Cryptographic Security Platform

Compliance Manager user guide

Document issue: 1.0
Issue date: September 10, 2026

Contents

1 About this guide

This guide describes how to use the cloud-based Compliance Manager application.

Revision information

See the table below for changes in each document issue.

Issue	Date	Section	Changes
1.5	Sep 2026	AWS Elastic Load Balancer plugin credentials	Fix broken links
1.4	Aug 2026	Plugin reference	Document new plugins
1.3	Aug 2026	All	Fix broken links
1.2	Aug 2026	All	Update fonts
1.1	Aug 2026	Cover	Update title
1.0	Jul 2026	All	Document release

Other documentation

For instructions on installing and administering the on-premises version of Compliance Manager, see:

https://api.managed.entrust.com/kcm/latest

Documentation feedback

You can rate and provide feedback about product documentation by completing the online feedback form:

https://go.entrust.com/documentation-feedback

Any information you provide goes directly to the documentation team and is used to improve and correct the information in our guides.

2 About CSP Compliance Manager

Entrust Cryptographic Security Platform (CSP) Compliance Manager provides centralized visibility into an enterprise's cryptographic assets and a policy engine that enables fine-grained control of keys, secrets, and certificates, regardless of location. This allows businesses to easily establish and maintain an inventory and gain visibility into all related information for cryptographic assets, whether on-premises or in cloud environments.

> Inventory information provides granular details on which assets are used and can include information about the asset's history, ownership, environment, purpose, and whether it is used in a critical system.

3 Using the Compliance Manager interface

The Cryptographic Security Platform Compliance Manager Dashboard lets you view a snapshot of your data sources, their compliance status, and the percentage of completed documentation. You can

- Filter by individual data sources or view them all together.
- View the total number of cryptographic assets and the type.

Tenant administrators will receive an email with the URL, username, and temporary password for Cryptographic Security Platform Compliance Manager.

> ⚠ If email notifications are not available, then the tenant manager will need to send you the information separately.

Cryptographic Security Platform Compliance Manager has the following users.

Role	Permissions
Administrators	Add, edit, and view all collections
Auditors	View access for all collections.

See below for accessing the Cryptographic Security Platform Compliance Manager interface for the first time.

1. Read and agree to the Entrust End User License.
2. Change the temporary password to continue.
3. Optionally:
    - Add data sources as explained in Creating a Data Source Connection.
    - Create a collection to organize your vault, as explained in Creating a Collection.
4. Manage vault contents with the following dashboards.
    - World Map
    - Data Sources
    - Total Assets
    - Risk of Cryptographic Assets (Risk Score)
    - Compliance Trends
    - Documentation Trends

World Map

This section displays a world map with pins indicating where your vaults are located.

- Click the **Collections** drop-down menu at the top to filter what collections you want to view.
- Click the **Data Source** drop-down menu at the bottom to filter what data sources you want to view.

Data Sources

The data sources row displays the number of data sources, their current compliance, and their current documentation. Click the data sources number to go directly to the Data Sources page.

Total Assets

The total assets row displays all cryptographic assets and the data source they belong to.

- Click the total objects number to go directly to the Cryptographic Assets page
- Click a data source link to view all cryptographic assets in that data source.

Risk of Cryptographic Assets (Risk Score)

This section displays the total risk score for all objects in all data sources. Each vault's overall security risk score is calculated using a weighted average that reflects the individual risk scores of the objects it contains, providing a nuanced measure of the vault's security posture.

The risk score provides a quantifiable measure of the security risk associated with the vault, taking into account all the cryptographic assets it contains. Click the risk score percentage to go directly to the Cryptographic Assets page, filtered by that risk score.

Compliance Trends

This section displays trending information for your compliance policies.

Documentation Trends

This section displays trending information about your documentation.

4 Collections and Data Sources

This section contains:

Managing Data Sources

Data sources are managed in the webGUI for the specific data source. For example, if you want to manage a Cryptographic Security Platform Vault for CloudKeys, you would use the Cryptographic Security Platform Vault for CloudKeys webGUI. If you want to manage an HSM in the Cryptographic Security Platform Vault, you would use the KeySafe5 webGUI for that HSM.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Data Sources**.
3. Highlight the data source that you want to manage and click **Manage**.
   The webGUI for your data source will launch in a separate window.

Disconnecting or Reconnecting a Data Source Connection

You can disconnect or reconnect a data source connection in the Cryptographic Security Platform Compliance Manager webGUI.

Disconnecting a Data Source Connection

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to disconnect and click **View Details**.
4. Highlight the data source that you want to disconnect.
5. On the far right, click the **Disconnect** icon.
6. In the Disconnect Data Source window, click **Disconnect**.

Reconnecting a Data Source Connection

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to reconnect and click **View Details**.
4. Highlight the data source that you want to reconnect.
5. On the far right, click the **Reconnect** icon.

You will be taken to the Initialize Connection to a Data Source page for the data source that you previously disconnected.
6. Click **Create and Download File**.
7. On the Data Sources page, click the **Manage** link for the data source.
   You will be taken to the data source that you want to reconnect.
8. Log in to the data source, and click the Connect Now link at the top.
   Alternatively, you can click Settings and then Cryptographic Security Platform Compliance Manager.
9. Click **Connect** and upload the file that you just downloaded.
10. Click **Close** to close the window.

Moving a Data Source to a Different Collection

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to move and click **View Details**.
4. Highlight the data source that you want to move and click the **Move** icon.
5. In the Move Data Source to Collection window, select the collection where you want to move the Data Source and click **Move**.

Viewing Connected Data Source Connections

You can only edit data source details from the Collections page. To actually make changes to the data source, see Managing Data Sources.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to edit and click **View Details**.
4. Highlight the data source that you want to edit and click the **Edit** icon.
5. In the Edit Details window, modify the following:
   - Friendly Name—How this data source is referenced in the Cryptographic Security Platform Compliance Manager.
   - Description—An optional description to describe the data source.
   - Location—The best location to represent this data source on the world map. For US locations, you can select the closest city and state. For all other locations, select the closest city.

   Note: The friendly name and the location are displayed on the dashboard.
6. When you are finished, click **Apply**.

Editing Data Source Details

You can only edit data source details from the Collections page. To actually make changes to the data source, see Managing Data Sources.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to edit and click **View Details**.
4. Highlight the data source that you want to edit and click the **Edit** icon.
5. In the Edit Details window, modify the following:
   - Friendly Name—How this data source is referenced in the Cryptographic Security Platform Compliance Manager.

- Description—An optional description to describe the data source.
- Location—The best location to represent this data source on the world map. For US locations, you can select the closest city and state. For all other locations, select the closest city.

Note: The friendly name and the location are displayed on the dashboard.

6. When you are finished, click **Apply**.

Deleting a Data Source from a Collection

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that contains the data source that you want to delete and click **View Details**.
4. On the Data Sources tab, highlight the data source that you want to delete and click the **Delete** icon.
5. In the Delete Data Source window, click **Delete**.

Creating a Data Source Connection

Initializing a connection to a data source allows the Cryptographic Security Platform Compliance Manager to import metadata from that data source.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Data Sources**.
   Note: For Cryptographic Security Platform Vault, the data sources are the individual vaults, for example, the Cryptographic Security Platform Vault for KMIP. If you want to add the Cryptographic Security Platform Vault appliance, select **Appliance Clusters** from the side menu bar.
3. Select **Actions > Initialize Connection**.
   If you do not already have a data source connected, click **Initialize Connection** on the Data Sources page.
4. On the Initialize Connection to a Data Source page, enter the URL for the data source that you want to add.
5. Click **Create and Download File**.
   The Cryptographic Security Platform Compliance Manager downloads a file in .JSON format that will be used to connect to your data source.
   Important: This file should be considered a password. Please save it in a safe location in case you need to reconnect the same data source.
6. Log into the data source that you created the connection to.
7. At the top of the page, you will see a warning that you are not connected to Cryptographic Security Platform Compliance Manager. Click **Connect Now**.
8. Click **Connect** and upload the JSON file that you downloaded from Cryptographic Security Platform Compliance Manager.
9. Click **Close** to close the Connect CSP Compliance Manager window.

Deleting a Collection

You can only delete a collection if there are no data sources in the collection, and it is not the default.

Note: The Default Collection can not be deleted.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight a collection that you have created and click **View Details**.
4. Click the **About** tab and then click **Delete Collection**.

Editing a Collection

You can edit the details of a collection and the data sources contained in that collection.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight a collection that you have created and click **View Details**.
4. Click the **Data Sources** tab, mouseover one of the data sources, and click one of the following icons:
    - Disconnect / Reconnect—See Disconnecting or Reconnecting a Data Source Connection.
    - Move—See Moving a Data Source to a Different Collection.
    - Edit—See Editing Data Source Details.
    - Delete—See Deleting a Data Source from a Collection.
5. Click the **About** tab and update the collection name or description.
6. Click **Apply**.

Creating a Collection

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Click the **Create Collection** button.
4. Enter the name of the collection. This name can be modified later.
5. Enter an optional description to help identify the collection.
6. Click **Create**.

Viewing Collections

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Collections**.
3. Highlight the collection that you want to view and click **View Details**.
    - The Data Sources tab displays all of the connected data sources in this collection. From here you can move a data source to a different collection, edit a data source, or delete an existing data source.
    - The About tab displays the collection name and description. From here you can delete a collection as long as no data sources are in the collection, and it is not the default.

About Collections and Data Sources

Data sources are organized in Cryptographic Security Platform Compliance Manager in Collections. Each collection can have one or more data sources connected to it. All data sources are added to the Default Collection when they are first connected.

Data sources include the entities that are being analyzed for compliance, including CSP Vaults (CloudKeys, Secrets, KMIP, etc.) or File Encryption. Data sources are not managed in the Compliance Manager, however, when a data source is connected, information (metadata) about the cryptographic assets within the data source is synced with the Compliance Manager every 8 hours.

5 Cryptographic Assets

This section contains:

Sending Documentation Requests

You can send an email directly to the owner of a cryptographic asset to ask them to complete the documentation.

Note: If email notifications are not enabled, then

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Cryptographic Assets**.
3. Select the cryptographic asset for which you want to send a documentation request.
4. Select the Documentation tab.
5. In the Ask Owner field, enter the email address of the owner of the cryptographic asset.

The owner of the cryptographic asset receives an email with a direct link to the documentation page for this cryptographic asset. When the owner clicks the link, they are taken directly to a Cryptographic Security Platform Compliance Manager page that does not require a login.

Note: If email is not enabled, the Ask Owner field will not be visible.

Documenting Cryptographic Assets

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Cryptographic Assets**.
3. Select the cryptographic asset that you want to document.
4. Click the documentation tab.
5. Fill in the documentation and click **Submit**.

Viewing Cryptographic Assets

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Cryptographic Assets**.
   The Cryptographic Assets page displays the cryptographic assets based upon the filters above. You can filter on collection, vault, or type of cryptographic asset as well as use the search window to narrow down your results. The columns displayed are:
   - Name—The name of the cryptographic asset.
   - Collection—The collection to which the data source belongs.
   - Data Source—The vault to which the cryptographic asset belongs.
   - Status—The status of the cryptographic asset.
   - Documented—Displays the completion percentage of the of the documentation, or No Template.
   - Compliance—Displays the compliance percentage of the cryptographic asset, or Unassessed if the cryptographic asset has not been assessed.

- Age—Displays the age of the cryptographic asset.
- Risk—Displays the risk score for the cryptographic asset.

3. Select the cryptographic asset that you want to view. The cryptographic asset page displays the following tabs:
   - **Details**—Displays information about the cryptographic asset, such as when it was created and any the identifying information. For example:
     - Data Source—The name of the data source that the cryptographic asset belongs to.
     - Collection—The name of the Collection the cryptographic asset belongs to.
     - HSM Protected—Whether or not the cryptographic asset is protected by an HSM.
     - Dormant—Whether or not the cryptographic asset is active in a Cryptographic Security Platform Vault. When cryptographic assets are deleted, Cryptographic Security Platform Compliance Manager retains the metadata for 6 months. It takes a few days after an object is deleted in a Cryptographic Security Platform Vault to show up as Dormant.
     - Expire Action—The action, if any, to be taken when the cryptographic asset expires.
     - LastSync—The date and time that Cryptographic Security Platform Compliance Manager was last synced with the Cryptographic Security Platform Vault. The sync occurs every six hours.
     - Status—The status of the cryptographic asset.
       Note: Dormant cryptographic assets may still be listed as Active or Available if that was what the last synced metadata showed.
     - Compliance Status—Whether compliance has been run against the cryptographic asset.
     - Is Documented—Whether or not documentation has been created for the cryptographic asset.

     Note: This list is not complete.
   - **Compliance**—Displays all of the compliance runs, the score, the policy and schedule used, and the status.
   - **Documentation**—Displays the documentation template that was assigned. If there is no template, click the **View vault documentation** link to add a template.
     Whether or not there is a template, there is a Documented or Not Documented box telling you if the cryptographic asset documentation has been completed. If there is documentation, you will also see an Ask Owner box that allows you to email this page to someone who can fill out the details.

About Cryptographic Assets

All objects contained within your data sources are called cryptographic assets. For each cryptographic asset, Cryptographic Security Platform Compliance Manager tracks the following information:

- Name—The name of the cryptographic asset.
- Collection—The collection that contains the data source where the cryptographic asset is located.
- Data Source—The name of the data source that contains the cryptographic asset.
- Status—The status of the cryptographic asset. This can be one of the following:
  - Available
  - Preactive
  - Deactivated
  - Disabled
- Documented—Whether or not the cryptographic asset has documentation. This can be one of the following:
  - Documented—A documentation template has been assigned to the cryptographic asset and it is completed.
  - Not Documented—A documentation template has been assigned to the cryptographic asset, but it is not completed.
  - No Template—No documentation template has been assigned to the cryptographic asset.
- Compliance—Whether or not the cryptographic asset has been assessed by a compliance policy.
  - Compliant—The cryptographic asset has been assessed and is compliant.
  - Not Compliant—The cryptographic asset has been assessed and is not compliant.
  - Unassessed—The cryptographic asset has not been assessed.

- Age—The age of the cryptographic asset in days.
- PQ Status—If PQ Safe, the cryptographic asset is considered to be secure against a cryptographic attack by a quantum computer.
- Risk—The risk score of the cryptographic asset.

6 Reports

This section contains:

Deleting a Report

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Reports**.
3. On the Manage Reports page, you can delete reports in one of two ways:
    - On the Reports tab, locate the report that you want to delete, and click the Delete icon at the end of the row.
      This deletes the report definition which includes the schedule. The historical reports will remain on the History tab, but no more reports will be run.
    - On the History tab, locate the report version that you want to delete, and click the Delete icon at the end of the row.
      This deletes the individual version of the report, but does not delete the report definition or any other versions of the report.

Editing a Report

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Reports**.
3. On the Manage Reports page, locate the report that you want to edit, and either click the name of the report or click the Edit button at the end of the row.
4. In the Edit Report window, you can update the following information:
    - The Name of the report.
    - The Description of the report.
    - Whether or not email is enabled.
    - If email is enabled, the email addresses of the people who will receive the report.
    - The collections where the data sources are found.
    - The data sources to be included in the report.
    - How long the report history should be retained.
    - The recurring schedule for the report.
    - The frequency when the report should be run.
5. Click **Apply** to save your changes.

Creating a Report

When you create a report the report definition remains on the Reports tab. When that report is run, any versions of the report will be located on the History tab.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Reports**.
3. On the Manage Reports page, in the About section, enter the following:

Field	Description
Type	Enter the type of report you would like to create. This can be one of the following: • Executive Summary • License
Name	Enter the name for the report.
Description	Enter an optional description for the report.
Email Report To	If email is enabled, and you would like to email the report, enter the email address.

4. Click **Next**.
5. On the Manage Reports page, in the Details section, enter the following:

Field	Description
Collections	Choose the collections that you would like to include.
Data Sources	Choose the Data Sources that you would like to include.
Report Retention	Enter the amount of time you would like to keep this report.

6. Click **Next**.
7. On the Manage Reports page, in the Schedule section, enter the following:

Field	Description
Recurring Schedule	Choose Active to set a recurring schedule or Inactive if you do not want to set a schedule. If you choose Inactive, you cannot select a frequency or time.
Frequency	Select how often the report should run. This can be one of the following: • Daily • Hourly • Weekly
at	Enter the time when you want the report to run. This depends on your Frequency selection. • Daily—Select the time that you want the report to run. • Hourly—Select how many hours to wait before you run the report again. • Weekly—Select the days of the week and the time that you want the report to run.

8. Click **Create**.

Viewing Reports

The Manage Reports page displays all existing reports in your system. The reports are a high-level overview suitable for a management summary. All reports are generated as a PDF. You can generate the following types of reports:

- Executive Summary Report
- Licensing Report

To access the Manage Reports page, select Reports from the side menu bar.

The Reports tab displays all reports that have not been deleted. On this page, you can:

- Create a new report.
- Use the Filter bar to narrow down your results.
- Click the report name to view the report details and edit the report.
- Click the Last Run link to view the report.
- Click the icons at the end of each row to run the report without a schedule, edit the report, or delete the report.

The History tab displays all the report versions that do not exceed the retention period. On this page, you can:

- Use the Filter bar to narrow down your results.
- Click the icons at the end of each row to download the report or delete the report version.

7 Compliance

This section contains:

Deleting a Schedule

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Compliance page, select the **Schedules** tab.
4. On the Schedules tab, locate the schedule that you want to run and click the **Delete** icon.

Editing a Schedule

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Compliance page, select the **Schedules** tab.
4. On the Schedules tab, locate the schedule that you want to run and click the **Edit** icon.
5. Make any changes that you require and click **Save**.

Manually Running a Schedule

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Compliance page, select the **Schedules** tab.
4. On the Schedules tab, locate the schedule that you want to run and click the **Run Now** icon.

Creating a Schedule

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Compliance page, select the **Schedules** tab.
4. On the Schedules tab, click **Create Schedule**.
5. Enter the name and optional description for the schedule.
6. Chose whether the schedule should be Enabled or Disabled.
   Enabled policies will run automatically per the schedule or manually. You cannot run a disabled policy.

7. In the Policy section, select the category and the policy that you want to run.
8. In the Vaults section, add the vault or vaults that you want to run the policy against. The vault must be compatible with the selected policy.
9. In the Schedule, select the date and time that you want the schedule to run.
10. Click **Create**.

About Schedules

Schedules will automatically run a compliance policy against the defined vaults at the time that you set. You can manually run a schedule at any time.

Deleting a Custom Compliance Policy

You can only delete custom compliance policies. System policies are read-only. The Edit and Delete icon for each custom policy is only visible when you highlight the policy.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Policies page, highlight the policy that you want to delete and click the **Delete** icon.

The policy is immediately deleted.

Modifying an Existing Compliance Policy

If you are modifying a custom policy, you can save it directly. If modifying a system policy, you must save it under a different name and then open the policy.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
3. On the Policies page, select the policy that you want to modify.
   Note: For system policies, click **Save As**, enter a new policy name and optional description, and click **Save**. Select the new policy from the Policies page.
4. Review the operations that exist for the policy. You can edit or delete the operations.
5. Click **Add Operations**.
6. In the Add Operation window, select the catalog or policy from which you would like to add operations. All operations available in the catalog or policy will be displayed beneath the policy.
7. Select the operations that you would like to add and click **Add** or **Add & Close**.
8. If you clicked **Add**, repeat the previous step. Click **Add & Close** when you are finished.
9. Edit the parameters for each operation that you have added.
10. When you have finished, click **Save**.

Creating a Compliance Policy

If you only want to modify operation parameter values in an existing system policy, we recommend that you save a copy of that policy instead of creating a new one.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance.**
3. Click the **Create Policy** button.
4. In the Create Policy window, enter the name of the policy. This name can be modified later.

5. Enter an optional description to help identify the policy.
6. Select the object type that will be included in the policy
7. Click **Add Operations**.
   The Cryptographic Security Platform Compliance Manager takes you to the Policies page for your new policy.
8. On the Policies page, click **Add Operations**.
9. In the Add Operation window, select the catalog or policy from which you would like to add operations.
   All operations available in the catalog or policy will be displayed beneath the policy.
10. Select the operations that you would like to add and click **Add** or **Add & Close**.
11. If you clicked **Add**, repeat the previous step. Click **Add & Close** when you are finished.
12. Edit the parameters for each operation that you have added.
13. When you have finished, click **Create**.

Post-Quantum Compliance

Compliance Manager now supports Post-Quantum Cryptography (PQC) with NIST aligned guidelines. The Entrust Post Quantum Reference Policy allows you to check if an asset is in compliance with post-quantum (PQ) cryptographic standards by validating the algorithms used. This policy contains the following operations:

- Verify Not Quantum Safe Cryptographic Algorithm—This operation checks to see if the algorithm being assessed is found in the Not Allowed PQ Algorithm list.
  - If the algorithm **is** on the list, then it is not allowed, and the operation will fail.
  - If the algorithm **is not** on the list, then it is allowed, and the operation will pass.
- Verify Quantum Safe Cryptographic Algorithm—This operation checks to see if the algorithm being assessed is found in the Allowed PQ Algorithm list.
  - If the algorithm **is not** on the list, then it is not allowed, and the operation will fail.
  - If the algorithm **is** on the list, then it is allowed, and the operation will pass.

The Allowed and Not Allowed lists have sample values in them, but you can edit them to meet your requirements.

Viewing Compliance Policies

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Compliance**.
   The Policies page displays. The Policies page contains a list of all policies and can be sorted by the following:
   - The name of the policy
   - The optional description of the policy
   - The type of object for which the policy is meant
   - Whether the policy is a System or Custom policy.
     Note: System policies are read-only. Open a system policy and click **Save As** to keep any changes that you make.
   - The number of operations contained in the policy
   - The date the policy was last updated
3. Select the policy that you want to view.
   The individual policy page displays the operations contained in the policy and whether they are enabled and/or configured. From here you can:
   - Click **Edit** to make changes to an operation. If the policy is a system policy, you'll need to save it as a custom policy to keep any changes.
   - Click **Delete** to remove the operation from this policy.

4. Click **Cancel** to return to the Policies page.

About Compliance

On the Compliance page, you can manage the compliance policies and schedules that you want to apply to your vaults. Entrust creates sample policies which include operations and parameters for those operations. There are two types of policies:

- System—System policies are created by Entrust as a reference and cannot be modified. However, you can save them as a custom policy.
- Custom—Custom policies include Best Practice policies that are created by Entrust and policies that you have created yourself. You can either create a policy by saving a system policy under a different name, or create a policy from scratch.

All policies can be run by adding them to a schedule. Schedules automatically run a selected policy against a specified vault at the time you choose. You can also choose to run a schedule immediately.

8 Documentation

This section contains:

Deleting a Custom Template

System templates cannot be deleted. The Edit and Delete icon for each custom template is only visible when you highlight the template.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select Documentation.
3. On the Templates page, highlight the template that you want to delete and click the **Delete** icon.

The template is immediately deleted.

Modifying a Template

System templates cannot be modified, however you can save them as a new template under a different name.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select Documentation.
3. On the Templates page, select the template that you want to modify.
4. On the template page, click **Edit Template**.
   On the template page, you can now add or remove metadata, edit fields, or delete fields.
   Note: If a field is referenced by a compliance policy, you cannot edit the field.
   - Click **Reorder Fields** to change the order that the fields are listed in.
   - Click **Add Field** to add a new field to the template.
5. If the template is a System template, click **Save As** and enter the new name and optional description of the template, then click **Save**.
6. If the template is not a system template, you can click **Save** to update the template with the existing name, or **Save As** to change the name.
7. Click **Cancel** to exit without saving.

Assigning a Documentation Template

Documentation templates are assigned to an individual vault.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Vaults**.
3. Select the vault for which to assign a documentation template.
4. Click the Documentation tab.
5. Select the documentation template and click **Apply**.

Creating a Documentation Template

Documentation templates contain information that helps you identify and run compliance reports against your cryptographic assets.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select Documentation.
3. Click the **Create Template** button.
4. In the Create Template window, enter the name of the template. This name can be modified later.
5. Enter an optional description to help identify the template.
6. Select the category type that will be included in the template.
   The Cryptographic Security Platform Compliance Manager takes you to the Templates page for your new template.
   On the Templates page, you can add metadata and user input fields to help identify the cryptographic assets.
7. In the Metadata field, select the metadata that you would like to add to the template. Click **x** to remove any metatdata that you do not want to use.
8. Click **Add Field**.
9. Choose one of the following:
   - Browse Catalog—Choose an existing field from the catalog and modify it to meet your specifications.
   - Custom Field—Create the fields you want from the provided options.
10. Follow the on screen instructions to create or modify your fields.
11. Click **Add** to add the field and create an additional field, or **Add & Close** to add the field and close the window.
12. When you have finished adding all of your fields, click **Save**.

Viewing Documentation Templates

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select Documentation.
3. On the Templates page, you can view all of the document templates have been created.
   System templates are created by Cryptographic Security Platform Compliance Manager, and cannot be modified. Custom templates can be edited or deleted.

About Documentation

Documentation allows you to track information about the cryptographic assets contained in each data source. The owners of the cryptographic assets will need to manually complete the documentation for their objects.

Cryptographic Security Platform Vault uses templates to determine which information must be collected for the cryptographic assets. By default, we supply several documentation templates that you can use as is or modify under a different name. You can also create a template from scratch.

9 Appliance Clusters

This section contains:

About Appliance Clusters

Appliance clusters are the cluster nodes that contain the Cryptographic Security Platform Vault that you want to monitor in Cryptographic Security Platform Compliance Manager.

Note: If you are creating a new cluster, or your cluster is degraded and you replace the affected node, the Cryptographic Security Platform Appliance Management webGUI may show that the cluster is healthy, but it may still take up to 30 minutes to restore all functionality.

Creating an Appliance Cluster Connection

Appliance clusters consist of all of the individual cluster nodes for the Cryptographic Security Platform Vault appliance. If you want to add an individual vault, for example, the Cryptographic Security Platform Vault for KMIP, select Data Sources from the side menu bar. You need to create a connection with the Appliance Cluster to view the cluster in the Cryptographic Security Platform Compliance Manager.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Appliance Clusters**.
3. Select **Actions > Initialize Connection**.
4. On the Initialize Connection to a Cluster page, enter the appliance URL for one of the nodes of a Cryptographic Security Platform Vault.
   Note: The appliance URL can be found by logging into the Cryptographic Security Platform Vault Management appliance, and then clicking Switch to Appliance Management in the top right corner. The URL should be similar to https://<IP_Address>/appliance/.
5. Click **Create and Download File**.
   The Cryptographic Security Platform Compliance Manager downloads a file in .JSON format that will be used to connect to your appliance cluster.
   Important: This file should be considered a password. Please save it in a safe location in case you need to reconnect the same appliance cluster.
6. Log into the Cryptographic Security Platform Vault that you created the connection to.
7. At the top of the page, you will see a warning that you are not connected to Cryptographic Security Platform Compliance Manager. Click **Connect Now**.
8. Click **Connect** and upload the JSON file that you downloaded from Cryptographic Security Platform Compliance Manager.
9. Click **Close** to close the Connect CSP Compliance Manager window.

Viewing Appliance Cluster Details

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Appliance Clusters**.
3. Select the cluster that you want to view.
   The Appliance Cluster Details window displays the following:

- the connection status
- the date and time of the last sync
- the IP address of the Cryptographic Security Platform Vault cluster
- the installed software version
- the date the connection was established

4. Click **Close** to close the window.

10 Compliance Manager Authentication and Settings

The Settings page consists of the following tabs:

- About—Allows you to view the name and UID of this Compliance Manager tenant and add an image.
- Members—Allows you to invite a user and assign them a role.
- Local Users—Allows you to manage local users.
- Authentication—Allows you to configure Active Directory or OIDC authentication.

This section contains:

Configuring OpenID Connect

Cryptographic Security Platform Compliance Manager supports user authentication through an OpenID Connect provider. Once enabled, you will log in using the OIDC provider.

Before You Begin

The OpenID Connect provider must be configured to accept the Cryptographic Security Platform Compliance Manager URIs for each node of the cluster. The URIs are located in the Cryptographic Security Platform Compliance Manager webGUI for each tenant. The Client ID and Client Secret are located in the OIDC provider. For more information, see External Authentication Providers.

Procedure

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Authentication** tab.
4. In the OpenID Connect section, click **Configure**.
5. In the Disable Local Authentication dialog box, click **Continue**.
6. Enter the following:

Field	Description
Name	A user-defined name for the OpenID Connect provider. Cryptographic Security Platform Compliance Managerdisplays this name on the button on the login dialogs.
Client ID	The organizational identity assigned by the OpenID Connect provider when you sign up for the service.

Field	Description
Client Secret	A cryptographic component used to secure the organization's access to the OpenID Connect provider. Important: This field is write-only. It will never be displayed again after it has been initially created. It can be reentered if necessary.
Base URL	Enter the base URL. For Entrust IDaaS, the base URL will be: https://<IDaaS server>/api/oidc
Admin Name	The name of the tenant administrator.
Admin Email	The email address of the tenant administrator.

7. Click **Apply**.

Reconfiguring Active Directory

After you have successfully configured Active Directory, you can reconfigure it. This will wipe out the current Active Directory settings and start a new configuration.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Authentication** tab.
4. On the Authentication tab, click **Manage** under Active Directory.
5. In the Authentication > Manage Active Directory section, click **Reconfigure AD Now**.
6. In the Details section, enter the following:

Field	Description
Configuration Method	Choose whether to use Automatic or Manual configuration.
Domain Name	The Domain name to use for account authentication.
Security	Choose None or SSL.
Service Account	The AD account that the tenant should use when logging into the AD server.
Service Account Password	The password for the Service Account.

7. Click **Next**.
8. If you selected Automatic configuration, do the following:
    a. In the Domains section, verify the domain that you want to use. The default domain is displayed with a star icon.

Important: Cryptographic Security Platform Compliance Manager automatically adds all of the discovered domain controllers and global catalogs, starting with the closest. If you have a large number, then this will be done in the background. If the domain that you want to use is not visible, and you do not want to wait, then we recommend that you complete the configuration process, then edit your AD configuration later.

b. Click **Next** and proceed to step 10.

9. If you selected Manual configuration, do the following:
   a. In the Domains section, click the **Edit** icon to add at least one domain controller and global catalog.
   b. Click **Add Domain Controller** in the Edit Domain bar.
   c. In the Add Domain Controller section, complete the following:

Field	Description
Name	Enter the name of the domain controller.
Port	Enter the port number for the domain controller.
User Search Context (Base DN)	Enter the Distinguished Name (DN) of the node where the search for users should start. For performance reasons, the base DN should be as specific as possible. For example, `dc=ldapserver,dc=com`.
Group Search Context (Base DN)	The Distinguished Name (DN) of the node where the search for Security groups should start. This option applies to AD Security groups being associated with a Cloud Admin Group.

   d. Click **Add Global Catalog** in the Edit Domain bar.
   e. In the Add Global Catalog section, complete the following:

Field	Description
Name	Enter the name of the global catalog.
Port	Enter the port number for the global catalog.
User Search Context (Base DN)	Enter the Distinguished Name (DN) of the node where the search for users should start. For performance reasons, the base DN should be as specific as possible. For example, `dc=ldapserver,dc=com`.
Group Search Context (Base DN)	The Distinguished Name (DN) of the node where the search for Security groups should start. This option applies to AD Security groups being associated with a Cloud Admin Group.

f. Click **Update**.
g. Click **Next**.

10. In the Administrator section, enter the following:

Field	Description
Domain	Enter the name of the domain.
User	Enter the name of the Active Directory user.

11. If you are using SSL for security, click **Next**. If not, proceed to step 13.
12. In the Certificates section, check the **Approve All Certificates** checkbox. This will automatically approve all of the certificates in the domain controllers. Approving a certificate will download and approve all related certificates.
13. Click **Complete**.

You are logged out of the Cryptographic Security Platform Compliance Manager webGUI and can log back in with your AD credentials.

Configuring Active Directory

If you want to specify AD-managed Security groups whose members will have access to Cryptographic Security Platform Compliance Manager, you must specify a Windows Active Directory (AD) server.

Cryptographic Security Platform Compliance Manager uses the same settings for both local account authentication and AD Security group authentication. You can specify up to two AD domain controllers for failover, but both controllers must manage the same AD domain.

Note: Configuring Active Directory will disable Local Authentication.

Procedure

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Authentication** tab.
4. In the Active Directory section, click **Configure**.
5. In the Disable Local Authentication dialog box, click **Continue**.
6. In the Details section, enter the following:

Field	Description
Configuration Method	Choose whether to use Automatic or Manual configuration.
Domain Name	The Domain name to use for account authentication.
Security	Choose None or SSL.
Service Account	The AD account that the tenant should use when logging into the AD server.

Field	Description
Service Account Password	The password for the Service Account.

7. Click **Next**.
8. If you selected Automatic configuration, do the following:
    a. In the Domains section, verify the domain that you want to use. The default domain is displayed with a star icon.
    Important: Cryptographic Security Platform Compliance Manager automatically adds all of the discovered domain controllers and global catalogs, starting with the closest. If you have a large number, then this will be done in the background. If the domain that you want to use is not visible, and you do not want to wait, then we recommend that you complete the configuration process, then edit your AD configuration later.
    b. Click **Next** and proceed to step 10.
9. If you selected Manual configuration, do the following:
    a. In the Domains section, click the **Edit** icon to add at least one domain controller and global catalog.
    b. Click **Add Domain Controller** in the Edit Domain bar.
    c. In the Add Domain Controller section, complete the following:

Field	Description
Name	Enter the name of the domain controller.
Port	Enter the port number for the domain controller.
User Search Context (Base DN)	Enter the Distinguished Name (DN) of the node where the search for users should start. For performance reasons, the base DN should be as specific as possible. For example, `dc=ldapserver,dc=com`.
Group Search Context (Base DN)	The Distinguished Name (DN) of the node where the search for Security groups should start. This option applies to AD Security groups being associated with a Cloud Admin Group.

    d. Click **Add Global Catalog** in the Edit Domain bar.
    e. In the Add Global Catalog section, complete the following:

Field	Description
Name	Enter the name of the global catalog.
Port	Enter the port number for the global catalog.

Field	Description
User Search Context (Base DN)	Enter the Distinguished Name (DN) of the node where the search for users should start. For performance reasons, the base DN should be as specific as possible. For example, `dc=ldapserver,dc=com`.
Group Search Context (Base DN)	The Distinguished Name (DN) of the node where the search for Security groups should start. This option applies to AD Security groups being associated with a Cloud Admin Group.

f. Click **Update**.
g. Click **Next**.

10. In the Administrator section, enter the following:

Field	Description
Domain	Enter the name of the domain.
User	Enter the name of the Active Directory user.

11. If you are using SSL for security, click **Next**. If not, proceed to step 13.
12. In the Certificates section, check the **Approve All Certificates** checkbox. This will automatically approve all of the certificates in the domain controllers. Approving a certificate will download and approve all related certificates.
13. Click **Complete**.

You are logged out of the Cryptographic Security Platform Compliance Manager webGUI and can log back in with your AD credentials.

Adding and Deleting Local Users

Local authentication is configured by default with the tenant administrator as a member.

Adding Local Users

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click **Local Users** to view your existing users.
4. Click **Create User**.
5. Enter the email address and the name for the new local user.
6. Click **Create User**.

Deleting Local Users

Note: For local authentication, you cannot delete a local user until it has been removed from the Members tab.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click **Local Users** to view your existing users.
4. Highlight the user that you want to delete and click the **Delete** icon.
5. In the Delete User window, click **Yes, Delete**.

Single Sign-On (SSO) Authentication

Cryptographic Security Platform Compliance Manager has added Single Sign-On (SSO) authentication. The account credentials that you set up for your specific authentication mode (local mode, OIDC, AD) for the Default tenant can now be used to log in to the Compliance Manager and Tenant Management appliances.

Note: When you log into the Compliance Manager and Tenant Management appliances, there are two options:

- If you have the Administrator role, you can login with your default credentials.
- If you are the secroot user, you can log in as secroot.

Adding and Removing Active Directory (AD) Members

A member can be either an administrator that can manage all collections or an auditor that only has view access of all collections. It is recommended to have at least two administrators.

Adding a Member

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Members** tab to view your existing members and their roles.
4. Click **Add Member**.
5. On the Add Member page, enter the AD domain, the name of the user that you want to add, and their email address.
6. Select whether the new member will be an administrator or an auditor.
7. If you selected Administrator, choose the additional roles that you would like to assign to this tenant.
    - Compliance Manager appliance and Tenant Management Administrator—Grants administrative access to the Appliance and Tenant Management applications.
8. Click **Add Member**.
   If SMTP is enabled, the new user will receive an email with the login URL for Cryptographic Security Platform Compliance Manager webGUI. If SMTP is not enabled, you will need to send the login URL to the new user.

Removing a Member

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Members** tab.
4. Highlight the member that you want to remove, and click the **Delete** icon.
5. In the Remove Member window, click **Yes, Delete**.

Inviting and Removing Local Authentication or OpenID Connect (OIDC) Members

A member can be either an administrator that can manage all collections or an auditor that only has view access of all collections. It is recommended to have at least two administrators.

> Note: If you are using local authentication, members must be existing local users. This means you must add a local user before you can invite them to be a member, and remove them as members before you can delete them from the Local Users page.

Inviting a Member

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Members** tab to view your existing members and their roles.
4. Click **Invite Member**.
5. On the Invite Member page, enter the name of the user that you want to add and their email address.
6. Select whether the new member will be an administrator or an auditor.
7. If you selected Administrator, choose the additional roles that you would like to assign to this tenant.
   - Compliance Manager appliance and Tenant Management Administrator—Grants administrative access to the Appliance and Tenant Management applications.
8. Click **Invite Member**.
   For Local Authentication: If SMTP is enabled, the new user will receive an email with the login URL and a temporary password for the Cryptographic Security Platform Compliance Manager webGUI. If SMTP is not enabled, you will need to send the login URL and temporary password to the new user.
   For OIDC: If SMTP is enabled, the new user will receive an email with the registration URL. This URL is active for 24 hours only, and the user must click on the URL and sign in with their credentials before logging in to the Cryptographic Security Platform Compliance Manager webGUI. If SMTP is not enabled, you will need to send the registration URL to the new user.

Removing a Member

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **Members** tab.
4. Highlight the member that you want to remove, and click the **Delete** icon.
5. In the Remove Member window, click **Yes, Delete**.

Adding an Image

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the side menu bar, select **Settings**.
3. Click the **About** tab.
4. Click **Select Image**.
5. Select the image that you want to use to represent this app.
6. Click **Apply**.

11 Licensing and Entitlements

This section contains:

About Licenses and Entitlements

When you first start using your Cryptographic Security Platform Compliance Manager tenant, a trial license is installed by default. The trial license is valid for 30 days, and includes all data sources that you add to your collections. All entitlements are tracked in the Cryptographic Security Platform Compliance Manager. For the Cryptographic Security Platform Vault, all vault-related enforcement occurs in the Cryptographic Security Platform Vault.

Important: Entitlements are only available through the Cryptographic Security Platform Compliance Manager webGUI.

Adding a License

To add a license, you must log in to a Cryptographic Security Platform Compliance Manager tenant.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the top-right corner, click the question mark (**?**) and select **License**.
3. Click the **Licenses** tab.
4. Click **Add License**.
5. Add your license in one of the following ways:
    - Import File
        i. Click the Import File radio button.
        ii. Click **Browse** and select the license file.
        iii. Click **Add**.
    - Enter License
        i. Click the **Enter License** radio button.
        ii. Paste the license into the text box.
        iii. Click **Add**.

Viewing Entitlements and Licenses

To view your entitlements and licenses, you must log in to a Cryptographic Security Platform Compliance Manager webGUI.

License counts are updated once daily, as well as:

- When a data source has been added to Cryptographic Security Platform Compliance Manager.
- When a Cryptographic Security Platform Vault cluster is deleted.
- When cryptographic assets are synced.
- When cryptographic assets are marked dormant.

Note: Cryptographic assets count differently depending on their state. For example, if a key has been rotated and has multiple versions, it will show up as one key in inventory, but the entitlement count is based on the number of keys that are pre-active, active, or available. Keys that have been archived or

disabled only count as 1/10th of a key. Key fractions are rounded down, so if you have 19 disabled cryptographic assets, it will count as 1 asset. 20 disabled cryptographic assets will count as 2 assets.

Procedure

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with your standard account credentials.
2. In the top-right corner, click the question mark (**?**) and select **Entitlements**.
   The Entitlements page displays the names of your entitlements, the limit, and how many are currently in use
3. In the Entitlements tab, you can view the names of your entitlements, the limit, and how many are currently in use.
4. In the Licenses tab, you can view the license activation keys, expiration date, and status.
5. Click the Activation key to view additional details about the license.

12 External Authentication Providers

This section contains:

Example: Configuring Azure OIDC to use with CSP Compliance Manager

The following example shows how to configure Azure OIDC to use with Cryptographic Security Platform Compliance Manager for External Authentication using Open ID Connect. You will need to register Entrust Cryptographic Security Platform Compliance Manager as a Microsoft Azure application, then copy the Client Secret, the Application (client) ID, and the base URL from Azure and paste it into Cryptographic Security Platform Compliance Manager.

1. In Microsoft Azure, click **App Registrations** in the sidebar.
2. Click **New Registration**.
3. In the Register an application page, enter the name to use for the application and enter the following redirect URIs.
    - Set the Login Redirect URIs for all nodes in the cluster to:
      `https://<IP or FQDN of Cryptographic Security Platform Compliance Manager node>/hytrust-authentication-rest/api/v2/login/`
    - Set the Logout Redirect URIs for all nodes in the cluster to:
      `https://<IP or FQDN of Cryptographic Security Platform Compliance Manager node>/hytrust-authentication-rest/api/v2/sso/logout/`
4. Click **Register**.
   After the application is created, you will be taken to the new application page.
5. Copy the Application (client) ID to a text window for later use.
   This will be the application (client) ID used in Cryptographic Security Platform Compliance Manager.
6. Click **Certificates & secrets** in the sidebar to create a client secret.
7. In the Client secrets section, click **New client secret**.
8. In the Add a client secret window, enter a description, set the expiration date, and click **Add**.
   The new client secret appears in the Client secrets section.
9. Copy the client secret value to a text window for later use.
   This will be the client secret used in Cryptographic Security Platform Compliance Manager.
10. Click **Token configuration** in the sidebar and then click **Add optional claim**.
11. In the Add optional claim window, select 'ID' for the token type and 'upn' and 'sid' for the claim.
12. Click **Add**.
13. In the pop-up window, check the 'Turn on the Microsoft Graph profile permission (required for claims to appear in token) checkbox.
14. Click **Overview** in the sidebar, and then click **Endpoints**.
15. In the Endpoints window, copy the OpenID Connect metadata document (up to and including the v2.0) to a text window to find the OpenID Connect URI or Issuer URI.
    For example, https://login.microsoftonline.com/a995284f-7628-4646-b755-ja0e3c7f0264/v2.0/. The '/.well-known/openid-configuration/' section is not needed.
    Note: This will be the base URL used in Cryptographic Security Platform Compliance Manager.
16. Close the Endpoints window to return to the Overview page.

Example: Configuring Entrust Identity as a Service

The following example is configuring Entrust Identity as a Service (IDaaS) to use with Cryptographic Security Platform Compliance Manager for External Authentication using OpenID Connect.

1. Log into Entrust IDaaS with your user name and one time password (OTP).
2. After you have logged in, click **Applications**.
3. Click the + icon to create a new Generic OpenID Connect and OAuth Cloud Integration.
4. Create a Generic Web Application using the following:
    - General Settings:
        - Copy and paste the Client ID and the Client Secret to a safe location. These will be used when configuring OIDC in Cryptographic Security Platform Compliance Manager.
        - Set the Token / Revocation Endpoint Client Authentication Method to Client Secret Post.
        - Set the Login Redirect URIs for all nodes in the cluster to:
          `https://<IP or FQDN of Cryptographic Security Platform Compliance Manager node>/hytrust-authentication-rest/api/v2/login`
          The URIs are located in the Cryptographic Security Platform Compliance Manager webGUI for each tenant.
        - Set the Logout Redirect URIs for all nodes in the cluster to:
          `https://<IP or FQDN of Cryptographic Security Platform Compliance Manager node>/hytrust-authentication-rest/api/v2/sso/logout`
          The URIs are located in the Cryptographic Security Platform Compliance Manager webGUI for each tenant.

      Important: If you use an IP address in the login and logout URLs, you can only log in to your Cryptographic Security Platform Compliance Manager tenant using the IP address. If you use the FQDN in the login and logout URLs, you can only log in to your Cryptographic Security Platform Compliance Manager tenant using the FQDN.
    - Authentication Settings:
        - Check the Require Consent checkbox.
        - Under Grant Types Supported, check the Authorization Code checkbox.
    - Supported Scopes
        - Select the Your unique identifier checkbox.
        - Select the Email address checkbox.

    Use the default for all other settings.
5. Add a resource rule to the AD Group so that the AD group and users from that AD group can access the application.

13 System Console

This section contains:

Exiting the Console

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Quit TUI session`.

Manage Timeouts and Appearance

In the Cryptographic Security Platform Compliance Manager System Console, you can set your TUI, GUI, and SSH session timeouts, as well as the TUI 3D appearance.

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Manage Timeouts and Appearance`.
3. On the Manage TUI Timeout and Appearance page, select one of the following and then select **OK**:
    - `Manage TUI Session Timeout`
      Allows you to change the TUI session timeout. The TUI timeout can be between 60 and 86400 seconds.
        i. Enter the new timeout value and select **OK**.
        ii. Review the new timeout value and select **OK** to return to the **Manage Timeouts and Appearance** page.
    - `Toggle TUI 3D Appearance`
      Selecting this option automatically updates the appears of the TUI.
    - `Manage GUI Settings`
      Allows you to change the Web Session (also called the GUI session) Idle Timeout. The default is 15 minutes. The maximum allowed is 1440 minutes.
        i. Enter the new timeout value and select **OK**.
        ii. Review the new timeout value and select **OK**.
        iii. Select Return to Previous Menu to return to the **Manage Timeouts and Appearance** page.
    - `Manage SSH Session Idle Timeout`
      Allows you to change the SSH Session idle timeout value. The default is 15 minutes.
        i. Enter the new timeout value and select **OK**. Enter 0 to disable timeouts.
        ii. Review the new timeout value and select **OK** to return to the **Manage Timeouts and Appearance** page.

Recover Compliance Manager Cluster

If your Cryptographic Security Platform Compliance Manager system is not responding, you can reset the cluster.

Important: This operation can take a long time.

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Manage Cryptographic Security Platform Compliance Manager Node`.
3. Select `Recover Compliance Manager cluster`.
4. Select `Yes` to proceed.

Delete CSP Compliance Manager Snapshots

Cryptographic Security Platform Compliance Manager automatically creates a pre-upgrade snapshot whenever you upgrade to a newer version. You can delete these pre-upgrade snapshots.

Important: If you delete the pre-upgrade snapshots, you cannot revert your upgrade to the previous installed version.

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Manage Cryptographic Security Platform Compliance Manager Node`.
3. Select `Delete Cryptographic Security Platform Compliance Manager pre-upgrade snapshot`.
4. In the confirmation screen, select **Yes**.

Change the secroot Account

The default user for the Cryptographic Security Platform Appliance Management webGUI is secroot. If the secroot user can no longer remember their credentials or is locked out, you can reset their credentials with a temporary password.

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Support Accounts`.
3. Select `secroot(Cryptographic Security Platform Compliance Manager webGUI default account)`.
4. Select one of the following:
   - If email is enabled, select **Generate random password and send it via email to secroot (Recommended)**.
     After receiving the email, the secroot user can log into the Cryptographic Security Platform Appliance Management webGUI and reset their password.
   - If email is not enabled, select **Enter new temporary password for secroot**.
     You will need to communicate the new password to the secroot user. After that, the secroot user can log into the Cryptographic Security Platform Appliance Management webGUI and reset their password.
5. Select `Return to previous menu` when you are finished.

Manage Read Only Support Access

The htrestricted account provides read-only access to debug Cryptographic Security Platform Compliance Manager.

Enabling the Support User

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Accounts`.

3. Select `htrestricted (read only support access)`.
4. Select Yes to enable the htrestricted account.
5. Enter and confirm a password for the Entrust support user.
   The password must be at least 8 characters long and must include at least one lowercase letter, one uppercase letter, one digit, and one symbol.
   Note: The password is only valid for 24 hours. The account is automatically disabled after 24 hours, or you can disable it when no longer needed.
6. Exit the console, and then log back in as htrestricted with the password that you created. You can also use an ssh client.

Entrust Support can now assist you further.

Disabling the Support User

The htrestricted user password will automatically expire in 24 hours. You can disable the support user at any time before that.

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Accounts`.
3. Select `htrestricted(read only support access)`.
4. Select Yes to disable the htrestricted account.

Manage Full Support Access

The htsupport user allows Entrust support to access your system. You can enable and disable the support user.

Enabling the Support User

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Support Accounts`.
3. Select `htsupport (full support access)`.
4. Select Yes to enable the htsupport account.
5. Enter and confirm a password for the Entrust support user.
   The password must be at least 8 characters long and must include at least one lowercase letter, one uppercase letter, one digit, and one symbol.
   Note: The password is only valid for 24 hours. The account is automatically disabled after 24 hours, or you can disable it when no longer needed.
6. Exit the console, and then log back in as htsupport with the password that you created. You can also use an ssh client.

Entrust Support can now assist you further.

Disabling the Support User

The htsupport user password will automatically expire in 24 hours. You can disable the support user at any time before that.

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Support Accounts`.
3. Select `htsupport (full support access)`.
4. Select Yes to disable the htsupport account.

Manage htadmin Account and SSH Access

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Manage htadmin and SSH Access`.

3. Select one of the following:

Option	Prompt	Description
1	Change htadmin Account Password	Select to change your htadmin password. a. Enter the current password at the prompt. b. Enter your new password. The password must be at least 8 characters long and must include at least one lowercase

Option	Prompt	Description
		letter, one uppercase letter, one digit and one of the following special characters: ~, !, @, #, $, %, ^, &, *, (,), -, +. c. Reenter your new password. If the password was changed successfully, the console will display: Password successfully updated. If not, you will receive a message stating why the change was denied.

Option	Prompt	Description
2	Manage SSH Access to System Menu	Select to allow SSH login access to the htadmin account. You can enable or disable access.
3	Manage SSH Session Idle Timeout	Select to modify the SSH session idle timeout value. The default is 15 minutes. Select 0 to disable timeouts.

4. When you are finished, select `Return to Previous Menu`.

Manage your Network Settings

You can modify your network settings for standalone nodes and the primary nodes of a cluster.

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. Select `Manage Network Settings` and press Enter.
3. Select one of the following options:

Option	Prompt	Description
1	Show Current Network Configuration	Select to view your Cryptographic Security Platform Compliance Manager current network configuration.

Option	Prompt	Description
2	Manage IP Address Settings	Select to change your current management interface. On the Interfaces page, select the network interface to configure and select OK. On the modify IP address settings, select Yes, and then update the settings that you want to change. Click OK, then Yes on the confirmation screen. Click OK to return to the Manage Network Settings page.
3	Disable a Network Interface	Select to disable a network interface.
4	Manage DNS Settings	Select to change your DNS settings. On the Modify DNS settings page, select Yes, and then update the settings that you want to change. Click OK, then Yes on the confirmation screen. You are automatically returned to the Manage Network Settings page.
5	Manage NTP Settings	Select to change your NTP settings. On the Modify NTP network settings page, select Yes, and then update the settings that you want to change. Click OK, then Yes on the confirmation screen. You are automatically returned to the Manage Network Settings page.

Option	Prompt	Description
6	Manage Static Routes	Select to manage your static routes. See Manage your Static Routes.
7	Network Diagnostic Tools	Select to test your network connectivity. See Test Your Network Connectivity.

4. When you are finished, select `Return to Previous Menu`.

Manage Your Static Routes

You can view, add, or delete static routes.

