Using and Managing SaltStack SecOps

October 2021

VMware vRealize Automation SaltStack Config 8.6

vmware®

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

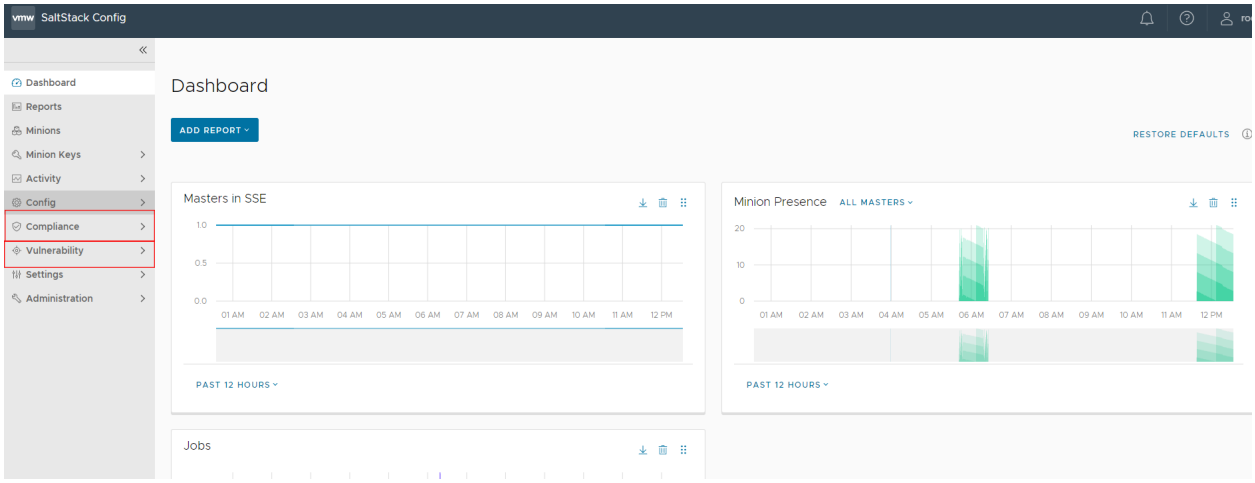
VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

What is SaltStack SecOps

1

SaltStack SecOps is a compliance and vulnerability management application that can automate security remediation.



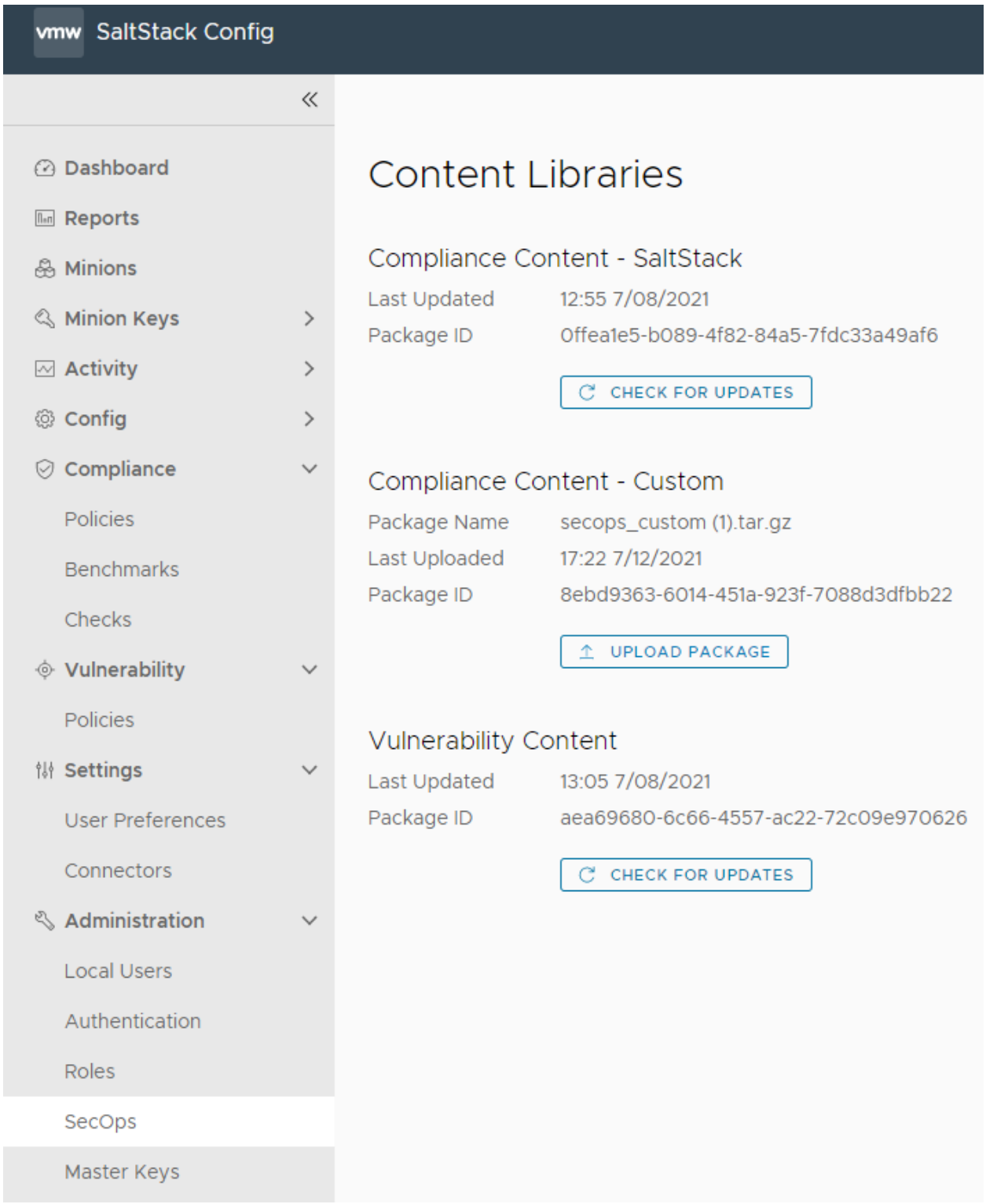
SaltStack SecOps allows you to scan your system for compliance against various security benchmarks, detect system vulnerabilities, and remediate your results. SaltStack SecOps includes both SaltStack SecOps Compliance and SaltStack SecOps Vulnerability services.

SaltStack SecOps Compliance scans your system for compliance with supported security benchmarks from accredited institutations (such as CentOS Linux Level 1 and 2 Senior and Workstation) and allows you to remediate nodes that are not in compliance. It includes two compliance libraries:

- Compliance Content Library - Built-in security content
- Compliance Content Custom Library - Custom checks and benchmarks defined and uploaded by your organization.

SaltStack SecOps Vulnerability scans your system for common vulnerabilities and exposures (CVEs), and then remediates any identified advisories. It includes a built-in advisories vulnerability library.

Content libraries are updated regularly as security standards change. You can also configure content to download automatically as updates become available, or download content manually. To manually check for updates and download content, click **Administration > Secops >** and click either **Check for Updates** or **Upload Package** underneath the desired content library.



Note Before you can use SaltStack SecOps, you must have SaltStack Config installed and configured. However, SaltStack SecOps requires a separate product license from SaltStack Config. Contact a sales representative for more information.

This chapter includes the following topics:

Prerequisites

Before you can use SaltStack SecOps you must ensure these prerequisites are met.

Apply a DLF license

When installing SaltStack SecOps without using vRealize Suite Lifecycle Manager, you must apply a DLF license in addition to your SaltStack SecOps license.

1 Create a file named vra_license in `/etc/raas`.

2 Edit the `/etc/rass/vra_license` file and add your SecOps key.

3 Save the vra_license file.

4 Run `chown rass:raas vra_license`

5 Run `systemct1 restart raas`.

Configure SaltStack SecOps

2

vRealize Automation SaltStack SecOps is add-on for SaltStack Config that provides two security libraries. Both content libraries update regularly as security standards change. You can configure content to download (or ingest) automatically as security standards change, which is recommended for most standard systems.

The following types of content are provided as part of SaltStack SecOps:

- **Compliance** - Automated compliance detection and remediation for your infrastructure. The compliance content library consists of industry best-practice security and compliance content, such as CIS.
- **Vulnerability** - Manages vulnerabilities on all the systems in your environment. Its content library includes advisories based on the latest Common Vulnerabilities and Exposures (CVE) entries.

As an alternative, the library includes the option to download content manually, or to access content from the RaaS node via an HTTP(s) proxy. Manual ingestion is useful for air-gapped systems, while downloading via proxy is useful to avoid downloading content directly from the internet. Downloading via proxy also provides more control and visibility into what's being downloaded and where.

Before you start

Configuring SaltStack SecOps is one post-installation step in a series of several steps that should be followed in a specific order. First, complete one of the installation scenarios and then read the following post-installation pages:

- Install the license key
- Install and configure the Master Plugin
- Check the RaaS configuration file
- Log in for the first time and change default credentials
- Accept the Salt master key and back up data
- Set up SSL certificates

Install Python 3 rpm libraries

SaltStack SecOps uses the Python 3 rpm libraries to reliably compare package versions. These programs need the increased accuracy provided by these libraries to determine version compliance or assess vulnerabilities.

Currently, any minions using RedHat or CentOS 7 might need the Python 3 rpm libraries in order to run accurate compliance or vulnerability assessments. If you intend to run assessments on minions that use these versions of RedHat or CentOS, you need to manually install the Python 3 rpm library on these machines.

Note Other workarounds are available. If you need an alternate workaround, Contact Support.

To install the Python 3 rpm library on the Salt master running the Master Plugin:

1 Install the EPEL repository using the following command:

```
yum install -y epel-release
```

2 Install the Python 3 rpm library:

```
yum install -y python3-rpm
```

Automatic content ingestion for standard systems

For non-air-gapped RaaS systems, content is downloaded and ingested on a periodic basis as determined by the settings in the configuration file. By default, automatic content ingestion is already configured in SaltStack Config and no further action is required.

If you installed SaltStack Config manually, follow these steps to configure automatic SaltStack SecOps content ingestion:

1 Add the following to the RaaS service configuration file `/etc/raas/raas` in the `sec` section, adapting it as necessary:

```
sec:
  stats_snapshot_interval: 3600
  username: secops
  content_url: https://enterprise.saltstack.com/secops_downloads
  ingest_saltstack_override: true
  ingest_custom_override: true
  locke_dir: locke
  post_ingest_cleanup: true
  download_enabled: true
  download_frequency: 86400
  compile_stats_interval: 10
  archive_interval: 300
  old_policy_file_lifespan: 2
```

```
  delete_old_policy_files_interval: 86400
  ingest_on_boot: true
  content_lock_timeout: 60
  content_lock_block_timeout: 120
```

2 Save the file.

3 Restart the RaaS service:

```
systemctl restart raas
```

After the service restarts, SaltStack SecOps content begins to download. This may take up to five minutes, depending on your internet connection.

