



Splunk® SOAR (On-premises)

Administer Splunk SOAR (On-premises) 5.4.0

Generated: 10/28/2022 9:28 pm

Table of Contents

Introduction to Splunk SOAR (On-premises)	1
Administer Splunk SOAR (On-premises)	1
Take a tour of Splunk SOAR (On-premises) and perform product onboarding when you log in for the first time	2
Splunk SOAR (On-premises) security information	5
Splunk SOAR (On-premises) security information	9
Configure your company's settings in Splunk SOAR (On-premises)	14
Configure your company settings in Splunk SOAR (On-premises)	14
Configure the ROI Settings dashboard	14
Obtain and configure a Splunk SOAR (On-premises) license	15
Configure administration settings in Splunk SOAR (On-premises)	18
Configure a source control repository for your Splunk SOAR (On-premises) playbooks	18
Customize email templates in Splunk SOAR (On-premises)	21
Configure search in Splunk SOAR (On-premises)	23
Configure Google Maps for visual geolocation data	28
Run playbooks in parallel with vertical scaling	29
Manage your organization's credentials with a password vault	31
Set global environment variables	34
Set the global action concurrency limit	35
Add tags to objects in Splunk SOAR (On-premises)	36
Create custom CEF fields in Splunk SOAR (On-premises)	38
Reset the admin and root passwords in Splunk SOAR (On-premises)	39
Delete containers from your Splunk SOAR (On-premises) deployment	41
Configure a source control repository for your Splunk SOAR (On-premises) playbooks	42
Configure settings for your Splunk SOAR (On-premises) instance	46
Enable clickable URLs in CEF data	46
Clustering	46
View cluster status and enable or disable a cluster	46
Configure multiple tenants on your Splunk SOAR (On-premises) instance	46
View related data using aggregation rules	49
Define tasks using workbooks	50
Tune performance by managing Splunk SOAR (On-premises) features	52
Use data retention strategies to schedule and manage your database cleanup	54
Configure settings for your Splunk SOAR (On-premises) system's events	57
Create custom status labels in Splunk SOAR (On-premises)	57
Create custom severity names	57
Create custom fields to filter Splunk SOAR (On-premises) events	58
Filter indicator records in Splunk SOAR (On-premises)	60
Track information about an event or case using HUD cards	61
Configure the response times for service level agreements	61
Configure how events are resolved	62
Configure labels to apply to containers	62
Use authorized users to grant authorized access	63

Table of Contents

Manage your Splunk SOAR (On-premises) users and accounts.....	65
Manage Splunk SOAR (On-premises) users.....	65
Manage roles and permissions in Splunk SOAR (On-premises).....	68
Configure password requirements and timeout intervals to secure your Splunk SOAR (On-premises) accounts.....	71
Configure single sign-on authentication for Splunk SOAR (On-premises).....	72
Secure Splunk SOAR (On-premises) using two factor authentication.....	76
Configure role based access control inside Splunk apps.....	76
Secure Splunk SOAR (On-premises) by configuring an account password expiration.....	77
Manage your registered mobile devices.....	79
Enable or disable registered mobile devices.....	79
Monitor your Splunk SOAR (On-premises) system activity.....	81
Monitor the health of your Splunk SOAR (On-premises) system.....	81
View how much data is ingested in Splunk SOAR (On-premises) using ingestion summary.....	82
View ingested container statistics using Ingestion Status.....	83
Configure the logging levels for Splunk SOAR (On-premises) daemons.....	83
Enable and download audit trail logs in Splunk SOAR (On-premises).....	88
Locate long-running playbooks for debugging or troubleshooting in Splunk SOAR (On-premises).....	90
View the playbook run history in Splunk SOAR (On-premises).....	90
View Playbook Run Statistics.....	91
View the action run history.....	92
Use ITSI to monitor the health of your Splunk SOAR (On-premises) deployment.....	92
Use Python scripts and the REST API to manage your Splunk SOAR (On-premises) deployment.....	92
Manage Splunk SOAR (On-premises) Certificate Store.....	96
Splunk SOAR (On-premises) certificate store overview.....	96
Add or remove certificates from the Splunk SOAR (On-premises) certificate store.....	96
Troubleshooting certificate issues.....	96
Backup or restore your Splunk SOAR (On-premises) instance.....	98
Splunk SOAR (On-premises) backup and restore overview.....	98
Back up a Splunk SOAR (On-premises) deployment.....	102
Restore Splunk SOAR (On-premises) from a backup.....	107
Splunk SOAR (On-premises) backup tools.....	110
Use ibackup.pyc with warm standby.....	112
Create and manage a warm standby.....	116
Warm standby feature overview.....	116
Create a warm standby.....	119
Failover to the warm standby.....	122
Disable warm standby for Splunk SOAR (On-premises).....	123
Recreate warm standby after a failover.....	123
Upgrade or maintain warm standby instances.....	125
Warm standby tools.....	126

Table of Contents

Manage your Splunk SOAR (On-premises) Apps and Assets.....	128
Add and configure apps and assets to provide actions in Splunk SOAR (On-premises).....	128
Splunk SOAR (On-premises) telemetry.....	136
Share data from Splunk SOAR (On-premises).....	136
Share data from Splunk SOAR (On-premises).....	136

Introduction to Splunk SOAR (On-premises)

Administer Splunk SOAR (On-premises)

Splunk SOAR (On-premises) is a cloud-based Security Orchestration, Automation, and Response (SOAR) system that is delivered as a SaaS (software-as-a-service) solution hosted and managed by Splunk.

The Splunk SOAR (On-premises) platform combines security infrastructure orchestration, playbook automation, and case management capabilities to integrate your team, processes, and tools to help you orchestrate security workflows, automate repetitive security tasks, and quickly respond to threats.

This manual is intended to be used by the person or team administering the Splunk SOAR (On-premises) system.

The following topics are discussed in this manual:

Feature	Description
Company Settings	Information about your company, contacts, and your Splunk SOAR (On-premises) license.
Administration Settings	All the settings to configure the behavior and appearance of Splunk SOAR (On-premises).
Product Settings	Settings for the Splunk SOAR (On-premises) product that apply to your deployment, such as clickable URLs, aggregation, and workbooks.
Event Settings	Settings to configure the organization, handling, and presentation.
User Management	Settings related to user accounts, permissions, and authentication.
View how much data is ingested in Splunk SOAR (On-premises) using ingestion summary	Information and reports for monitoring the activity of your Splunk SOAR (On-premises) deployment.
Apps and Assets	How to add and configure apps and assets to provide actions in Splunk SOAR (On-premises).
Telemetry	Information about sharing data from Splunk SOAR (On-premises).

Splunk Technical Support

Splunk Standard Support is included in every Splunk SOAR (On-premises) subscription. For details about the levels of technical support provided, read [Support Programs](#). Only authorized support contacts from your company can open cases. Your Splunk support agreement specifies who your authorized contacts are. Your Support contract specifies a number of authorized contacts, and an expiration date. One of your contacts is a Support portal administrator, who can update the list. Only an authorized contact can open a case and track its status. An authorized contact can file a case by logging in to [splunk.com](#), then navigating to the **Support Portal**.

Splunk Support portal

Designated Splunk SOAR (On-premises) users can manage operational contacts for their account and file support cases using the Support portal. Operational contacts are the people in your organization who are notified when their Splunk SOAR (On-premises) environment undergoes maintenance or experiences an event that affects performance.

To manage operational contacts:

1. Go to **My Operational Contacts** in the Support portal.
2. Follow the instructions on the page to add, edit, and remove operational contacts for your Splunk SOAR (On-premises) environment.

To file a case on the Support portal:

1. From the **Splunk installation is?** dropdown, select the state of your deployment.
2. In **Subject**, summarize your issue. Splunk Support sees the first 250 characters in this field.
3. In **What Product are you having trouble with?** select **Splunk SOAR (On-premises)**.
4. In **What OS are you using?** select **Linux**.
5. Leave **What OS Version are you using?** blank.
6. In **I need help with...** select a category that applies to your issue.
7. In **What is the impact...** explain briefly how this issue disrupts your work.
8. In the **Problem Description**, be thorough. For issues (as opposed to enhancement requests), include the exact time of the issue and its duration, the type of Splunk instance experiencing the issue (for example, forwarder, search head, or indexers), and any relevant screen shots.
9. Include **Steps to reproduce** if you've found a specific scenario that triggers the issue.
10. Click **Submit**. The portal directs you to a screen with a case number and sends you an email containing the case number.

Splunk Support replies to the case creator by email. You can update the case by replying to the email (be sure to keep the tracking ID in the email subject line). You can also update the case, check on its status, or close a case using the support portal.

Splunk community

The Splunk user community is a great resource. Check out Splunk Answers, where you can ask and answer questions about the product. There are also a number of other ways to get involved in the Splunk community, such as user groups or the Splunk Trust. For more information about getting involved with the Splunk community, see the Community portal.

See also

- Use playbooks to automate analyst workflows in Splunk SOAR (On-premises) in the *Build Playbooks with the Playbook Editor* manual.

Take a tour of Splunk SOAR (On-premises) and perform product onboarding when you log in for the first time

When you log in to Splunk SOAR (On-premises) for the first time, there are several screens you must navigate before arriving at the home page. The screens appear in the following order:

- [Read and accept the Splunk End User License Agreement.](#)
- [Review and understand how Splunk collects and uses aggregated product usage data.](#)
- (Optional) [Take a tour of Splunk SOAR \(On-premises\) and create some sample data.](#)
- (Optional) [Configure basic settings for your Splunk SOAR \(On-premises\) instance, data sources, playbooks, and apps and assets.](#)

Read and accept the Splunk End User License Agreement

When you log in to Splunk SOAR (On-premises) for the first time, you must read and accept the Splunk End User License Agreement.

1. Scroll to the bottom of the End User License Agreement.
2. Click **I Accept**.

Review and understand how Splunk collects and uses aggregated product usage data

Splunk collects and sends anonymized usage data to Splunk. This behavior is enabled by default. Read the text on the **Helping You Get More Value from Splunk Software** page and click **Got it**.

See [Share data from Splunk SOAR \(On-premises\)](#) for information about how to opt out, what information is shared, and how it is used.

Take a tour of Splunk SOAR (On-premises) and create some sample data

Generate some sample data and get a guided tour of Splunk SOAR (On-premises)'s main pages.

Click **Exit Tour** at any time to leave the tour and go to the onboarding tutorial, where you can [Configure basic settings for your Splunk SOAR \(On-premises\) instance, data sources, playbooks, and apps and assets](#).

Perform the following tasks to create some sample data and take the guided tour:

1. Click **Get Started** to begin the product tour and create sample events.
2. Generate some sample events. Click the number of sample events you want to generate. After the events are generated, the **Sources** page shows you the sample events.
3. Click **View Event** to view the details for an event on the Investigation page.
4. Click **Run Playbook** to run a playbook against this event. In Investigation, the **Activity** tab shows the automated actions taken against the event by the playbook.
5. Click **View Playbook** to view the playbook in the Playbook Editor. Playbooks run from the **Start** block and perform the actions up to the **End** block.
6. Click **Configure Splunk SOAR (On-premises)** to complete the tour and go to the onboarding tutorial, where you can [Configure basic settings for your Splunk SOAR instance, data sources, playbooks, and apps and assets](#).

Configure basic settings for your Splunk SOAR (On-premises) instance, data sources, playbooks, and apps and assets

Click **Skip on-boarding** at any time to go directly to the Splunk SOAR (On-premises) home page. See Log in and navigate Splunk SOAR (On-premises) in *Use Splunk SOAR (On-premises)*.

- [Configure basic settings](#).
- [Configure a data source](#).
- [Run a demo playbook](#).
- [Configure apps and assets](#).

Configure basic settings

Configure basic administrative and email settings for your Splunk SOAR (On-premises) instance.

1. Configure the administrative password, company name, IT contact email address, system time zone, and the appliance base URL for this Splunk SOAR (On-premises) instance. If you skip the on-boarding, you can configure these fields later. See [Configure your company settings in Splunk SOAR \(On-premises\)](#) for more information about these fields.
2. Configure email server settings. Splunk SOAR (On-premises) requires an email server to send users email for action approvals, when SLAs are breached, and when items that they are tracking change. If you skip the on-boarding, you can configure the email server and asset later. See [Add and configure apps and assets to provide actions in Splunk SOAR \(On-premises\)](#).
 1. Use **smtp** as the default asset name, or enter a new name.
 2. Enter the IP address or hostname of the email server.
 3. Select the SSL method that your Splunk SOAR (On-premises) instance should use to connect to the email server.
 4. Complete the email asset configuration by providing a tag, username, password, sender address, and port.
 5. Click **Enable Unicode Support** to enable Splunk SOAR (On-premises) to properly display Unicode characters in the emails.

Configure a data source

Configure a data source from which Splunk SOAR (On-premises) can ingest data. In this on-boarding procedure, you can add one data source. You can add additional data sources later at any time. See [Add and configure apps and assets to provide actions in Splunk SOAR \(On-premises\)](#).

Perform the following tasks to configure a data source during the on-boarding procedure.

1. Select a data source.
2. Select or specify an asset name.
3. Select or specify a container name.
4. (Optional) Click **Additional Information** to expand the section.
 1. Enter one or more **Tags** to attach to the objects from this data source.
 2. Enter a description for the asset.
 3. Complete other fields specific to the asset type. The fields may vary depending on the data source you selected.
5. Click **Save**.
6. In some cases, you are asked to perform additional tasks. For example, if you configure a Splunk data source, you must record the authorization token that is provided and also download a separate app from Splunkbase in order for the integration between Splunk SOAR (On-premises) and the Splunk platform to work.
7. Click **Continue**.

Run a demo playbook

A list of playbooks is available based on the data source you configured. Select a playbook you want to run, then click **Save and Continue**.

Configure apps and assets

Configure apps and assets that will provide actions for your playbooks.

1. Select the apps that will provide the actions for the selected playbook.
 - ♦ If you selected the **investigate** playbook, select one app in each of the Information Services, File Reputation Services, Domain Reputation Services, Sandbox, and Threat Intel.

- ◆ If you selected the **hunting** playbook, select one app in each of the Information Services, Endpoint Services, File Reputation Services, and Sandbox.
- 2. In the **Select Apps to Configure** section, click on each app and provide the required information to configure an instance of the app, called an asset.
- 3. Click **Additional Information** to expand the section and provide additional information.
- 4. Click **Save and Test Connectivity** to verify the configuration of each asset.