Listing your Current Static Routes

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Network Settings` and press Enter.
3. On the Manage Network Settings page, select `Manage Static Routes` and press Enter.
4. On the Manage Static Routes page, select `List Current Static Routes` and select OK.
5. The current static routes are displayed. Select OK to return.

Adding a Static Route

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Network Settings` and press Enter.
3. On the Manage Network Settings page, select `Manage Static Routes` and press Enter.
4. On the Manage Static Routes page, select `Add Static Route` and select OK.
5. On the Add a static route page, enter the network address and gateway of the static route to add.
6. Select OK to save and return.

Deleting a Static Route

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Network Settings` and press Enter.
3. On the Manage Network Settings page, select `Manage Static Routes` and press Enter.
4. On the Manage Static Routes page, select `Delete Static Route` and select OK.
5. On the Delete a static route page, enter the network address and gateway of the static route to delete.
6. Select OK to save and return.

Test Your Network Connectivity

Cryptographic Security Platform Compliance Manager attempts to connect to different devices in your network and displays the connection information.

1. Log in to the Cryptographic Security Platform Compliance Manager as htadmin.
2. Select `Manage Network Settings` and press Enter.
3. On the Manage Network Settings page, select `Network Diagnostic Tools` and press Enter.
4. On the Network Diagnostics page, select one of the following and follow the prompts:
    - `Verify DNS Server Response`
    - `Verify NTP Server Response`
    - `Test Remote Server is Reachable`

- `Test Inbound Ports of Another Server`

5. Select `Return to previous menu` when you are finished.

Setting Console Settings

If you do not remember the credentials of any user with the Security Administrator (`secroot`) privilege, or if you are locked out of the Cryptographic Security Platform Compliance Manager webGUI, Cryptographic Security Platform Compliance Manager provides a self-service option using the Cryptographic Security Platform Compliance Manager System Console to reset the `secroot`) user credentials with a temporary password. You can disable this option if you do not want this feature.

1. Log into the Cryptographic Security Platform Compliance Manager webGUI with with Security Admin privileges.
2. In the top right, click the Switch to **Appliance Management** link.
3. In the top menu bar, click **Settings**.
4. In the **System Settings** section, click **Console Settings**.
5. On the Console Settings page, select one of the following:
   - Select **YES** to allow the `htadmin` user to reset the `secroot` password.
   - Select **NO** if you do not want to allow the `htadmin` user to reset the `secroot` password.
6. Click **Apply**.

Using the CSP Compliance Manager System Console

1. Log in to the Cryptographic Security Platform Compliance Manager System Console as htadmin.
2. In the Cryptographic Security Platform Compliance Manager System Console, enter your selection at the prompt. This can be one of the following:

Option	Prompt	Description
1	Manage Network Settings	Allows you to update your network settings. See Manage your Network Settings. Note: You cannot change networking settings for a cluster member.
2	Manage htadmin and SSH Access	Allows you to modify your htadmin pasword and SSH access. See Manage htadmin Account and SSH Access.
3	Manage Accounts	Allows you to enable Support access or change the secroot password. See Manage Full Support Access, Manage Read Only Support Access, or Change the secroot Account .
4	Manage HSM Client Account	If you are connecting Cryptographic Security Platform Compliance Manager to multiple HSM servers, this option allows you to configure those servers as an HA group..

Option	Prompt	Description
5	Download Entrust Internal Certificate	Allows you to copy your CA certificate to a remote system. You will need to enter the remote system's hostname, username, and password.
6	Gather Diagnostic Logs	Creates a support bundle with diagnostic information and log files that Entrust Support can use to diagnose issues with your Cryptographic Security Platform Compliance Manager cluster.
7	Manage Cryptographic Security Platform Compliance Manager Node	Allows you to delete a Compliance Manager pre-upgrade snapshot or recover the kubernetes cluster. See Delete Compliance Manager Snapshots or Recover Cryptographic Security Platform Compliance Manager Cluster.
8	Reboot or Shut Down Cryptographic Security Platform Compliance Manager Node	Allows you to reboot or shut down your Cryptographic Security Platform Compliance Manager instance. Choose whether to shut down or reboot Cryptographic Security Platform Compliance Manager, and then select Yes.
9	Manage Timeouts and Appearance	Allows you to set your TUI, GUI, and SSH session idle timeouts. See Manage Timeouts and Appearance.
10	Quit TUI Session	Exits the console. See Exiting the Console.

Accessing the System Console

The Cryptographic Security Platform Compliance Manager System Console is a TUI (Text-based User Interface), and is accessed by logging in as htadmin.

You can access the Cryptographic Security Platform Compliance Manager System Console using one of the following:

- the vCenter Server Console where you installed Cryptographic Security Platform Compliance Manager
- an SSH session using the htadmin account

Note: SSH is disabled by default. To modify the settings, see Manage htadmin Account and SSH Access.

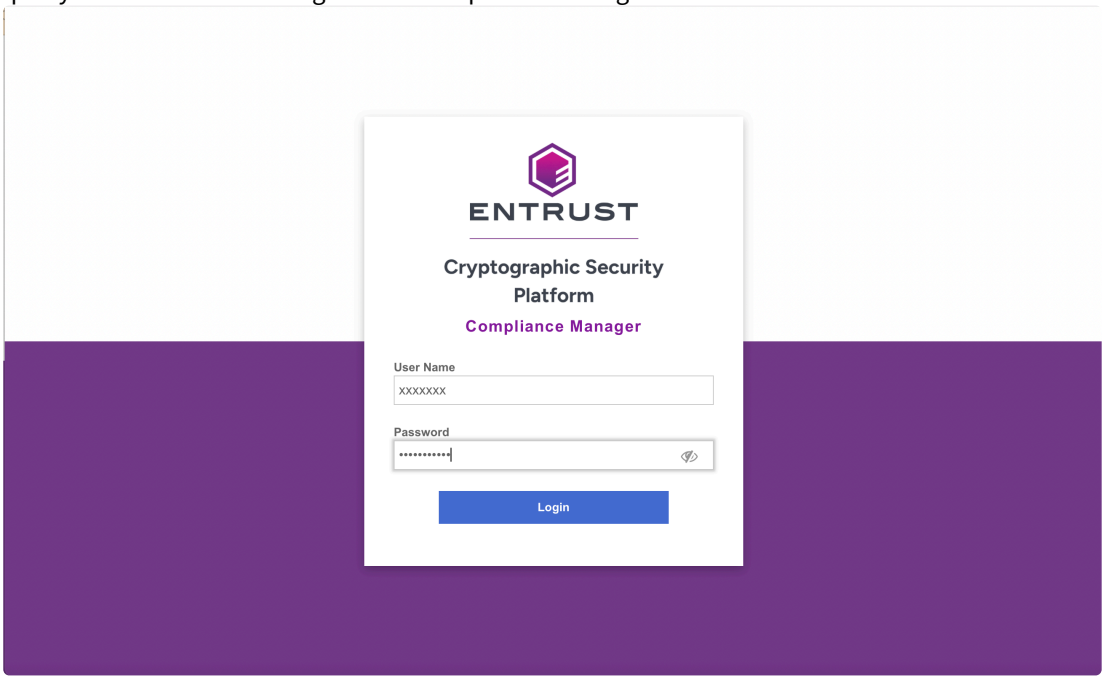
14 Using CSP Discovery

Entrust Cryptographic Security Platform Discovery is a comprehensive network discovery and scanning application that helps you understand what's running on your network, identify vulnerabilities, and manage your digital infrastructure.

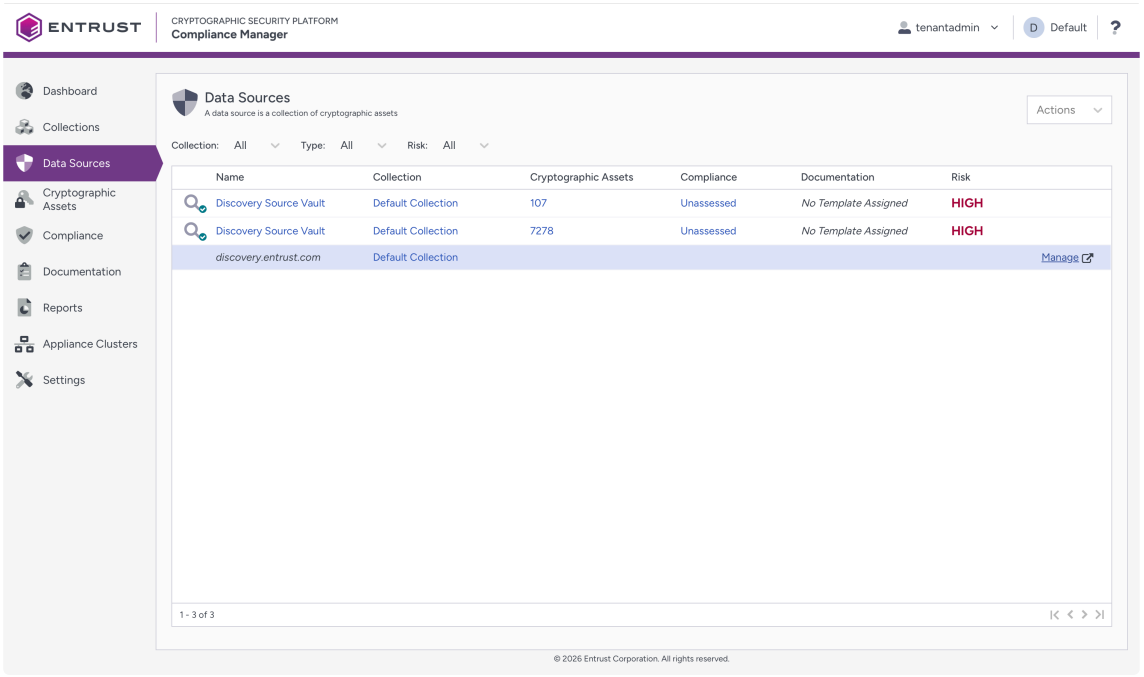
> CSP Discovery is only available for on-prem Compliance Manager installations.

To launch the Discovery application, follow the steps below:

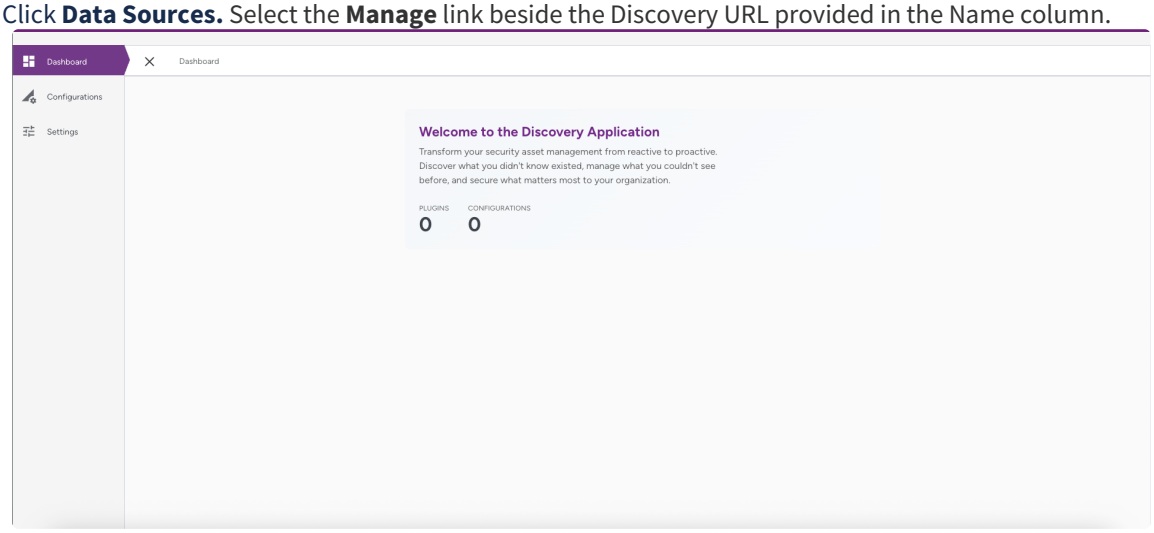
1. Open your web browser and go to the Compliance Manager URL.



2. Log in to the Entrust CSP Compliance Manager.



3. Click **Data Sources.** Select the **Manage** link beside the Discovery URL provided in the Name column.



4. Bookmark the URL if you'd like to access the Discovery application directly.
5. See below for the tool options.
    - Discovery Dashboard
    - Discovery configurations
    - Discovery Settings
    - Managing Discovery Plugins
    - Discovery Plugins Output
    - Plugin reference

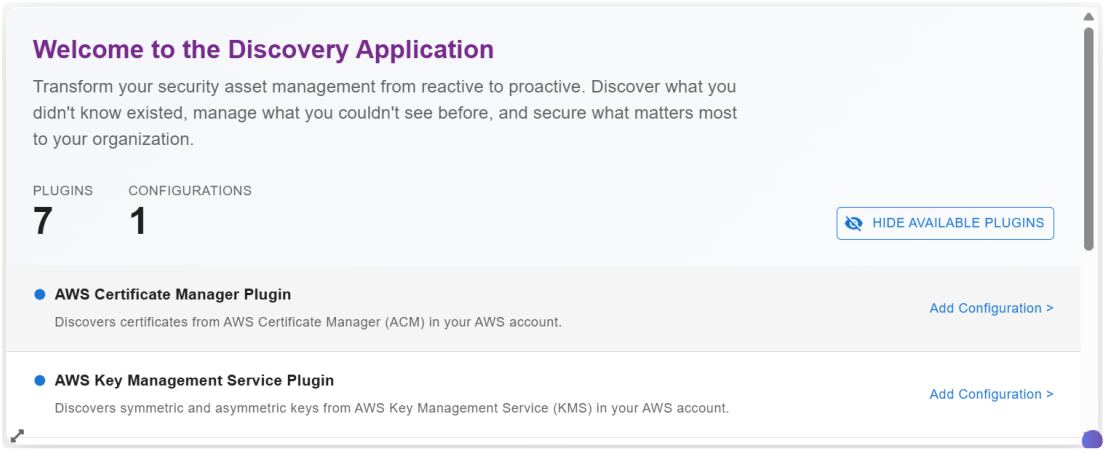
Discovery Dashboard

The Dashboard page provides an overview of the Discovery operations through informational widgets.

Changes made to the dashboard, such as widget height and width, layout, and options, are retained in your browser's memory; however, you can reset all the information by clicking the Reset Layout button.

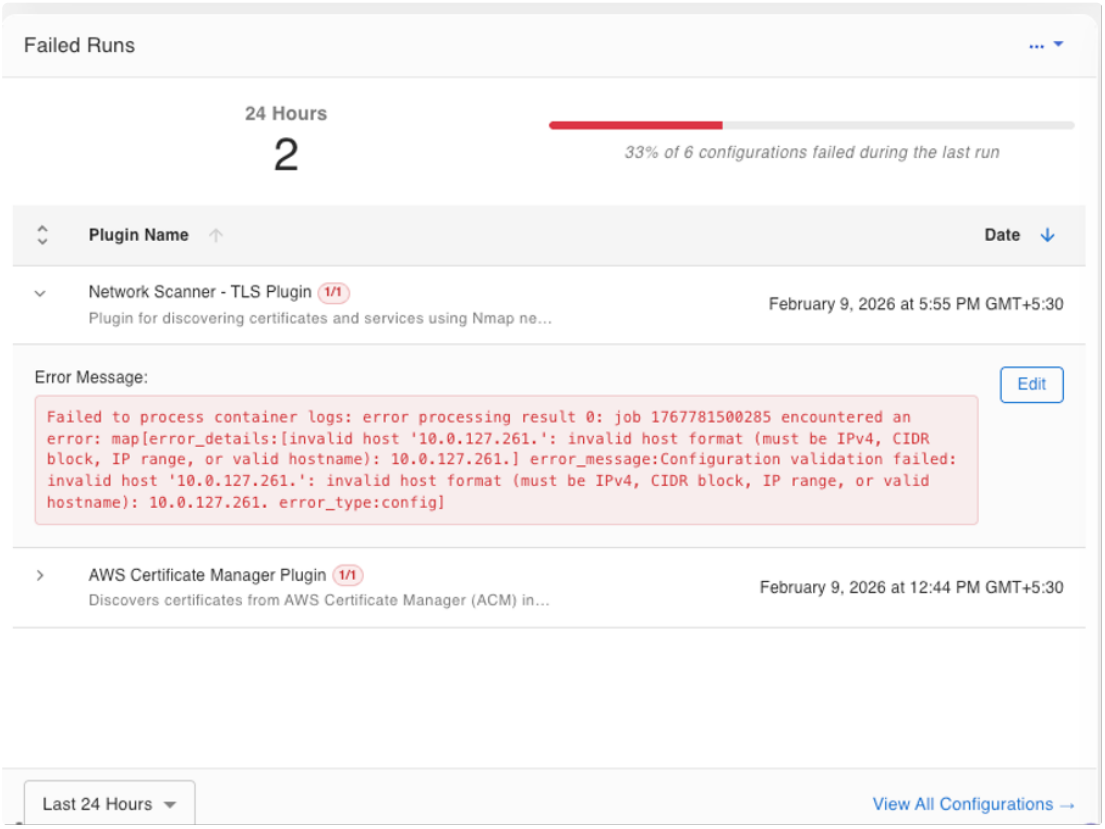
Welcome

The Welcome widget greets you when you first open your dashboard and helps you get started with configuration setup. Think of it as your launchpad for creating new configurations.



Failed Runs

This widget shows which configurations encountered issues during their most recent execution. It acts as a health monitor for your scans, alerting you when problems occur so you can resolve them quickly.

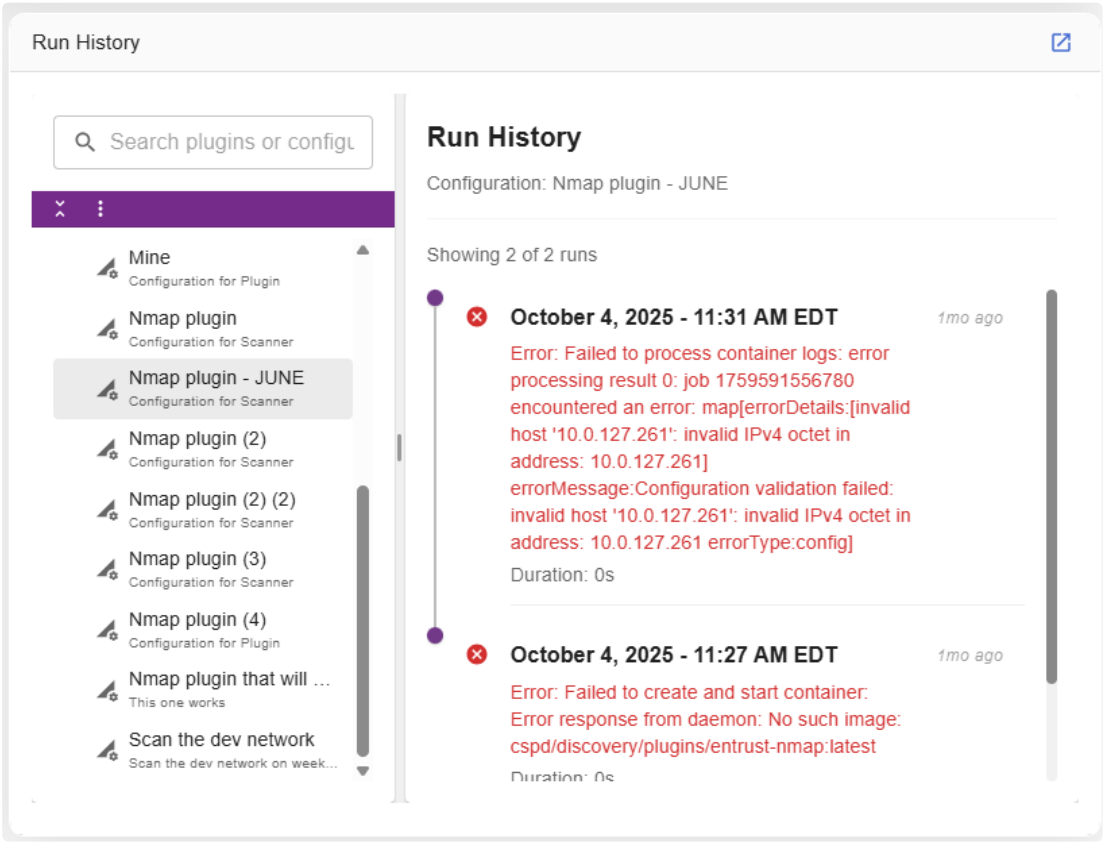


You can:

- Select the **>** beside the entry to view error details.
- Click the **Edit** button to modify that configuration.
- Change the date range.

Run History Widget

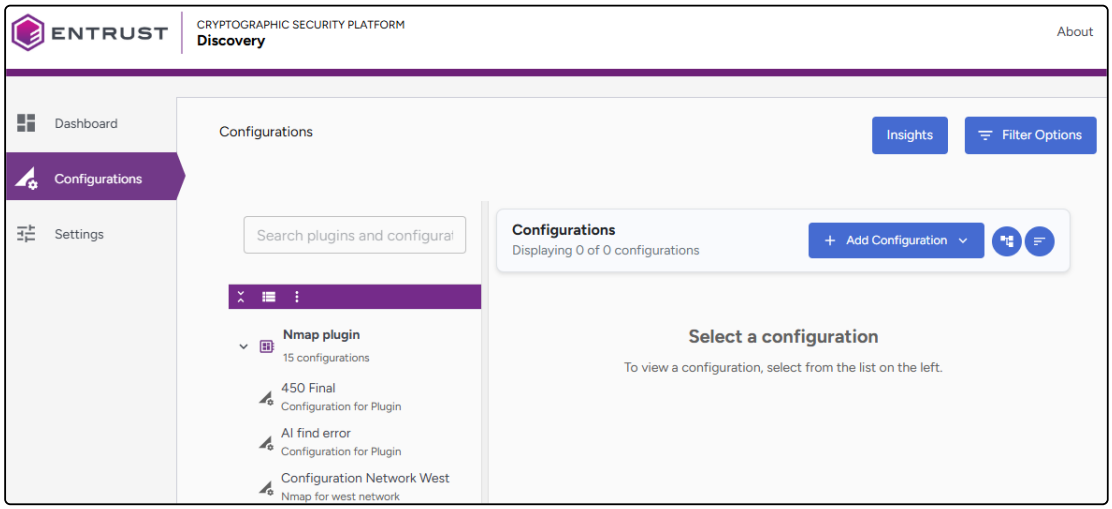
The Run History widget displays the complete execution history for all your configurations. It acts as a logbook and records every scan run. This includes when the run started, how long it took, whether the run succeeded or failed, and the totals of what was found. A scan will appear in this list only after it has finished running.



Discovery configurations

A configuration is a scan job that defines the plug-in to run, the plug-in's required parameters, and the scan schedule.

The Configurations page serves as the central hub for managing all configurations.

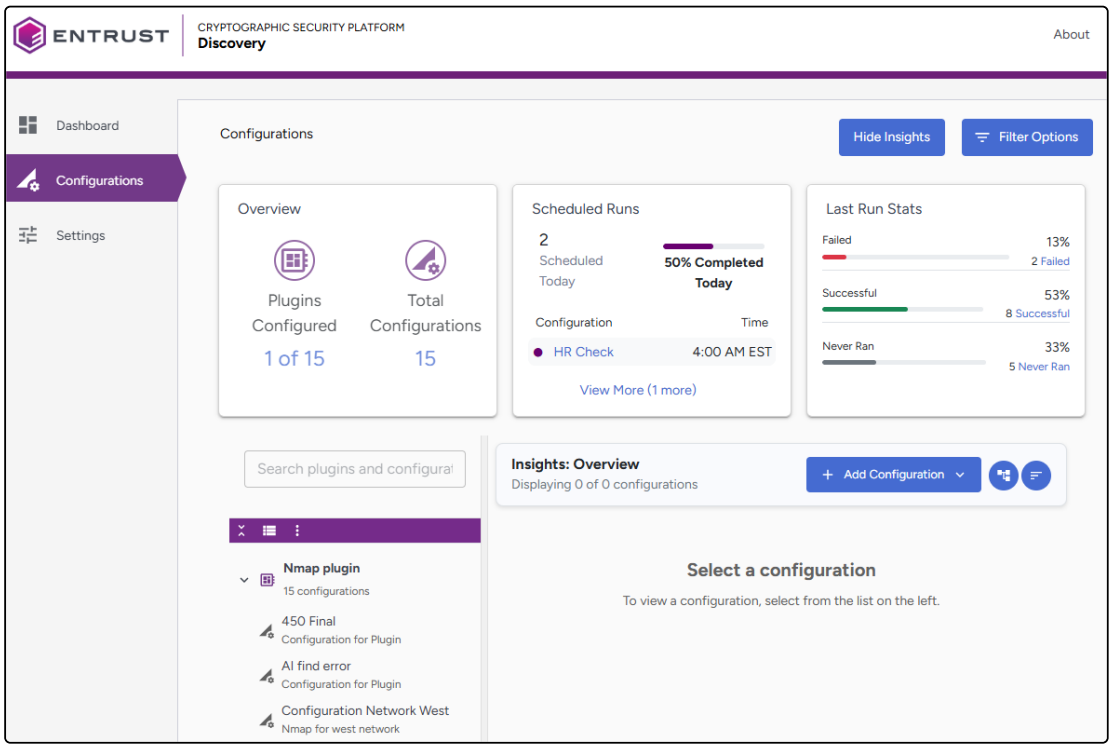


This page displays all available configurations in the system and allows you to create, edit, run, and monitor the configurations you have set up. For more information, see:

- Discovery Insights panel
- Discovery Filter options
- Discovery Content Area

Discovery Insights panel

Click the Insights button at the top to display a statistics dashboard that slides down above the configurations.



The following widgets are available.

- Overview
- Scheduled Runs
- Last Run Stats

Overview

The Overview widget displays two key metrics:

- **Plugins Configured: Displays the number of available plugins that have at least one configuration. For example, 1 of 15.**
- **Total Configurations: Displays the total number of individual configurations set up. For example, 15.**

When you select a plugin tree, the plugin tree will change accordingly.

Scheduled Runs

This widget displays today's upcoming scheduled runs. It displays the configuration names along with their scheduled run times.

Clicking a configuration name filters the plugin tree, allowing you to view more configuration information in the content panel.

Last Run Stats

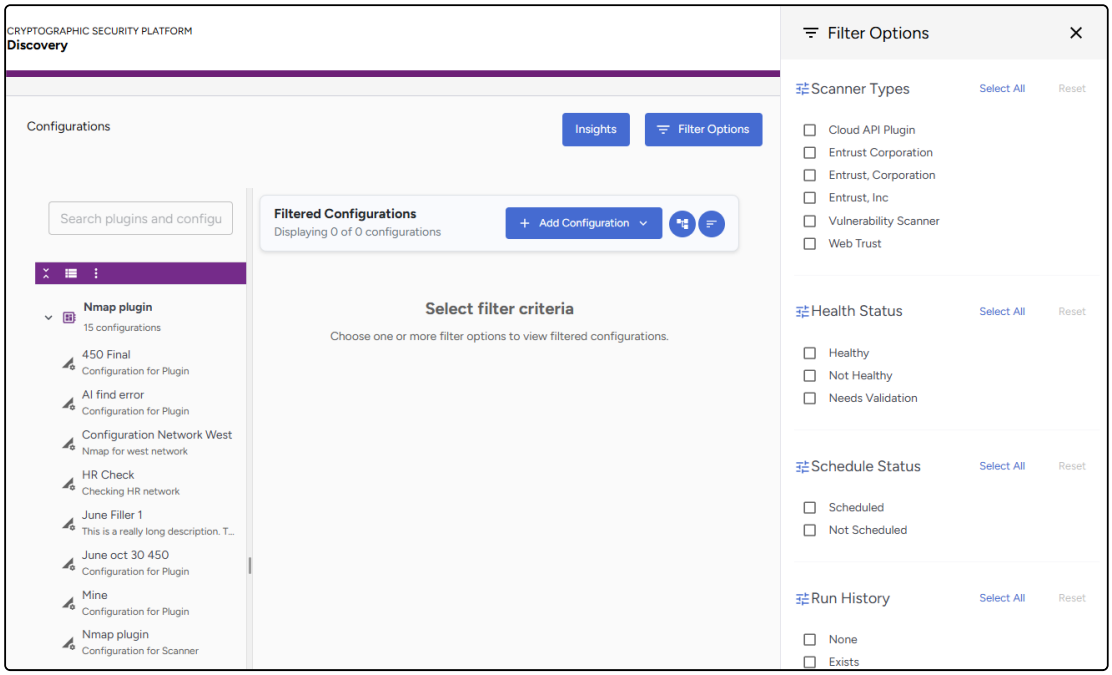
This widget displays execution results from recent runs in the following categories:

- Failed: Configurations that encountered errors during execution.
- Successful: Configurations that completed without errors.
- Never Ran: Configurations that have not been executed yet.

Click any category to filter the plugin tree and display only those configurations.

Discovery Filter options

Click the Filter Options button to open the available filtering options.



The following filters are available:

- Scanner Types
- Health Status
- Schedule Status
- Run History

After running a filter, you can perform actions such as:

- Check the active filter count on the **Filter Options** button.
- Click **Clear All Filters** inside the drawer to reset all applied filters.

Scanner Types

This filter allows you to filter by scanner type.

- Third-party scanners are identified by the plugin manufacturer's name.
- Built-in scanner plugins are identified by their category. For example, Cloud API, Vulnerability Scanner, Web Trust, etc.

Health Status

This filter allows you to filter by health status.

Status	Description
Healthy	Configurations passing connection tests
Invalid	Configurations with connection problems
Needs Validation	Configurations not yet tested

Schedule Status

This filter allows you to filter by schedule status.

Status	Description
Scheduled	Configurations with active schedules
Not Scheduled	No associated schedule

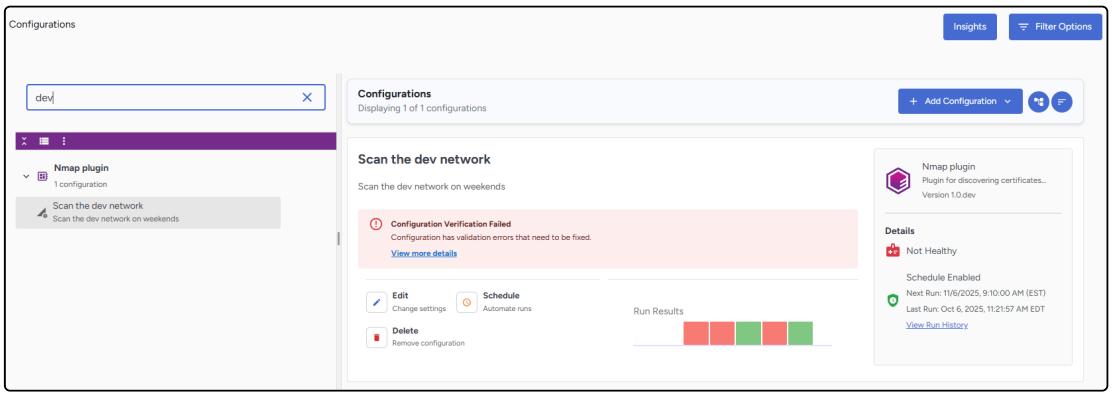
Run History

This filter allows you to filter by run history:

Status	Description
Scheduled	Configurations with active schedules
Not Scheduled	No associated schedule

Discovery Content Area

The main area consists of a browsable left panel (tree) that allows you to select configurations, and a right panel (content area) that provides detailed configuration information and available actions.



In the left panel, use the search box to quickly find specific plugins or configurations by typing their names. The tree filters instantly as you type. Click the X button to clear your search.

A selection of menus appears in the purple bar, allowing you to apply different views to the tree and configurations:

- When you display configurations by plugin name, you can quickly expand or collapse the entire tree.
- Click View All Configurations to see all configurations displayed on the right panel.
- You can view the items in the tree as follows:
  - Group By Plugin Name—Displays the plugins as expandable folders with configurations.
  - List by Configuration Name—Displays a flat, alphabetical list of all your configuration names without grouping.

When a Configuration is selected in the Left panel, the right panel displays the complete details, including:

- Configuration name
- Description,
- Action buttons to perform operations
- If applicable, a small graph containing the run results of the last 5 runs.

Position your mouse over the separate items in the graph to view the run date and total results.

The gray box on the right shows the plugin's name, description, and version. Further details are listed below:

- The health status
- Schedule information,
- If the configuration was run, a **View Run History** link will provide more information on all completed runs.

You can perform the following tasks:

- Editing an Existing Configuration
- Manually Running a Configuration
- Scheduling Automatic Runs
- Creating a New Configuration
- Deleting a Configuration

Editing an Existing Configuration

To edit an existing configuration:

1. Click the configuration name in the left tree.
2. Click the **Edit** button in the right panel.

3. Make your changes.
4. Test the connection again if you have changed connectivity settings.
5. Save your changes.

Manually Running a Configuration

To manually run a configuration:

1. Select the configuration in the tree
2. Click "Run Now" in the right panel.
   Note: This option is only available if the automatic configuration test has passed.
3. Confirm in the pop-up dialog.
4. Allow sufficient time for the configuration to complete, and check **Run History** to view the results.

Scheduling Automatic Runs

To schedule an automatic run:

1. Select your configuration.
2. Click **Schedule** in the right panel.
3. Enable the schedule toggle.
4. Set your schedule according to your local time.
5. Save your configuration to run automatically at the specified time.

Creating a New Configuration

To create a new configuration:

1. Click **Add New Configuration** in the header of the right panel.
2. Click a plugin name to select it.
3. Fill out the configuration form, including server addresses, credentials, scan settings, and other necessary details.
4. Test your connection to ensure settings are correct
5. Save the configuration.

Deleting a Configuration

To delete a configuration:

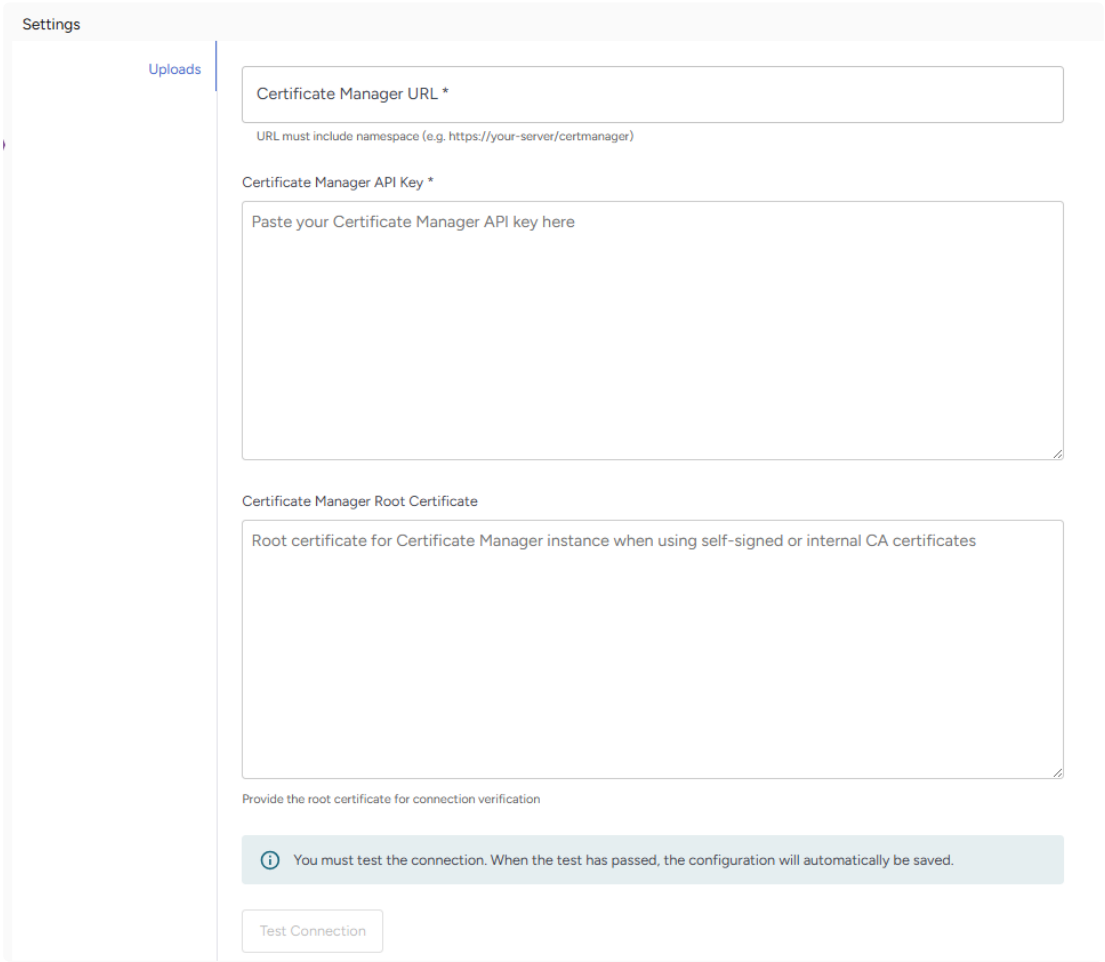
1. Select the configuration to delete.
2. Click **Delete** in the right panel.
3. Confirm the deletion in the pop-up dialog.
   The configuration is permanently removed.

Discovery Settings

This section explains how to configure system-wide Discovery settings and upload a plugin.

Uploads

This setting enables you to configure a connection to the Certificate Manager. Once configured, scan results are automatically forwarded to the Certificate Manager for further review and management.



Explanation of fields include:

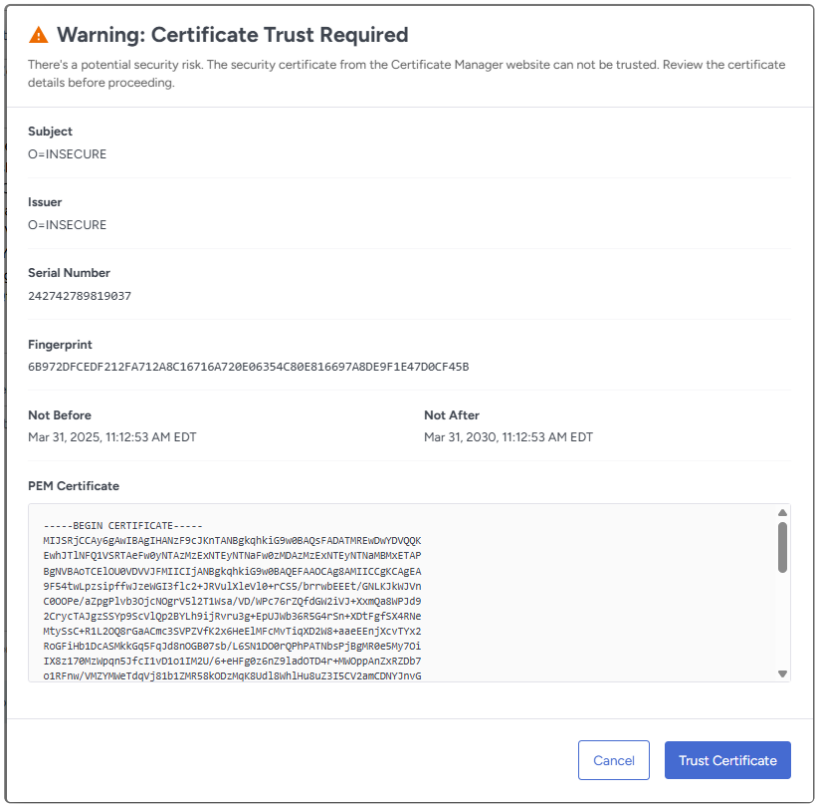
Field	Value
Certificate Manager URL	The web address of your Certificate Manager instance (must include the namespace path, not just the server address)
Certificate Manager API Key	Add your API authentication token. Get this token from your Certificate Manager administrator.

Field	Value
Certificate Manager Root Certificate (Optional)	Provide the root certificate if your Certificate Manager uses self-signed or internal CA certificates. Leave blank if using publicly trusted certificates.

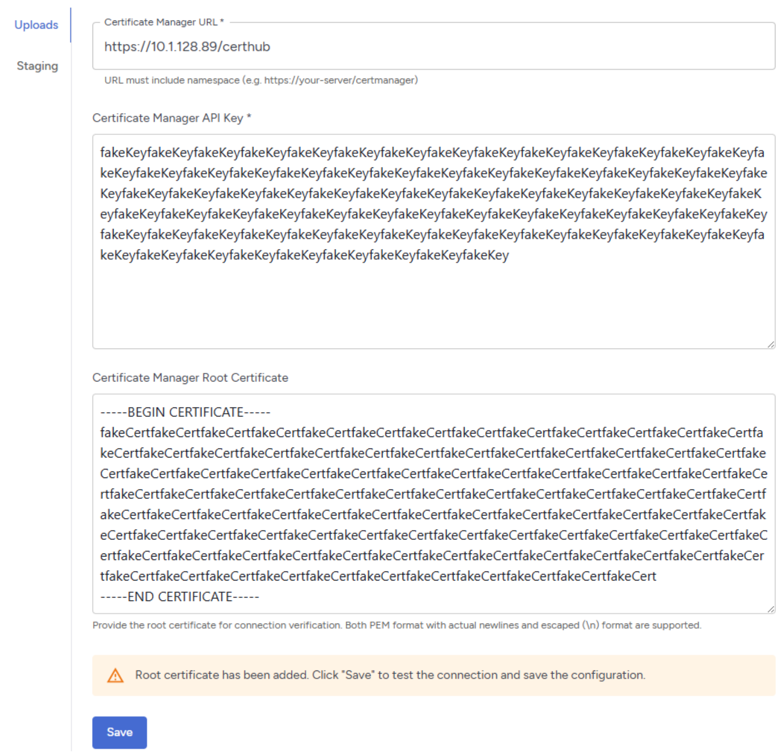
If SSL certificate verification is needed, a dialog appears showing certificate details:

- Certificate subject (who it's issued to)
- Issuer (who signed it)
- Serial number and fingerprint
- Valid dates (from/to)
- Full PEM certificate text

Review the certificate information carefully and click Trust Certificate.



Verify the connection is tested then Click Save.



> ⚠ You cannot save your configuration until you've tested the connection.

Staging

Follow the below steps to upload a plugin:

Creating the Plugin Tar File

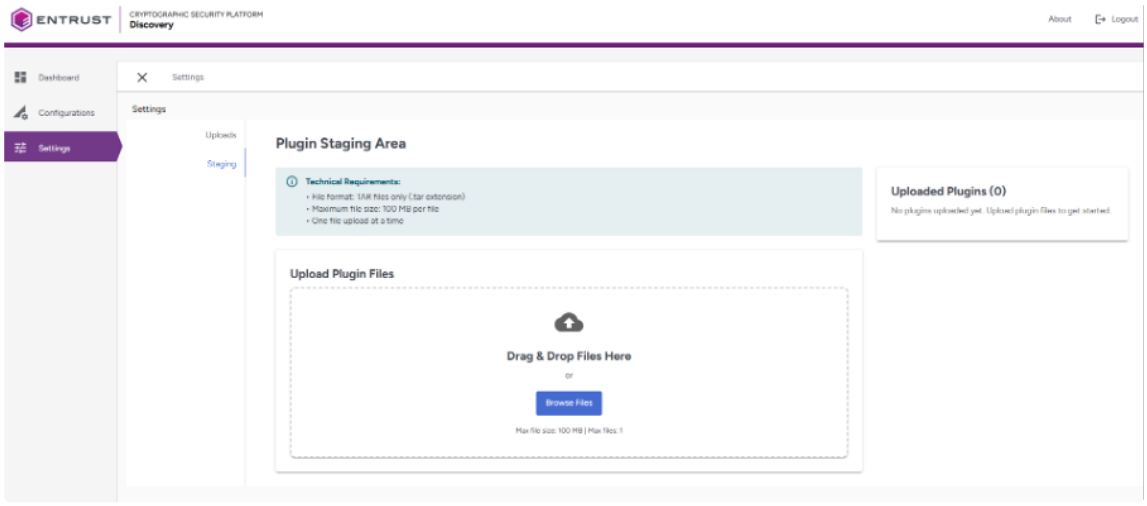
First, build your plugin Docker image according to the Discovery Plugin Spec. Then create a tar file:

```
# Example: Save your plugin image to a tar file
docker save my-plugin:1.0.0 -o my-plugin.tar
```

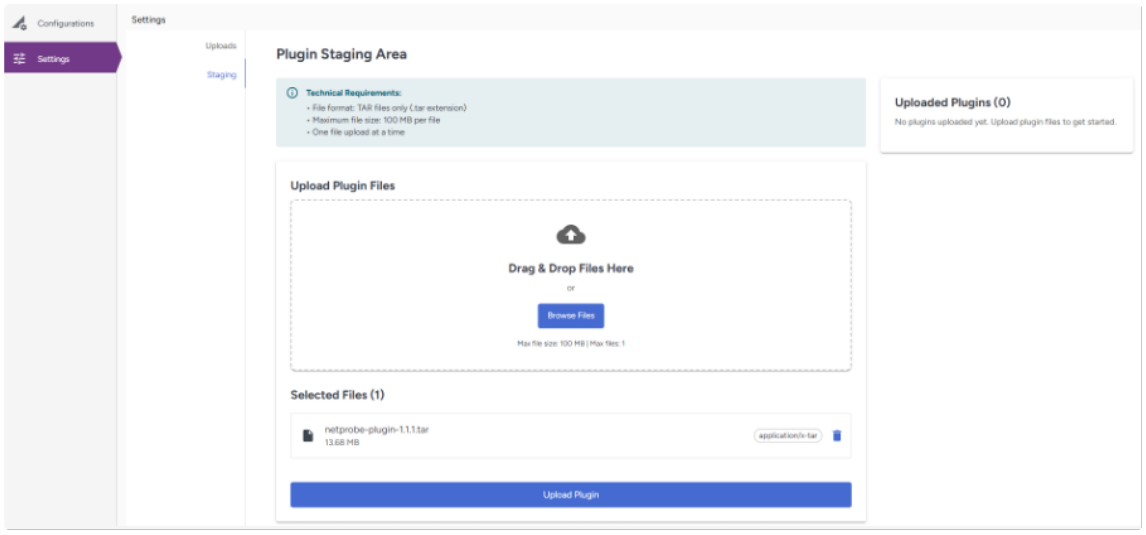
Note that the image name cannot begin with "cspd/discovery/", or the upload endpoint (below) will reject it. So, in the above example, my-plugin:1.0.0 is fine, but cspd/discovery/plugins/my-plugin:1.0.0 would be rejected. You must re-tag the image (and remove the un-allowed tag), then do the docker save.

1. Click the Staging tab, (where you will find the staging area to upload a plugin)

To select the plugin file, click the Browse File button or simply drag and drop the plugin file.

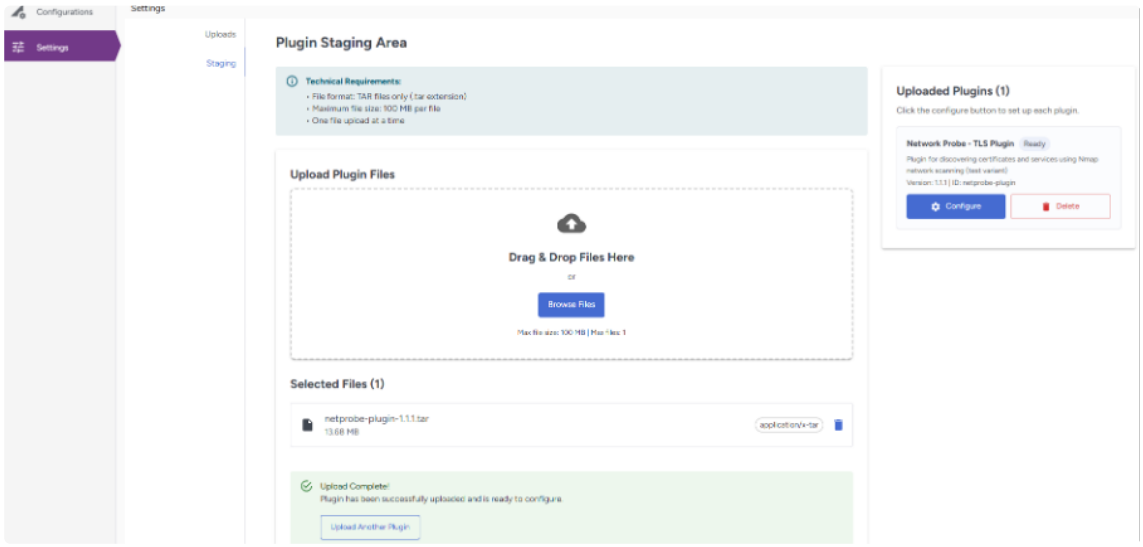


2. Once the plugin is selected, click the Upload Plugin button to proceed with the upload.



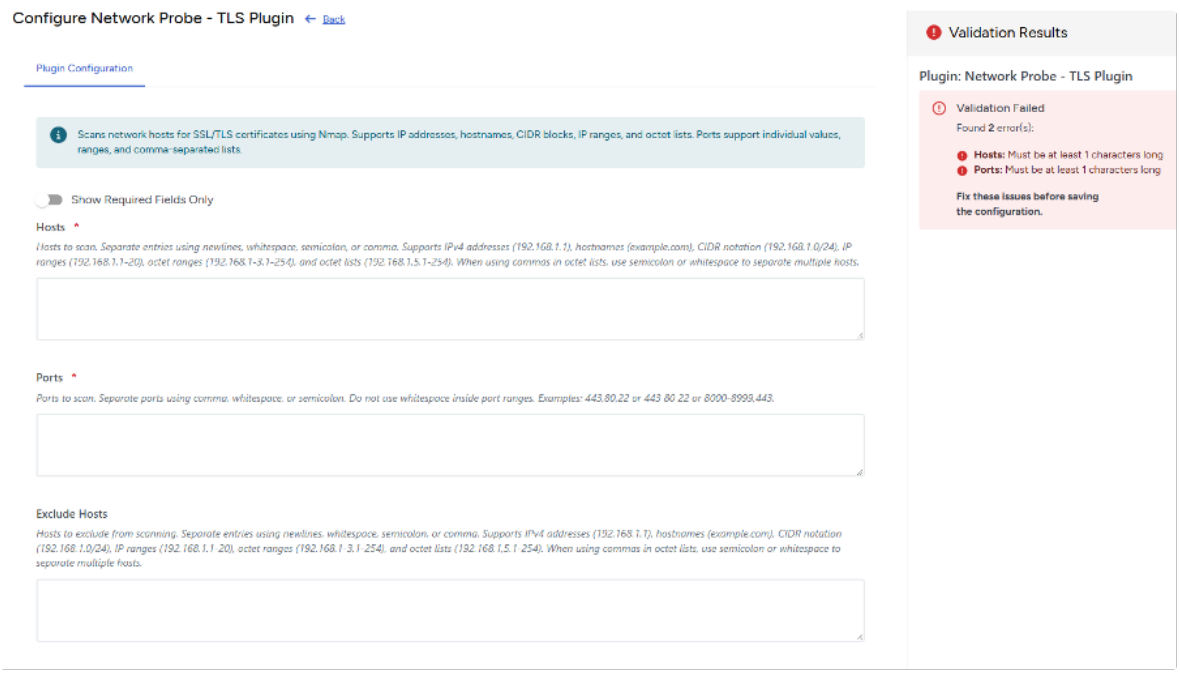
3. The plugin has been uploaded and is now visible in the right-hand panel.