Ingesting content via http(s) proxy

For ingestion via proxy, you'll need to create an override to the RaaS service and add new environment variables for `httpproxy` and `httpsproxy`.

To configure the RaaS node to use https proxy:

1 Complete the previous steps to enable automatic ingestion.

2 On the master in the command line, edit the RaaS service:

```
systemctl edit raas
```

3 Add the following lines to the generated file.

```
[Service]
Environment="http_proxy=http://<hostname>:234"
Environment="https_proxy=https://<hostname>:234"
Environment="HTTP_PROXY=http://<hostname>:234"
Environment="HTTPS_PROXY=http://<hostname>:234"
```

4 If your proxy requires password authentication, you may need to set this as part of the proxy environment variables. For example:

```
Environment="HTTP_PROXY=http://USER:PASSWORD@<hostname>:234"
```

5 If your proxy uses an internal Certificate Authority, you may also need to set the `REQUESTS_CA_BUNDLE` environment variable to ensure that the proxy is able to use it. For example:

```
Environment="REQUESTS_CA_BUNDLE=/etc/pki/tls/certs/ca-bundle.crt"
```

6 Restart the RaaS service:

```
systemctl restart raas
```

After the service restarts, content begins to download. This may take up to 20 minutes.

Manual content ingestion for SaltStack SecOps Compliance

If your environment is air-gapped, which means it cannot connect to an external site to download updates, you must manually update SaltStack SecOps Compliance content by downloading the tarball from Customer Connect and transferring it to your RaaS node.

Also, if your system is air-gapped, change the download configuration setting in the RaaS configuration file to False:

```
sec:
  download_enabled: False
```

The RaaS configuration file is located in `/etc/raas/raas`. You might also need to restart the RaaS service after applying these configuration settings:

```
systemctl restart raas
```

To manually ingest the SaltStack SecOps Compliance tarball:

1 Download the SaltStack SecOps Compliance content.

2 Log in to an RaaS node.

3 Copy the compliance content tarball to the RaaS node in the `tmp` folder.

   This content could be delivered by email or any other means.

4 Ingest the tarball contents.

   ```
   su - raas -c "raas ingest /path/to/locke.tar.gz.e"
   ```

   This returns:

   ```
   Extracting: /tmp/locke.tar.gz -> /tmp/extracted-1551290468.5497127

   Cleaning up: /tmp/extracted-1551290468.5497127

   Results:

   {'errors': [], 'success': True}
   ```

Manual content ingestion for SaltStack SecOps Vulnerability

If your environment is air-gapped, which means it cannot connect to an external site to download updates, you must manually update SaltStack SecOps Vulnerability content by downloading the tarball from Customer Connect and transferring it to your RaaS node.

Also, if your system is air-gapped, change the download configuration setting in the RaaS configuration file to False:

```
sec:
  download_enabled: False
```

The RaaS configuration file is located in `/etc/raas/raas`. You might also need to restart the RaaS service after applying these configuration settings:

```
systemctl restart raas
```

To manually ingest the SaltStack SecOps Vulnerability tarball:

1 Download the SaltStack SecOps Vulnerability content.

2 Log in to an RaaS node.

3 Copy the vulnerability content tarball to the RaaS node in the `tmp` folder.

   This content could be delivered by email or any other means.

4 Ingest the tarball contents, replacing the name of the tarball in this command with the exact file name of the tarball:

   ```
   su - raas -c "raas vman_ingest /tmp/vman_date_example123.tar.gz.e"
   ```

   This returns:

   ```
   'adv': {'error': 0, 'success': 60334},
     'adv_cve_xref': {'error': 0, 'success': 243781},
     'cve': {'error': 0, 'success': 162251},
     'pkgfile': {'error': 0, 'success': 42},
     'py': {'error': 0, 'success': 7},
     'sls': {'error': 0, 'success': 3}
   ```

Troubleshooting manual ingestion

If you try running the manual ingestion commands for either SaltStack SecOps Compliance or SaltStack SecOps Vulnerability content, you might see an error message similar to this message:

```
/home/centos/locke_date_example123.tar.gz.e not found or not readable
```

This error message sometimes appears if you do not place the tarball in the `tmp` folder. Placing the tarball in the `tmp` folder resolves the issue.

Set up Splunk integration

SaltStack Config integrates the vulnerability library with Splunk to help you optimize and secure your digital infrastructure using the SaltStack Config Add-On for Splunk Enterprise. The add-on is available on Splunkbase, and requires SaltStack Config version 6.3 or higher.

The SaltStack Config add-on in Splunk takes advantage of a Prometheus-compatible metrics endpoint which reports over 25 unique SaltStack Config metrics. These metrics provide insight into the health of your infrastructure. Accessing them in Splunk is useful for monitoring for outages, identifying abnormal activity, and more. It also gives you the ability to take automated actions based on a specific Splunk event using SaltStack Config.

For instructions on how to install and configure the add-on, see the full add-on documentation in the VMware knowledge base.

For more on the SaltStack Config metrics endpoint, see the product documentation forSaltStack SecOps.

Configuration options

The following table describes the configuration options available for compliance content:

Option	Description
`stats_snapshot_interval`	How often (in seconds) SaltStack SecOps Compliance stats will be collected
`compile_stats_interval`	How often (in seconds) SaltStack SecOps Compliance stats will be compiled
`username`	Username to use when connecting to SaltStack Config to download the most recent SSaltStack SecOps Compliance content (default: `secops`)
`content_url`	URL used to download SaltStack SecOps Compliance content
`ingest_override`	When ingesting new content, overwrite existing benchmarks and checks (default: `True`)
`locke_dir`	Path where ingestion expects to find new content (default: `locke`). If you use a relative path (no leading `/`), then it is relative to the RaaS service cache dir `/var/lib/raas/cache`
`post_ingest_cleanup`	Remove the expanded content from the file system after ingestion (default: `True`)
`download_enabled`	Whether SaltStack SecOps Compliance content downloads are allowed (default: `True`). Set this to `False` for air gapped systems.
`download_frequency`	How often in seconds will the RaaS service attempt to download SaltStack SecOps Compliance content (default: `86400` for 24 hours)
`ingest_on_boot`	Should the RaaS service attempt to download SaltStack SecOps Compliance content on boot? (default: `True`)
`content_lock_timeout`	How long in seconds will content download locks last (default: `60`)
`content_lock_block_timeout`	How long in seconds will content download locks block before failing (default: `120`)

The following table describes the configuration options that are available for vulnerability content:

Option	Description
`vman_dir`	Location where SaltStack SecOps Vulnerability content is expanded before ingestion. If the path is relative (no leading `/`), then it is relative to the RaaS service cache dir `/var/lib/raas/cache`
`download_enabled`	If `True`, SaltStack SecOps Vulnerability content downloading is enabled. Set to `False` for air gapped systems
`download_frequency`	The frequency in seconds of automated SaltStack SecOps Vulnerability content downloads and ingestion
`username`	Username used to log in to `enterprise.saltstack.com` to get content
`content_url`	URL from which SaltStack SecOps Vulnerability content will be downloaded
`ingest_on_boot`	If `True`, SaltStack SecOps Vulnerability content will be downloaded and ingested soon after the RaaS service boots (default: `True`)
`compile_stats_interval`	How often (in seconds) SaltStack SecOps Vulnerability stats will be compiled
`stats_snapshot_interval`	How often (in seconds) SaltStack SecOps Vulnerability stats will be collected
`old_policy_file_lifespan`	Lifespan (in days) of old policy files that will remain in the RaaS file system
`delete_old_policy_files_interval`	How often (in seconds) old SaltStack SecOps Vulnerability policy files will be deleted from the RaaS file system
`tenable_asset_import_enabled`	If `True`, minion grains in SaltStack Config will be sent to Tenable.io for matching assets (default: `True`)
`tenable_asset_import_grains`	List of minion grains to send to Tenable.io, if tenable asset import is enabled. SaltStack SecOps Vulnerability supports only `fqdn`, `ipv4`, `ipv6`, and `hostname` out of the box, however you can send other information by defining custom grains. For more on grains, including how to write custom grains, see Salt documentation: Grains. If you have only a subset keys in your grains data, only those in the subset will be synced. `fqdn` and `ipv4` will be sent even if you do not list them here. For more information, see the Tenable Import assets documentation.

FAQ

- **Q: How often is new SaltStack Vulnerability content released?**
  - A: The current release frequency is about once per quarter. However, content might be released more frequently in the future.
- **Can I get access to new content sooner if I use automatic content ingestion instead of manual ingestion?**
  - A: The same content is available, whether you ingest manually or automatically.

    However, if you use manual ingestion, you need to plan to check for security content updates and develop a process to manually ingest updated content when it is available.

What to do next

After configuring SaltStack SecOps, there may be additional post-installation steps. Check the list of post-installation steps to ensure you have completed all the necessary steps.

Supported Security and Compliance Benchmarks

3

SaltStack SecOps supports a variety of security and compliance benchmarks.

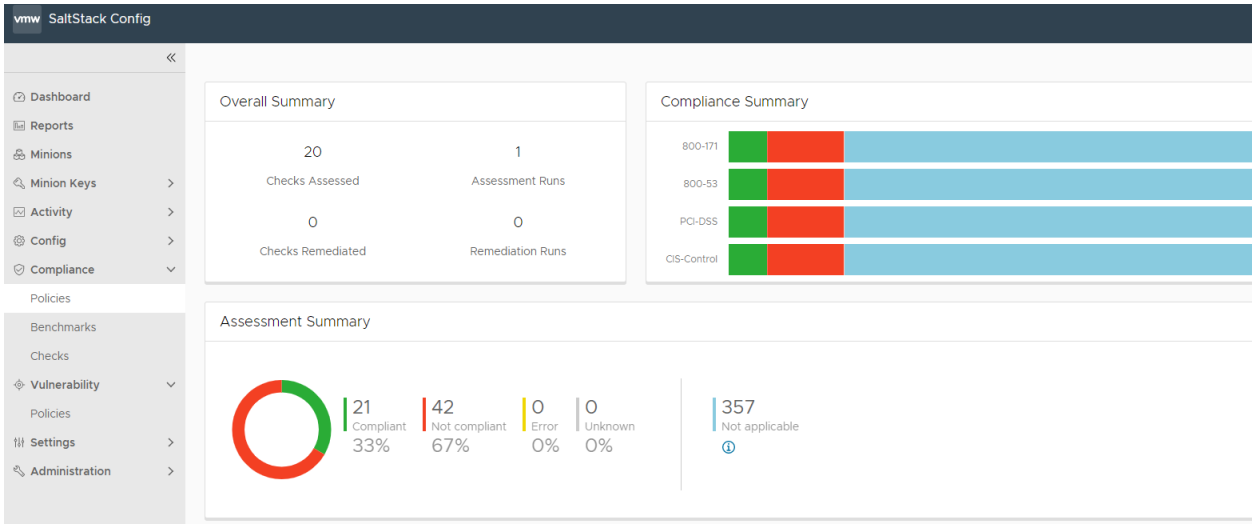
To view a list of supported security and compliance benchmarks and instructions to subsribe to future updates, see

Supported Security and Compliance Benchmarks.