Splunk SOAR (On-premises) security information

This topic explains the fundamentals of the Splunk SOAR (On-premises) system design and base security measures, as well as the parameters and limitations for that design.

Operating System

Splunk SOAR (On-premises) runs on top of one of the supported operating systems:

1. Red Hat Enterprise Linux 7.6 through 7.9
2. CentOS 7.6 through 7.9

If you deployed Splunk SOAR (On-premises) using the Amazon Marketplace Image (AMI), the base operating system is CentOS 7.9.

Splunk SOAR (On-premises) does not monitor or control the operating system on which it is deployed.

Basic OS privilege separation is utilized, partitions are mounted with limited capabilities, and SELinux is on.

Processes and daemons

Splunk SOAR (On-premises) runs multiple processes and daemons:

- The web-based user interface runs in the http process as the nginx user. Splunk SOAR (On-premises) uses a custom httpd configuration. Use caution if you update http.
- The watchdogd daemon runs as the phantom user and is responsible for starting or stopping other processes, and collecting system and process information.
- All other daemons run as the phantom user.

Start up

This section provides a brief overview of what happens when Splunk SOAR (On-premises) starts.

- In cloud and unprivileged deployments, because Splunk SOAR (On-premises) does not have root level access to configure systemd items, the user account that runs Splunk SOAR (On-premises) has its crontab modified to run <PHANTOM_HOME>/bin/start_phantom.sh at system boot time.

Access to the operating system

Splunk SOAR (On-premises) users do not have access to the operating system of their Splunk SOAR (On-premises) deployment.

Access to the operating system is separate from access to the web-based user interface. It is managed by a systems administrator. Accounts added or removed using the operating system shell do not affect the ability to log in to the web-based user interface. Splunk SOAR (On-premises) utilizes local database accounts or remote identity providers for authentication to the web-based user interface.

If you deployed Splunk SOAR (On-premises) from a virtual machine image, remote SSH access as the root user is disabled by default.

Ultimately, it is impossible for Splunk SOAR (On-premises) to be secure against an attacker who has access to or control of the local operating system or virtualization platform where it is deployed. Splunk SOAR (On-premises) can have assets with configured credentials, such as firewalls, mail servers, for Active Directory, or other critical infrastructure. Because an attacker with root access can access everything they need to reverse engineer or bypass access controls, it is vitally important that the operating system and any virtualization platform be made secure.

Ports and endpoints

Splunk SOAR (On-premises) requires access to several ports and endpoints in order to function. Lists of the needed ports and endpoints are available at [Splunk SOAR \(On-premises\) ports and endpoints](#).

System maintenance and updates

System maintenance tasks, such as system software patching, maintaining disk space, and managing operating system access are the responsibility of customer's systems administrators.

System backups or virtual machine snapshots should be made before performing any system changes. If you have concerns about whether an update or change is likely to affect the operation of Splunk SOAR (On-premises) open a Support case.

Authentication

Splunk SOAR (On-premises) uses its own authentication database, independent of the linux operating system.

There are several options for web UI authentication. The local user database uses the default Django PBKDF2 hash. See the Wikipedia article <https://en.wikipedia.org/wiki/PBKDF2> for more information. Other options include:

- LDAP/LDAPS
- OAUTH
- SAML

Splunk SOAR (On-premises) supports using Duo for two-factor authentication.

Splunk SOAR (On-premises) supports password complexity for its local accounts. Users that require the most advanced account security features are encouraged to use an external identity provider.

Splunk SOAR (On-premises) does use a certificate store for authenticating the LDAPS authentication server.

For more on information configuring users, two-factor authentication, and passwords, see the section [Manage your Splunk SOAR \(On-premises\) users and accounts](#) in *Administer Splunk SOAR (On-premises)*.

Clustering

Splunk SOAR (On-premises) can be deployed as a cluster, using multiple nodes which can share a PostgreSQL database, filesystem, Splunk Enterprise instance, and distribute running apps, playbooks, and action runs between them.

Because any node may introduce new Python code into the system as an app or playbook, an attacker compromising any one single Splunk SOAR (On-premises) node has the ability to compromise all the other nodes and services in the same way.

If one cluster node is determined to be compromised at the OS level, all the other cluster nodes and services should be assumed to be compromised as well.

SSL and TLS

Splunk SOAR (On-premises) has a certificate store used to validate certificates when opening connections to other servers.

The certificates in the store are trusted certificate authority (CA) certificates from mkcert.org. In almost all cases, Splunk SOAR (On-premises) can use its certificate store to validate any certificate issued by a commercial certificate authority (CA).

If an asset uses TLS and has a self-signed certificate, or if you have an in-house certificate authority, then those certificates must be imported into the store for verification to work.

This includes any necessary intermediate certificates. Note that the requirement for the Common Name to match still applies, so if the certificate is for server.example.com, then the Splunk SOAR (On-premises) asset must also be configured to connect to it as server.example.com, and not a different form of the name such as "server", or an IP address.

See [Splunk SOAR \(On-premises\) certificate store overview](#)

Embedded git client

The git client uses the OpenSSL certificate store, which includes most commercial CAs. Git repositories can be configured to use an HTTPS URI if that repository uses a signed certificate from a commercial certificate authority.

If you need to connect to a git repo that uses an unrecognized CA, you have to disable git certificate checking system-wide.

Playbooks, apps, and Python code

Splunk SOAR (On-premises) uses user-supplied Python code in several ways.

- Apps are collections of Python code and JSON configuration files that allow Splunk SOAR (On-premises) to connect to, use, and control other products or services. Apps provide Actions to Splunk SOAR (On-premises), to make controlling your security infrastructure easy.
- Playbooks are specially-crafted Python code that utilize Splunk SOAR (On-premises) Python libraries run actions, use apps, or run custom code.

Apps and playbooks are available from several sources:

- Splunk provides several apps and playbooks from Splunkbase, the Phantom Portal, and a GitHub repository.
- You can develop apps and playbooks yourself.
- You can get apps or playbooks from third parties, such as other Splunk SOAR (On-premises) users.

When running, the python code from Apps and Playbooks are running as either the linux user account phantom, or the specified account that runs Splunk SOAR (On-premises).

It is critical that any untrusted code you obtain from other sources be examined thoroughly.

Python code runs without restrictions other than that it is running as a user account without any special privileges. There is no sandbox of any kind. Anything that the Python language with common libraries can do can be done from an app's or playbook's code. If apps, or assets have configured credentials, obtaining those credentials is possible from an app or playbook's code.

Malicious apps can also introduce hostile HTML into the web user interface in two places:

- App documentation can include HTML.
- Apps often include widgets which render HTML in the web user interface.

Any sort of attack that one could typically perform with an XSS exploit could also be performed by a malicious app. This can allow complete control of the Splunk SOAR (On-premises) UI for a logged-in administrator account.

In addition to malicious behavior, it's also possible for app authors to inadvertently introduce security holes.

For example, an app author may make a system call to run a command, and pass user-supplied data to the shell without properly sanitizing the inputs, potentially leading to a command injection vulnerability. Worse, they might pick up input from a security alert that ultimately came from an outside attacker, and do the same.

User accounts, roles, and privileges

Splunk SOAR (On-premises) supports multiple types of users, has a number of built-in roles which can be assigned to users, and the ability to define custom roles with customized individual privileges.

See also:

- [Manage Splunk SOAR \(On-premises\) users](#)
- [Manage roles and permissions in Splunk SOAR \(On-premises\)](#)

Noteworthy user privileges and roles

When auditing your Splunk SOAR (On-premises) deployment's security, these user accounts and privileges which are especially important. While these roles privileges are expected to be given only to trusted users, it is vitally important to know what capabilities you are trusting them with before doing so.

Administrator role

Administrators can perform any function in the web UI. They can modify users, roles, edit or install apps, manage assets, edit or manage playbooks, change system settings, and more.

Administrators can manipulate users and assets.

- Local user accounts have passwords associated with them. Once the password is set, the UI will not display it to any account. However, an Administrator can simply change passwords.
- Assets can have stored credentials configured. Once credentials are stored, the UI will not display them to any account.

Edit Users and Roles

A user with the "Edit Users and Roles" permission, even if that is the only permission they have, can grant themselves any and all other privileges. Any user or role with this permission is effectively an administrator.

Edit Apps and/or Edit Playbooks

Users with these permissions can edit apps or playbooks, which gives them the ability to execute arbitrary Python code. This means they could leverage Python code to get access to the system shell and attempt other attacks or privilege escalations.

Edit Assets

This permission does not provide a direct path to escalate privileges on Splunk SOAR (On-premises) itself. With this permission a malicious actor could change an asset to connect to a different IP address or hostname for malicious server to obtain asset credentials.

Edit System Settings

The Edit System Settings privilege allows a user to modify system settings. One set of system settings are the identity providers in use. A malicious user with the Edit System Settings privilege could redirect authentication requests to an authentication server they control to obtain user credentials.

Splunk SOAR (On-premises) security information

This topic explains the fundamentals of the Splunk SOAR (On-premises) system design and base security measures, as well as the parameters and limitations for that design.

Operating System

Splunk SOAR (On-premises) runs on top of one of the supported operating systems:

1. Red Hat Enterprise Linux 7.6 through 7.9
2. CentOS 7.6 through 7.9

If you deployed Splunk SOAR (On-premises) using the Amazon Marketplace Image (AMI), the base operating system is CentOS 7.9.

Splunk SOAR (On-premises) does not monitor or control the operating system on which it is deployed.

Basic OS privilege separation is utilized, partitions are mounted with limited capabilities, and SELinux is on.

Processes and daemons

Splunk SOAR (On-premises) runs multiple processes and daemons:

- The web-based user interface runs in the http process as the nginx user. Splunk SOAR (On-premises) uses a custom httpd configuration. Use caution if you update http.
- The watchdogd daemon runs as the phantom user and is responsible for starting or stopping other processes, and collecting system and process information.
- All other daemons run as the phantom user.

Start up

This section provides a brief overview of what happens when Splunk SOAR (On-premises) starts.

- In cloud and unprivileged deployments, because Splunk SOAR (On-premises) does not have root level access to configure systemd items, the user account that runs Splunk SOAR (On-premises) has its crontab modified to run `<PHANTOM_HOME>/bin/start_phantom.sh` at system boot time.

Access to the operating system

Splunk SOAR (On-premises) users do not have access to the operating system of their Splunk SOAR (On-premises) deployment.

Access to the operating system is separate from access to the web-based user interface. It is managed by a systems administrator. Accounts added or removed using the operating system shell do not affect the ability to log in to the web-based user interface. Splunk SOAR (On-premises) utilizes local database accounts or remote identity providers for authentication to the web-based user interface.

If you deployed Splunk SOAR (On-premises) from a virtual machine image, remote SSH access as the root user is disabled by default.

Ultimately, it is impossible for Splunk SOAR (On-premises) to be secure against an attacker who has access to or control of the local operating system or virtualization platform where it is deployed. Splunk SOAR (On-premises) can have assets with configured credentials, such as firewalls, mail servers, for Active Directory, or other critical infrastructure. Because an attacker with root access can access everything they need to reverse engineer or bypass access controls, it is vitally important that the operating system and any virtualization platform be made secure.

Ports and endpoints

Splunk SOAR (On-premises) requires access to several ports and endpoints in order to function. Lists of the needed ports and endpoints are available at [Splunk SOAR \(On-premises\) ports and endpoints](#).

System maintenance and updates

System maintenance tasks, such as system software patching, maintaining disk space, and managing operating system access are the responsibility of customer's systems administrators.

System backups or virtual machine snapshots should be made before performing any system changes. If you have concerns about whether an update or change is likely to affect the operation of Splunk SOAR (On-premises) open a Support case.

Authentication

Splunk SOAR (On-premises) uses its own authentication database, independent of the linux operating system.

There are several options for web UI authentication. The local user database uses the default Django PBKDF2 hash. See the Wikipedia article <https://en.wikipedia.org/wiki/PBKDF2> for more information. Other options include:

- LDAP/LDAPS
- OAUTH
- SAML

Splunk SOAR (On-premises) supports using Duo for two-factor authentication.

Splunk SOAR (On-premises) supports password complexity for its local accounts. Users that require the most advanced account security features are encouraged to use an external identity provider.

Splunk SOAR (On-premises) does use a certificate store for authenticating the LDAPS authentication server.

For more on information configuring users, two-factor authentication, and passwords, see the section Manage your Splunk SOAR (On-premises) users and accounts in [Administer Splunk SOAR \(On-premises\)](#).

Clustering

Splunk SOAR (On-premises) can be deployed as a cluster, using multiple nodes which can share a PostgreSQL database, filesystem, Splunk Enterprise instance, and distribute running apps, playbooks, and action runs between them.

Because any node may introduce new Python code into the system as an app or playbook, an attacker compromising any one single Splunk SOAR (On-premises) node has the ability to compromise all the other nodes and services in the same way.

If one cluster node is determined to be compromised at the OS level, all the other cluster nodes and services should be assumed to be compromised as well.

SSL and TLS

Splunk SOAR (On-premises) has a certificate store used to validate certificates when opening connections to other servers.

The certificates in the store are trusted certificate authority (CA) certificates from mkcert.org. In almost all cases, Splunk SOAR (On-premises) can use its certificate store to validate any certificate issued by a commercial certificate authority (CA).

If an asset uses TLS and has a self-signed certificate, or if you have an in-house certificate authority, then those certificates must be imported into the store for verification to work.

This includes any necessary intermediate certificates. Note that the requirement for the Common Name to match still applies, so if the certificate is for server.example.com, then the Splunk SOAR (On-premises) asset must also be configured to connect to it as server.example.com, and not a different form of the name such as "server", or an IP address.

See [Splunk SOAR \(On-premises\) certificate store overview](#)

Embedded git client

The git client uses the OpenSSL certificate store, which includes most commercial CAs. Git repositories can be configured to use an HTTPS URI if that repository uses a signed certificate from a commercial certificate authority.

If you need to connect to a git repo that uses an unrecognized CA, you have to disable git certificate checking system-wide.

Playbooks, apps, and Python code

Splunk SOAR (On-premises) uses user-supplied Python code in several ways.

