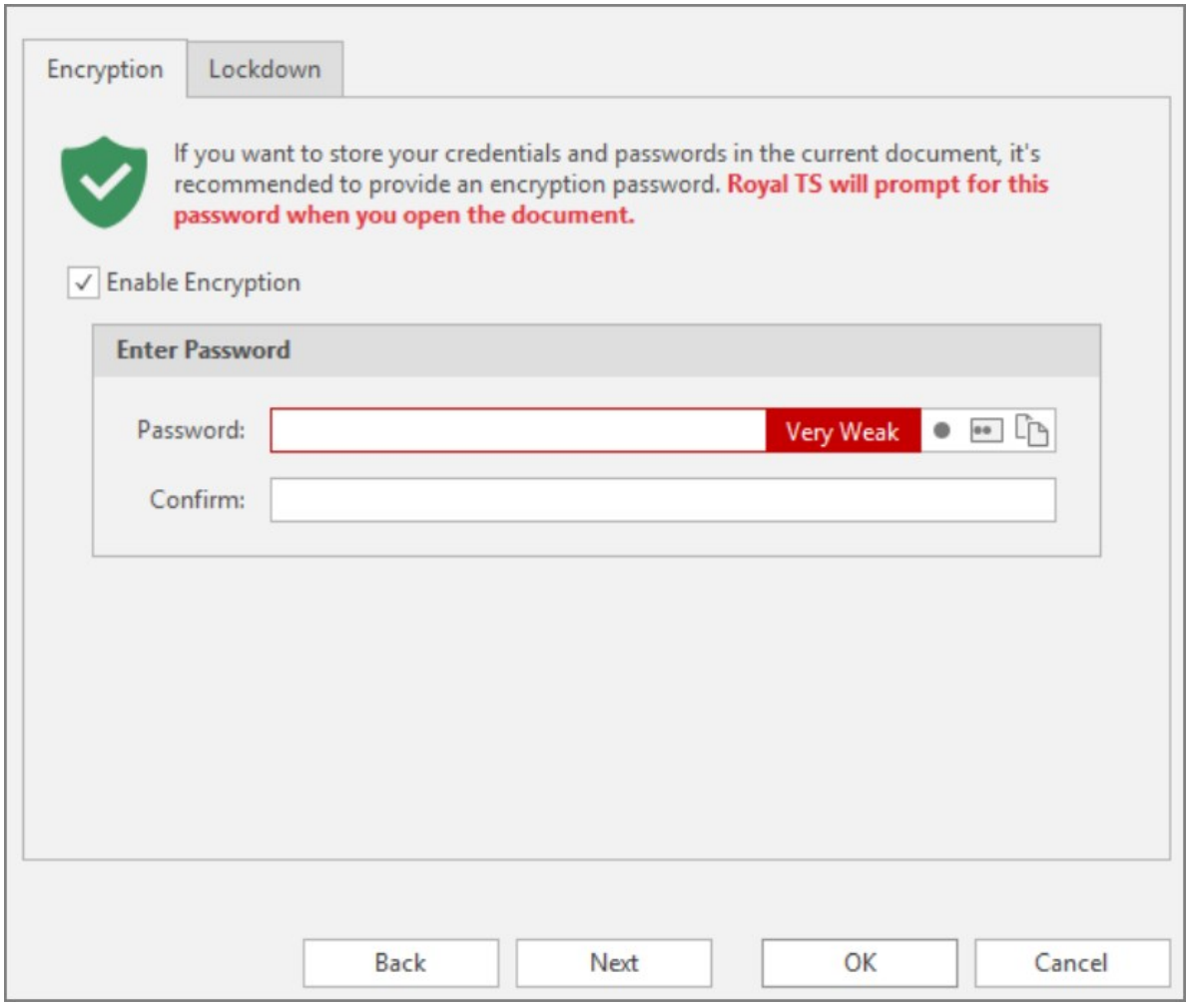
2. Click the **New** button. The Add Royal TS Document wizard appears:



3. Type a name for the document in the **Display Name** text box.
4. Click the **Next** button:



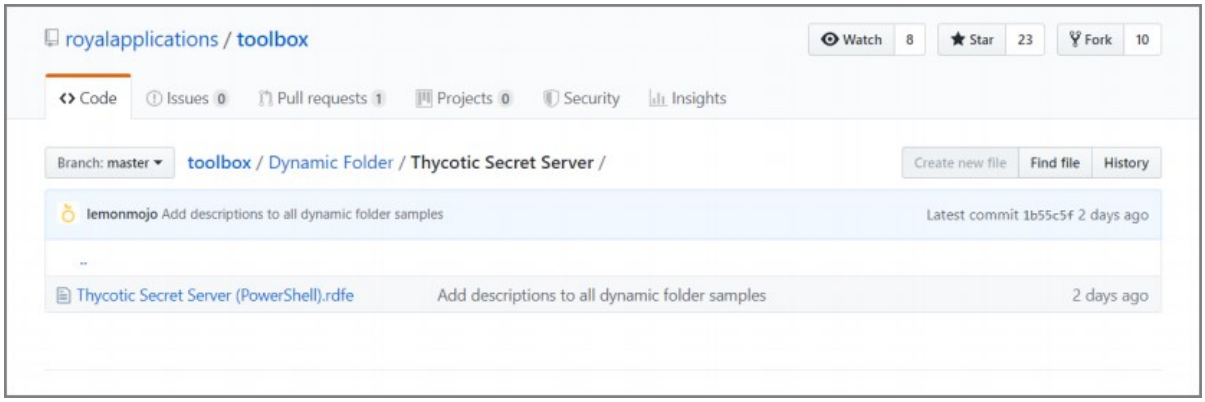
5. Click to enable the **Enable Encryption** check box:



6. Type the desired password in the **Password** and **Confirm** text boxes. You can also click the **Generate Password** button to the right to automatically create one.

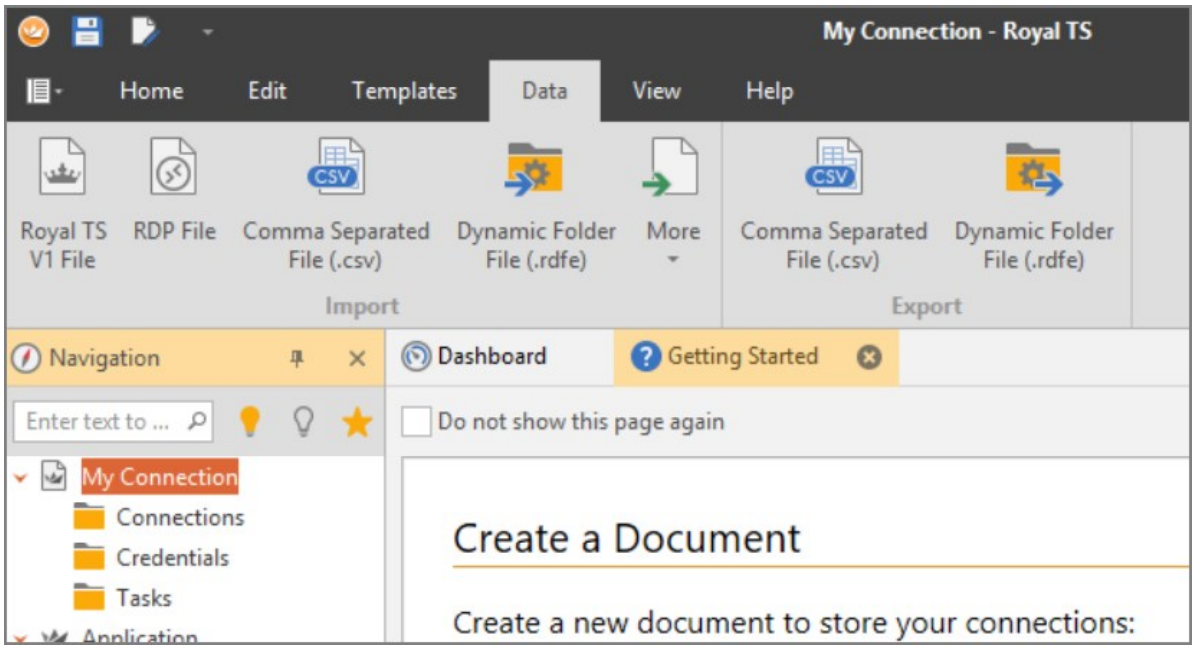
7. Click the **OK** button. The wizard closes and your new connection appears as a node in the tree in the left pane.

8. Navigate to the dynamic folder PowerShell file for Secret Server at https://github.com/royalapplications/toolbox/tree/master/Dynamic%20Folder/Thycotic%20Secret%20Server:



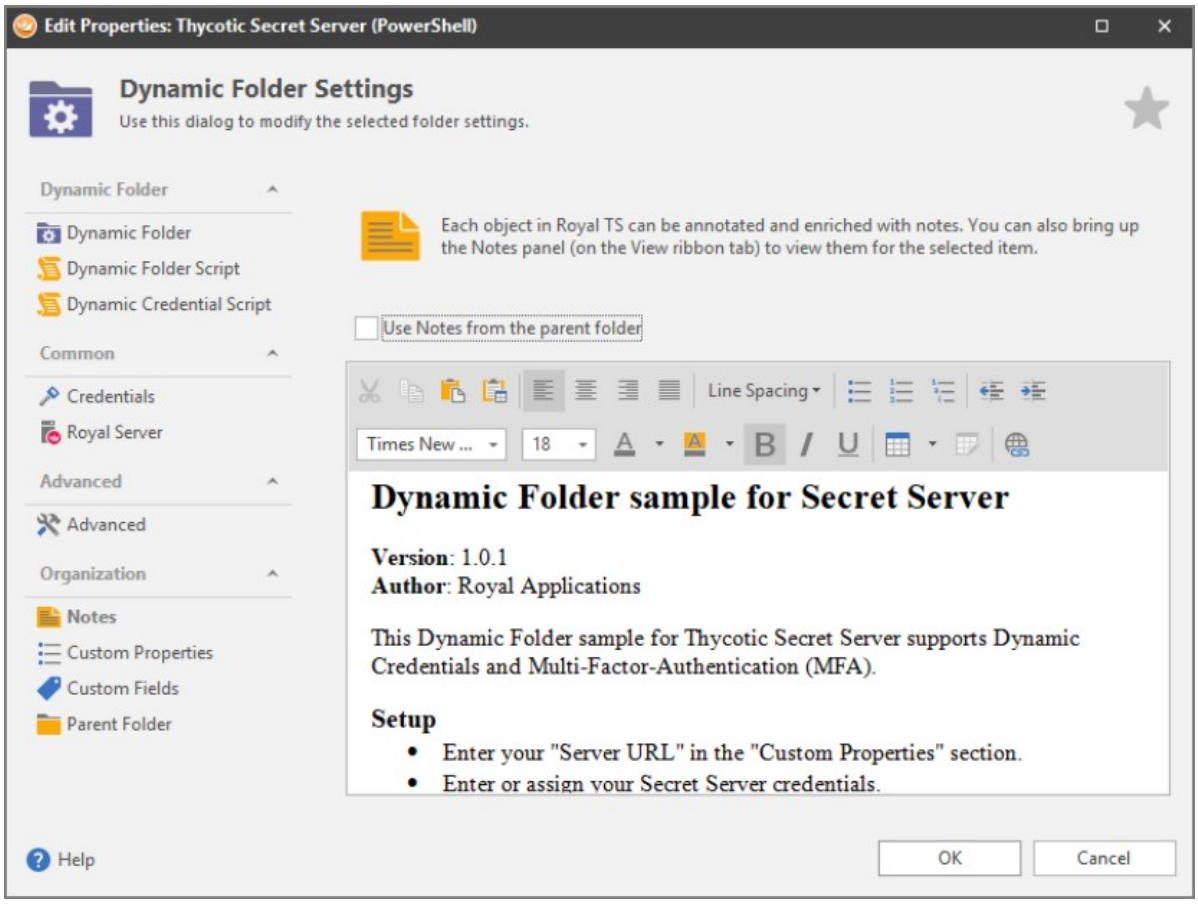
9. Click the **Thycotic Secret Server (PowerShell).rdfe** link to open the file.