How do I use SaltStack SecOps Compliance

4

SaltStack SecOps Compliance is an IT infrastructure security and compliance solution that combines security with IT operations all in a single platform.



SaltStack SecOps Compliance harnesses SaltStack Config's powerful configuration management and remote execution capabilities to bring all of your infrastructure assets to compliance with a range of industry-wide security benchmarks. It integrates with SaltStack Config to apply security measures at scale, while respecting custom exemptions based on your organization's needs.

The built-in compliance library contains the latest security standards based on industry best practice hardening guides. SaltStack SecOps Compliance uses the library to assess your infrastructure security and remediate noncompliant systems instantly. Occasionally, you might need to update the security library and download content. You can update and download content to the security library by clicking **Administration > SecOps** on the side menu, and then selecting **Compliance Content - SaltStack > Check for updates**.

To use SaltStack SecOps Compliance, you first define a compliance policy, then scan systems against the policy. The scan detects non-compliant systems and allows you to remediate issues instantly. Additionally, you can enter exemptions and specify user permissions to ensure all paths to remediation are customized for your organization's needs.

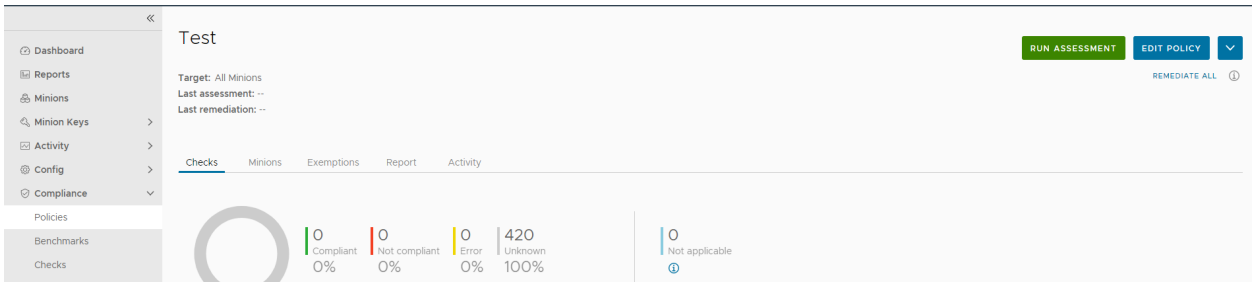
This chapter includes the following topics:

- How do I create a compliance policy

- How do I run a compliance assessment
- How do I view and remediate my assessment results

How do I create a compliance policy

To secure your infrastructure assets with SaltStack SecOps Compliance, you must start by defining policies.



SaltStack SecOps Compliance provides different industry benchmarks to choose from including checks for Center for Internet Security (CIS) and more. Each benchmark includes a collection of security checks. You can choose to apply all available checks for a given benchmark, or use only a subset of available checks. Using a subset of checks is useful for customizing SaltStack SecOps Compliance for your unique infrastructure needs, for example if remediating a given check poses the risk of breaking a known dependency.

When creating your policy, you must select a target to apply the policy to, along with which benchmarks and checks to run against your system.

To connect to the SDK directly, see vRealize Automation SaltStack Config SecOps.

Target

A target is the group of minions, across one or many Salt masters, that a job's Salt command applies to. A Salt master is managed similarly to a minion and can be a target if it is running the minion service. When creating a policy and selecting a target, you are defining the nodes that the security checks are run against. You can choose an existing target or create a new one.

For more information, see Minions.

Benchmarks

SaltStack SecOps Compliance simplifies the process of defining your security policy by grouping security checks by benchmark.

Benchmarks are category of security checks. SaltStack SecOps Compliance benchmarks are defined by widely-accepted experts, while custom benchmarks are defined by your own organization's standards. You can use benchmarks to help create a range of different policies optimized for different groups of nodes. For example, you might create an Oracle Linux policy that applies CIS checks to your Oracle Linux minions, and a Windows policy that applies CIS checks to your Windows minions. For more information on creating custom content, see Creating and testing custom compliance components.

Note Specifically for Windows Server benchmarks, the CIS content for certain benchmarks (notated with a tooltip ⓘ) is distributed across three different benchmarks:

- Domain master content
- Member content
- Domain master and member content

If you want to include all Member content, you must select both the benchmarks for Member and benchmarks for Domain master and member.

Checks

A check is a security standard that SaltStack SecOps Compliance assesses for compliance. The SaltStack SecOps Compliance library updates checks frequently as security standards change. In addition to checks included the SaltStack SecOps Compliance content library, you can create your own custom checks. Custom checks are indicated by a custom-checks-user-icon, instead of the built-in-checks-shield-icon. For more information on creating custom content, see Creating and testing custom compliance components. Each check includes several information fields.

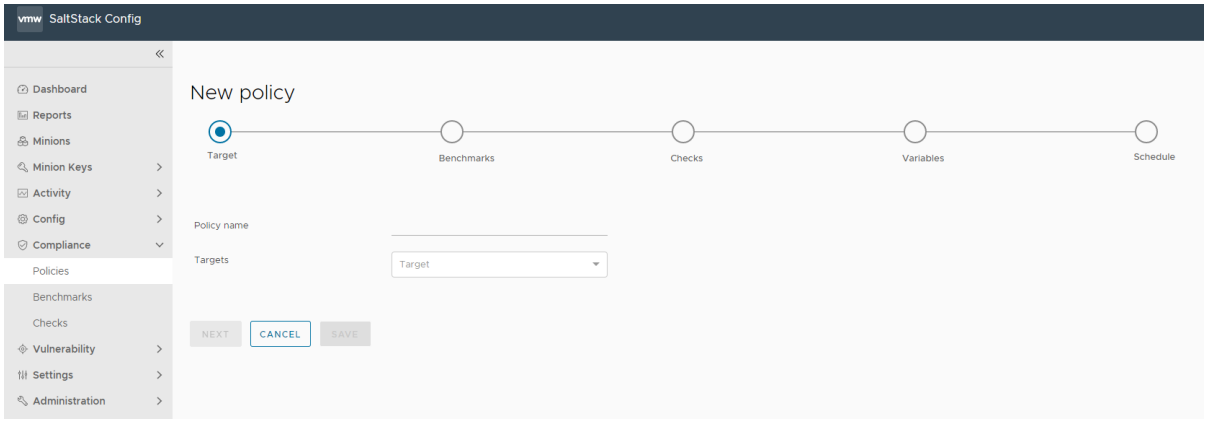
Information field	Description
Description	Description of the check.
Action	Description of the action that is performed during remediation.
Break	Used for internal testing only. For more information, contact your administrator.
Global Description	Detailed description of the check.
Osfinger	List of osfinger values that the check is implemented for. Osfinger is found in grains items for each minion to identify the minion's operating system and major release version. Grains are collected for the operating system, domain name, IP address, kernel, OS type, memory, and other system properties.
Profile	List of configuration profiles for different benchmarks.
Rationale	Description of rationale for implementing the check.
Refs	Compliance cross-references between benchmarks.

Information field	Description
Remediate	Values that indicate if SaltStack SecOps Compliance is capable of remediating noncompliant nodes, as not all checks include specific, actionable remediation steps.
Remediation	Description of how any noncompliant systems are remediated, if applicable.
Scored	CIS benchmark Scored value. Score recommendations impact the target's benchmark scroe, while recommendation that are not scored do not affect the score. True indicates scored and false indicates not scored.
State file	Copy of the Salt state that is applied to perform the check and if applicable, the subsequent remediation.
Variables	Variables in SaltStack SecOps Compliance used to pass values into the Salt states that make up security checks. For best results, use the default values.For more information, see How do I use Salt States.
Schedules	Select the schedule frequency from Recurring, Repeart Date & Time, Once, or Cron Expression. Additional options are available, depending on the scheduled activity, and on the schedule frequency you choose. ■ Recurring - set an interval for repeating the schedule, with optional fields for start or end date, splay, and maximum number of parallel jobs. ■ Repeat Date & Time - repeat the schedule weekly or daily, with optional fields for start or end date, and maximum number of parallel jobs. ■ Once - set a date and time to run the job once. ■ Cron - enter a cron expression to define a custom schedule based on Croniter syntax. See the CronTab Editor for syntax guidelines. Avoid scheduling jobs fewer than 60 secs apart when defining a custom cron expression. **Note** In the schedule editor, the terms “Job” and “Assessment” are used interchangeably. When you define a schedule for the policy, you are scheduling the assessment only - not the remediation. **Note** When defining an assessment schedule, you can choose the **Not Scheduled (on demand)** option. If you select this option, you choose to run a one-time assessment, and no schedule is defined.

Note You can exempt checks and minions from remediation by clicking **Add Exemption**, entering the reason for exemption and clicking **Add Exemption** again to confirm. Remediation is skipped for exempted items.

Procedure

1 On the SaltStack SecOps Compliance home page, click **Create Policy**.



2 Enter the policy name and select a target to apply the policy. Click **Next**.

3 On the Benchmarks tab, select all benchmarks you want to include in the policy and then click **Next**.

Note If no benchmarks are available, you might need to download compliance content. You can update and download content to the security library by clicking **Administration > SecOps** on the side menu, and then selecting **Compliance Content - SaltStack > Check for updates**.