- Apps are collections of Python code and JSON configuration files that allow Splunk SOAR (On-premises) to connect to, use, and control other products or services. Apps provide Actions to Splunk SOAR (On-premises), to make controlling your security infrastructure easy.
- Playbooks are specially-crafted Python code that utilize Splunk SOAR (On-premises) Python libraries run actions, use apps, or run custom code.

Apps and playbooks are available from several sources:

- Splunk provides several apps and playbooks from Splunkbase, the Phantom Portal, and a GitHub repository.
- You can develop apps and playbooks yourself.
- You can get apps or playbooks from third parties, such as other Splunk SOAR (On-premises) users.

When running, the python code from Apps and Playbooks are running as either the linux user account phantom, or the specified account that runs Splunk SOAR (On-premises).

It is critical that any untrusted code you obtain from other sources be examined thoroughly.

Python code runs without restrictions other than that it is running as a user account without any special privileges. There is no sandbox of any kind. Anything that the Python language with common libraries can do can be done from an app's or playbook's code. If apps, or assets have configured credentials, obtaining those credentials is possible from an app or playbook's code.

Malicious apps can also introduce hostile HTML into the web user interface in two places:

- App documentation can include HTML.
- Apps often include widgets which render HTML in the web user interface.

Any sort of attack that one could typically perform with an XSS exploit could also be performed by a malicious app. This can allow complete control of the Splunk SOAR (On-premises) UI for a logged-in administrator account.

In addition to malicious behavior, it's also possible for app authors to inadvertently introduce security holes.

For example, an app author may make a system call to run a command, and pass user-supplied data to the shell without properly sanitizing the inputs, potentially leading to a command injection vulnerability. Worse, they might pick up input from a security alert that ultimately came from an outside attacker, and do the same.

User accounts, roles, and privileges

Splunk SOAR (On-premises) supports multiple types of users, has a number of built-in roles which can be assigned to users, and the ability to define custom roles with customized individual privileges.

See also:

- [Manage Splunk SOAR \(On-premises\) users](#)
- [Manage roles and permissions in Splunk SOAR \(On-premises\)](#)

Noteworthy user privileges and roles

When auditing your Splunk SOAR (On-premises) deployment's security, these user accounts and privileges which are especially important. While these roles privileges are expected to be given only to trusted users, it is vitally important to know what capabilities you are trusting them with before doing so.

Administrator role

Administrators can perform any function in the web UI. They can modify users, roles, edit or install apps, manage assets, edit or manage playbooks, change system settings, and more.

Administrators can manipulate users and assets.

- Local user accounts have passwords associated with them. Once the password is set, the UI will not display it to any account. However, an Administrator can simply change passwords.
- Assets can have stored credentials configured. Once credentials are stored, the UI will not display them to any account.

Edit Users and Roles

A user with the "Edit Users and Roles" permission, even if that is the only permission they have, can grant themselves any and all other privileges. Any user or role with this permission is effectively an administrator.

Edit Apps and/or Edit Playbooks

Users with these permissions can edit apps or playbooks, which gives them the ability to execute arbitrary Python code. This means they could leverage Python code to get access to the system shell and attempt other attacks or privilege escalations.

Edit Assets

This permission does not provide a direct path to escalate privileges on Splunk SOAR (On-premises) itself. With this permission a malicious actor could change an asset to connect to a different IP address or hostname for malicious server to obtain asset credentials.

Edit System Settings

The Edit System Settings privilege allows a user to modify system settings. One set of system settings are the identity providers in use. A malicious user with the Edit System Settings privilege could redirect authentication requests to an authentication server they control to obtain user credentials.

Configure your company's settings in Splunk SOAR (On-premises)

Configure your company settings in Splunk SOAR (On-premises)

Set the Company Name, IT Contact email address, System Time Zone, and the appliance Base URL for this Splunk SOAR (On-premises) instance. The settings are described in the following table:

Setting	Description
Company Name	The name of the company used in emails sent by Splunk SOAR (On-premises).
IT Contact	The email address of the OS system administrator for Splunk SOAR (On-premises). System-level alerts are sent to this email address.
Instance Name	A unique name used to identify a certain instance of Splunk SOAR (On-premises). Instance names are randomly generated and can be changed if desired, but changing the instance name is not required.
System Time Zone	The time zone for the host system of the virtual appliance that the Splunk SOAR (On-premises) instance runs on.
Base URL for Splunk SOAR (On-premises)	The URL used to access Splunk SOAR (On-premises). This field is set for you when your Splunk SOAR (On-premises) instance is created.

Configure the ROI Settings dashboard

Configure the parameters used to estimate the data displayed in the Automation ROI Summary dashboard.

Setting	Description
FTE Gained	<p>Enable this toggle make the FTE Gained widget available in the Automation ROI Summary dashboard.</p> <p>To calculate this value, Splunk SOAR (On-premises) divides the number of actions run by automation (calculated in Splunk SOAR (On-premises)) by the number of expected actions an analyst would take, based on minutes per action and analyst hours per day (configured on the ROI settings page).</p>
Time Saved	<p>Enable this toggle make the Time saved widget available in the Automation ROI Summary dashboard.</p> <p>To calculate this value, Splunk SOAR (On-premises) sums the difference between the Analyst Minutes Per Action (configured on the ROI settings page) and the actual minutes per action (calculated in Splunk SOAR (On-premises)) over all actions for the past 24 hours.</p>
Money Saved	<p>Enable this toggle make the Dollars saved widget available in the Automation ROI Summary dashboard.</p> <p>To calculate this value, Splunk SOAR (On-premises) multiplies the average time an analyst spends per action and the average analyst salary (configured on the ROI settings page) by the number of actions run by automation (calculated in Splunk SOAR (On-premises)).</p>
Annual analyst salary	The average annual salary paid to each analyst.

Setting	Description
Currency	The national currency value you want to use in the display.
Analyst hours per day	The typical number of hours each analyst works per day.
Minutes per action	The average number of minutes an analyst typically spends on any action in a case.

Use the Splunk App for SOAR to view more granular breakdowns of individual action runtimes. The Splunk App for SOAR sends Splunk SOAR (On-premises) log data back to the Splunk platform. You can use this data to generate or modify reports as needed.

The Splunk App for SOAR is not supported by Splunk. See App support types in Working with Splunkbase.

Obtain and configure a Splunk SOAR (On-premises) license

From the main menu, select **Administration > Company Settings > License** to view information about the license on your system.

There are three types of licenses available for Splunk SOAR (On-premises):

- **Community License**
This is the default, free license for everyone who registers for Splunk SOAR (On-premises) Community access and downloads Splunk SOAR (On-premises). This license is limited to a set number of actions per day. See Community License.
- **Event-based License**
This license type is based on the number of events updated in the twenty-four hour tracking period. Individual licenses vary in terms of volume.
- **Seat-based License**
This license is governed by the number of users allowed to log in to Splunk SOAR (On-premises). Seat-based licensing is available in blocks of five seats and can vary by the number of tenants.

The number of tenants is purchased as an additional parameter for both event-based and seat-based licenses.

If a license is removed or expires, Splunk SOAR reverts to the community license after 15 days.

Community license

Splunk SOAR (On-premises) installs with a default license, the Community License. The Community License is limited to:

- 100 licensed actions per day
- 50 containers
- 1 tenant
- 5 cases in the **New** or **Open** states

Splunk SOAR (On-premises) licensed actions

- `phantom.act()`
- `phantom.prompt()`

Using these actions via the REST API, a Playbook, or by executing an action in the Splunk SOAR (On-premises) graphical user interface counts as a licensed action. When used in the Visual Playbook Editor's debugger, these actions are not counted against the number of licensed actions.

No actions called from the Visual Playbook Editor's debugger count as a licensed action.

The action limit is specifically the number of actions run, as opposed to Playbooks run. Running one Playbook may invoke several actions. Also, an action run against multiple assets will count as only one action. Keep this in mind if you are managing the number of actions taken per day.

Event-based license

The Event-based license limits events.

An event is a container. A container is a top-level composite object that collects artifacts. An event-based license tracks the number of events that are updated in the twenty-four hour tracking period.

Seat-based license

Customers using a seat-based license are limited to a number of user accounts that can log in to Splunk SOAR (On-premises). This number includes local accounts in Splunk SOAR (On-premises) and accounts authenticated or managed by external services such as SAML2, LDAP, or OpenID. The built-in user accounts for the automation and the admin users do not count against a seat-based license. Other users assigned the admin role still count against a seat-based license.

Seat limits must be purchased in increments of five.

Obtaining a license

To obtain a license, you must submit a license request and obtain a Splunk SOAR (On-premises) license file.

To obtain a trial license for Splunk SOAR (On-premises), contact the Splunk SOAR (On-premises) Sales department.

To request an updated copy of a current Splunk SOAR (On-premises) license, open a license request case at <https://support.splunk.com> or call +1(855)SPLUNK-S or +1(855)775-8657.

International Splunk Support numbers are located at https://www.splunk.com/en_us/about-us/contact.html#tabs/customer-support.

The number of events permitted and expiration of the license is based on the terms listed in your company's entitlement.

Once you have your license file:

1. From the main menu, select **Administration**.
2. Select **Company Settings > License**.

3. Click **Upload Key**.
4. Provide the location of the key file on your system.
5. Click **Accept & Install**. The license is applied automatically.

The information obtained from the license file is displayed on the page.

If any of the information shown is incorrect or you experience any difficulty loading the license file, open a support case at <https://support.splunk.com> or call +1(855)SPLUNK-S or +1(855)775-8657.

Configure administration settings in Splunk SOAR (On-premises)

Configure a source control repository for your Splunk SOAR (On-premises) playbooks

You can save your Splunk SOAR (On-premises) playbooks in Git repositories. By default, playbooks are managed in a Git repository called local. You can create additional Git repositories as needed. Doing so enables you to perform the following tasks:

- Import and export playbooks and share facilities among Splunk SOAR (On-premises) instances. For example, you can use Git to publish playbooks from a development Splunk SOAR (On-premises) environment to a separate production environment.
- Edit playbooks using a tool of your choice instead of the Splunk SOAR (On-premises) web interface.

Once you edit a playbook outside of the Visual Playbook Editor (VPE), you can no longer use drag and drop blocks in the VPE to edit that playbook. Any subsequent edits in the VPE are only possible by editing the full playbook. This is not recommended.

Splunk SOAR (On-premises) also uses a Git repository to publish company-authored playbooks for customers to download. This repository is called the community repository and is configured on Splunk SOAR (On-premises) by default. You can restore this repository if you accidentally remove it. See [Restore the community playbook repository](#).

You can transfer playbooks to Git using HTTP, HTTPS, or Git. Other protocols can be authenticated or anonymous if supported by the server.

Access the source control settings in Splunk SOAR (On-premises)

To access the Splunk SOAR (On-premises) source control settings, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Source Control**.

You can also access the source control settings from any Playbooks page by clicking **Manage source control**.

Set up a playbook repository using HTTP, HTTPS, or Git

To set up a Git repository using HTTP, HTTPS, or Git protocols, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Source Control**.
3. Select **Configure a new repository** from the Repositories drop-down list.
4. Provide a repository URL, repository name, and branch name. The repository name can be any name that describes your repository.
5. For HTTP and HTTPS, specify a username and password. Splunk SOAR (On-premises) attempts to connect anonymously if no username or password is provided. When crafting the URI, Splunk SOAR (On-premises) converts `https://server...` to `https://username:password@server....`. The Git protocol is not authenticated and does not require a username or password.
6. Click **Save Changes**.

A repository that is added to Splunk SOAR (On-premises) can't be edited. If you need to make a change, delete the repository and then add it again.

The username and password strings are separated so that the password can be encrypted and stored and not displayed to other administrators. However, passwords are stored as clear text in the Git configuration file for that repository.

Git hooks and the SOAR Playbook Editor

Splunk SOAR (On-premises) does not directly support Git hooks. If you choose to use git hooks in your system, be aware of the following:

- There is a risk that the playbook editor will not be able to save or push changes because the Git configuration rejects a commit.
- To avoid this issue, direct Splunk SOAR (On-premises) to push to a staging repository or branch that will not reject pushes. This prevents the playbook editor from being blocked from saving and pushing changes. Handle merge conflicts or other issues manually when pushing from the staging repository to the original repository.

If Git remote rejects your commits, causing the push to fail, you have two options:

- Delete and recreate
 1. Delete the repository and recreate it in Splunk SOAR (On-premises). The playbook reverts to the last successful push and removes all changes made after the last successful push.
 2. Recreate your changes and try to push again.
- Remediate the issue from the command line. Continue reading the next section.

To remediate the issue from the command line, follow the set of the instructions that matches your scenario.

You must have some expertise with Git or refer to Git documentation to perform some of the steps described in this section.

Problem commit is your most recent commit

If your most recent local commit is causing the problem, and you have not made subsequent commits, choose one of the following options:

Option 1:

1. Delete the most recent commit by running `git reset --hard HEAD~1`.
2. Recreate your changes and try to push again.

Option 2:

1. Make the necessary changes in your repository so it will pass the hook.
2. Replace your previous commit with this current state of the repository by running `git commit --amend`
3. Try to push again.

Problem commit is not your most recent commit

Choose the option that best matches your scenario:

Option 1:

If your playbook editor is blocked from pushing to the remote repository, follow these steps:

1. Delete the repository and recreate it in Splunk SOAR (On-premises). The playbook reverts to the last successful push and removes all changes made after the last successful push.
2. Recreate your changes and try to push again.

Option 2:

Otherwise, perform an interactive rebase in Git by following these steps:

1. Start the interactive rebase by running `git rebase -i <identifier>~1`
 - The identifier refers to the working version of the branch, such as `origin/master` or the commit hash for the last working commit.
 - Append `~1` to start the rebase on the commit before the last working version.
- Git displays a file that lists all of the commits in order. Choose an option:
 - *To delete the problematic commit entirely*, delete the line with the commit, while keeping the subsequent commits.
 - *To amend the problematic commit*, modify line for that commit, replacing `pick <hash>` with `edit <hash>`. After you write and close this file, git starts rebasing. Git pauses when it reaches that line. Then you can use `git commit --amend` to repair the commit.
- After the rebase completes successfully, push to the repository. If performing the rebase caused the commit history to diverge, you must perform a force push.