4. To Configure a plugin, Click Configure.

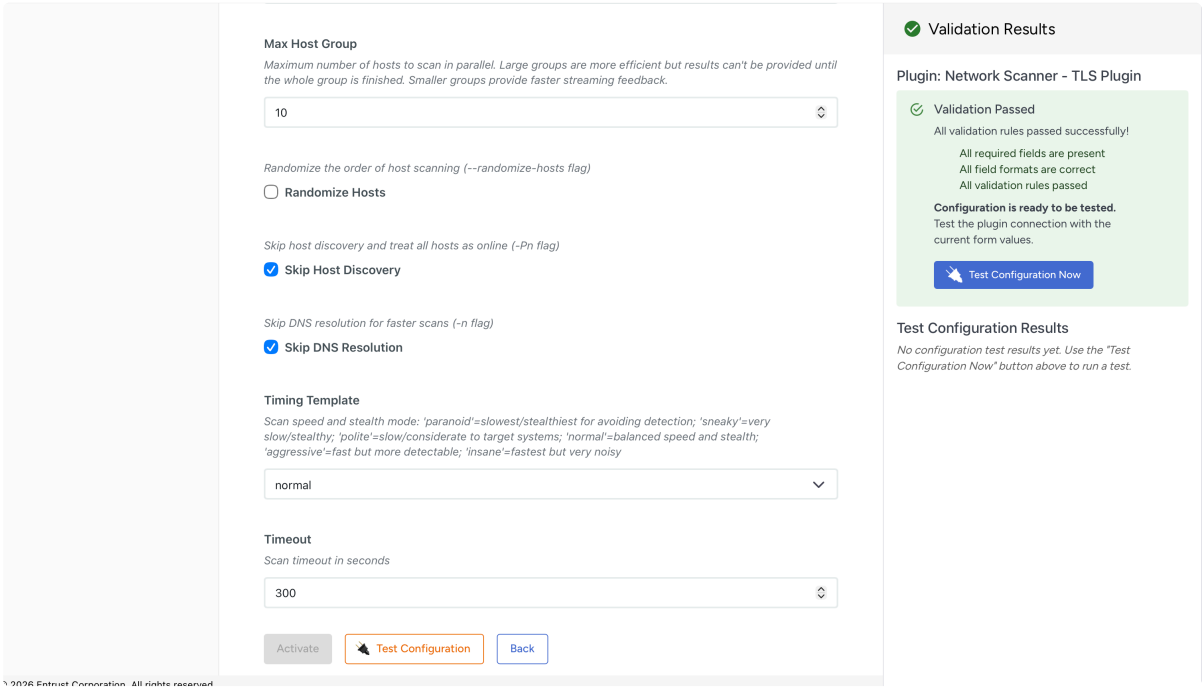


5. Enter the required details in the Hosts and Ports fields.

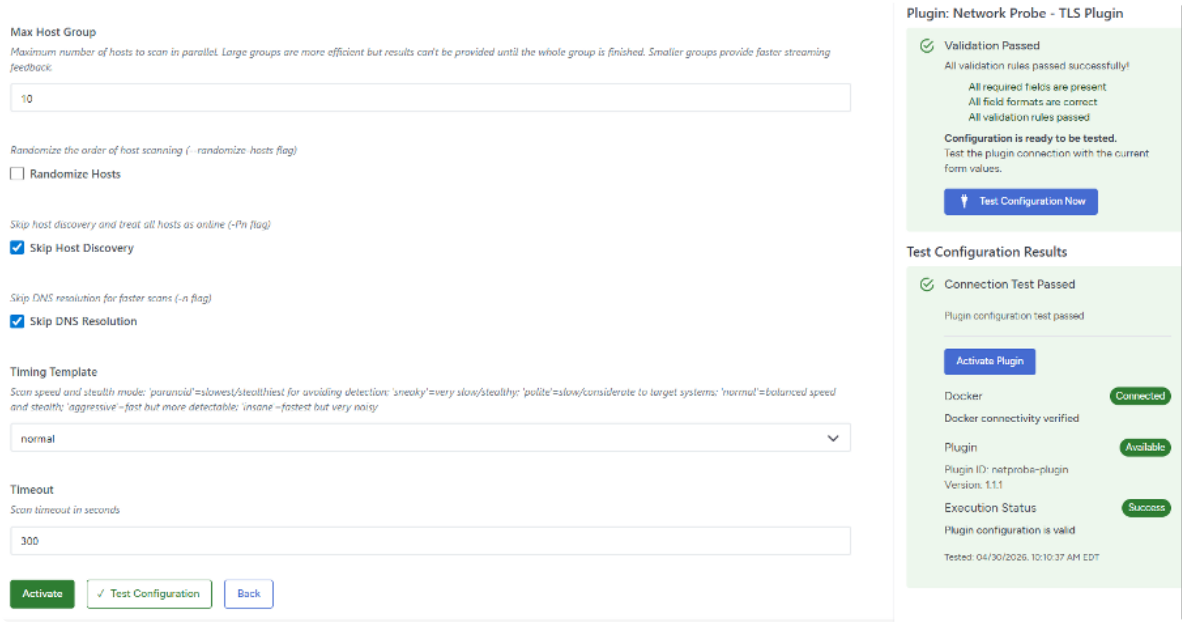
Note: To proceed with the next step, scroll down using the scrollbar.



6. Click Test Configuration button.

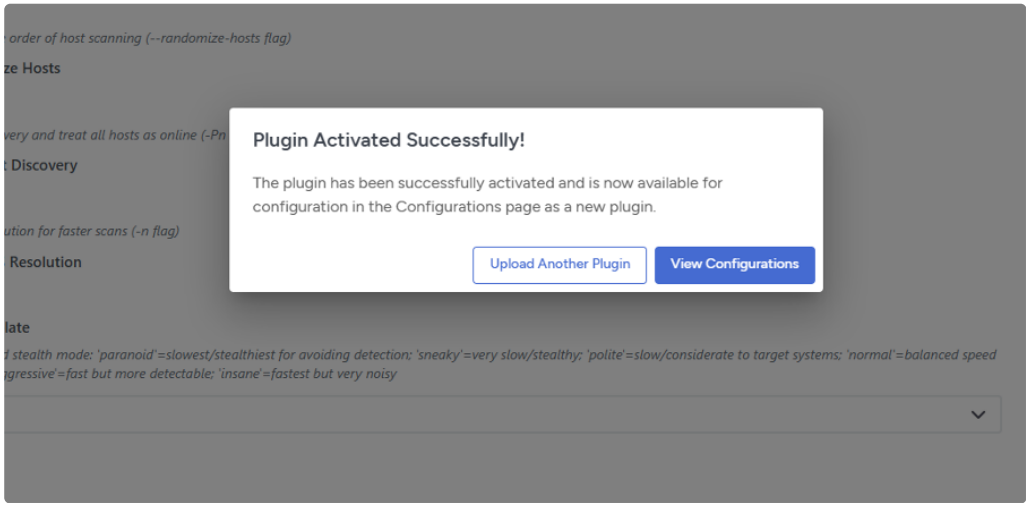


7. Once the test passed successfully, click Activate to make the configuration available for general use.



8. A confirmation message, 'Plugin Activated Successfully,' is displayed.

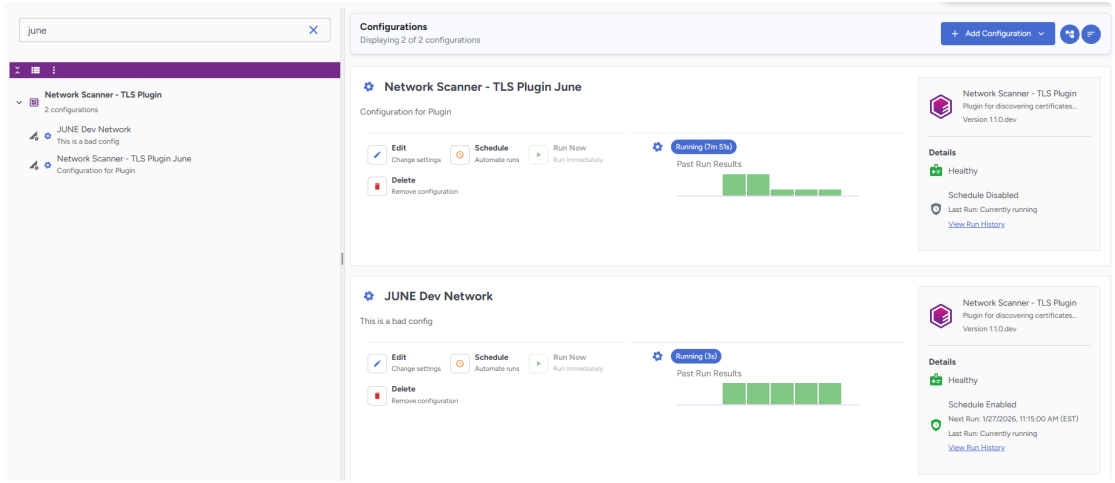
Note: User will notice that the uploaded plugin is no longer listed on the Staging page.



Managing Discovery Plugins

Plugin configuration is divided into the following sections.

- Plugin Configuration
- Schedule
- Validation Results



Click Edit for a configuration that is currently running and navigates to the Test Configuration Results section.

Plugin Configuration

In this tab, configure the scanner's operational parameters - target systems, authentication credentials, and scan-specific settings.

- AWS Certificate Manager plugin configuration
- AWS Key Management Service plugin configuration
- AWS Secrets Manager plugin configuration
- AWS CloudFront plugin configuration
- AWS Elastic Load Balancer plugin configuration
- Azure Key Vault plugin configuration
- GCP Certificate Manager plugin configuration
- GCP Key Management Service plugin configuration
- GCP Secret Manager plugin configuration
- HashiCorp Vault plugin configuration
- Network Scanner - TLS plugin configuration
- Network Scanner - OIDC plugin configuration
- Network Scanner - SSH Scanner plugin configuration
- Tenable Vulnerability Management plugin configuration
- Qualys CertView plugin configuration

> The interface adapts based on your selected plugin type, presenting only the relevant configuration options for that scanning methodology.

AWS Certificate Manager plugin configuration

To configure the AWS Certificate Manager plugin, see:

- AWS Certificate Manager plugin credentials
- AWS Certificate Manager plugin settings

For more information, see AWS Certificate Manager plugin.

AWS Certificate Manager plugin credentials

The AWS Certificate Manager plugin supports the following types of credentials:

- Creating permanent IAM user credentials
- Creating temporary credentials

Once set, run the following code in the AWS CLI to test the permissions.

```
# Test list permission
aws acm list-certificates --region us-east-1

# Test describe permission
aws acm describe-certificate --certificate-arn "arn:aws:acm:..." --region us-east-1

# Test get certificate permission
aws acm get-certificate --certificate-arn "arn:aws:acm:..." --region us-east-1
```

Creating permanent IAM user credentials

The IAM user or role must have the following ACM Permissions

acm:ListCertificates	Required to discover all certificates in the specified region. Without this permission, the plugin cannot enumerate certificates, and the scan will fail immediately.
acm:DescribeCertificate	Required to retrieve detailed metadata for each certificate, including domain names, validity periods, in-use status, and certificate authority. Without this permission, the plugin can list certificates but cannot retrieve their details.
acm:GetCertificate	Required to retrieve the actual PEM-encoded certificate body. Without this permission, the plugin can list and describe certificates, but cannot extract the certificate data itself.

To create permanent IAM user credentials

1. Log in to the AWS IAM Console at https://console.aws.amazon.com/iam/
2. Navigate to **Users** → **Add user**
3. Enter a username - for example: "acm-discovery-plugin".
4. Select **Programmatic access** as the access type
5. Click **Next: Permissions**
6. Attach the required permissions policy. For example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "acm:ListCertificates",
                "acm:DescribeCertificate",
                "acm:GetCertificate"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Complete the wizard and click **Create user**
8. Save the Access Key ID and Secret Access Key displayed on the final screen.
   Note: You won't be able to retrieve the secret key again.
9. Use these credentials in the plugin configuration and leave `sessionToken` empty.

Creating temporary credentials

For Temporary Credentials (STS), you may need the following permissions.

sts:GetSessionToken	Required only when using temporary credentials to generate or validate session tokens.

To create temporary credentials

1. Ensure you have an IAM role configured with the required permissions.
2. Use AWS CLI or SDK to assume the role:

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/ACMReadRole \
  --role-session-name acm-plugin-session
```

3. Extract the following fields from the response:
   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`
4. Use all three values in the plugin configuration.
   Note: Temporary credentials typically expire within 1-12 hours and must be refreshed.

AWS Certificate Manager plugin settings

This section includes:

Required AWS Certificate Manager plugin settings

The following configuration settings are required by the plugin:

- accessKeyId
  The AWS Access Key ID for authentication. This value is the first part of your AWS programmatic access credentials.
  Example:

```
AKIA1A2B3C4D5E6F7G8H
```

- secretAccessKey
  The AWS Secret Access Key for authentication. This is the secret part of your AWS programmatic access credentials and should be kept secure.
  Example:

```
abcdef1234567890abcdef1234567890abcdef12
```

- Region
  The AWS region to scan for certificates.
  Example:

```
us-east-1
```

Optional AWS Certificate Manager plugin settings

The plugin configuration supports the following optional settings.

sessionToken

An AWS session token for temporary credentials. This value is only required when using temporary credentials obtained through:

- AWS STS (Security Token Service) for role assumption
- MFA-protected operations
- Federated access

Example:

```
IQoJb3JpZ2luX2VjEBoaCXVzLWVhc3QtMSJHMEUCIQDExample+SessionToken+Data
```

AWS Key Management Service plugin configuration

To configure the AWS Key Management Service plugin, see:

- AWS Key Management Service plugin credentials
- AWS Key Management Service plugin settings

For more information, see AWS Key Management Service plugin.

AWS Key Management Service plugin credentials

The AWS Key Management Service plugin supports the following types of credentials.

- Creating permanent IAM user credentials
- Creating temporary credentials

Once set, run the following code in the AWS CLI to test the permissions.

```
# Test list keys permission
aws kms list-keys --region us-east-1

# Test describe key permission
aws kms describe-key --key-id "key-id-or-arn" --region us-east-1

# Test list aliases permission
aws kms list-aliases --region us-east-1

# Test get public key permission (for asymmetric keys only)
aws kms get-public-key --key-id "key-id-or-arn" --region us-east-1
```

Creating permanent IAM user credentials

The IAM user or role must have the following ACM Permissions

Identiifer	Description
kms:ListKeys	Required to discover all keys in the specified region. Without this permission, the plugin cannot enumerate keys, and the scan will fail immediately.

Identiifer	Description
kms:DescribeKey	Required to retrieve detailed metadata for each key, including key spec, key usage, key state, creation date, and origin. Without this permission, the plugin can list keys but cannot retrieve their details.
kms:ListAliases	Required to retrieve friendly alias names for keys. Without this permission, keys will be identified only by their key ID, not their alias names.
kms:GetPublicKey	Required to retrieve the public key data for asymmetric keys. Without this permission, the plugin can discover asymmetric keys but cannot extract their public key PEM data. This permission is not needed for symmetric or HMAC keys.

Follow the steps below to create a credential with these permissions.

To create permanent IAM user credentials

1. Log in to the AWS IAM Console at https://console.aws.amazon.com/iam/
2. Navigate to **Users** → **Add user**
3. Enter a username - for example, "kms-discovery-plugin".
4. Select **Programmatic access** as the access type
5. Click **Next: Permissions**
6. Attach the required permissions policy. For example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListKeys",
                "kms:DescribeKey",
                "kms:ListAliases",
                "kms:GetPublicKey"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Complete the wizard and click **Create user**
8. Save the Access Key ID and Secret Access Key displayed on the final screen.

   ⚠ You won't be able to retrieve the secret key again.

9. Use these credentials in the plugin configuration and leave `sessionToken` empty.

Creating temporary credentials

For Temporary Credentials (STS), you may need the following permissions.

Identiifer	Allowed actions
sts:GetSessionToken	Required only when using temporary credentials to generate or validate session tokens.

Follow the steps below to create a credential with these permissions.

To create temporary credentials

1. Ensure you have an IAM role configured with the required permissions.
2. Use AWS CLI or SDK to assume the role:

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/ACMReadRole \
  --role-session-name acm-plugin-session
```

3. Extract the following fields from the response:
   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`
4. Use all three values in the plugin configuration.

> ⚠ Temporary credentials typically expire within 1-12 hours and must be refreshed.

AWS Key Management Service plugin settings

This section includes:

- Required AWS Key Management Service plugin settings
- Optional AWS Key Management Service plugin settings

Required AWS Key Management Service plugin settings

Refer to the following sections for the configuration settings required by the plugin.

- accessKeyId
- secretAccessKey
- region

accessKeyId

The AWS Access Key ID for authentication.

> ⚠ This value is the first part of your AWS programmatic access credentials.

Example:

```
AKIA1A2B3C4D5E6F7G8H
```

secretAccessKey

The AWS Secret Access Key for authentication.

⚠ This is the secret part of your AWS programmatic access credentials and should be kept secure.

Example:

```
abcdef1234567890abcdef1234567890abcdef12
```

region

The AWS region to scan for certificates. Common regions include:

- us-east-1
- us-west-2
- eu-west-1
- ap-southeast-1
- etc.us-east-1

Example:

```
us-east-1
```

Optional AWS Key Management Service plugin settings

The plugin configuration supports the following optional settings.

sessionToken

An AWS session token for temporary credentials. This value is only required when using temporary credentials obtained through:

- AWS STS (Security Token Service) for role assumption
- MFA-protected operations
- Federated access

Example:

```
IQoJb3JpZ2luX2VjEBoaCXVzLWVhc3QtMSJHMEUCIQDExample+SessionToken+Data
```

AWS Secrets Manager plugin configuration

To configure the AWS Key Management Service plugin, see:

- AWS Secrets Manager plugin configuration
- AWS Secrets Manager plugin settings

For more information, see AWS Secrets Manager plugin.

AWS Secrets Manager plugin credentials

The AWS Secrets Manager plugin supports the following types of credentials.

- Creating permanent IAM user credentials

- Creating temporary credentials

Once set, run the following code in the AWS CLI to test the permissions.

```
# Test list secrets permission
aws secretsmanager list-secrets --region us-east-1

# Test describe secret permission
aws secretsmanager describe-secret --secret-id "secret-name-or-arn" --region us-
east-1

# Test with date filters for incremental scanning
aws secretsmanager list-secrets --region us-east-1 --filters Name=name,Values=test
```

Creating permanent IAM user credentials

The IAM user or role must have the following ACM Permissions

Identiifer	Description
secretsmanager :ListSecrets	Required to discover all secrets in the specified region. Without this permission, the plugin cannot enumerate secrets, and the scan will fail immediately.
secretsmanager :DescribeSecret	Required to retrieve detailed metadata for each secret, including rotation settings, version information, tags, and KMS key ID. Without this permission, the plugin can list secrets but cannot retrieve their comprehensive details.

Follow the steps below to create a credential with these permissions.

To create permanent IAM user credentials

1. Log in to the AWS IAM Console at https://console.aws.amazon.com/iam/
2. Navigate to **Users** → **Add user**
3. Enter a username - for example: "secretsmanager-discovery-plugin".
4. Select **Programmatic access** as the access type
5. Click **Next: Permissions**
6. Attach the required permissions policy. For example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecrets",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

7. Complete the wizard and click **Create user**
8. Save the Access Key ID and Secret Access Key displayed on the final screen.

   ⚠ You won't be able to retrieve the secret key again.

9. Use these credentials in the plugin configuration and leave `sessionToken` empty.

Creating temporary credentials

For Temporary Credentials (STS), you may need the following permissions.

Identiifer	Allowed actions
sts:GetSessionToken	Required only when using temporary credentials to generate or validate session tokens.

Follow the steps below to create a credential with these permissions.

To create temporary credentials

1. Ensure you have an IAM role configured with the required permissions.
2. Use AWS CLI or SDK to assume the role:

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/ACMReadRole \
  --role-session-name acm-plugin-session
```

3. Extract the following fields from the response:
   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`
4. Use all three values in the plugin configuration.

   ⚠ Temporary credentials typically expire within 1-12 hours and must be refreshed.

AWS Secrets Manager plugin settings

This section includes:

- Required AWS Secrets Manager plugin settings
- Optional AWS Secrets Manager plugin settings

Required AWS Secrets Manager plugin settings

Refer to the following sections for the configuration settings required by the plugin.

- accessKeyId
- secretAccessKey
- region

accessKeyId

The AWS Access Key ID for authentication.

⚠ This value is the first part of your AWS programmatic access credentials.

Example:

```
AKIA1A2B3C4D5E6F7G8H
```

secretAccessKey

The AWS Secret Access Key for authentication.

⚠ This is the secret part of your AWS programmatic access credentials and should be kept secure.

Example:

```
abcdef1234567890abcdef1234567890abcdef12
```

region

The AWS region to scan for certificates. Common regions include:

- us-east-1
- us-west-2
- eu-west-1
- ap-southeast-1
- etc.us-east-1

Example:

```
us-east-1
```

Optional AWS Secrets Manager plugin settings

The plugin configuration supports the following optional settings.

sessionToken

An AWS session token for temporary credentials. This value is only required when using temporary credentials obtained through:

- AWS STS (Security Token Service) for role assumption
- MFA-protected operations
- Federated access

Example:

```
IQoJb3JpZ2luX2VjEBoaCXVzLWVhc3QtMSJHMEUCIQDExample+SessionToken+Data
```

AWS CloudFront plugin configuration

See below for configuring the AWS CloudFront plugin.

- AWS CloudFront plugin credentials
- AWS CloudFront plugin settings

AWS CloudFront plugin credentials

The AWS CloudFront plugin supports these credential types.

- Creating permanent IAM user credentials
- Creating temporary credentials

After you set them up, run the following AWS CLI command to test permissions.

```
# Test list secrets permission aws secretsmanager list-secrets --region us-east-1 #
Test describe secret permission aws secretsmanager describe-secret --secret-id
"secret-name-or-arn" --region us-east-1 # Test with date filters for incremental
scanning aws secretsmanager list-secrets --region us-east-1 --filters
Name=name,Values=test
```

Creating permanent IAM user credentials

The IAM user or role must have the following ACM permissions:

secretsmanager:ListSecrets	Required to discover all secrets in the specified region. Without this permission, the plugin cannot enumerate secrets, and the scan will fail immediately.
secretsmanager:DescribeSecret	Required to retrieve detailed metadata for each secret, including rotation settings, version information, tags, and KMS key ID. Without this permission, the plugin can list secrets but cannot retrieve their comprehensive details.

Follow the steps below to create a credential with these permissions.

To create permanent IAM user credentials

1. Log in to the AWS IAM Console at https://console.aws.amazon.com/iam/
2. Navigate to **Users** → **Add user**
3. Enter a username - for example: "secretsmanager-discovery-plugin".
4. Select **Programmatic access** as the access type
5. Click **Next: Permissions**
6. Attach the required permissions policy. For example:
   `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "secretsmanager:ListSecrets", "secretsmanager:DescribeSecret" ], "Resource": "*" } ] }`
7. Complete the wizard and click **Create user**
8. Save the Access Key ID and Secret Access Key displayed on the final screen.
   You won't be able to retrieve the secret key again.
9. Use these credentials in the plugin configuration and leave `sessionToken` empty.

Creating temporary credentials

For Temporary Credentials (STS), you may need the following permissions.

sts:GetSessionToken	Required only when using temporary credentials to generate or validate session tokens.

Follow the steps below to create a credential with these permissions.

To create temporary credentials

1. Ensure you have an IAM role configured with the required permissions.
2. Use AWS CLI or SDK to assume the role:
   ```
   aws sts assume-role \ --role-arn arn:aws:iam::123456789012:role/ACMReadRole \ --role-session-name acm-plugin-session
   ```
3. Extract the following fields from the response:
   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`
4. Use all three values in the plugin configuration.
   Temporary credentials typically expire within 1-12 hours and must be refreshed.

AWS CloudFront plugin settings

See below for the AWS CloudFront plugin configuration settings.

- Required AWS CloudFront plugin settings
- Optional AWS CloudFront plugin settings

Required AWS CloudFront plugin settings

The following configuration settings are required by the plugin:

- accessKeyId
  The AWS Access Key ID for authentication. This value is the first part of your AWS programmatic access credentials.
  Example:

  ```
  AKIA1A2B3C4D5E6F7G8H
  ```

- secretAccessKey
  The AWS Secret Access Key for authentication. This is the secret part of your AWS programmatic access credentials and should be kept secure.
  Example:

  ```
  abcdef1234567890abcdef1234567890abcdef12
  ```

- Region
  The AWS region to scan for certificates.
  Example:

```
us-east-1
```

Optional AWS CloudFront plugin settings

The plugin configuration supports the following optional settings.

sessionToken

An AWS session token for temporary credentials. This value is only required when using temporary credentials obtained through:

- AWS STS (Security Token Service) for role assumption
- MFA-protected operations
- Federated access

Example:

```
IQoJb3JpZ2luX2VjEBoaCXVzLWVhc3QtMSJHMEUCIQDExample+SessionToken+Data
```

AWS Elastic Load Balancer plugin configuration

See below for configuring the AWS Elastic Load Balancer plugin.

- AWS Elastic Load Balancer plugin credentials
- AWS Elastic Load Balancer plugin settings

AWS Elastic Load Balancer plugin credentials

The AWS Elastic Load Balancer plugin supports the following types of credentials.

- Creating permanent IAM user credentials
- Creating temporary credentials

Once set, run the following code in the AWS CLI to test the permissions.

```
# Test list secrets permission aws secretsmanager list-secrets --region us-east-1 #
Test describe secret permission aws secretsmanager describe-secret --secret-id
"secret-name-or-arn" --region us-east-1 # Test with date filters for incremental
scanning aws secretsmanager list-secrets --region us-east-1 --filters
Name=name,Values=test
```

Creating permanent IAM user credentials

The IAM user or role must have the following ACM Permissions

secretsmanager:ListSecrets	Required to discover all secrets in the specified region. Without this permission, the plugin cannot enumerate secrets, and the scan will fail immediately.
secretsmanager:DescribeSecret	Required to retrieve detailed metadata for each secret, including rotation settings, version information, tags, and KMS key ID. Without this permission, the plugin can list secrets but cannot retrieve their comprehensive details.

Follow the steps below to create a credential with these permissions.

To create permanent IAM user credentials

1. Log in to the AWS IAM Console at https://console.aws.amazon.com/iam/
2. Navigate to **Users** → **Add user**
3. Enter a username - for example: "secretsmanager-discovery-plugin".
4. Select **Programmatic access** as the access type
5. Click **Next: Permissions**
6. Attach the required permissions policy. For example:
   ```
   { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action":
   [ "secretsmanager:ListSecrets", "secretsmanager:DescribeSecret" ], "Resource":
   "*" } ] }
   ```
7. Complete the wizard and click **Create user**
8. Save the Access Key ID and Secret Access Key displayed on the final screen.
   You won't be able to retrieve the secret key again.
9. Use these credentials in the plugin configuration and leave `sessionToken` empty.

Creating temporary credentials

For Temporary Credentials (STS), you may need the following permissions.

sts:GetSessionToken	Required only when using temporary credentials to generate or validate session tokens.

Follow the steps below to create a credential with these permissions.

To create temporary credentials

1. Ensure you have an IAM role configured with the required permissions.
2. Use AWS CLI or SDK to assume the role:
   ```
   aws sts assume-role \ --role-arn arn:aws:iam::123456789012:role/ACMReadRole \ --
   role-session-name acm-plugin-session
   ```
3. Extract the following fields from the response:
   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`
4. Use all three values in the plugin configuration.
   Temporary credentials typically expire within 1-12 hours and must be refreshed.

AWS Elastic Load Balancer plugin settings

See below for the AWS Elastic Load Balancer plugin configuration settings.

Required AWS Elastic Load Balancer plugin settings

The following configuration settings are required by the plugin:

- accessKeyId
  The AWS Access Key ID for authentication. This value is the first part of your AWS programmatic access credentials.
  Example:

```
AKIA1A2B3C4D5E6F7G8H
```

- secretAccessKey
  The AWS Secret Access Key for authentication. This is the secret part of your AWS programmatic access credentials and should be kept secure.
  Example:

```
abcdef1234567890abcdef1234567890abcdef12
```

- Region
  The AWS region to scan for certificates.
  Example:

```
us-east-1
```

Optional AWS Elastic Load Balancer plugin settings

The plugin configuration supports the following optional settings.

sessionToken

An AWS session token for temporary credentials. This value is only required when using temporary credentials obtained through:

- AWS STS (Security Token Service) for role assumption
- MFA-protected operations
- Federated access

Example:

```
IQoJb3JpZ2luX2VjEBoaCXVzLWVhc3QtMSJHMEUCIQDExample+SessionToken+Data
```

Azure Key Vault plugin configuration

To configure the Azure Key Vault plugin, see:

- Azure Key Vault plugin credentials
- Azure Key Vault plugin settings

For more information, see Azure Key Vault plugin.

Azure Key Vault plugin credentials

Create Azure Key Vault credentials with the following permissions.

Scope	Permission	Description
Certificate	get	Required to retrieve certificate details and PEM-encoded certificate bodies. Without this permission, the plugin can list certificates but cannot retrieve their data.
	list	Required to enumerate all certificates in the vault. Without this permission, the plugin cannot discover certificates.
Key	get	Required to retrieve key details and public key data for asymmetric keys. Without this permission, the plugin can list keys but cannot retrieve public keys or metadata.
	list	Required to enumerate all keys in the vault. Without this permission, the plugin cannot discover keys.
Secret	get	Required to retrieve secret metadata. The plugin does not retrieve secret values; however, this permission is needed for metadata access. Without this permission, the plugin can list secrets but cannot retrieve metadata.
	list	Required to enumerate all secrets in the vault. Without this permission, the plugin cannot discover secrets.

The following credential creation modes are supported:

- Creating Azure Key Vault credentials with the Azure CLI
- Creating Azure Key Vault credentials with the web console

Creating Azure Key Vault credentials with the Azure CLI

See below for creating Azure Key Vault credentials with the Azure CLI.

To create Azure Key Vault credentials with the Azure CLI

1. Create a service principal.

```
az ad sp create-for-rbac --name "KeyVault-Discovery-Plugin"
```

2. Copy the following from the command output.
   - The `clientId` application identifier.

- The `clientSecret` password.
- The `tenantId` tenant identifier.

3. Grant Key Vault permissions.

```
az keyvault set-policy --name your-keyvault-name \
  --spn <clientId> \
  --certificate-permissions get list \
  --key-permissions get list \
  --secret-permissions get list
```

Creating Azure Key Vault credentials with the web console

Follow the steps below to create Azure Key Vault credentials with the Azure web console.

- Registering an Azure AD Application
- Creating a client secret
- Granting permissions to Key Vault

Registering an Azure AD Application

Register an Azure AD (Active Directory) application as explained below.

To register an Azure AD application

1. Log in to the Azure Portal at https://portal.azure.com
2. Navigate to **Azure Active Directory → App registrations**
3. Click **New registration**
4. Enter a name – for example: "KeyVault-Discovery-Plugin".
5. Select **Accounts in this organizational directory only** for supported account types.
6. Click **Register**
7. On the application overview page, note the **Application (client) ID** and **Directory (tenant) ID.**

Creating a client secret

Create a client secret as explained below.

To create a client secret

1. In your registered application, navigate to **Certificates & secrets**
2. Click **New client secret**
3. Enter a description – for example: "Discovery Plugin Secret".
4. Select an expiration period – for example: 6 months, 12 months, 24 months, or custom.
5. Click **Add.**
6. Immediately copy the secret **Value.**

> ⚠ You won't be able to see this value again.

7. Use this value as `clientSecret` in the plugin configuration

Granting permissions to Key Vault

Grant permissions to your Key Vault in Azure.

To grant permissions to Key Value

1. Navigate to your Key Vault in the Azure Portal

2. Click **Access policies** in the left menu
3. Click **Create** (or **+ Add Access Policy** on older portal versions)
4. Select the following permissions:
    - **Certificate permissions**: Get, List
    - **Key permissions**: Get, List
    - **Secret permissions**: Get, List
5. Click **Next** and search for your application name
6. Select your application and click **Next**
7. Review and click **Create** to save the access policy

Azure Key Vault plugin settings

See the following sections for the configuration settings required by the Key Vault plugin.

- vaultUrl
- clientId
- clientSecret
- tenantId

> This plugin does not have optional values.

vaultUrl

The full URL of your Azure Key Vault endpoint.

Example:

```
https://your-keyvault-name.vault.azure.net
```

clientId

The Azure AD Application (client) identifier.

> This value is the unique identifier for the Azure AD application (service principal) that will authenticate to Key Vault.

Example:

```
12345678-1234-1234-1234-123456789abc
```

clientSecret

The Azure AD Application client secret.

> This value is the secret for the service principal and should be kept secure.

Example:

```
your-client-secret-value~abc123
```

tenantId

The Azure AD Directory (tenant) identifier.

> ⚠ This value is the unique identifier for your Azure AD tenant/directory.

Example:

```
87654321-4321-4321-4321-cba987654321
```

GCP Certificate Manager plugin configuration

To configure the GCP Certificate Manager plugin, see:

- GCP Certificate Manager plugin credentials
- GCP Certificate Manager plugin settings

GCP Certificate Manager plugin credentials

GCP Certificate Manager plugin requires

- **GCP Project**: Active Google Cloud project with Certificate Manager API enabled
- **Service Account**: Service account with appropriate permissions
- **IAM Permissions**: The service account needs the following permissions:
  - `certificatemanager.certs.list`
  - `certificatemanager.certs.get`

Service Account Setup

1. **Create a Service Account**:
   ```
   gcloud iam service-accounts create cert-discovery \
     --description="Service account for Certificate Manager discovery" \
     --display-name="Certificate Discovery"
   ```
2. **Grant Certificate Manager Permissions**:
   ```
   gcloud projects add-iam-policy-binding YOUR_PROJECT_ID \
     --member="serviceAccount:cert-discovery@YOUR_PROJECT_ID.iam.gserviceaccount.com"
    \
     --role="roles/certificatemanager.viewer"
   ```
3. **Generate Service Account Key**:
   ```
   gcloud iam service-accounts keys create cert-discovery-key.json \
     --iam-account=cert-discovery@YOUR_PROJECT_ID.iam.gserviceaccount.com
   ```

GCP Certificate Manager plugin settings

The following configuration settings are required:

`projectId`	GCP Project ID to scan	`my-gcp-project`

`privateKeyId`	Private key identifier from service account	`8d911f4e6fd27332a575283a182bafc13460093f`
`privateKey`	Service account private key in PEM format	`-----BEGIN PRIVATE KEY-----...`
`clientEmail`	Service account email address	`cert-discovery@project.iam.gserviceaccount.com`
`clientId`	OAuth2 client ID	`111813473337277308137`
`clientX509CertUrl`	OAuth2 client certificate URL	`https://www.googleapis.com/robot/v1/metadata/x509/...`
`location`	Specific GCP region (scans all locations if empty)	`us-central1`, `global`

GCP Key Management Service plugin configuration

To configure the GCP Key Management Service plugin, see:

- GCP Key Management Service plugin credentials
- GCP Key Management Service plugin settings

GCP Key Management Service plugin credentials

GCP Key Management Service plugin requires:

- **GCP Project**: Active Google Cloud project with Cloud KMS API enabled
- **Service Account**: Service account with appropriate KMS permissions
- **IAM Permissions**: The service account needs the following permissions:
  - `cloudkms.keyRings.list`
  - `cloudkms.cryptoKeys.list`
  - `cloudkms.cryptoKeyVersions.list`
  - `cloudkms.cryptoKeyVersions.get`
  - `cloudkms.cryptoKeyVersions.getPublicKey`

Service Account Setup

1. **Create a Service Account**:
```
gcloud iam service-accounts create kms-discovery \
  --description="Service account for KMS key discovery" \
  --display-name="KMS Discovery"
```
2. **Grant KMS Permissions**:
```
gcloud projects add-iam-policy-binding YOUR_PROJECT_ID \
  --member="serviceAccount:kms-discovery@YOUR_PROJECT_ID.iam.gserviceaccount.com"
```

```
  \
   --role="roles/cloudkms.viewer"
```

3. **Generate Service Account Key**:

```
gcloud iam service-accounts keys create kms-discovery-key.json \
  --iam-account=kms-discovery@YOUR_PROJECT_ID.iam.gserviceaccount.com
```

GCP Key Management Service plugin settings

This section includes:

- Required GCP Key Management Service plugin settings
- Optional GCP Key Management Service plugin settings

Required GCP Key Management Service plugin settings

GCP Key Management Servicer plugin requires following configuration settings:

Parameter	Description	Example
`projectId`	GCP Project ID to scan	`my-gcp-project`
`location`	GCP location to scan	`global` , `us-central1`
`privateKey`	Service account private key in PEM format	`-----BEGIN PRIVATE KEY-----...`
`clientEmail`	Service account email address	`kms-discovery@project.iam.gserviceaccount.com`

Optional GCP Key Management Service plugin settings

GCP Key Management Service plugin configuration supports the following optional settings:

Parameter	Description	Example
`privateKeyId`	Private key identifier (metadata only)	`8d911f4e6fd27332a575283a182bafc13460093f`
`clientId`	OAuth2 client ID (metadata only)	`111813473337277308137`
`clientX509CertUrl`	OAuth2 client certificate URL (metadata only)	`https://www.googleapis.com/robot/v1/metadata/x509/...`

GCP Secret Manager plugin configuration

To configure the GCP Secret Manager plugin, see:

- GCP Secret Manager plugin credentials
- GCP Secret Manager plugin settings

GCP Secret Manager plugin credentials

GCP Secret Manager plugin requires:

1. **GCP Project**: A valid GCP project with Secret Manager API enabled
2. **Service Account**: A service account with appropriate permissions
3. **Authentication**: Service account key file or credentials

Required Permissions

The service account must have the following IAM permissions:

- `secretmanager.secrets.list`
- `secretmanager.versions.list`
- `secretmanager.versions.access`

Recommended IAM role: `Secret Manager Secret Accessor` or a custom role with the above permissions.

GCP Secret Manager plugin settings

The following configuration settings are required:

`projectId`	GCP Project ID to scan	`my-gcp-project`
`privateKeyId`	Private key identifier from service account	`5094569f8086a9ae...`
`privateKey`	Private key in PEM format	`-----BEGIN PRIVATE KEY-----...`
`clientEmail`	Service account email address	`discovery@my-project.iam.gserviceaccount.com`
`clientId`	OAuth2 client ID	`104319692654298486174`
`clientX509CertUrl`	OAuth2 client certificate URL	`https://www.googleapis.com/robot/v1/metadata/x509/...`

HashiCorp Vault plugin configuration

To configure the HashiCorp Vault plugin, see:

- HashiCorp Vault plugin credentials
- HashiCorp Vault plugin settings

HashiCorp Vault plugin credentials

The HashiCorp Vault plugin supports the following types of credentials:

- Vault Token (recommended)
- AppRole (for automation/service accounts)
- Other Auth Methods (Kubernetes, LDAP, AWS IAM, OIDC, GitHub, etc.)
    - Use the resulting Vault token in the plugin configuration

---

Token Generation

Login to Vault

```
vault login
```

Use your organization's auth method (userpass/LDAP/OIDC/GitHub/etc.).

Create a Token with TTL

```
vault token create -policy="root" -ttl=24h
```

- Replace `root` with a policy that grants read/list permissions to the engines you intend to scan.
- Adjust `-ttl` (e.g., `1h`, `24h`, `72h`). Tokens expire after TTL and must be renewed or recreated.

Non-Expiring Tokens (Use with caution)

```
vault token create -policy="root"
```

- Omitting `-ttl` creates a token with default TTL; in many setups this can be long-lived or effectively non-expiring depending on token role constraints.
- Prefer short-lived, renewable tokens for security. If you must use a non-expiring token, scope it narrowly via policy and rotate periodically.

AppRole (Programmatic)

```
vault write auth/approle/login role_id="<ROLE_ID>" secret_id="<SECRET_ID>"
```

- Extract the `client_token` from the response and use it as `vaultToken` in the plugin.

Required Policies / Permissions

Grant list/read on relevant paths. Example for KV v2 mounted at `secret/`:

```
# KV v2 data read/list
path "secret/data/*" {
  capabilities = ["read", "list"]
}
```

```
# KV v2 metadata list (needed to enumerate)
path "secret/metadata/*" {
  capabilities = ["list"]
}
```

For other engines (Transit, PKI, etc.), grant the minimal read/list capabilities needed to enumerate and read metadata.

Security Notes

- Prefer short-lived, renewable tokens. Rotate credentials regularly.
- Scope policies to the minimal paths required for discovery.
- Avoid broadly-permissive tokens; use narrow mounts and namespaces.

HashiCorp Vault plugin settings

This section includes:

- Required HashiCorp Vault plugin settings
- Optional HashiCorp Vault plugin settings

Required HashiCorp Vault plugin settings

HashiCorp Vault plugin requires following configuration settings:

Parameter	Description	Example
vaultURL	Vault address (HTTP/HTTPS)	https://10.1.127.31:8200
token	Token generated via login/AppRole/ other auth	hvs.EkUYG34L76vGpt8jfCDqCpdS
engineTypes	Engines to include in discovery search	["pki", "transit", "kv"]
timeoutSeconds	Request timeout in seconds	30

Example Plugin Configuration

```
{
  "vaultUrl": "https://your-vault-server.example.com:8200",
  "vaultToken": "<your-vault-token>",
  "engineTypes": ["kv", "transit", "pki"],
  "namespace": "<optional-enterprise-namespace>",
  "skipTlsVerify": optional-true/false,
  "timeoutSeconds": <timeout-in-seconds>
```

```
}
```

Troubleshooting

- Connection issues: verify `vaultUrl`, network reachability, TLS settings.
- Permission errors: review policies and engine mount paths (KV v2 uses `data/` and `metadata/`).
- Docker on Linux: use `--network=host` or the bridge IP (`172.17.0.1`).

Optional HashiCorp Vault plugin settings

HashiCorp plugin configuration support following optional settings.

Parameter	Description	Example
namespace	For Vault Enterprise (optional).	"root"
skipTlsVerify	Skip TLS certificate verification (optional, not recommended for production)	true

Network Scanner - TLS plugin configuration

To configure the TLS plugin, see:

- Network Scanner - TLS plugin credentials
- Network Scanner - TLS plugin settings

For more information, see Network Scanner - TLS plugin.

Network Scanner - TLS plugin credentials

The Network Scanner - TLS plugin does not require cloud service credentials.

Network Scanner - TLS plugin settings

This section includes:

- Required Network Scanner - TLS plugin settings
- Optional Network Scanner - TLS plugin settings

Required Network Scanner - TLS plugin settings

See the following sections for the configuration settings required by the Network Scanner - TLS plugin.

- host
- ports

host

Comma-delimited list of targets to scan. Supports IPv4 addresses, IPv6 addresses (with brackets), hostnames, and CIDR blocks. Multiple targets can be combined.

For example:

```
192.168.1.1,10.0.0.0/24,example.com,2001:db8::1
```

ports

Comma-delimited list of TCP ports to scan for SSL/TLS services. Supports individual ports and ranges. Standard SSL/TLS ports include:

- 22 (SSH)
- 25/465/587 (SMTP)
- 443 (HTTPS)
- 993/995 (IMAP/POP3)
- 636 (LDAPS)
- 990 (FTPS)

For example:

```
443,22,25,465,587,993,995,8443,8000-8999
```

Optional Network Scanner - TLS plugin settings

See the following sections for the optional Network Scanner - TLS plugin configuration settings.

- excludeHosts
- timeout
- timingTemplate
- maxHostGroup
- skipHostDiscovery
- skipDNSResolution:
- randomizeHosts
- state.lastRunDateTime

excludeHosts

Comma-delimited list of hosts to exclude from scanning. Uses the same format as the hosts field. Useful for excluding routers, firewalls, or critical infrastructure from scans.

For example:

```
10.1.127.1,192.168.1.254
```

timeout

Total scan timeout in seconds. For large networks, increase this value to prevent premature scan termination.

- Default: 300 seconds (5 minutes).
- Range: 1-86400 seconds (1 second to 24 hours).
- Example: 60

timingTemplate

The name of the TLS timing template controlling scan speed and network impact. See the following table for the supported options.

Template name	Template identifier	Scan speed
paranoid	T0	Slowest, most stealthy, suitable for IDS evasion
sneaky	T1	Very slow and stealthy
polite	T2	Slow and considerate to target systems
normal	T3	Balanced speed and stealth (default)
aggressive	T4	Fast but more detectable
insane	T5	Fastest but very noisy, and may miss results

Example: `polite`

maxHostGroup

The maximum number of hosts to scan in parallel.

- For small networks, use 5-10;
- For large networks, use 50-100.

Larger values increase scan speed but delay streaming feedback and consume more resources.

- Default: 10 hosts.
- Range: 0-1000.
- Example: 20

skipHostDiscovery

`true` to treat all hosts as online and skip ping probes using:

```
Nmap -Pn flag
```

Use this value when firewalls block ICMP ping requests. May increase scan time as TLS will attempt to scan all specified hosts.

- Default: `false`
- Example: `true`

skipDNSResolution:

`true` to skip reverse DNS lookups using:

```
Nmap -n flag
```

Enabling this speeds up scans and reduces network traffic; however, hostnames won't be resolved in the output.

- Default: `false`
- Example: `true`

randomizeHosts

`true` to randomize the order of host scanning using:

```
Nmap --randomize-hosts flag
```

This value helps distribute scan load and avoid detection patterns.

- Default: `false`
- Example: `true`

state.lastRunDateTime

The timestamp of the last scan execution. This value is updated automatically after each scan and is for informational and tracking purposes only.

Example:

```
2025-10-02T12:00:00Z
```

Network Scanner - OIDC plugin configuration

To configure the Network Scanner - OIDC plugin, see:

- Network Scanner - OIDC plugin credentials
- Network Scanner - OIDC plugin settings

Network Scanner - OIDC plugin credentials

The Network Scanner - OIDC plugin does not require cloud service credentials.

Network Scanner - OIDC plugin settings

This section includes:

- Required Network Scanner - OIDC settings
- Optional Network Scanner - OIDC settings

Required Network Scanner - OIDC settings

The plugin configuration supports the following required settings:

issuerURL

List of OIDC provider issuer URLs to scan

Example:

"https://accounts.google.com",

"https://login.microsoftonline.com/common"

The plugin validates that at least one issuer URL is provided and that each URL is properly formatted. URLs must use HTTPS protocol (except for localhost/127.0.0.1 for testing).

Optional Network Scanner - OIDC settings

The plugin configuration supports the following optional settings:

timeout

HTTP request timeout in milliseconds

- **default: 5000**
- **minimum: 100**

Network Scanner - SSH Scanner plugin configuration

To configure the SSH Scanner plugin, see:

- Network Scanner - SSH plugin credentials
- Network Scanner - SSH plugin settings

For more information, see Network Scanner - SSH plugin.

Network Scanner - SSH plugin credentials

The Network Scanner - SSH plugin does not require SSH authentication. Specifically, it does not require SSH usernames, passwords, or private keys, as it operates by:

1. Establishing TCP connections to SSH ports
2. Performing the initial SSH protocol handshake
3. Retrieving publicly advertised server information, such as host keys, certificates, or algorithms
4. Disconnecting before authentication

Therefore, the plugin:

- Does not require SSH credentials in configuration
- Does not make authentication attempts
- Only collects visible SSH server information
- Cannot access authenticated SSH resources
- Is safe to run against production SSH servers

Network Scanner - SSH plugin settings

This section includes:

- Required Network Scanner - SSH plugin settings
- Optional Network Scanner - SSH plugin settings

Required Network Scanner - SSH plugin settings

See the following sections for the configuration settings required by the Network Scanner - SSH plugin.

- hosts
- ports

hosts

A comma-delimited list of hosts to scan. Supports multiple formats:

Format	Example
IP addresses	192.168.1.10
Hostnames	server.example.com
CIDR blocks (scans all hosts in subnet)	10.1.127.0/24
IP ranges (scans last octet range)	192.168.1.1-50

This value can combine all formats in a single configuration. For example:

```
192.168.1.10,server.example.com,10.1.127.0/24,192.168.1.1-50
```

ports

A comma-delimited list of TCP port numbers to scan for SSH services.

> Port 22 is the standard SSH port, but many systems use non-standard ports for security.

This value supports a list of up to 100 ports. For example:

```
22,2222,2200,8022
```

Optional Network Scanner - SSH plugin settings

The Network Scanner - SSH plugin configuration supports the following optional setting.

timeout

The SSH connection timeout in milliseconds.

- Default: 1000ms (1 second).
- Range: 100ms to 3600000ms (1 hour).
- Example: `2000`

Increase for slow or distant networks; decrease for fast local networks to speed up scans.

Tenable Vulnerability Management plugin configuration

To configure the Tenable Vulnerability Management plugin, see:

- Tenable Vulnerability Management plugin credentials
- Tenable Vulnerability Management plugin settings

For more information, see Tenable Vulnerability Management plugin.

Tenable Vulnerability Management plugin credentials

To use the Tenable Vulnerability Management plugin, you must generate API access and secret keys from your Tenable account.

To generate secret keys

1. Navigate to your Tenable.io instance, typically:

```
https://cloud.tenable.com
```

2. Log in with your Tenable credentials
3. Click on your username in the top-right corner
4. Select **My Account** from the dropdown menu
5. Navigate to the **API Keys** tab
6. Click **Generate** to create new API keys.
7. Immediately copy both the **Access Key** and **Secret Key** values.
8. Store these keys securely.

> ⚠ The secret keys will not be displayed again.

9. Optionally, name the keys – for example: "Certificate Discovery Plugin".
10. Grant the following permissions and roles to your account.
    - Required Tenable.io permissions
    - Role-Based access
    - Custom Roles

Required Tenable.io permissions

Grant the following permissions to the credential.

Permissions	Description
View Scan Results	Required to access vulnerability scan data containing certificate information. Without this permission, the plugin cannot retrieve vulnerability details and will fail to function.
View Assets	Required to export asset inventory and validate which assets are current. Without this permission, the plugin cannot perform asset validation or incremental scanning.
Use Exports API	Required to export large datasets of vulnerabilities and assets. The plugin uses the Export APIs (/vulns/export and /assets/v2/export), which require this permission. Without this, the plugin cannot bulk-export vulnerability and asset data.

Role-Based access

Tenable uses role-based access control. The following built-in roles have sufficient permissions.

Role	Description
Administrator	Full access (includes all required permissions)
Scan Manager	Can view scan results and assets
Scan Operator	Can view scan results and assets
Standard (with appropriate permissions)	May work when granting the View Scan Results and View Assets permissions.

Custom Roles

If using custom roles, ensure the role includes:

- Can View Scan Results
- Can View Assets
- Can Use Exports API
- Can Access API (basic API access)

Tenable Vulnerability Management plugin settings

This section includes:

- Required Tenable Vulnerability Management plugin settings
- Optional Tenable Vulnerability Management plugin settings

Required Tenable Vulnerability Management plugin settings

See the following sections for the configuration settings required by the Tenable Vulnerability Management plugin.

- accessKey
- secretKey

accessKey

The Tenable API access key. This 64-character hexadecimal string serves as the username for API authentication.

Example:

```
1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

secretKey

The Tenable API secret key. This 64-character hexadecimal string serves as the password for API authentication and should be kept secure.

```
abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890
```

Optional Tenable Vulnerability Management plugin settings

The Tenable Vulnerability Management plugin supports the following optional configuration setting.

baseUrl

Tenable base URL. See below for the supported values.

US data center (default)
`https://cloud.tenable.com`

EU data center
`https://cloud.tenable.eu`

Canada data center
`https://cloud.tenable.ca`

Asia-Pacific data center
`https://cloud.apac.tenable.com`

Custom URLs for on-premise Tenable.sc installations support valid URLs with a length of at least three characters.

Qualys CertView plugin configuration

See below for configuring the Qualys CertView plugin.

- Qualys CertView plugin credentials
- Qualys CertView plugin settings

Qualys CertView plugin credentials

Qualys CertView plugin credentials are as below:

gatewayUrl": "https://gateway.qg3.apps.qualys.com",
"username": "entru3dp1",
"password": "apg_gpc3PJK*udh3uht"

Qualys CertView plugin settings

This section includes:

- Required Qualys CertView plugin settings

Required Qualys CertView plugin settings

See the following sections for the configuration settings required by the Qualys CertView plugin.