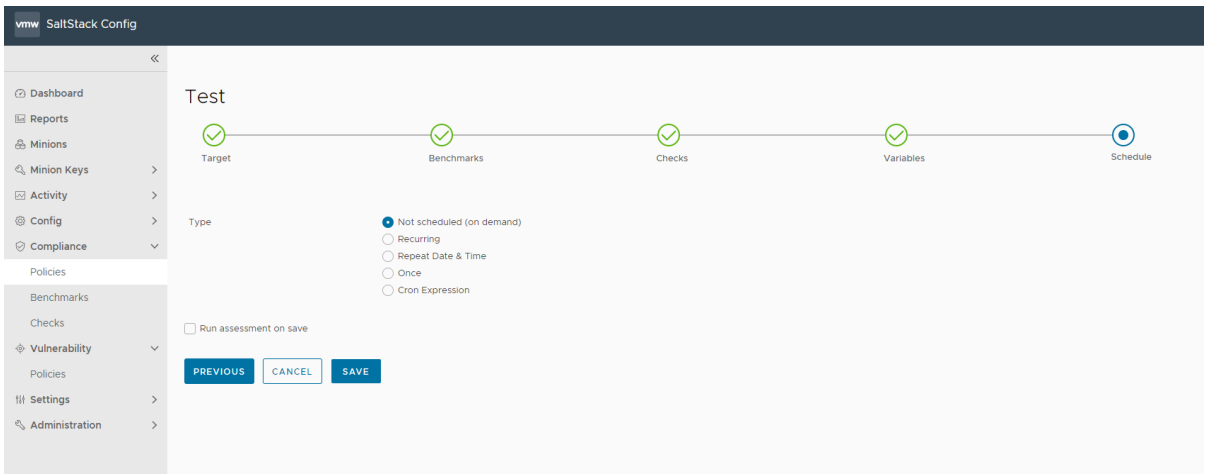
4 On the Checks tab, select all checks that you would like to include in the policy. The available checks are determined by the benchmarks you selected in step 3. Click **Next**.

5 On the Variables tab, enter or modify variables as needed. You can also choose to accept the default values (recommended). Click **Next**.

6 On the schedule page, define the schedule frequency and click **Save**.

7 (Optional) To run an assessment immediately after saving your policy, select **Run assessment on save**.

8 Click **Save**.



The policy is saved. If you selected **Run assessment on save**, the assesment is run immediately after saving.

Results

The compliance policy is saved and used to run an assessment. You can edit the policy by selecting the policy from the home page, and clicking **Edit Policy**.

Creating and testing custom compliance components

Custom Compliance content allows you to define your own security standards that supplement the library of security benchmarks and checks built into SaltStack SecOps Compliance.

Custom content is useful for enhancing SaltStack SecOps Compliance policies to fit your internal requirements.

SaltStack SecOps Compliance includes a Custom Content Software Development Kit (SDK) you can use to create, test, and build your own custom security content. You can import your custom security content to use alongside the SaltStack SecOps Compliance built-in security library for assessment and remediation. The ability to import custom content also allows you to version your content using a version control system of your choice, such as Git.

To use custom checks, you must first initialize the SaltStack SecOps Compliance Custom Content SDK. The SDK includes sample files you can modify to create your own custom checks, as well as benchmarks. The SDK also includes a Docker-based testing environment where you can test your new content.

Once your custom content is created and tested, you can build a content file and import it into SaltStack SecOps Compliance to begin assessing and remediating. Custom checks include a user icon custom-checks-user-icon , in contrast with SaltStack checks built-in-checks-shield-icon . SaltStack SecOps Compliance tracks dependencies between policies and your custom content, and provides a list of dependencies that might break if you delete the content.

Prerequisites

- Download the SaltStack SecOps Compliance Custom Content SDK.
- Install Docker. For more information, see Get Docker.

Procedure

1 From the command line, navigate to the directory containing the file and run the command:

Operating System	Command
Mac OS or Linux	`./secops_sdk init`
Windows	`secops_sdk.exe init`

No output is showed, which is expected. Your directory contains these folders and files:

- benchmarks - Contains custom benchmark meta (.meta) files

- salt/locke/custom - Contains custom check state (.sls) and meta (.meta) files
- sample_tests - Contains example files for testing using Docker
- README.md - Provides more detailed information about the SDK

2. (Optional) Commit changes to a version-controlled repository.
3. To create custom checks, in the Custom Content SDK, navigate to `salt/locke/custom`. To create custom benchmarks, skip to step 8.

   Note All custom checks must be configured in both a state (.sls) and corresponding meta (.meta) file.

4. Create a copy for both a sample state (.sls) and corresponding meta (.meta) file, and rename both with your desired custom name. Save both of these files together in any subdirectory of `salt/locke/custom`.

   Both files must be in the same directory and start with the same name, for example: `my_first_check.meta` and `my_first_check.sls`.

5. Edit the contents of the meta file to customize the check based on your needs.

   Note Check meta files contain references to different benchmarks. When creating custom content, ensure that you include all associated benchmarks in your check meta file.

6. Edit the contents of the state file.
7. Ensure both files are saved in the same directory.
8. To create custom benchmarks, in the Custom Content SDK, navigate to the `benchmarks` directory. This directory contains a sample benchmark meta (.meta) file.
9. Make a copy of `Sample_benchmark.meta`, and rename it with your desired custom name.
10. Edit the contents of the meta file to customize the benchmark based on your needs.

Results

Your custom checks and benchmarks are created. If needed, you can delete a custom check or benchmark by navigating to **SecOps > Checks** or **SecOps > Benchmarks**, clicking the menu icon ⋮ next to the custom content, and clicking **Delete**.

What to do next

After creating your custom content, you can test it by opening the command line, navigating to the Custom Content SDK `sample_tests` directory, and running these commands:

Command	Result
1. `./build.sh`	Builds a docker image of CentOS7 with Salt for testing.
2. `./up.sh`	Starts the testing container

Command	Result
3. `./test.sh salt-call --local state.apply locke.custom.mounts.my_first_check test=True`	Runs sample tests on checks you created in the `salt/locke/custom` directory. You can initiate custom checks as you would normal Salt states. For more information on Salt States, see How do I use Salt States.
4. `./down.sh`	After testing is complete, run this command to shut down the testing container.

After testing your custom content, you can Building your custom content library.

Building your custom content library

After creating your custom checks and benchmarks, you can create a custom content library.

Prerequisites

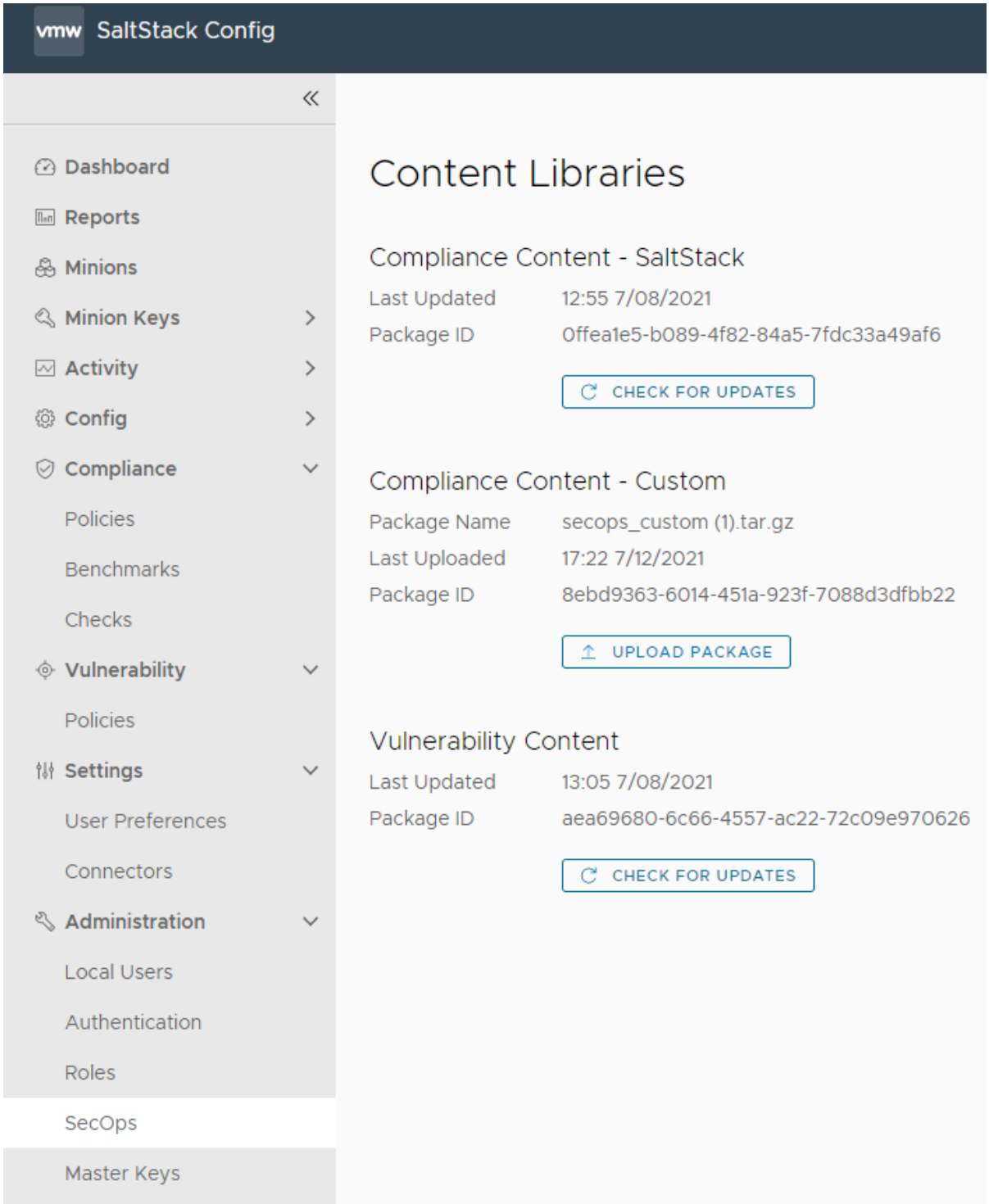
Creating and testing custom compliance components

Procedure

1. Open the command line and navigate to the Custom Content SDK root directory.
2. To build your custom content library, run the `./secops_sdk build -a` command.

   After running the command, the SDK root directory includes the `_dist` subdirectory and two `tar.gz` files you can use to import your custom content.
3. To import your content, click **Administration > SecOps** from the side menu.
4. Under **Compliance Content - SaltStack**, click **Check for updates**.

5 Click **Upload Package** and select the `tar.gz` file.



Note To make your custom checks easier to navigate, use the file containing a timestamp in the filename. You can also import content using the API (RaaS), or alternatively, through the command line during installation. See Sec API interface or the SaltStack Config Installation documentation.

Results

Your custom content is available in SaltStack SecOps Compliance for building policies, running assessments, and remediating your systems.

How do I run a compliance assessment

After creating a compliance policy, you can run a compliance assessment.

When running a compliance assessment, your system is scanned for compliance against the built-in Compliance Content library and (if applicable) the Compliance Custom Content library. These libraries contain checks and benchmarks and are updated regularly as security standards change. For more information on the custom content library, see Creating and testing custom compliance components.

After running your assessment, you can view your results and remediate any nodes that are out of compliance.