10. Right-click the **Raw** button and select **Save link as...** or **Save linked content as...** to save the PowerShell .rdfe file to a location of your choice. It probably defaults to your Downloads folder.
11. Return to RTS.
12. Click the new **Connection** node in the navigation tree.
13. Click the **Data** tab:

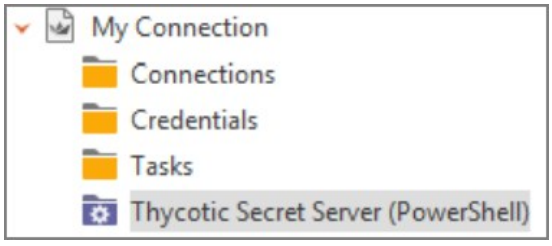


14. Click the **Dynamic Folder File** button in the **Import** section.

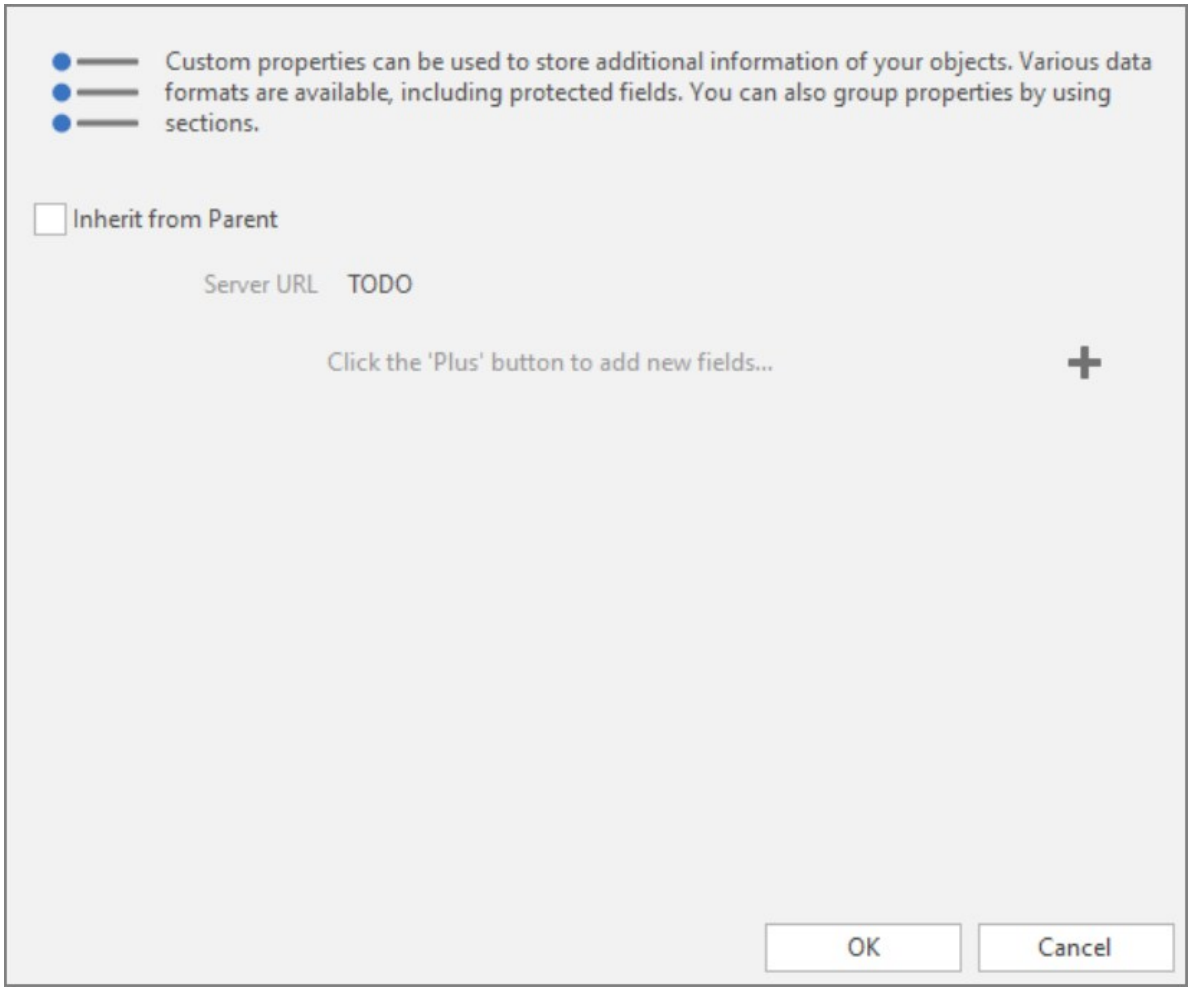
15. Locate the file you downloaded.
16. Click the **Open** button to import the file. A confirmation popup and then the Edit Properties popup appears:



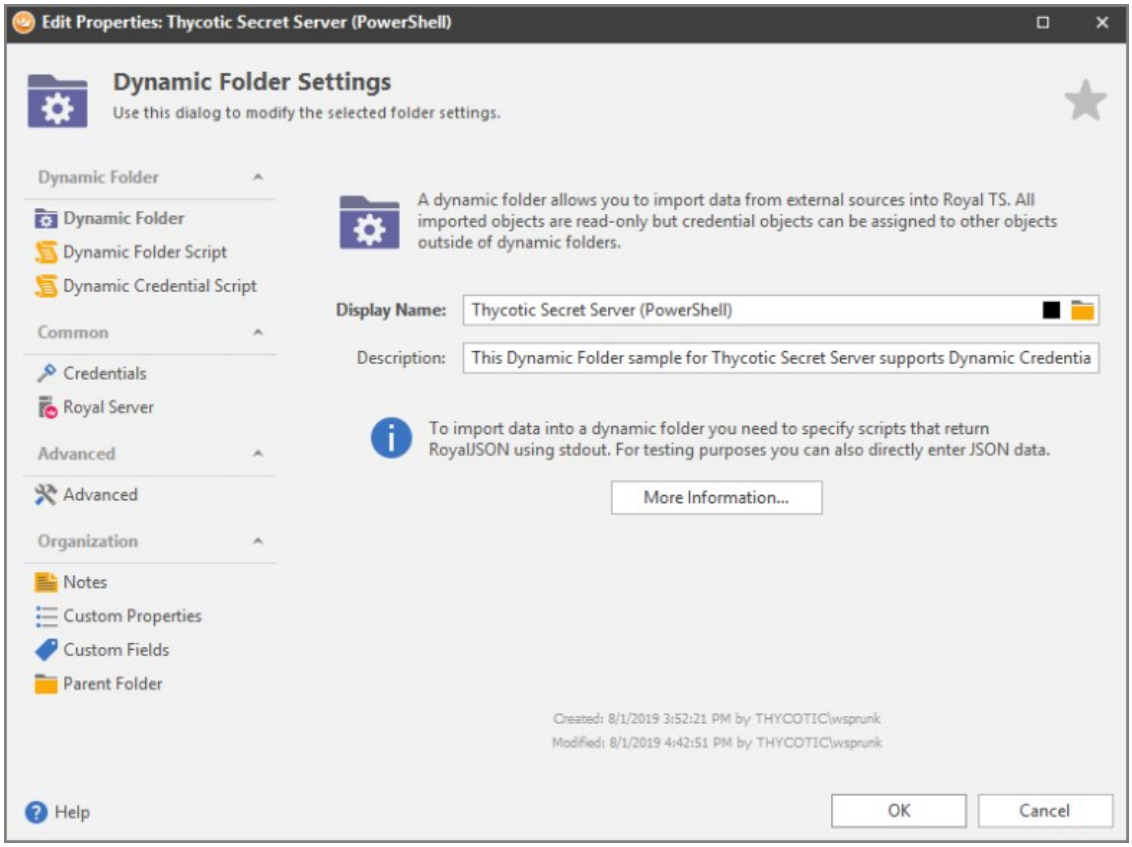
You cannot see it yet, but a new dynamic folder appears under your connections



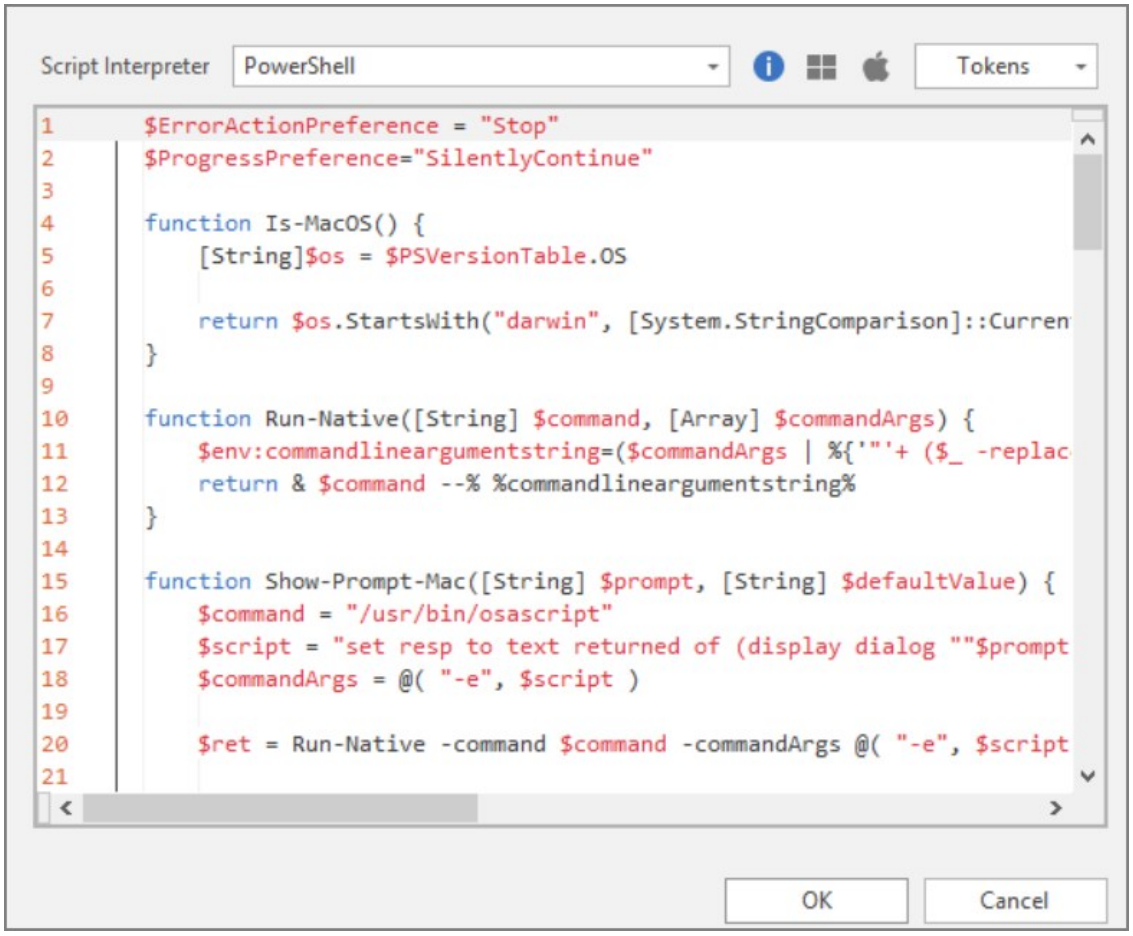
17. Click the Custom Properties button on the left:



18. Click the "TODO" next to Server URL to make a text box appear.
19. Type your server URL in the text box.
20. Click the **OK** button.
21. If multi-factor authentication (MFA) is required by your server or user:
    1. Right click the Thycotic Secret Server (PowerShell) connection and select **Properties**. The Edit Properties wizard appears:



2. Click the **Dynamic Folder Script** button. The script appears:

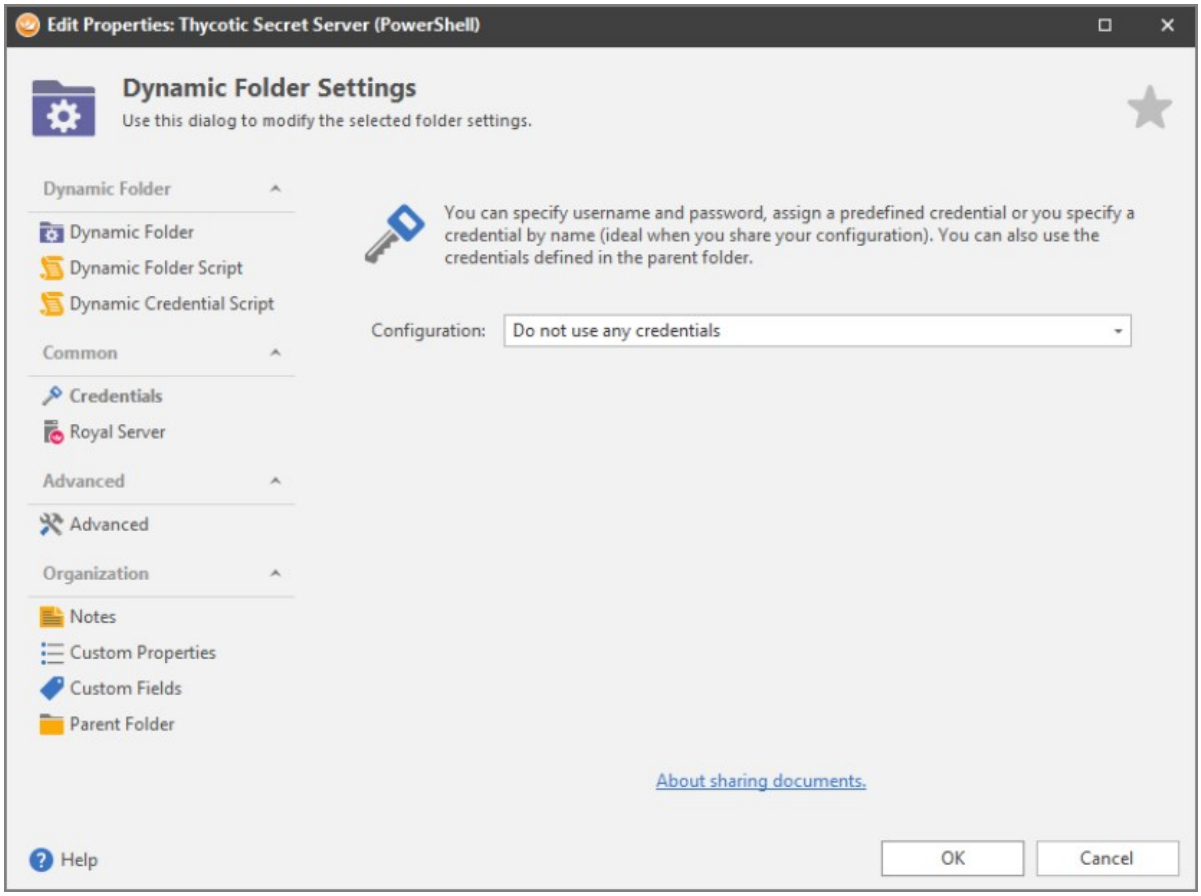


3. Scroll down to the very last line of the script:

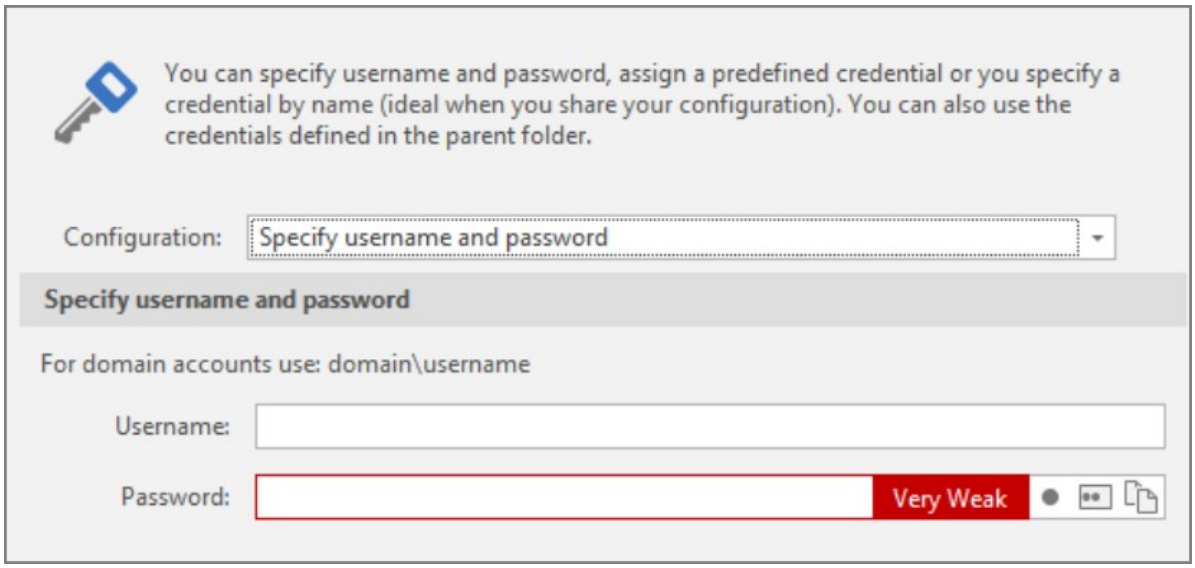
   Get-Entries -url "$CustomProperty.ServerURL$" -username "$EffectiveUsername$" -password "$EffectivePassword$" -requiresMFA $false

4. Change the $false to $true.
5. Click the **OK** button.
6. Repeat the process for the Dynamic Credential script.

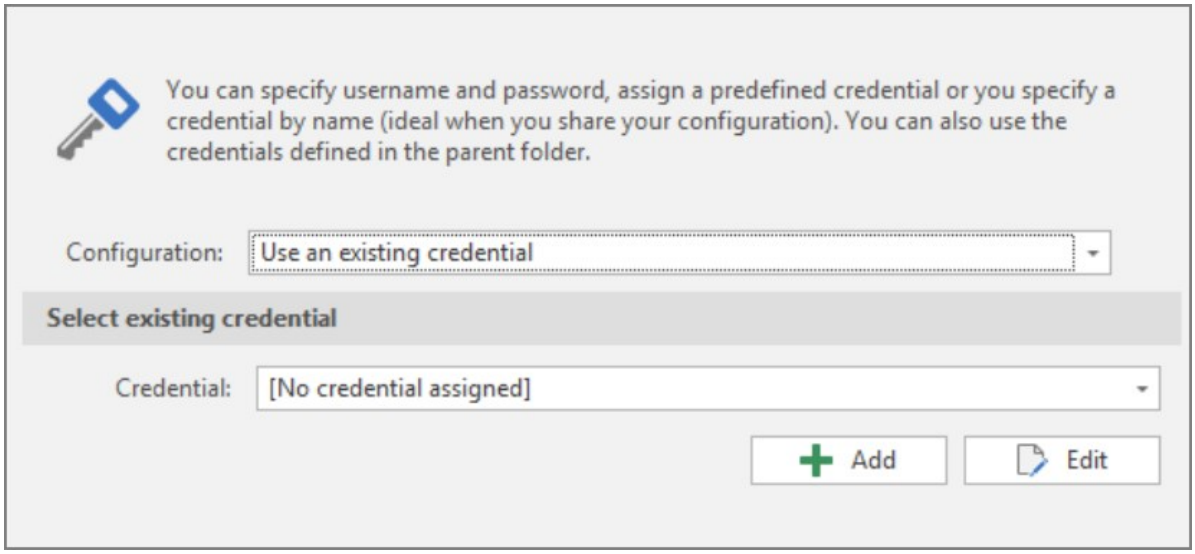
22. Click the **Credentials** button. The Edit Properties wizard appears:



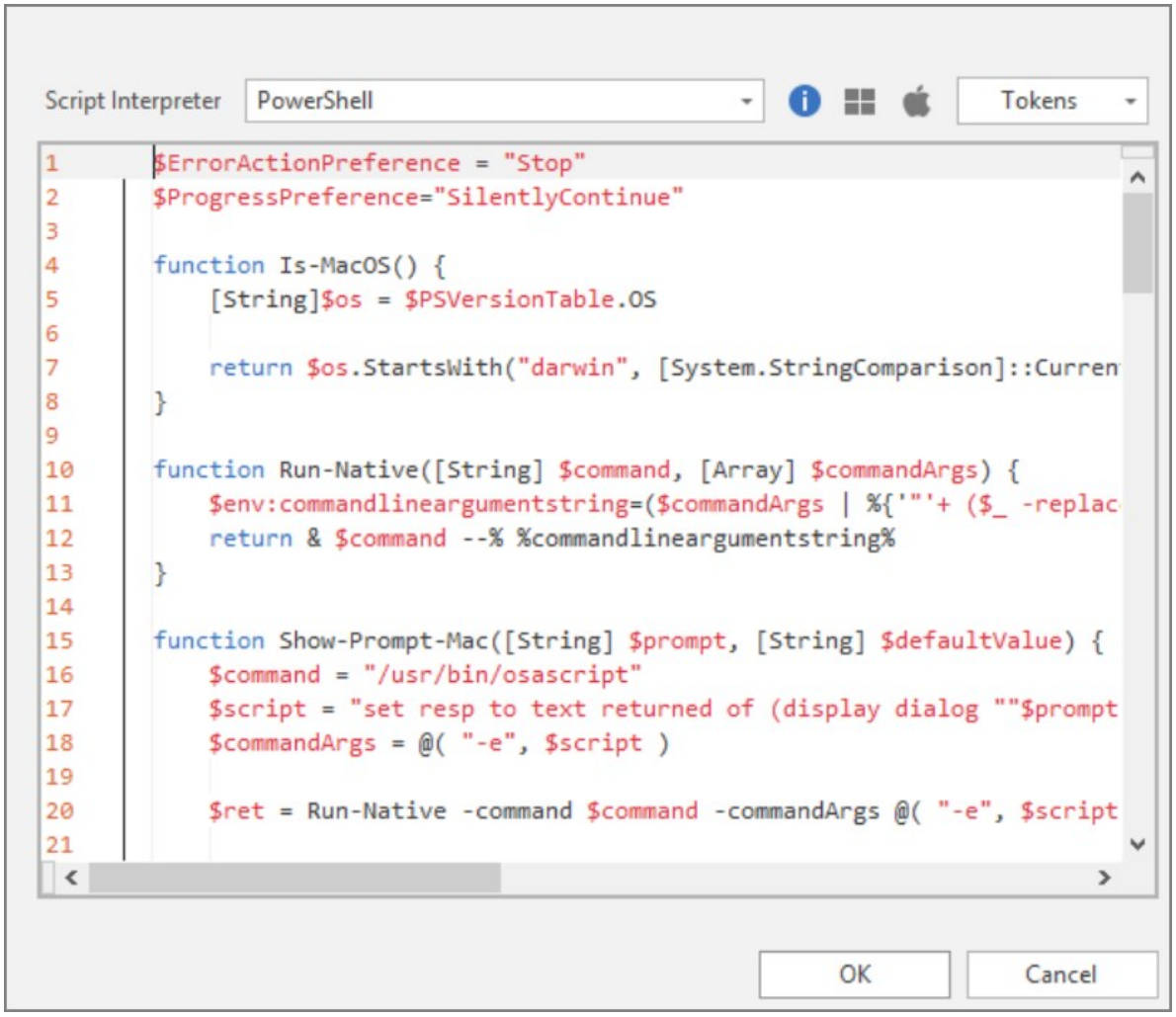
23. Click the **Configuration** dropdown list and select either **Specify username and password** or **Use an existing credential**.
24. If chose the former, type your information in the **Username** and Password text boxes:



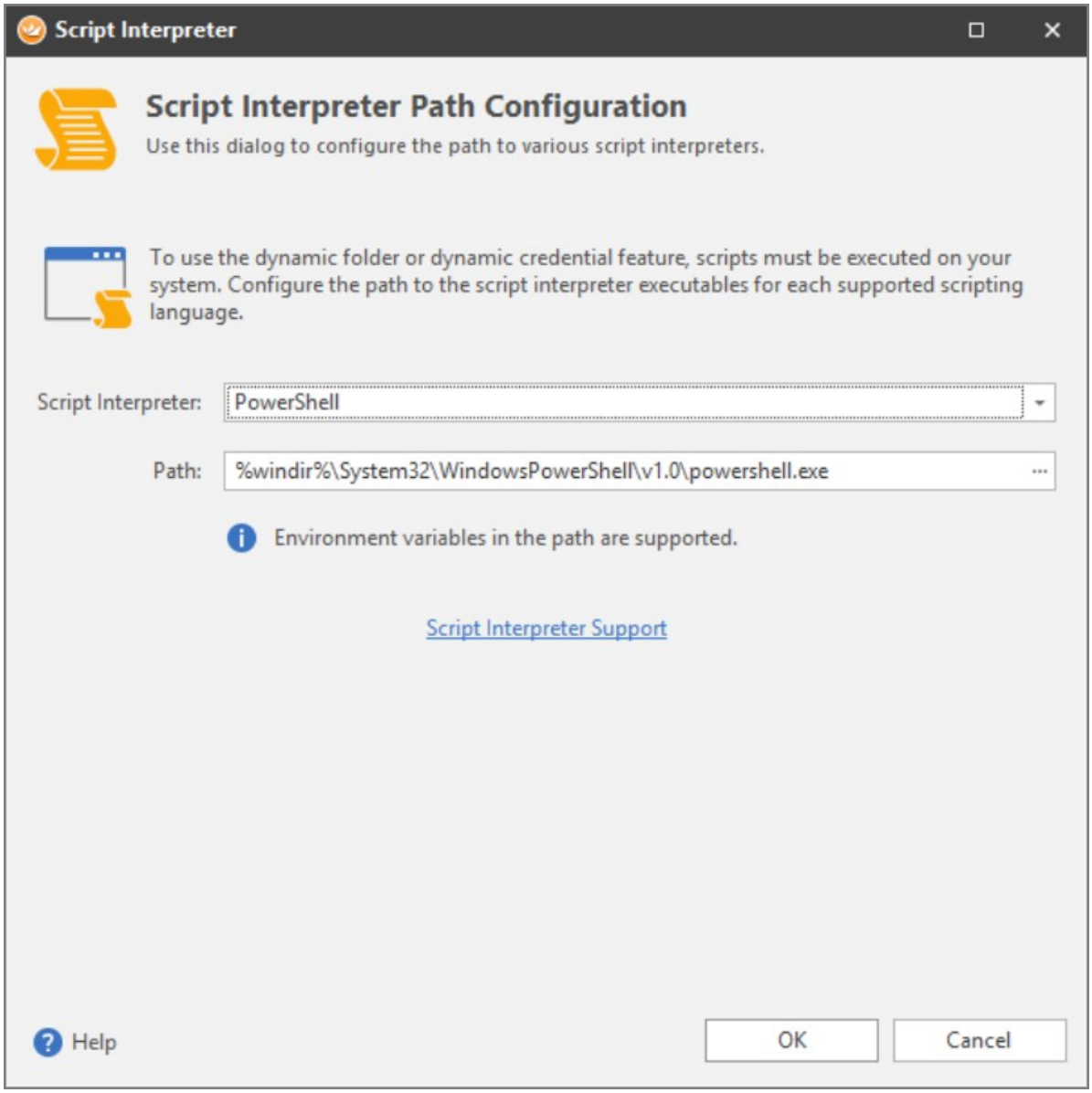
25. If you chose the latter, select an existing credential from the **Credential** dropdown list. If your credential is not in the list, click the **+ Add** button to add it.



26. Click the **OK** button.
27. Click the **Dynamic Folder Script** button. The script editor appears:



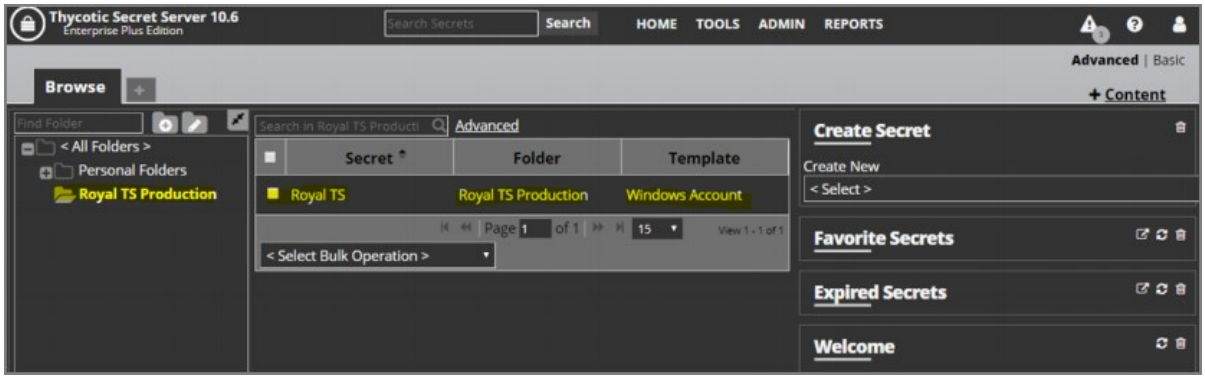
28. Click the blue information button. The Script Interpreter Path Configuration dialog box appears:



29. Ensure the path to your installed powershell.exe file is correct. If not navigate to the correct path and click the **Open** button.

Configure Secret Server for Integration

1. Log into Secret Server using the credentials assigned to the dynamic folder
2. Create a **RTS folder** and **Secret**:

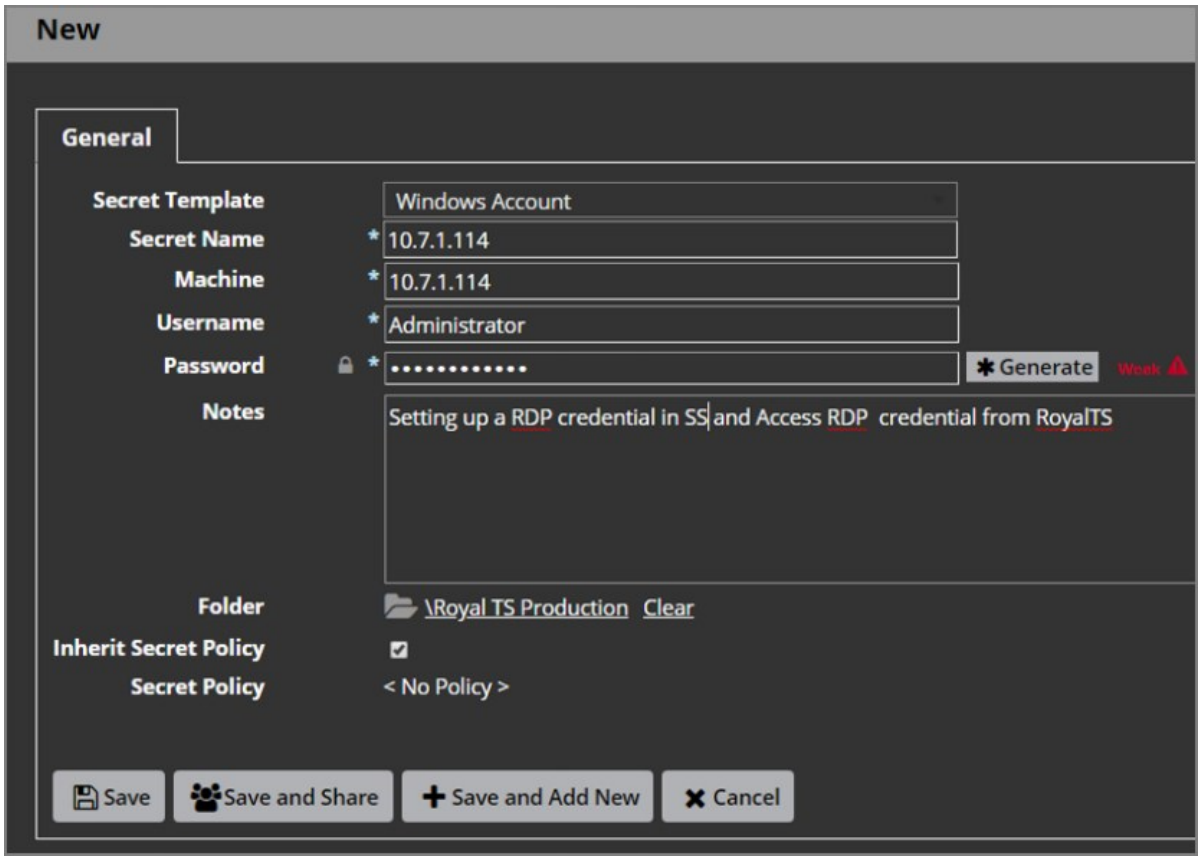


3. Return to the RTS application.
4. Right-click the Secret Server dynamic folder in the tree.
5. Click **Reload**. The secret you created in SS is pulled into your dynamic folder:

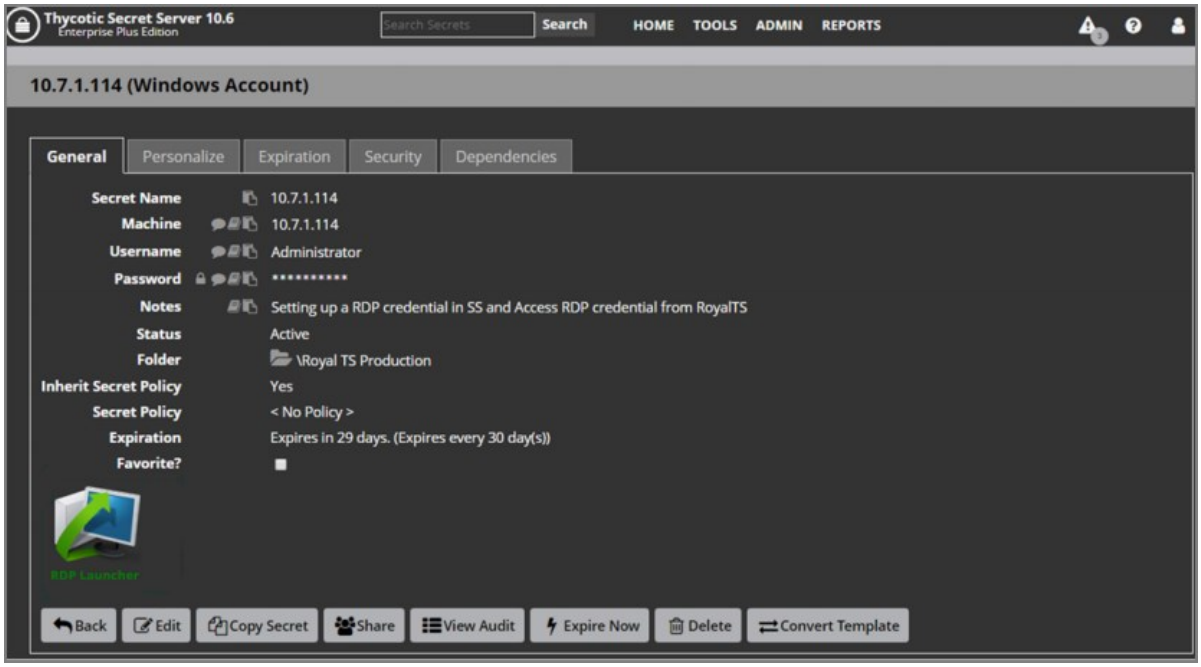
Note: If you want to view your user ID and password, you can right-click the credential for the dynamic folder and select Copy to Clipboard.