Required Fields

- `username` (string): Qualys user login with CertView permissions
- `password` (string): Qualys user password
- `gatewayUrl` (string): Qualys Gateway base URL (e.g., `https://gateway.qg3.apps.qualys.com`)

Finding Your Gateway URL

Find your Qualys Gateway URL in your Qualys portal under Help → About. This URL is specific to your subscription and platform.

Permissions

Required permissions:

- CertView Module Access (read)
- API Access enabled
- Certificate View Permission

Recommended: Use a dedicated service account with read-only CertView access.

Optional Qualys CertView plugin settings

The plugin configuration supports the following optional settings:

Usage

CLI Commands

info

Returns plugin metadata:

```
./qualys-certview-plugin info
```

schema

Returns JSON schema for configuration validation:

```
./qualys-certview-plugin schema
```

uischema

Returns UI schema for form rendering:

```
./qualys-certview-plugin uischema
```

test

Verifies connectivity and validates credentials:

```
echo '{"username":"...","password":"...","gatewayUrl":"https://gateway.qg3.apps.qualys.com"}' | ./qualys-certview-plugin test
```

scan

Discovers and returns certificates from all asset types:

`echo '{"username":"...","password":"...","gatewayUrl":"https://gateway.qg3.apps.qualys.com"}' | ./qualys-certview-plugin scan`

Returns JSON array of certificate scan results with URNs, PEM data, and Qualys-specific metadata. The `asset_type` field in results indicates whether each certificate is from "managed" or "unmanaged" assets.

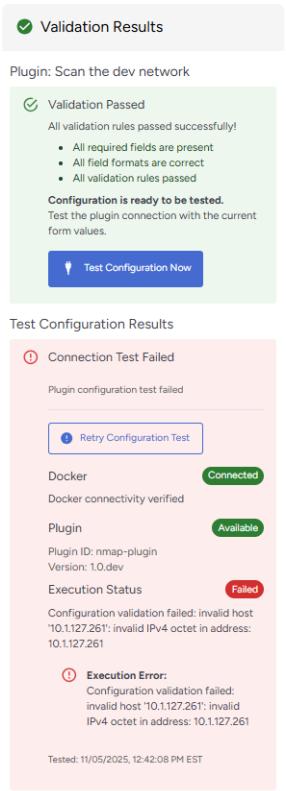
Schedule

In this tab, define the execution timeline for your scanning operations. Recurring schedules include:

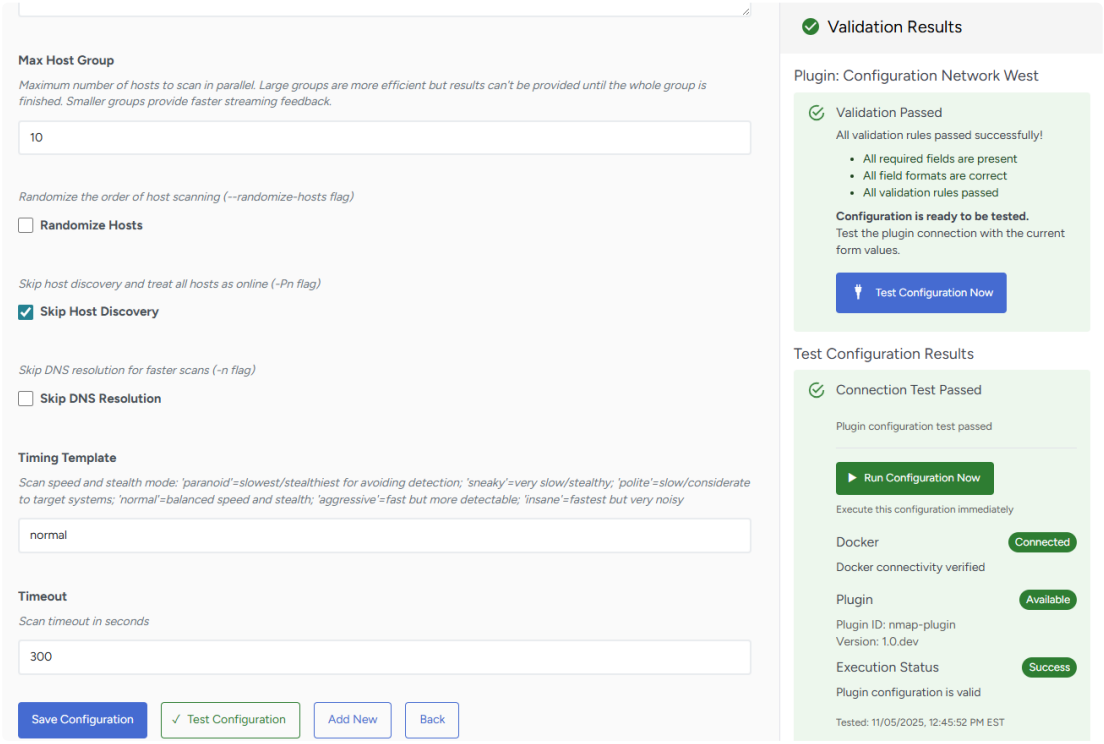
- Daily, weekly, monthly, and yearly intervals,
- Setting a CRON expression.

Validation results

The system continuously validates your configuration inputs, providing immediate feedback on the correctness and completeness of parameters. Once validation passes, you must verify connectivity by testing the actual connection to your target systems.

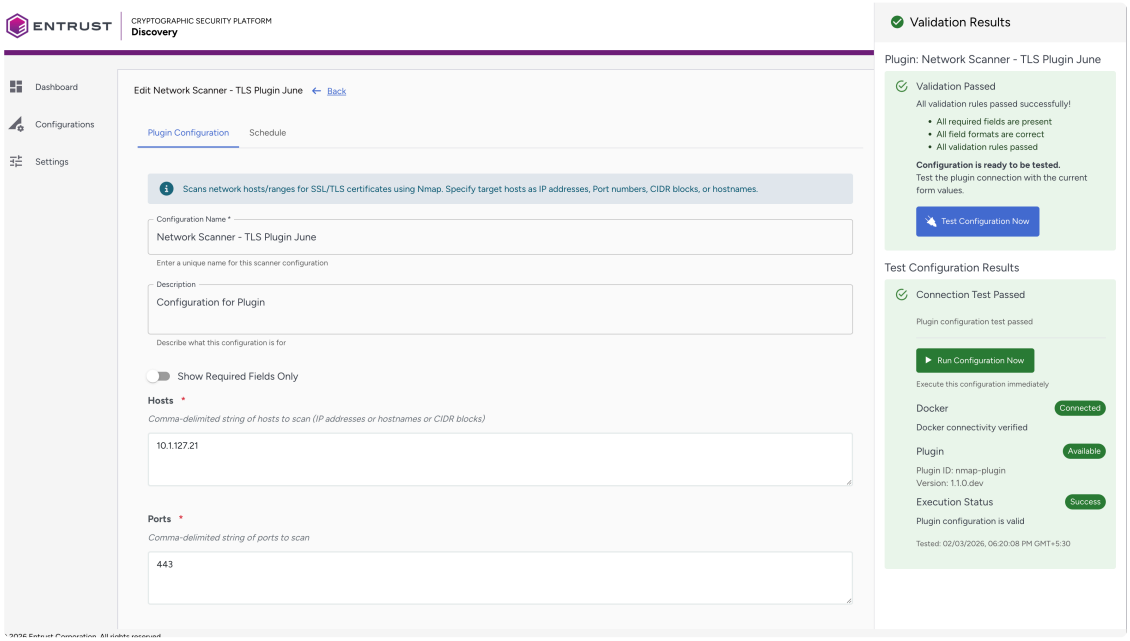


This workflow ensures that only validated, tested configurations are deployed in your environment, reducing the likelihood of failed scans due to configuration errors or connectivity issues.



Following the successful connection testing, you must save your configuration to make it operational. At this point, you have three options:

- **Run Configuration Now:** Execute immediately to begin scanning with the current configuration
- **Add New:** Create additional configurations for the same plugin type to target different network segments or use different parameters.
- **Back:** Return to the main configuration page to manage your complete configuration portfolio.



The Test Configuration Results section shows three areas as below:

- Docker: Connected (Docker connectivity verified)
- Plugin: Available (Plugin ID: nmap-plugin, Version: 1.1.0.dev)
- Execution status: Success (Plugin configuration is valid)

Discovery Plugins Output

This section provides details about the plugin output, including cryptographic asset's raw data and additional properties being sent to CSP Compliance Manager.

- AWS Certificate Manager plugin output
- AWS Key Management Service plugin output
- AWS Secrets Manager plugin output
- AWS CloudFront plugin output
- AWS Elastic Load Balancer Plugin output
- AWS CloudHSM plugin output
- Azure Key Vault plugin output
- GCP Certificate Manager plugin output
- GCP Key Management Service plugin output
- GCP Secret Manager plugin output
- HashiCorp Vault plugin output
- Network Scanner - TLS plugin output
- Network Scanner - OIDC plugin output
- Network Scanner - SSH plugin output
- Tenable Vulnerability Management plugin output
- Filesystem Scanner plugin output
- Qualys CertView plugin output
- Qualys TotalCloud plugin output
- Qualys Container Security plugin output

AWS Certificate Manager plugin output

Output Format

Certificate Scan Results

The plugin generates certificate scan results in the following standardized format:


```
{
    "type": "cert",
    "timestamp": "2025-12-15T16:18:46-05:00",
    "urn":
"urn:cert:sha256:9b33a57a6e9e9416a24e5e7748912b96e87c1990aab1e7ea60eeb303293c1a0a",
    "url": "https://console.aws.amazon.com/acm/home?region=us-east-1#/?id=arn:aws:acm:us-
east-1:558162184545:certificate/1275ea29-2cbb-4ff4-98f2-e1cc6196e9d8",
    "extra": {
      "account_id": "558162184545",
      "arn": "arn:aws:acm:us-east-1:558162184545:certificate/1275ea29-2cbb-4ff4-98f2-
e1cc6196e9d8",
      "options": {
        "cert_transparency_logging_preference": "disabled"
      },
      "origin": "imported",
      "platform_type": "aws",
      "region": "us-east-1",
      "renewal_eligibility": "ineligible",
      "status": "issued"
    },
    "cert_pem": "MIIDizCC......"

}
```


When Certificate Body Is Unavailable


```
{
    "type": "cert",
    "timestamp": "2025-12-15T16:18:45-05:00",
    "urn": "urn:cert:arn:arn:aws:acm:us-east-1:558162184545:certificate/
b9464c90-29f8-4d88-bb18-71dcaa3e031f",
    "url": "https://console.aws.amazon.com/acm/home?region=us-east-1#/?id=arn:aws:acm:us-
east-1:558162184545:certificate/b9464c90-29f8-4d88-bb18-71dcaa3e031f",
    "extra": {
      "account_id": "558162184545",
      "arn": "arn:aws:acm:us-east-1:558162184545:certificate/b9464c90-29f8-4d88-
bb18-71dcaa3e031f",
```

```
      "domain_name": "*.example.com",
      "issuer": "Amazon",
      "options": {
        "cert_transparency_logging_preference": "enabled"
      },
      "origin": "amazon_issued",
      "platform_type": "aws",
      "public_key_algorithm": "RSA-2048",
      "region": "us-east-1",
      "renewal_eligibility": "ineligible",
      "signature_algorithm": "SHA256WITHRSA",
      "status": "failed",
      "subject": "CN=*.example.com",
      "subject_alternative_names": [
        "*.example.com"
      ]
    },
    "cert_pem": ""

}
```


URN Generation: The `urn` field is generated using a hash of the certificate PEM body for uniqueness and consistency.

Error Output: If an error occurs, the plugin outputs a structured JSON object with error details under `data.errorDetails` and a summary message in `data.errorMessage`.

Field Descriptions

- `type` : Always "cert" for certificate assets
- `timestamp` : RFC3339 timestamp of when the scan was performed, in local timezone (e.g., `2024-08-07T10:30:00-07:00` for PDT or `2024-08-07T17:30:00Z` for UTC)
- `urn` : Unique identifier generated from certificate SHA-256 hash
- `url` : Direct link to certificate in AWS Console
- `certPem` : RFC 7468 compliant PEM-encoded certificate (base64 body without headers/footers)
- `extra` : Additional metadata from ACM (see Extra Fields section below)

Extra Fields

The `extra` object contains AWS-specific metadata about the certificate. The following fields are included:

Standard Fields (Always Included)

- `platform_type` (string): Always `"aws"` indicating the certificate is from AWS Certificate Manager
- `renewal_eligibility` (string): Whether the certificate is eligible for renewal (values: `"eligible"` , `"ineligible"`)
- `options` (object): Certificate configuration options
    - `cert_transparency_logging_preference` (string): Certificate transparency logging preference (values: `"enabled"` , `"disabled"`)
- `account_id` (string): AWS account ID extracted from the certificate ARN. Empty string if extraction fails
- `region` (string): AWS region where the certificate is stored (e.g., `"us-east-1"`)

- `arn` (string): Full AWS ARN of the certificate (e.g., `"arn:aws:acm:us-east-1:123456789012:certificate/abcd-1234"`)
- `status` (string): Current certificate status (values: `"issued"`, `"pending_validation"`, `"failed"`, `"expired"`, `"revoked"`, `"inactive"`)
- `origin` (string): Certificate type/origin (values: `"amazon_issued"`, `"imported"`)

Fallback Fields (When Certificate Body Is Unavailable)

When the certificate body cannot be retrieved via `GetCertificate`, the following additional fields are populated from `DescribeCertificate` metadata:

- `issuer` (string): Certificate issuer distinguished name
- `subject` (string): Certificate subject distinguished name
- `public_key_algorithm` (string): Public key algorithm type (e.g., "RSA_2048", "EC_prime256v1")
- `signature_algorithm` (string): Signature algorithm used
- `subject_alternative_names` (array of strings): List of Subject Alternative Names (SANs)
- `serial_number` (string): Certificate serial number
- `not_after` (string): Certificate expiration date (RFC3339 format)
- `not_before` (string): Certificate validity start date (RFC3339 format)

Note: All status, origin, and preference values are converted to lowercase for consistency.

AWS Key Management Service plugin output

Output Format

Symmetric Key Scan Results


```
{
    "type": "symkey",
    "timestamp": "2025-12-15T11:39:49-05:00",
    "urn": "urn:symkey:name:alias/aws/secretsmanager:arn:aws:kms:us-
east-1:558162184545:key/4b3daa9e-1474-4629-8f8a-95f2003caf0f",
    "url": "https://console.aws.amazon.com/kms/home?region=us-east-1#/kms/keys/
4b3daa9e-1474-4629-8f8a-95f2003caf0f",
    "extra": {
      "account_id": "558162184545",
      "alias": "alias/aws/secretsmanager",
      "arn": "arn:aws:kms:us-east-1:558162184545:key/
4b3daa9e-1474-4629-8f8a-95f2003caf0f",
      "cipher_suites": [
        "SYMMETRIC_DEFAULT"
      ],
      "created_date": "2025-10-22T18:29:42Z",
      "cryptographic_algorithm": "AES",
      "cryptographic_length": 256,
      "description": "Default key that protects my Secrets Manager data when no other key
is defined",
```

```
      "hsm_backed": true,
      "key_id": "4b3daa9e-1474-4629-8f8a-95f2003caf0f",
      "key_manager": "aws",
      "key_spec": "symmetric_default",
      "multi_region": false,
      "name": "alias/aws/secretsmanager",
      "origin": "aws_kms",
      "platform_type": "aws",
      "purpose": "encrypt_decrypt",
      "region": "us-east-1",
      "rotation": {
        "enabled": true,
        "rotation_period_days": 365
      },
      "status": "enabled"
    }
}
```


Public Key Scan Results


```
{
   "type": "pubkey",
    "timestamp": "2025-12-15T11:39:50-05:00",
    "urn":
"urn:pubkey:sha256:b05aaba4c71cb0acd32f4c24bf16ea328bafd1da265824d6ff168113cf12859d",
    "url": "https://console.aws.amazon.com/kms/home?region=us-east-1#/kms/keys/
4fa5d66e-2c5f-4504-a3fe-30545631eeb6",
    "extra": {
      "account_id": "558162184545",
      "alias": "",
      "arn": "arn:aws:kms:us-east-1:558162184545:key/4fa5d66e-2c5f-4504-
a3fe-30545631eeb6",
      "created_date": "2025-10-22T18:34:56Z",
      "description": "Test RSA 4096 asymmetric signing key for high security",
      "hsm_protected": true,
      "key_id": "4fa5d66e-2c5f-4504-a3fe-30545631eeb6",
      "key_manager": "customer",
      "multi_region": false,
      "name": "4fa5d66e-2c5f-4504-a3fe-30545631eeb6",
      "origin": "aws_kms",
      "platform_type": "aws",
      "purpose": "sign_verify",
      "region": "us-east-1",
      "status": "enabled"
    },
    "pubkey_pem": "MIICIjANB.."
}
```


URN Generation:

- For symmetric and HMAC keys: Generated using key name and ARN
- For asymmetric keys: Generated using hash of the public key PEM body for uniqueness

Error Output: If an error occurs, the plugin outputs a structured JSON object with error details under `data.errorDetails` and a summary message in `data.errorMessage`.

Field Descriptions

- `type` : "symkey" for symmetric/HMAC keys, "pubkey" for asymmetric keys
- `timestamp` : RFC3339 timestamp of when the scan was performed (e.g., "2024-08-07T10:30:00-04:00"). May include a timezone offset depending on system configuration.
- `urn` : Unique identifier generated based on key type and content
- `url` : Direct link to key in AWS Console
- `pubkey_pem` : RFC 7468 compliant PEM-encoded public key (base64 body without headers/footers/newlines) - asymmetric keys only
- `extra` : Additional deployment-specific metadata from KMS with standardized snake_case field names (see Extra Fields section below)

Extra Fields

The `extra` object contains AWS-specific metadata about the key. The fields differ based on key type:

Symmetric and HMAC Key Extra Fields

For symmetric and HMAC keys (`type: "symkey"`), the following fields are included:

- `platform_type` (string): Always `"aws"` indicating the key is from AWS KMS
- `name` (string): Human-readable key name (alias if available, otherwise key ID)
- `status` (string): Current key state (values: `"enabled"` , `"disabled"` , `"pending_deletion"` , `"pending_import"` , `"unavailable"`)
- `region` (string): AWS region where the key resides (e.g., `"us-east-1"`)
- `account_id` (string): AWS account ID extracted from the key ARN
- `key_id` (string): Unique identifier for the key (UUID format)
- `alias` (string): Key alias name (e.g., `"alias/aws/s3"`) or empty string if no alias exists
- `key_spec` (string): Key specification indicating algorithm (values: `"symmetric_default"` , `"hmac_224"` , `"hmac_256"` , `"hmac_384"` , `"hmac_512"`)
- `description` (string): User-provided description of the key
- `created_date` (string): ISO 8601 timestamp of when the key was created (e.g., `"2025-10-22T18:29:42Z"`)
- `arn` (string): Full AWS ARN of the key (e.g., `"arn:aws:kms:us-east-1:123456789012:key/abcd-1234"`)
- `purpose` (string): Key usage purpose (values: `"encrypt_decrypt"` , `"generate_verify_mac"`)
- `origin` (string): Key origin source (values: `"aws_kms"` , `"external"` , `"aws_cloudhsm"` , `"external_key_store"`)
- `hsm_backed` (boolean): Whether the key is backed by a Hardware Security Module (all AWS KMS keys are HSM-backed)
- `cipher_suites` (array of strings): Supported encryption algorithms (e.g., `["SYMMETRIC_DEFAULT"]` , `["HMAC_SHA_256"]`)
- `cryptographic_algorithm` (string): Primary cryptographic algorithm (e.g., `"AES"` , `"HMAC"`)
- `cryptographic_length` (integer): Key length in bits (e.g., 256, 384, 512)
- `key_manager` (string): Entity managing the key (values: `"aws"` , `"customer"`)

- `multi_region` (boolean): Whether the key is replicated across multiple regions
- `rotation` (object): Key rotation information
  - `enabled` (boolean): Whether automatic key rotation is enabled
  - `rotation_period_days` (integer): Rotation period in days (only present when rotation is enabled)

Asymmetric Key Extra Fields (Public Keys)

For asymmetric keys (`type: "pubkey"`), the following fields are included:

- `platform_type` (string): Always `"aws"` indicating the key is from AWS KMS
- `created_date` (string): ISO 8601 timestamp of when the key was created
- `status` (string): Current key state
  (values: `"enabled"` , `"disabled"` , `"pending_deletion"` , `"pending_import"` , `"unavailable"`)
- `name` (string): Human-readable key name (alias if available, otherwise key ID)
- `origin` (string): Key origin source (values: `"aws_kms"` , `"external"` , `"aws_cloudhsm"` , `"external_key_store"`)
- `purpose` (string): Key usage purpose (values: `"sign_verify"` , `"encrypt_decrypt"`)
- `hsm_protected` (boolean): Whether the key is protected by a Hardware Security Module
- `alias` (string): Key alias name or empty string if no alias exists
- `arn` (string): Full AWS ARN of the key
- `description` (string): User-provided description of the key
- `key_id` (string): Unique identifier for the key (UUID format)
- `key_manager` (string): Entity managing the key (values: `"aws"` , `"customer"`)
- `region` (string): AWS region where the key resides
- `multi_region` (boolean): Whether the key is replicated across multiple regions
- `account_id` (string): AWS account ID extracted from the key ARN (only included if successfully extracted)

Note: For asymmetric keys, intrinsic cryptographic properties like `key_spec` , `cipher_suites` , `cryptographic_algorithm` , `cryptographic_length` , and `cryptographic_curve` are NOT included in the extras as they can be derived from the `pubkey_pem` field. Rotation is also not applicable to asymmetric keys.

Note: Timestamps in output may include a timezone offset (e.g., `-04:00`) or `Z` for UTC, depending on the system time configuration where the plugin or mock server is run.

AWS Secrets Manager plugin output

Output Format

Secret Scan Results

The plugin generates secret scan results in the following standardized format:

```
{
    "type": "secret",
    "timestamp": "2025-12-15T12:13:41-05:00",
    "urn": "urn:secret:name:test/database/mysql-prod:5eba86d3-
c1be-47dc-95ee-0da8d188fd76",
```

```
    "url": "https://console.aws.amazon.com/secretsmanager/home?region=us-east-1#/secret?
name=arn:aws:secretsmanager:us-east-1:558162184545:secret:test/database/mysql-prod-
bpkYrP",
    "extra": {
      "account_id": "558162184545",
      "created_date": "2025-10-22T18:29:42Z",
      "current_version": "5eba86d3-c1be-47dc-95ee-0da8d188fd76",
      "description": "Production MySQL database credentials",
      "expiration_date": "",
      "last_access_time": "",
      "last_modified_date": "2025-10-22T18:29:43Z",
      "name": "test/database/mysql-prod",
      "platform_type": "aws",
      "region": "us-east-1",
      "revision": "5eba86d3-c1be-47dc-95ee-0da8d188fd76",
      "rotation": {
        "enabled": false
      },
      "status": "active",
      "version_count": 1
    }
}
```


URN Generation: The `urn` field is generated using the secret name and current version ID for uniqueness and consistency.

Error Output: If an error occurs, the plugin outputs a structured JSON object with error details under `data.errorDetails` and a summary message in `data.errorMessage`.

Field Descriptions

- `type`: Always "secret" for secret assets
- `timestamp`: RFC3339 timestamp of when the scan was performed (e.g., "2024-08-25T10:30:00-04:00"). May include a timezone offset or Z for UTC, depending on system configuration.
- `urn`: Unique identifier generated from secret name and version
- `url`: Direct link to secret in AWS Console
- `extra`: Additional metadata from Secrets Manager (see Extra Fields section below)

Extra Fields

The `extra` object contains AWS-specific metadata about the secret. The following fields are included:

Standard Fields (Always Included)

- `platform_type` (string): Always `"aws"` indicating the secret is from AWS Secrets Manager
- `name` (string): Name/identifier of the secret (e.g., `"test/database/mysql-prod"`)
- `created_date` (string): ISO 8601 timestamp of when the secret was created (e.g., `"2025-10-22T18:29:42Z"`)
- `last_modified_date` (string): ISO 8601 timestamp of when the secret was last modified (e.g., `"2025-10-22T18:29:43Z"`)
- `expiration_date` (string): ISO 8601 timestamp of when the secret expires or was deleted (empty string if not applicable)

- `status` (string): Current secret state (values: `"active"`, `"deleted"`)
- `rotation` (object): Secret rotation configuration
  - `enabled` (boolean): Whether automatic rotation is enabled
  - `next_rotation_date` (string): ISO 8601 timestamp of the next scheduled rotation (only present when rotation is enabled)
  - `lambda_arn` (string): ARN of the Lambda function used for rotation (only present when rotation is enabled)
  - `rotation_rules` (object): Rotation schedule rules (only present when rotation is enabled)
    - `automatically_after_days` (integer): Number of days between automatic rotations
    - `duration` (string): Duration window for rotation
    - `schedule_expression` (string): Cron or rate expression for rotation schedule
- `revision` (string): Current version ID of the secret (UUID format)
- `current_version` (string): Current version ID of the secret (same as revision, UUID format)
- `version_count` (integer): Total number of versions for this secret
- `last_access_time` (string): ISO 8601 timestamp of when the secret was last accessed (empty string if not available)
- `region` (string): AWS region where the secret is stored (e.g., `"us-east-1"`)

Conditional Fields (Included When Available)

- `account_id` (string): AWS account ID extracted from the secret ARN (only included if successfully extracted from ARN)
- `description` (string): User-provided description of the secret (only included if non-empty)
- `kms_key_id` (string): ARN or ID of the KMS key used to encrypt the secret (only included if non-empty)
- `tags` (object): Key-value pairs of tags associated with the secret (only included if tags exist)

Note: The plugin extracts only metadata about secrets, not the actual secret values themselves. All timestamp fields use ISO 8601 format.

AWS CloudFront plugin output

Output Format

The plugin generates JSON output in the following format for each discovered certificate:


```
{
 "result_type": "scan",
 "plugin_id": "aws-cloudfront-plugin",
 "plugin_version": "1.1.0",
 "data": {
  "type": "cert",
  "timestamp": "2026-01-21T15:33:19+05:30",
  "urn": "URN_NUMBER",
  "url": "https://console.aws.amazon.com/cloudfront/v3/home?region=us-east-1#/distributions/DISTRIBUTION_ID",
```

```
  "extra": {
    "distribution_id": "DISTRIBUTION_ID",
    "distribution_arn": "arn:aws:cloudfront::ACCOUNT_ID:distribution/DISTRIBUTION_ID",
    "domain_name": "d21rsch7py4ncb.cloudfront.net",
    "distribution_status": "Deployed",
    "certificate_domain_name": "example.com",
    "certificate_arn": "arn:aws:acm:us-east-1:ACCOUNT_ID:certificate/...",
    "certificate_source": "acm",
    "minimum_protocol_version": "TLSv1.3_2025",
    "ssl_support_method": "sni-only",
    "issuer": "Entrust",
    "subject": "CN=example.com",
    "serial_number": "2b:41:0b:12:32:dc:fa:f0:84:86:a5:3c:ca:b7:f4:c6:f7:2d:55:18",
    "not_before": "2025-12-17T23:42:49Z",
    "not_after": "2026-12-17T23:42:49Z",
    "key_algorithm": "RSA-2048",
    "signature_algorithm": "SHA256WITHRSA"
  },
  "cert_pem": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n"
 }
}
```


Output Fields Explanation

Field	Description
distribution_id	CloudFront distribution ID (e.g., E1ZU6UPGYYF07)
distribution_arn	AWS ARN of the distribution
domain_name	CloudFront domain name (e.g., d21rsch7py4ncb.cloudfront.net)
distribution_status	Distribution status (Deployed, Deploying, Disabled)
certificate_domain_name	Certificate domain name (e.g., example.com)
certificate_arn	ACM certificate ARN
certificate_source	Source of certificate (acm = AWS Certificate Manager)

Field	Description
minimum_protocol_version	Minimum TLS version supported (e.g., TLSv1.3_2025)
ssl_support_method	SSL support method (sni-only, vip)
issuer	Certificate issuer name
subject	Certificate subject (CN, O, C, etc.)
serial_number	Certificate serial number
not_before	Certificate validity start date (RFC3339 format)
not_after	Certificate expiration date (RFC3339 format)
key_algorithm	Key algorithm (RSA-2048, EC-prime256v1, etc.)
signature_algorithm	Signature algorithm (SHA256WITHRSA, etc.)
cert_pem	Full certificate chain in PEM format (RFC 7468 compliant)

Supported Distributions

The plugin supports CloudFront distributions that use:

- **Custom ACM Certificates**: Certificates from AWS Certificate Manager
- **Multiple Distributions**: Scans all distributions with custom SSL certificates in a single operation
- **Distribution Configuration**: Captures aliases, origins, cache behaviors, logging, geo-restrictions

The plugin does not currently support:

- IAM certificates (legacy - use ACM instead)
- Default CloudFront certificates

Limitations

1. **No Incremental Scans**: Each scan processes all distributions (see Why No Incremental Scans section)
2. **CloudFront Global Service**: Even when scanning a specific region, CloudFront distributions are global resources but must be accessed through the CloudFront API
3. **Certificate Chain**: The plugin extracts metadata from the leaf certificate in the chain. Full chain is included in PEM output for browser/tool validation
4. **No Private Key Export**: Private keys are never exported - only public certificate data is returned
5. **No IAM Certificates**: Legacy IAM certificates are logged but not processed

Troubleshooting

Connection Issues

Error: "InvalidClientTokenId"

- Cause: Access key ID is invalid or credentials are expired
- Solution: Verify credentials in config file and ensure access key is active in IAM console

Error: "AccessDenied"

- Cause: IAM user lacks required permissions
- Solution: Verify IAM policy includes all required actions (see IAM Permissions section)

Error: "NoSuchEntity"

- Cause: Invalid AWS account or region
- Solution: Verify account ID and region in configuration

Certificate Issues

No Certificates Found

- Cause: No CloudFront distributions with custom ACM certificates
- Solution: Create a CloudFront distribution with custom ACM certificate (see AWS CloudFront documentation)

"Failed to decode PEM certificate"

- Cause: Certificate data is malformed or corrupted
- Solution: Verify certificate in ACM console is valid; regenerate if necessary

"Failed to parse certificate"

- Cause: Certificate parsing error or unsupported certificate format
- Solution: Ensure certificate is in valid X.509 format; contact AWS Support if issue persists

Related Documentation

- AWS CloudFront Documentation
- AWS Certificate Manager Documentation

Support

For issues or questions about this plugin:

1. Review the troubleshooting section above
2. Check CloudFront plugin logs for detailed error messages
3. Verify AWS credentials and IAM permissions
4. Consult AWS CloudFront and Certificate Manager documentation

AWS Elastic Load Balancer Plugin output

Output Format

Certificate Scan Result


```
{
 "result_type": "scan",
 "plugin_id": "aws-elasticloadbalancer-plugin",
 "plugin_version": "1.1.0",
 "data": {
  "type": "cert",
  "timestamp": "2026-02-02T10:00:00Z",
  "urn": "urn:cert:sha256:...",
  "url": "https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:...",
  "extra": {
   "load_balancer_arn": "arn:aws:elasticloadbalancing:...",
   "load_balancer_name": "my-alb",
   "load_balancer_type": "application",
   "scheme": "internet-facing",
   "dns_name": "my-alb-123456789.us-east-1.elb.amazonaws.com",
   "listener_arn": "arn:aws:elasticloadbalancing:...",
   "listener_port": 443,
   "listener_protocol": "HTTPS",
   "ssl_policy": "ELBSecurityPolicy-TLS13-1-2-2021-06",
   "certificate_arn": "arn:aws:acm:...",
   "certificate_domain_name": "example.com",
   "certificate_status": "ISSUED",
   "is_default": true,
   "target_groups": [...],
   "ssl_policy_details": {...},
   "platform_type": "aws"
  },
  "cert_pem": "-----BEGIN CERTIFICATE-----\n..."
 }
}
```

Error Handling and Troubleshooting

Common Errors

1. Authentication Errors

Error Message:

```
Error: InvalidClientTokenId: The security token included in the request is invalid
```

Causes:

- Incorrect Access Key ID
- Incorrect Secret Access Key
- Access Key has been deleted or rotated
- Access Key is inactive

Resolution:

```
# Step 1: Verify credentials in config.json are correct
cat config.json

# Step 2: Re-generate access keys in AWS IAM console
# AWS Console → IAM → Users → Your User → Security Credentials → Create Access Key

# Step 3: Update config.json with new credentials

# Step 4: Re-run test command
./aws-elasticloadbalancer-plugin test -i config.json
```

2. Permission Errors

Error Message:

```
Error: AccessDenied: User is not authorized to perform: elasticloadbalancing:DescribeLoadBalancers
```

Causes:

- IAM policies not attached to user
- Policies are expired or have conditions
- User doesn't have required permissions

Resolution:

1. Go to **AWS IAM Console** → **Users** → Your User
2. Click **Add Permissions** → **Attach Existing Policies**
3. Attach: `ElasticLoadBalancingReadOnly` + `AWSCertificateManagerReadOnly`
4. Wait 30 seconds for policy to propagate
5. Retry plugin test

3. Region Errors

Error Message:

```
Error: The region is not a valid AWS region
```

Causes:

- Typo in region name
- Region not enabled for your AWS account
- Region doesn't exist

Resolution:

```
# Valid AWS regions:
us-east-1, us-west-1, us-west-2
eu-west-1, eu-central-1, eu-north-1
ap-southeast-1, ap-northeast-1, ap-south-1
ca-central-1, sa-east-1, ap-east-1
```

4. No Certificates Found

Error Message:

`Scan completed: 0 assets found`

This is normal if:

- No load balancers exist in the region
- Load balancers only have HTTP listeners (not HTTPS/TLS)
- Certificates are not attached to listeners

Verification:

1. Go to **AWS EC2 Console → Load Balancers**
2. Check if any load balancers exist
3. Click on each load balancer → **Listeners** tab
4. Verify at least one HTTPS or TLS listener exists
5. Check if certificate is selected in listener configuration

Connection Troubleshooting

Error	Cause	Solution
`Unable to reach AWS`	Network connectivity	Check internet connection, firewall rules
`Connection timeout`	AWS service unavailable	Wait and retry, check AWS status page
`Certificate parsing failed`	Invalid PEM format	Ensure certificate is valid, use `openssl verify`
`Invalid JSON output`	Plugin error	Check plugin version, review logs

Validation Steps

Step 1: Verify Credentials

```
# Test with AWS CLI to ensure credentials work
aws iam get-user --profile default
```

Step 2: Verify Policies

```
# List policies attached to your user
aws iam list-attached-user-policies --user-name your-username
```

Step 3: Verify Load Balancers Exist

```
# List all load balancers in region
aws elbv2 describe-load-balancers --region us-east-1
```

Step 4: Verify HTTPS Listeners

```
# List all listeners for a specific load balancer
aws elbv2 describe-listeners --load-balancer-arn your-lb-arn --region us-east-1
```

Security Best Practices

Credential Management

1. **Use AWS Managed Policies**
   - Easier to maintain
   - AWS regularly updates for new services
   - Least-privilege by design
2. **Rotate Access Keys Regularly**
   - Rotate every 90 days
   - Maintain two active keys for zero-downtime rotation
   - Delete old keys after rotation verified
3. **Use Temporary Credentials (STS)**
   - For production environments
   - Use IAM roles for EC2/Lambda execution
   - Set token expiration to 1-2 hours
4. **Secure Configuration Files**
   - Never commit credentials to version control
   - Use `.gitignore` for config files
   - Store in secure credential manager (Vault, Secrets Manager)

Runtime Security

- **Enable CloudTrail**: Monitor API calls
- **Use VPC Endpoints**: Private connectivity to AWS services
- **Enable MFA**: Multi-factor authentication for IAM users
- **Regular Audits**: Review access patterns and permissions

Performance Optimization

Scan Performance

Factor	Impact	Optimization
Load balancer count	Direct	Run scans during off-peak hours
Region count	Linear	Parallelize scans across regions
Certificate complexity	Minimal	No optimization needed
API rate limits	Rare	Unlikely to hit AWS rate limits

Multi-Region Scanning

```
# Sequential (slow)
for region in us-east-1 eu-west-1 ap-southeast-1; do
  ./aws-elasticloadbalancer-plugin scan -i config-$region.json -o output-$region.json
done

# Parallel (fast)
for region in us-east-1 eu-west-1 ap-southeast-1; do
  ./aws-elasticloadbalancer-plugin scan -i config-$region.json -o output-$region.json &
done
wait
```

Supported Load Balancer Types

The plugin supports the following ELB load balancers:

Application Load Balancer (ALB)

- **Purpose**: HTTP/HTTPS traffic distribution
- **Listener Protocols**: HTTP, HTTPS
- **Certificate Support**: ACM, IAM
- **SSL Policies**: Supported (extract supported protocols/ciphers)
- **Target Types**: Instance, IP, Lambda

Network Load Balancer (NLB)

- **Purpose**: Ultra-high performance TCP/TLS
- **Listener Protocols**: TLS, TCP, UDP
- **Certificate Support**: ACM
- **SSL Policies**: Supported
- **Target Types**: Instance, IP, ALB

Classic Load Balancer (CLB)

- **Purpose**: Legacy load balancing (deprecated)
- **Listener Protocols**: HTTP, HTTPS, TCP, SSL
- **Certificate Support**: ACM, IAM (limited)
- **SSL Policies**: Basic support
- **Note**: AWS recommends migration to ALB/NLB

Comparison: ALB vs NLB vs CLB

Feature	ALB	NLB	CLB
Layer	Layer 7	Layer 4	Layer 4/7
Use Case	Web apps	High performance	Legacy
Throughput	Moderate	Ultra-high	Low
Latency	Medium	Ultra-low	Higher
SSL Policy	Advanced	Advanced	Basic
Certificate Types	ACM, IAM	ACM	ACM, IAM
Recommended	YES	For performance	Deprecated

Version History

Version	Date	Changes
1.1.0	2026-02-03	Real AWS API integration, AWS managed policies, comprehensive error handling
1.0.0	2026-01-15	Initial POC with mock data

Support and Troubleshooting

For issues or questions:

1. **Check this README** - Most issues are covered in the Troubleshooting section
2. **Verify AWS Permissions** - Ensure IAM policies are correctly attached
3. **Review Plugin Logs** - Check console output for detailed error messages
4. **Test Connection First** - Always run `test` command before `scan` command
5. **Review AWS Console** - Verify load balancers and certificates exist in AWS

Additional Resources

- **AWS ELB Documentation**: https://docs.aws.amazon.com/elasticloadbalancing/
- **AWS ACM Documentation**: https://docs.aws.amazon.com/acm/
- **AWS IAM Documentation**: https://docs.aws.amazon.com/iam/
- **SSL/TLS Certificates**: https://tools.ietf.org/html/rfc7468
- **AWS SDKv2 for Go**: https://github.com/aws/aws-sdk-go-v2

AWS CloudHSM plugin output

Output Format

Certificate Discovery


```
{
  "result_type": "scan",
  "plugin_id": "aws-cloudhsm-plugin",
  "plugin_version": "1.1.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-02-27T10:11:37Z",
    "urn":
"urn:cert:sha256:1009c0242c95817ff7323482637a7127a63d91bb50c0789a74b2d812ab2fcd34",
    "url": "https://console.aws.amazon.com/cloudhsm/home?region=us-east-1#/clusters/
cluster-123/certificates",
    "cert_pem": "MIIDfTCCAm...(base64 without headers)...",
    "extra": {
      "platform_type": "aws",
      "hsm_backed": true,
      "cluster_id": "cluster-123",
      "region": "us-east-1",
      "label": "my-cert",
      "cert_reference": "0x0000000000000001",
      "cert_type": "X.509",
      "token_label": "hsm1",
      "token_serial": "",
      "firmware_version": "10.23"
    }
  }
}
```


Private Key Discovery


```
{
  "result_type": "scan",
  "plugin_id": "aws-cloudhsm-plugin",
  "plugin_version": "1.1.0",
  "data": {
    "type": "privkey",
    "timestamp": "2026-02-27T10:11:37Z",
    "urn": "urn:privkey:name:my-rsa-key:0x0000000000000003",
    "url": "https://console.aws.amazon.com/cloudhsm/home?region=us-east-1#/clusters/
cluster-123/keys",
```

```
    "extra": {
      "platform_type": "aws",
      "hsm_backed": true,
      "cluster_id": "cluster-123",
      "region": "us-east-1",
      "label": "my-rsa-key",
      "key_reference": "0x0000000000000003",
      "key_type": "CKK_RSA",
      "key_length": 2048,
      "extractable": false,
      "sensitive": true,
      "origin": "aws_created",
      "token_label": "hsm1",
      "token_serial": "",
      "firmware_version": "10.23",
      "usage_flags": {
        "sign": true,
        "decrypt": true,
        "unwrap": true,
        "derive": false
      }
    }
  }
}
```


Public Key Discovery


```
{
  "result_type": "scan",
  "plugin_id": "aws-cloudhsm-plugin",
  "plugin_version": "1.1.0",
  "data": {
    "type": "pubkey",
    "timestamp": "2026-02-27T10:11:37Z",
    "urn":
"urn:pubkey:sha256:b05aaba4c71cb0acd32f4c24bf16ea328bafd1da265824d6ff168113cf12859d",
    "url": "https://console.aws.amazon.com/cloudhsm/home?region=us-east-1#/clusters/
cluster-123/keys",
    "pubkey_pem": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A...(base64 without headers)...",
    "extra": {
      "platform_type": "aws",
      "hsm_backed": true,
      "cluster_id": "cluster-123",
      "region": "us-east-1",
      "label": "my-rsa-key-pub",
      "key_reference": "0x0000000000000004",
      "key_type": "CKK_RSA",
      "extractable": false,
      "sensitive": false,
      "token_label": "hsm1",
      "token_serial": "",
      "firmware_version": "10.23",
      "usage_flags": {
```

```
        "verify": true,
        "encrypt": true,
        "wrap": true,
        "derive": false
      }
    }
  }
}
```


Symmetric Key Discovery


```
{
  "result_type": "scan",
  "plugin_id": "aws-cloudhsm-plugin",
  "plugin_version": "1.1.0",
  "data": {
    "type": "symkey",
    "timestamp": "2026-02-27T10:11:37Z",
    "urn": "urn:symkey:name:my-aes-key:0x0000000000000005",
    "url": "https://console.aws.amazon.com/cloudhsm/home?region=us-east-1#/clusters/
cluster-123/keys",
    "extra": {
      "platform_type": "aws",
      "hsm_backed": true,
      "cluster_id": "cluster-123",
      "region": "us-east-1",
      "label": "my-aes-key",
      "key_reference": "0x0000000000000005",
      "key_type": "CKK_AES",
      "key_length": 256,
      "extractable": false,
      "sensitive": true,
      "origin": "aws_created",
      "token_label": "hsm1",
      "token_serial": "",
      "firmware_version": "10.23",
      "usage_flags": {
        "encrypt": true,
        "decrypt": true,
        "wrap": true,
        "unwrap": true,
        "sign": false,
        "verify": false,
        "derive": false
      }
    }
  }
}
```


URN Generation

- **Certificates:** `urn:cert:sha256:<fingerprint>` - SHA256 fingerprint of DER-encoded certificate
- **Public Keys:** `urn:pubkey:sha256:<fingerprint>` (when PEM available)
  or `urn:pubkey:name:<label>:<reference>` (fallback)
- **Private Keys:** `urn:privkey:name:<label>:<reference>` - Name-based (no key material exported)
- **Symmetric Keys:** `urn:symkey:name:<label>:<reference>` - Name-based (no key material exported)

Extra Fields

The `extra` object contains CloudHSM-specific metadata. Fields vary based on object type:

Common Fields (All Types)

Field	Type	Description
`platform_type`	string	Always `"aws"`
`hsm_backed`	boolean	Always `true` for CloudHSM objects
`cluster_id`	string	CloudHSM cluster ID
`region`	string	AWS region
`label`	string	Object label from PKCS#11
`token_label`	string	HSM token label
`token_serial`	string	HSM token serial number
`firmware_version`	string	HSM firmware version (e.g., `"10.23"`)

Key-Specific Fields

Field	Type	Description
`key_reference`	string	PKCS#11 object handle (hex format)
`key_type`	string	PKCS#11 key type (`CKK_RSA` , `CKK_EC` , `CKK_AES` , `CKK_DES3`)
`key_length`	int	Key length in bits (when available)
`extractable`	boolean	Whether key can be exported from HSM
`sensitive`	boolean	Whether key is marked sensitive
`origin`	string	Key origin: `"aws_created"` (generated on HSM) or `"imported"`
`usage_flags`	object	Allowed operations: `sign` , `verify` , `encrypt` , `decrypt` , `wrap` , `unwrap` , `derive`

Certificate-Specific Fields

Field	Type	Description
`cert_reference`	string	PKCS#11 object handle (hex format)
`cert_type`	string	Certificate type (typically `"X.509"`)

Note: Certificate intrinsic fields (subject, issuer, serial number, validity dates) are NOT included in `extra` when `cert_pem` is present, as they can be derived from the PEM data.

Azure Key Vault plugin output

Output Format

The plugin outputs a JSON file containing discovered assets, their metadata, and (for public keys/certificates) the PEM-encoded public key body.

Certificate Output Format


```
{
   "type": "cert",
    "timestamp": "2025-12-15T12:15:40-05:00",
    "urn":
"urn:cert:sha256:dbe504b31dfecf6bb7ec13a4b3debae8d145c8ff1c2b3dc7d5969794320e846c",
    "url": "https://certhubsourcedestination.vault.azure.net/certificates/
autorenew20240818/27f0e041f6ae498393cf5f2bf194d223",
    "extra": {
      "cert_name": "autorenew20240818",
      "exportable": true,
      "issuer_name": "Unknown",
      "origin": "imported",
      "platform_type": "azure",
      "purpose": "key_encipherment",
      "recovery_level": "Recoverable+Purgeable",
      "reuse_key": false,
      "status": "enabled",
      "tenant_id": "586c621b-e6b8-416e-8f9c-3acba8ec8fe4",
      "vault_name": "certhubsourcedestination"
    },
    "cert_pem": "MIIFIDCCBAigAw..."
}
```


Public Key Output Format


```
{
    "type": "pubkey",
    "timestamp": "2025-12-15T12:15:45-05:00",
    "urn":
"urn:pubkey:sha256:32f5b4f5d669a0079fa742ee45ef5e6f7d5e9f2ea0eccf0e8264d9869ef1bcc2",
    "url": "https://certhubsourcedestination.vault.azure.net/keys/
AzureTest02/029ea1d008ac43c89f47df652f4704ab",
    "extra": {
      "created_date": "2023-06-16T11:12:00-04:00",
      "expiration_date": "2026-06-16T11:11:59-04:00",
      "exportable": false,
      "hsm_protected": false,
      "key_version": "029ea1d008ac43c89f47df652f4704ab",
      "name": "AzureTest02",
      "origin": "azure_keyvault",
      "platform_type": "azure",
      "purpose": "encrypt_decrypt",
      "status": "enabled",
      "tenant_id": "586c621b-e6b8-416e-8f9c-3acba8ec8fe4",
      "updated_date": "2023-06-16T11:12:00-04:00",
      "vault_name": "certhubsourcedestination"
    },
    "pubkey_pem": "MIIBIjANBgk.."
}
```

Private Key Output Format


```
{
    "type": "privkey",
    "timestamp": "2025-12-15T12:15:59-05:00",
    "urn": "urn:privkey:name:damazure:b8819c19e7184dab9dbb0bf026c4b39a",
    "url": "https://certhubsourcedestination.vault.azure.net/secrets/damazure/
b8819c19e7184dab9dbb0bf026c4b39a",
    "extra": {
      "content_type": "application/x-pem-file",
      "created_date": "2025-09-05T13:19:53-04:00",
      "cryptographic_algorithm": "RSA",
      "cryptographic_length": 2048,
      "expiration_date": "2028-09-05T13:19:48-04:00",
      "key_name": "damazure",
      "key_version": "b8819c19e7184dab9dbb0bf026c4b39a",
      "platform_type": "azure",
      "recovery_level": "Recoverable+Purgeable",
      "status": "enabled",
      "tenant_id": "586c621b-e6b8-416e-8f9c-3acba8ec8fe4",
      "updated_date": "2025-09-05T13:19:53-04:00",
      "vault_name": "certhubsourcedestination"
    }
}
```


URN Generation: The `urn` field is generated using a hash of the certificate PEM body for uniqueness and consistency.

Error Output: If an error occurs, the plugin outputs a structured JSON object with error details under `data.errorDetails` and a summary message in `data.errorMessage`.

Field Descriptions

- `type` : Asset type - `"cert"` for certificates, `"pubkey"` for public keys, `"privkey"` for private keys, `"symkey"` for symmetric keys, `"secret"` for generic secrets
- `timestamp` : RFC3339 timestamp of when the scan was performed, in local timezone (e.g., `2024-08-07T10:30:00-07:00` for PDT or `2024-08-07T17:30:00Z` for UTC)
- `urn` : Unique identifier generated from asset content hash or name
- `url` : Direct link to asset in Azure Portal
- `cert_pem` / `pubkey_pem` : RFC 7468 compliant PEM-encoded certificate or public key (base64 body without headers/footers or newlines)
- `extra` : Additional metadata from Azure Key Vault (see Extra Fields section below)

Extra Fields

The `extra` object contains Azure-specific metadata about the asset. The fields differ based on asset type.

Common Fields (Present in All Asset Types)

The following fields are present across all asset types:

- `platform_type` (string): Always `"azure"` indicating the asset is from Azure Key Vault
- `vault_name` (string): Name of the Azure Key Vault containing the asset
- `status` (string): Current asset state (values: `"enabled"`, `"disabled"`)
- `created_date` (string): ISO 8601 timestamp of when the asset was created
- `updated_date` (string): ISO 8601 timestamp of last modification

Conditional Common Fields (only present if vault metadata is available):

- `tenant_id` (string): Azure Active Directory tenant ID
- `tenant_name` (string): Azure AD tenant name
- `subscription_id` (string): Azure subscription ID
- `subscription_name` (string): Azure subscription name
- `resource_group` (string): Azure resource group name
- `location` (string): Azure region/location

Certificate-Specific Fields (`type: "cert"`)

In addition to the common fields, certificates include:

- `cert_name` (string): Name/identifier of the certificate
- `recovery_level` (string): Recovery protection level (values: `"Recoverable+Purgeable"`, `"Recoverable"`, `"Purgeable"`, `"CustomizedRecoverable+Purgeable"`, `"CustomizedRecoverable"`)
- `issuer_name` (string): Name of the certificate issuer (e.g., `"Self"`, `"Unknown"`, or CA name)
- `origin` (string): Certificate origin (values: `"self_signed"`, `"imported"`, `"ca_issued"`)
- `ca_name` (string): Certificate Authority name (only present when origin is `"ca_issued"`)
- `exportable` (boolean): Whether the certificate's private key can be exported
- `reuse_key` (boolean): Whether the key can be reused for certificate renewal
- `purpose` (string): Comma-separated list of certificate purposes derived from Key Usage and Extended Key Usage (e.g., `"digital_signature, key_encipherment, server_auth, client_auth"`)

Fallback fields (when certificate body is unavailable):

- `cert_version` (string): Specific version identifier of the certificate
- `not_after` (string): Certificate expiration date (ISO 8601 format)
- `not_before` (string): Certificate validity start date (ISO 8601 format)

Note: Key Usage and Extended Key Usage fields are NOT included in extras when `cert_pem` is present, as they are intrinsic to the certificate and can be parsed from the PEM.