After running an assessment, the assessment results are identified and shown on the policy home page as:

- Compliant - setting is in its intended state compared to the standard or benchmark.
- Not compliant - setting is not in its intended state compared to the standard or benchmark. Further investigation and possible remediation are recommended.
- Not applicable - The setting is not applicable to this system. For example, if running a CentOS check on AIX.
- Unknown - Assessment or remediation has not been run.
- Error - SaltStack SecOps Compliance encountered an error while running the assessment or remediation.

Note Policies that include many checks might result in longer assessment processing time, which can delay other processes in SaltStack Config. It is recommended to plan for processing time before initiating an assessment.

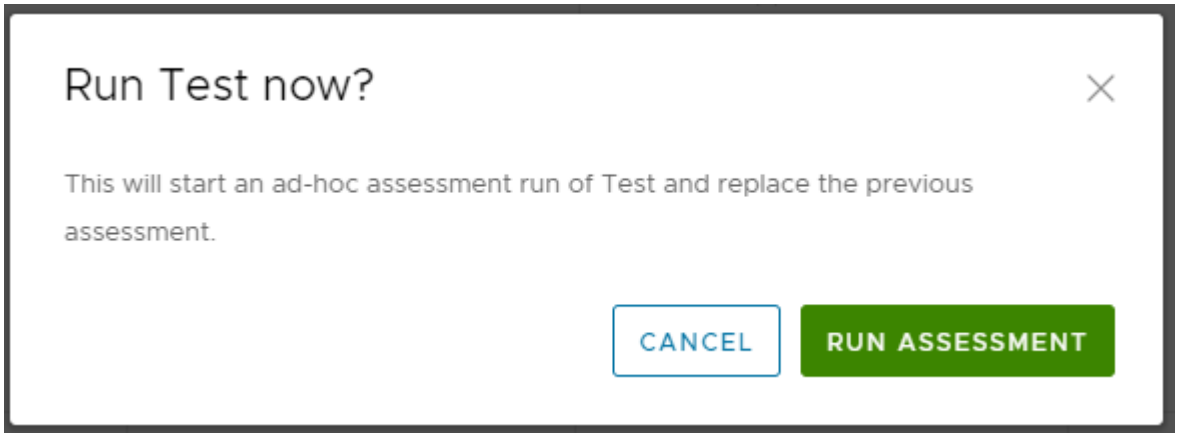
Prerequisites

Before you can run an assessment you must create a compliance policy. For more information, see How do I create a compliance policy.

Procedure

1. On the SaltStack SecOps Compliance home page, select a policy.

2 On the policy home page, click **Run assessment** and then **Run assessment** again on the confirmation window.



The Activity window opens. Completed assessments are listed in the Activity window along with their status, Job ID (JID), and other information.

3 After the assessment is finished, to view your assessment results, select the policy from the SaltStack SecOps Compliance home page.

The policy home page shows results from the most recent assessment, organized by check. You can filter the list, or select column headings to sort your results. To view assessment results by minion, select **Minion**.

Results

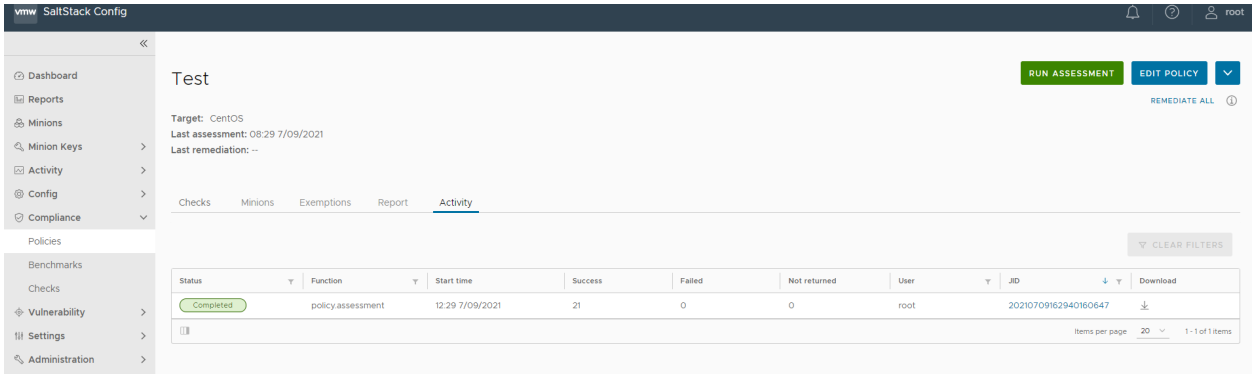
The assessment is complete and you can review your results. To download the assessment report, select the policy and click **Report tab > Download > JSON**.

What to do next

After reviewing your assessment results, you can How do I view and remediate my assessment results.

How do I view and remediate my assessment results

After running an assessment and reviewing your assessment results, you can remediate any non-compliant items.



On the policy home page, the Activity tab shows a list of completed or in-progress assessments and remediations and their statuses:

Status	Description
Queued	The operation is ready to run but the minions have not started the operation.
Completed	The operation is finished running.
Partial	The operation is still waiting for some minions to return, although the operation has finished running.

When non-compliant systems have been identified during a compliance assessment, you can remediate them to bring them into compliance. You can choose to remediate individual checks or nodes (minions), or you can alternatively remediate all checks on all nodes.

Note When you select to Remediate all, any exempted checks or nodes are not remediated.

You can add check and minion exemptions by selecting either the check from the policy home page, or the minion from the **Minions** tab and clicking **Add Exemption**. To remove an exemption, click the **Exemptions** tab, click the drop down arrow next to the exemption you want to remove, and click **Remove Exemption**.

Prerequisites

Before you can remediate your results, you must first run a compliance assessment. For more information, see How do I run a compliance assessment.

Procedure

1 On the SaltStack SecOps Compliance home page, select a policy to view the most recent assessment results.

2 You can choose to remediate all, remediate by check, or remediate by minion.

   a To remediate all: On the **Checks** tab, click **Remediate all**.

   b To remediate by check: Select a check to open the check description, scroll to the list of results from your last assessment, select all minions you want to remediate, and click **Remediate**.

   c To remediate by minion: On the policy home page, click the **Minions** tab, select a minion, and click **Remediate**.

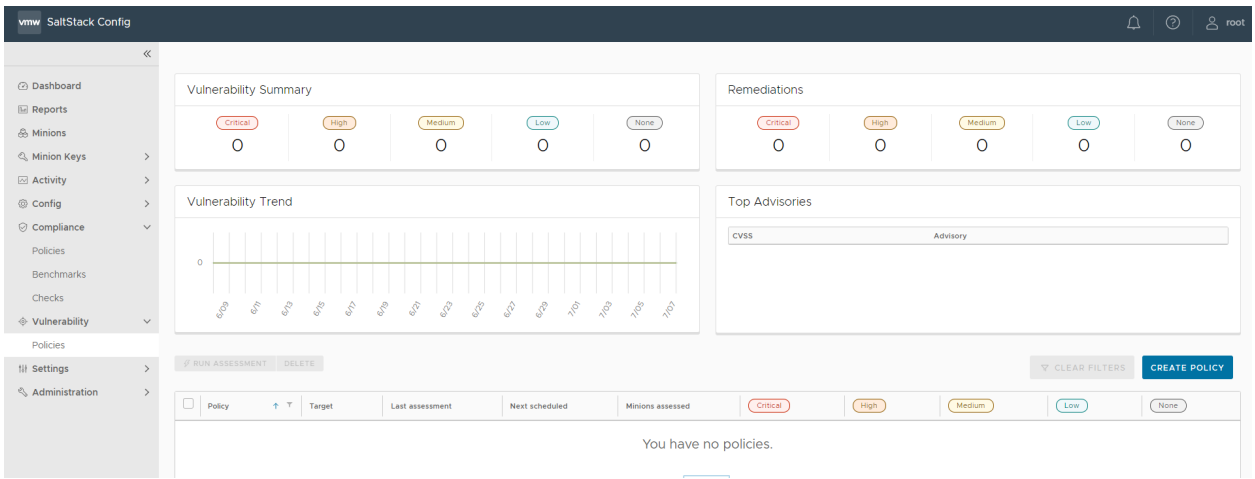
Results

Non-compliant results are remediated and made compliant.

How do I use SaltStack SecOps Vulnerability

5

SaltStack SecOps Vulnerability is a vulnerability remediation solution that allows Security and IT teams to work together to assess the vulnerability status of your systems against the latest security advisories, including those that reference Common Vulnerabilities and Exposures (CVE).



After scanning and detecting advisories, SaltStack SecOps Vulnerability can remediate any advisory that has an available repair package. You can optionally exempt certain advisories or assets to customize your vulnerability management strategy around other existing security controls.

SaltStack SecOps Vulnerability also supports importing security scans from third-party vendors, and remediating those advisories on impacted assets if a remediation is available. This currently includes imported scans from Tenable, Rapid7, Qualys, and Kenna Security, with a built-in API connector for importing from Tenable.io.

SaltStack SecOps Vulnerability provides various vulnerability reporting options including a quick, printable dashboard view to help assess your vulnerability trend over time.

Following a scan, you can access a downloadable list of all detected vulnerabilities, along with their corresponding advisory name, severity, vulnerability score, and affected assets. As a SaltStack Config add-on, SaltStack SecOps Vulnerability goes beyond assessment, and takes advantage of Salt to actively remediate vulnerabilities while also giving you full control over when and what to remediate.