Create an RDP Credential in Secret Server

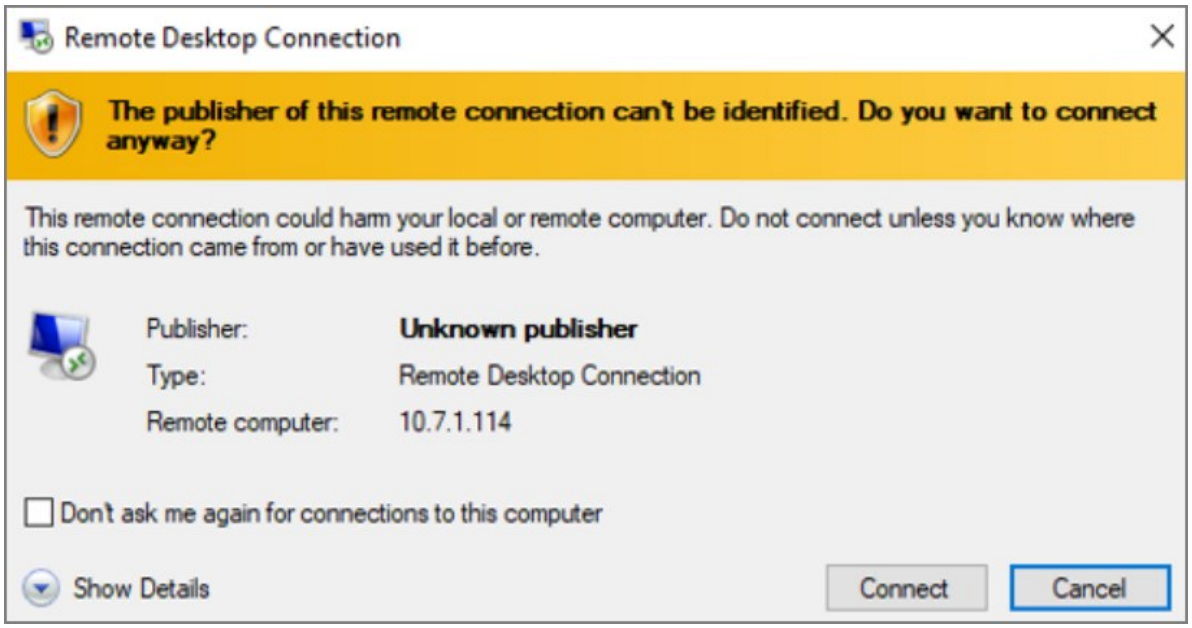
1. Create a secret in Secret Server using the **Window Account** secret template:



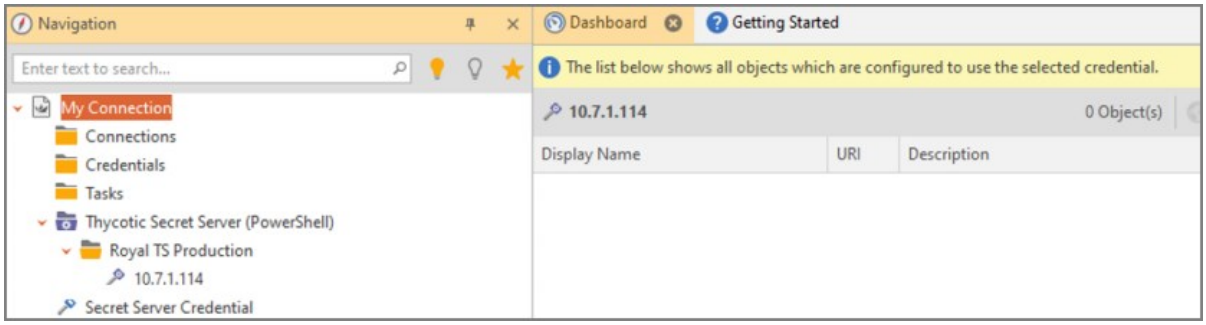
2. Click the Save button. The new secret appears:



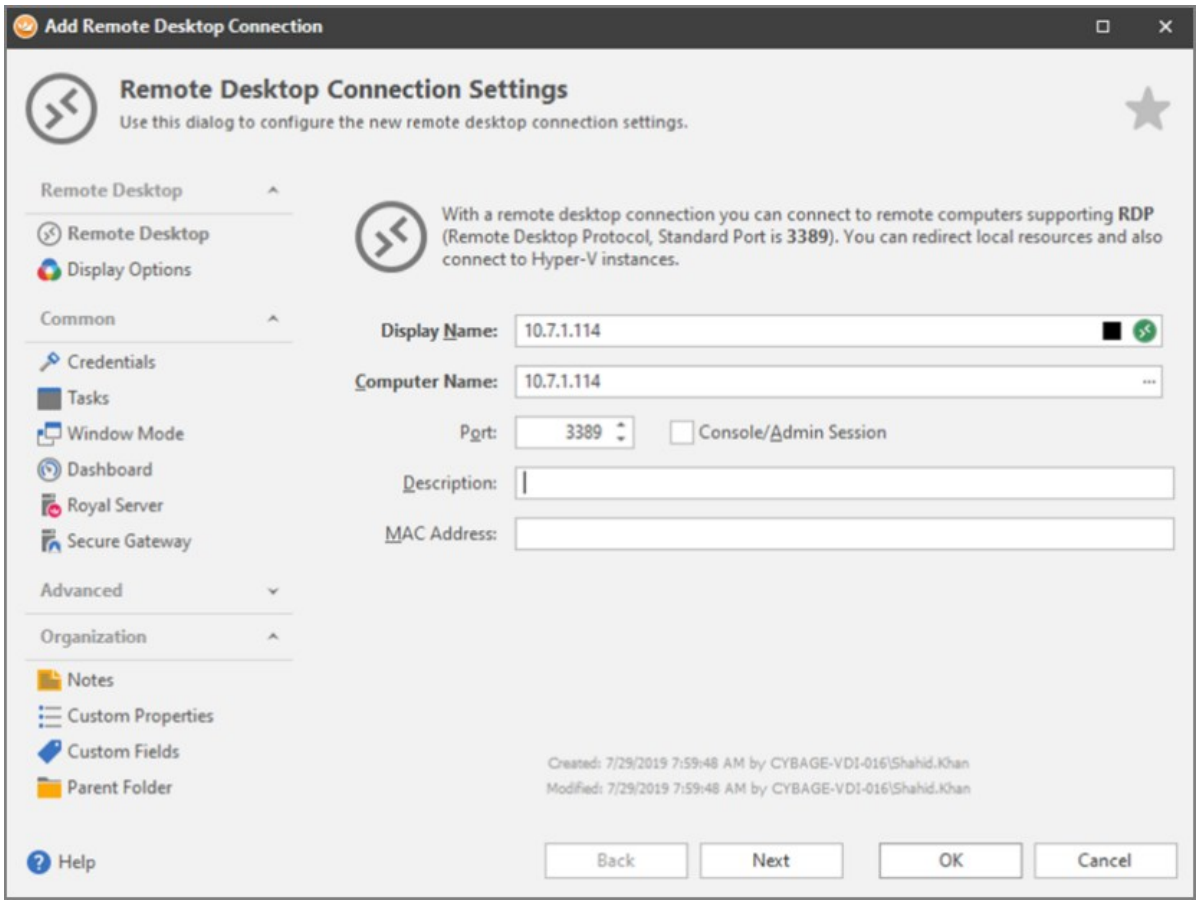
3. Click the **RDP Launcher** button. The session launches, and a confirmation dialog box appears.
4. Click the **Open RDPWinBootstrapper** button. The session connects with the target server, and an RDP dialog box appears:



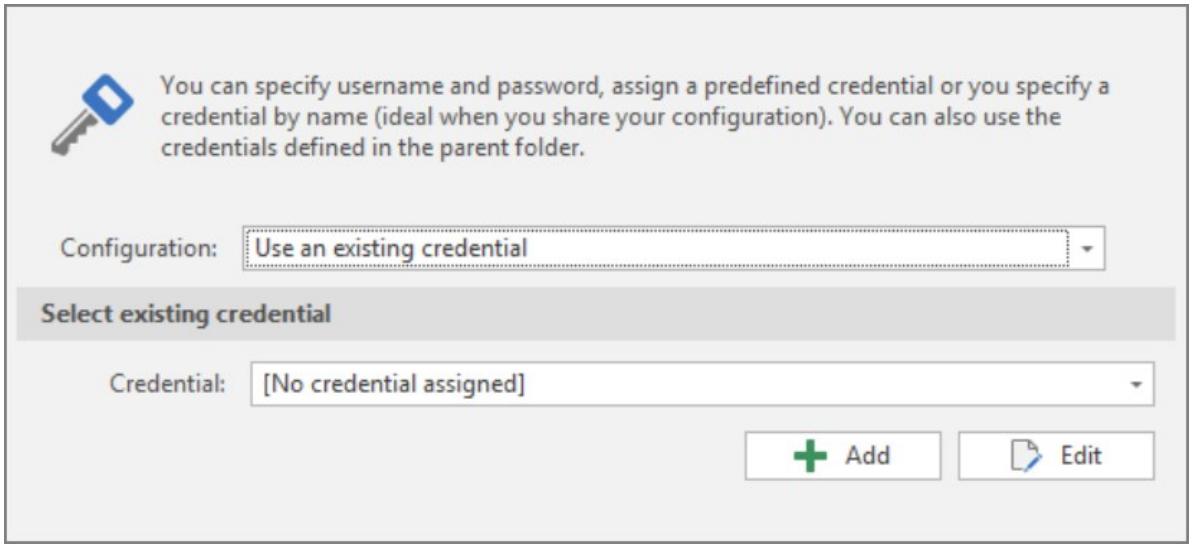
5. Click the **Connect** button. The text "Active Session" appears below the RDP Launcher button.
6. Return to RTS.
7. Once again, reload the dynamic folder. The Window Account secret appears in RTS:



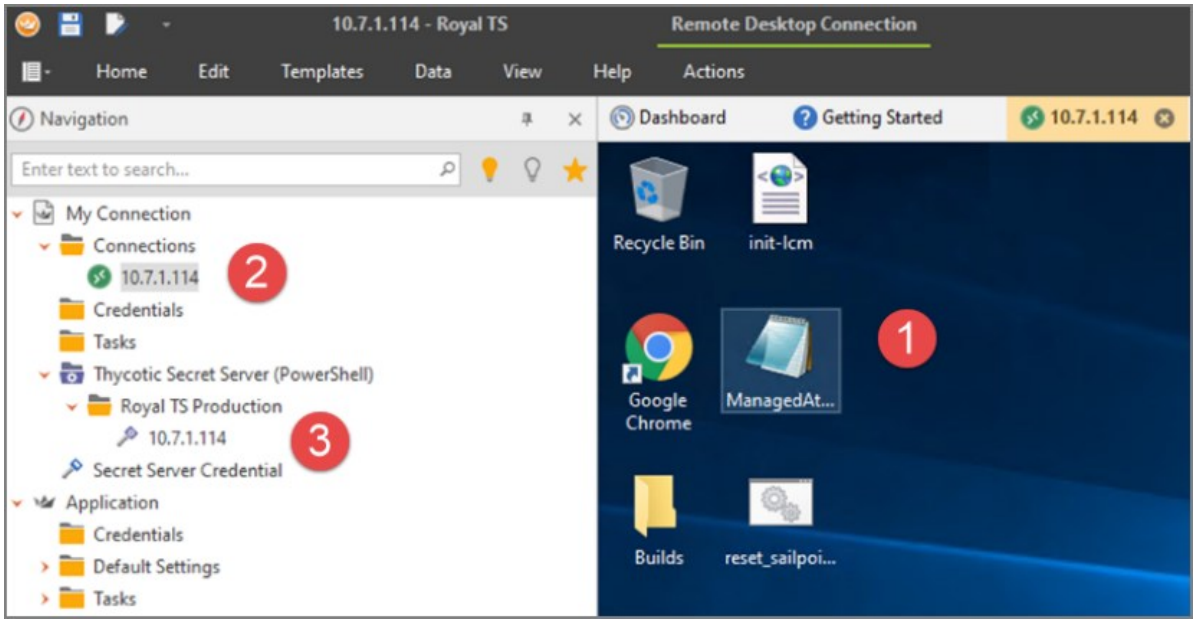
8. Right-click the **Connections** folder and select **Add > Remote Desktop**. The Add Remote Desktop Connection wizard appears:



9. Click the Credentials button:

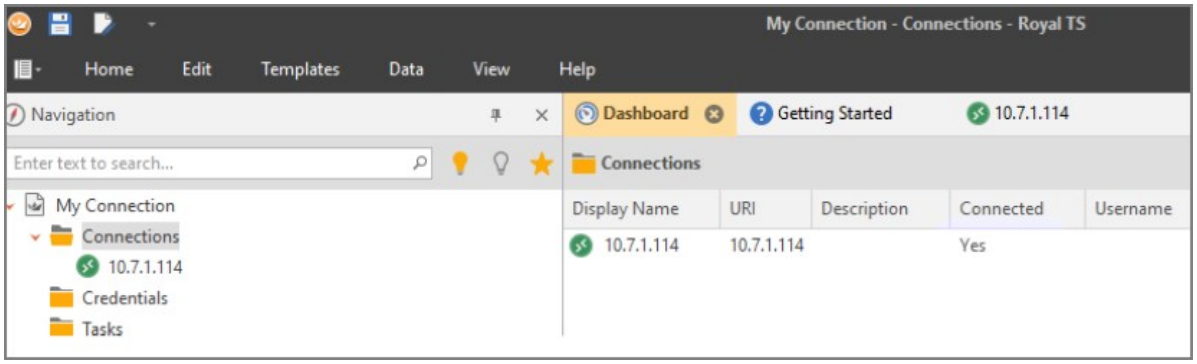


10. If necessary, click the **Configuration** dropdown list and select **Use an existing credential**.
11. Click the **Credential** dropdown list and select the credential you created earlier.
12. Click the **OK** button. The wizard closes.
13. Click to expand the **Connections** folder under your connection.
14. Right-click the connection you created and select Connect. Your new session will appear as a new tab in RTS:



- The new RDP session appears in its own tab.
- The existing credential is used to connect with the 10.7.1.114 server.
- The fetched credential from the Window Account secret from Secret Server.

15. "Yes" appears in the Connected column for that server's connection.



Note: If you try to connect to the server and get a "Windows Security: Your credentials did not work" error, try correcting the password in the Windows Account secret in SS and reloading the dynamic folder in RTS.

SailPoint

- SailPoint IdentityIQ
- SailPoint IdentityNOW

SailPoint IdentityIQ

This section reviews the SailPoint IdentityIQ pre-configuration for use with the SCIM Connector, as well as an use examples and known usability issues.

SailPoint Documentation

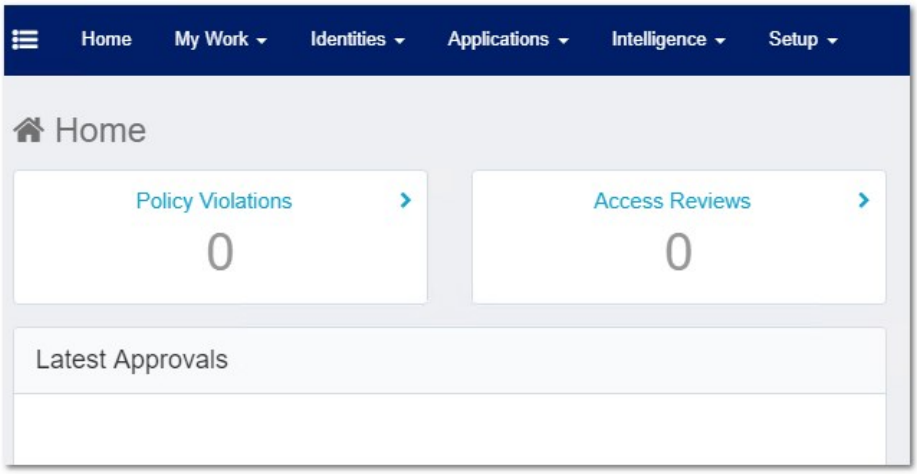
- General SailPoint IdentityIQ Documentation: https://community.sailpoint.com/community/identityiq/product-downloads
- SailPoint SCIM Verification Plugin User Guide: https://community.sailpoint.com/docs/DOC-10479
- Privileged Account Management Deployment Guide: https://community.sailpoint.com/docs/DOC-12926
- Privileged Account Management in IdentityIQ: https://community.sailpoint.com/docs/DOC-9014

Configuring a SailPoint IdentityIQ Endpoint

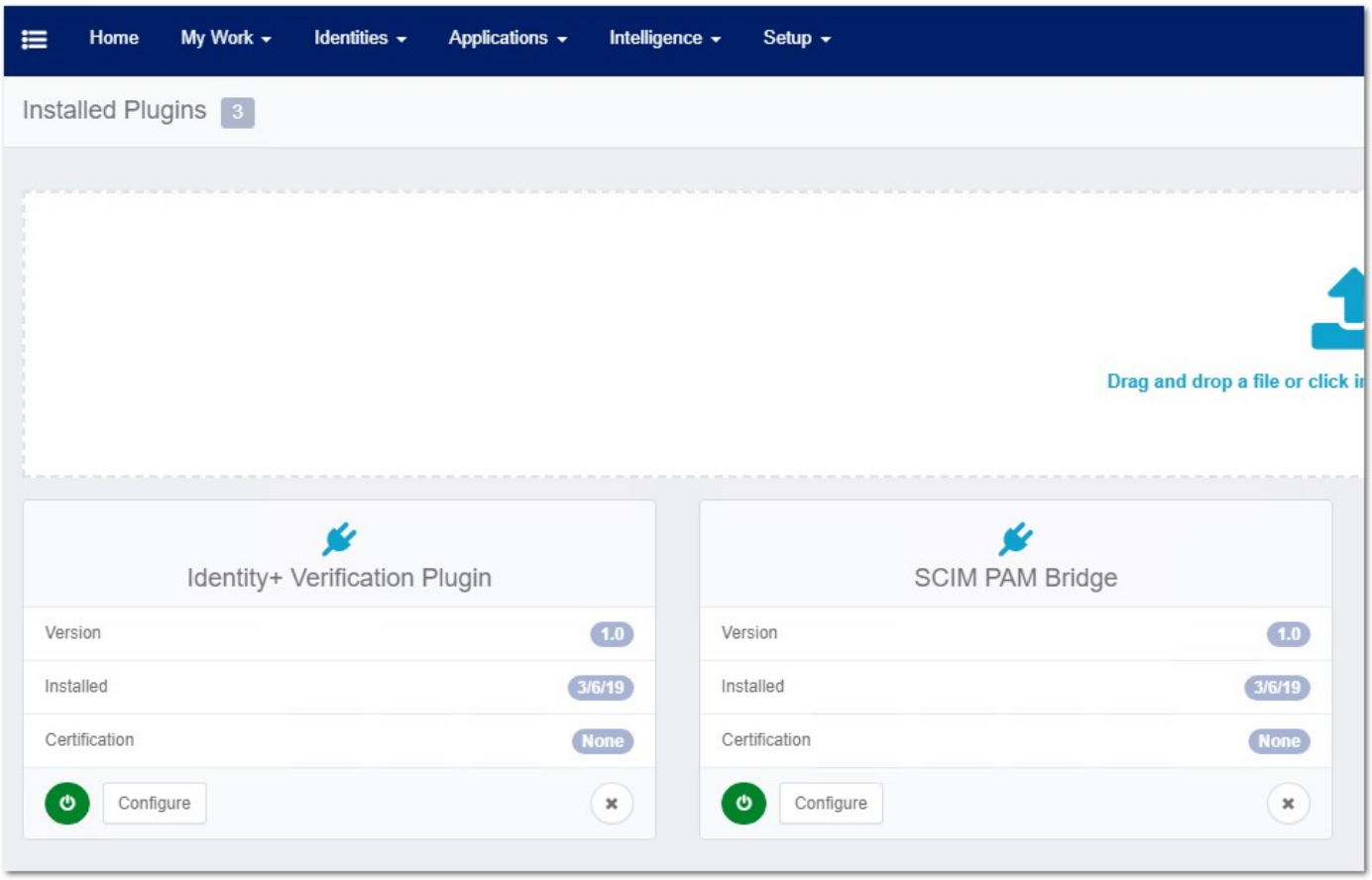
The steps in this section are required to configure SailPoint's IdentityIQ platform for use as a SCIM Endpoint for the Thycotic SCIM Connector application. These steps are taken within SailPoint IdentityIQ, in addition to the steps from the Making a SCIM Endpoint section.

> **Note**: We suggest referring to SailPoint documentation for the most recent instructions, showing the most recent user interface.

1. Open SailPoint IdentityIQ:



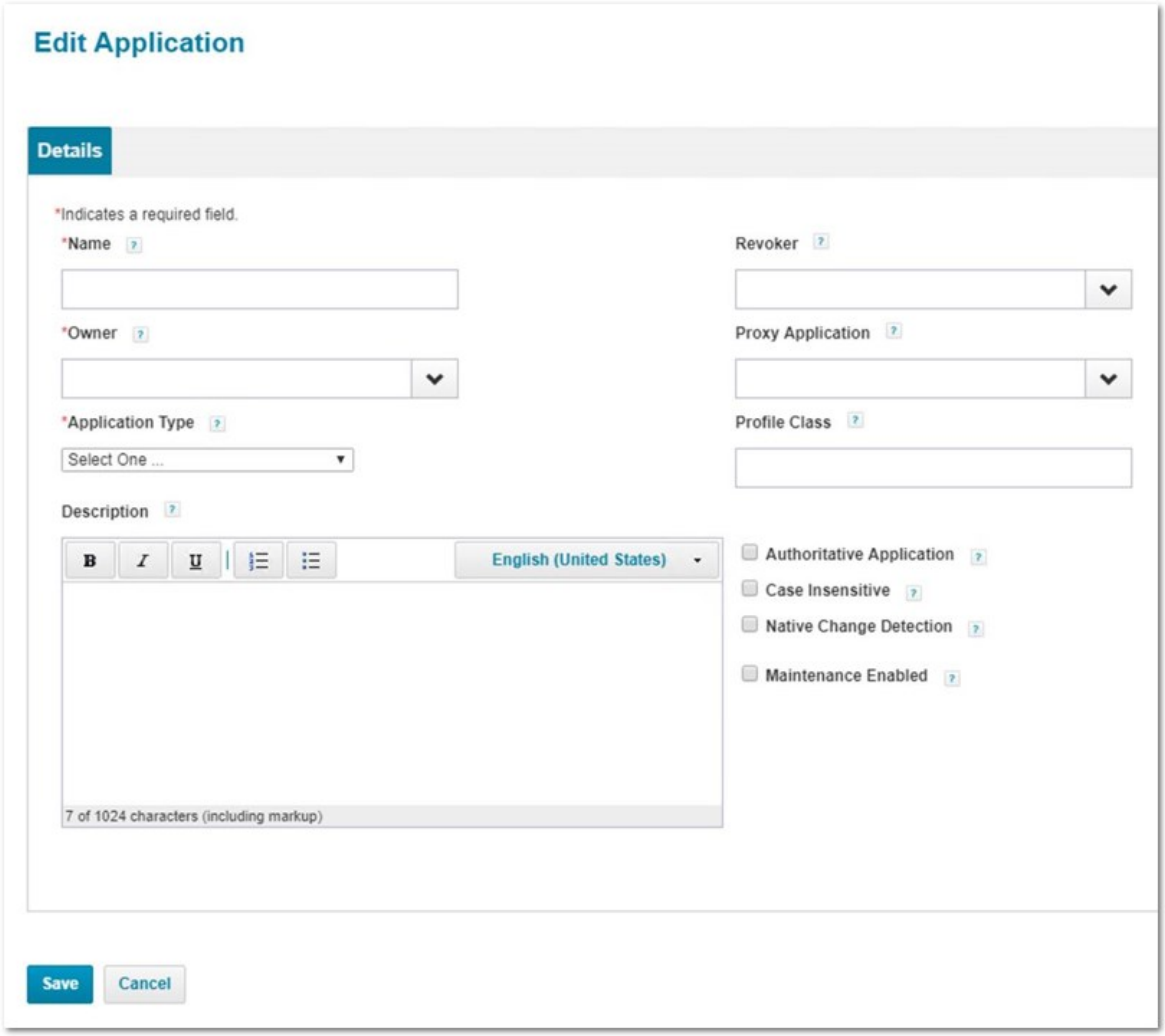
2. Navigate to **Setup** and select the **Plugins**.