Public Key-Specific Fields (`type: "pubkey"`)

In addition to the common fields, public keys include:

- `name` (string): Key name from Azure Key Vault
- `key_version` (string): Specific version identifier of the key (UUID format)
- `origin` (string): Key origin (value: `"azure_keyvault"`)

- `hsm_protected` (boolean): Whether the key is backed by a Hardware Security Module
- `exportable` (boolean): Whether the key can be exported from the vault
- `purpose` (string): Key usage purpose (values: `"encrypt_decrypt"`, `"sign_verify"`, `"wrap_unwrap"`)
- `expiration_date` (string): ISO 8601 timestamp of key expiration (only present if set)

Note: Intrinsic cryptographic properties like algorithm type, key length, and curve are NOT included in extras as they can be derived from the `pubkey_pem` field.

Private Key-Specific Fields (`type: "privkey"`)

In addition to the common fields, private keys include:

- `key_name` (string): Name of the private key
- `key_version` (string): Specific version identifier of the key
- `content_type` (string): MIME type of the secret content (e.g., "application/x-pem-file")
- `recovery_level` (string): Recovery protection level
- `expiration_date` (string): ISO 8601 timestamp of key expiration (only present if set)

Symmetric Key-Specific Fields (`type: "symkey"`)

In addition to the common fields, symmetric keys include:

- `key_name` (string): Name of the symmetric key
- `key_version` (string): Specific version identifier of the key
- `content_type` (string): MIME type of the secret content
- `recovery_level` (string): Recovery protection level
- `expiration_date` (string): ISO 8601 timestamp of key expiration (only present if set)

Generic Secret-Specific Fields (`type: "secret"`)

In addition to the common fields, generic secrets include:

- `name` (string): Name/identifier of the secret
- `revision` (string): Current version ID of the secret
- `current_version` (string): Current version ID of the secret (same as revision)
- `content_type` (string): MIME type of the secret content
- `recovery_level` (string): Recovery protection level
- `expiration_date` (string): ISO 8601 timestamp of secret expiration (only present if set)

Note: Azure Key Vault doesn't expose rotation policies via SDK for secrets, so rotation information is not included.

GCP Certificate Manager plugin output

Output Format

Certificate Scan Results

Each discovered certificate generates a result with the following structure:


```
{
     "type": "cert",
    "timestamp": "2025-12-15T12:18:07-05:00",
    "urn":
"urn:cert:sha256:302595b8e1a7d6a1baa33577a22ddd64b6c07e5846117bbd22ebb07727f11e75",
    "url": "https://console.cloud.google.com/security/ccm/list/lbCertificates?
project=gcp-project-corp-dev-discovery",
    "extra": {
      "create_time": "2025-11-17T20:26:42.777178748Z",
      "description": "",
      "location": "global",
      "name": "projects/gcp-project-corp-dev-discovery/locations/global/certificates/
test2",
      "origin": "imported",
      "platform_type": "gcp",
      "project_id": "gcp-project-corp-dev-discovery",
      "self_managed": {
        "has_pem_certificate": false,
        "has_pem_private_key": false
      },
      "update_time": "2025-11-18T02:31:08.609078336Z"
    },
    "cert_pem": "MIID0TCCA...."

}```

### Certificate Scan Results without Cert PEM

```json
{
    "type": "cert",
    "timestamp": "2025-12-15T12:18:07-05:00",
    "urn": "urn:cert:gcp:projects/gcp-project-corp-dev-discovery/locations/global/
certificates/testcert",
    "url": "https://console.cloud.google.com/security/ccm/list/lbCertificates?
project=gcp-project-corp-dev-discovery",
    "extra": {
      "create_time": "2025-11-17T20:00:16.553991295Z",
      "description": "",
      "location": "global",
      "managed": {
        "authorization_attempt_info": [
          {
            "domain": "www.example.com",
            "state": "authorizing"
```
```

```
          }
        ],
        "dns_authorizations": null,
        "domains": [
          "www.example.com"
        ],
        "issuance_config": "",
        "provisioning_issue": {
          "reason": "authorization_issue"
        },
        "state": "provisioning"
      },
      "name": "projects/gcp-project-corp-dev-discovery/locations/global/certificates/
testcert",
      "origin": "gcp_created",
      "platform_type": "gcp",
      "project_id": "gcp-project-corp-dev-discovery",
      "subject_alternative_names": [
        "www.example.com"
      ],
      "update_time": "2025-11-18T02:31:08.415811203Z"
    },
    "cert_pem": ""
}
```


Output Fields

Field	Description	Present For
type	Always "cert" for certificate assets	All certificates
timestamp	RFC3339 timestamp of when the scan was performed	All certificates
urn	Unique resource identifier based on certificate fingerprint or resource name	All certificates
url	Direct link to GCP Console	All certificates
cert_pem	Base64-encoded certificate (PEM without headers/footers)	Self-managed only (empty for Google-managed)
extra	Additional metadata (see Extra Fields section below)	All certificates

Extra Fields

The `extra` object contains GCP-specific metadata about the certificate.

Common Fields (Present in All Certificates)

- `name` (string): Full GCP resource name (e.g., "projects/my-project/locations/global/certificates/my-cert")
- `origin` (string): Source of certificate (values: "imported", "gcp_created")
- `create_time` (string): ISO 8601 timestamp of when the certificate was created in GCP
- `update_time` (string): ISO 8601 timestamp of last update
- `description` (string): User-provided description of the certificate
- `project_id` (string): GCP project ID containing the certificate
- `region` (string): GCP region or location (empty string for global certificates)

Conditional Common Fields

- `labels` (object): Key-value pairs of user-defined labels (only present if labels exist)
- `scope` (string): Certificate scope (values: "default", "edge_cache", "all_regions") (only present if scope is set)

Fallback Fields (When cert_pem is Empty)

These fields are only included when the certificate PEM is not available (typically for Google-managed certificates in provisioning state):

- `not_after` (string): Certificate expiration date (ISO 8601 format)
- `subject_alternative_names` (array of strings): List of Subject Alternative Names (DNS names)

Note: These fields are NOT included when `cert_pem` is present, as they are intrinsic to the certificate and can be parsed from the PEM.

Self-Managed Certificate Fields

For imported/self-managed certificates (`origin: "imported"`), the following additional field is included:

- `self_managed` (object): Information about self-managed certificate
  - `has_pem_certificate` (boolean): Whether a PEM certificate is present
  - `has_pem_private_key` (boolean): Whether a PEM private key is present (not retrieved by plugin)

Google-Managed Certificate Fields

For Google-managed certificates (`origin: "gcp_created"`), the following additional field is included:

- `managed` (object): Information about Google-managed certificate provisioning
  - `domains` (array of strings): List of domains covered by the certificate
  - `dns_authorizations` (array of strings): DNS authorization resource names used for domain validation
  - `issuance_config` (string): Certificate issuance configuration resource name
  - `state` (string): Current provisioning state (values: `"active"` , `"provisioning"` , `"failed"`)
  - `authorization_attempt_info` (array of objects): Authorization attempt details for each domain

- `domain` (string): Domain being authorized (lowercase)
- `state` (string): Authorization state (lowercase, e.g., `"authorizing"`, `"authorized"`, `"failed"`)
- `details` (string): Additional details about authorization (lowercase)
- `failure_reason` (string): Reason for authorization failure if applicable (lowercase)

- `provisioning_issue` (object): Information about provisioning issues if any
  - `reason` (string): Reason for provisioning failure (lowercase, e.g., `"authorization_issue"`, `"rate_limited"`)
  - `details` (string): Additional details about the issue (lowercase)

Note: All string values in the `managed` object are converted to lowercase for consistency.

GCP Key Management Service plugin output

Output Format

The plugin outputs discovered keys in JSON format with detailed metadata:

Symmetric Key Example


```
{
    "type": "symkey",
    "timestamp": "2025-12-15T12:23:41-05:00",
    "urn": "urn:symkey:name:ouldriw-gcp-test-ekm-aes-3:1",
    "url": "https://console.cloud.google.com/security/kms/key/manage/us/gcp-byok-entrust-kc-key-ring/ouldriw-gcp-test-ekm-aes-3?project=htdc-project",
    "extra": {
      "account_id": "htdc-project",
      "created_date": "2025-10-02T14:01:56Z",
      "cryptographic_algorithm": "AES",
      "cryptographic_length": 256,
      "external_protection_level_options": {
        "external_key_uri": "https://xkc8.entrust.com/v0/ekm/422c5783-36d5-40cf-82a9-632ff382b8ae"
      },
      "generated_date": "2025-10-02T14:01:56Z",
      "hsm_backed": false,
      "key_ring": "projects/htdc-project/locations/us/keyRings/gcp-byok-entrust-kc-key-ring",
      "key_version": "1",
      "location": "us",
      "name": "ouldriw-gcp-test-ekm-aes-3",
      "origin": "gcp_kms",
      "platform_type": "gcp",
      "project_id": "htdc-project",
      "protection_level": "external",
      "purpose": "encrypt_decrypt",
      "reimport_eligible": false,
      "status": "enabled"
```

```
    }
}
```


Asymmetric Key Example


```
{
    "type": "pubkey",
    "timestamp": "2025-12-15T12:23:42-05:00",
    "urn":
"urn:pubkey:sha256:9b11b522b4888047012b525c54554767042f0184b9246672a6ab2b99dfadbaa8",
    "url": "https://console.cloud.google.com/security/kms/key/manage/us/gcp-byok-entrust-
kc-key-ring/ouldriw-gcp-test-ekm-ec?project=htdc-project",
    "extra": {
      "account_id": "htdc-project",
      "created_date": "2025-10-02T13:53:51Z",
      "external_protection_level_options": {
        "external_key_uri": "https://xkc8.entrust.com/v0/ekm/db685b6f-7cc0-4883-8634-
e2139c5a9bfb"
      },
      "generated_date": "2025-10-02T13:53:51Z",
      "hsm_backed": false,
      "key_ring": "projects/htdc-project/locations/us/keyRings/gcp-byok-entrust-kc-key-
ring",
      "key_version": "projects/htdc-project/locations/us/keyRings/gcp-byok-entrust-kc-
key-ring/cryptoKeys/ouldriw-gcp-test-ekm-ec/cryptoKeyVersions/1",
      "location": "us",
      "name": "ouldriw-gcp-test-ekm-ec",
      "origin": "gcp_kms",
      "platform_type": "gcp",
      "project_id": "htdc-project",
      "protection_level": "external",
      "purpose": "sign_verify",
      "reimport_eligible": false,
      "status": "enabled"
    },
    "pubkey_pem": "MFkwEwYHKoZI...."
}
```


Output Fields

Field	Description
`type`	Key type (`symkey` for symmetric/MAC keys, `pubkey` for asymmetric keys)
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name for the key (SHA256 hash of public key for asymmetric keys, name-based for symmetric keys)

Field	Description
`url`	Direct link to GCP Console for the key
`pubkey_pem`	Public key in RFC 7468 PEM format (base64 body without headers/footers) - asymmetric keys only
`extra`	Additional metadata (see Extra Fields section below)

Extra Fields

The `extra` object contains GCP-specific metadata about the key. The fields differ based on key type.

Common Fields (Present in All Keys)

- `platform_type` (string): Always `"gcp"` indicating the key is from GCP KMS
- `name` (string): Key name from GCP KMS
- `status` (string): Current key state (values: `"enabled"`, `"disabled"`, `"destroyed"`, `"destroy_scheduled"`, `"pending_generation"`, `"pending_import"`, `"import_failed"`)
- `origin` (string): Key origin (value: `"gcp_kms"`)
- `account_id` (string): GCP project ID
- `location` (string): GCP location/region (e.g., `"us"`, `"us-east1"`, `"global"`)
- `key_ring` (string): Full resource name of the key ring (e.g., `"projects/my-project/locations/us/keyRings/my-keyring"`)
- `protection_level` (string): Protection level (values: `"software"`, `"hsm"`, `"external"`, `"external_vpc"`)
- `purpose` (string): Key usage purpose
- `hsm_backed` (boolean): Whether the key is backed by a Hardware Security Module
- `created_date` (string): ISO 8601 timestamp of when the key version was created
- `reimport_eligible` (boolean): Whether the key can be reimported

Conditional Common Fields

The following fields are only present when the data is available:

- `generated_date` (string): ISO 8601 timestamp of key generation (only present if key was generated)
- `destroy_time` (string): ISO 8601 timestamp when key will be destroyed (only present if scheduled for destruction)
- `destroy_event_time` (string): ISO 8601 timestamp when destroy event occurred (only present if applicable)
- `import_job` (string): Import job resource name (only present for imported keys)
- `import_time` (string): ISO 8601 timestamp of key import (only present for imported keys)
- `import_failure_reason` (string): Reason for import failure (only present if import failed)
- `external_protection_level_options` (object): External key manager options (only present for external keys)

  - `external_key_uri` (string): URI of the external key
- `labels` (object): Key-value pairs of user-defined labels (only present if labels exist)

Asymmetric Key-Specific Fields (`type: "pubkey"`)

In addition to the common fields, asymmetric keys include:

- `key_version` (string): Full resource name of the specific key version (e.g., `"projects/.../cryptoKeys/.../cryptoKeyVersions/1"`)
- `project_id` (string): GCP project identifier
- `purpose` (string): Key purpose (values: `"sign_verify"` , `"encrypt_decrypt"`)

Note: Intrinsic cryptographic properties like `cryptographic_algorithm` , `cryptographic_length` , and `cryptographic_curve` are NOT included in extras as they can be derived from the `pubkey_pem` field.

Symmetric Key-Specific Fields (`type: "symkey"`)

In addition to the common fields, symmetric keys include:

- `project_id` (string): GCP project identifier from resource name
- `key_version` (string): Version number (e.g., `"1"` , `"2"`)
- `purpose` (string): Key purpose (values: `"encrypt_decrypt"` , `"mac"`)
- `cryptographic_algorithm` (string): Algorithm type (values: `"AES"` , `"HMAC"`) (only present if determinable)
- `cryptographic_length` (integer): Key length in bits (only present if determinable)

GCP Secret Manager plugin output

Output Format

Scan Results

Each discovered secret generates a result with the following structure:


```
{
    "type": "secret",
    "timestamp": "2025-12-15T12:20:46-05:00",
    "urn": "urn:secret:name:database-password:1",
    "url": "https://console.cloud.google.com/security/secret-manager/secret/database-password/versions?project=gcp-project-corp-dev-discovery",
    "extra": {
      "created_date": "2025-11-14T17:55:33.143792Z",
      "current_version": "1",
      "labels": {
        "environment": "test",
        "team": "backend"
      },
      "last_modified_date": "2025-11-14T17:55:34.492888Z",
      "location": "automatic",
      "name": "database-password",
      "platform_type": "gcp",
```

```
      "project_id": "gcp-project-corp-dev-discovery",
      "replication": {
        "automatic": {}
      },
      "replication_status": {
        "automatic": {}
      },
      "revision": "1",
      "rotation": {
        "enabled": false
      },
      "status": "enabled",
      "version_count": 1,
      "version_details": {
        "client_specified_payload_checksum": true,
        "create_time": "2025-11-14T17:55:34.492888Z",
        "etag": "\"164391b44c76d8\"",
        "name": "projects/415567878177/secrets/database-password/versions/1"
      }
    }
}
```


Output Fields

Field	Description
`type`	Always "secret" for secret assets
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name following format `urn:secret:name:{secret-name}:{version}`
`url`	Direct link to the secret in Google Cloud Console
`extra`	Additional metadata (see Extra Fields section below)

Extra Fields

The `extra` object contains GCP-specific metadata about the secret.

Standard Fields (Always Included)

- `platform_type` (string): Always `"gcp"` indicating the secret is from GCP Secret Manager
- `name` (string): Secret name/identifier
- `project_id` (string): GCP project ID containing the secret
- `created_date` (string): ISO 8601 timestamp of when the secret was created

- `last_modified_date` (string): ISO 8601 timestamp of when the current version was created
- `status` (string): Current secret version state (values: `"enabled"`, `"disabled"`, `"destroyed"`)
- `revision` (string): Current version number
- `current_version` (string): Current version number (same as revision)
- `version_count` (integer): Total number of versions for this secret
- `rotation` (object): Secret rotation configuration
  - `enabled` (boolean): Whether automatic rotation is enabled
  - `next_rotation_time` (string): ISO 8601 timestamp of next scheduled rotation (only present when rotation is enabled)
  - `rotation_period` (string): Duration between rotations in seconds format (e.g., `"2592000s"` for 30 days) (only present when rotation is enabled)

Conditional Fields (Included When Available)

- `location` (string): Replication location (value: `"automatic"` for automatic replication, or specific location for user-managed replicas)
- `expiration_date` (string): ISO 8601 timestamp when the secret expires (only present if expiration is set)
- `version_details` (object): Detailed information about the current version (only present if version information is available)
  - `name` (string): Full resource name of the version
  - `create_time` (string): ISO 8601 timestamp of version creation
  - `etag` (string): Version entity tag for concurrency control
  - `destroy_time` (string): Timestamp when version was destroyed (only present if destroyed)
  - `scheduled_destroy_time` (string): Timestamp when version is scheduled for destruction (only present if scheduled)
  - `client_specified_payload_checksum` (boolean): Whether client specified a payload checksum (only present if true)
- `labels` (object): Key-value pairs of user-defined labels (only present if labels exist)
- `annotations` (object): Key-value pairs of annotations (only present if annotations exist)
- `ttl` (string): Time-to-live duration for the secret (only present if TTL is set)
- `version_destroy_ttl` (string): Duration before version destruction (only present if set)
- `topics` (array of objects): Pub/Sub topics for notifications (only present if topics are configured)
- `replication` (object): Replication configuration details (only present if replication is configured)
  - `automatic` (object): Automatic replication settings (present for automatic replication)
  - `user_managed` (object): User-managed replication settings (present for user-managed replication)
    - `replicas` (array): List of replica locations
- `replication_status` (object): Current replication status (only present if replication status is available)
  - `automatic` (object): Status for automatic replication
  - `user_managed` (object): Status for user-managed replication

Note: The plugin extracts only metadata about secrets, not the actual secret values themselves. All timestamp fields use ISO 8601 format.

HashiCorp Vault plugin output

Output Format

Public Key Scan Results


```
{
    "type": "pubkey",
    "timestamp": "2025-12-15T12:29:20+05:30",
    "urn": "urn:pubkey:name:document-signing:1",
    "url": "https://10.1.127.31:8200/ui/vault/secrets/transit/show/document-signing",
    "extra": {
      "allow_plaintext_backup": false,
      "cryptographic_algorithm": "Ed25519",
      "cryptographic_length": 256,
      "deletion_allowed": false,
      "derived": false,
      "engine_path": "transit",
      "engine_type": "transit",
      "exportable": false,
      "key_id": "transit/document-signing",
      "key_operations": [
        "sign",
        "verify"
      ],
      "key_version": "1",
      "latest_version": 1,
      "name": "document-signing",
      "purpose": "signing/verification",
      "status": "enabled",
      "vault_url": "https://10.1.127.31:8200"
    },
    "pubkey_pem": "a1XEMzVXj0TX...
}
```


Symmetric Key Scan Results


```
{
    "type": "symkey",
    "timestamp": "2025-12-15T12:29:19+05:30",
    "urn": "urn:symkey:name:backup-restore:1",
    "url": "https://10.1.127.31:8200/ui/vault/secrets/transit/show/backup-restore",
    "extra": {
      "allow_plaintext_backup": false,
      "cryptographic_algorithm": "AES-GCM",
      "cryptographic_length": 256,
      "deletion_allowed": false,
      "derived": true,
      "engine_path": "transit",
      "engine_type": "transit",
      "exportable": true,
      "key_id": "transit/backup-restore",
      "key_operations": [
```

```
        "encrypt",
        "decrypt"
      ],
      "key_version": "1",
      "latest_version": 1,
      "name": "backup-restore",
      "purpose": "encryption/decryption",
      "status": "enabled",
      "vault_url": "https://10.1.127.31:8200"
    }
}
```


Certificates Scan Results


```
{
    "type": "cert",
    "timestamp": "2025-12-15T12:29:16+05:30",
    "urn":
"urn:cert:sha256:2c9ee1f798c8e6a29a6ea445d1bed05a0e310e4109ae81424b609e3c03be4373",
    "url": "https://10.1.127.31:8200/ui/vault/secrets/pki/show/
01:93:c1:e6:42:0c:6c:2d:d4:c8:8a:f5:1e:06:6c:12:92:4a:05:d0",
    "extra": {
      "engine_path": "pki",
      "engine_type": "pki",
      "status": "valid",
      "vault_serial": "01:93:c1:e6:42:0c:6c:2d:d4:c8:8a:f5:1e:06:6c:12:92:4a:05:d0",
      "vault_url": "https://10.1.127.31:8200"
    },
    "cert_pem": "MIIDHzCCA...."
}
```


Secrets Scan Results


```
{
    "type": "secret",
    "timestamp": "2025-12-15T12:29:32+05:30",
    "urn": "urn:secret:name:database/dev:1",
    "url": "https://10.1.127.31:8200/ui/vault/secrets/secret/show/database/dev",
    "extra": {
      "created_date": "2025-11-28T10:26:31.281551177Z",
      "current_version": "1",
      "engine_path": "secret",
      "engine_type": "kv",
      "has_cert_data": false,
      "has_key_data": false,
      "key_count": 4,
      "keys": [
        "host",
        "password",
        "port",
        "username"
      ],
      "last_modified_date": "2025-11-28T10:26:31.281551177Z",
      "name": "database/dev",
```

```
      "vault_url": "https://10.1.127.31:8200"
    }
}
```

Output Fields

Field	Description
`type`	Asset type (`cert` for certificates, `pubkey` for asymmetric keys, `symkey` for symmetric keys, `secret` for KV secrets)
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name based on asset type (SHA256 hash for certificates/public keys, name-based for symmetric keys/secrets)
`url`	Direct link to Vault UI for the asset
`cert_pem`	Base64-encoded certificate (PEM without headers/footers) - certificates only
`pubkey_pem`	Base64-encoded public key (PEM without headers/footers) - asymmetric keys only
`extra`	Additional metadata (see Extra Fields section below)

Extra Fields

The `extra` object contains HashiCorp Vault-specific metadata about the asset. The fields differ based on asset type.

Common Fields (Present in Most Assets)

- `platform_type` (string): Always `"hashicorp_vault"` indicating the asset is from HashiCorp Vault
- `name` (string): Asset name/identifier from Vault
- `vault_url` (string): HashiCorp Vault server URL
- `engine_path` (string): Path of the secrets engine where the asset is stored
- `engine_type` (string): Type of secrets engine (values: `"pki"` , `"transit"` , `"kv"` , `"kv-v2"` , `"ssh"`)
- `status` (string): Current asset state (values: `"enabled"` , `"valid"` , `"expired"` , `"not_yet_valid"`)

Certificate-Specific Fields (`type: "cert"`)

For PKI certificates, the following fields are included:

- `vault_serial` (string): Certificate serial number as stored in Vault
- `namespace` (string): Vault namespace (only present for Vault Enterprise)

Note: Intrinsic certificate properties (issuer, subject, validity dates, key usage) are NOT included in extras as they can be parsed from the `cert_pem` field.

Public Key-Specific Fields (`type: "pubkey"`)

For asymmetric keys from Transit engine, the following fields are included:

- `key_id` (string): Full key identifier including engine path (e.g., `"transit/api-signing"`)
- `purpose` (string): Key usage purpose (value: `"signing/verification"`)
- `exportable` (boolean): Whether the key can be exported from Vault
- `key_operations` (array of strings): Supported cryptographic operations (e.g., `["sign", "verify"]`)
- `created_date` (string): ISO 8601 timestamp of when the key was created (only present if available)
- `cryptographic_algorithm` (string): Algorithm type (e.g., `"RSA"` , `"ECDSA"` , `"Ed25519"`)
- `cryptographic_length` (integer): Key length in bits (e.g., 2048, 4096)
- `key_version` (string): Current key version number
- `latest_version` (integer): Latest version number available for this key
- `deletion_allowed` (boolean): Whether the key can be deleted
- `derived` (boolean): Whether the key uses key derivation
- `allow_plaintext_backup` (boolean): Whether plaintext backup is allowed
- `namespace` (string): Vault namespace (only present for Vault Enterprise)

Symmetric Key-Specific Fields (`type: "symkey"`)

For symmetric keys from Transit engine, the following fields are included:

- `key_id` (string): Full key identifier including engine path (e.g., `"transit/backup-encryption"`)
- `purpose` (string): Key usage purpose (values: `"encryption/decryption"` , `"MAC generation/verification"`)
- `key_operations` (array of strings): Supported cryptographic operations (e.g., `["encrypt", "decrypt"]`)
- `created_date` (string): ISO 8601 timestamp of when the key was created (only present if available)
- `key_version` (string): Current key version number
- `cryptographic_algorithm` (string): Algorithm type (e.g., `"AES-GCM"` , `"ChaCha20-Poly1305"` , `"HMAC"`)
- `cryptographic_length` (integer): Key length in bits (e.g., 256, 512)
- `latest_version` (integer): Latest version number available for this key
- `exportable` (boolean): Whether the key can be exported from Vault
- `deletion_allowed` (boolean): Whether the key can be deleted
- `derived` (boolean): Whether the key uses key derivation
- `allow_plaintext_backup` (boolean): Whether plaintext backup is allowed
- `namespace` (string): Vault namespace (only present for Vault Enterprise)

Secret-Specific Fields (`type: "secret"`)

For secrets from KV engines, the following fields are included:

- `created_date` (string): ISO 8601 timestamp of when the secret was created
- `last_modified_date` (string): ISO 8601 timestamp of last modification (only present if available)
- `has_cert_data` (boolean): Whether the secret contains certificate data
- `has_key_data` (boolean): Whether the secret contains key data
- `key_count` (integer): Number of key-value pairs in the secret
- `keys` (array of strings): List of keys in the secret (only present if keys exist)
- `current_version` (string): Current version number (only present for KV v2 secrets)
- `namespace` (string): Vault namespace (only present for Vault Enterprise)

Note: The plugin analyzes KV secrets to determine if they contain cryptographic material but does not extract the actual secret values.

Supported Secrets Engines

The plugin can discover assets from the following HashiCorp Vault secrets engines:

- **PKI**: Public Key Infrastructure certificates
- **Transit**: Encryption-as-a-service keys (symmetric and asymmetric)
- **KV v1**: Key-Value secrets (version 1)
- **KV v2**: Versioned Key-Value secrets (version 2)

Note: If no specific secrets engines are configured, the plugin will automatically discover and scan all available engines in the Vault instance.

Network Scanner - TLS plugin output

Output Format

Certificate Results

The plugin generates certificate scan results in standardized format:


```
{
   "type": "cert",
    "timestamp": "2025-12-10T22:04:02+05:30",
    "urn":
"urn:cert:sha256:d4db1d3029b6d4bfb66b3cbca043c83d90bb6caae35a02b3d8ee62b64a8ce2a6",
    "url": "https://10.1.127.26:443",
    "cert_pem": "MIIFODCCBCC....",
    "endpoint": {
      "host": "10.1.127.26",
      "port": 443,
      "service": "http",
      "service_vendor": "nginx",
      "service_vendor_version": "1.18.0",
      "cipher_suites": "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, ...."
```

```
    }
}
```

Output Fields

Field	Description
`type`	Asset type (`cert` for certificates, `pubkey` for asymmetric keys, `symkey` for symmetric keys, `secret` for KV secrets)
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name based on asset type (SHA256 hash for certificates/public keys, name-based for symmetric keys/secrets)
`url`	Direct link to Vault UI for the asset
`cert_pem`	Base64-encoded certificate (PEM without headers/footers) - certificates only
`endpoint`	object

Endpoint Fields

Field	Type	Description
`host`	string	Host address or hostname where the certificate was discovered
`port`	integer	Port number where the certificate was discovered (1-65535)
`service`	string	Detected service type (e.g., `http` , `https` , `smtp` , `imap` , `pop3` , `ldap` , `ftp` , `ssh`)
`service_vendor`	string	Detected service vendor (e.g., `nginx` , `Apache` , `OpenSSH` , `Postfix` , `IIS`)
`service_vendor_version`	string	Detected service vendor version (e.g., `1.18.0` , `2.4.41`)

Field	Type	Description
cipher_suites	string	Comma-separated list of supported SSL/TLS cipher suites detected by ssl-enum-ciphers script

Note: The service , service_vendor , service_vendor_version , and cipher_suites fields are optional and populated when nmap successfully detects the service information.

Network Scanner - OIDC plugin output

Output Format


```
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:5e7e618024db381bdd758a00f5a861542d00face4b7008669be7ee1587080eb0",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "issuer_url": "https://login.microsoftonline.com/common",
      "jwk_algorithm": "",
      "jwk_key_id": "yEUwmXWL107Cc-7QZ2WSbeOb3sQ",
      "jwk_use": "sig",
      "jwk_x5t": "yEUwmXWL107Cc-7QZ2WSbeOb3sQ",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "platform_type": "oidc",
      "total_certs": 1
    },
    "cert_pem": "MIIC/jCCAeagAwIBAgIJAM/
j60cyShQEMA0GCSqGSIb3DQEBCwUAMC0xKzApBgNVBAMTImFjY291bnRzLmFjY2Vzc2NvbnRyb2wud2lu
ZG93cy5uZXQwHhcNMjUxMDAxMDUxNjM2WhcNMzAxMDAxMDUxNjM2WjAtMSswKQYDVQQDEyJhY2Nv
dW50cy5hY2Nlc3Njb250cm9sLndpbmRvd3MubmV0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCA
QEAsFO9W05FQwiZenzhV4wYRMpok4hgSMR+TZKRZKt26FojHgHfOhySAjcCesHo/
QVvj0Wos9KSgIB3Pt55qo2lc4TmRLdpFsKZ5XEyWaCZL87pGb6fqsCNIqw+768vhGUk3nR2NHOjNrzwJb/
soyZO0/2Wq7cZRqD8yFeVWRFxY8qDm1e/
qutcxSQm+d0rvgGP3ZSbcx5HpTjQIz+HmiRDJoIfhRBgFRriBNmF7mG8H9Tug2GsZObGl4PbNyMYzVfJvqU
GVvxdpX0TOgZ9hnva5QSfPaJCpyya1diUX6rJAxXU5JU7/
yoWXW49nHAd1LUtxnSrVgD8tY84HNcyAmNZ6wIDAQABoyEwHzAdBgNVHQ4EFgQUYdczXud2h1dPEOc
A7tnbvbh/+sAwDQYJKoZIhvcNAQELBQADggEBABpmhrnzarikYdrPOr72YQhi2Hw/
GMswnvVyrf9ORSwynpHR1jKki+rr/rtEUTZbArRsvy0U/0PwnmDHM/
J1R7o9mi4sbf56o0CuZOifk362i2THIiS9s7QCuOJneGrZo6YsIvcTubo2C13taowzhvar/
cJgMTeWcvqw2UAnOr8uXkM5NWbUQGj2r2aAg0U71dJrSQGukkujzxj2wdQshg9BKC3QkIH8mvaQAVkSYT
BDI2whF1JcYRmNzIsZGGG3kWUncKzIw6zmPQRYK96E/n4Tq6yYybq5CsfrvP9/9iDg+hhOt5gzbuQSX/
0yYA6vbk4y8ZaOr51OCvfEj1ew2Bg="
```

```
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:bc0ff35a25a9217c0cfe5f7ee25f808dacaa7aa332f4690adbbdee8097483033",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "jwk_algorithm": "",
      "jwk_key_id": "yEUwmXWL107Cc-7QZ2WSbeOb3sQ",
      "jwk_use": "sig",
      "jwk_x5t": "yEUwmXWL107Cc-7QZ2WSbeOb3sQ",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "platform_type": "oidc",
      "purpose": "sign_verify"
    },
    "pubkey_pem":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsFO9W05FQwiZenzhV4wYRMpok4hgSMR+TZK
RZKt26FojHgHfOhySAjcCesHo/
QVvj0Wos9KSgIB3Pt55qo2lc4TmRLdpFsKZ5XEyWaCZL87pGb6fqsCNIqw+768vhGUk3nR2NHOjNrzwJb/
soyZO0/2Wq7cZRqD8yFeVWRFxY8qDm1e/
qutcxSQm+d0rvgGP3ZSbcx5HpTjQIz+HmiRDJoIfhRBgFRriBNmF7mG8H9Tug2GsZObGl4PbNyMYzVfJvqU
GVvxdpX0TOgZ9hnva5QSfPaJCpyya1diUX6rJAxXU5JU7/
yoWXW49nHAd1LUtxnSrVgD8tY84HNcyAmNZ6wIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:f2bd70b8aaa376a859d8026703b482b3043466cd5d24865e1070b8f6776b722a",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "issuer_url": "https://login.microsoftonline.com/common",
      "jwk_algorithm": "",
      "jwk_key_id": "-MyGFduIUViaL6NDYyTV0FGATGk",
      "jwk_use": "sig",
      "jwk_x5t": "-MyGFduIUViaL6NDYyTV0FGATGk",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "platform_type": "oidc",
      "total_certs": 1
```

```
    },
    "cert_pem": "MIIC/
jCCAeagAwIBAgIJAJcPVeVovVVRMA0GCSqGSIb3DQEBCwUAMC0xKzApBgNVBAMTImFjY291bnRzLmFjY
2Vzc2NvbnRyb2wud2luZG93cy5uZXQwHhcNMjUxMDE1MTkzMTM1WhcNMzAxMDE1MTkzMTM1WjAtMSs
wKQYDVQQDEyJhY2NvdW50cy5hY2Nlc3Njb250cm9sLndpbmRvd3MubmV0MIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAp5cmEcNwRuuH+AloG+bA4At8VAkpIuMlu7Zl0AhFewUUfOEf0tOKNGH7LGc
6pzjYGwZuey94xelj4SRLHMyeTopjjHtmyy1LJyXZLUBT6tVTIwGWECpHQKNcIjXbSgRR80C6lrAcBSErXV43
h9awnu1JrhlfVrCaDYuE2oBFO5PZR6H0ZyfehCIRYt/
69MCxUMmbx+nyj+bnlJXmRuF87ll6hmceEYXTUnpuWNeIHp6UMSTKE2UCna6KSsvOJ8WPXQJq2SImefNy
F67hSSKgdPJcnid+E4hb63mrkfxyFm+bLMtd4twEKZv9F5q9Wk3NrHZuMIqjjcSdyt/
aEcFBTQIDAQABoyEwHzAdBgNVHQ4EFgQUjulxdRp7rNoGxO9r/
r4vh+sOVOUwDQYJKoZIhvcNAQELBQADggEBAG0FMXXZzggEePMs1cxHLViL9/fP5QF50P2qm1snTsN8/
SfNcpUPOy+cH+DP58omGXz4GfC8J28Yq/Ct8aSh7AX6Sn88QdEzojaKB/
maCcfINH9YUxaYXW0aVawr2Qz7DH1gP453shqduay+d9vlglNNtqqeK9Xfyi3npHIJloLtD5B8/
YPlOQ69dUTNVtNeZVlw/GyM+aca/V+WHr8YocxybAM/
8ZPjS8Z7i587y6HN9ojDiVPZ3R3kgDM2C1o0mBVh2ZkX3zJVUKcehJslGhWtFJH39/1QOIQGpPtLMy+fdWX
hsoz8OI4Kc9tIlnrXBlwSdTlrIK2owYyMbzX+EQU="
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:a989f058e2fa368d25bd4f6be447c5a4179a6a3e3b26dcd52fe709ceec4945ff",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "jwk_algorithm": "",
      "jwk_key_id": "-MyGFduIUViaL6NDYyTV0FGATGk",
      "jwk_use": "sig",
      "jwk_x5t": "-MyGFduIUViaL6NDYyTV0FGATGk",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "platform_type": "oidc",
      "purpose": "sign_verify"
    },
    "pubkey_pem":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAp5cmEcNwRuuH+AloG+bA4At8VAkpIuMlu7Zl0
AhFewUUfOEf0tOKNGH7LGc6pzjYGwZuey94xelj4SRLHMyeTopjjHtmyy1LJyXZLUBT6tVTIwGWECpHQKN
cIjXbSgRR80C6lrAcBSErXV43h9awnu1JrhlfVrCaDYuE2oBFO5PZR6H0ZyfehCIRYt/
69MCxUMmbx+nyj+bnlJXmRuF87ll6hmceEYXTUnpuWNeIHp6UMSTKE2UCna6KSsvOJ8WPXQJq2SImefNy
F67hSSKgdPJcnid+E4hb63mrkfxyFm+bLMtd4twEKZv9F5q9Wk3NrHZuMIqjjcSdyt/aEcFBTQIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "cert",
```

```
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:2fb708c7a140214aa085ff2696d90dbccdcf7edd7ea280a3a1a1b94413b91457",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "issuer_url": "https://login.microsoftonline.com/common",
      "jwk_algorithm": "",
      "jwk_key_id": "rtsFT-b-7LuY7DVYeSNKclJ7Vnc",
      "jwk_use": "sig",
      "jwk_x5t": "rtsFT-b-7LuY7DVYeSNKclJ7Vnc",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "total_certs": 1
    },
    "cert_pem": "MIIC/
jCCAeagAwIBAgIJAPJGEpowIhBNMA0GCSqGSIb3DQEBCwUAMC0xKzApBgNVBAMTImFjY291bnRzLmFjY
2Vzc2NvbnRyb2wud2luZG93cy5uZXQwHhcNMjUxMDI2MTk0NjQ1WhcNMzAxMDI2MTk0NjQ1WjAtMSswK
QYDVQQDEyJhY2NvdW50cy5hY2Nlc3Njb250cm9sLndpbmRvd3MubmV0MIIBIjANBgkqhkiG9w0BAQEFAA
OCAQ8AMIIBCgKCAQEAsE+vzm1BhzJJ5KKgJKPGX4M3GbeM0c25HOVQL1aLbOEHm92HBFk1djM9a8WL
Dfg/
d8SLh3Ehta0i0ctATwU0CSeeodvsqL4mKEOXYEqIi1f8ixCX0c7vJ0ESNcyWeAm18F9WNtFKKDOM7gzCn0z
uuAZR3m/rBaPDOkoX1AULrkMZjnantrw4z8hL344dLAneta5JiulJor2NiJGNU5EHcVjw7eMDunPTpC/
lAxDKF5/hTQ0Hj4+R2AzuSBO0DZ3T2G7/6lmIguOIanfGoYGKev4JvumXahkVGf/
tgZ3WuoUqB8KEIM8VGjS0MjBFgCtxX6GmvRD+H3F58x4bsBAZxwIDAQABoyEwHzAdBgNVHQ4EFgQU+
A6C3/
xdVe7vu2wezFXPLQE0nyMwDQYJKoZIhvcNAQELBQADggEBADAoTCjqbO+Ku6E1nbOUkq513ETV+7iL6g
7FnxY4ysl2qPAsgPcLOO/
HoWGLNfu4fbqyBqtSpoHYQUEe2e4FNF9T0EB5B5NShFiSlLVcQyp23PcrcInQRnb7x9iX/
ztxm1bpNnLXrQrh/RTsdev6LqiIfhC2XH70Avb6LTYcBMkUuo9Y2kxT3WtyklSl0Ogr3td/
lPZne1vcPP4h64uzE9+GKcm+2iZRyWGMjtG6DnC1whmoetqDDmQ9pmHi2xlxSjcTS8oq/
FwEA20sjNO4DdBN9tS2VMwVZldZ/Z594sRKOPo3kPVdKhJZud5Yt2nt+xiHcjKY48HmOXRnF8AOto="
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:66dd1457a9c3a6c5fa059ae4e35b0eb8899e11556fa96c99b28e4d376eec444d",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "jwk_algorithm": "",
      "jwk_key_id": "rtsFT-b-7LuY7DVYeSNKclJ7Vnc",
      "jwk_use": "sig",
      "jwk_x5t": "rtsFT-b-7LuY7DVYeSNKclJ7Vnc",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "purpose": "sign_verify"
    },
    "pubkey_pem":
```

"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsE+vzm1BhzJJ5KKgJKPGX4M3GbeM0c25HOV
QL1aLbOEHm92HBFk1djM9a8WLDfg/
d8SLh3Ehta0i0ctATwU0CSeeodvsqL4mKEOXYEqIi1f8ixCX0c7vJ0ESNcyWeAm18F9WNtFKKDOM7gzCn0z
uuAZR3m/rBaPDOkoX1AULrkMZjnantrw4z8hL344dLAneta5JiulJor2NiJGNU5EHcVjw7eMDunPTpC/
IAxDKF5/hTQ0Hj4+R2AzuSBO0DZ3T2G7/6ImIguOIanfGoYGKev4JvumXahkVGf/
tgZ3WuoUqB8KEIM8VGjS0MjBFgCtxX6GmvRD+H3F58x4bsBAZxwIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:c4be8f65fcc69cb67c1ad4424c1060af912cd4a91f0513ad17df61d237f6cbf4",
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "issuer_url": "https://login.microsoftonline.com/common",
      "jwk_algorithm": "",
      "jwk_key_id": "QhLMpTTJogmIIa5vrgJiSFgh97I",
      "jwk_use": "sig",
      "jwk_x5t": "QhLMpTTJogmIIa5vrgJiSFgh97I",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "total_certs": 1
    },
    "cert_pem":
"MIIC6jCCAdKgAwIBAgIJAL1iBsaLYB8iMA0GCSqGSIb3DQEBCwUAMCMxITAfBgNVBAMTGGxvZ2luLm1pY
3Jvc29mdG9ubGluZS51czAeFw0yNTExMDkxNzAwNDFaFw0zMDExMDkxNzAwNDFaMCMxITAfBgNVBAMT
GGxvZ2luLm1pY3Jvc29mdG9ubGluZS51czCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIx2
U7dO5uf0n7CwD8BTs2Od214jGUYSpzGmPjEBfUuidBGWzDU8jE4OWuE5vwbDirfcxscW5+WryOPBSHyKh
9MKN9xasqxQqZxz8WfALTILGdebz3nl/2stDw2RBj15EFdFHN9OB0VybT2dYRonelxIE44Gseh0650ZV/
kd+Z89BiYufHtlTECDal9UqM9k/2yS00nO/
ZyPYoczfBGG+yy6BleAL8H3zt2VzkaBUrOOvGX0EcCdrrH6BFLej7SBBk3SfLkufRHQFaUlADSNbA1ZyZ6JyD
Vl8q9dEApWcDapueCAYf7ZPfGYHlbo9bQqCKAqTlMvFimourVpFry9xc8CAwEAAaMhMB8wHQYDVR0OBB
YEFBiWesvUh7Ny+crgGVrEmwQc+X/zMA0GCSqGSIb3DQEBCwUAA4IBAQA3N+vcAcpceN7s/
uOORi+5eB9FtL03ZVUmJtoUMMme6Q6588oIaTg4HhEM8XSzQJyljDJtsjJnw1YoZWCVi5ju0c3f+L+ue8Ap
hB+9VkjqSvwlR9gOrNhhO3ZFG8tvvlf7dZTbI5EaiMdHrpPCtRLruWUugbVZwqjM3foMixnOgxG1mMPNYNnS
URxo1ygS0WbS4dSzjXVX3GyuXIMjk+75CWtOan8JgY7KJCFqgXHgAb/
QpWjY7a5H0blu79rD6RFawH+zH5V6//9CgeiKPSmd4mfkS5IrVJ0jceek5ttqmYrRTYSQknl/
mVMP4pRX6rKLwwTahNWVHCvcaNlgXECX"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:59f5e6dc48d57a1d4287694c56126b555b442e9bf0e8f0cd2e68438f1a8cd978",

```
    "url": "https://login.microsoftonline.com/common",
    "extra": {
      "oidc_issuer": "https://sts.windows.net/{tenantid}/",
      "jwk_algorithm": "",
      "jwk_key_id": "QhLMpTTJogmIIa5vrgJiSFgh97I",
      "jwk_use": "sig",
      "jwk_x5t": "QhLMpTTJogmIIa5vrgJiSFgh97I",
      "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
      "origin": "oidc_jwks",
      "purpose": "sign_verify"
    },
    "pubkey_pem": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAjHZTt07m5/
SfsLAPwFOzY53bXiMZRhKnMaY+MQF9S6J0EZbMNTyMTg5a4Tm/
BsOKt9zGxxbn5avI48FIfIqH0wo33FqyrFCpnHPxZ8AtOUsZ15vPeeX/
ay0PDZEGPXkQV0Uc304HRXJtPZ1hGid4jEgTjgax6HTrnRlX+R35nz0GJi58e2VMQINqX1Soz2T/
bJLTSc79nl9ihzN8EYb7LLoGV4AvwffO3ZXORoFSs468ZfQRwJ2usfoEUt6PtIEGTdJ8uS59EdAVpSUANI1sD
VnJnonINWXyr10QClZwNqm54IBh/tk98ZgeVuj1tCoIoCpOUy8WKai6tWkWvL3FzwIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:a8f2498a81ee69b679074f3ae9308076f3f8efadba7e7855d0a981d134277653",
    "url": "https://token.actions.githubusercontent.com",
    "extra": {
      "oidc_issuer": "https://token.actions.githubusercontent.com",
      "jwk_algorithm": "RS256",
      "jwk_key_id": "cc413527-173f-5a05-976e-9c52b1d7b431",
      "jwk_use": "sig",
      "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
      "origin": "oidc_jwks",
      "purpose": "sign_verify"
    },
    "pubkey_pem":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAw4M936N3ZxNaEblcUoBm+xu0+V9JxNx5S7Tm
F0M3SBK+2bmDyAeDdeIOTcIVZHG+ZX9N9W0u1yWafgWewHrsz66BkxXq3bscvQUTAw7W3s6TEeYY7o9s
hPkFfOiU3x/
KYgOo06SpiFdymwJfIRs9cnbaU88i5fZJmUepUHVIlP2tpPWTi+7UA3AdP3cdcCs5bnFfTRKzH2W0xqKsY/
jIG95aQJRBDpbiesefjuyxcQnOv88j9tCKWzHpJzRKYjAUM6OPgN4HYnaSWrPJj1v41eEkFM1kORuj+GSH2q
MVD02VklcqaerhQHIqM+RjeHsN7G05YtwYzomE5G+fZuwgvQIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
```

```
  "urn": "urn:cert:sha256:5080fcb18b6cdce2069478d67d8ffaf2d67a772e190c8c99c44f9a9dc3641999",
  "url": "https://token.actions.githubusercontent.com",
  "extra": {
   "cert_index": 0,
   "oidc_issuer": "https://token.actions.githubusercontent.com",
   "issuer_url": "https://token.actions.githubusercontent.com",
   "jwk_algorithm": "RS256",
   "jwk_key_id": "38826b17-6a30-5f9b-b169-8beb8202f723",
   "jwk_use": "sig",
   "jwk_x5t": "ykNaY4qM_ta4k2TgZOCEYLkcYIA",
   "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
   "total_certs": 1
  },
  "cert_pem": "MIIDKzCCAhOgAwIBAgIUDnwm6eRIqGFA3o/P1oBrChvx/
nowDQYJKoZIhvcNAQELBQAwJTEjMCEGA1UEAwwaYWN0aW9ucy5zZWxmLXNpZ25lZC5naXRodWIwHh
cNMjQwMTIzMTUyNTM2WhcNMzQwMTIwMTUyNTM2WjAlMSMwIQYDVQQDDBphY3Rpb25zLnNlbGYtc2l
nbmVkLmdpdGh1YjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOTGp5svs8LJN8BH7VzXS
hWXnOK0lhDVuI0xnr5bwHFPc924CwaIEFb6mC7bvW2lZtgd633uaJ2naG6vKaOVGpCdGLE4ohH11nUk+2C
NknZL7/
oTmDHGSmGeHRb7kjtb0Ng4BJMPzmTYmCNUudfDFhHDcZz1Obuu85GsABrC5ZlzWzspYFXwUSaxvII+rH
K/
rAbOC2gmt5lOSLmgh3taQfp0mB6Lxlf89HoBPNwtPfBX8DtXTWQVnqODm4W+WfmWBSyXGX54DGNMyZ
wlTZqR0FjoMXxopId3MIuDGKxa2weDU5cW60N2y/
qxikeV99fL3sg5aPA8s9iljKG0+MAfVNUCAwEAAaNTMFEwHQYDVR0OBBYEFIPALo5VanJ6E1B9eLQgGO+u
GV65MB8GA1UdIwQYMBaAFIPALo5VanJ6E1B9eLQgGO+uGV65MA8GA1UdEwEB/
wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAGS0hZE+DqKIRi49Z2KDOMOaSZnAYgqq6ws9HJHT09
MXWlMHB8E/
apvy2ZuFrcSu14ZLweJid+PrrooXEXEO6azEakzCjeUb9G1QwlzP4CkTcMGCw1Snh3jWZIuKaw21f7mp2rQ+Y
NltgHVDKY2s8AD273E8musEsWxJl80/MNvMie8Hfh4n4/
Xl2r6t1YPmUJMoXAXdTBb0hkPy1fUu3r2T+1oi7Rw6kuVDfAZjaHupNHzJeDOg2KxUoK/GF2/M2qpVrd19Pv/
JXNkQXRE4DFbErMmA7tXpp1tkXJRPhFui/Pv5H9cPgObEf9x6W4KnCXzT3ReeeRDKF8SqGTPELsc="
 }
}
{
 "result_type": "scan",
 "plugin_id": "oidc-plugin",
 "plugin_version": "1.0.dev",
 "data": {
  "type": "pubkey",
  "timestamp": "2025-12-01T08:31:22-05:00",
  "urn": "urn:pubkey:sha256:92b790cd3edf64a3b1d369fe7e777a01753b0feb323766a3fbcd1cabf3b6f3be",
  "url": "https://token.actions.githubusercontent.com",
  "extra": {
   "oidc_issuer": "https://token.actions.githubusercontent.com",
   "jwk_algorithm": "RS256",
   "jwk_key_id": "38826b17-6a30-5f9b-b169-8beb8202f723",
   "jwk_use": "sig",
   "jwk_x5t": "ykNaY4qM_ta4k2TgZOCEYLkcYIA",
   "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
   "origin": "oidc_jwks",
   "purpose": "sign_verify"
  },
  "pubkey_pem":
```