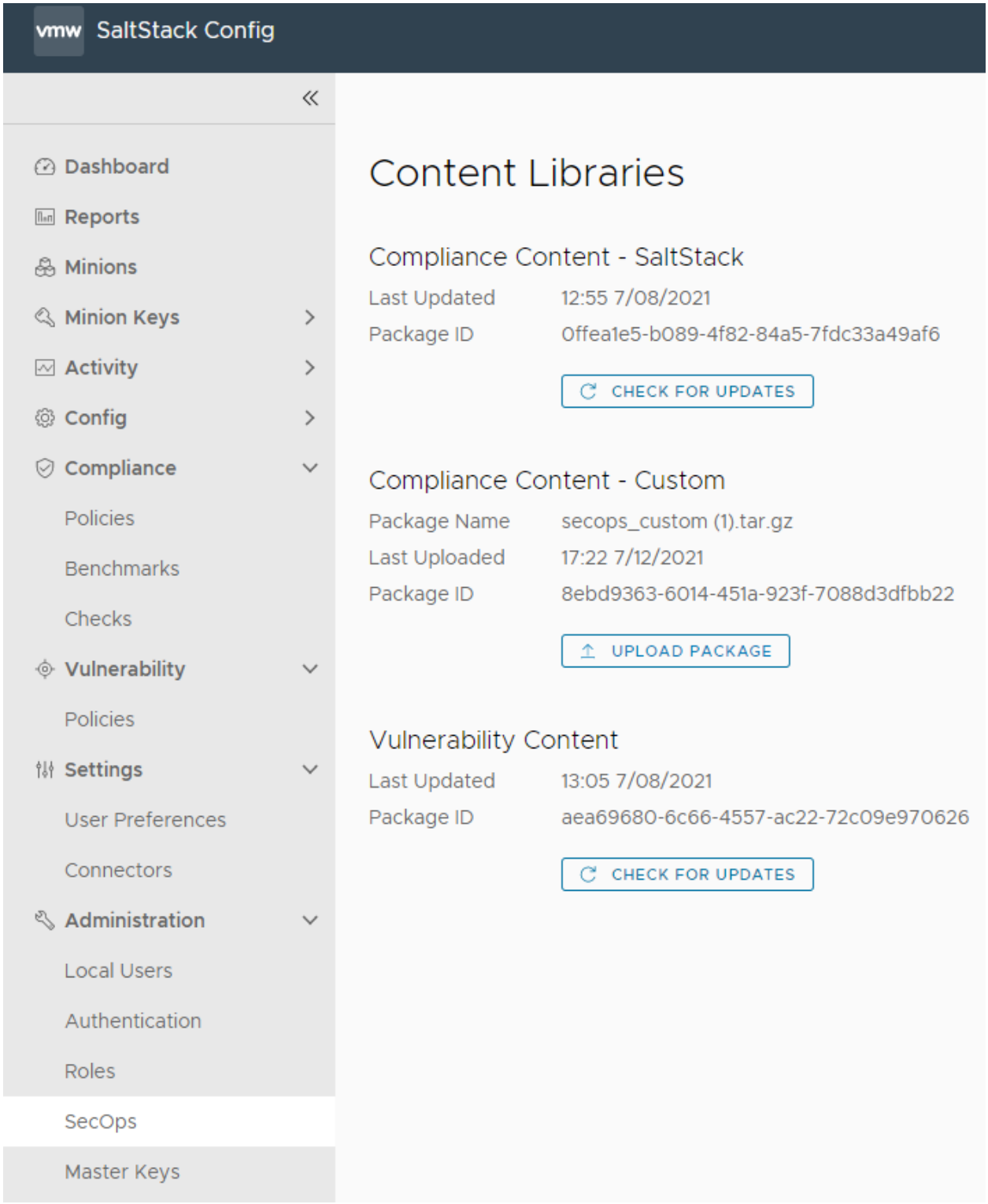
This chapter includes the following topics:

Using the vulnerability library

SaltStack Config uses an automated process to search for the latest security advisories along with the software packages or versions to fix nodes that are impacted by those vulnerabilities. This content is built and updated continuously in the vulnerability library.

When a new advisory or remediation is available, SaltStack Config bundles the library into a tarball and makes it available for download to SaltStack SecOps Vulnerability customers. By default, SaltStack Config checks for new content daily.

The vulnerability library tarball is encrypted before it is made available to SaltStack SecOps Vulnerability customers to ensure data integrity. A SaltStack SecOps Vulnerability license comes with the necessary keys to decrypt the tarball once it is downloaded. When SaltStack SecOps Vulnerability ingests a new tarball, it may take 15-20 minutes to get the latest content, which can impact performance. If you use the default process to update the vulnerability library, you should expect this performance delay the first time you download this content after installing and activating SaltStack SecOps Vulnerability. Then, when a new tarball is available for download, it is ingested and updated. However, you can experience the 15-20 minute delay during ingestion again, depending on the timing of the update. You can reduce the possibility of this delay by updating the vulnerability library manually. To update the content manually, click **Administration > SecOps**. Under **Vulnerability Content** click **Check for updates**.



Note The Package ID matches the UUID of the content tarball provided by SaltStack. However, if you notice a mismatch between the two IDs, this is because the tarball has been renamed. Check the name of the file you uploaded to ensure it still has the original filename provided by SaltStack. It might have been modified by a user, or by a computer.

How do I create a vulnerability policy

To begin using SaltStack SecOps Vulnerability, first create your security policy. In your policy, add the minions you want to target in an assessment and determine the assessment's run schedule.

A vulnerability policy is comprised of a target and an assessment schedule. The target determines which minions to include in an assessment and the schedule determines when assessments will be run. A security policy also stores the results of the most recent assessment in SaltStack SecOps Vulnerability. Policies can also include schedules, as well as specifications for handling exemptions.

Component	Description
Target	A target is the group of minions, across one or many Salt masters, that a job's Salt command applies to. A Salt master can also be managed like a minion and can be a target if it is running the minion service. When you choose a target in SaltStack SecOps Vulnerability, you define the group of assets (referred to as minions) your policy will apply to. You can choose an existing target or create a new one.
Schedule	Choose the schedule frequency from Recurring, Repeat Date & Time, Once, or Cron Expression. Additional options are available, depending on the scheduled activity, and on the schedule frequency you choose. ■ Recurring - Set an interval for repeating the schedule, with optional fields for start or end date, splay, and maximum number of parallel jobs. ■ Repeat Date and Time - Choose to repeat the schedule weekly or daily, with optional fields for start or end data, and maximum number of parallel jobs. ■ Once - Specify a date and time to run the job. ■ Cron - Enter a cron expression to define a custom schedule based on Croniter syntax. For best results, avoid scheduling jobs less than 60 seconds apart when defining a custom cron expression. For more information, see the Cron Editor for guidelines. **Note** In the schedule editor, the terms "Job" and "Assessment" are used interchangeably. When you define a schedule for the policy, you are scheduling the assessment only - not the remediation. ■ Not Scheduled (on demand) - Choose to run only single assessments as needed. No set schedule is defined.

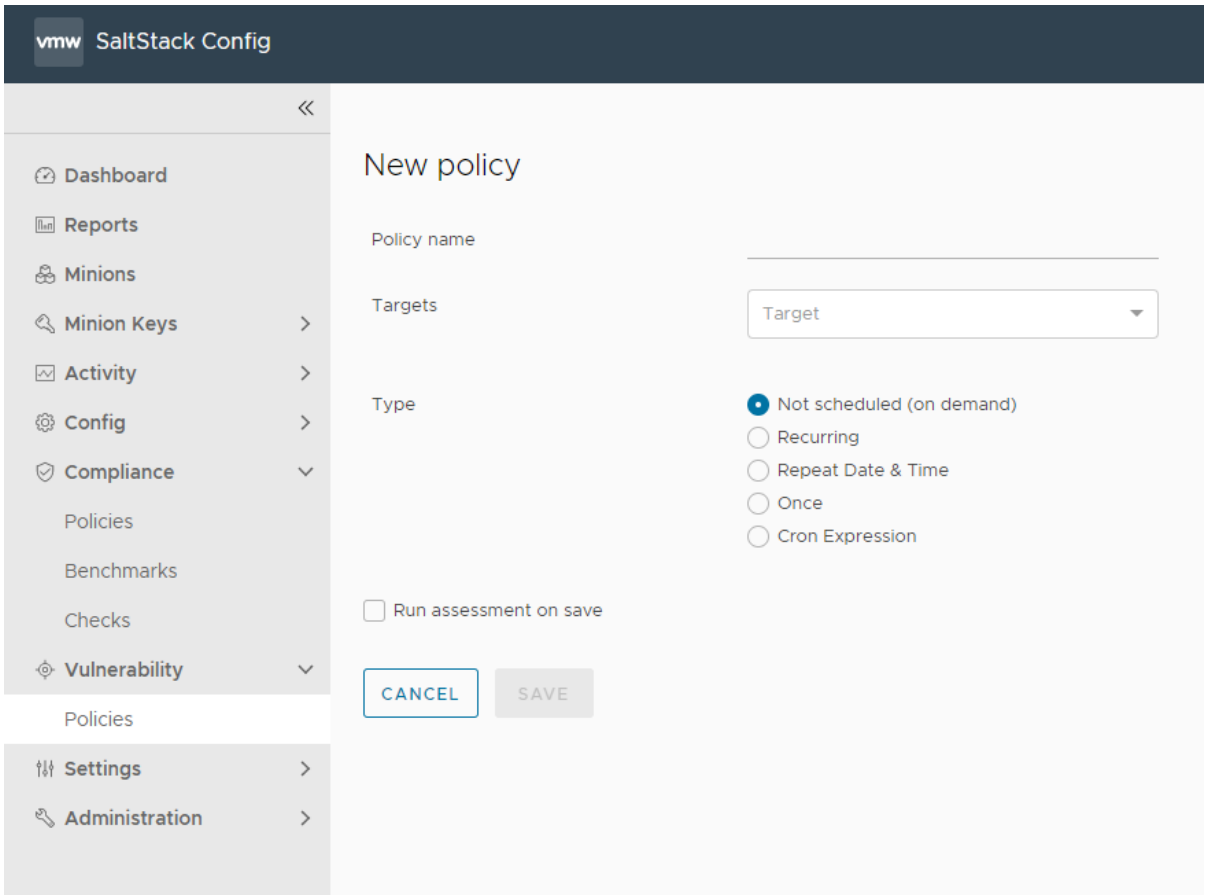
Prerequisites

Before creating your first security policy, you need access to the vulnerability library. For more information, see Using the vulnerability library for more information.

You must also create the targets that you want to assess before creating your policy. A target is the group of assets (referred to as minions) your policy will apply to. For more information, see Minions.

Procedure

1 In the Vulnerability workspace, click **Create Policy**.



2 Enter a policy name and select the target you want to access.

Note Scanning a large number of minions might result in long processing times. This could also delay other processes, such as jobs running, in SaltStack Config. Make sure to account for extra time required to run large assessments.

3 Define a schedule frequency.

4 (Optional) Select **Run assessment on save**.

5 Click **Save**.

Results

The policy is saved. If you selected Run assessment on save, the policy is run immediately after saving. If necessary, you can edit a policy by selecting the policy from the Vulnerability workspace and clicking **Edit Policy** and then **Save**.

How do I run a vulnerability assessment

After you've created a policy, you can run an assessment that scans the targeted assets against the latest advisories.

SaltStack SecOps Vulnerability scans for available packages that can repair vulnerabilities identified by the advisory.

Note After initial installation, SaltStack Config takes about 15-20 minutes to ingest vulnerability content. For best results, wait at least 20 minutes after installing SaltStack Config before you run your first vulnerability scan. For more information, see Using the vulnerability library.

From the vulnerability workspace, you can run assessments from one more policies at once by clicking the checkboxes next to each policy and clicking **Run assessment**.