1. Add the following plugins:
    - Identity+ Verification
    - SCIM PAM Bridge

3. Navigate to **Applications** and select **Application Definition**.

4. Click **Add New Application**.



Enter the following details:

- **Name**: Enter the name for the application.
- **Owner**: Enter the user account used to connect to the SCIM Connector application.
- **Application Type**: Select "Privileged Account Management" from the drop-down menu.
- Check the box for the **Authoritative application** option

1. Click **Save**. This takes you back to the **Application Definitions** page.

5. Select **Configuration**.

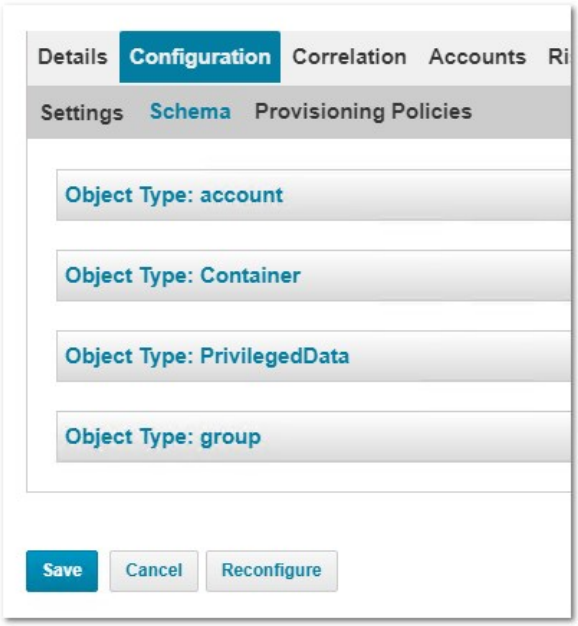
Enter the following details:

- **Base URL**: This should be the URL for the installed SCIM Connector application with "/v2" appended to the end of the URL.
- **Authentication Type**: Select the **API Token** option.
- **API Token**: Copy and paste the API token that was generated in the SCIM Connector application when creating the SCIM endpoint (using the **Non-expiring token** option). See Creating the SCIM Endpoint.
- **Permissions**: Click the **+** button to the right and add **View** permissions.

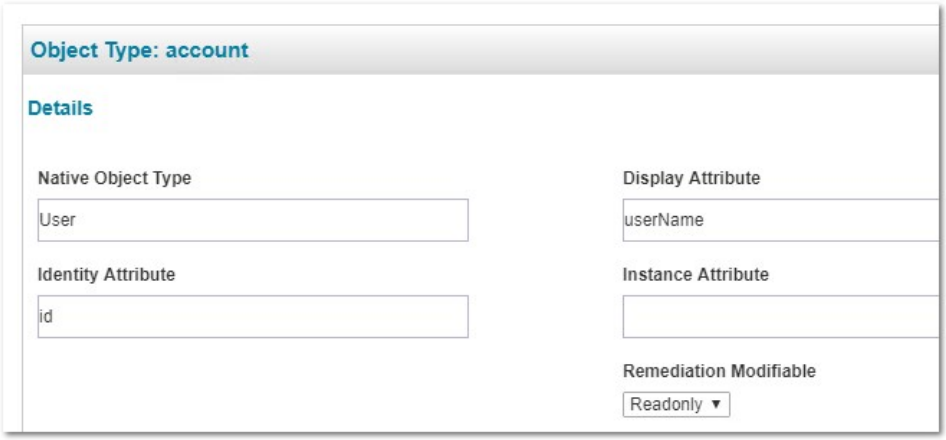
  Note: We recommend you remember these values because you will need them again in a later step.

6. Once the above settings are set, click **Test Connection** to see if the configuration works.

7. Navigate to **Configuration | Schema**.



8. Open **Object Type: account**.



1. Ensure the **Identity Attribute** is set to **id**.
2. In the **Attributes** section of the page, click the **Add New Schema Attribute** button. A new attribute row appears.
3. Add the following attribute row entries:

Name	Description	Type	Properties
source		string	Correlation Key
nativeidentifier		string	Correlation Key

9. Go the **Object Type: container** section and add the following four attribute row entries:

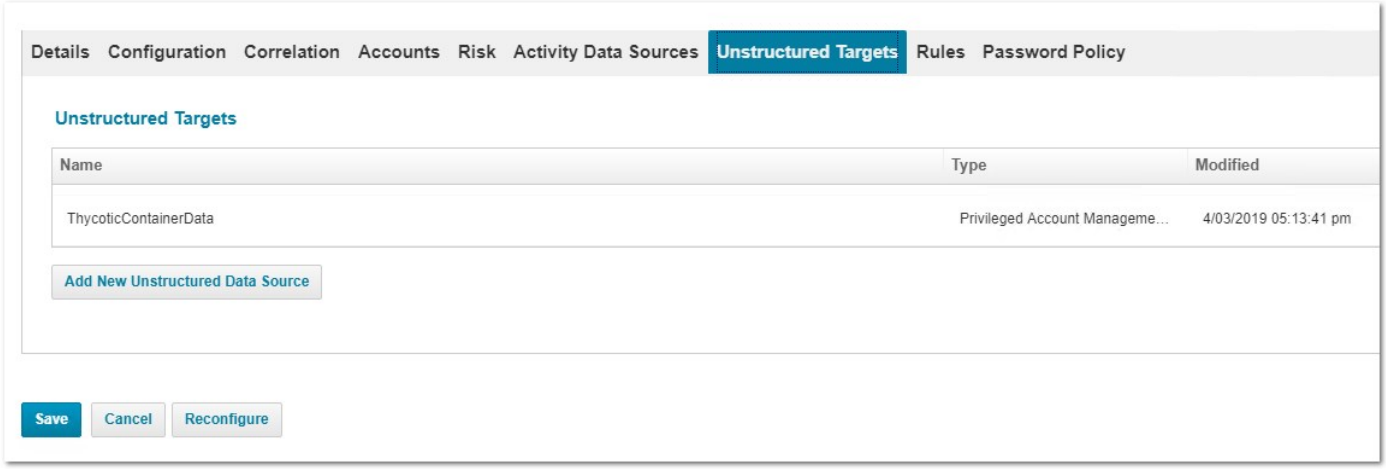
Name	Description	Type	Properties

Name	Description	Type	Properties
privilegedData.value	The ID of the privileged data	string	Multi-Valued
privilegedData.$ref	A URI reference to the PrivilegedData	string	Multi-Valued
privilegedData.display	The displayable value of the PrivilegedData	string	Multi-Valued
privilegedData.type	The type of the PrivilegedData	string	Multi-Valued

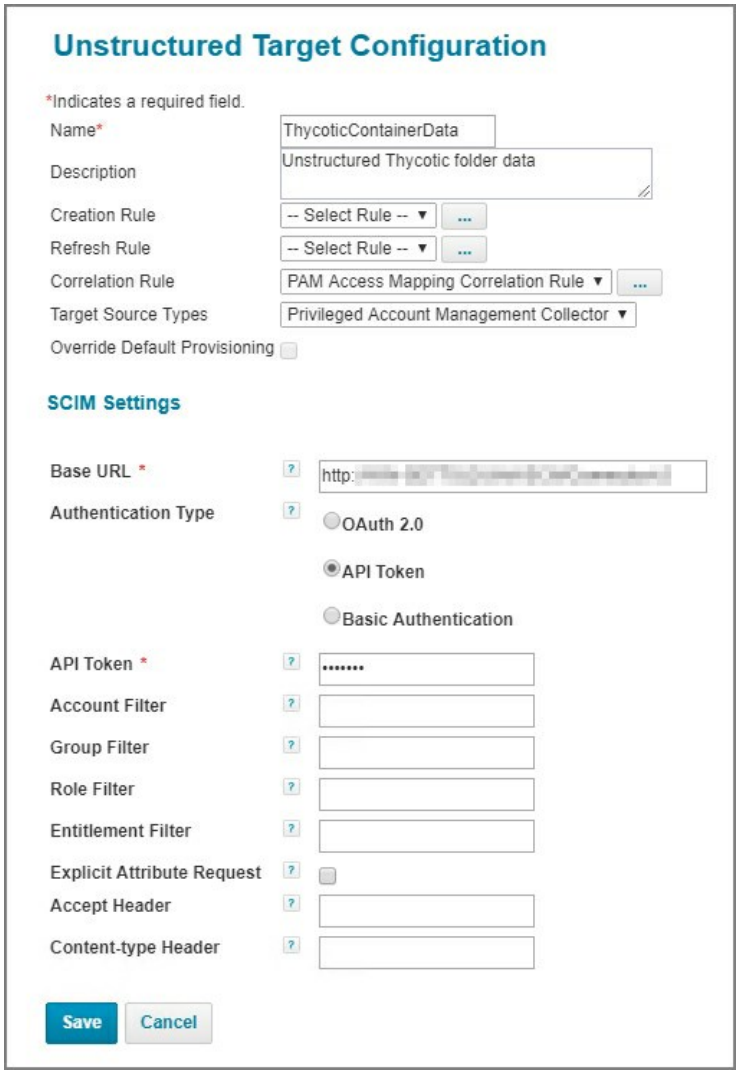
10. Using the same method, go to the **Object Type: group** section and add the following attribute rows:

Name	Description	Type	Properties
displayName	A human-readable name for the Group. REQUIRED	string	Multi-Valued
members	A list of the members Group		
id	The unique identifier of the Group	string	
source		string	Correlation Key
nativeidentifier		string	Correlation Key