"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5Manmy+zwsk3wEftXNdKFZec4rSWENW4jTGe
vlvAcU9z3bgLBogQVvqYLtu9baVm2B3rfe5onadobq8po5UakJ0YsTiiEfXWdST7YI2Sdkvv+hOYMcZKYZ4dFv
uSO1vQ2DgEkw/
OZNiYI1S518MWEcNxnPU5u67zkawAGsLlmXNbOylgVfBRJrG8gj6scr+sBs4LaCa3kg5luaCHe1pB+nSYHovG
V/
z0egE83C098FfwO1dNZBWeo4Obhb5Z+ZYFLJcZfngMY0zJnCVNmpHQWOgxfGikh3cwi4MYrFrbB4NTlxb
rQ3bL+rGKR5X318veyDlo8Dyz2KWMobT4wB9U1QIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:16be8ed6932596b33c3c8d03b29480edfdeac2fb023013d42f2063fbf24a0252",
    "url": "https://token.actions.githubusercontent.com",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://token.actions.githubusercontent.com",
      "issuer_url": "https://token.actions.githubusercontent.com",
      "jwk_algorithm": "RS256",
      "jwk_key_id": "38E9B30B3A023A1B72309921A69A42FCC496C42C",
      "jwk_use": "sig",
      "jwk_x5t": "OOmzCzoCOhtyMJkhpppC_MSWxCw",
      "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
      "total_certs": 1
    },
    "cert_pem": "MIIDrDCCApSgAwIBAgIQbuIOJTcGQ4GOQs29F1/
uLzANBgkqhkiG9w0BAQsFADA2MTQwMgYDVQQDEyt2c3RzLXZzdHNnaHJ0LWdoLXZzby1vYXV0aC52aX
N1YWxzdHVkaW8uY29tMB4XDTI1MDkxMTE5MzcxOFoXDTI3MDkxMTE5NDcxOFowNjE0MDIGA1UEAxMrdn
N0cy12c3RzZ2hydC1naC12c28tb2F1dGgudmlzdWFsc3R1ZGlvLmNvbTCCASIwDQYJKoZIhvcNAQEBBQAD
ggEPADCCAQoCggEBALRKthafR3HU+TMDLAf1E5vl/
8KCSS4t7APstEV9rYqE5ufNbHsn7zh+pSeSZh1bUvv32/AdrUU9ac5x0EjateD7CXbvW6eXoG1piM/
uUb4aqDPqCnpvzCRDQjRbql4o0S0+OxUXXx94p6nXtbyZQPySRwqmDCiKRx+5KjRWHAqcK9L5PNB3Nq08
i9LgVyOzagWW7VIDaMrnFjuTu0EKzzTppa/hiPGhXJBUX6Gwmgjoh/
altNhbZFe+H8ARlQjYTX92tQivNfjvMRBopr1cnSP6UAOQfEKlbkl2P57TrueeR3YqwNISpGOn7BPXmQjBZlgx2
dpEli44Wm8VsSM0s10CAwEAAaOBtTCBsjAOBgNVHQ8BAf8EBAMCBaAwCQYDVR0TBAIwADAdBgNVHS
UEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwNgYDVR0RBC8wLYIrdnN0cy12c3RzZ2hydC1naC12c28tb2F1
dGgudmlzdWFsc3R1ZGlvLmNvbTAfBgNVHSMEGDAWgBQLxdObdnfWzaBcxau87tdSUtEuQjAdBgNVHQ4
EFgQUC8XTm3Z31s2gXMWrvO7XUlLRLklwDQYJKoZIhvcNAQELBQADggEBAD2Eo703wXQgB2vJn/
RwTTcHGkeMkYXm0mWCxOSh4iCKVvqypBJrmLzRkMMJN0/10qIGciYWUl6EkL7yj48tpXXH01Ep0ONDdo9
UYmKGp81Z4j3u3FBJTVQSdj2tnPOPZlYWaBkerlkcleyWBRKvne1UBaobbk84epfBmUfAMFmyJEk+x+q7cqm
sbjDtdrmhiWaInqCijpS2dW2MitJ5F7tBBS26SMTqLQteA2IOwIW1BMlYIPuSO3dKn/
rYVS8RjL+x+MxP98vla5sichoEZwVWnXiXgFZ4n/asGqc+Da9q6lLLtInvgI5bi7kjJJ2ARTRC5/a+J/
v3EL+t8SdnOO8="
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",

```
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn":
"urn:pubkey:sha256:c1c474f69e0bab1bc47a20fd08037a6e33a6308685903d4d329aa6ee8d9a997e",
    "url": "https://token.actions.githubusercontent.com",
    "extra": {
      "oidc_issuer": "https://token.actions.githubusercontent.com",
      "jwk_algorithm": "RS256",
      "jwk_key_id": "38E9B30B3A023A1B72309921A69A42FCC496C42C",
      "jwk_use": "sig",
      "jwk_x5t": "OOmzCzoCOhtyMJkhpppC_MSWxCw",
      "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
      "origin": "oidc_jwks",
      "purpose": "sign_verify"
    },
    "pubkey_pem": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtEq2Fp9HcdT5MwMsB/UTm8j/
woJJLi3sA+y0RX2tioTm581seyfvOH6lJ5JmHVtS+/
fb8B2tRT1pznHQSNq14PsJdu9bp5egbWmIz+5RvhqoM+oKem/MJENCNFuqXijRLT47FRdfH3inqde1vJlA/
JJHCqYMKIpHH7kqNFYcCpwr0vk80Hc2rTyL0uBXI7NqBZbtUgNoyucWO5O7QQrPNOmlr+GI8aFckFRfobC
aCOiH9qW02FtkV74fwBGVCNhNf3a1CK81+O8xEGimvVydI/pQA5B8QqVuQjY/
ntOu555HdirA0hKkY6fsE9eZCMFmWDHZ2kSWLjhabxWxIzSzXQIDAQAB"
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "cert",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:cert:sha256:f8fac529321f1d8ac92fd260cac84988f49b29d1acc41f3b5bb5c854db0824d1",
    "url": "https://token.actions.githubusercontent.com",
    "extra": {
      "cert_index": 0,
      "oidc_issuer": "https://token.actions.githubusercontent.com",
      "issuer_url": "https://token.actions.githubusercontent.com",
      "jwk_algorithm": "RS256",
      "jwk_key_id": "4F3E9AD8C9A6F5EB3173006F4FA630E28F43DCE9",
      "jwk_use": "sig",
      "jwk_x5t": "Tz6a2Mmm9esxcwBvT6Yw4o9D3Ok",
      "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
      "total_certs": 1
    },
    "cert_pem":
"MIIDrDCCApSgAwIBAgIQPQS35v3ITW6fNLO8GX5QBjANBgkqhkiG9w0BAQsFADA2MTQwMgYDVQQDEyt
2c3RzLXZzdHNnaHJ0LWdoLXZzby1vYXV0aC52aXN1YWxzdHVkaW8uY29tMB4XDTI1MDgwNjE0MTEzMloX
DTI3MDgwNjE0MjEzMlowNjE0MDIGA1UEAxMrdnN0cy12c3RzZ2hydC1naC12c28tb2F1dGgudmlzdWFsc3R1
ZGlvLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALRnr6oZARp/DQdHSsbPA3cYZ/
sWT5uebL3EpZCWRDsAzlwy5f9PolTlSjSX3Bxyx7zKmus4dgw3KQFmYthaXgQCbqbK8pqsM5ZcTjg1ilqFSKGi
lzju4dSS4hfSp8LqvqOgT7SncDk/
mzq10cwoTLULHOQtQLnsfg9GepaDmwbitS5JtESvSEkLC6gRMKvINDjrqSP7lVNa9q09htJ5GdAr8VZUHIBk
oRYvDHcrNrOSz6gcIwuCysI5PJqFkFd9ihHHG0zysZxv+d/
```

Q8UYJRyYBzx4rElRrH2SFF4tnIbV3UB2T3Vb3vGUZpJwspGbZCeivO5ZUlF4u6Ms4QiGXYlECAwEAAaOBtT
CBsjAOBgNVHQ8BAf8EBAMCBaAwCQYDVR0TBAIwADAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQU
HAwIwNgYDVR0RBC8wLYIrdnN0cy12c3RzZ2hydC1naC12c28tb2F1dGgudmlzdWFsc3R1ZGlvLmNvbTAfBg
NVHSMEGDAWgBRKoYOga736JYE15vT7b4gWjC1hwTAdBgNVHQ4EFgQUSqGDoGu9+iWBNeb0+2+IFowt
YcEwDQYJKoZIhvcNAQELBQADggEBAJVZIPtoZUlvqgu+Pl0nj8WopA8iuy1m7JRg5fg+bOIGFhXFR8+mH8p
rpeodjUQ40q2Hq6IwnVir+G56zVwAPf2HHksqdp8be9qjkTjD0mJorPCt/
lumrKoNGOVmYffYulyr73hwsl8fN6sGjAyXLFBkozE4s5ssbeodFxiYE1A61SXnzldC00M7qWleMWjTUBixiZ+R/
eroddkLNBGDv9ewDrTQv1ipNec89+Wi7Wb6SAXNxBADiC5kVlFylBgHo3oZNg3KFzZS01REyc4zdH7v1wfZz
ilLlul6ygTyYRYpJCsKrX5D9JW196f2PCzcs+VXMfneRDnvyfjep7Y1Pi8="
  }
}
{
  "result_type": "scan",
  "plugin_id": "oidc-plugin",
  "plugin_version": "1.0.dev",
  "data": {
    "type": "pubkey",
    "timestamp": "2025-12-01T08:31:22-05:00",
    "urn": "urn:pubkey:sha256:759fff3dba00ce211d31cce857a1d1a0b538c2c0ca33b881ad7bf14db9540c28",
    "url": "https://token.actions.githubusercontent.com",
    "extra": {
      "oidc_issuer": "https://token.actions.githubusercontent.com",
      "jwk_algorithm": "RS256",
      "jwk_key_id": "4F3E9AD8C9A6F5EB3173006F4FA630E28F43DCE9",
      "jwk_use": "sig",
      "jwk_x5t": "Tz6a2Mmm9esxcwBvT6Yw4o9D3Ok",
      "jwks_uri": "https://token.actions.githubusercontent.com/.well-known/jwks",
      "origin": "oidc_jwks",
      "purpose": "sign_verify"
    },
    "pubkey_pem":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtGevqhkBGn8NB0dKxs8Ddxhn+xZPm55svcSlkJ
ZEOwDOXDLl/
0+iVOVKNJfcHHLHvMqa6zh2DDcpAWZi2FpeBAJupsrymqwzIlxOODWKWoVIoaIjOO7h1JLiF9Knwuq+o6BP
tKdwOT+bOrXRzChMtQsc5C1Auex+D0Z6loObBuK1Lkm0RK9ISQsLqBEwq8g0OOupl/
shU1r2rT2G0nkZ0CvxVlQeUGShFi8Mdys2s5LPqBwjC4LKwjk8moWQV32KEccbTPKxnG/
539DxRglHJgHPHisSVGsfZIUXi2chtXdQHZPdVve8ZRmknCykZtkJ6K87llSUXi7oyzhCIZdiUQIDAQAB"
  }
}

Network Scanner - SSH plugin output

Output Format

Certificate Scan Results (X.509)

For X.509 certificates found in PKIX-SSH implementations:


```
{
 "resultType": "scan",
  "pluginId": "ssh-scanner-plugin",
  "pluginVersion": "1.0.dev",
```

```
  "data": {
    "type": "cert",
    "timestamp": "2025-10-29T09:06:02-04:00",
    "urn":
"urn:cert:sha256:887ae4b0e6d30e43f120cd0a2d9d5bddc876f126057d2513d7803123b277d9a4",
    "url": "ssh://localhost:2223",
    "extra": {
      "negotiated_algorithms": {
        "compression": "none",
        "encryption": "aes128-gcm@openssh.com",
        "host_key": "rsa-sha2-256-cert-v01@openssh.com",
        "kex": "curve25519-sha256",
        "mac": "hmac-sha2-256-etm@openssh.com"
      }
    },
    "certPem": "MIIDRTCCAi2gAwI...",
    "endpoint": {
      "host": "localhost",
      "port": 2223,
      "service": "ssh",
      "serviceVendor": "Go"
    }
  }
}
```


OpenSSH Certificate Scan Results

For OpenSSH certificate format host keys:


```
{
  "resultType": "scan",
  "pluginId": "ssh-scanner-plugin",
  "pluginVersion": "1.0.dev",
  "data": {
    "type": "openssh-cert",
    "timestamp": "2025-10-29T09:05:30-04:00",
    "urn":
"urn:cert:sha256:bf83eed618d9b012303de3241a69ad797b27f744f9e99eec65f31ea80273aafc",
    "url": "ssh://localhost:2222",
    "extra": {
      "CertType": 2,
      "KeyId": "test-host",
      "Serial": 0,
      "ValidAfter": 1761742980,
      "ValidBefore": 1793192647,
      "ValidPrincipals": [
        "localhost"
      ],
      "negotiated_algorithms": {
        "compression": "none",
        "encryption": "aes128-gcm@openssh.com",
        "host_key": "rsa-sha2-512",
        "kex": "curve25519-sha256",
        "mac": "hmac-sha2-256-etm@openssh.com"
```

```
      }
    },
    "certPem": "MIIBIjANBg....",
    "endpoint": {
      "host": "localhost",
      "port": 2222,
      "service": "ssh",
      "serviceVendor": "OpenSSH_9.9"
    }
  }
}
```


Public Key Scan Results

For SSH host public keys:


```
{
 "type": "pubkey",
    "timestamp": "2025-12-01T08:24:18-05:00",
    "urn":
"urn:pubkey:sha256:1ccfacbc6b13228c1b94b79a0e367107fd8d12b279793b8b084f3c947e5c519a",
    "url": "ssh://10.1.127.26:22",
    "extra": {
      "negotiated_algorithms": {
        "compression": "none",
        "encryption": "aes128-gcm@openssh.com",
        "host_key": "rsa-sha2-512",
        "kex": "curve25519-sha256",
        "mac": "hmac-sha2-256-etm@openssh.com"
      },
      "ssh_key_type": "ecdsa-sha2-nistp256"
    },
    "pubkey_pem": "MFkwEwYHKoZI....",
    "endpoint": {
      "host": "10.1.127.26",
      "port": 22,
      "service": "ssh",
      "service_vendor": "OpenSSH_8.2p1",
      "service_vendor_version": "Ubuntu-4ubuntu0.13"
    }
}
```


State Result

After each scan, the plugin outputs state information:


```
{
  "resultType": "state",
  "pluginId": "ssh-scanner-plugin",
  "pluginVersion": "1.0.1",
  "data": {
    "lastRunDateTime": "2025-10-14T11:12:21-04:00"
  }
}
```

Output Fields

Field	Description
`type`	Asset type: `"cert"` for X.509 certificates, `"openssh-cert"` for OpenSSH certificates, `"pubkey"` for public keys
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name based on SHA256 hash of the certificate or public key
`url`	SSH URL in format `ssh://<host>:<port>`
`certPem`	Base64-encoded certificate (PEM body without headers/footers) - certificates only
`pubkey_pem`	Base64-encoded public key (PEM body without headers/footers) - public keys only
`endpoint`	Object containing endpoint details (see Endpoint Fields below)
`extra`	Additional metadata (see Extra Fields section below)

Endpoint Fields

Field	Description
`host`	Target host address (IP or hostname)
`port`	Target SSH port number
`service`	Always `"ssh"` for this plugin
`serviceVendor`	SSH server software identifier (e.g., `"OpenSSH_9.9"`, `"Go"`)
`serviceVendorVersion`	SSH server version information (e.g., `"Ubuntu-4ubuntu0.13"`). Only present when available

Extra Fields

Common Fields (All Asset Types)

Field	Type	Description
`negotiated_algorithms`	object	Cryptographic algorithms negotiated during the SSH handshake (see Negotiated Algorithms Fields below)

Negotiated Algorithms Fields

The `negotiated_algorithms` object contains the following fields:

Field	Type	Description
`kex`	string	Key exchange algorithm (e.g., `"curve25519-sha256"`, `"diffie-hellman-group14-sha256"`)
`host_key`	string	Host key algorithm (e.g., `"rsa-sha2-512"`, `"ecdsa-sha2-nistp256"`, `"ssh-ed25519"`)
`encryption`	string	Symmetric encryption algorithm when client-to-server and server-to-client are identical (e.g., `"aes128-gcm@openssh.com"`, `"chacha20-poly1305@openssh.com"`)
`encryption_client_to_server`	string	Client-to-server encryption algorithm (only present when different from server-to-client)
`encryption_server_to_client`	string	Server-to-client encryption algorithm (only present when different from client-to-server)
`mac`	string	Message authentication code algorithm when both directions are identical (e.g., `"hmac-sha2-256-etm@openssh.com"`). Not present for AEAD ciphers
`mac_client_to_server`	string	Client-to-server MAC algorithm (only present when different from server-to-client)
`mac_server_to_client`	string	Server-to-client MAC algorithm (only present when different from client-to-server)

Field	Type	Description
`compression`	string	Compression algorithm when both directions are identical (typically `"none"`)
`compression_client_to_server`	string	Client-to-server compression (only present when different from server-to-client)
`compression_server_to_client`	string	Server-to-client compression (only present when different from client-to-server)

Public Key-Specific Fields

Field	Type	Description
`ssh_key_type`	string	SSH key type (e.g., `"ssh-rsa"`, `"ecdsa-sha2-nistp256"`, `"ssh-ed25519"`, `"ssh-dss"`)

Note: Cryptographic algorithm details (algorithm type, key length, curve) are derivable from the PEM-encoded public key and are not duplicated in extra fields.

OpenSSH Certificate-Specific Fields

Field	Type	Description
`valid_after`	integer	Unix timestamp for certificate validity start time
`valid_before`	integer	Unix timestamp for certificate validity end time
`valid_principals`	array	Array of valid principal names for the certificate
`key_id`	string	Certificate key identifier string
`serial`	integer	Certificate serial number
`cert_type`	integer	Certificate type: `1` = user certificate, `2` = host certificate

Tenable Vulnerability Management plugin output

Output Format

The plugin generates certificate scan results in the following standardized format:

```
{
    "type": "cert",
    "timestamp": "2025-10-02T12:01:34-04:00",
    "urn":
"urn:cert:sha256:9bf5e2d39583fe759a2b2343ab0c449f6a27e98aaa4d00f241fa70800d8b83de",
    "url": "https://example.com:443",
    "cert_pem": "MIIFODCCBCC...",
    "endpoint": {
      "host": "example.com",
      "port": 443,
      "service": "https",
      "service_vendor": "nginx",
      "service_vendor_version": "1.20.1",
      "cipher_suites": "TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256,
TLS_AES_128_GCM_SHA256"
    }
}
```

Output Fields

Field	Description
`type`	Asset type (`cert` for certificates, `pubkey` for asymmetric keys, `symkey` for symmetric keys, `secret` for KV secrets)
`timestamp`	RFC3339 timestamp of when the scan was performed
`urn`	Unique resource name based on asset type (SHA256 hash for certificates/public keys, name-based for symmetric keys/secrets)
`url`	Direct link to Vault UI for the asset
`cert_pem`	Base64-encoded certificate (PEM without headers/footers) - certificates only
`endpoint`	Base64-encoded certificate body (without PEM headers/footers or newlines)

Field	Description
extra	Additional metadata (see Extra Fields section below)

Endpoint Fields

Field	Type	Description
host	string	Host address or hostname where the certificate was discovered (from Tenable asset hostname or IPv4)
port	integer	Port number where the certificate was discovered (1-65535)
service	string	Detected service type (e.g., http , https , smtp , ssh). Defaults to port's service from Tenable
service_vendor	string	Detected service vendor (e.g., nginx , Apache , OpenSSH , Postfix). Enriched from Tenable plugin 10107 (HTTP Server) and similar service detection plugins
service_vendor_version	string	Detected service vendor version (e.g., 1.20.1 , 2.4.41). Extracted from service detection output
cipher_suites	string	Comma-separated list of supported SSL/TLS cipher suites detected by Tenable plugin 21643

Note: The service_vendor , service_vendor_version , and cipher_suites fields are populated when Tenable successfully detects service information through plugins like 10107 (HTTP Server), 10263 (SMTP Server), and 21643 (SSL Cipher Suites). These fields may be empty if service detection data is not available.

Filesystem Scanner plugin output

Output Format

Each discovered asset is returned as a JSON object with the following structure:

```
{
  "result_type": "scan",
  "plugin_id": "filesystem-scanner-plugin",
  "plugin_version": "1.1.0",
  "type": "cert",
  "timestamp": "2026-03-05T14:30:45+05:30",
  "urn": "urn:cert:sha256:abcd1234efgh5678",
```

```
  "url": "ssh://admin@10.1.127.33:22/etc/ssl/certs/server.crt",
  "extra": {
    "file_path": "/etc/ssl/certs/server.crt",
    "keystore_type": "certificate",
    "file_size_bytes": 2048,
    "owner": "root",
    "permissions": "644",
    "created_date": "2025-12-15T10:30:00Z",
    "modified_date": "2025-12-15T10:30:00Z",
    "accessed_date": "2026-03-05T10:15:30Z",
    "is_encrypted": false,
    "cryptographic_algorithm": "RSA",
    "cryptographic_length": "2048"
  },
  "cert_pem": "-----BEGIN CERTIFICATE-----\nMIID...\n-----END CERTIFICATE-----"
}
```


Field Descriptions

Field	Type	Description
result_type	string	Always "scan" for discovery results
plugin_id	string	Always "filesystem-scanner-plugin"
plugin_version	string	Plugin version (e.g., "1.1.0")
type	string	Asset type: "cert", "privkey", "pubkey", or "secret"
timestamp	string	ISO8601 discovery timestamp with timezone
urn	string	Unique identifier (URN) for result deduplication
url	string	SSH locator showing discovery path
extra	object	Asset metadata (see table below)
cert_pem	string	Full PEM content (only for cert type)
pubkey_pem	string	Full PEM content (only for pubkey type)

Metadata Fields (in `extra` object)

Field	Type	Description
`file_path`	string	Full filesystem path on remote host
`keystore_type`	string	Detected type (certificate, java_keystore, pkcs12, etc.)
`file_size_bytes`	number	File size in bytes
`owner`	string	File owner username
`permissions`	string	File permissions (e.g., "644", "600")
`created_date`	string	File creation timestamp (ISO8601)
`modified_date`	string	File last modified timestamp (ISO8601)
`accessed_date`	string	File last accessed timestamp (ISO8601)
`is_encrypted`	boolean	true if file appears encrypted, false otherwise
`cryptographic_algorithm`	string	Algorithm if detected (RSA, EC-P256, EdDSA, etc.)
`cryptographic_length`	string	Key/cert length if detected (2048, 4096, 256, etc.)

Security Notes

- **Private Key Material**: Never included in output - only metadata returned for `privkey` type
- **Secret Content**: Never included in output - only metadata returned for keystores/ `secret` type
- **Public Key/Certificate PEM**: Always included - these are non-sensitive public data

Security Considerations

Credential Management

- **Password & Private Key Fields**: Marked `writeOnly` in configuration - never logged or exposed in error messages
- **Credentials in Memory**: Cleared after SSH authentication completes
- **Best Practice**: Use SSH key authentication instead of passwords for better security

SSH Access Control

- Plugin requires valid SSH credentials with filesystem read permissions
- Some directories (`/root/.ssh` , `/etc/ssl/private`) may need elevated privileges
- Configure SSH user with minimal necessary permissions
- Consider using passwordless `sudo` if root access is needed for certain paths

Output Security

- **Private Keys**: Never included in output - only metadata (algorithm, length, owner)
- **Keystores/Secrets**: Never included in output - only metadata (size, owner, encryption status)
- **Certificates & Public Keys**: Fully included (these are non-sensitive public data)

Scan Configuration

- **maxDepth Limit**: Controls recursion depth to prevent excessive scanning and symlink loops
- **Recommended Depth**: 6-10 for typical systems, 3-5 for performance-critical environments
- **Custom Paths**: Override default paths to focus on specific locations and improve performance

Troubleshooting

SSH Connection Errors

Error: "failed to resolve hostname" - Cause: Target host not reachable or hostname not resolvable - Solution: Verify hostname/IP address and network connectivity `bash nslookup hostname ping hostname ssh -v username@hostname`

Error: "SSH authentication failed" - Cause: Incorrect username, password, or private key - Solution: Verify credentials and SSH access `bash ssh -i your-key.pem username@hostname # For key auth ssh username@hostname # For password auth`

Error: "SSH key format error" - Cause: Private key in unsupported format (OpenSSH, PKCS#8, etc.) - Solution: Convert to PEM format ```bash # Convert OpenSSH format to PEM ssh-keygen -p -N "" -m pem -f ~/.ssh/id_rsa

Convert PKCS#8 to PEM openssl pkcs8 -in key.p8 -out key.pem -nocrypt ```

Scan Issues

No Assets Found - Cause 1: Specified directories don't contain cryptographic assets - Solution: Verify paths contain files, adjust `scanPaths` - Cause 2: Permission denied reading files - Solution: Check file permissions, ensure SSH user has read access `bash ssh username@hostname ls -la /etc/ssl`

Scan Timeout or Slow Performance - Cause 1: `maxDepth` too high scanning many directory levels - Solution: Reduce `maxDepth` to 6-8 - Cause 2: `scanPaths` includes large directories with many files - Solution: Focus paths on specific certificate locations - Cause 3: Network latency or slow SSH connection - Solution: Use fast network, check SSH server performance

Configuration Errors

Error: "invalid port: XXXXX" - Solution: Port must be 1-65535

Error: "invalid maxDepth: XXXXX" - Solution: maxDepth must be 1-20

Error: "either password or privateKey must be provided for SSH authentication" - Solution: Provide either `password` OR `privateKey` field for authentication

Error: "only one authentication method allowed: provide either password or privateKey, not both" - Solution: Remove either `password` or `privateKey` - you cannot provide both simultaneously

Building & Testing

Prerequisites

- Go 1.21 or later
- `make` command-line tool
- Docker (optional, for containerized deployment)

Build Plugin

```
cd go-modules/filesystem-scanner-plugin
make build
```

Creates executable: `filesystem-scanner-plugin`

Run Tests

```
# Run unit tests
make test

# Run full test suite including integration tests
make integration-test

# Run linter
make lint
```

Build Docker Image

```
make docker
```

Creates Docker image: `cspd/discovery/plugins/filesystem-scanner-plugin:1.1.0`

Clean Build Artifacts

```
make clean
```

Implementation Details

Architecture

The plugin is implemented as a single file:

- **filesystem-scanner.go** (1,442 lines): CLI interface, SSH client management, filesystem scanning, asset discovery, metadata extraction, configuration parsing and validation

Note: This follows the repository standard where 19 out of 20 plugins use a single hyphenated filename (e.g., `aws-certmanager.go`, `azure-keyvault.go`).

How It Works

1. **Configuration Parsing**: Reads and validates JSON configuration from stdin
2. **SSH Connection**: Connects to remote host using password or key authentication
3. **Auto-Discovery** (if no scanPaths provided): Tests common certificate locations for accessibility
4. **Path Preparation**: Deduplicates and normalizes scan paths to prevent redundant scanning
5. **Recursive Scanning**: Traverses directories up to configured depth, discovering assets
6. **Certificate Chain Detection**: Files with multiple certificates are split into separate results
7. **Asset Classification**: Identifies file types (certificates, keys, keystores) by pattern matching
8. **PEM Validation**: Validates certificate and public key PEM format before including in results
9. **Metadata Extraction**: Reads file metadata (owner, permissions, timestamps) via SSH
10. **Content Processing**: For PEM files, extracts algorithm/key length from certificate or public key
11. **Result Generation**: Creates standardized result objects with URN and metadata
12. **Streaming Output**: Returns results in JSONL format (one result per line)

Key Design Features

Retry Logic - All SSH operations retry up to 3 times with 500ms delays - Transient network failures automatically recover - Persistent failures (auth, permissions) fail immediately

Path Deduplication - Overlapping or duplicate scan paths are automatically consolidated - Parent path `/etc/ssl` eliminates child paths like `/etc/ssl/certs` - Improves performance and prevents error proliferation

Error Context - All errors include specific details (hostname, file path, operation) - Operators can diagnose issues without enabling debug logging

Graceful Degradation - File content read failures don't prevent metadata extraction - Assets reported with metadata URN even if content unreadable

PEM Validation - Certificates and public keys validated during discovery - Invalid/non-PEM cert files skipped to prevent KCM rejection (fixes 550 vs 549 issue) - OpenSSH public keys automatically converted to PEM format

File Pattern Recognition

The plugin recognizes these patterns:

```
Java Keystores:   *.jks, *.keystore
PKCS#12:          *.p12, *.pfx
PEM Files:        *.pem
Certificates:     *.crt, *.cer
Private Keys:     *.key
Public Keys:      *.pub, authorized_keys
OpenSSH Keys:     id_rsa, id_ecdsa, id_ed25519
GnuPG:            *.gpg
NSS Databases:    cert*.db, key*.db
```

URN Generation

Each discovery result gets a unique identifier (URN):

For Certificates:

`Format: urn:cert:sha256:<fingerprint>`
`Example: urn:cert:sha256:a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6`

`Fingerprint: SHA256 hash of DER-encoded certificate`
`Purpose: Content-based deduplication (same cert = same URN)`

For Public Keys:

`Format: urn:pubkey:sha256:<fingerprint>`
`Example: urn:pubkey:sha256:c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6a7b8`

`Fingerprint: SHA256 hash of DER-encoded SubjectPublicKeyInfo (for valid PEM public keys)`
`Fallback: urn:pubkey:name:<filename>:<version> (if PEM parsing fails)`
`Purpose: Content-based deduplication for valid public keys`

For Private Keys:

`Format: urn:privkey:name:<filename>:<version>`
`Example: urn:privkey:name:id_rsa:a1b2c3d4e5f6g7h8`

`Filename: Base filename of the private key file`
`Version: First 16 hex characters of SHA256 hash of full file path`
`Purpose: Path-based identification (no key material in URN for security)`

For Secrets/Keystores:

`Format: urn:secret:name:<filename>:<version>`
`Example: urn:secret:name:keystore.jks:b2c3d4e5f6g7h8i9`

`Filename: Base filename of the keystore/secret file`
`Version: First 16 hex characters of SHA256 hash of full file path`
`Purpose: Path-based identification (no secret content in URN for security)`

Performance Characteristics

Typical scan times on 10GB filesystem:

- **Focused Scan** (e.g., `/etc/ssl,/etc/pki` with depth 4-6): 5-15 seconds
- **Standard Scan** (e.g., `/etc` , `/opt` , `/var/lib` with depth 8-10): 30-60 seconds
- **Deep Scan** (multiple paths with depth 15-20): 60-120 seconds

Performance depends on: - Number of files in scanned directories - Network latency to SSH server - SSH server I/O performance - Configured depth limit

Limitations

- **No Incremental Scans**: Full scan each time - filesystem doesn't provide modification timestamps for filtering
- **SSH-Only**: Cannot scan local filesystem, requires SSH access

- **Keystore Decryption**: Basic encryption detection only - doesn't attempt password-protected keystore decryption
- **Symlink Handling**: Follows symlinks one level only to prevent infinite loops

Qualys CertView plugin output

Output Format


```
{
  "result_type": "scan",
  "plugin_id": "qualys-certview-plugin",
  "plugin_version": "1.1.0.dev",
  "data": {
    "type": "cert",
    "timestamp": "2026-01-30T12:42:49-05:00",
    "urn": "urn:cert:sha256:e6fe22bf45e4f0d3b85c59e02c0f495418e1eb8d3210f788d48cd5e1cb547cd4",
    "url": "https://qualysguard.qg3.apps.qualys.com/certview/#/certificates/189259654",
    "extra": {
      "asset_count": 2,
      "asset_type": "all",
      "certhash": "e6fe22bf45e4f0d3b85c59e02c0f495418e1eb8d3210f788d48cd5e1cb547cd4",
      "certificate_id": 189259654,
      "certificate_type": "Intermediate",
      "created_date": "2026-01-22T18:56:12.000+00:00",
      "endpoint_details": [
        {
          "asset_id": 1060774298,
          "asset_interfaces": [
            {
              "address": "192.178.192.113"
            }
          ],
          "asset_name": "lcyyzn-in-f113.1e100.net",
          "asset_uuid": "a5a21a66-4988-4e8a-bf94-ab19d927d018",
          "host_instances": [
            {
              "fqdn": "",
              "grade": "C",
              "port": 443,
              "pqc_support": false,
              "protocol": "tcp",
              "service": "https",
              "ssl_protocols": [
                "TLSv1",
                "TLSv1.3",
                "TLSv1.2",
                "TLSv1.1"
              ]
            }
          ],
          "primary_ip": "192.178.192.113"
        },
```

```
      {
        "asset_id": 1065786012,
        "asset_interfaces": [
          {
            "address": "142.251.181.147"
          }
        ],
        "asset_name": "www.google.com",
        "asset_uuid": "9db074fa-c935-4bd6-a181-47b59141d0f7",
        "host_instances": [
          {
            "fqdn": "",
            "grade": "C",
            "port": 443,
            "pqc_support": false,
            "protocol": "tcp",
            "service": "https",
            "ssl_protocols": [
              "TLSv1",
              "TLSv1.3",
              "TLSv1.2",
              "TLSv1.1"
            ]
          }
        ],
        "primary_ip": "142.251.181.147"
      }
    ],
    "instance_count": 2,
    "issuer_category": "unapproved",
    "last_found": "2026-01-30T12:42:49-05:00",
    "revocation_status": "Not Revoked",
    "sources": [
      "VM"
    ],
    "updated_date": "2026-01-30T17:42:49.000+00:00"
  },
  "cert_pem": "MIIFCzCCAvOgAwIBAgIQf/
AFoHxM3tEArZ1mpRB7mDANBgkqhkiG9w0BAQsFADBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR2
9vZ2xlIFRydXN0IFNlcnZpY2VzIExMQzEUMBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMjMxMjEzMDkwMDAw
WhcNMjkwMjIwMTQwMDAwWjA7MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVR29vZ2xlIFRydXN0IFNlc
nZpY2VzMQwwCgYDVQQDEwNXUjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCp/5x/
RR5wqFOfytnlDd5GV1d9vl+aWqxG8YSau5HbyfsvAfuSCQAWXqAc+MGr+XgvSszYhaLYWTwO0xj7sfUkDSb
utltkdnwUxy96zqhMt/
TZCPzfhyM1IKjiaeKMTj+xWfpgoh6zySBTGYLKNINtYE3pAJH8do1cCA8Kwtzxc2vFE24KT3rC8gIcLrRjg9ox9i1
1MLL7q8Ju26nADrn5Z9TDJVd06wW06Y613ijNzHoU5HEDy01hLmFXxRmpC5iEGuh5KdmyjS//
V2pm4M6rlagplmNwEmceOuHbsCFx13ye/
aoXbv4r+zgXFNFmp6+atXDMyGOBOozAKql2N87jAgMBAAGjgf4wgfswDgYDVR0PAQH/
BAQDAgGGMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjASBgNVHRMBAf8ECDAGAQH/
AgEAMB0GA1UdDgQWBBTeGx7teRXUPjckwyG77DQ5bUKyMDAfBgNVHSMEGDAWgBTkrysmcRorSCeFL1
JmLO/
wiRNxPjA0BggrBgEFBQcBAQQoMCYwJAYIKwYBBQUHMAKGGGh0dHA6Ly9pLnBraS5nb29nL3IxLmNydDA
rBgNVHR8EJDAiMCCgHqAchhpodHRwOi8vYy5wa2kuZ29vZy9yL3IxLmNybDATBgNVHSAEDDAKMAgGBm
```

eBDAECATANBgkqhkiG9w0BAQsFAAOCAgEARXWL5R87RBOWGqtY8TXJbz3S0DNKhjO6V1FP7sQ02hYST
L8Tnw3UVOIIecAwPJQI8hr0ujKUtjNyC4XuCREINJThb0Lbgpt7fyqaqf9/qdLeSiDLs/
sDA7j4BwXaWZIvGEaYzq9yviQmsR4ATb0IrZNBRAq7x9UBhb+TV+PfdBJTDhEI05vc3ssnbrPCuTNiOcLgNeF
bpwkuGcuRKnZc8d/KI4RApW//mkHgte8y0YWuryUJ8GLFbsLIbjL9uNrizkqRSvOFVU6xddZIMy9vhNkSXJ/
UcZhjJY1pXAprffJBvei7j+Qi151lRehMCofa6WBmiA4fx+FOVsV2/7R6V2nyAiIJJkEd2nSi5SnzxJrlXdaqev3htyt
mOPvoKWa676ATL/
hzfvDaQBEcXd2Ppvy+275W+DKcH0FBbX62xevGiza3F4ydzxl6NJ8hk8R+dDXSqv1MbRT1ybB5W0k8878XS
OjvmiYTDIfyc9acxVJrY/
cykHipa+te1pOhv7wYPYtZ9orGBV5SGOJm4NrB3K1aJar0RfzxC3ikr7Dyc6QwqDTBU39CluVIQeuQRgwG3M
uSxI7zRERDRilGoKb8uY45JzmxWuKxrfwT/478JuHU/
oTxUFqOl2stKnn7QGTq8z29W+GgBLCXSBxC9epaHM0myFH/FJlniXJfHeytWt0="
  }
}

Each scan result includes:

- **Standard fields**: `result_type`, `plugin_id`, `plugin_version`, `type`, `timestamp`, `urn`, `url`, `cert_pem`
- **Extra fields** (Qualys-specific):
  - **Identifiers**: `certificate_id`, `certhash`
  - **Properties**: `certificate_type`, `revocation_status`, `self_signed`, `issuer_category`, `instance_count`, `asset_count`, `imported`, `asset_type` ("managed" or "unmanaged")
  - **Dates**: `created_date`, `updated_date`, `last_found` (ISO8601)
  - **Sources**: Array (CLOUD_AGENT, VM_SCANNER, WAS, EASM, SSL_LABS)
  - **Endpoints**: Asset info, host instances (port/protocol/service/grade/FQDN/SSL protocols/cipher suites)
  - **PQC**: `pqc_supported_algorithms` array (KYBER, DILITHIUM, FALCON, etc.)

Empty fields are omitted per plugin framework standards.

Expanding Coverage

To discover more certificates, configure additional discovery sources in your Qualys portal (Cloud Agent, VM Scanner, WAS, EASM). All sources automatically sync to CertView; no plugin changes needed.

Error Handling

Errors include `error_type` (machine-readable), `error_message` (user-facing), and `error_details` (diagnostic).

Common error types: `JSON_PARSE_ERROR`, `JSON_VALIDATION_ERROR`, `CONNECT_ERROR`, `NETWORK_ERROR`.

Building and Testing

```
# Build
cd go-modules/qualys-certview-plugin
make build

# Run unit tests
make test
```

```
# Run integration tests
make integration-test

# Lint
make lint        # Fast (development)
make lint-full   # Comprehensive (pre-commit)
```

Security & Limitations

Security:

- Credentials marked `writeOnly` in schema for UI masking
- Service accounts with read-only CertView access recommended
- Tokens not persisted; plugin re-authenticates each scan

Limitations:

- Subject to Qualys API rate limits based on subscription tier
- Does not validate certificate cryptography (trusts Qualys data)
- Requires correct gateway URL for authentication

Support

Contact: support@entrust.com

Qualys TotalCloud plugin output

Output Format

Each scan result includes:

Common Fields

- `result_type` : Always `"scan"`
- `plugin_id` : `"qualys-totalcloud-plugin"`
- `plugin_version` : Plugin version string
- `type` : Resource type (`cert` , `pubkey` , `privkey` , `symkey` , `secret`)
- `timestamp` : ISO8601 scan timestamp with timezone
- `urn` : Unique resource identifier (format depends on type)
- `url` : Cloud provider console URL for the resource
- `cert_pem` : Base64-encoded certificate body (certificates only)

URN Formats

- **Certificates**: `urn:cert:sha256:<sha256-fingerprint>` (when PEM available)
  or `urn:cert:<cloud>:<account>:<resource-id>`

- **Public Keys**: `urn:pubkey:sha256:<sha256-fingerprint>` (when key material available) or `urn:pubkey:<cloud>:<account>:<resource-id>`
- **Secrets** (including KMS keys, Key Vault keys, and other identifier-only key references): `urn:secret:name:<name>:<version>`

Extra Fields (Cloud-Specific Metadata)

AWS:

- `platform_type` : "aws"
- `account_id` : AWS account identifier
- `region` : AWS region (e.g., "us-east-1")
- `resource_type` : Qualys resource type (e.g., "SECRETS", "KMS_KEY")
- `arn` : AWS ARN (when available)
- `control_id` : Compliance control identifier (Evaluations API)
- `service` : Service name (e.g., "ACM", "KMS", "Secrets Manager")
- Service-specific fields: `name` , `key_id` , `algorithm` , `key_usage` , `rotation` , etc.

Azure:

- `platform_type` : "azure"
- `subscription_id` : Azure subscription identifier
- `resource_type` : Qualys resource type (e.g., "SQL_SERVER", "KEY_VAULT_KEY")
- `location` : Azure region (e.g., "eastus")
- `control_id` : Compliance control identifier (Evaluations API)
- `service` : Service name (e.g., "Key Vault", "Application Gateway")
- Service-specific fields: `vault_name` , `key_name` , `cert_name` , `name` , `server_name` , etc.

GCP:

- `platform_type` : "gcp"
- `project_id` : GCP project identifier
- `resource_type` : Qualys resource type
- `control_id` : Compliance control identifier
- `service` : Service name (e.g., "Cloud KMS", "Secret Manager")
- Service-specific fields: `key_name` , `name` , `algorithm` , `purpose` , etc.

Example Output


```
{
 "result_type": "scan",
 "plugin_id": "qualys-totalcloud-plugin",
 "plugin_version": "0.0.1",
 "type": "cert",
 "timestamp": "2026-02-18T14:22:00-05:00",
 "urn": "urn:cert:sha256:a1b2c3d4e5f6...",
```

```
 "url": "https://console.aws.amazon.com/acm/home?region=us-east-1#/certificates/abc123",
 "cert_pem": "MIIFjDCCA3SgAwIBAgIS...",
 "extra": {
  "platform_type": "aws",
  "account_id": "123456789012",
  "region": "us-east-1",
  "resource_type": "ACM_CERTIFICATE",
  "domain_name": "example.com",
  "status": "ISSUED",
  "in_use": true
 }
}
```


Resource Type Classification

The plugin applies intelligent classification based on:

1. **Qualys Resource Type**: Primary indicator (e.g., `KMS_KEY`, `KEY_VAULT_CERTIFICATE`)
2. **Service Context**: Service name influences classification (e.g., AWS KMS → keys, Azure Key Vault → keys/certs/secrets)
3. **Control Name Keywords**: Keywords in compliance control names (e.g., "certificate", "encryption key", "secret")
4. **Resource ID Structure**: Path analysis for Azure/GCP resource IDs (e.g., `/keys/`, `/secrets/`, `/certificates/`)

Classification Logic:

- **Certificates**: ACM certificates, IAM server certificates, Application Gateway SSL certs, load balancer listener certs
- **Secrets**: Secrets Manager secrets, AWS KMS keys (identifier only — Qualys returns no key material or KeySpec), Azure Key Vault keys (identifier only — Qualys returns no kty/key_ops), Key Vault secrets, connection strings, API keys, credentials, Route 53 DNSSEC keys (identifier only), Azure Storage Account encryption metadata, Azure CosmosDB encryption metadata

Expanding Coverage

To discover more crypto assets:

1. **Configure Cloud Connectors**: Add AWS accounts, Azure subscriptions, or GCP projects in Qualys CloudView
2. **Enable Services**: Ensure crypto-related services are enabled in your cloud environments (KMS, Key Vault, Secret Manager, ACM, etc.)
3. **Run Compliance Scans**: Qualys evaluates controls automatically; plugin discovers assets from evaluation results
4. **No Plugin Changes Needed**: All cloud resources sync automatically to Qualys CloudView

Error Handling

Errors include `error_type` (machine-readable), `error_message` (user-facing), and `error_details` (diagnostic).

Common error types:

- `JSON_PARSE_ERROR` : Invalid JSON input
- `JSON_VALIDATION_ERROR` : Configuration doesn't match schema
- `CONNECT_ERROR` : Cannot connect to Qualys API or authentication failed
- `NETWORK_ERROR` : Network connectivity issues

API Behaviour and Constraints

API Constraints:

- **Rate Limiting Resilience**: The Qualys CloudView API may throttle requests (HTTP 429) during large scans or when request rate is high. The plugin automatically handles this by retrying up to 5 times with exponential backoff (1s, 2s, 4s, 8s delays between attempts, ~15 seconds total per endpoint). This retry limit is a *defensive circuit breaker* to prevent wasting scan time on struggling endpoints—the Qualys API continues returning 429 regardless of retry count, so after 5 failures, the plugin skips that resource, logs a warning, and continues scanning others. For large environments (1000+ resources) or high-frequency scans, rate limiting may occur; this is expected Qualys API behavior. Consider adjusting scan schedules or consulting Qualys documentation on rate limits for your account tier.
- **Discovery timing depends on Qualys's compliance evaluation schedule (not real-time)**: The plugin discovers resources from Qualys's most recent compliance evaluation results, not directly from cloud providers. For example, if Qualys evaluates CIS controls daily at 2:00 AM, a certificate created at 10:00 AM won't appear in scan results until after the next evaluation run (after 2:00 AM the following day). Incremental scans use `lastScanDate` to filter by Qualys evaluation timestamps, so if Qualys re-evaluates resources after that timestamp, those resources will be returned even if the underlying cloud resources did not change.
- **Assets must be visible in Qualys CloudView to be discovered**: This plugin retrieves data from Qualys CloudView's inventory and compliance evaluations. If cloud provider connectors are not configured, accounts/subscriptions/projects are not added to CloudView, or proper permissions are not granted, resources will not appear in scan results. Ensure connectors are active and syncing before running scans.

Cloud Provider Coverage:

- **AWS**: Full Inventory + Evaluations API coverage for 10 resource types
- **Azure**: Full Inventory + Evaluations API coverage for 7 resource types
- **GCP**: Evaluations API only (Inventory API doesn't include Cloud KMS, Secret Manager, Certificate Manager)
- **OCI**: Not currently supported

Resource Identification:

- Certificates/public keys use SHA256 fingerprint URNs when PEM/key material is available
- When material unavailable, URNs use cloud provider resource identifiers
- Private keys and symmetric keys never include key material (security best practice)

Certificate Data Limitations:

- Qualys CloudView API provides certificate metadata (domain names, expiration, status) but often does not include full PEM-encoded certificate bodies
- When PEM data is unavailable, the plugin generates URNs using cloud provider identifiers (account ID + resource ID)
- This is a Qualys API limitation, not a plugin limitation
- To obtain full certificate details, use cloud provider-native APIs (AWS SDK, Azure SDK, GCP SDK) or other specialized certificate discovery plugins

Data Completeness

Qualys TotalCloud is a cloud security posture management (CSPM) tool — it monitors compliance and vulnerability posture across cloud environments rather than managing cryptographic assets directly. This has two important implications for discovered data:

Why certificate coverage is slim

TotalCloud does not focus on certificate lifecycle management. Certificates are only surfaced incidentally when they appear in compliance controls (e.g., ACM certificates attached to CloudFront distributions). For comprehensive certificate discovery, use dedicated plugins such as `aws-certmanager-plugin` or `azure-keyvault-plugin`.

Why metadata is sparser than dedicated plugins

Dedicated plugins (e.g., `aws-kms-plugin`, `azure-keyvault-plugin`) call asset-specific APIs and return rich metadata — key algorithms, rotation policies, expiry dates, HSM backing, etc. TotalCloud's evaluations API only returns what is needed to assess compliance posture: resource identifiers, control status, and basic properties. It reports *how compliant a resource is*, not *what the resource contains*.

Why secrets and keys have broader coverage

IAM keys, KMS keys, Key Vault secrets/keys, and service account credentials are directly relevant to cloud compliance posture, so TotalCloud surfaces these more consistently across cloud providers.

Building and Testing

```
# Build
cd go-modules/qualys-totalcloud-plugin
make build

# Run unit tests
make test

# Run integration tests (requires mock server)
make integration-test

# Lint
make lint        # Fast (development)
make lint-full   # Comprehensive (pre-commit)
```

Security Best Practices

- **Credentials**: Mark password as `writeOnly` in schema for UI masking
- **Service Accounts**: Use dedicated read-only Qualys accounts
- **State Security**: State contains only timestamps, no sensitive data
- **API Tokens**: Plugin authenticates per-scan; tokens not persisted
- **Cloud Permissions**: Follow least-privilege principle for Qualys cloud connectors

Troubleshooting

No Resources Discovered

- **Check Cloud Connectors**: Verify at least one cloud connector is configured and active in Qualys CloudView
- **Verify Permissions**: Ensure Qualys account has CloudView API and TotalCloud module access
- **Run Compliance Scans**: Qualys must evaluate controls before assets appear in Evaluations API
- **Check Services**: Ensure crypto services (KMS, Key Vault, Secrets Manager, ACM) are deployed in your cloud environments

Authentication Failures

- **Verify Gateway URL**: Ensure URL matches your Qualys platform (qg1, qg2, qg3, or private cloud)
- **Check Credentials**: Username/password must be valid Qualys user credentials
- **API Access**: Confirm API access is enabled for the Qualys account

Incremental Scan Returns All Resources

- **First Scan**: Normal behavior when no state provided (full scan)
- **Qualys Re-Evaluation**: If Qualys runs compliance evaluations on all resources between scans, all resources will match the time filter—this is expected behavior, not a bug
- **Evaluation Schedule**: Incremental scanning filters by Qualys's evaluation timestamps (`evaluatedOn` for Evaluations API, `updated` for Inventory API), not by actual resource modification dates
- **Check State**: Verify state with `lastScanDate` is being passed to plugin
- **Wait Between Scans**: Running scans too frequently (e.g., every minute) may always return full results if Qualys evaluates on a longer schedule (e.g., hourly)

Slow Scans

- **Large Inventories**: Scans with hundreds of controls and thousands of resources may take several minutes
- **API Rate Limits**: Qualys may throttle requests; plugin respects API rate limits
- **Network Latency**: Consider network connectivity between plugin runner and Qualys Gateway

References

- Qualys CloudView API Documentation
- Qualys TotalCloud Platform Overview
- Plugin Framework Documentation

Qualys Container Security plugin output

Example Output

Sample Secret Discovery

```
{
 "result_type": "scan",
 "plugin_id": "qualys-container-security-plugin",
 "plugin_version": "1.1.0.dev",
 "data": {
   "type": "secret",
```

```
  "timestamp": "2026-03-12T16:50:13-04:00",
  "urn": "urn:secret:name:GitHub Personal Access
Token:2f1f1810aa3ee99d340e8b17b41048120374934de6a0aaa6313c6b31f8a926ce:app/.env:2",
  "url": "https://558162184545.dkr.ecr.us-east-1.amazonaws.com/entrust-discovery-test:latest#app/.env",
  "extra": {
   "category": "Github",
   "name": "GitHub Personal Access Token",
   "severity": "critical",
   "file_path": "app/.env",
   "layer_sha": "2f1f1810aa3ee99d340e8b17b41048120374934de6a0aaa6313c6b31f8a926ce",
   "rule_uuid": "d3fa61b2-c95f-11ed-afa1-0242ac120002",
   "last_modified_date": "2026-03-12T16:49:55-04:00",
   "start_line": 2
  }
 }
}
```


Incremental Scanning

This plugin currently supports full scans only.

You may include a `state` object for plugin-runner compatibility:


```
{
 "config": {
  "username": "your-username",
  "password": "your-password",
  "gatewayUrl": "https://gateway.qg3.apps.qualys.com"
 },
 "state": {
  "lastRunDateTime": ""
 }
}
```


If `state.lastRunDateTime` is non-empty, the plugin returns a `state` error:

- `incremental scans not supported currently`

To run scans successfully, either omit `state` or set `lastRunDateTime` to an empty value.

Troubleshooting

Authentication Failures

Error: "authentication failed" or HTTP 401

Causes:

- Invalid username or password
- User permissions insufficient (missing Container Security role)
- Gateway URL incorrect or unreachable
- Account locked or disabled

Solutions:

1. Verify username and password in Qualys console
2. Confirm Container Security module is active in your subscription
3. Check user has "ContainerSecurity - User" role (or higher)
4. Verify gateway URL matches your region
5. Test connectivity: `ping <gateway-hostname>`
6. Check firewall rules allow HTTPS (443) to Qualys gateway

Connection Timeouts

Error: "connection refused" or "i/o timeout"

Causes:

- Network connectivity issues
- Gateway URL unreachable
- Firewall/proxy blocking HTTPS
- Qualys API service unavailable

Solutions:

1. Test connectivity to gateway: `curl -I https://<gateway-url>`
2. Verify firewall allows outbound HTTPS (port 443)
3. Check proxy settings if behind corporate proxy
4. Try a different network connection to rule out ISP blocks
5. Contact Qualys support to check service status

No Secrets Discovered

Possible Causes:

- No container images configured in Qualys Container Security
- Images don't contain detectable secrets
- User permissions don't include Container Security
- Container Security module not active

Solutions:

1. Log into Qualys console and verify images are listed in Container Security
2. Test with known image containing secrets for validation
3. Confirm Container Security module is visible in Qualys console
4. Check user permissions in Qualys (needs Container Security role)
5. Verify API token generation is allowed for user account

API Errors

Error: "list request failed" or HTTP 400/404 errors

Causes:

- API endpoint unavailable or changed

- Qualys API version mismatch
- Gateway URL incorrect
- API rate limiting

Solutions:

1. Verify gateway URL is correct for your region
2. Check Qualys Container Security API documentation version matches plugin
3. Review plugin logs for detailed API error messages
4. If rate limited, wait and retry after delay
5. Contact Qualys support with error details

Building the Plugin

Standard Build

```
make build
```

Creates executable: `qualys-container-security-plugin`

Build with Custom Version

```
go build -ldflags "-X main.Version=1.2.3" -o qualys-container-security-plugin
```

Testing

Run Unit Tests

```
make test
```

Runs unit tests only (skips integration tests due to `-short` flag).