Set up a playbook repository using SSH

To set up a playbook repository using SSH, perform the following steps:

1. From the main menu, select **Administration**.
2. Select **Administration Settings > Source Control**.
3. Select **Configure a new repository** from the Repositories drop-down list.
4. Provide a repository URL starting with **ssh://** and including the username. For example:
`ssh://<username>@10.4.5.6/opt/repos`
5. Add the SSH public key from Splunk SOAR (On-premises) to your Git server's authorized keys file.
 1. Copy the contents in the **SSH Public Key** field.
 2. Log in to your Git server as a user with permissions to edit the Git server's `authorized_keys` file.
 3. Add the SSH public key to the authorized key file, such as `~/.ssh/authorized_keys`.
6. Provide a repository name and branch name. The repository name can be any name that describes your repository.

If you get the following error when setting up an external repo with SSH Auth:

```
Cmd('git') failed due to: exit code(128) cmdline: git fetch -v origin stderr: 'fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.'
```

This indicates that the `/home/<phantom_user>/.ssh/known_hosts` file is not being updated with the external repo and ssh key info.

You can manually correct this through the CLI by running the following command as the Splunk SOAR (On-premises) user:

```
ssh-keyscan -t rsa <external_repo> >> /home/<phantom_user>/.ssh/known_hosts
```


Establish trust with the git repository from your Splunk SOAR (On-premises) deployment

You must inform git that it can trust the remote hosts of your Splunk Phantom deployment before you can use the source repository.

On your Splunk SOAR (On-premises) instance, or in the case of a cluster, on each Splunk SOAR (On-premises) cluster node:

1. SSH to the Splunk SOAR (On-premises) instance or cluster node. Log in as the user account that runs Splunk SOAR (On-premises).

```
ssh phantom@<phantom instance or cluster node>
```

2. Run the command to establish trust with the git repository.

```
git ls-remote git@<address of the git repository>
```

3. Verify that the information returned is correct. Example:

```
git ls-remote git@your-git-repository:phantom/phantom.git
```

4. If the returned values are correct, type **yes**.

Use repositories from the Playbooks page

You can make use of configured repositories on the Playbooks page. See View the list of configured playbooks for more information.

Restore the community playbook repository

The community playbook repository is a collection of playbooks vetted by the Splunk SOAR (On-premises) community. This repository is configured by default when Splunk SOAR (On-premises) is installed. Follow the procedure to restore the community repository if it is accidentally altered or deleted.

1. From the **Home** menu, select **Administration**.
2. Select **Source Control**.
3. In the Repositories drop-down list, select **Configure a new repository**.
4. In the Repo URL field, type the URL: `https://github.com/phantomcyber/playbooks.git`
5. In the Repo Name field, type **community**.
6. In the Branch Name field, enter the version of Splunk SOAR (On-premises) you are running, up to the second set of digits. For example, if you are running version 4.10.3 enter **4.10** in this field.
7. Check the **Read Only** check box.
8. Click **Save Changes**.

Customize email templates in Splunk SOAR (On-premises)

Customize email templates in Splunk SOAR (On-premises) by inserting real-time information into the emails using special variables. For example, to use the name of the incident in the email, use the `{name}` variable where you want the incident name to appear. Variables can be used in both the subject and body of the email.

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Email Settings**.
3. Select a template from the drop-down list. Templates provided by default are **New Incident Assigned** and **Approvals**.
4. Modify the email template for your use. You can use the variables listed in the following table.

The term **container** refers to the type of object generating the email. Incidents are the only container used for generating emails. See [Add and configure apps and assets to provide actions in Splunk SOAR \(On-premises\)](#) for more information about containers.

Variable	Description
{name}	The name of the container or incident.
{label}	The label of the container, such as "incident" or "vulnerability," which is configured on the asset.
{container_url}	The URL to view the container.
{first_name}	The first name of the user being notified.
{from_first_name}	The first name of the user who was the previous owner.
{from_email}	The email address of the previous owner. This is not a template, but can be configured in settings.
{due_time}	The due time of the container in the respective time zone.
{severity}	The severity of the container, such as high, medium, or low.
{your_expired_containers}	The details of the expired containers assigned to the user.
{your_expiring_containers}	The details of the containers assigned to the user that are about to expire.
{your_closed_containers}	The details of the containers assigned to the user that have been closed.
{all_expired_containers}	The details of all containers that have expired.
{all_expiring_containers}	The details of all containers that are about to expire.
{all_closed_containers}	The details of all containers that have been closed.
{task_count}	The amount of tasks assigned to you.
{task_list}	The list of tasks associated with the case.
{phase}	The case management phase associated with the task.
{ownership_type}	Denotes the owner type as either user or role.
{invitee_first_name}	The first name of the person receiving the email.
{inviter_first_name}	The first name of the person sending the email.
{user_message}	A custom message that can be written and added as part of the notification.
{from_first_name}	The name of the person the incident was reassigned to.
{action_name}	The name of the action that will be run on the asset.
{action_executor}	The rule name or name of the user running or executing the action.
{asset_name}	The name of the asset.
{user_owner_type}	This denotes whether the owner is the primary or secondary approver.
{approval_due_time}	The time in which the action to be run on an asset must be approved by.
{approval_url}	Use this URL to navigate to a place where you can approve, deny, delegate or change the action parameters.
{approval_message}	A custom message that can be added to a manual action sent with the approval request.
{task_name}	The name of an assigned task.

Configure search in Splunk SOAR (On-premises)

Splunk SOAR (On-premises) uses an embedded, preconfigured version of Splunk Enterprise as its native search engine. Your organization might want to use a different Splunk Enterprise deployment with Splunk SOAR (On-premises) or use an external Elasticsearch instance.

Configure Splunk SOAR (On-premises) to use an external Splunk Enterprise or Splunk Cloud Platform instance for search

This table summarizes the available options for configuring a Splunk Enterprise or Splunk Cloud Platform instance for search in Splunk SOAR (On-premises).

Search Option	Description
Embedded Splunk Enterprise Instance	This is the default. No additional configuration is required.
External Standalone Splunk Enterprise Instance	<p>Use this option to connect your Splunk SOAR (On-premises) instance or cluster to a single, external instance of Splunk Enterprise or Splunk Cloud Platform.</p> <p>This option requires the Splunk App for SOAR.</p> <ol style="list-style-type: none">1. See Check prerequisites for Splunk App for SOAR in the <i>Install and Configure Splunk App for SOAR</i> manual to verify version compatibility and requirements.2. See Set up remote search on a standalone Splunk Cloud Platform or Enterprise instance in the <i>Install and Configure Splunk App for SOAR</i> manual for instructions.
External Distributed Splunk Enterprise Instance	<p>Use this option to connect your Splunk SOAR (On-premises) instance or cluster to a Splunk Enterprise or Splunk Cloud Platform deployment that contains one or more search heads, or one or more indexers with or without a search head cluster or indexer cluster.</p> <p>This option requires the Splunk App for SOAR.</p> <ol style="list-style-type: none">1. See Check prerequisites for Splunk App for SOAR in the <i>Install and Configure Splunk App for SOAR</i> manual to verify version compatibility and requirements.2. See Set up remote search on a distributed Splunk Cloud Platform or Enterprise instance in the <i>Install and Configure Splunk App for SOAR</i> manual for instructions.

Clustered deployments of Splunk SOAR require an external Splunk Enterprise, as either a single instance or a distributed deployment, or a Splunk Cloud Platform deployment.

Integrating with Splunk Cloud Platform requires the following additional information and actions:

- You must use a public certificate from a verified or trusted certificate authority (CA).
- You must contact Splunk Customer Support for assistance with Splunk Cloud Platform integration. You will need to provide the path to your certificate and your CA.
- You must enable certificate verification on your Splunk SOAR (On-premises) assets.

Splunk SOAR (On-premises) also provides support for an external Elasticsearch instance for single-instance deployments of Splunk SOAR (On-premises). Clustered deployments of Splunk SOAR (On-premises) cannot use Elasticsearch as their search endpoint. See [Configure Splunk SOAR \(On-premises\) to use an external Elasticsearch instance](#)

Configure Splunk SOAR (On-premises) to use an external Elasticsearch instance for search

When you configure Splunk SOAR (On-premises) to use an external instance of Elasticsearch, a copy of all indexed and searchable data is sent to the Elasticsearch instance. The embedded Splunk Enterprise remains active and is used as the search provider for searches in the Splunk SOAR (On-premises) web interface.

Verify the following requirements before configuring the external Elasticsearch instance:

- If you are using SSL to secure your connection to the Elasticsearch instance, the SSL certificate is imported to the Splunk Phantom certificate store.
- You know the host name and port for the Elasticsearch instance.
- You know the username and password of an Elasticsearch user account, or the client certificate and client key.

Perform the following tasks to connect to an external Elasticsearch instance:

1. From the main menu in Splunk SOAR (On-premises), select **Administration**.
2. Click **Administration Settings**.
3. Click **Search Settings**.
4. From **Search Endpoint**, select the radio button for **External Elasticsearch Instance**.
5. Select the **Use SSL** check box to enable SSL.
6. If your Elasticsearch instance is version 6 or newer, select the **Use one index per section** check box.
7. Type the host name in the **Host** field.
8. Type the port number in the **Port** field.
9. Choose your authentication method, either basic authentication with a username and password, or a client certificate.
10. If you are using basic authentication with a username and password:
 1. Type the username of the authorized Elasticsearch account in the **Username** field.
 2. Type the password of the authorized Elasticsearch account in the **Password** field.
11. If you are using certificate-based authentication, select the **Client Authentication** check box.
 1. Type the path to the client certificate in the **Client Certificate** field. This certificate is often a file with the .pem extension.
 2. Type the path to the client key in the **Client Key** field. This key is often a file with the .key extension.
12. Test the connection to your Elasticsearch instance by clicking **Test Connection**.
13. When you are finished, click **Save Changes**.

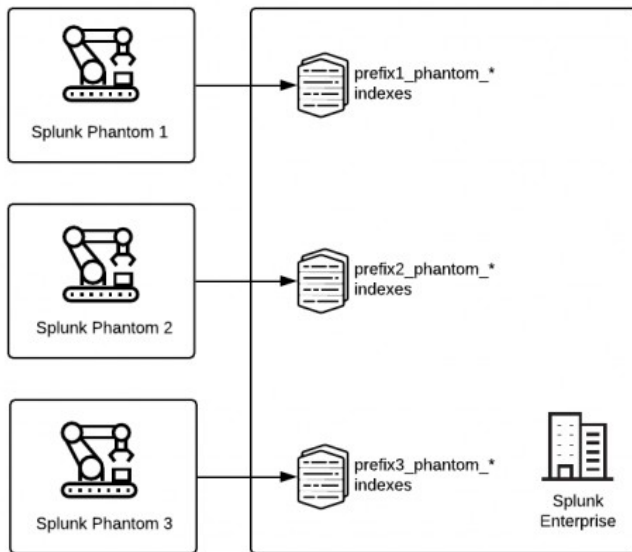
If you want to use a client certificate to connect to your Elasticsearch instance, provide the paths on the Splunk SOAR instance's operating system to the public and private keys. The private key, often a file with the .pem extension, is the Client Certificate. The public key, often a file with the .key extension, is the Client Key. Both files must be readable by the nginx user. You can store the files in the nginx user's home directory, /var/cache/nginx.

Reindex data to make newly added information searchable

There are some situations where data coming in to Splunk SOAR (On-premises) can't be indexed, and therefore can't be searched. You can reindex information sections to make this information searchable. See Reindex all indexes to search for the data created while using the new prefixes. See Reindex data in the *Install and Configure Splunk App for SOAR* manual.

Define a custom index per Splunk SOAR (On-premises) instance

If you have multiple Splunk SOAR (On-premises) instances in your environment, you can append a custom prefix to the index created on the Splunk platform. Use the custom prefix to create separate indexes for each Splunk SOAR (On-premises) instance, which provides data separation and the ability to correlate each index with the appropriate Splunk SOAR (On-premises) instance.



Define a custom prefix with a standalone external Splunk platform deployment

Perform the following tasks on each Splunk SOAR instance to create a custom prefix for each instance with a standalone external Splunk platform deployment for search:

1. Verify that your Splunk SOAR (On-premises) instance is connected to the Splunk platform by setting up the search settings using a standalone external Splunk instance:
 1. Follow the instructions in Configure the service with Splunk App for SOAR in *Install and Configure Splunk App for SOAR*.
 2. Make sure to click **Test Connection** at the end of the procedure and verify that Splunk SOAR (On-premises) and the Splunk platform are connected.
2. Log in to the Splunk SOAR (On-premises) instance as the **root** user. In unprivileged environments, run the script as the specific user configured to run Splunk SOAR (On-premises).
3. On each Splunk SOAR (On-premises) instance, run the `set_preference` command: `phenv set_preference --splunk-index-prefix=<prefixstring> --splunk-admin-username <splunkadminusername>`

For example, to set a custom prefix called **prefix1** using **admin** as the admin user for the Splunk platform:

```
phenv set_preference --splunk-index-prefix="prefix1" --splunk-admin-username admin
```

Use an empty prefix string to remove a custom prefix. For example:

```
phenv set_preference --splunk-index-prefix="" --splunk-admin-username admin
```

In Splunk SOAR clusters, the script updates the prefix for all nodes in the cluster.