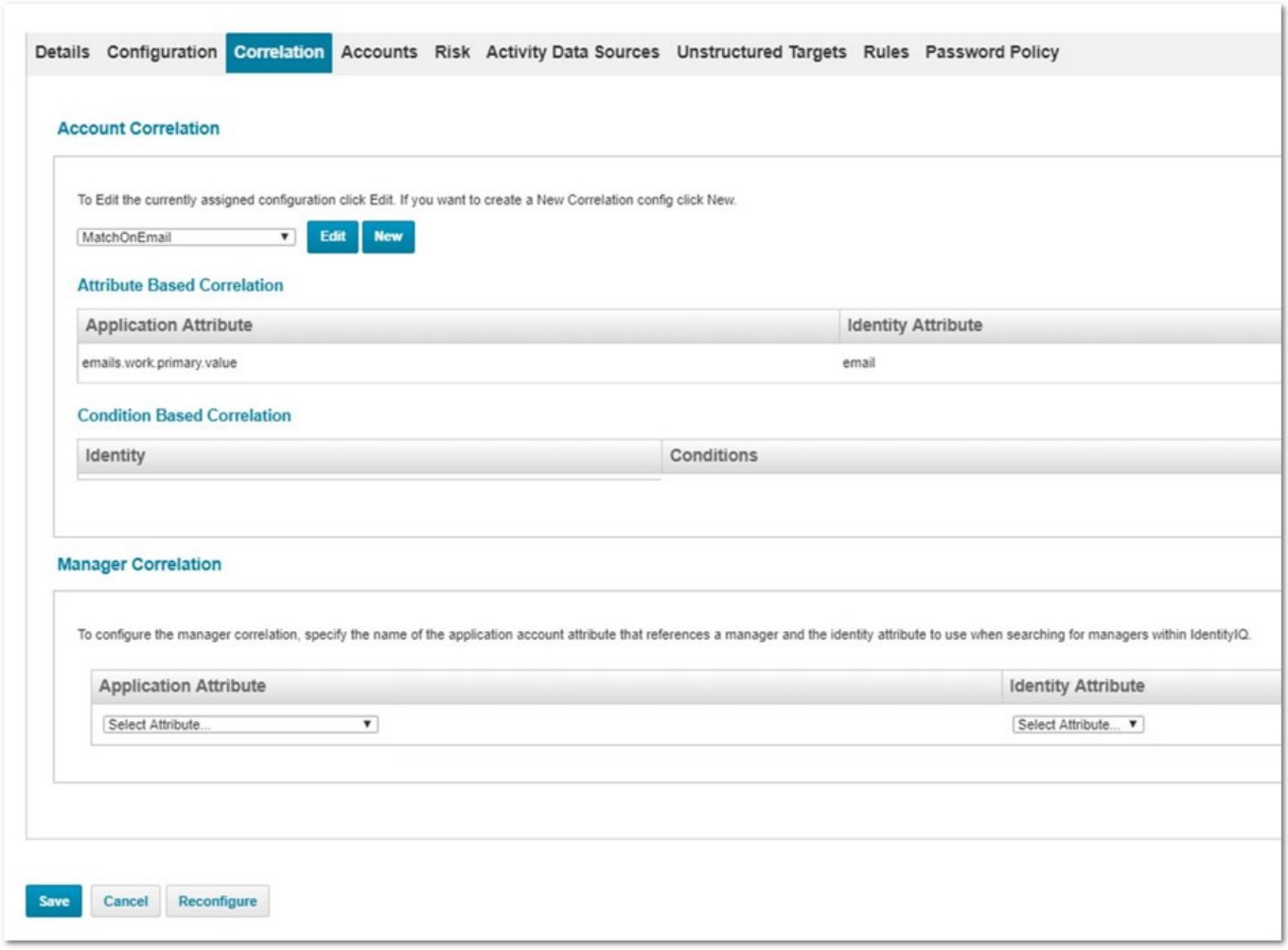
11. Click **Save** at the bottom of the page.
12. Return to the **Application Definition** page.
13. Click the **Unstructured Targets** tab.



14. Click the **Add New Unstructured Data Source** button. The Unstructured Target Configuration page appears:



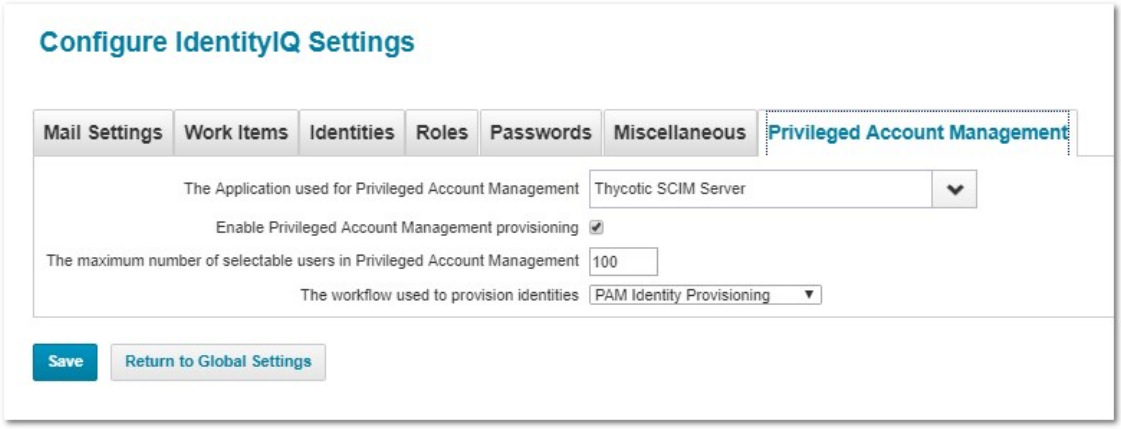
15. In the **Unstructured Target Configuration** dialog that opens, enter the following details:
    - **Name**: Enter the name of the application that was used earlier.
    - **Description**: Enter a short description.
    - **Correlation Rule**: Select the **PAM Access Mapping Correlation Rule** value.
    - **Target Source Types**: Select the **Privileged Account Management Collector** value.
    - **Base URL**: Enter the same **Base URL** value used earlier.
    - **Authentication Type**: Select the **API Token** value.
    - **API Token**: Enter the same token that was used earlier.
16. Click **Save**.
17. On the **Application Definition** page, select the **Correlation** tab:



18. Add a new correlation and follow the wizard dialog that pops-up. Ensure that "email.work.primary.value" correlates to the "email" value.

    Note: This correlation value **must** be set for the integration to correctly match user values.

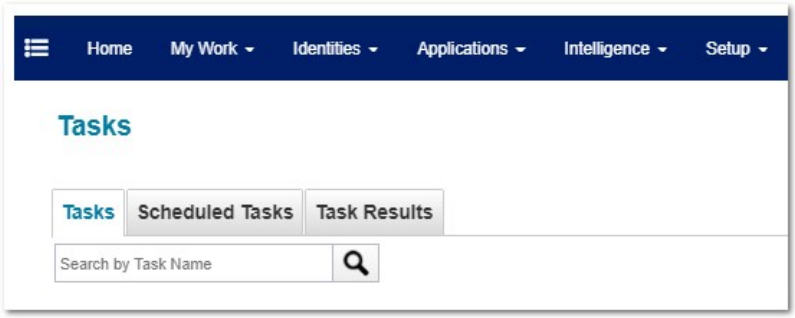
19. Under the Global Settings, navigate to **Identity IQ Configuration**.
20. Select the **Privileged Account Management** tab:



Enter the following:

- **The Application used for Privileged Account Management**: Set the application that was created earlier.
- **Enable Privileged Account Management provisioning**: Enable the check box.
- **The workflow used to provision identities**: PAM Identity Provisioning

21. Click **Save**.
22. Navigate to **Setup | Tasks**.



1. Add the following tasks:
    - **Account Group Aggregation**: Select the application that was created earlier.
    - **Account Aggregation**: Select the application that was created earlier.
    - **Target Aggregation**: Select the Target source that was created.
2. After the tasks are added, right click each task and select **Run in Background** to execute them in the background.

The basic configuration is now complete.

Using SailPoint IdentityIQ After a Connection Is Made

Once SailPoint IdentityIQ is configured to work as an endpoint for the SCIM Connector and a connection has been established from the SCIM Connector itself, users can run actions directly from IdentityIQ.

Examples

Adding a User to a Group

1. In the main SailPoint dashboard, click the menu icon in the top left corner and select **Manage Access**.
2. Click to select the **Manage User Accounts** sub-option.
3. Search for the user name for the account that you want to add to a group.
4. Once the user is found, click to select the radio button in the top left corner of the user card. The button turns green.
5. Click **Next**.
6. Search to find the group that the user should be added into.
7. Click to select the checkmark circle in the top left corner of the user card. The circle turns green.
8. Click **Next**.
9. Review the setting to ensure the correct user and group are selected.
10. Click **Submit** at the bottom of the page to complete the action.

After add the user to the group, you should be able to view their access to the container on the Privileged Account Management page, and they will be listed in the Effective Access tab for the container that the group is applied to.

Please see the SailPoint Documentation section for more information on navigating SailPoint IdentityIQ.

Directly Adding a User to a Container

1. In the main SailPoint dashboard, click the menu icon in the top left corner and select **Manage Access** from the menu that opens.
2. Click to select the **Privileged Account Management** sub-option. The Privileged Account Management page appears, and you see all of the PAM containers and folders the logged in user has access to.
3. Find the container you want to add access to (manually or using the search feature).
4. Click **Manage** at the bottom of that container box.
5. Click **Add Identities** on the right-hand side. This starts the process of giving a user direct permissions access to the container. The Choose Identities dialog opens.
6. Search for the user account that should be added.
7. Click **Next** to continue. The dialog name changes to Add Container Permissions, and you now can select the access level you want to apply to the account.
8. Click to select the check box for the permission you want to add.
9. Click **Submit** to complete the action.

After the user has been added to the container, you should be able to view their access to the container on the Privileged Account Management page, and they should be listed in the Direct Access tab.

Please see the SailPoint Documentation section for more information on navigating SailPoint IdentityIQ.

SailPoint Integration Concepts and Limitations

This section reviews any SailPoint-specific limitations. For a list of the SCIM Connector application limitations, please see the SCIM Connector Limitations section.

- In SailPoint IdentityIQ, there are "containers" and "privileged data." The containers map to Secret Server folders, and privileged data maps to secrets.
- SailPoint allows adding permissions to containers, but they cannot be directly added to privileged data. That is, they cannot be added directly to a Secret Server secret. So when a user gets access to a container, the user is really getting access to a Secret Server folder.
- While there is no direct way to give users access to a specific secret, they can still be given access indirectly by adding a user into a group that already has access to both the folder/container and the secret/privileged data.
- When a users are given access to a container/folder, either with direct access or by adding them to a group, they only have "view" access to the container. More granular assignment of permission levels can only be defined in SS.
- If the "view" permission setting seen in the Configuring a "SailPoint IdentityIQ Endpoint" section is not configured correctly, an incorrectly formatted POST call to the SCIM Connector application will result, which returns a HTTP 400 error message.
- Any sensitive information that is associated with a secret/privileged data, such as a password, is not shared over the SCIM Connector and must be viewed in SS.
- Personal folders in Secret Server can be viewed in SailPoint, but users cannot be given direct access to the folders. However, users can be given access by adding them to an existing group. The owner of the personal folder cannot have their access removed from the folder.
- Using custom attributes or extensions with SailPoint IdentityIQ and the SCIM Connector is not currently supported.

SailPoint IdentityNow

What is IdentityNow?

IdentityNow by Sailpoint is a cloud based Identity and Access Management tool designed to simplify access changes during HR events such as onboarding, job function changes, and offboarding. IdentityNow centralizes access management decisions for business systems into a single console with unified workflows. Thycotic Secret Server is one of the tools for which IdentityNow can manage access granting roles to users based on their position inside the business which can come with access to specific privileged secrets. When a user transitions business roles the secrets to which they will have access can also be modified without further effort on the part of the administrator. In addition if an employee or contractor leaves the organization their access to secrets can be removed ensuring that critical business assets remain protected.

Why Use Secret Server with IdentityNow?

IdentityNow customers still require Thycotic's Secret Server for privileged account management because although IdentityNow excels at managing individual user access to systems you would not want to use it to manage shared resources such as service accounts. Additionally Secret Server offers more fully featured tools specific to privileged account management with audit, password rotation, dependency management, and others. By using both products in combination clients can achieve the automation benefits for provisioning and managing user accounts and access through IdentityNow along with the security advantages of Privileged Access Management through Secret Server.

How do Secret Server and IdentityNow communicate?

IdentityNow and Secret Server are both capable of using the System for Cross-domain Identity Management (SCIM) to send messages about triggering events. SCIM is a standard data exchange format used in managing identity between cloud enabled systems. Thycotic provides a SCIM connector which can be installed on any server in the client's environment and will ingest messages from IdentityNow or other SCIM supporting IDAM systems and convert them into REST calls to the Secret Server API. When a user is provisioned or modified inside of IdentityNow it will trigger a message to the Thycotic SCIM connector, the connector will parse the message and then make the appropriate changes inside of Secret Server