Run Integration Tests

```
make integration-test
```

Runs full test suite including integration tests against a mock Qualys server.

Requirements for Integration Tests:

- Go 1.25.7+ installed
- `git` , `make` available
- No specific Qualys credentials needed (uses mock server)

For detailed integration testing information, see QUALYS_CONTAINER_SECURITY_INTEGRATION_TESTING.md.

Detected Secret Types

The plugin detects secret categories based on Qualys Container Security rules, including but not limited to:

- **AWS Credentials** - AWS Access Keys, Secret Access Keys
- **Private Keys** - RSA, EC, DSA private keys; PEM-encoded keys
- **TLS Certificates** - X.509 certificates, self-signed certificates
- **API Keys** - Google Cloud API keys, GitHub tokens, generic API keys
- **Database Credentials** - Database connection strings with embedded credentials

- **OAuth Tokens** - OAuth bearer tokens, refresh tokens
- **Generic Credentials** - Username/password combinations in config files
- **Certificates in Bundles** - Multiple certificates or keys within a single file

Detection is based on Qualys Container Security pattern matching rules which you can configure and extend within Qualys console. Custom rules can be configured in the Qualys UI; underlying rule definitions are regex-based rule entries in the custom-rules workflow.

Limitations

- Plugin queries Qualys Container Security API, which must have already scanned the container images
- Container images must be accessible to Qualys (either in a registry or on a scanned host)
- Scan speed depends on Qualys API responsiveness and number of images in Container Security module
- Cannot detect secrets outside enabled Qualys Container Security rules (coverage can be extended with custom rules in the Qualys UI)
- Secrets stored as environment variables (both `ENV` in Dockerfile and runtime `-e` / `--env-file`) are not detected; only secrets embedded in image filesystem files are scanned
- Secret detection is supported on **registry and CI/CD sensors only** (Linux, Docker/Containerd/CRI-O); the general/host sensor does not support secret detection
- **Note:** If you encounter multi-arch image rejection in registry-sensor mode, consider trying single-arch builds (e.g., `--platform linux/amd64`) without provenance or SBOM attestations as a potential workaround

References

- Qualys Container Security Documentation
- Qualys Container Security API Guide
- Qualys Gateway URLs and Regions

Tenable Security Center plugin output

Output Format

Example Output


```
{
  "result_type": "scan",
  "plugin_id": "tenable-sc-plugin",
  "plugin_version": "1.1.0",
  "data": {
    "type": "cert",
    "timestamp": "2025-07-01T12:00:00Z",
    "urn": "urn:cert:sha256:a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2",
    "url": "https://web.example.com:443",
    "cert_pem": "MIIDazCCAlOgAwIBAgIUExampleCertificateForDocumentation0DQYJKoZIhvcNAQEL...",
    "endpoint": {
      "host": "web.example.com",
      "port": 443,
      "service": "https",
      "service_vendor": "nginx",
      "service_vendor_version": "1.18.0",
      "cipher_suites": "TLSv12: TLS_AES_256_GCM_SHA384 TLS_AES_128_GCM_SHA256
TLS_CHACHA20_POLY1305_SHA256",
```

```
    "os_type": "Linux"
   }
  }
}
```


Microsoft Defender plugin output

Output Format

Example Output


```
{
  "result_type": "scan",
  "plugin_id": "ms-defender-cert-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-04-15T10:00:00-05:00",
    "urn": "urn:cert:sha1:f7def4b6da84ec5d8264dc0233023313f99454ef",
    "url": "https://security.microsoft.com/vulnerabilities/certificates/
F7DEF4B6DA84EC5D8264DC0233023313F99454EF",
    "extra": {
      "device_id": "15d113df6e2589f6b277699700bc93d0bb1d789c",
      "device_name": "defender-test-e",
      "path": "Microsoft.PowerShell.Security\\Certificate::LocalMachine\\REQUEST\
\F7DEF4B6DA84EC5D8264DC0233023313F99454EF",
      "is_self_signed": false,
      "name": "defender-test-endpoint.eastus.cloudapp.azure.com",
      "signature_algorithm": "sha1RSA",
      "key_length": 2048,
      "serial_number": "6F7B40327AE7BA8E4785CBAACA3A10E2",
      "not_after": "2027-04-09T16:47:48Z",
      "not_before": "2026-04-09T16:27:48Z",
      "subject": "CN=defender-test-
endpoint.eastus.cloudapp.azure.com,OU=PKI,O=Entrust,L=Kanata,ST=ON,C=CA",
      "issuer": "CN=defender-test-
endpoint.eastus.cloudapp.azure.com,OU=PKI,O=Entrust,L=Kanata,ST=ON,C=CA",
      "key_usage": ["Data Encipherment", "Digital Signature", "Key Encipherment", "Non-Repudiation"],
      "extended_key_usage": ["Server Authentication"],
      "rbac_group_id": 0,
      "rbac_group_name": "Unassigned"
    },
    "cert_pem": "",
    "endpoint": {
      "host": "defender-test-e"
    }
  }
}
{
  "result_type": "state",
  "plugin_id": "ms-defender-cert-plugin",
  "plugin_version": "1.2.0",
```

```
  "data": {
    "lastRunDateTime": "2026-04-15T10:00:00-05:00"
  }
}
```


Jenkins Script Plugin output

Output

{"data":{"cert_pem":"MIIEUzCCAzugAwIBAgIIQ2/
LW5uggKMwDQYJKoZIhvcNAQELBQAwaDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAk1BMQ8wDQYDVQ
QHEwZOZXd0b24xETAPBgNVBAoTCEN5YmVyYXJrMQ8wDQYDVQQLEwZCaXpEZXYxFzAVBgNVBAMTD
mN5YmVyYXJrLmxvY2FsMB4XDTIxMDEyODE4MDcwMFoXDTI5MDExMDIwMTYwMFowbzELMAkGA1UEBh
MCVVMxCzAJBgNVBAgTAk1BMQ8wDQYDVQQHEwZOZXd0b24xETAPBgNVBAoTCEN5YmVyYXJrMQ8w
DQYDVQQLEwZCaXpEZXYxHjAcBgNVBAMTFWNsaWVudC5jeWJlcmFyay5sb2NhbDCCASIwDQYJKoZIhvc
NAQEBBQADggEPADCCAQoCggEBAN9n1FpUpJIpbR9+KghwvZbSLk00qlGFDMtV1Za3+CQdjGojzJwNOP
aNOStEBMO9MCv7r+JnAJSGGZaGeGr8ZB9OQHND8vck51RW45Sx+RY33PmD1HV9wtkKtCLESZmZF9UZ
cA/ocAoHRpLBTHiJ5l9KL0V9w1PwJ2aeFXjmUwkUZgdqZZpzgmZMqCh+o/
YuglpJgtZnN1r9dKx9DC9goZU7+/qjB8fBY8Yg7+qkXVYS13ifcGq9w3ulhGxEo5KeuRIfph/
jyd9f96mC2u4Af6UjIvi3ViWqoKRPIFFbpC5lLkDWP3GEVyud6bkOCvya9ZvKdv8u7k8VE+hYBX26DeUCAwE
AAaOB+TCB9jAJBgNVHRMEAjAAMB0GA1UdDgQWBBR1OQHL0fEjJYaGqx21gGzIyzTXnTALBgNVHQ8EBA
MCBPAwIAYDVR0lAQH/
BBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMGgGA1UdEQRhMF+CFWNsaWVudC5jeWJlcmFyay5sb2N
hbIIGY2xpZW50gh1zZXJ2aWNlcy11c2NlbnRyYWwuc2t5dGFwLmNvbYIfKi5zZXJ2aWNlcy11c2NlbnRyYWwu
c2t5dGFwLmNvbTARBglghkgBhvhCAQEEBAMCBaAwHgYJYIZIAYb4QgENBBEWD3hjYSBjZXJ0aWZpY2F0
ZTANBgkqhkiG9w0BAQsFAAOCAQEAQQqqT/TlcTUvrnliMRgHPdZUn1Cae/
G5KzTo1nOKBSUstPUafMxWd7NejDELIievzmMthosSsCjcP0Czy4rq8TrzBjpUlLrKwewyu75mS1ZVpSxU2euV
VFsOvaVFTlWbmQOVcSlZIF4v0Q2Lt4h2zEAnPM1ZKAT2ipQ5FbQ6JXkgfu2vsEVg+Hx2ma3Vs8ITP50PHO6
BK+3hLL5FS/glvd9GP3uJO8z43lOvOMEgf8wG4WUsD7549/
RHK2VQ+rfs4mpHFqUWFHNWSWhjEjdNLedSspBFFcTs6TPfgukXUNKjgnaDXfCZRBAon+0jUjGksdl5tqMC
olFzQpctWRsW1Q==","extra":
{"credential_id":"tesing","credential_type":"FileCredentialsImpl","description":"cyberarck
clinet","display_name":"cyberarck clinet","domain":"_","full_name":"system/_/
tesing","is_self_signed":false,"jenkins_url":"http://
jenkins.example.com:8080","name":"client.cyberark.local","origin":"imported","platform_type":"jenkins","scope":"
GLOBAL","status":"enabled","store":"system","type_name":"Secret
file","id":"tesing"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/
credentials/store/system/domain/_/credential/tesing/
config.xml","urn":"urn:cert:sha256:757e0c78b4fe1c3422ca58821fbaba5e599420ccbce2235916e628f7761fd4
f0"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"db-prod-
credentials","credential_type":"UsernamePasswordCredentialsImpl","display_name":"Production Database
Credentials","domain":"_","full_name":"system/_/db-prod-credentials","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"Username with
password","username":"db_admin"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://
jenkins.example.com:8080/credentials/store/system/domain/_/credential/db-prod-credentials/
config.xml","urn":"urn:secret:name:system/_/db-prod-credentials:1"},"plugin_id":"jenkins-script-
plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"api-key-
github","credential_type":"StringCredentialsImpl","display_name":"GitHub API
Token","domain":"_","full_name":"system/_/api-key-github","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s

tore":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/api-key-github/config.xml","urn":"urn:secret:name:system/_/api-key-github:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"api-key-slack","credential_type":"StringCredentialsImpl","display_name":"Slack Webhook Secret","domain":"_","full_name":"system/_/api-key-slack","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/api-key-slack/config.xml","urn":"urn:secret:name:system/_/api-key-slack:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIC+DCCAeACCQD3+LsTOZ2nkDANBgkqhkiG9w0BAQsFADA+MR0wGwYDVQQDDBR3ZWJhcHAuaW50ZXJuYWwuY29ycDEQMA4GA1UECgwHRW50cnVzdDELMAkGA1UEBhMCVVMwHhcNMjYwNzEzMTAxMzA1WhcNMjcwNzEzMTAxMzA1WjA+MR0wGwYDVQQDDBR3ZWJhcHAuaW50ZXJuYWwuY29ycDEQMA4GA1UECgwHRW50cnVzdDELMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDED/
gITQ54f9T5X8l1S1uarL8uC5JMimfUXFHhAvLfpnzh9JZKiblfwKQFZU27vOYfxvn+VR8a6kV7YMm4EGyp+fc51IXZ9lvuHX08XZYjQWSGstf99E+9MyzNJBjlPKncMrknHWeoyagWedvDaSf4PExHtb4B3igfeue3yIon5AB7/
VIesUa2d6ATS4yJPw0jrM+aK0n0EcSczIxt4aXIlJ/wV6ai6m+CkdcJds/
+Ksi5kqvPVuM+HEUEUKqAL8c4jtfjlOP+3KdBh3i95g3bNPS8+rrlNBnpOmm/MTYE4yHl/
MvUG0l9PLo9QBtGFNsrMWGLxzOYMtZ0cZt30oAhAgMBAAEwDQYJKoZIhvcNAQELBQADggEBACylLaDzVh8IIC5H0cJVT7r2UiDLr3ir48DS03WL4k+kkMJGNhyt9IlOGqlUqfxYh1JXq2mpnq3s0b8ZWkv0g9lm7ejnbvUPM1zI1M8MK5PdAmD260mcb22tj95uFZyAreQAB/3Jkq2WuhuLBQ8e2KX2hFic/7Agan/
XYRsthMilwRn84N7OlmPsTEr5dPJFbeEkQ3DoDCNre6FQsEifYBJkfDa6Z6AcPUnvfjoAQJqg2mnnkrLD7lLW/
8lUV8gpbnf2PUyvmSPBLv9CkFMM3pWMYQ8obKgjsmDgo06xzWx3BYkdS0UgXJKsGeEPHrVW6qu1vdDTiLNsakH0bK2l//Q=","extra":{"credential_id":"web-server-cert","credential_type":"FileCredentialsImpl","description":"Web Application TLS Certificate","display_name":"Web Application TLS Certificate","domain":"_","full_name":"system/_/web-server-cert","is_self_signed":true,"jenkins_url":"http://jenkins.example.com:8080","name":"webapp.internal.corp","origin":"imported","platform_type":"jenkins","status":"enabled","store":"system","type_name":"Secret file","id":"web-server-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/web-server-cert/config.xml","urn":"urn:cert:sha256:e1fe12713a3fc327f601368aea25e9eb0e5b4d6f2038046d7b3ee3ebb1ab260e"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIDHDCCAgQCCQCh+cJgJ3NgHDANBgkqhkiG9w0BAQsFADBQMRkwFwYDVQQDDBBhcGkuZ2F0ZXdheS5jb3JwMRAwDgYDVQQKDAdFbnRydXN0MRQwEgYDVQQLDAtFbmdpbmVlcmluZzELMAkGA1UEBhMCQ0EwHhcNMjYwNzEzMTAxMzA1WhcNMjYwODEyMTAxMzA1WjBQMRkwFwYDVQQDDBBhcGkuZ2F0ZXdheS5jb3JwMRAwDgYDVQQKDAdFbnRydXN0MRQwEgYDVQQLDAtFbmdpbmVlcmluZzELMAkGA1UEBhMCQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDCVMhsCM5xymf2P1/7m/
Fxg71YRYExICYmmCi/
M+LulZjiBd5KMTc2g30sfhKQPv1Nby9TXEX7E79wKmk15N0uk7Q7kaRt5aRxiqOmQuigzugo3bEUQfOPQp3pA5TbxiCkETnt+8BnkV+YzVJdLypVyFzbGRfH7VkxMKKL+FqzBZ3iY34btAmTCkvZEuga8YJfhoj3OxSmDNwkxd5vTY5o5mQNfMY7r23YEXdJGRwyYlCsVnLBOV9p4jQ06VwdxGCsFDEIrS596TMirWiSw+zRAffNq7B94eSY/
I74kTWqW1VZ0o7saKNWK2zDbUx7Ecbu2gX6S7p4fw0IMpgER31hAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAI+qXg1KM1R0EslkEeeWz1eYwyLeKc+zTKByysvfhdIsyNvH8KL+/
C4jsSSi4kJk0ttq8Eau9NyPpLx7cL95IPI5F0pJKUkAHF4eO9nGwMaT7Rda60VyXFMZ4eTjCUHeE7fz5T3RD6Xvg8003chfDHX28W2JswCZTpXUEXtQyBLgRTfhztNDNCvLDePXaOOoPYNCckfSRCW5BLXo/
VUAxqZ0vHDsQupstsc5uWUsOfyBz5pq9mN3MQ/vactE+tqnNiI9ByhQGcqG/
pscs+6UFUTJmqKavgldUH18+R7/S1gPUFNgDYzfjxir+fzOqZqWL7OTQ1TnthHxTddzKHlygPE=","extra":

{"credential_id":"api-gateway-cert","credential_type":"FileCredentialsImpl","description":"API Gateway Certificate (expiring soon)","display_name":"API Gateway Certificate (expiring soon)","domain":"_","full_name":"system/_/api-gateway-cert","is_self_signed":true,"jenkins_url":"http://jenkins.example.com:8080","name":"api.gateway.corp","origin":"imported","platform_type":"jenkins","status":"enabled","store":"system","type_name":"Secret file","id":"api-gateway-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/api-gateway-cert/config.xml","urn":"urn:cert:sha256:34f8cb2bd5dd90f6855aa0c583331c382eff31313278183802add645c27f9c72"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIBnjCCAUQCCQDgccKwMDVfMTAKBggqhkjOPQQDAjBXMSMwIQYDVQQDDBpwYXltZW50cy5taWNyb3NlcnZpY2UuY29ycDEQMA4GA1UECgwHRW50cnVzdDERMA8GA1UECwwIUGxhdGZvcm0xCzAJBgNVBAYTAlVTMB4XDTI2MDcxMzEwMTMwNVoXDTI4MDcxMjEwMTMwNVowVzEjMCEGA1UEAwwacGF5bWVudHMubWljcm9zZXJ2aWNlLmNvcnAxEDAOBgNVBAoMB0VudHJ1c3QxETAPBgNVBAsMCFBsYXRmb3JtMQswCQYDVQQGEwJVUzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABBh23OatFF04qO9BjbmCuJANI2LxUFSHW6AWi/k4fk7ELjiIP98wg7wJLU6C1IkGgR9ALntIC9Grrv98lhO77tMwCgYIKoZIzj0EAwIDSAAwRQIgavQUiNBs9WnfWtBvnVU680xxizOV3ROXqsHLQG8nyNMCIQDZb2ISz3rmYGdSZNTDK1APZJMj0TIo5w/a7D9x3fo3qw==","extra":{"credential_id":"payments-microservice-cert","credential_type":"FileCredentialsImpl","description":"Payments Microservice mTLS Certificate (ECDSA P-256)","display_name":"Payments Microservice mTLS Certificate (ECDSA P-256)","domain":"_","full_name":"system/_/payments-microservice-cert","is_self_signed":true,"jenkins_url":"http://jenkins.example.com:8080","name":"payments.microservice.corp","origin":"imported","platform_type":"jenkins","status":"enabled","store":"system","type_name":"Secret file","id":"payments-microservice-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/payments-microservice-cert/config.xml","urn":"urn:cert:sha256:0ba6ac20de61723c66b485a57a8fcc841456a6097707d0642802cbca22caeed4"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIE8jCCAtoCCQDUouRJ14UZMDANBgkqhkiG9w0BAQsFADA7MRowGAYDVQQDDBEqLmRldi5lbnRydXN0LmNvbTEQMA4GA1UECgwHRW50cnVzdDELMAkGA1UEBhMCVVMwHhcNMjYwNzEzMTAxMzA1WhcNMjcwMTA5MTAxMzA1WjA7MRowGAYDVQQDDBEqLmRldi5lbnRydXN0LmNvbTEQMA4GA1UECgwHRW50cnVzdDELMAkGA1UEBhMCVVMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDLF2UYatc6XHb83n6vrNuytsfXq8z6YIMEbFYTgiemrj3fpV2VwLZMQPxmkHjOYJV334RdWW3GdfkrH9ELLUww2apUi5Gb/YRMgclVERO/bs8ISFfxlxzOBHJ/pbflwK0cXK5uMXlhCvObipNbsM6G9aQX1yjXrO7GvX/937qaXqP2e3RrVAGclv0gPO0e2B21PXUfrJfdRkqassPB/JejuuR95PhayENmkFnj8S4CdPDcyZ7VbcJLO/GVnXfIagnlAIlVPtFzWXmYk/oghEe0dvs7Nm5Zjn9Rg8dE3ktQkPZYwTMdbWvN2KEmoKxWKLw6PJnrsZZuk1c0WZ85sVqJtmSsNm2Fk80xCxyXqeXMqNfPp5C5wqmpB8TKgUsnyXx+/4ngK8naZnmYztZFWq1ky7h8H2VIzqkm47q/q5SECVnJigy0PncW139mg/T67EQSpWasIdjcz4xEu2nEPXdvT0jeCacA25xqVxon3ptVTw4gOdujC1yM1Y2/kLKjU6OaIL/epVyCbV5r4BzCqcOY8pcAWoWLMrAIQdePULpC3WpS4VUvuGEmmOoTzOOi22ZRKZcn0h6PdJhc7rOZQI2SfeREDy+FX1Uk+n4h/VObxolDzRmYb3Lpmsr/ngmVc39oC06HI1VERMPWhNfA2HZU0tKp7JkP2qUr833ij1zXnwIDAQABMA0GCSqGSIb3DQEBCwUAA4ICAQBOiRa/L//bWQG1MI/mxi2MOjP66bI2nWeV+jug+BR0RU4RjkmWwDuoKTRi4G/XvJBSKHkTaNTwTH33ccmZp0AqetcYVQxlKSN84t4s3Pyuf3I3/GcTA9SMc/DQ95jYGlMxOfQqDyWe+QFhPOYgMfCMKaD94Mmtc9zMDJ2X1cRPhaFNPSQVW2W0dpew3PKnzvjPqsCZf2A7F+S4DZySvuGjPkEmpGmyupO2nJnXexT8QGyZz2wrBd8T7kjTW9mSyrbpJKe9L4Z6cQYKgBuQy4pSksRQy9ZWQ31oQlAqlAifv10nHd5KTJSYP2UZyy7D0l8LwjReeRMrAn/JiM2YdjJ3+i1MgiZcwP6YBktQ5lW8H7bpCnLW2GL2ptZC36tzbuUUfQq5/ec8gLAR1G3TekerZvCa6WJexGtkkdEwNLZ3PCCEzr4ZiYmy1u8cyDAZx0s6jIzOth9//PM7CMX6L5t6PHaQZV4so1sCG65z1rbPNshihWY9QiOdSIJIiifHwza3IvWLcWFT63zidscjxcCRklWWXFP5MR

Mk5cScLfKn+rroQc+PlMh1+oTAfp3NdciAC2Q1WOMIY7PdT98ENWNi5Y9PmwcPQGgCRpxwYa35tnqo+OMBYl+Z0WnJ1Rqk49yrouJdNrgjOLqxTTNkHTVLzuqqKiPlRWZ+2tLcSsxT0A==","extra":{"credential_id":"wildcard-dev-cert","credential_type":"FileCredentialsImpl","description":"Wildcard Certificate for *.dev.entrust.com (RSA-4096)","display_name":"Wildcard Certificate for *.dev.entrust.com (RSA-4096)","domain":"_","full_name":"system/_/wildcard-dev-cert","is_self_signed":true,"jenkins_url":"http://jenkins.example.com:8080","name":"*.dev.entrust.com","origin":"imported","platform_type":"jenkins","status":"enabled","store":"system","type_name":"Secret file","id":"wildcard-dev-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/wildcard-dev-cert/config.xml","urn":"urn:cert:sha256:3a10ba8d5607f1ef9dba410c47e2f33b43f63522f105163e03c1ed11a1ae6985"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"docker-registry-creds","credential_type":"UsernamePasswordCredentialsImpl","display_name":"Docker Hub Registry Credentials","domain":"_","full_name":"system/_/docker-registry-creds","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Username with password","username":"entrust-ci"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/docker-registry-creds/config.xml","urn":"urn:secret:name:system/_/docker-registry-creds:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"aws-secret-access-key","credential_type":"StringCredentialsImpl","display_name":"AWS Secret Access Key for CI/CD","domain":"_","full_name":"system/_/aws-secret-access-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/aws-secret-access-key/config.xml","urn":"urn:secret:name:system/_/aws-secret-access-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"artifactory-deploy","credential_type":"UsernamePasswordCredentialsImpl","display_name":"Artifactory Deploy Credentials","domain":"_","full_name":"system/_/artifactory-deploy","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Username with password","username":"ci-deployer"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/artifactory-deploy/config.xml","urn":"urn:secret:name:system/_/artifactory-deploy:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"signing-private-key","credential_type":"FileCredentialsImpl","display_name":"Code Signing Private Key (DO NOT EXPORT)","domain":"_","full_name":"system/_/signing-private-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/signing-private-key/config.xml","urn":"urn:secret:name:system/_/signing-private-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"tls-cert-chain","credential_type":"FileCredentialsImpl","display_name":"TLS Certificate Chain (Root CA)","domain":"_","full_name":"system/_/tls-cert-chain","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/tls-cert-chain/config.xml","urn":"urn:secret:name:system/_/tls-cert-chain:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"deploy-ssh-key","credential_type":"BasicSSHUserPrivateKey","display_name":"Production Deploy SSH Key (Ed25519)","domain":"_","full_name":"system/_/deploy-ssh-key","jenkins_url":"http://

jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"SSH Username with private key"},"timestamp":"2026-07-22T11:05:50Z","type":"privkey","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/deploy-ssh-key/config.xml","urn":"urn:privkey:name:system/_/deploy-ssh-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"backup-ssh-key","credential_type":"BasicSSHUserPrivateKey","display_name":"Backup Server SSH Key (RSA)","domain":"_","full_name":"system/_/backup-ssh-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"SSH Username with private key"},"timestamp":"2026-07-22T11:05:50Z","type":"privkey","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/backup-ssh-key/config.xml","urn":"urn:privkey:name:system/_/backup-ssh-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIB1DCCAXqgAwIBAgIJANd+Q5vdZkZkMAoGCCqGSM49BAMDMF4xCzAJBgNVBAYTAlVTMREwDwYDVQQLEwhQbGF0Zm9ybTEQMA4GA1UEChMHRW50cnVzdDEqMCgGA1UEAxMhbXRscy1zZXJ2aWNlLmludGVybmFsLmVudHJ1c3QuY29tMB4XDTI2MDcxMzEwNDEwMloXDTI4MDcxMjEwNDEwMlowXjELMAkGA1UEBhMCVVMxETAPBgNVBAsTCFBsYXRmb3JtMRAwDgYDVQQKEwdFbnRydXN0MSowKAYDVQQDEyFtdGxzLXNlcnZpY2UuaW50ZXJuYWwuZW50cnVzdC5jb20wWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAARWVLNGF8ihKK3CU+2UN+HDXkfJ5lTyJrHfUYgsNKr/ODES6Ziu2bThOA/c9J+madMQCigcXU+6D8o0Tq7dL2sKoyEwHzAdBgNVHQ4EFgQU9bL33JJSs6Sf0muOEy6XWFhqRIAwCgYIKoZIzj0EAwMDSAAwRQIgQrjYDXLY/RWTkinwZOtK5LuhDJdewpGLcC1ExCS6o7UCIQCmVRJQeAXdgL0LnZ7hMhjYO1lXqwB4KXQJyWjHO6Eo1g==","extra":{"credential_id":"mtls-service-cert","credential_type":"CertificateCredentialsImpl","description":"mTLS Service Certificate (ECDSA P-256, PKCS12)","display_name":"mTLS Service Certificate (ECDSA P-256, PKCS12)","domain":"_","full_name":"system/_/mtls-service-cert","is_self_signed":false,"jenkins_url":"http://jenkins.example.com:8080","name":"mtls-service.internal.entrust.com","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Certificate","id":"mtls-service-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/mtls-service-cert/config.xml","urn":"urn:cert:sha256:4a0a1addc35dff8f5123064c143f16f7cc49144bb4c7704689ab50d700f4e7b0"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":
{"cert_pem":"MIIDJzCCAg+gAwIBAgIIa8HD9RXuwocwDQYJKoZIhvcNAQEMBQAwQjELMAkGA1UEBhMCVVMxEDAOBgNVBAoTB0VudHJ1c3QxITAfBgNVBAMTGGNvZGUtc2lnbmluZy5lbnRydXN0LmNvbTAeFw0yNjA3MTMxMDQ1MjdaFw0yNzA3MTMxMDQ1MjdaMEIxCzAJBgNVBAYTAlVTMRAwDgYDVQQKEwdFbnRydXN0MSEwHwYDVQQDExhjb2RlLXNpZ25pbmcuZW50cnVzdC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDC4b9psZrLAXXMM7JjTuVBmlxtYszXbFeHs48d3Nt2gt6NJWnMA471XBSbplX/1LRon1s7+oft53DQ/Yefbd1XdxHNg1223W7IHExVGGUpAG86OEPrFQar6bbEdrZFYwuNbb8xiTkwl1eJ5+7h2ovbmGmWmbsK67OOD0un/tB9loreIQ/CyQuNtUIBwv17D5EM68wuTftkfgFWL6lqBx7HG/mAkoJT2BBtYcW56sGKbnanR5vCxGfVjhwpiB25LPrSf7b3kPVPUjuL2iuXQJV265PKPeF9cRNwSkAHcwum/tiaMDdn+bJyDWXCgRwCk+8IiEpXyJwEb43wk2/nDwcfAgMBAAGjITAfMB0GA1UdDgQWBBTD1EAIh9sujwG7cXvkJPev+o52xzANBgkqhkiG9w0BAQwFAAOCAQEAwkqSpqYQgz6H6iDs7AQZz1Gzf5Yb7982nKHByWzWUHrFI/htEzyJiN9DyV/QEh4X/Gw+V4TRL39DP2GboZj6kZLN3IeFnjJ20V7iQTCU4YUfwA3JbcuRdGKMISDzj5xmA+1CgInMcu6hpN1wLvMp5VrElbvE4sWhHZxfpY0WcrsC77stYAWsdIF0GK2LKEo+Ew3MRi8PXx1M7N8vMUJj7HZ+NfLMP4ZsIplFuJ1EYzUoqdKmyqxXe/Qrb5Xz7lVtWTUET6RrVeAXrQRR9kbcLjNLuQZ9kfWD/1utSbobo03NvF7HgeHrMR7Blrn5xlgwczh07nZQRKfDY3fJHZDVoA==","extra":{"credential_id":"code-signing-cert","credential_type":"CertificateCredentialsImpl","description":"Code Signing Certificate (RSA-2048, PKCS12)","display_name":"Code Signing Certificate (RSA-2048, PKCS12)","domain":"_","full_name":"system/_/code-signing-cert","is_self_signed":false,"jenkins_url":"http://jenkins.example.com:8080","name":"code-

signing.entrust.com","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Certificate","id":"code-signing-cert"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/code-signing-cert/config.xml","urn":"urn:cert:sha256:a78a6e1da18dc43d6afb5926598f34f8df00186661f5d15d3784f7ab90c2e0b3"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"vault-token","credential_type":"VaultTokenCredential","display_name":"Vault Root Token","domain":"_","full_name":"system/_/vault-token","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Vault Token Credential"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/vault-token/config.xml","urn":"urn:secret:name:system/_/vault-token:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"github-app-cred","credential_type":"GitHubAppCredentials","display_name":"GitHub App for org","domain":"_","full_name":"system/_/github-app-cred","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"GitHub App"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/github-app-cred/config.xml","urn":"urn:secret:name:system/_/github-app-cred:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"openshift-token","credential_type":"OpenShiftTokenCredentials","display_name":"OpenShift Cluster Token","domain":"_","full_name":"system/_/openshift-token","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"OpenShift Token for OpenShift Client Plugin"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/openshift-token/config.xml","urn":"urn:secret:name:system/_/openshift-token:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"bitbucket-pat","credential_type":"StringCredentialsImpl","display_name":"Bitbucket Personal Access Token","domain":"_","full_name":"system/_/bitbucket-pat","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/bitbucket-pat/config.xml","urn":"urn:secret:name:system/_/bitbucket-pat:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"gcp-service-account","credential_type":"FileCredentialsImpl","display_name":"GCP Service Account Key (JSON)","domain":"_","full_name":"system/_/gcp-service-account","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/gcp-service-account/config.xml","urn":"urn:secret:name:system/_/gcp-service-account:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"nexus-deploy","credential_type":"UsernamePasswordCredentialsImpl","display_name":"Nexus Repository Credentials","domain":"_","full_name":"system/_/nexus-deploy","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Username with password","username":"nx-deployer"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/nexus-deploy/config.xml","urn":"urn:secret:name:system/_/nexus-deploy:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}

{"data":{"extra":{"credential_id":"sonarqube-token","credential_type":"UsernamePasswordCredentialsImpl","display_name":"SonarQube Analysis Token","domain":"_","full_name":"system/_/sonarqube-token","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Username with password","username":"sqa_xxxxxx"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/sonarqube-token/config.xml","urn":"urn:secret:name:system/_/sonarqube-token:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"k8s-kubeconfig","credential_type":"FileCredentialsImpl","display_name":"Kubernetes Cluster Kubeconfig","domain":"_","full_name":"system/_/k8s-kubeconfig","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/k8s-kubeconfig/config.xml","urn":"urn:secret:name:system/_/k8s-kubeconfig:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"ssh-known-hosts","credential_type":"FileCredentialsImpl","display_name":"SSH Known Hosts File","domain":"_","full_name":"system/_/ssh-known-hosts","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/ssh-known-hosts/config.xml","urn":"urn:secret:name:system/_/ssh-known-hosts:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"docker-config-json","credential_type":"FileCredentialsImpl","display_name":"Docker Registry Config","domain":"_","full_name":"system/_/docker-config-json","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/docker-config-json/config.xml","urn":"urn:secret:name:system/_/docker-config-json:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"npm-token","credential_type":"StringCredentialsImpl","display_name":"NPM Registry Token","domain":"_","full_name":"system/_/npm-token","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/npm-token/config.xml","urn":"urn:secret:name:system/_/npm-token:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"datadog-api-key","credential_type":"StringCredentialsImpl","display_name":"Datadog API Key","domain":"_","full_name":"system/_/datadog-api-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/datadog-api-key/config.xml","urn":"urn:secret:name:system/_/datadog-api-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"jira-token","credential_type":"StringCredentialsImpl","display_name":"JIRA API Token","domain":"_","full_name":"system/_/jira-token","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/credentials/store/system/domain/_/credential/jira-token/config.xml","urn":"urn:secret:name:system/_/jira-token:1"},"plugin_id":"jenkins-script-

plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"slack-bot-
token","credential_type":"StringCredentialsImpl","display_name":"Slack Bot
Token","domain":"_","full_name":"system/_/slack-bot-token","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://
jenkins.example.com:8080/credentials/store/system/domain/_/credential/slack-bot-token/
config.xml","urn":"urn:secret:name:system/_/slack-bot-token:1"},"plugin_id":"jenkins-script-
plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"pagerduty-
key","credential_type":"StringCredentialsImpl","display_name":"PagerDuty Integration
Key","domain":"_","full_name":"system/_/pagerduty-key","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://
jenkins.example.com:8080/credentials/store/system/domain/_/credential/pagerduty-key/
config.xml","urn":"urn:secret:name:system/_/pagerduty-key:1"},"plugin_id":"jenkins-script-
plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":
{"credential_id":"cyberarkck","credential_type":"FileCredentialsImpl","display_name":"cyberarkckpem","domain":"
_","full_name":"system/_/cyberarkck","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://
jenkins.example.com:8080/credentials/store/system/domain/_/credential/cyberarkck/
config.xml","urn":"urn:secret:name:system/_/cyberarkck:1"},"plugin_id":"jenkins-script-
plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":
{"credential_id":"abcdef","credential_type":"VaultTokenCredential","display_name":"abcdef","domain":"_","full_na
me":"system/_/abcdef","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"SYSTEM","status":"enabled","s
tore":"system","type_name":"Vault Token
Credential"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/
credentials/store/system/domain/_/credential/abcdef/config.xml","urn":"urn:secret:name:system/_/
abcdef:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"test-pubkey-001","credential_type":"FileCredentialsImpl","display_name":"Test
Public Key for Discovery","domain":"_","full_name":"system/_/test-pubkey-001","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"Secret file"},"pubkey_pem":"-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAyc8oNAZMOa8/
TRonwktG\ng6YIg4DISxp3kMMrz3ZxCZYX6fKlpxfYOj71St37W65sto2S04LrtoT2Ipd2yrFr\nFhmmpZvADB/
wtMxkbDPmu1BeLtJ/iNmEwlwcIgH7ghfTbr8RhZUn+lwmrFVttMts\nxC6gTHJg3MN2UyPy6EWdkD/
VstF+rk3qwp7wQ3p4VcqLNz7YHg9uYdd/
6w8Pwx35\neAusaoIK4tWzDEifsx9W9yp0yMJcQUSkHNsUEbG2XHRDmpf9x8Ld/
R01kT9CN0K7\nmcGWbMPnsoEoR6s0shTQFc7GJA5cyZT+afRUJm+G9L8pPs2hL6yLhqHbf+qFEyzl\noQID
AQAB\n-----END PUBLIC KEY-----","timestamp":"2026-07-22T11:05:50Z","type":"pubkey","url":"http://
jenkins.example.com:8080/credentials/store/system/domain/_/credential/test-pubkey-001/
config.xml","urn":"urn:pubkey:name:system/_/test-pubkey-001:1"},"plugin_id":"jenkins-script-
plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"cert_error":"no certificates found in
keystore","credential_id":"testing1","credential_type":"DockerServerCredentials","description":"testing1","display
name":"testing1","domain":"","full_name":"system/_/testing1","jenkins_url":"http://
jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","s
tore":"system","type_name":"X.509 Client
Certificate"},"timestamp":"2026-07-22T11:05:50Z","type":"cert","url":"http://jenkins.example.com:8080/

credentials/store/system/domain/_/credential/testing1/config.xml","urn":"urn:cert:name:system/_/testing1:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"prod-db-password","credential_type":"StringCredentialsImpl","display_name":"Production DB Password","domain":"_","folder":"production","full_name":"production/folder/_/prod-db-password","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"folder","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/job/production/credentials/store/folder/domain/_/credential/prod-db-password/config.xml","urn":"urn:secret:name:production/folder/_/prod-db-password:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"prod-api-key","credential_type":"StringCredentialsImpl","display_name":"Production API Key","domain":"_","folder":"production","full_name":"production/folder/_/prod-api-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"folder","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/job/production/credentials/store/folder/domain/_/credential/prod-api-key/config.xml","urn":"urn:secret:name:production/folder/_/prod-api-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"deploy-tls-cert","credential_type":"FileCredentialsImpl","display_name":"Deployment TLS Certificate","domain":"_","folder":"production/deployments","full_name":"production/deployments/folder/_/deploy-tls-cert","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"folder","type_name":"Secret file"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/job/production/job/deployments/credentials/store/folder/domain/_/credential/deploy-tls-cert/config.xml","urn":"urn:secret:name:production/deployments/folder/_/deploy-tls-cert:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"deploy-secret","credential_type":"StringCredentialsImpl","display_name":"Deployment K8s Secret","domain":"_","folder":"production/deployments","full_name":"production/deployments/folder/_/deploy-secret","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"folder","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/job/production/job/deployments/credentials/store/folder/domain/_/credential/deploy-secret/config.xml","urn":"urn:secret:name:production/deployments/folder/_/deploy-secret:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{"extra":{"credential_id":"staging-api-key","credential_type":"StringCredentialsImpl","display_name":"Staging API Key","domain":"_","folder":"production/staging","full_name":"production/staging/folder/_/staging-api-key","jenkins_url":"http://jenkins.example.com:8080","origin":"imported","platform_type":"jenkins","scope":"GLOBAL","status":"enabled","store":"folder","type_name":"Secret text"},"timestamp":"2026-07-22T11:05:50Z","type":"secret","url":"http://jenkins.example.com:8080/job/production/job/staging/credentials/store/folder/domain/_/credential/staging-api-key/config.xml","urn":"urn:secret:name:production/staging/folder/_/staging-api-key:1"},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"scan"}
{"data":{},"plugin_id":"jenkins-script-plugin","plugin_version":"1.2.0","result_type":"state"}

Jenkins Rest Plugin output

Output Format

Example Output


```
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/system/_/bitbucket-http-access-token:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/bitbucket-http-access-token/
config.xml",
    "extra": {
      "credential_id": "bitbucket-http-access-token",
      "credential_type": "UsernamePasswordCredentialsImpl",
      "display_name": "bitbucket-http-access-token",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Username with password",
      "username": "rzhang"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "privkey",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:privkey:name:jenkins:a3dcb54241ac7a18/system/_/bitbucket-ssh-key:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/bitbucket-ssh-key/config.xml",
    "extra": {
      "credential_id": "bitbucket-ssh-key",
      "credential_type": "BasicSSHUserPrivateKey",
      "display_name": "rzhang",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "SSH Username with private key",
      "username": "rzhang"
    }
  }
```

```
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/system/_/rzhang-kubeconfig-config:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/rzhang-kubeconfig-config/
config.xml",
    "extra": {
      "credential_id": "rzhang-kubeconfig-config",
      "credential_type": "FileCredentialsImpl",
      "display_name": "kubeconfig.yaml",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Secret file"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/system/_/rzhang-secret-token:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/rzhang-secret-token/
config.xml",
    "extra": {
      "credential_id": "rzhang-secret-token",
      "credential_type": "StringCredentialsImpl",
      "display_name": "rzhang-secret-token",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Secret text"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "key",
```

```
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:key:name:jenkins:a3dcb54241ac7a18/system/_/rzhang-p12-example:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/rzhang-p12-example/
config.xml",
    "extra": {
      "credential_id": "rzhang-p12-example",
      "credential_type": "CertificateCredentialsImpl",
      "display_name": "CN=apiclient",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Certificate"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/system/_/1password-service-account:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/1password-service-account/
config.xml",
    "extra": {
      "credential_id": "1password-service-account",
      "credential_type": "StringCredentialsImpl",
      "display_name": "1password-service-account",
      "domain": "_",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Secret text"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/system/pmsplbbitbucket01.corporate.datacard.com/
rzhang-domain-username-password:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/
pmsplbbitbucket01.corporate.datacard.com/credential/rzhang-domain-username-password/config.xml",
    "extra": {
      "credential_id": "rzhang-domain-username-password",
```

```
      "credential_type": "UsernamePasswordCredentialsImpl",
      "display_name": "rzhang/******",
      "domain": "pmsplbbitbucket01.corporate.datacard.com",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Username with password",
      "username": "rzhang"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/folder_for_test_folder_credentials/folder/_/folder-
username-pwd:1",
    "url": "http://10.1.127.33:8080/job/folder_for_test_folder_credentials/credentials/store/folder/domain/_/
credential/folder-username-pwd/config.xml",
    "extra": {
      "credential_id": "folder-username-pwd",
      "credential_type": "UsernamePasswordCredentialsImpl",
      "display_name": "folder-rzhang/******",
      "domain": "_",
      "folder": "folder_for_test_folder_credentials",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "folder",
      "type_name": "Username with password",
      "username": "folder-rzhang"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-07-15T15:10:07-04:00",
    "urn": "urn:secret:name:jenkins:a3dcb54241ac7a18/folder_for_test_folder_credentials/
sub_folder_for_test_folder_credentials/folder/_/sub-folder-username-pwd:1",
    "url": "http://10.1.127.33:8080/job/folder_for_test_folder_credentials/job/
sub_folder_for_test_folder_credentials/credentials/store/folder/domain/_/credential/sub-folder-username-
pwd/config.xml",
    "extra": {
      "credential_id": "sub-folder-username-pwd",
      "credential_type": "UsernamePasswordCredentialsImpl",
```

```
      "display_name": "sub-rzhang/******",
      "domain": "_",
      "folder": "folder_for_test_folder_credentials/sub_folder_for_test_folder_credentials",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "folder",
      "type_name": "Username with password",
      "username": "sub-rzhang"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "key",
    "timestamp": "2026-07-22T10:57:57-04:00",
    "urn": "urn:key:name:jenkins:a3dcb54241ac7a18/system/_/pem-with-privatekey-no-pwd:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/pem-with-privatekey-no-
pwd/config.xml",
    "extra": {
      "credential_id": "pem-with-privatekey-no-pwd",
      "credential_type": "CertificateCredentialsImpl",
      "description": "Certificate with Private Key no password",
      "display_name": "CN=apiclient (Certificate with Private Key no password)",
      "domain": "_",
      "full_name": "system/_/pem-with-privatekey-no-pwd",
      "jenkins_url": "http://10.1.127.33:8080",
      "platform_type": "jenkins",
      "scope": "GLOBAL",
      "store": "system",
      "type_name": "Certificate"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "type": "key",
    "timestamp": "2026-07-22T10:57:57-04:00",
    "urn": "urn:key:name:jenkins:a3dcb54241ac7a18/system/_/pem-with-privatekey-with-pwd:1",
    "url": "http://10.1.127.33:8080/credentials/store/system/domain/_/credential/pem-with-privatekey-with-
pwd/config.xml",
    "extra": {
      "credential_id": "pem-with-privatekey-with-pwd",
      "credential_type": "CertificateCredentialsImpl",
      "description": "Certificate with Private Key with password",
      "display_name": "CN=apiclient (Certificate with Private Key with password)",
      "domain": "_",
```

```
    "full_name": "system/_/pem-with-privatekey-with-pwd",
    "jenkins_url": "http://10.1.127.33:8080",
    "platform_type": "jenkins",
    "scope": "GLOBAL",
    "store": "system",
    "type_name": "Certificate"
   }
  }
}
{
  "result_type": "state",
  "plugin_id": "jenkins-rest-plugin",
  "plugin_version": "1.2.0",
  "data": {
    "lastScanDate": "2026-07-15T15:10:07-04:00"
  }
}
```


Veeam Backup and Replication Plugin output

Output Format

Example Output


```
{
   "result_type": "scan",
   "plugin_id": "veeam-br-plugin",
   "plugin_version": "1.0.0",
   "data": {
      "type": "cert",
      "timestamp": "2026-08-27T11:10:11-04:00",
      "urn": "urn:cert:sha1:017fe59f68de25a2afc5da9b6359221835d92327",
      "url": "https://10.1.127.31/api/v1/serverCertificate",
      "extra": {
         "is_trusted": false,
         "issuer": "WIN-PKVC22U2T6V",
         "key_length": 2048,
         "name": "WIN-PKVC22U2T6V",
         "not_after": "2036-07-12T08:39:12-07:00",
         "not_before": "2026-07-12T08:39:12-07:00",
         "public_key_algorithm": "RSA",
         "serial_number": "06FA8D52FFFC554C",
         "source": "veeam_backup_replication",
         "subject": "CN=WIN-PKVC22U2T6V"
      },
      "cert_pem": ""
   }
}
{
   "result_type": "scan",
   "plugin_id": "veeam-br-plugin",
   "plugin_version": "1.0.0",
```

```
  "data": {
    "type": "cert",
    "timestamp": "2026-08-27T11:10:11-04:00",
    "urn": "urn:cert:sha1:efeeacc7ea19ea0395db5b78876484120713181a",
    "url": "https://10.1.127.31/api/v1/kmsServers/3ddb8686-e1a8-49ba-b9ed-d4465fcfc350",
    "extra": {
      "description": "KMS-server 3ddb8686-e1a8-49ba-b9ed-d4465fcfc350 (10.1.127.50) server certificate",
      "name": "10.1.127.50 server certificate",
      "port": 5696,
      "source": "veeam_backup_replication",
      "veeam_id": "3ddb8686-e1a8-49ba-b9ed-d4465fcfc350"
    },
    "cert_pem": ""
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-08-27T11:10:11-04:00",
    "urn": "urn:cert:sha1:1f5ef3b735591b7a70569818dd7f3c0aa5a3fd53",
    "url": "https://10.1.127.31/api/v1/kmsServers/3ddb8686-e1a8-49ba-b9ed-d4465fcfc350",
    "extra": {
      "description": "KMS-server 3ddb8686-e1a8-49ba-b9ed-d4465fcfc350 (10.1.127.50) client certificate",
      "name": "10.1.127.50 client certificate",
      "port": 5696,
      "source": "veeam_backup_replication",
      "veeam_id": "3ddb8686-e1a8-49ba-b9ed-d4465fcfc350"
    },
    "cert_pem": ""
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:8b9efe28-32af-4b5a-9030-5f45c5f55460",
    "url": "https://10.1.127.31/api/v1/credentials/8b9efe28-32af-4b5a-9030-5f45c5f55460",
    "extra": {
      "created_date": "2026-08-26T10:55:24-07:00",
      "credential_type": "ManagedService",
      "description": "entrust.com/managedacct Managed Service account",
      "name": "8b9efe28-32af-4b5a-9030-5f45c5f55460",
      "source": "veeam_backup_replication",
      "user_name": "entrust.com\\managedacct",
      "veeam_id": "8b9efe28-32af-4b5a-9030-5f45c5f55460"
    }
  }
```

```
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:2adaeb23-9c8a-4ec3-93db-8451b1acc864",
    "url": "https://10.1.127.31/api/v1/credentials/2adaeb23-9c8a-4ec3-93db-8451b1acc864",
    "extra": {
      "created_date": "2026-08-25T13:00:16-07:00",
      "credential_type": "Standard",
      "description": "svc_kasten_acct",
      "name": "2adaeb23-9c8a-4ec3-93db-8451b1acc864",
      "source": "veeam_backup_replication",
      "user_name": "svc_kasten_acct",
      "veeam_id": "2adaeb23-9c8a-4ec3-93db-8451b1acc864"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:10056d89-b50a-4ca5-b71d-c578aaed4e76",
    "url": "https://10.1.127.31/api/v1/credentials/10056d89-b50a-4ca5-b71d-c578aaed4e76",
    "extra": {
      "authentication_type": "SshKey",
      "created_date": "2026-08-26T13:43:07-07:00",
      "credential_type": "Linux",
      "description": "testsshuser with private key",
      "name": "10056d89-b50a-4ca5-b71d-c578aaed4e76",
      "port": 22,
      "source": "veeam_backup_replication",
      "user_name": "testsshuser",
      "veeam_id": "10056d89-b50a-4ca5-b71d-c578aaed4e76"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:6e232983-0205-4f5d-a7a0-45705683d90d",
    "url": "https://10.1.127.31/api/v1/credentials/040023d3-012c-4125-863d-2368ada9fdbb",
    "extra": {
```

```
        "created_date": "2026-07-23T15:28:10-07:00",
        "credential_type": "Standard",
        "description": "Test of add credentials",
        "name": "6e232983-0205-4f5d-a7a0-45705683d90d",
        "source": "veeam_backup_replication",
        "user_name": "Testuser",
        "veeam_id": "040023d3-012c-4125-863d-2368ada9fdbb"
      }
    }
}
{
    "result_type": "scan",
    "plugin_id": "veeam-br-plugin",
    "plugin_version": "1.0.0",
    "data": {
      "type": "secret",
      "timestamp": "2026-08-27T11:10:12-04:00",
      "urn": "urn:secret:name:7a232983-0205-4f5d-a7a0-45705683d99f",
      "url": "https://10.1.127.31/api/v1/credentials/21aabff2-3485-4af2-bc0e-517f61f841fe",
      "extra": {
        "created_date": "2026-07-23T15:35:28-07:00",
        "credential_type": "Standard",
        "description": "Test of add credentials",
        "name": "7a232983-0205-4f5d-a7a0-45705683d99f",
        "source": "veeam_backup_replication",
        "user_name": "Testuser Two",
        "veeam_id": "21aabff2-3485-4af2-bc0e-517f61f841fe"
      }
    }
}
{
    "result_type": "scan",
    "plugin_id": "veeam-br-plugin",
    "plugin_version": "1.0.0",
    "data": {
      "type": "secret",
      "timestamp": "2026-08-27T11:10:12-04:00",
      "urn": "urn:secret:name:ee81a656-1150-4c14-be59-2d0dd57e21b7",
      "url": "https://10.1.127.31/api/v1/cloudCredentials/ee81a656-1150-4c14-be59-2d0dd57e21b7",
      "extra": {
        "access_key": "8d911f4e6fd27332a575283a182bafc13460093f",
        "cloud_provider_type": "Google",
        "description": "Discovery GCP credentials",
        "name": "ee81a656-1150-4c14-be59-2d0dd57e21b7",
        "platform_type": "gcp",
        "source": "veeam_backup_replication",
        "veeam_id": "ee81a656-1150-4c14-be59-2d0dd57e21b7"
      }
    }
}
{
    "result_type": "scan",
    "plugin_id": "veeam-br-plugin",
```

```
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:785902ca-334e-4a93-8c82-7ba0c3f8cbde",
    "url": "https://10.1.127.31/api/v1/cloudCredentials/785902ca-334e-4a93-8c82-7ba0c3f8cbde",
    "extra": {
      "access_key": "AKIAYD5IPBFQUWNMLOH3",
      "cloud_provider_type": "Amazon",
      "description": "Entrust Discovery AWS Cert Manager credentials",
      "name": "785902ca-334e-4a93-8c82-7ba0c3f8cbde",
      "platform_type": "aws",
      "source": "veeam_backup_replication",
      "veeam_id": "785902ca-334e-4a93-8c82-7ba0c3f8cbde"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:b994aea1-742e-44c5-9e54-ff02215765c4",
    "url": "https://10.1.127.31/api/v1/cloudCredentials/b994aea1-742e-44c5-9e54-ff02215765c4",
    "extra": {
      "account": "azzurestorageacct",
      "cloud_provider_type": "AzureStorage",
      "description": "Azure Storage Account with shared key",
      "name": "b994aea1-742e-44c5-9e54-ff02215765c4",
      "platform_type": "azure",
      "source": "veeam_backup_replication",
      "veeam_id": "b994aea1-742e-44c5-9e54-ff02215765c4"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:7c3cf454-abf4-4748-a0e6-e019999614c1",
    "url": "https://10.1.127.31/api/v1/cloudCredentials/7c3cf454-abf4-4748-a0e6-e019999614c1",
    "extra": {
      "access_key": "GOOGEG7XYZ456789ABCDEF12",
      "cloud_provider_type": "Google",
      "description": "GCP account",
      "name": "7c3cf454-abf4-4748-a0e6-e019999614c1",
      "platform_type": "gcp",
      "source": "veeam_backup_replication",
```

```
        "veeam_id": "7c3cf454-abf4-4748-a0e6-e019999614c1"
      }
    }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:ab0cbf59-dc26-4b36-b0d3-8a7154c6ab57",
    "url": "https://10.1.127.31/api/v1/cloudCredentials/ab0cbf59-dc26-4b36-b0d3-8a7154c6ab57",
    "extra": {
      "application_id": "9bbe135b-db22-4b04-b933-3fc5784742c3",
      "cloud_provider_type": "AzureCompute",
      "connection_name": "MS Azure Compute account",
      "deployment_type": "MicrosoftAzure",
      "description": "Microsoft Azure Compute Account",
      "name": "ab0cbf59-dc26-4b36-b0d3-8a7154c6ab57",
      "platform_type": "azure",
      "region": "Global",
      "source": "veeam_backup_replication",
      "subscriptions": [
        {
          "id": "0ad08f73-db75-4405-81f1-df06435278ce",
          "azureSubscriptionId": "31e070fa-08ca-4fcf-9b5f-caaaeb91d779",
          "azureSubscriptionName": "Entrust Corp - Discovery Dev/Test"
        }
      ],
      "tenant_id": "2e62aa76-a049-46c9-9235-5f9c90964887",
      "veeam_id": "ab0cbf59-dc26-4b36-b0d3-8a7154c6ab57"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "veeam-br-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-27T11:10:12-04:00",
    "urn": "urn:secret:name:vcenter01:3fc9fa0e-314f-431d-98a6-6d02b8abb4b4",
    "url": "https://10.1.127.31/api/v1/encryptionPasswords/3fc9fa0e-314f-431d-98a6-6d02b8abb4b4",
    "extra": {
      "current_version": "3fc9fa0e-314f-431d-98a6-6d02b8abb4b4",
      "description": "Encryption password last modified at 2026-07-22T11:24:05.787-07:00",
      "hint": "****name",
      "name": "vcenter01",
      "source": "veeam_backup_replication",
      "veeam_id": "3fc9fa0e-314f-431d-98a6-6d02b8abb4b4"
    }
  }
```

```
}
{
   "result_type": "state",
   "plugin_id": "veeam-br-plugin",
   "plugin_version": "1.0.0",
   "data": {
      "lastScanDate": "2026-08-27T11:10:13-04:00"
   }
}
```