4. Users on the Splunk platform inherit index permissions from their roles. After creating the new indexes, you can update roles to give all users in the role access to the new indexes, or create new users and new roles to give access to the new indexes. This example shows how to edit the **phantomsearch** and **phantomdelete** roles to grant users access to the new indexes.
 1. From Splunk Web, select **Settings > Roles**.
 2. Click the name of the role you want to edit, such as **phantomsearch**.
 3. Click the **Indexes** tab.
 4. Check the boxes next to the names of the new indexes.
 5. Click **Save**.
 6. Perform this procedure again to grant access to the new indexes for the **phantomdelete** role.
5. If you need additional custom roles to manage only the new indexes this example shows how to create them.
 1. From Splunk Web, select **Settings > Roles**.
 2. Click **New Role**.
 3. Type a name for the role.
 4. On the **Inheritance** tab, select the existing role you want your new role to inherit from, such as **phantomsearch**.
 5. Click the **Indexes** tab.
 6. Check the boxes next to the names of the new indexes.
 7. Uncheck the boxes next to the names of the indexes the new role should not be able to access.
 8. Click **Create**.
 9. Click the name of the role you want to edit, such as **phantomsearch**.
 10. Click the **Indexes** tab.
 11. Uncheck the boxes next to the names of the new indexes. This will prevent items managed by the new role from being repeated in indexes by **phantomsearch**.
 12. Click **Save**.
 13. Perform this procedure again to create a new role with access to the new indexes for the **phantomdelete** role. Custom roles used for deletions must inherit permissions from the **phantomdelete** role.
6. After the prefix is created, update the Splunk administration for the HEC token to grant access to the new indexes. See Set up the HTTP Event Collector on the standalone Splunk platform instance in the *Install and Configure Splunk App for SOAR* manual for instructions.
7. Perform this step if you are using a Splunk SOAR (On-premises) cluster. Run the following commands on each node in your Splunk SOAR (On-premises) cluster:

```
pkill --full add_to_searchindex
<PHANTOM_HOME>/bin/phsvc restart uwsgi
```
8. Reindex all indexes to search for the data created while using the new prefixes. See Reindex data in the *Install and Configure Splunk App for SOAR* manual.

Define a custom prefix with a distributed external Splunk platform deployment

Perform the following tasks on each Splunk SOAR (On-premises) instance to create a custom prefix for each instance with a distributed external Splunk platform deployment for search:

The custom prefix script is not supported for use with distributed Splunk platform deployments which are built in the Splunk Cloud Platform.

1. Verify that your Splunk SOAR (On-premises) instance is connected to the Splunk platform by setting up the search settings using a distributed external Splunk instance:

1. Follow the instructions in Set up remote search on a distributed Splunk Cloud Platform or Enterprise instance in the *Install and Configure Splunk App for SOAR* manual.. The Splunk App for SOAR must be installed on all search heads in the cluster.
2. Make sure to click **Test Connection** at the end of the procedure and verify that Splunk SOAR (On-premises) and the Splunk platform are connected.
2. Log in to the Splunk SOAR (On-premises) instance as the **root** user. In unprivileged environments, run the script as the specific user configured to run Splunk SOAR (On-premises).
3. On each Splunk SOAR (On-premises) instance, run the `set_preference` command: `phenv python set_preference --splunk-index-prefix=<prefixstring> --splunk-admin-username <splunkadminusername>`

For example, to set a custom prefix called **prefix1** using **admin** as the admin user for the Splunk platform:

```
phenv python set_preference --splunk-index-prefix="prefix1" --splunk-admin-username admin
```

Use an empty prefix string to remove a custom prefix. For example:

```
phenv python set_preference --splunk-index-prefix="" --splunk-admin-username admin
```

In Splunk SOAR clusters, the script updates the prefix for all nodes in the cluster.

Below is sample output from the command run in a Splunk SOAR (On-premises) unprivileged cluster with a distributed Splunk Enterprise deployment:

```
[phanru@phantom ~]$ phenv set_preference --splunk-index-prefix prefix1 --splunk-admin-username admin
Are you sure you wish to apply search index prefix prefix1 for this Phantom instance [yes/no]? yes
Proceeding ... index configuration stored: /home/phanru/phantomcyber/tmp/indexes.conf
Done! Next steps:
- indexes.conf must be updated via splunk cluster master node.
- On Splunk platform, edit permissions to allow the current or new HEC token to access new indexes.
- On Splunk platform, edit permissions to allow the current or new search/delete users to access new indexes.
- If new HEC token or users are created, update the Phantom search settings.
Run `pkill --full add_to_searchindex` on each Phantom cluster node
Run `/home/phanru/phantomcyber/bin/phsvc restart uwsgi` on each Phantom cluster node
- Rerun Test Connection.
- All phantom search indexes must now be re-indexed.
```

Note the location of the new `indexes.conf` file created by the script. You will need this information in the next step.

4. Edit and save the contents of the new `indexes.conf` file that was created by the `phenv set_preference --splunk-index-prefix` command. In our example, we can use `cat` to view and copy the contents of the `<PHANTOM_HOME>/tmp/indexes.conf` file.
5. In the master node of the Splunk search head cluster, append the contents of the new `indexes.conf` file to the local `indexes.conf` file on the master node, such as `/opt/splunk/etc/master-apps/_cluster/local/indexes.conf`.
6. Run the following commands to push the new `indexes.conf` to the other indexers in the cluster and verify:

```
/opt/splunk/bin/splunk apply cluster-bundle --answer-yes
/opt/splunk/bin/splunk show cluster-bundle-status
```
7. Users on the Splunk platform inherit index permissions from their roles. After creating the new indexes, you can update roles to give all users in the role access to the new indexes, or create new users and new roles to give access to the new indexes. This example shows how to edit the **phantomsearch** and **phantomdelete** roles to grant users access to the new indexes.

1. From Splunk Web, select **Settings > Roles**.

2. Click the name of the role you want to edit, such as **phantomsearch**.
3. Click the **Indexes** tab.
4. Check the boxes next to the names of the new indexes.
5. Click **Save**.
6. Perform this procedure again to grant access to the new indexes for the **phantomdelete** role.
8. If you need additional custom roles to manage only the new indexes this example shows how to create them.
 1. From Splunk Web, select **Settings > Roles**.
 2. Click **New Role**.
 3. Type a name for the role.
 4. On the **Inheritance** tab, select the existing role you want your new role to inherit from, such as **phantomsearch**.
 5. Click the **Indexes** tab.
 6. Check the boxes next to the names of the new indexes.
 7. Uncheck the boxes next to the names of the indexes the new role should not be able to access.
 8. Click **Create**.
 9. Click the name of the role you want to edit, such as **phantomsearch**.
 10. Click the **Indexes** tab.
 11. Uncheck the boxes next to the names of the new indexes. This will prevent items managed by the new role from being repeated in indexes by **phantomsearch**.
 12. Click **Save**.
 13. Perform this procedure again to create a new role with access to the new indexes for the **phantomdelete** role. Custom roles used for deletions must inherit permissions from the **phantomdelete** role.
9. After the prefix is created, update the Splunk administration for the HEC token to grant access to the new indexes. See Set up the HTTP Event Collector on the distributed Splunk Cloud Platform or Enterprise instance in the *Install and Configure Splunk App for SOAR* manual for instructions.
10. Perform this step if you are using a Splunk SOAR (On-premises) cluster. Run the following commands on each node in your Splunk SOAR (On-premises) cluster:


```
pkill --full add_to_searchindex
<PHANTOM_HOME>/bin/phsvc restart uwsgi
```
11. Reindex all indexes to search for the data created while using the new prefixes. See Reindex data in the *Install and Configure Splunk App for SOAR* manual.

Use a custom prefix when you want to change your Splunk platform instance

If you have a situation where you want to use the same custom prefix on your Splunk SOAR (On-premises) instance with a different or new Splunk platform instance, perform the following tasks:

1. Follow the instructions in either Set up remote search on a standalone Splunk Cloud Platform or Enterprise instance or Set up remote search on a distributed Splunk Cloud Platform or Enterprise instance in the *Install and Configure Splunk App for SOAR* manual to connect your Splunk SOAR (On-premises) instance with the Splunk platform.
2. Run the `set_preference` command to create the new prefix.
3. Update the Splunk administration for the HEC token to grant access to the new indexes.
4. Reindex all indexes to search for the data created while using the new prefixes.

Configure Google Maps for visual geolocation data

The MaxMind app provides a `geolocate_IP` action that uses Google Maps functionality to show a world map with a marker indicating the approximate location of the IP under investigation. You must provide a Google Maps API key to enable this functionality. See the Maps JavaScript API page in the Google Maps Platform documentation for more information about

obtaining a Google Maps API key.

After obtaining an API key, perform the following steps:

1. From the **Home** Menu, select **Administration**.
2. Select **Administration Settings > Google Maps**.
3. Enter your API key into the field.
4. Click **Save Changes**.

With a proper API key applied, MaxMind Geolocate IP displays a map with searches.

Run playbooks in parallel with vertical scaling

Splunk SOAR (On-premises) supports vertical scaling for playbook execution.

The DECIDED daemon now spawns a number of runners when started. Each runner is a dedicated instance of the Python 3 environment. The default is four runners.

When you upgrade Splunk SOAR (On-premises) from a lower release, the number of runners you have set is not changed.

When you set multiple runners in **Main Menu > Administration > Administration Settings > Playbook Execution**, DECIDED spawns that many runners for each Python version when it starts.

Playbooks and custom functions are normally run by a single runner in a single queue. A playbook or custom function must be completed before the runner executes the next playbook or custom function in sequence. By defining additional runners, playbooks and custom functions can be assigned to different runners to increase the number of playbooks which are executed at once.

When a playbook run is started, the DECIDED daemon assigns the playbook to an available runner. Playbooks are assigned to an available runner as follows:

1. Playbooks are assigned to available runners in a round-robin fashion.
2. All the blocks of a playbook and child playbooks are run by the same runner.

Differences in behavior from a single runner vs. multiple python runners

Some Splunk SOAR (On-premises) behaviors are different when multiple python runners are enabled.

- The playbook API **save_data** may return incorrect results when playbooks are run in parallel if the key:value pairs are not unique across playbook runs. Use the **save_object()** API instead of the **save_data()** API.
- The playbook API **save_object** may return incorrect results if the same playbook is run against the same container multiple times. Use the optional **playbook_name** and **container_id** parameters with **save_object** to make sure that saved objects are unique across multiple runs of the same playbook. If you need to save information specifically about the playbook run, use the **save_run_data()** and **get_run_data()** APIs.
- When the number of playbook runners is increased some deployments may reach the maximum size defined for `decided.log` more quickly than with the default number of runners. Splunk SOAR (On-premises) administrators may want to increase the settings for logrotate on their deployments to values higher than `rotate 10` and `size 50M`.

Use local variables instead of global variables

When creating or editing playbook source code, it is better to use local variables than global ones. Values stored in a global variable may be modified by another instance of the playbook, a child playbook, or another process that uses the same variable resulting in unexpected or incorrect results.

Exchange data between playbooks or blocks when you have multiple python runners

When you have multiple Python runners, you must exercise greater care around data consistency in your playbooks. When there is only a single Python runner, playbooks run in series so no other playbook will attempt to access or modify information during the playbook run. When multiple Python runners are operating, multiple playbooks could be using the same variables or objects, so data consistency is more difficult to guarantee. Using local variables, the correct APIs for passing data between playbook runs, and implementing a locking method helps ensure your Splunk SOAR (On-premises) data is consistent and correct.

You must implement a locking solution, such as `NamedAtomicLock`, to ensure data consistency. See `NamedAtomicLock` on pypi.org for details. Use locking carefully. Incorrectly implementing locking can lead to problems or hang a playbook.

- If you want to share data within the same playbook run use the APIs **save_run_data** and **get_run_data** to exchange information about specific keys. In this case, locking is not required. See `get_run_data` and `save_run_data` in the *Python Playbook API Reference for Splunk SOAR (On-premises)*.
- If you want to share data between playbook runs with the same playbook name, between playbook runs that operate on the same container, or across all playbook runs, then you should use the APIs **clear_object**, **get_object**, and **save_object**. In each of these cases, locking is required. See `get_object`, `save_object`, and `clear_object` in the *Python Playbook API Reference for Splunk SOAR (On-premises)*.
- If all you want to do is read data from a container in a playbook, then no locking is required.
- When exchanging data between playbook runs, remember that the order in which playbooks are run is not guaranteed. If you need to ensure results from a specific playbook are available, call it as a dependent playbook.

DECIDED settings for vertical scaling

You configure DECIDED for vertical scaling by changing the settings in **Main Menu > Administration > Administration Settings > Playbook Execution**.

Changing these settings restarts DECIDED and causes any running playbooks to fail. It is better to change these settings only when there are no active playbooks.

Code Block Execution Time Out (in minutes) Set the number of minutes a playbook code block or custom function can run before DECIDED will stop the runner and spawn a replacement runner. Any playbook that hits this limit without completing will be marked as failed.

Number of Python 3 Runners Set the number of Python 3 runners. For Splunk SOAR (On-premises) the default is 4 runners. The minimum is one runner. The maximum is 10. This number of runners will be created when DECIDED starts and will be active even if no playbooks or custom functions are assigned to them.

These settings can also be modified by using the REST API. See [REST System Settings](#).

When to add more Python runners

By default, Splunk SOAR (On-premises) starts with four runners for Python 3 and is designed to support up to 10 runners. Because every deployment is unique, and the factors that influence performance are varied, there are no hard rules for when, or by how many to increase the number of runners for your deployment.

When deciding whether or not to add more runners, some factors that influence performance are:

- Number and kind of actions performed in your playbooks.
- Number of child playbooks or custom functions executed by playbooks.
- Actions that require responses from assets or external services.
- Available CPU resources.

If your Splunk SOAR (On-premises) deployment is queuing playbooks to run, and your hardware or virtual machine still has unused CPU capacity (such as idle cores, or low core usage percentages) you should consider increasing the number of playbook runners.

- Increase the number of runners by one and measure performance before adding additional runners. Repeat this until you either achieve the performance gains desired, reach the maximum number of runners, or encounter resource limits.

When you increase the number of Python runners you can see a decrease in the length of time it takes to complete a playbook. Many deployments can expect to see gains by adding between one and four more runners, with gains from adding additional Python runners tapering off after a total of five runners.

Not all playbooks and deployments are the same. Your results may vary based on the number of playbooks, the kinds of actions or processing each playbook is doing, the amount of CPU cores available to Splunk SOAR (On-premises), and other effects.

Manage your organization's credentials with a password vault

Use credential vaults to centrally manage and monitor credential usage in your organization. Splunk SOAR (On-premises) supports the following password vaults:

- Hashicorp Vault
- CyberArk Enterprise Password Vault
- Thycotic Secret Server

As an administrator, you can configure Splunk SOAR (On-premises) to retrieve credentials from these vaults and use them in assets or use them as a client to other identity providers such as LDAP and OpenID.

Use Hashicorp Vault with Splunk SOAR (On-premises)

Splunk SOAR (On-premises) supports Hashicorp Vault's KV store REST API version 2.