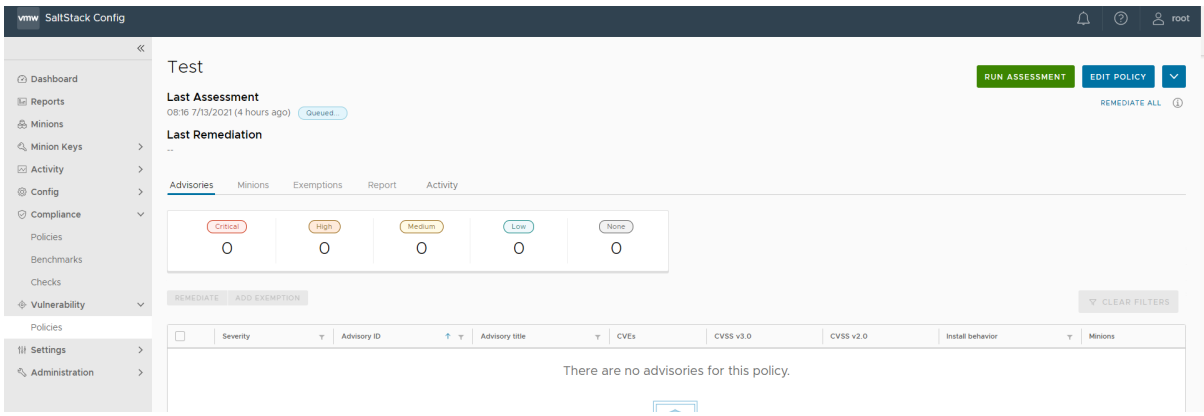
To view policy details and then run assessment on a single policy:

Prerequisites

Before you can run a vulnerability assessment, you must have an existing vulnerability policy. For more information, see How do I create a vulnerability policy.

Procedure

1. In the Vulnerability workspace, select a policy to open the policy's dashboard.
2. In the policy dashboard, click **Run assessment** and then click **Run assessment** in the confirmation dialog box.



Results

SaltStack SecOps Vulnerability scans your system against the latest advisories. During assessment, no changes are made to any of your systems. After the assessment is complete, you can remediate any advisories. You can view the status of current or past assessments by clicking a policy in the Vulnerabilty workspace and the clicking on the **Activity** tab. The results page lists all queued, in progress, and completed scans.

What to do next

You can view your assessment results on the policy dashboard. To sort your results by node, click the **Minions** tab. If desired, you can download the assessment report in JSON format by clicking **Report > Download > JSON** from the policy dashboard.

Use Case: How do I import a third-party security scan as an alternative to running an assessment

As an alternative to running an assessment on a vulnerability policy, SaltStack SecOps Vulnerability supports importing security scans generated by a variety of third-party vendors.

Instead of running an assessment on a vulnerability policy, you can import a third-party security scan directly into SaltStack Config and remediate the security advisories it identified using SaltStack SecOps Vulnerability. See How do I run a vulnerability assessment for more information about running a standard assessment.

SaltStack SecOps Vulnerability supports third-party scans from:

- Tenable
- Rapid7
- Qualys
- Kenna Security

You can also use a Tenable.io connector for Tenable scans.

When you import a third-party scan into a security policy, SaltStack Config matches your minions to the nodes that were identified by the scan. The Import Staging workspace displays the list of advisories that can be imported and another list showing the advisories that cannot be imported currently. The list of unsupported advisories includes an explanation of why they cannot be imported.

Note By default, all SaltStack Config users can access the Connectors workspace. However, permission to run Vulnerability Vendor Import, as well as a SaltStack SecOps Vulnerability license, are required for a user to successfully import vulnerabilities from a connector.

The security policy dashboard lists the advisories identified by the third-party scan, as well as whether each advisory is supported or unsupported for remediation.

Note If the size of your export file is large, you might need to scan a smaller segment of nodes in your network with your third-party tool. Alternatively, you could import large scans using the command line interface (CLI) or API.

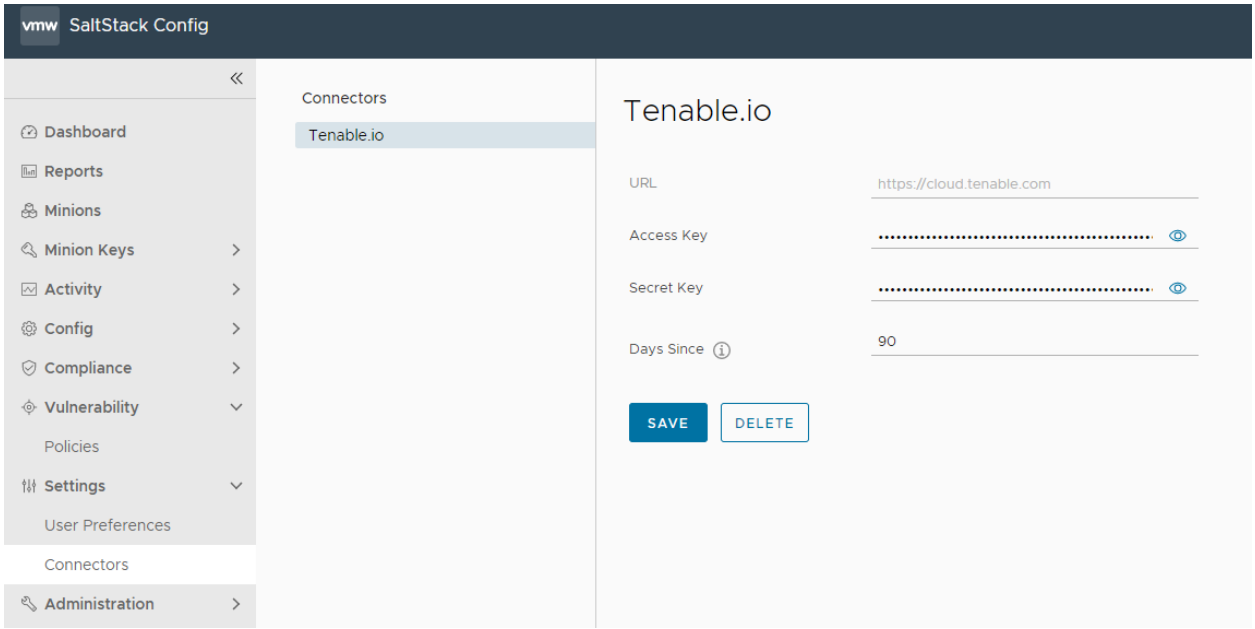
After importing your scan, the Import Staging workspace displays an import summary and two tables: a list of Supported Vulnerabilities and a list of Unsupported Vulnerabilities. Supported vulnerabilities are the advisories that are available for remediation. Unsupported vulnerabilities are the advisories that cannot currently be remediated. The list of unsupported vulnerabilities includes an explanation of why they cannot be imported.

You can import a third-party from a file, from a connector, or by using the command line.

Prerequisites

Before you can import a third-party security scan you must configure a connector. The connector must first be configured using the third-party tool's API keys.

To configure a Tenable.io connector:



To configure a Tenable.io connector, navigate to **Settings > Connectors > Tenable.io**, enter the required details for the connector, and click **Save**.

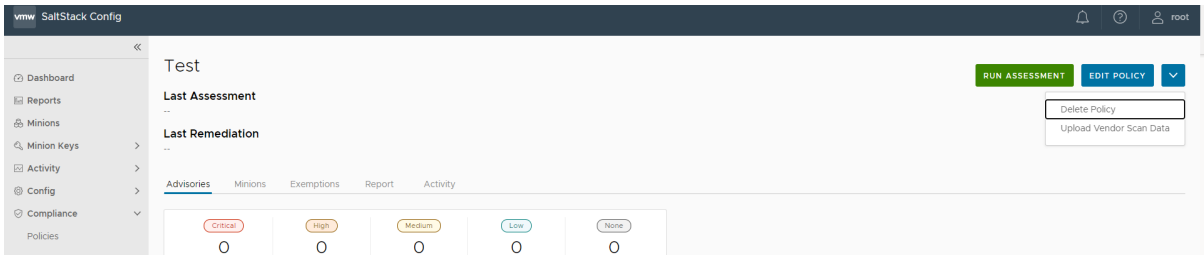
Connector field	Description
Secret Key and Access Key	Key pair required to authenticate with the connector API. For more information on generating your keys, see the Tenable.io documentation.
URL	Base URL for API requests. This defaults to `https://cloud.tenable.com`.
Days since	Query Tenable.io scan history beginning this number of days ago. Leave blank to query an unlimited period of time. When you use a connector to import scan results, SaltStack SecOps Vulnerability uses the most recent results per node available within this period. **Note** To ensure your policy contains the latest scan data, make sure to rerun your import after each scan. SaltStack SecOps Vulnerability does not poll Tenable.io for the latest scan data automatically.

Procedure

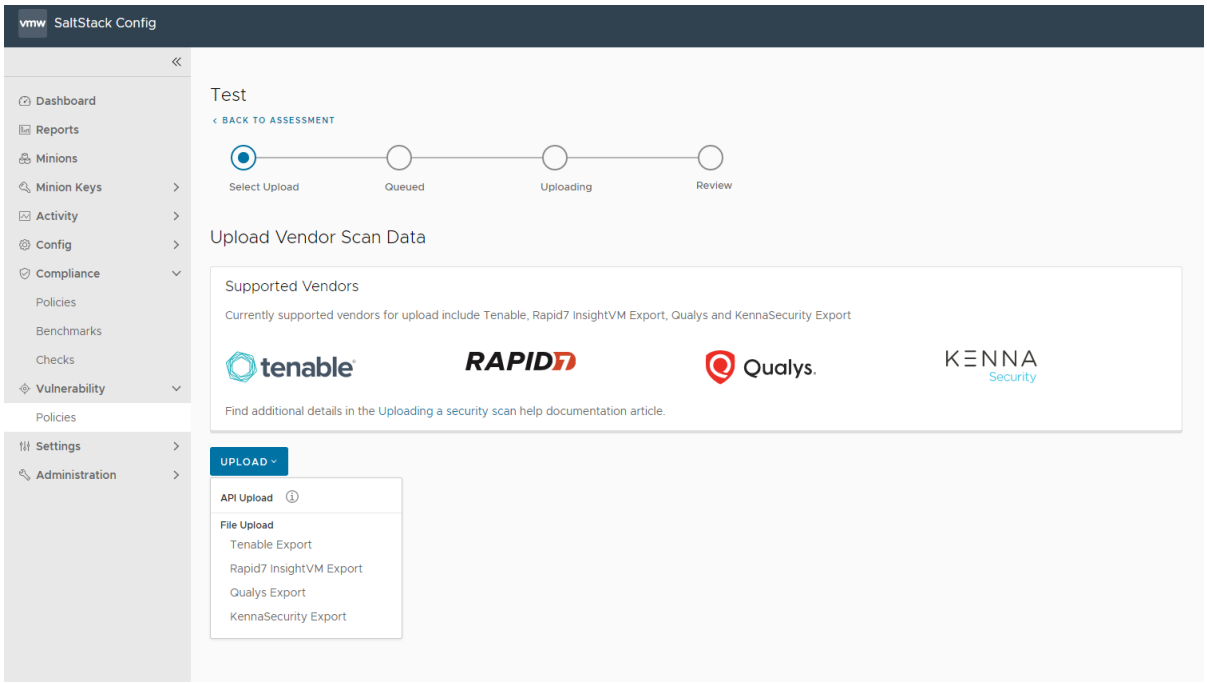
1. In your third-party tool, run a scan and make sure to pick a scanner that is in the same network as the nodes you want to target. Then, indicate the IP addresses you want to scan. If you are importing a third-party scan from a file, export the scan in one of the supported file formats (Nessus, XML, or CSV).
2. In SaltStack Config, make sure you have downloaded SaltStack SecOps Vulnerability content.
3. In the SaltStack SecOps Vulnerability workspace, create a security policy targeting the same nodes that were included in the third-party scan. Ensure the nodes you scanned in your third-party tool are also included as targets in the security policy. See How do I create a vulnerability policy for more information.