Nutanix Prism Element Discovery plugin output

Output Format

Example Output


```
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-08-13T12:44:52+05:30",
    "urn": "urn:cert:sha256:aee8f6bf9cb9e12fdabb0aa17a13e6edcf634018e25dd0bfb3956eae96449455",
    "url": "https://10.8.58.37:9440",
    "extra": {
      "api_source": "tls-handshake",
      "cert_index": 0,
      "cert_type": "tls-server-certificate",
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "name": "prism.nutanix.local",
      "origin": "nutanix-pe"
    },
    "cert_pem":
"MIID9DCCAtygAwIBAgIUPiX1TtWxwLITAde54HDEWUiEdMUwDQYJKoZIhvcNAQELBQAwejELMAkGA1UEB
hMCVVMxCzAJBgNVBAgMAkNBMREwDwYDVQQHDAhTYW4gSm9zZTEVMBMGA1UECgwMTnV0YW5pe
CBJbmMuMRYwFAYDVQQLDA1NYW5hZ2VhYmlsaXR5MRwwGgYDVQQDDBNwcmlzbS5udXRhbml4LmxvY
2FsMB4XDTI1MTIxMzAyNDIxMVoXDTM1MTIxMTAyNDIxMVowejELMAkGA1UEBhMCVVMxCzAJBgNVBAgM
AkNBMREwDwYDVQQHDAhTYW4gSm9zZTEVMBMGA1UECgwMTnV0YW5peCBJbmMuMRYwFAYDVQQL
DA1NYW5hZ2VhYmlsaXR5MRwwGgYDVQQDDBNwcmlzbS5udXRhbml4LmxvY2FsMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAsCVj+frW2PSiiz0Cbu1W3FK8mL8xN1JNO3B6fbdMqMl/pLORqi/W/
Me9AgusQvCSJQKaMY54oKq+0Y0tNm9rdCU3I2gsdJbhauEP0sUTmP0/ztW7Op/
Mk112OCzAyNgMos56XyV+SggzKAqhTZDewx9dB3TLkNpfYLOitTbb30DjgZMylhOoz5xLyMf5kfnbUeHqgTO
iuMHoP8ypDkIoGtp8+BgAsO3vSTfHeNSMLBHeyEwnnWPbRTc7IYG5chqFt7n8Q2K4IXowx2owrGmjK/
a5go7RblqYj7w0NsxG28cxLJ0rs8zng2/
pnYOjHlx7U6JWfgt1WPqvitiMkidZTQIDAQABo3IwcDAdBgNVHQ4EFgQU/nRvOy9HP/
KzoFP6DIRM8GKSkGgwHwYDVR0jBBgwFoAU/nRvOy9HP/KzoFP6DIRM8GKSkGgwDwYDVR0TAQH/
BAUwAwEB/
zAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwDQYJKoZIhvcNAQELBQADggEBAJwW5rH1Df7I
V4Bq6X8qIvFm9lx7sT8R+WKChU+84SCvOmQcU+8nOmf5ZngEzrsk+SEqtQAVmV9baHsOyml0M4Ek5o3C
0AWCS3ADYeHqFzfR0GeIgE70OG1yM8TnJfOMUZ6n3BUbGUX8H/
```

KNfIB6PMfaj1pXQDsCkZwGLHBDFHirAGEXx4UM7uGg+ZLc/
rIF270uTz5GQHysk9wRu6SRuGswefkV0NRSJdYoar7iL/
chL2Lfav7IYmabvzqHymcrK9oto6+LlwMGGBR5z06j/
CvDqgy+9nby0d7fkcPF33088NvfoWhoa4oni8cYFiqP964bNqB82HyEP0bkRz8+07A="
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-13T12:44:52+05:30",
    "urn": "urn:secret:name:nutanix-cert-metadata-000645cc-2b3a-3b1c-06a8-7cc255ec3615:",
    "url": "https://10.8.58.37:9440/PrismGateway/services/rest/v1/keys/pem",
    "extra": {
      "api_source": "v1",
      "cert_city": "San Jose",
      "cert_common_name": "prism.nutanix.local",
      "cert_country_code": "US",
      "cert_expiry_date": "2035-12-11T02:42:11Z",
      "cert_key_type": "rsa2048",
      "cert_organization_name": "Nutanix Inc.",
      "cert_organizational_unit_count": 1,
      "cert_signature_algorithm": "SHA256withRSA",
      "cert_state": "CA",
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "description": "Certificate metadata from Prism Element",
      "id": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "name": "nutanix-cert-metadata-000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "origin": "nutanix-pe"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "pubkey",
    "timestamp": "2026-08-13T12:44:53+05:30",
    "urn": "urn:pubkey:name:vx:",
    "url": "https://10.8.58.37:9440/PrismGateway/services/rest/v2.0/cluster/public_keys",
    "extra": {
      "api_source": "v2",
      "asset_type": "ssh-public-key",
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "name": "vx",
      "origin": "nutanix-pe",
      "ssh_key_data": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDRjA3wZvymvH7CV0U12f0jWnqaQKbSjBnwe30j8ctUEUvVWdR

axqUyPQpJB9/rAFLYTGiVYBJRo7jMMLpfVTvZ4WshYz7DsR/R7Lj24GUPD4nbgPluT7fLbBl5/
uAJRel5KjaD658ZEJrkc9AXoxMxkkER9DrD6AeNI+DIRteWccBdpymtyyKpTlk2elSmmmqy4UUetMw7ibJ/
rZXuTzLaEu9S1T7v4jtxPAWKIZQwUuDTHBxtegIpFmMCOvyXxeHQJx1YEGy0u4OTgWfewzJS0T/
uSUSy2bn+1bLHeooaXKwV3/iQl7U7anQPz2hFPw5aspvi1xwPKMqZCIJHr7Xm/
sZ17oHBgsVNxPi3nG8Y2r4uidTQg9HgtyTlhos7CT/
TXQFryTnA1TAaUp+BrjpmSHLguOF6vM34v7olFBJPv7yTwXhsSUpooz8VL77cgMm0eQZC7iH3ne6AfC9Vpg
eQIQahZ4oUwGi6fkEnF87zZylx3PMAfW2A7dD6TKN9/VTzdpFzxzQEKCUZ1FzRDMnery7i/
L+IHt8KOcHnaL+mlykwMVvsc1glCIca8co/
p7d48pnLl1igljFfQ0gwCidPKVJmKhtizevI1sfXPR0AWQjnw4OPse4gkC6hCla67ntUeNnPCDSOPv42avbFYgi
WOmZ04dZ4tIliKX4dH3GDcQ=="
    },
    "pubkey_pem": ""
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-13T12:44:54+05:30",
    "urn": "urn:secret:name:nutanix-client-auth-000645cc-2b3a-3b1c-06a8-7cc255ec3615:",
    "url": "https://10.8.58.37:9440/PrismGateway/services/rest/v2.0/authconfig/client_auth/",
    "extra": {
      "api_source": "v2",
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "description": "Client authentication configuration",
      "enable_client_auth": false,
      "id": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "name": "nutanix-client-auth-000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "origin": "nutanix-pe"
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-13T12:44:55+05:30",
    "urn": "urn:secret:name:nutanix-encryption-status-000645cc-2b3a-3b1c-06a8-7cc255ec3615:",
    "url": "https://10.8.58.37:9440/PrismGateway/services/rest/v2.0/data_at_rest_encryption/",
    "extra": {
      "api_source": "v2",
      "asset_type": "encryption-status",
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "description": "Data-at-Rest Encryption configuration status",
      "desired_password_protection": false,
      "disk_count": 18,
      "disk_encrypted_count": 0,

```
      "id": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "is_cluster_empty": false,
      "is_ready_for_cloud_kms": true,
      "is_ready_for_ekmpc": true,
      "is_ready_for_password_protection": false,
      "is_ready_for_software_encryption": false,
      "name": "nutanix-encryption-status-000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "origin": "nutanix-pe",
      "pckey_manager_status": "NOTALLOWED",
      "storage_container_count": 11,
      "storage_container_encrypted_count": 0
    }
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-13T12:44:56+05:30",
    "urn": "urn:secret:name:nutanix-auth-config-000645cc-2b3a-3b1c-06a8-7cc255ec3615:",
    "url": "https://10.8.58.37:9440/PrismGateway/services/rest/v2.0/authconfig/",
    "extra": {
      "api_source": "v2",
      "asset_type": "auth-config",
      "auth_type_count": 2,
      "cluster_name": "RNO-NXRDY003",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "description": "Authentication configuration",
      "directory_count": 1,
      "id": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "name": "nutanix-auth-config-000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "origin": "nutanix-pe"
    }
  }
}
{
  "result_type": "state",
  "plugin_id": "nutanix-pe-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "lastScanDate": "2026-08-13T07:14:58Z"
  }
}
```

Nutanix Prism Central Discovery plugin output

Output Format

Example Output


```
{
  "result_type": "scan",
  "plugin_id": "nutanix-pc-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-08-10T18:06:56+05:30",
    "urn": "urn:cert:sha256:aee8f6bf9cb9e12fdabb0aa17a13e6edcf634018e25dd0bfb3956eae96449455",
    "url": "https://pc.example.com:9440",
    "extra": {
      "api_source": "tls-handshake",
      "cert_index": 0,
      "cert_type": "tls-server-certificate",
      "cluster_name": "PrismCentral",
      "cluster_uuid": "0806a141-8916-471b-99fd-b5119ff280ef",
      "name": "prism.nutanix.local",
      "origin": "nutanix-pc",
      "platform_type": "nutanix"
    },
    "cert_pem": "MIID9DCC..."
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pc-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "cert",
    "timestamp": "2026-08-10T18:06:58+05:30",
    "urn": "urn:cert:sha256:ba969a18b65b61c83051f7da0c0857699bf75dd9eb685337432bb6aea7eb8fae",
    "url": "https://pc.example.com:9440/api/clustermgmt/v4.3/config/clusters/
000645cc-2b3a-3b1c-06a8-7cc255ec3615/ssl-certificate",
    "extra": {
      "api_source": "v4",
      "api_version": "v4.3",
      "cert_index": 0,
      "cert_type": "ssl-certificate",
      "cluster_name": "ExampleCluster",
      "cluster_uuid": "000645cc-2b3a-3b1c-06a8-7cc255ec3615",
      "has_ca_chain": false,
      "has_private_key": false,
      "name": "prism.nutanix.local",
      "origin": "nutanix-pc",
      "platform_type": "nutanix",
      "private_key_algorithm": "RSA_2048"
    },
```

```
    "cert_pem": "MIID9DCC..."
  }
}
{
  "result_type": "scan",
  "plugin_id": "nutanix-pc-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "type": "secret",
    "timestamp": "2026-08-10T18:07:00+05:30",
    "urn": "urn:secret:name:nutanix-kms-example-server-id:",
    "url": "https://pc.example.com:9440/api/security/v4.1/config/key-management-servers/example-server-id",
    "extra": {
      "api_source": "v4",
      "api_version": "v4.1",
      "description": "Key Management Server configuration",
      "has_dek_key": true,
      "id": "example-server-id",
      "kms_provider": "NATIVE",
      "name": "nutanix-kms-example-server-id",
      "origin": "nutanix-pc",
      "platform_type": "nutanix"
    }
  }
}
{
  "result_type": "state",
  "plugin_id": "nutanix-pc-plugin",
  "plugin_version": "1.0.0",
  "data": {
    "lastScanDate": "2026-08-10T12:37:03Z"
  }
}
```


Plugin reference

See below for the available plugins.

- AWS Certificate Manager plugin
- AWS Key Management Service plugin
- AWS Secrets Manager plugin
- AWS CloudFront plugin
- AWS Elastic Load Balancer plugin
- AWS CloudHSM Plugin
- Azure Key Vault plugin
- GCP Certificate Manager plugin
- GCP Key Management Service plugin
- GCP Secret Manager plugin
- HashiCorp Vault plugin
- Network Scanner - TLS plugin
- Network Scanner - OIDC plugin
- Network Scanner - SSH plugin
- Tenable Vulnerability Management plugin

AWS Certificate Manager plugin

AWS Certificate Manager (ACM) is a service provided by Amazon Web Services that makes it easy to provision, manage, and deploy SSL/TLS certificates for use with AWS services and internally connected resources.

The AWS Certificate Manager plugin does the following:

- Connects to your AWS account using IAM credentials (either permanent access keys or temporary session tokens from AWS STS).
- Discovers all X.509 SSL/TLS certificates stored in ACM within a specified AWS region. These include:
  - Public Certificates—Free SSL/TLS certificates issued by ACM for use with AWS services like CloudFront, Application Load Balancers, and API Gateway.
  - Private Certificates—Certificates issued by AWS Private Certificate Authority for internal organizational use.
  - Imported Certificates—Third-party certificates that have been imported into ACM.
- Exports certificate data in PEM format along with comprehensive metadata.

Note:

- The plugin only extracts public certificate information. No private keys are accessed or exported.
- The plugin does not support incremental scanning. Each scan operation processes all certificates in the specified region regardless of when they were last modified.

AWS Key Management Service plugin

AWS Key Management Service (KMS) is a managed service provided by Amazon Web Services that enables easy creation and control of cryptographic keys used to encrypt data. KMS is integrated with other AWS services and supports the following:

- Symmetric keys
- Asymmetric key pairs for signing/encryption
- HMAC keys for message authentication

The Key Management Service plugin:

1. Connects to your AWS account using IAM credentials (either permanent access keys or temporary session tokens from AWS STS).
2. Discovers all cryptographic keys and key states stored in KMS within a specified AWS region, including:
   - Symmetric Keys—the plugin discovers the following types of symmetric keys:

Key	Description
SYMMETRIC_DEFAULT	AES-256-GCM encryption keys for encrypt/decrypt operations
HMAC	HMAC_224, HMAC_256, HMAC_384, HMAC_512 for message authentication

For Symmetric keys:

- The plugin does not export key material. AWS KMS does not allow exporting symmetric key or HMAC key material for security reasons.
- The plugin returns comprehensive metadata, but does not return the key body.

- Asymmetric keys—the plugin discovers the following types of asymmetric keys:

Algorithm	Description
RSA	RSA_2048, RSA_3072, RSA_4096 for signing/verification or encryption/decryption
Elliptic curve	ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521, ECC_SECG_P256K1 for signing/verification
SM2	SM2 (China national standard) for encryption/decryption

For Asymmetric keys:

- The plugin extracts the public key in PEM format.
- The plugin does not export the private key. AWS KMS does not allow export of private key material.

- Key states

Key State	Description
Enabled	The key is available for use
Disabled	The key exists but cannot be used for cryptographic operations
PendingDeletion	The key is scheduled for deletion (7-30 day waiting period)
PendingImport	Waiting for the key material to be imported
Unavailable	The key is temporarily unavailable
Creating	The key is being created

3. Retrieves key metadata.
4. Exports the public key data for asymmetric keys in PEM format.

Note: The plugin does not support incremental scanning; each scan operation processes all keys in the specified region.

AWS Secrets Manager plugin

AWS Secrets Manager is a fully managed service provided by Amazon Web Services that helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

The AWS Secrets Manager plugin:

1. Connects to your AWS account using IAM credentials (either permanent access keys or temporary session tokens from AWS STS).
2. Discovers all secrets stored in Secrets Manager within a specified AWS region. This can be one of the following:
    - Database credentials—Credentials for RDS, Aurora, Redshift, and DocumentDB with optional automatic rotation.
    - API Keys—API keys and tokens for external services
    - Secrets—Application secrets (configuration values, connection strings, and sensitive application data) and custom secrets (any sensitive information stored in JSON or plain text format).
      The plugin does not retrieve secret values:
        - SecretString
        - SecretBinary

      Note: The plugin only collects metadata about the secrets. This ensures the plugin does not require `GetSecretValue` permission and maintains security best practices by not exposing actual secret values during discovery operation
3. Exports comprehensive metadata, including rotation settings, version information, and tags.

AWS CloudFront plugin

Amazon CloudFront is a content delivery network (CDN) service provided by Amazon Web Services that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds. CloudFront uses SSL/TLS certificates to protect data in transit between clients and CloudFront edge locations.

The AWS CloudFront Plugin is a discovery tool that connects to your AWS account to scan and extract certificate information from CloudFront distributions. It retrieves certificates deployed on CloudFront distributions along with their configuration context, validates certificate formats according to RFC 7468 standards, and exports certificate data in a standardized format for certificate inventory and compliance management purposes.

Features

- **Certificate Discovery**: Scans and discovers all certificates deployed on CloudFront distributions
- **Distribution Context**: Provides CloudFront distribution configuration context for each certificate (domain names, origins, cache behaviors)
- **RFC 7468 Compliance**: Ensures extracted certificate data conforms to RFC 7468 PEM encoding standards
- **Flexible Authentication**: Supports both permanent IAM credentials and temporary session tokens
- **Multi-Region Support**: Can scan CloudFront distributions from any AWS region
- **Standardized Output**: Generates certificate scan results in consistent JSON format with URN generation
- **Comprehensive Validation**: Includes configuration validation and schema generation
- **Security-First Design**: Handles sensitive credentials securely with masked input fields

AWS Elastic Load Balancer plugin

Elastic Load Balancing (ELB) is a service provided by Amazon Web Services that automatically distributes incoming application traffic across multiple targets such as Amazon EC2 instances, containers, and IP addresses. ELB uses SSL/TLS certificates to securely terminate connections between clients and load balancers.

ELB provides three types of load balancers:

- **Application Load Balancer (ALB)**: Layer 7 (Application layer) load balancing for web applications, HTTP/HTTPS traffic
- **Network Load Balancer (NLB)**: Layer 4 (Transport layer) load balancing for extreme performance, TLS/TCP traffic
- **Classic Load Balancer (CLB)**: Legacy load balancer (deprecated), supports both Layer 4 and Layer 7

The AWS Elastic Load Balancer Plugin is a discovery tool that connects to your AWS account to scan and extract certificate information from all load balancer types. It retrieves certificates deployed on load balancers along with their configuration context (listeners, target groups, SSL policies), validates certificate formats according to RFC 7468 standards, and exports certificate data in a standardized format for certificate inventory and compliance management purposes.

Features

- **Multi-LB-Type Discovery**: Scans and discovers certificates from ALB, NLB, and CLB load balancers
- **Comprehensive Context**: Provides load balancer configuration context for each certificate (listeners, target groups, SSL policies, availability zones)
- **SSL Policy Details**: Extracts supported protocols and ciphers from SSL policies
- **RFC 7468 Compliance**: Ensures extracted certificate data conforms to RFC 7468 PEM encoding standards
- **Flexible Authentication**: Supports both permanent IAM credentials and temporary session tokens
- **Multi-Region Support**: Can scan load balancers across any AWS region
- **Standardized Output**: Generates certificate scan results in consistent JSON format with URN generation
- **Comprehensive Validation**: Includes configuration validation and schema generation
- **Security-First Design**: Handles sensitive credentials securely with masked input fields

AWS CloudHSM Plugin

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to generate and use your own encryption keys on the AWS Cloud. CloudHSM provides FIPS 140-2 Level 3 validated HSMs, giving you full control over your cryptographic keys and operations.

CloudHSM supports various cryptographic objects:

- **Private Keys**: RSA, EC private keys for signing and decryption
- **Public Keys**: RSA, EC public keys for verification and encryption
- **Symmetric Keys**: AES, DES3 keys for encrypt/decrypt operations
- **Certificates**: X.509 certificates stored in the HSM

The AWS CloudHSM Plugin is a discovery tool that connects to your CloudHSM cluster via PKCS#11 to scan and extract cryptographic object information. It uses AWS Systems Manager (SSM) Session Manager to securely tunnel to an EC2 instance in your VPC that has access to the CloudHSM cluster.

Features

- **PKCS#11 Discovery**: Scans cryptographic objects using standard PKCS#11 interface
- **Multi-Object Type Support**: Discovers private keys, public keys, symmetric keys, and X.509 certificates

- **Certificate Extraction**: Retrieves X.509 certificates in PEM format with SHA256 fingerprint generation
- **Public Key Extraction**: Retrieves public key data for RSA and EC keys in RFC 7468 compliant format
- **SSM Tunneling**: Secure connectivity via AWS Systems Manager Session Manager port forwarding
- **Dual Authentication**: AWS IAM credentials for API access plus CloudHSM crypto user credentials for PKCS#11 access
- **FIPS 140-2 Level 3**: All discovered objects are HSM-backed with compliance context
- **Mock Mode**: Built-in support for testing with mock PKCS#11 server (localhost/127.0.0.1)
- **Standardized Output**: Generates scan results in consistent JSON format with URN generation
- **Key Origin Tracking**: Distinguishes between keys created on HSM vs imported keys

Azure Key Vault plugin

Azure Key Vault is a cloud service provided by Microsoft Azure for securely storing and managing cryptographic keys, certificates, secrets, and other sensitive information. It is widely used to safeguard cryptographic keys and secrets used by cloud applications and services.

The Azure Key Vault plugin:

1. Connects to your Azure Key Vault using Azure Active Directory (Azure AD) service principal authentication with client credentials.
2. Discovers all certificates, keys, and secrets in the vault.
    - X.509 certificates—the plugin does the following:
        - Discovers X.509 certificates stored in Azure Key Vault.
        - Exports the PEM-encoded certificate body in RFC 7468-compliant format.
        - Returns comprehensive certificate metadata.
    - Asymmetric keys—The plugin discovers the following asymmetric keys and exports the PEM-encoded public key.

Algorithm	Description
RSA	RSA-2048, RSA-3072, RSA-4096, RSA-HSM variants
Elliptic curve	P-256, P-384, P-521, P-256K, EC-HSM variants

    - Symmetric key—The plugin discovers the following symmetric keys and exports metadata only. No key material is exported.

OKP	Ed25519 only; no HSM variant
OCT	Symmetric AES key (OCT-HSM variants)

    - Secrets—The plugin discovers generic secrets, including those containing PEM-encoded keys, but only exports metadata. Secret values are not retrieved. For example, if a secret contains a PEM-encoded private or symmetric key, it is classified and reported, but the key material is not exported.

    The plugin does not support incremental scanning; each scan processes all assets in the vault.
3. Exports PEM-encoded certificate bodies and public keys for asymmetric keys, along with comprehensive metadata.
    For security reasons, private key bodies and symmetric key bodies are never exported. Only metadata is returned for these asset types.

GCP Certificate Manager plugin

GCP Certificate Manager (GCM) is a Google Cloud service for centrally managing TLS/SSL certificates at scale, simplifying their acquisition, deployment, and renewal for Google Cloud services, supporting both Google-managed certificates and self-managed certificates imported into it.

The GCP Certificate Manager plugin:

1. Connects to your GCP account using Service account key and IAM permissions.
2. Discovers all X.509 SSL/TLS certificates stored in GCM within a specified GCP region.
    - Public Certificates—Free SSL/TLS certificates for use with GCP services such as Google Cloud load balancers by acquisition from trusted CAs. For example, Let's Encrypt or Google Trust Services.
    - Private Certificates—Certificates issued through integration between Certificate Manager and a private CA pool in CA Service.
    - Imported Certificates—Third-party certificates that have been imported into GCP.
3. Exports certificate data in PEM format along with comprehensive metadata.

The plugin supports incremental scanning to discover only new or updated certificates since the last scan.

GCP Key Management Service plugin

GCP Key Management Service (GCP KMS) is a cloud-hosted key management service provided by Google Cloud Platform (GCP) that allows you to manage symmetric and asymmetric cryptographic keys for their cloud services. These can be either compatible Google cloud services or your own applications.

Google Cloud KMS supports:

- Google-owned and Google-managed encryption keys
- Customer-managed encryption keys (CMEKs)
- Cloud KMS keys
- Cloud HSM keys
- Customer-supplied encryption keys

The GCP KMS plugin:

1. Connects to your GCP account using a Service account key and IAM permissions.
2. Discovers cryptographic keys stored in Google Cloud KMS for specified region.
    - Asymmetric Keys—The plugin discovers the following types of asymmetric keys:

Algorithm	Description
RSA Keys	RSA_SIGN_PKCS1_2048_SHA256, RSA_SIGN_PKCS1_3072_SHA256, RSA_SIGN_PKCS1_4096_SHA256, RSA_SIGN_PKCS1_4096_SHA512, RSA_SIGN_PSS_2048_SHA256, RSA_SIGN_PSS_3072_SHA256, RSA_SIGN_PSS_4096_SHA256, RSA_SIGN_PSS_4096_SHA512
RSA Decryption	RSA_DECRYPT_OAEP_2048_SHA256, RSA_DECRYPT_OAEP_3072_SHA256, RSA_DECRYPT_OAEP_4096_SHA256, RSA_DECRYPT_OAEP_4096_SHA512
EC Keys	EC_SIGN_P256_SHA256, EC_SIGN_P384_SHA384, EC_SIGN_SECP256K1_SHA256

    - Symmetric Keys—The plugin discovers the following types of symmetric keys:

Algorithm	Description
AES	GOOGLE_SYMMETRIC_ENCRYPTION
HMAC	HMAC_SHA1, HMAC_SHA224, HMAC_SHA256, HMAC_SHA384, HMAC_SHA512

3. Retrieves key metadata
4. Exports public key data for asymmetric keys in PEM format.

The plugin does not support incremental scanning.

GCP Secret Manager plugin

GCP Secret Manager is a secrets and credential management service that lets one store and manage sensitive data such as API keys, usernames, passwords, certificates, and more. The service enables one to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

The GCP Secrets Manager plugin:

1. Connects to your GCP account using a Service account key and IAM permissions.
2. Discovers all secrets stored in GCP projects. These include:
    - API keys
    - Secrets
    - Certificates
3. Extracts detailed metadata with support for various secret configurations. These include labels, annotations, expiration times, and rotation policies.

This plugin does not support incremental scanning

HashiCorp Vault plugin

HashiCorp Vault is a widely-used secrets management platform that provides secure storage and access control for sensitive data including API keys, passwords, certificates, and cryptographic keys. It is used to centralize secrets management, implement dynamic secrets, automate certificate lifecycle management, and encrypt data in transit and at rest.

The HashiCorp Vault Discovery Plugin can discover and catalog cryptographic assets stored across various vault secrets engines and extract metadata, including:

- PKI Certificates—TX.509 certificates managed by a Vault's PKI secrets engine.
- Transit Keys—Cryptographic keys used for encryption-as-a-service operations.
- KV Secrets—Key-value secrets that may contain certificates, keys, or other sensitive data.
- Other Secrets Data—Other secrets supported by various Vault secrets engines. This can include database credentials, cloud credentials, and more.

The HashiCorp Vault Discovery Plugin supports the following Vault engines (when enabled and permitted by policy):

- KV (Key-Value) v1/v2
- Transit (cryptographic keys)
- PKI (certificates/issuers)
- Others as supported by plugin version

The plugin does not support incremental scanning. Each scan operation processes all certificates in the specified region regardless of when they were last modified.

Network Scanner - TLS plugin

Network Mapper (TLS) is a powerful open-source network scanning tool used for security auditing, network discovery, and vulnerability assessment. It can identify hosts, services, operating systems, and security configurations across networks.

The Nmap plugin:

1. Leverages the SSL/TLS certificate extraction capabilities to discover X.509 certificates from various network services.
    - Traditional algorithm certificates—The plugin can discover certificates using the following traditional algorithms:

Algorithm	Examples
RSA	1024-bit, 2048-bit, 3072-bit, 4096-bit
ECDSA (Elliptic Curve)	P-256, P-384, P-521 curves
EdDSA	Ed25519, Ed448 (Edwards-curve signatures)
DSA	Legacy DSA implementations (discouraged)

    - Post-Quantum cryptography certificates (supported in future releases)—The plugin is designed to detect and report certificates issued with emerging post-quantum algorithms. For example:

Algorithm	Examples
ML-DSA	(Module-Lattice-Based Digital Signature Algorithm): ML-DSA-44, ML-DSA-65, ML-DSA-87
FALCON	Compact lattice-based signatures
SPHINCS+	Hash-based signatures
ML-KEM (Key Encapsulation)	ML-KEM-512, ML-KEM-768, ML-KEM-1024

    Note: Failed connection attempts due to unreachable hosts or closed ports do not stop the scan.

2. Uses the TLS `ssl-cert` and `ssl-enum-ciphers` NSE scripts to:
    - Detect service types and versions
    - Extract certificates in real-time as services are discovered.

- Enumerate cipher suites

Network Scanner - OIDC plugin

OpenID Connect (OIDC) is an identity layer built on top of the OAuth 2.0 protocol. It allows clients to verify the identity of end-users based on authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner.

The OIDC plugin:

- Connects to OpenID Connect identity providers to scan and extract cryptographic assets from their public endpoints.
- Retrieves key metadata and validates key formats.
- Exports certificate and public key data in PEM format along with comprehensive metadata.

Note: The OIDC plugin does not support incremental scanning

The plugin works with any standards-compliant OIDC provider, including:

OIDC Provider	Login URL
Google Identity Platform	https://accounts.google.com/
Microsoft Azure AD	https://login.microsoftonline.com/{tenant}
Auth0	https://{domain}.auth0.com
GitHub Actions	https://token.actions.githubusercontent.com/
Amazon Cognito	https://cognito-idp.{region}.amazonaws.com/{userPoolId}
Keycloak	https://{keycloak-server}/auth/realms/{realm}
Okta	https://{domain}.okta.com/oauth2/default

Network Scanner - SSH plugin

SSH (Secure Shell) is a cryptographic network protocol used for:

- Secure remote login
- Command execution,
- Other secure network services over insecure networks

SSH servers use public key cryptography for authentication and encryption, making them critical components in cryptographic infrastructure management.

The SSH Scanner plugin:

1. Scans network hosts and IP ranges to identify SSH servers
2. Perform SSH protocol handshakes to retrieve the following:
    - SSH host public keys

- OpenSSH certificates
- X.509 certificates
- SSH server information including the server vendor, server version, supported SSH protocol versions, and negotiated cryptographic algorithms.

Note: The plugin does not require SSH authentication. It only performs initial protocol handshakes to discover publicly available information. Each scan processes all specified hosts and ports completely.

Tenable Vulnerability Management plugin

Tenable Vulnerability Management (TVM) is a cloud-based vulnerability scanning and assessment service that continuously discovers, assesses, and reports on vulnerabilities across an organization's IT infrastructure. Tenable scanners perform comprehensive vulnerability assessments, including:

- Network scanning
- Service detection
- SSL/TLS certificate discovery.

The Tenable Vulnerability Management plugin integrates information retrieved by the following Tenable Vulnerability plugins.

Plugin ID	Description
10863	Retrieves SSL certificate information (primary source of certificate data)
10107	Identifies HTTP server type and version
10263	Identifies SMTP mail server information
21643	Enumerates the supported SSL Cipher Suites

Tenable Vulnerability Management plugin:

1. Authenticates to Tenable using API access and secret keys
2. Retrieves vulnerability data specifically related to SSL certificates (Plugin ID 10863),
3. Enriches the certificate information with service detection data (web server types/versions, SMTP servers) and cipher suite information captured by Tenable scanners.

This plugin supports incremental scanning, which significantly improves performance in environments with large numbers of assets.

Filesystem Scanner Plugin

Introduction

The Filesystem Scanner Plugin discovers cryptographic assets (certificates, keys, and keystores) stored on remote Linux systems by connecting via SSH. It recursively scans specified directories, identifies all cryptographic assets, and extracts comprehensive metadata for inventory and compliance purposes.

This plugin is useful for organizations that: - Maintain cryptographic assets across multiple Linux servers - Need to inventory certificates, keys, and keystores distributed in custom locations - Must track certificate

expiration, key algorithms, and file ownership across their infrastructure - Run compliance or security audits requiring a complete picture of cryptographic assets

The plugin connects to target systems using SSH (password or key-based authentication), scans specified directories recursively, and returns all discovered assets with rich metadata including ownership, permissions, algorithms, and timestamps.

Features

- **Cryptographic Asset Discovery**: Finds 25+ file types including PEM certificates, SSH keys, Java keystores, PKCS#12 containers, GnuPG keyrings, and NSS databases
- **Flexible SSH Authentication**: Supports both password-based and private key authentication
- **Recursive Directory Scanning**: Scans specified directories with configurable depth limits (1-20 levels)
- **Rich Metadata Extraction**: File ownership, permissions, modification times, encryption status, algorithm detection
- **Smart Path Optimization**: Automatic deduplication prevents redundant scanning of overlapping paths
- **Resilient Retry Logic**: Automatic recovery from transient SSH failures with exponential backoff (3 attempts, 500ms delay)
- **Contextual Error Handling**: Detailed error messages with diagnostic information for troubleshooting

Qualys CertView Plugin

Overview

The Qualys CertView Plugin discovers SSL/TLS certificates from Qualys CertView, which aggregates certificate data from multiple sources:

- **Cloud Agent**: Host-based scanning of endpoint certificate stores
- **VM Scanner**: Network-based SSL/TLS certificate probing
- **Web Application Scanning (WAS)**: Certificates found during security assessments
- **External Attack Surface Management (EASM)**: Passive discovery of internet-facing certificates

The plugin exports certificate inventory in standardized JSON format for cryptographic asset management and compliance.

Features

- **Multi-Source Certificate Discovery**: Retrieves certificates from Cloud Agent, VM Scanner, WAS, and EASM
- **Comprehensive Asset Discovery**: Retrieves both MANAGED and UNMANAGED assets with metadata for downstream filtering
- **Incremental Scanning**: Uses Qualys's `updateDate` field for delta scans
- **Post-Quantum Cryptography**: Extracts PQC algorithm metadata (CertView v2.3 API)
- **Rich Metadata**: Asset details, host instances, SSL/TLS protocols, cipher suites, security grades, revocation status
- **Standardized Output**: Consistent JSON format with URN generation
- **State Tracking**: Persists scan state for incremental discovery
- **Secure Credentials**: Masked input fields for sensitive data

Qualys TotalCloud plugin

Overview

The Qualys TotalCloud Plugin discovers cryptographic assets (certificates, keys, and secrets) from cloud infrastructure managed by Qualys CloudView. It integrates with Qualys TotalCloud's compliance and security posture management platform to extract crypto assets from multiple cloud providers.

Supported Cloud Providers:

- **AWS** (Amazon Web Services)
- **Azure** (Microsoft Azure)
- **GCP** (Google Cloud Platform)

Note: OCI (Oracle Cloud Infrastructure) is not currently supported.

Multi-Connector Support: The plugin automatically scans across all cloud connectors configured in your Qualys account, discovering crypto assets from multiple AWS accounts, Azure subscriptions, and GCP projects in a single scan execution.

The plugin uses Qualys CloudView's dual API approach to comprehensively discover crypto assets:

- **Inventory API**: Retrieves resources by type (e.g., AWS Secrets, CloudFront distributions, Azure Key Vault resources)
- **Evaluations API**: Discovers crypto assets through compliance control evaluations

Features

- **Multi-Cloud Discovery**: Scans AWS, Azure, and GCP from a single plugin
- **Dual API Strategy**: Combines Inventory and Evaluations APIs for comprehensive coverage
- **Incremental Scanning**: Time-based filtering using Qualys relative time syntax (`now-5m` , `now-2h` , `now-3d`)
- **Intelligent Filtering**: Client-side control filtering focuses on crypto-relevant compliance checks
- **Resource Type Detection**: Automatically classifies discovered assets as certificates, keys, or secrets
- **Standardized Output**: Consistent JSON format with URNs and rich metadata
- **State Tracking**: Persists scan state for efficient delta scans
- **Secure Credentials**: Password fields masked in UI schema

When the `scan` command-line option is executed, the plugin validates the input configuration and produces the scan output.

- This plugin needs to emit the cryptographic assets from the Qualys TotalCloud endpoints.
- **List of assets - Certs, keys, secrets from AZ, AWS, GCP clouds**
- Can accept input via stdin or a config file
- The output is emitted to stdout or the output file
- The output should be in a standard format, as per plugin documentation
- All the assets should have a urn, url, plugin id, plugin version, type, result type
- If applicable, plugin should output the state and should be able to filter scan results based on state if available in input

Qualys Container Security Plugin

Overview

A discovery plugin that identifies cryptographic assets (secrets, certificates, API keys, and credentials) within container images through the Qualys Container Security API.

This plugin discovers secrets in container images by querying the Qualys Container Security API. It returns detected secrets with:

- **Unique identifiers (URNs)** - Content-based fingerprints combining secret type, source layer, file location, and line number
- **Extrinsic locators (URLs)** - Registry-aware URLs pointing to the container image or local image location
- **Metadata** - Secret category, severity, detection rules, layer information, and timestamp information

Features

The plugin supports discovery across multiple container registries, including but not limited to:

- **Docker Hub** - Public and private Docker registries
- **Docker V2** - Docker Registry API v2 compatible registries
- **Docker V2-Private** - Docker V2 registries with private/authentication requirements
- **AWS ECR** - Amazon Elastic Container Registry
- **Azure Container Registry** - Azure container registry service
- **GitHub Container Registry** - GitHub Packages container registry
- **Google Artifact Registry** - Google Cloud artifact repositories
- **Google Cloud Registry** - Legacy Google Container Registry (GCR)
- **Harbor Container Registry** - Enterprise container registry
- **JFrog Artifactory Private** - JFrog Artifactory container registry
- **Red Hat Quay** - Red Hat Quay container registry

Tenable Security Center Plugin

Tenable Security Center (TSC) is an on-prem.-based vulnerability scanning and assessment service that continuously discovers, assesses, and reports on vulnerabilities across an organization IT infrastructure. This plugin discovers SSL/TLS certificates from Tenable Security Center by querying vulnerability scan results. It scans all hosts within the Tenable SC instance and extracts certificate details, including subject, validity dates, cipher suites, and associated services (such as HTTPS and SMTP).

The Tenable Security Center plugin integrates information retrieved by the following plugins.

Plugin ID	Description
10863	Retrieves SSL certificate information (primary source of certificate data)
10107	Identifies HTTP server type and version
10263	Identifies SMTP mail server information
21643	Enumerates the supported SSL Cipher Suites

Tenable Security Center plugin features are:

1. Authenticates to Tenable using API access and secret keys
2. Retrieves vulnerability data specifically related to SSL certificates (Plugin ID 10863),
3. Enriches the certificate information with service detection data (web server types/versions, SMTP servers) and cipher suite information captured by Tenable scanners.
4. Discovers SSL/TLS certificates from all scanned hosts in Tenable Security Center
5. Enriches certificate records with host operating system, HTTP server version, and SMTP service information
6. Collects cipher suite information per host
7. Deduplicates certificates by SHA-256 fingerprint across all hosts
8. Tracks scan timestamps in state for audit purposes

Microsoft Defender Plugin

This plugin integrates with Microsoft Defender for Endpoint to retrieve all SSL/TLS certificates discovered across enrolled Windows devices. For each certificate, it captures key metadata, including the subject, issuer, validity period, key usage, and the associated device.

Note: Microsoft Defender does not provide access to the raw certificate data. Therefore, this plugin records certificate attributes only and does not store the complete certificate content.

Before You Start

To configure access, ensure you have the following details from Microsoft Entra ID:

- **Tenant ID** – Identifies your organization within Entra ID
- **Client ID** – Identifies the application registered to access the API
- **Client Secret** – Acts as the authentication credential (password) for the application

---

Setting Up Microsoft Entra ID Access

Step 1: Register an Application

1. Sign in to the **Azure Portal**.
2. Navigate to **Microsoft Entra ID → App registrations → New registration**.
3. Provide a name for the application (for example, *Entrust-Discovery-MsDefender*), then click **Register**.
4. From the **Overview** page, copy the following values:
    - **Directory (tenant) ID**
    - **Application (client) ID**

Step 2: Create a Client Secret

1. Open the newly created app registration.
2. Go to **Certificates & Secrets → New client secret**.
3. Enter a description and select an expiry period.
4. Click **Add**.
5. Copy the **Value** immediately, as it is displayed only once.

Step 3: Grant API Permissions

1. Navigate to **API permissions → Add a permission → APIs my organization uses**.
2. Search for and select **WindowsDefenderATP**.
3. Choose **Application permissions** and enable:

- **Vulnerability.Read.All**

4. Click **Add permissions**.
5. Select **Grant admin consent** (this step requires Entra ID administrator privileges).

Once admin consent is successfully granted, the application is fully configured and ready for use.

Jenkins Script Plugin

Introduction

This plugin connects to your Jenkins server through the Jenkins Script Console (Groovy) and discovers credential metadata stored withinJenkins, including TLS certificates, SSH keys, secret tokens, passwords, and secret files.

For PKCS#12 keystore credentials and PEM-encoded secret file credentials, the plugin extracts the public certificate PEM and reportsit as a certificate asset. All other supported credential types, including those provided by more than 80 third-party Jenkins plugins, arereported as metadata-only assets. In these cases, the plugin records the credential's existence, type, and location without inspecting orexposing its contents. Any unrecognized credential types are reported as secret metadata assets.

> *Note: The plugin accesses decrypted credential content within the Jenkins controller process for classification purposes, such as opening keystores or reading file contents to identify PEM-encoded certificates. However, it never exports private keys,passwords, secret values, or other sensitive data. Only public certificate information and non-sensitive metadata are includedin the scan results.*

Which Jenkins Discovery Plugin Is This?

The platform provides two Jenkins discovery plugins:

Jenkins Script Plugin (`jenkins-script-plugin`)

This plugin executes a Groovy script on the Jenkins controller through the Jenkins Script Console. It can inspect supported credential types, including PKCS#12 keystores and PEM-encoded secret files, to extract and report public certificate PEM data as certificate assets. Othercredential types are discovered and reported as metadata-only assets. Because it runs directly on the Jenkins controller, it requires theOverall/ Administer permission.

Jenkins REST Plugin (`jenkins-rest-plugin`)

This plugin connects to Jenkins remotely using a username and API token over HTTP or HTTPS. It retrieves credential metadata onlyand classifies credentials based on their Jenkins credential type without inspecting their underlying content. For example:

- Credentials of the **Certificate** type are reported as key assets.
- Certificates uploaded as **Secret File** credentials are reported as secret assets.
- Certificate content and PEM data are not accessible through the Jenkins REST API and therefore cannot be extracted.

Veeam Backup and Replication Plugin

This plugin integrates with the Veeam Backup and Replication (BR) API, which is part of a Veeam BR server/ installation typically running on a enterprise-local virtual machine.

Note: The Veeam BR API does not provide access to the raw certificate. Therefore, this plugin records certificate attributes only and does not store the complete certificate content.

Before You Start

You need four pieces of information:

1. **Veeam REST API base URL** - this should usually be *https://<server/IP>:9419/api.* The 9419 port is a fixed value and cannot be changed in the current (July 2026) version of Veeam BR.
2. **API version** - as of this writing (July 2026), this should be set to "1.3-rev1"
3. **API administrator username** - User name for a user with the Veeam Backup Adminstrator role in your Veeam installation. For a Windows installation this is likely "Administrator". For a Linux installation, this is likely "veeamadmin"
4. **API administrator password** - Password for above user, likely set during Veeam installation.

You may also need the PEM certificate for your Veeam server or issuing CA if the API is protected by a self-signed or privately issued TLS certificate.

Required Access

The account used by the plugin must be able to authenticate to the Veeam REST API and read the resources queried by the plugin.

At minimum, the account must be able to access:

- `/oauth2/token`
- `/v1/serverCertificate`
- `/v1/kmsServers`
- `/v1/credentials`
- `/v1/encryptionPasswords`

If the account can log in but lacks permission to one of the scan endpoints, the scan will fail with an API error. This is why a user with Administrator role is required input - an Administrator is authorized for any endpoint.

Nutanix Prism Element Discovery Plugin

Introduction

Nutanix Prism Element (PE) is the cluster-level management interface for Nutanix AHV/AOS environments. It provides the legacy PrismGateway REST APIs (v1 and v2.0), which are used to manage SSL/TLS certificates, Data-at-Rest Encryption (DARE), key management servers, SSH public keys, and SNMPv3 credentials for an individual cluster.

The Nutanix Prism Element Discovery Plugin connects directly to a single Prism Element endpoint to discover and catalog cryptographic assets within that cluster. It identifies and reports certificates, certificate authority

(CA) chains, cryptographic keys, and security-sensitiveconfigurations, providing comprehensive visibility into the cluster's cryptographic posture.

Features

The plugin performs 15 discovery routines to identify and inventory cryptographic assets across the Prism Element cluster.

Certificates and Keys

- **Live TLS Certificate Capture**: Establishes a TLS handshake with the Prism Element endpoint and captures the presented leafcertificate.
- **CA Chain Discovery**: Retrieves the cluster CA certificate chain through `/v1/keys/ca_chain`, with `/v1/keys/pem` used as a metadatafallback.
- **File Server SSL Certificates**: Discovers SSL certificates configured for Nutanix Files file servers (vFilers) through `/v1/vfilers/{id}/ssl_certificate`.
- **Cluster SSH Public Keys**: Enumerates SSH public keys configured on the cluster.
- **Client Authentication (mTLS) CA Chain**: Reports the CA configuration used for mutual TLS client authentication.

Encryption and Security

- **Data-at-Rest Encryption (DARE)**: Discovers DARE CA certificates, retrieves pending DARE CSRs, and reports encryption-related cluster information, including disk and container counts.
- **Encryption Status**: Reports encryption configuration details and the number of enabled and disabled encryption instances.

Key and Authentication Management

- **Key Management Servers (KMS)**: Identifies configured Key Management Server entries and reports associated metadata.
- **SNMPv3 Keys**: Discovers SNMPv3 users configured with authentication and privacy keys.
- **Authentication Configuration**: Reports authentication settings, supported authentication methods, and configured LDAP directoryentries.
- **Remote Sites**: Discovers remote site and disaster recovery (DR) configurations.

Configuration and Monitoring

- **HTTP Proxy Configuration**: Reports configured HTTP proxies and associated CA certificates.
- **Syslog Configuration**: Discovers TLS certificate settings used for secure syslog forwarding.

Output Format

The plugin uses a shared formatting framework (`common/formatter` and `common/plugincli`) to generate consistent, standards-compliantasset records. Certificate data is exported in a normalized format, with `cert_pem` values stored without embedded newline characters. Certificate metadata is derived directly from the PEM content and is not duplicated in the additional attributes section.

Nutanix Prism Central Discovery Plugin

Introduction

Nutanix Prism Central (PC) is the multi-cluster management plane for a Nutanix environment. It exposes the modern v3 intent APIs and the v4 namespaced APIs (Cluster Management, Security) that manage certificates, certificate authorities, certificate signing requests, key management servers, credentials, and SNMPv3 configuration across every registered cluster.

The Nutanix Prism Central Discovery Plugin connects to a single Prism Central endpoint, automatically discovers all clusters it manages, and catalogs the cryptographic assets exposed through the v3/v4 APIs — across every cluster plus the Prism Central-scoped security configuration.

> *For direct single-cluster discovery through Prism Element (v1/v2 APIs), use the companion nutanix-pe-plugin instead.*

Features

- **Automatic multi-cluster discovery** — lists all managed clusters via the v4 Cluster Management API, with a v3 intent-API fallback, and scans each cluster.
- **Live TLS certificate capture** — performs a TLS handshake on the Prism Central URL (always) and per-cluster external IPs (when available). Clusters without an external IP are skipped for TLS probing.
- **SSL certificate + CA chain** — retrieves each cluster's SSL certificate and CA chain via the v4 Cluster Management API.
- **Managed certificates** — enumerates managed certificates per cluster (paginated).
- **Certificate authorities** — enumerates per-cluster certificate authorities.
- **Certificate signing requests** — lists pending CSRs per cluster and reports metadata (common name, purpose, PEM availability). CSR material is downloaded to confirm PEM availability but only metadata is emitted.
- **SNMPv3 keys** — reports SNMPv3 users carrying authentication/privacy keys via the v4.2 Cluster Management API. Key values are emitted as separate secret assets (auth-key and priv-key) when present; users with no keys are skipped.
- **Key Management Servers (Security v4.1)** — reports KMIP client and CA certificates (from `certPem` / `caPem`) and cloud KMS configuration with a key identifier.
- **Credentials (Security v4.1)** — reports externally-stored credential configurations that reference a target URL or address. Entries with only administrative metadata (name, validity) are omitted since credential material is write-only server-side.
- **Shared formatter library** — all output is produced through the shared `common/formatter` and `common/plugincli` packages for standards-compliant, consistent asset records (`cert_pem` without embedded newlines, PEM-intrinsic fields derived from the certificate, not duplicated in `extra`).