To use Hashicorp Vault with Splunk SOAR (On-premises), perform the following steps:

1. From the main menu, select **Administration**.
2. Select **Administration Settings > Password Vault**.
3. Get the URL and Token from your Hashicorp administrator.

4. Select the **Verify server certificate** checkbox to verify that the HTTPS certificate is trusted. If the certificate is not trusted by default, see [Manage the Splunk SOAR \(On-premises\) certificate store](#) for information about adding your own trusted certificate.
5. Click **Save Changes**.

Once you have Hashicorp access configured, you need to know the paths and names of the secrets you want to use from the Hashicorp Vault. You can use Hashicorp to supply credentials under OpenID and LDAP authentication configuration and with assets.

Use Hashicorp to provide credentials during authentication configuration

You can use Hashicorp to automatically supply credentials under OpenID and LDAP authentication configuration.

1. From the main menu, select **User Management**.
2. Select **Authentication**.
3. Select an identity provider such as **LDAP**.
4. Toggle the **LDAP** switch to enable LDAP authentication.
5. Check the **Manage password using Hashicorp Vault** check box.
6. Provide the value and key you want to retrieve from the vault.
7. (Optional) Click **Test Authentication** to verify authentication.
8. Click **Save Changes**.

Use Hashicorp to provide credentials with assets

You can use Hashicorp to automatically supply credentials when working with assets.

1. From the main menu, select **Apps**.
2. In the list of apps, find one to configure such as the Palo Alto Networks Firewall and click **Configure New Asset**.
3. Open the **Asset Settings** tab for that asset.
4. Click **Advanced** to expand the advanced configuration section.
5. In the Credential Management section, select the fields you want to get from Hashicorp Vault, and the path and key to use. For example, you can specify **/secret/autofocus** in the Path field and **apikey** in the Key field to retrieve an API key used to authenticate to the AutoFocus service.
6. Click **Save**.

Use CyberArk with Splunk SOAR (On-premises)

Integrate Splunk SOAR (On-premises) with CyberArk's Vault feature to retrieve passwords or other fields for assets. This allows you to utilize CyberArk account management features to change passwords on managed products and services without having to manually update Splunk SOAR (On-premises) assets after a password change.

For security purposes, utilizing CyberArk can greatly simplify password management but may not significantly change the security stance of the Splunk SOAR (On-premises) server. Splunk SOAR (On-premises) would no longer be the primary store for CyberArk-managed account passwords, but still has the ability to retrieve the same passwords from CyberArk in order to authenticate itself to other resources. Therefore, someone with administrative control over the Splunk SOAR (On-premises) server can gain access to those passwords.

Installing CyberArk on the Splunk SOAR (On-premises) server must be performed by a CyberArk administrator following the CyberArk documentation. Splunk SOAR (On-premises) was tested with the `CARKaim-9.70.0.3.x86_64.rpm` CyberArk installer package.

Perform the following tasks to use CyberArk with Splunk SOAR (On-premises):

1. From the main menu, select **Administration**.
2. Select **Administration Settings > Password Vault**.
3. Select **Cyberark** from the drop-down list in the Manager field. The CyberArk option in the drop-down list is inactive until the CyberArk components are installed. Splunk SOAR (On-premises) determines the presence of CyberArk in your environment by looking for the `/opt/CARKaim` directory.
4. Click **Save Changes**.

After the CyberArk options become visible, check the **Enable credential management at startup** check box to have the `watchdogd` daemon start CyberArk when Splunk SOAR (On-premises) is started. This is useful if you have disabled the system from starting CyberArk by removing the startup file from `/etc/init.d`.

To require a Splunk SOAR (On-premises) administrator to log in to perform an action in Splunk SOAR (On-premises) before CyberArk is available after a system restart, uncheck **Enable credential management at startup** and click **Save Changes**. In this situation, an administrator is someone who has the specific Administrator role. Click **Authorize** to require the logged-in administrative user to supply their own password to re-authenticate themselves, and then the credential management service will be started.

To use CyberArk to automatically supply credentials under authentication configuration, perform the following steps:

1. From the main menu, select **User Management**.
2. Select **Authentication**.
3. Select an identity provider such as **LDAP**.
4. Toggle the LDAP switch to enable LDAP authentication.
5. Check the **Manage password using CyberArk** check box.
6. Fill in the CyberArk Safe, Safe Path, and Object Name fields the same way you do for an Asset to select the CyberArk object that CyberArk is going to use to get the password field value.
7. Click **Save Changes**.

Use Thycotic Secret Server with Splunk SOAR (On-premises)

Splunk SOAR (On-premises) can use Thycotic's API to access secrets managed by Secret Server. Usernames and passwords can be stored in Thycotic Secret Server for both users and assets which require a login to use.

In order for Splunk SOAR (On-premises) to use secrets managed by Thycotic Secret Server you must provide:

- The URL to your organization's Thycotic Secret Server. Depending on your organization's DNS configuration, you may need to include the port number. `https://<your.organization's.secret.server>:<port number>`
- The username and password of the account which will retrieve secrets using the API.
- Optional: The Organization ID set in Secret Server for use in the Thycotic Secret Server API.

These values are used to make an oauth2 token for Thycotic Secret Server. Once authenticated, Splunk SOAR (On-premises) uses the `SearchSecretsByFolder` API to access the managed secrets.

Set the login secret in Thycotic Secret Server

You will need to setup the login information in Secret Server before it can be used to access Splunk SOAR (On-premises). For more information on Thycotic Secret Server, see the documentation on the Thycotic website.

1. Create the required folders.

2. Use the **Create Secret** widget, selecting the template as **Password**.
3. Enter the required items in the mandatory fields of **secret** and **Password**.

Set the Thycotic Secret Server settings in Splunk SOAR (On-premises)

Add the required information to create the oauth2 token for Thycotic Secret Server in Splunk SOAR (On-premises)'s administration settings. This token is for connecting to Thycotic Secret Server.

1. From the Main Menu, select **Administration**.
2. Select **Administration Settings > Password Vault**.
3. Select **Thycotic Secret Server** from the drop-down list in the **Manager** field.
4. Set the URL for your Thycotic Secret Server instance.
5. Specify the username and password Splunk SOAR (On-premises) will use to access secrets.
6. Optional: Set the organization id.
7. Click **Save Changes**.

Add the authentication settings in User Management. These will be the actual secrets for each user or asset. Only LDAP authentication is supported.

1. From the Main Menu, select **Administration**.
2. Select **User Management > Authentication**.
3. Select the **LDAP** tab.
4. Set **LDAP** to **ON**.
5. Add the information for your LDAP provider, server, domain, usernames, and passwords.
6. Check **Manage password using Thycotic Secret Server**.
7. Add the **Folder**, **Key**, and **Thycotic FieldName** that store the Splunk SOAR (On-premises) user credentials.
8. Test your LDAP integration by clicking **Test Authentication**.

For more information about configuring LDAP see [Configure single sign-on authentication for Splunk SOAR \(On-premises\)](#).

If you have assets which require logins and those logins are managed by Thycotic Secret Server, then you need to set credential management in the asset's configuration, in **Apps > <Asset Name> > Asset Settings > Advanced**.

Set global environment variables

You can set environment variables that apply globally across the Splunk SOAR (On-premises) runtime environment to manage proxies or other features. You can also override or provide these variables on a per-app basis in the app advanced configuration. Changes to global environment settings will not be applied until the Splunk SOAR (On-premises) platform is restarted.

To make changes to the global environment:

1. From the Splunk SOAR (On-premises) main menu, select **Administration**.
2. Click **Administration Settings > Environment Settings**.
3. Click **+Variable** to add a new environment variable.
4. In the **Name** field, specify **HTTP_PROXY**, **HTTPS_PROXY**, or **NO_PROXY** depending on the type of proxy connection. These environment variables are read by all Splunk SOAR (On-premises) processes and affect the entire product including external search connections, app and asset connections, and requests made from within playbooks.

5. In the **Value** field, include the following depending on the type of proxy configuration. Wildcards are not supported.
 1. HTTP and HTTPS proxy configurations: protocol, hostname or IP address, and the port of the proxy server. For example, `<protocol>://<hostname/IP>:<port>`
 2. NO_PROXY configurations: IP address, hostname, or domain of the asset.
 3. (Conditional) If the proxy server requires authentication, consider the following items:
 - `<scheme>://[<username>[:<password>]]@<host>[:<port>]` is the scheme (http or https), optional username and password, host name or IP address, and optional port number used to connect to the proxy server.
 - The scheme and host are required.
 - If using a proxy server that requires authentication Splunk SOAR (On-premises) may need a service account on the proxy server.
 - If authentication credentials (username/password) are specified, the "secret" box should be selected so that the username and password are stored in encrypted format.
 - If port is not specified it defaults to port 80 when the scheme is http, and port 443 when the scheme is https.
6. Check **Secret** to encrypt the **Value** field and stop it from being displayed.

When configuring the system to use an HTTP or HTTPS proxy, Splunk SOAR (On-premises) requires that you except calls to the loopback interface from the proxy list. You must set the environment variable "NO_PROXY" to include 127.0.0.1, localhost, and localhost.localdomain so that REST calls can be made on the loopback interface without being diverted to the proxy.

Apply environment variables to individual assets

You can also apply environment variables to configured assets individually. The asset environment variables take precedence over global environment variables. For more information, see [Configure environment variables for a Splunk SOAR \(On-premises\) asset](#).

Multi-tenancy and environment variables

When multi-tenancy is enabled, you can choose to set specific environment variables per tenant. To set specific environment variables per tenant, select the tenant you want to set the environment variables for in the **Tenant** drop-down menu on the **Environment Settings** screen. For more information on enabling and using multi-tenancy, see [Configure multiple tenants on your Splunk SOAR \(On-premises\) instance](#).

When multi-tenancy is enabled, per-asset variables take precedence over per-tenant variables and per-tenant variables take precedence over global environment variables. When multi-tenancy is not enabled, per-asset environment variables take precedence over global environment variables.

Set the global action concurrency limit

The global action concurrency limit designates the maximum number of concurrent actions across the Splunk SOAR (On-premises) platform.

- The default setting is 150 concurrent actions on the SOAR platform.

When changing the global action limit, ensure the existing action limits set on all of your assets is still within the new global limit. Use caution when changing the global action limit as it can significantly affect performance.

To change the concurrent action limit in Splunk SOAR (On-premises), follow these steps.

1. From the **Home** menu, select **Administration**.
2. Click **Administration Settings > Environment Settings**.
3. Enter your desired action limit in the box. Use caution when changing this limit because doing so can have a significant effect on performance.
4. Click **Save Changes**.
5. After changing this value, Splunk SOAR (On-premises) needs to be restarted for it to take effect.

See also

Concurrent actions limits can be controlled at the app or connector level and on individual assets.

- For information on controlling action concurrency in an app or connector's configuration metadata, see [Action Section: Synchronization in *Develop Apps for Splunk SOAR \(On-premises\)*](#).
- For information on setting concurrent actions for a specific asset, see [Set the concurrent action limit](#).
- For information on disabling action concurrency see, [Disable action lock or action concurrency](#).

Add tags to objects in Splunk SOAR (On-premises)

Add tags to objects in Splunk SOAR (On-premises) to help you perform the following tasks:

- Search for objects in Splunk SOAR (On-premises)
- Flag objects for other users
- Automation and workflow operations
- Affect the flow of playbooks

You can also require tags before a container can be closed. See [Configure how events are resolved](#) for more information.

Required user privileges to view, add, edit, or delete tags in Splunk SOAR (On-premises)

To view the Tags page, a user must have a role with the View System Settings privilege. To add, edit, or delete tags on the Tags page, a user must have a role with the Edit System Settings privilege.

Editing the tags on individual containers, artifacts, or assets requires a role with the matching Edit Containers, Edit Artifacts, or Edit Assets privileges. However, a user with the combination of View System Settings and Edit System Settings privileges can use the Tags page to delete or rename tags regardless of the object they are applied to, even without the edit privileges for those objects.

View tags in your Splunk SOAR (On-premises) instance

To view the Tags page, a user must have a role with the View System Settings privilege.

Perform the following steps to access the Tags page and view the existing tags in your Splunk SOAR (On-premises) instance:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Tags**.

Add a new tag to Splunk SOAR (On-premises)

To add a new tag to Splunk SOAR (On-premises), perform the following steps:

1. On the Tags page, click **+ Tag**.
2. Enter a new tag name.
3. Click **Create**.

Tags can be added on individual objects by editing or creating that object in Splunk SOAR (On-premises) and typing them into the Tags field. For example, to create a new tag for a container in Splunk SOAR (On-premises), do the following:

1. Navigate to the container.
2. Click **Event Info** to expand the section.
3. In the Tags field, enter the name of a new tag you want to associate with the container.

Edit existing Splunk SOAR (On-premises) tags

Renaming a tag affects all objects in Splunk SOAR (On-premises) currently using that tag. All containers, artifacts, or assets in Splunk SOAR (On-premises) with the existing tag name are updated to use the new tag name.

To edit an existing tag, perform the following steps:

1. On the Tags page, click the edit icon for the tag. If the existing tag is already in use by another Splunk SOAR (On-premises) component, its usage is summarized in the Edit Tag window. Review this information and make notes of where you must update the tag in Splunk SOAR (On-premises) to keep your playbooks operational.
2. Modify the name of the tag as desired.
3. Click **Save**.

Delete a tag in Splunk SOAR (On-premises)

A tag exists in Splunk SOAR (On-premises) as long as at least one object still uses that tag. If you remove a tag from all objects or delete all those objects, the tag no longer shows on the Tags page. Deleting a tag affects all objects in Splunk SOAR (On-premises) currently using that tag. The deleted tag is removed from all containers, artifacts, or assets in Splunk SOAR (On-premises) currently using the tag.

To delete an existing tag, perform the following steps:

1. On the Tags page, click the delete icon for the tag.
If the existing tag is already in use by another Splunk SOAR (On-premises) component, its usage is summarized in the Delete Tag window. Review this information before you proceed.
2. Click **Delete**.

Create custom CEF fields in Splunk SOAR (On-premises)

Splunk SOAR (On-premises) uses the Common Event Format (CEF). CEF is a system of key-value pairs for important pieces of information about an artifact.

An artifact might have several key pieces of information such as `sourceAddress`, `sourcePort`, `destinationAddress`, `destinationPort`, and a `timestamp`. Each of these is stored in a field.