   Note After exporting your scan and creating a policy, you can import your scan by running the `raas third_party_import "filepath" third_party_tool security_policy_name` command. For example, `raas third_party_import "/my_folder/my_tenable_scan.nessus" tenable my_security_policy`. It is recommended to import your scan using the CLI if the scan file is especially large.

4. In the policy dashboard, click the **Policy Menu** drop down arrow and select **Upload Vendor Scan Data**.



5 Import your scan:



- If importing your scan from a file, select **Upload > File Import** and select your third-party vendor. Then select the file to upload your third-party scan.
- If importing your scan from a connector, select **Upload > API Upload** and select the third-party. If no connectors are available, the menu directs you to the connectors settings workspace.

The import status timeline shows the status of your import as SaltStack SecOps Vulnerability maps your minions to the nodes that were identified by the scan. Depending on the number of advisories and affected nodes, this process might take some time.

6 Click **Import All Supported** to import all supported advisories. Alternatively, you can click the checkbox next to specific advisories in the **Support Vulnerabilities** table and click **Import Selected**.

Results

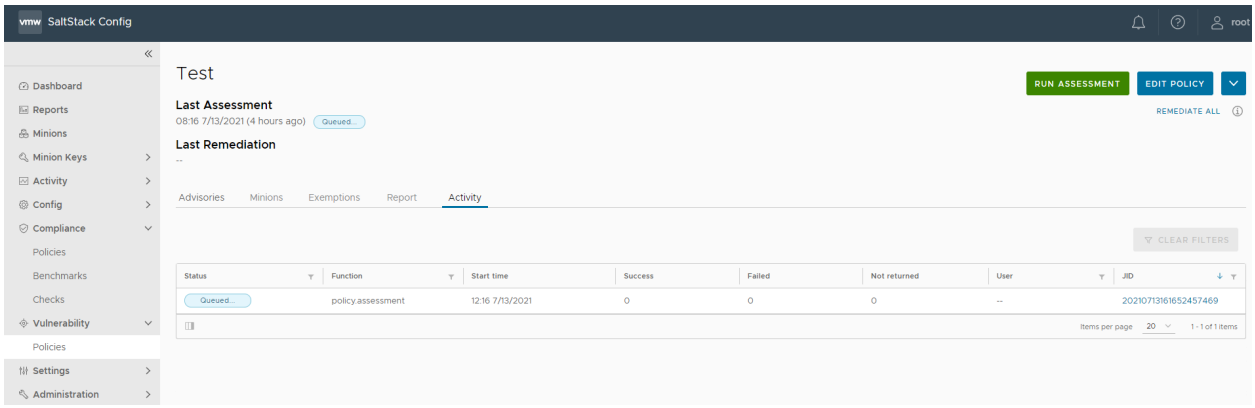
The selected advisories are imported to SaltStack SecOps Vulnerability and appear as an assessment in the policy dashboard. The policy dashboard also displays Imported from under the policy title to indicate that the latest assessment was imported from your third-party tool.

What to do next

You can now remediate these advisories. See How do I remediate my advisories for more information.

How do I remediate my advisories

After completing an initial assessment, you can then remediate the advisories that were detected in the assessment.



On the policy home page, the Activity tab shows a list of completed or in-progress assessments and remediations and their statuses:

Status	Description
Queued	The operation is ready to run but the minions have not started the operation.
Completed	The operation is finished running.
Partial	The operation is still waiting for some minions to return, although the operation has finished running.

During the remediation, all packages that are part of that advisory are applied to the selected nodes. You can remediate all advisories at once or you can remediate a specific advisory, a specific minion, or set of minions as needed.

SaltStack SecOps Vulnerability always installs the latest available version that is available from a vendor, even if the advisory was fixed by an earlier version.

After remediating an advisory, you must run another assessment to verify the remediation was successful.

You can choose which advisories or nodes to remediate as needed. These options include:

- Remediating all advisories at once
- Remediating a specific minion
- Remediating a set of minions

From the policy dashboard, when you run **Remediate all**, SaltStack SecOps Vulnerability remediates all advisories on all minions in your policy, which may result in long processing times.

Prerequisites

SaltStack SecOps Vulnerability triggers Windows nodes to receive the latest advisories from Microsoft. Windows nodes can receive these updates in one of two ways:

- Windows Update Agent (WUA) - By default, Windows nodes connect directly to Microsoft using the WUA, which is automatically installed on all Windows nodes. The WUA supports automated patch delivery and installation. It scans nodes to determine which security updates are not installed and then searches for and downloads the updates from Microsoft's update websites.
- Windows Server Update Services (WSUS) - A WSUS server acts as an intermediary between Microsoft and the minion. WSUS servers allow IT administrators to deploy updates to a network strategically to minimize down time and disruptions. See Windows Server Update Services (WSUS) in the official Microsoft documentation for more information.

Before using SaltStack SecOps Vulnerability on Windows nodes, verify which of these two methods your environment is currently using. If your environment is using the WSUS server method, you must:

- **Ensure that the WSUS is enabled and running**. If needed, you can configure the Windows minion to connect to a WSUS using a Salt state file provided by SaltStack. See Enabling Windows Server Update Services (WSUS) for this state file. After running this state file, verify that your minion is successfully connecting to the WSUS server and is receiving updates.
- **Approve updates related to advisories from Microsoft on the WSUS server**. When the WSUS server receives updates from Microsoft, the WSUS administrator must review and approve those updates in order to deploy the updates in the environment. In order for SaltStack SecOps Vulnerability to detect and remediate advisories, any updates that contain advisories must be approved.

If either of these two prerequisites are not met, SaltStack SecOps Vulnerability cannot accurately scan and remediate advisories. For systems that receive Microsoft updates through a WSUS server, assessments could return false positives that indicate Windows minions are secure from all CVEs even though they may not actually be secure.

Procedure

1. In the Vulnerability workspace, click a policy to open the policy's dashboard.
2. From the policy's dashboard:
   - To remediate by advisory, click the checkbox next to all advisories you want to remediate.
   - To remediate by minion, select the **Minions** tab, click a minion, and select all advisories you want to remediate for the active minion.
   - To remediate by both advisory and minion, click an advisory ID and click the checkbox next to all minons your want to remediate for the active advisory.
3. Click **Remediate**.

Results

Your vulnerability advisories are remediated. Occasionally, remediation might require a full system or minion reboot. For more information, see How do I reboot a minion as part of remediation.

How do I reboot a minion as part of remediation

A remediation might require a full system reboot in order for the patch or update to take effect. Occasionally, a remediation might even require a second reboot.

As a Windows administrator, to determine if an advisory or minion requires a reboot as part of a remediation, first run an assessment.

Note Rebooting as part of remediaton only applies to Windows minions.

Then to determine whether a reboot is needed:

For	Refer to
Advisory	On the Advisories tab of the policy dashboard, check the Install Behavior column for the advisory's status: ■ Never requires reboot - The advisory does not require a reboot when it is remediated. ■ Always requires a reboot - The advisory always requires a reboot when it is remediated. ■ Can require reboot - The advisory could possibly require a reboot under certain conditions as part of remediation. ■ (-) - The null value. This displays for Linux minions. Detecting whether a reboot is required is not supported for Linux minions.

Minion	On the Minions tab of the policy dashboard, check the Needs Reboot column for the minion's status:
	■ false - The minion either does not need a reboot for remediation or the minion has successfully rebooted.
	■ true - The status is true if:
	  ■ The minion needs a reboot and a reboot has not been started.
	  ■ The minion is currently rebooted and has not yet finished rebooting.
	  ■ The minion has rebooted but it will need a second reboot to apply additional changes.

If you detemine your system or minion needs a reboot follow these steps:

Procedure

1 On the **Minions** tab of the policy dashboard, click the checkbox next to a minion that shows true in the **Needs Reboot** column.

2 Click **Run Command**.

3 In the **Function** menu, select the `system.reboot` command.

4 In the **Arguments** field, add the necessary arguments.

- For Windows nodes, the `system.reboot` command needs two arguments: `timeout` and `in_seconds`. Set the first argument to 0 and the second argument to true. See the win_system.reboot module documentation for more information about these arguments.
- For Linux nodes, the `system.reboot` command takes one argument: `at_time`. See the system.reboot module documentation for more information about these arguments.

5 (Optional) If you want to schedule a reboot for a specific time, create a job that reboots the minion and then set that job to run at a scheduled time. See SaltStack Config jobs workflow for more information.

6 Click **Run Command** to run this command on the select minion.

Results

After initiating a reboot, the minion might take several minutes to reboot and come back online.

To check whether the minion is back online after a reboot, refresh the **Minions** tab in the Vulnerability workspace and check the minion's presence. See Minion presence for more information.

What to do next

After rebooting a minion as part of a remediation, you must run another assessment to verify the remediation was successful.

Troubleshooting

6

If you encounter problems using SaltStack SecOps, perform these troubleshooting steps.

Problem	Cause	Solution
After installing SaltStack SecOps without using vRealize Suite Lifecycle Manager, you encounter a licensing error.	SaltStack SecOps requires both a SaltStack SecOps license and a DLF license.	1 Create a file named vra_license in `/etc/raas`. 2 Edit the `/etc/rass/vra_license` file and add your SecOps key. 3 Save the vra_license file. 4 Run `chown rass:raas vra_license` 5 Run `systemctl restart raas`.