You can only have one of each CEF field per artifact. For example, you cannot have more than one `sourceAddress` per artifact. If you have a data set that includes multiple `sourceAddress` entries, separate those into multiple artifacts. Each of those artifacts can be placed in the same container.

You can extend or customize CEF to meet your organization's needs by adding custom CEF fields, and then using these fields in Investigation, add them to artifacts with the REST API, or using them in playbooks.

When an artifact is edited from Investigation, values set for a custom CEF appear as indicators. You can view these indicators by selecting **Indicators** in the **Home** menu.

You can add, delete, or modify a custom CEF using the REST API.

Create a custom CEF field

Perform the following steps to create a custom CEF field:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > CEF**.
3. Click **+ CEF**.
4. Type a name for your customized CEF.
5. (Optional) Select a data type for the field from the dropdown list.

Available choices are prepopulated with all enabled Apps actions. You can add your own data type or leave the data type blank. Leaving this blank allows users to enter a value while editing the artifact in Mission Control.

1. Click **Save**.

Modify a custom CEF field


Perform the following steps to modify a custom CEF field:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > CEF**.
3. Click the edit icon to the right of the CEF name.
4. Make the desired changes.
5. Click **Save**.

Delete a custom CEF field

Perform the following steps to delete a custom CEF field:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > CEF**.

3. Click the  icon to the right of the custom CEF field name.

Deleting a custom CEF does not remove it from existing artifacts that have the field applied.

Reset the admin and root passwords in Splunk SOAR (On-premises)

You can reset the passwords for the following accounts to meet your organization's hardening requirements, or if you misplace or forget them:

- The admin user for the Splunk SOAR (On-premises) web interface. This is a default account in Splunk SOAR (On-premises) that can't be deleted. It must always be available so that you can access Splunk SOAR (On-premises) in cases where other authentication methods such as LDAP fail. See [Reset the admin password in Splunk SOAR \(On-premises\)](#).
- The root user for the underlying CentOS Linux operating system. This account is required for maintenance tasks such as upgrades, and is also used to reset the admin password.

Reset the admin password in Splunk SOAR (On-premises)

To reset the admin user password, perform the following tasks:

1. Log in to the operating system with your normal user account.
2. Run the `sudo su` command to switch to the root user.
3. Run the following command in `{phantom_home}/www`:

```
phenv python manage.py changepassword admin
```

4. Enter a new password, then enter it again to confirm. Both passwords must match.
5. To verify, access the Splunk SOAR (On-premises) web interface and log in as the admin user using the new password.

If the admin account has Duo two factor authentication enabled and is no longer working properly, perform the following steps to temporarily disable the two factor authentication:

1. Run the following command as root:

```
phenv set_preference --disable-admin-2fa
```
2. Confirm that you want to disable two factor authentication for the admin account.

Reset the root password in Splunk SOAR (On-premises)

To reset the root password in Splunk SOAR (On-premises), perform the following tasks:

1. Configure the virtual machine to boot from a CD.
2. Mount the virtual machine root disk.
3. Edit the password file.
4. Mark the disk for re-labeling.
5. Set a new password.

Configure the virtual machine to boot from a CD

Perform the following steps to configure the virtual machine (VM) to boot from a CD.

1. Take a snapshot of the VM before performing this kind of recovery operation.
2. Obtain a Linux boot CD ISO that has the LVM tools on it. This has been successfully tested with SystemRescueCd-x86-4.7.2.
3. Configure the VM in your virtualization environment to boot from this ISO image.
4. Once configured, reset the VM so that it reboots.
5. Boot the VM from the CD image.

VMware products typically require that you press a key at the brief BIOS screen to make the VM boot from the CD rather than the virtual hard drive. This might take very careful timing. If you are unable to get it to boot from the CD image by manually pressing the button quickly enough, go to this VMware community page and search for "bios.bootDelay."

1. Follow the prompts for your boot CD until you are able to get to a shell.

Mount the virtual machine root disk

When you have a root shell, perform the following tasks to mount the Splunk SOAR (On-premises) VM drive.

1. Run the `lvscan` command to make sure you can see the LVM drives.
2. Use the following command to mount the drive:

```
mount /dev/VolGroup/lv_root /mnt
```

If your boot CD doesn't have a `/mnt` directory for mounting, substitute an appropriate mount location.

Edit the password file

Perform the following tasks to edit the `/etc/passwd` file:

1. Use a text editor to open the file. For example, to use `vi` type the following at the command line:

```
vi /mnt/etc/passwd
```
2. Find the line for the root user, which looks like the following: `root:x:0:0:root:/root:/opt/phantom/bin/setup`
3. Remove the "x" between the first two colons, so it looks like the following:

```
root::0:0:root:/root:/opt/phantom/bin/setup
```

 The "x" normally tells the operating system to look in `/etc/shadow` for the password hash. Having it blank means root has no password at all.

Mark the disk for relabeling

Because the Splunk SOAR (On-premises) virtual machine uses SELinux, perform the following steps to mark the disk for relabeling:

1. Run the following command to have Linux relabel the drives when they are booted:

```
touch /mnt/.autorelabel
```
2. To make sure the changes are written out, unmount the disk and reboot:

```
umount /mnt  
reboot
```

Set a new root password

To set a new root password, follow these steps:

1. Login as root to the VM console. You will not be prompted for a password.
2. When you are logged in, set a new root password immediately.
3. After setting the password, log out and then log back in with the new password to verify that a password is correct.

Delete containers from your Splunk SOAR (On-premises) deployment

Use the `delete_containers.pyc` script to remove containers from their Splunk SOAR (On-premises) deployment. Removing containers should only be done in compliance with your organization's legal and policy requirements for data retention.

Removing containers cannot be undone. The only way to recover containers is to restore your Splunk SOAR (On-premises) deployment from a backup.

Example: To delete all containers with the "test" label last updated before January 1, 2020 at 12:00:00 UTC:

```
phenv python /opt/phantom/bin/delete_containers.pyc --label test --before "2020-01-01T12:00:00Z"
```

Delete containers script arguments and record filters

Use these arguments for the `delete_containers.pyc` script to apply controls to the script.

Argument	Description
-h, --help	Show this help message and exit the script.
-b, --list-labels	List the available container labels and exit the script.
-d, --dry-run	Do not delete any containers, just show the results from the command. Use this option to test your command input before executing the script.
--non-interactive	Do not block script execution for user input. Use this flag for running <code>delete_containers.pyc</code> as part of an unsupervised script.
-c <number of containers to delete>, --chunk-size <number of containers to delete>	Maximum number of containers to delete in a single transaction. Maximum value is 10,000. Example: -c 100
-r <MAX_RETRY_COUNT>, --max-retry-count <MAX_RETRY_COUNT>	Maximum number of retries in case there is an error.

Use these filters for the `delete_containers.pyc` script to control on which containers the script deletes.

Filter	Description
-i <IDS>, --ids <IDS>	Delete the container IDs specified in a comma separated list.
-l <LABEL>, --label <LABEL>	Only delete containers with the specified label.
-m <string>, --matching <string>	Delete containers that title match the specified string. The match is not case sensitive.

Filter	Description
--before <date/time>	Only delete containers last updated before this date/time. Example: --before "2020-01-01T12:00:00Z"
--after <date/time>	Only delete containers last updated after this date/time. Example: --after "2020-01-01T12:00:00Z"
--status <STATUS>	Only delete containers the status values specified in a comma separated list.

Configure a source control repository for your Splunk SOAR (On-premises) playbooks

You can save your Splunk SOAR (On-premises) playbooks in Git repositories. By default, playbooks are managed in a Git repository called local. You can create additional Git repositories as needed. Doing so enables you to perform the following tasks:

- Import and export playbooks and share facilities among Splunk SOAR (On-premises) instances. For example, you can use Git to publish playbooks from a development Splunk SOAR (On-premises) environment to a separate production environment.
- Edit playbooks using a tool of your choice instead of the Splunk SOAR (On-premises) web interface.

Once you edit a playbook outside of the Visual Playbook Editor (VPE), you can no longer use drag and drop blocks in the VPE to edit that playbook. Any subsequent edits in the VPE are only possible by editing the full playbook. This is not recommended.

Splunk SOAR (On-premises) also uses a Git repository to publish company-authored playbooks for customers to download. This repository is called the community repository and is configured on Splunk SOAR (On-premises) by default. You can restore this repository if you accidentally remove it. See [Restore the community playbook repository](#).

You can transfer playbooks to Git using HTTP, HTTPS, or Git. Other protocols can be authenticated or anonymous if supported by the server.

Access the source control settings in Splunk SOAR (On-premises)

To access the Splunk SOAR (On-premises) source control settings, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Source Control**.

You can also access the source control settings from any Playbooks page by clicking **Manage source control**.

Set up a playbook repository using HTTP, HTTPS, or Git

To set up a Git repository using HTTP, HTTPS, or Git protocols, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **Administration Settings > Source Control**.
3. Select **Configure a new repository** from the Repositories drop-down list.
4. Provide a repository URL, repository name, and branch name. The repository name can be any name that describes your repository.

5. For HTTP and HTTPS, specify a username and password. Splunk SOAR (On-premises) attempts to connect anonymously if no username or password is provided. When crafting the URI, Splunk SOAR (On-premises) converts `https://server...` to `https://username:password@server....`. The Git protocol is not authenticated and does not require a username or password.
6. Click **Save Changes**.

A repository that is added to Splunk SOAR (On-premises) can't be edited. If you need to make a change, delete the repository and then add it again.

The username and password strings are separated so that the password can be encrypted and stored and not displayed to other administrators. However, passwords are stored as clear text in the Git configuration file for that repository.

Git hooks and the SOAR Playbook Editor

Splunk SOAR (On-premises) does not directly support Git hooks. If you choose to use git hooks in your system, be aware of the following:

- There is a risk that the playbook editor will not be able to save or push changes because the Git configuration rejects a commit.
- To avoid this issue, direct Splunk SOAR (On-premises) to push to a staging repository or branch that will not reject pushes. This prevents the playbook editor from being blocked from saving and pushing changes. Handle merge conflicts or other issues manually when pushing from the staging repository to the original repository.

If Git remote rejects your commits, causing the push to fail, you have two options:

- Delete and recreate
 1. Delete the repository and recreate it in Splunk SOAR (On-premises). The playbook reverts to the last successful push and removes all changes made after the last successful push.
 2. Recreate your changes and try to push again.
- Remediate the issue from the command line. Continue reading the next section.

To remediate the issue from the command line, follow the set of the instructions that matches your scenario.

You must have some expertise with Git or refer to Git documentation to perform some of the steps described in this section.

Problem commit is your most recent commit

If your most recent local commit is causing the problem, and you have not made subsequent commits, choose one of the following options:

Option 1:

1. Delete the most recent commit by running `git reset --hard HEAD~1`.
2. Recreate your changes and try to push again.

Option 2:

1. Make the necessary changes in your repository so it will pass the hook.

2. Replace your previous commit with this current state of the repository by running `git commit --amend`
3. Try to push again.

Problem commit is not your most recent commit

Choose the option that best matches your scenario:

Option 1:

If your playbook editor is blocked from pushing to the remote repository, follow these steps:

1. Delete the repository and recreate it in Splunk SOAR (On-premises). The playbook reverts to the last successful push and removes all changes made after the last successful push.
2. Recreate your changes and try to push again.

Option 2:

Otherwise, perform an interactive rebase in Git by following these steps:

1. Start the interactive rebase by running `git rebase -i <identifier>~1`
 - The identifier refers to the working version of the branch, such as `origin/master` or the commit hash for the last working commit.
 - Append `~1` to start the rebase on the commit before the last working version.
- Git displays a file that lists all of the commits in order. Choose an option:
 - *To delete the problematic commit entirely*, delete the line with the commit, while keeping the subsequent commits.
 - *To amend the problematic commit*, modify line for that commit, replacing `pick <hash>` with `edit <hash>`. After you write and close this file, git starts rebasing. Git pauses when it reaches that line. Then you can use `git commit --amend` to repair the commit.
- After the rebase completes successfully, push to the repository. If performing the rebase caused the commit history to diverge, you must perform a force push.

Set up a playbook repository using SSH

To set up a playbook repository using SSH, perform the following steps:

1. From the main menu, select **Administration**.
2. Select **Administration Settings > Source Control**.
3. Select **Configure a new repository** from the Repositories drop-down list.
4. Provide a repository URL starting with **ssh://** and including the username. For example:
`ssh://<username>@10.4.5.6/opt/repos`
5. Add the SSH public key from Splunk SOAR (On-premises) to your Git server's authorized keys file.
 1. Copy the contents in the **SSH Public Key** field.
 2. Log in to your Git server as a user with permissions to edit the Git server's `authorized_keys` file.
 3. Add the SSH public key to the authorized key file, such as `~/.ssh/authorized_keys`.
6. Provide a repository name and branch name. The repository name can be any name that describes your repository.

If you get the following error when setting up an external repo with SSH Auth:

```
Cmd('git') failed due to: exit code(128) cmdline: git fetch -v origin stderr: 'fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.'
```

This indicates that the `/home/<phantom_user>/.ssh/known_hosts` file is not being updated with the external repo and ssh

key info.

You can manually correct this through the CLI by running the following command as the Splunk SOAR (On-premises) user:

```
ssh-keyscan -t rsa <external_repo> >> /home/<phantom_user>/.ssh/known_hosts
```

Establish trust with the git repository from your Splunk SOAR (On-premises) deployment

You must inform git that it can trust the remote hosts of your Splunk Phantom deployment before you can use the source repository.

On your Splunk SOAR (On-premises) instance, or in the case of a cluster, on each Splunk SOAR (On-premises) cluster node:

1. SSH to the Splunk SOAR (On-premises) instance or cluster node. Log in as the user account that runs Splunk SOAR (On-premises).

```
ssh phantom@<phantom instance or cluster node>
```

2. Run the command to establish trust with the git repository.

```
git ls-remote git@<address of the git repository>
```

3. Verify that the information returned is correct. Example:

```
git ls-remote git@your-git-repository:phantom/phantom.git
```

4. If the returned values are correct, type **yes**.

Use repositories from the Playbooks page

You can make use of configured repositories on the Playbooks page. See View the list of configured playbooks for more information.

Restore the community playbook repository

The community playbook repository is a collection of playbooks vetted by the Splunk SOAR (On-premises) community. This repository is configured by default when Splunk SOAR (On-premises) is installed. Follow the procedure to restore the community repository if it is accidentally altered or deleted.

1. From the **Home** menu, select **Administration**.
2. Select **Source Control**.
3. In the Repositories drop-down list, select **Configure a new repository**.
4. In the Repo URL field, type the URL: `https://github.com/phantomcyber/playbooks.git`
5. In the Repo Name field, type **community**.
6. In the Branch Name field, enter the version of Splunk SOAR (On-premises) you are running, up to the second set of digits. For example, if you are running version 4.10.3 enter **4.10** in this field.
7. Check the **Read Only** check box.
8. Click **Save Changes**.

Configure settings for your Splunk SOAR (On-premises) instance

Enable clickable URLs in CEF data

When a Common Event Format (CEF) field on an artifact contains URL data, the user interface can display a clickable link for it.

- Use this setting to toggle whether clickable links are shown.
- Only CEF values generated by automation or included in an artifact are controlled by this setting.
- Since many URLs in CEF values are likely to be malicious, the default is **Off**.

Notes can contain URL data, and those URLs will be clickable unless they are escaped using the backtick or grave character (`). URLs in notes are not controlled by this setting.

Example:

```
`http://some.malicious.url.com`
```

Clustering

View cluster status and enable or disable a cluster

View the Clustering page to see the status of your Splunk SOAR (On-premises) clusters, enable or disable a cluster, or add additional nodes. See [Install and Upgrade Splunk SOAR \(On-premises\)](#) for information about setting up a cluster.

Perform the following steps to access the Clustering page:

1. From the main menu, select **Administration**.
2. Select **Product Settings > Clustering**.

The status of **online** means that the cluster node is up and running.

Disable a node by toggling the switch next to **Enabled** so that it is in the off position.

Click **View** to view the system health for that specific node. See [View the health of your Splunk SOAR \(On-premises\) system](#) to read more about the system health view for cluster nodes.

Configure multiple tenants on your Splunk SOAR (On-premises) instance

Enable multi-tenancy to allow one security team to manage multiple independent customers while segregating their customers' assets and data. For example, a Managed Security Service Provider (MSSP) business can use multi-tenancy to perform incident response for multiple clients with one analyst team on a single Splunk SOAR (On-premises) instance and maintain customer separation. The MSSP SOC can administer each customer's data set without needing a separate login and permissions configuration.

How many tenants can be configured?

The Splunk SOAR (On-premises) Community License only allows for one tenant if the multi-tenancy feature is enabled. You can view the number of allowed tenants in your Splunk SOAR (On-premises) instance by performing the following steps:

1. From the main menu, select **Administration**.
2. Select **Company Settings > License**.
3. View the information in the **Tenant Count** field.

The system default tenant doesn't count towards the total count.

Enable multi-tenancy

Splunk SOAR (On-premises) multi-tenancy isn't enabled by default. Perform the following steps to enable multi-tenancy:

1. From the main menu, click **Administration**.
2. Select **Product Settings > Multi-tenancy**.
3. Toggle **Enable Multi-tenancy** to **On**.
4. Click **Confirm** to confirm that you want to enable multi-tenancy.
5. Provide the information for the default system tenant.
6. Click **Save**.

View the tenants configured on your Splunk SOAR (On-premises) instance

To view the configured tenants in Splunk SOAR (On-premises), perform the following steps:

1. From the main menu, click **Administration**.
2. Select **Product Settings > Multi-tenancy**.

The default system tenant has an ID of 0. Each container in Splunk SOAR (On-premises) must have one tenant assigned. Before creating any additional tenants, all containers are assigned this default system tenant. Any containers that don't have an explicitly specified tenant and are created through an automated process are assigned to the default system tenant. If a container is created manually through the Splunk SOAR (On-premises) web interface you must select a tenant once you enable multi-tenancy.

Add a tenant to Splunk SOAR (On-premises)

To add a new tenant to Splunk SOAR (On-premises), perform the following steps:

1. From the main menu, click **Administration**.
2. Select **Product Settings > Multi-tenancy**.
3. Click **+ Tenant**.
4. Complete the information in the **Add Tenant** dialog box.
5. Click **Save**.

You can configure only as many tenants as your license allows, not including the default system tenant. If you already reached your limit, you must disable an existing tenant before you can add a new one.

Edit an existing tenant in Splunk SOAR (On-premises)

To edit the information for an existing tenant, hover and click the tenant you want to edit. Once a tenant is defined, you can't delete it. You must disable it instead. All tenant names must be unique.

Configure permissions for tenants and assets in Splunk SOAR (On-premises)

Each asset in Splunk SOAR (On-premises) must belong to one or more tenants. An asset can only be used by containers that share the same tenant as the asset. See [Add and configure apps and assets to provide actions in Splunk SOAR \(On-premises\)](#) for more information about configuring assets for tenants.

You can restrict access to tenant information based on role configuration in Splunk SOAR (On-premises). A role with no tenants specified means all users with the role have access to all tenants. To limit access to specific tenants, specify the tenants as part of the role configuration. See [Manage roles and permissions in Splunk SOAR \(On-premises\)](#) for information about configuring tenant user permissions.

Each container must have exactly one tenant. If no tenant is assigned to a container, then the container belongs to the default system tenant. An asset can have no tenants, which means it can be used with any tenant. See the following examples of assets and tenant usage:

- You can make assets based on public services, such as the whois databases, usable by all tenants.
- You can subscribe to a commercial service and make this service available for all tenants regardless of service level.
- Some assets such as a customer's firewall belong only to a specific tenant. Configure only one tenant for this type of asset.
- A premium commercial offering such as a commercial sandbox might be made available to a specific group of tenants. In order to ensure that only customers paying for that offering can use it, configure the asset so that it has only the paying customers.

Ingestion assets must have only one tenant, and this tenant is also assigned to any containers created by the ingestion asset. You can use separate assets for an app to separate data for different tenants. For example, consider if a Splunk Enterprise app is ingesting multiple customer logs tagged per customer. You can have a Splunk SOAR (On-premises) app that performs periodic polling of the Splunk Enterprise app based on a query containing the customer tag. One customer is called Initech, and a second customer is called Inirode. Create one asset for each company based on the Splunk Enterprise app:

- One query can contain `customer=initech`. Containers created by this asset belong to the Initech tenant.
- The second query can contain `customer=initrode`. Containers created by this asset belong to the Inirode tenant.

Containers can also be pushed to Splunk SOAR (On-premises) using the REST API. The REST API is accessed by automation users in Splunk SOAR (On-premises), each of whom is assigned a default tenant. The API caller can override this tenant, or use the default tenant if one is not specified. See REST Containers in the *REST API Reference for Splunk SOAR (On-premises)*.

In situations where you are not able to assign the correct tenant to a container, such as if you are unable to properly separate the data for different tenants, or do not have proper access to call the REST API to create containers, you can ingest the data using any default tenant, then use a playbook to assign the container to the desired tenant. For example, a container might have a field or artifact that maps directly to a customer name, or you might even need to look up custom IP address ranges to determine the customer before assigning the proper tenant.

View related data using aggregation rules

Define aggregation rules to view related data in a single location. Artifacts matching a defined rule are copied to a new container.

To view aggregation rules, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Aggregation**.

The Aggregation page shows a list of all container labels defined on your system. The number inside the parentheses next to each label is the number of rules defined for that label.

Container labels can be created by an ingestion asset or manually from **Home > Administration > Event Settings**. For example, you can choose a source label from an ingestion asset like the "Events" label or an "Email" label, then create a destination label such as "Aggregated Events" that makes it clear that containers with that label are aggregated.

Add a new aggregation rule

As an example, you may want to aggregate all containers with matching `sourceAddress` CEF fields from your "email" label into your "events" label.

To create the example aggregation rule:

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Aggregation**.
3. From the Aggregation page, click **+ Aggregation Rule**.
4. Specify **sourceAddress - Email to Events** as the name of the rule.
5. Select **email** from the drop-down list in the **Source Label** field.
6. Select **events** from the drop-down list in the **Destination Label** field.
7. Select **Exact** from the **Match** field to aggregate on the exact contents of the CEF field. You can click on the plus (+) icon to add additional match rules.
8. Select **sourceaddress** in the CEF field. You can start typing the field name to search through the list of available field names.
9. Click **Save**.

Edit an existing aggregation rule

After completing the previous example, perform the following steps to edit an existing aggregation rule in Splunk SOAR (On-premises).

1. Click on any existing rule. In this example, click **email** to view a summary of the aggregation rule.
2. Click **Edit** to make changes to the rule.
3. Click the trash can icon to remove the rule.

Click **+ Aggregation Rule** to create a new rule. If you create a new rule from the email label rule page, the new rule will automatically populate the Source Label field with email.

Using multiple matches in an aggregation rule

An aggregation rule can have multiple match lines, such as a match on both `sourceaddress` and `destinationaddress`.

For this example, both the `sourceaddress` and `destinationaddress` must match for it to be aggregated into the same container.

If you treat `sourceaddress` as the attacker's IP address, and `destinationaddress` as the target's IP address, then this means you have artifacts being aggregated in the same destination container for only the exact same attacker and victim. So with a target IP address of 1.1.1.1, there is one destination container for attacker IP address 2.2.2.2 and target IP address 1.1.1.1, and a different container for attacker IP address 3.3.3.3 and target IP address 1.1.1.1.

CEF fields are matched even if there is no value. For example, if you have artifacts with a `destinationaddress` of 1.1.1.1 and no `sourceaddress`, they are still aggregated together into a destination container.

Define tasks using workbooks

Workbooks are lists of standard tasks that analysts follow when they evaluate events or cases. You can create workbooks to analyze events. You can also combine multiple workbooks to create a more comprehensive workbook for cumulative events or cases, or cases that start out as one type of incident but end up to be a different type of incident.

Workbooks are available from Investigation, in both Summary View and Analyst View.

See Define a workflow in a case using workbooks in *Use Splunk SOAR (On-premises)* for information about how to use workbooks in a Splunk SOAR (On-premises) workflow.

Create a Splunk SOAR (On-premises) workbook

Perform the following tasks to create a new workbook in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Workbooks**.
3. Click **+ Workbook**.
4. Enter a name for your workbook.
5. (Optional) Enter a long description for your workbook.
6. Configure at least one phase for your workbook. A workbook can have multiple phases.
 1. Enter a name for the phase.
 2. (Optional) Configure a service level agreement (SLA) for the phase. See [Configure service level agreements in a workbook](#).
 3. Click the arrow next to **Task Name** to expand the section.
 4. Enter a name for the first task in the phase. You can have multiple tasks within each phase.
 5. (Optional) Assign an owner or role to the task. See [Notify task owners when they are assigned to a task](#).
 6. (Optional) Enter a long description or instructions for this task.
 7. (Optional) Configure an SLA for this task. The SLA must be shorter in length than the SLA for the phase.
 8. (Optional) Click **Actions** to select actions you want to run when this task is performed.
 9. (Optional) Click **Playbooks** to select playbooks you want to run when this task is performed.
 10. (Optional) Click **Add Task** to configure additional tasks for the phase.
7. (Optional) Click **Add Phase** to configure additional phases for the playbook.
8. Click **Save**.

Edit an existing Splunk SOAR (On-premises) workbook

Changes to a workbook only apply to future uses of the workbook. For example, if you change the SLA of a phase or add or remove a phase or task, the change is not reflected in any Splunk SOAR asset currently using the workbook.

To edit an existing workbook, do the following:

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Workbooks**.
3. Click on a workbook name to see the read-only summary of that page.
4. Use the drop-down list to expand the descriptions.
5. Click **Edit** to go to the workbook editing page.
6. Make the desired changes.
7. Click **Save**.

Reorder phases in a workbook

Suppose you need to add a phase to the middle of a series of phases in an existing workbook. New phases are added to the end by default, so you need to reorder the phases to place the new phase in its desired location.

Perform the following tasks to reorder a phase:

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Workbooks**.
3. Click on a workbook name to see the read-only summary of that page.
4. Use the drop-down list to expand the descriptions.
5. Click **Edit**.
6. Click **Reorder Phases**.
7. Enter the new phase at the bottom.
8. Click the three horizontal lines next to the phase and drag it to the order you want.
9. Click **Done Reordering**.
10. Click **Save**.

Configure service level agreements in a workbook

Service level agreements (SLAs) represent the default amount of time until a phase or task is due. You can adjust the time values to reflect your organization's requirements. The SLAs for phases and tasks are different from the SLAs that are set globally per severity across the entire platform.

Separate from severity SLAs, the phase and task SLAs allow for greater granularity when operating at the phase or task level. See [Create additional custom severity names](#) for more information about global SLAs and response settings.

The SLA time is tracked in minutes, days, or hours. It is based on the `start_time` timestamp when the phase or task is started and the `end_time` timestamp when the phase or task is completed. Each phase can have a total SLA that covers all the subtasks, or each task can have an individual SLA. However, if both the phase and task SLAs are used, there is no automatic validation to confirm that the phase SLA is greater than or equal to the total of all its subtask SLAs.

The owner of the phase or task sees SLA status messages in Investigation. You can also see the status of the current phase in the Summary View or in Analyst View, which is found under the Workbook tab. You can review if the SLAs are exceeded, how many tasks are completed, and how many of those tasks were completed on time.

To edit the phase or task SLA for the workbook, do the following:

1. From the **Home** menu, select **Administration**.
2. Select **Product Settings > Workbooks**.
3. Click on a workbook name to see the read-only summary of that page.
4. Use the drop-down list to expand the descriptions.
5. Click **Edit** to go to the workbook editing page.
6. Change the Phase SLA or from the Task Name drop-down list, in the Task SLA field, revise the time in which to complete the task.
7. Click **Save**.

Notify task owners when they are assigned to a task

You can notify owners that a workbook task is assigned to them. The table summarizes the methods.

Method of notification	Description
Email	When you assign a task to a role, Splunk SOAR (On-premises) sends an email notification to every member of the role. When a specific user assigns that task to themselves, the new owner and the previous owner both get an email notification.
In-product	When you assign a task to a role, every member of the role sees a bell notification in the Splunk SOAR (On-premises) menu bar. When a specific user assigns that task to themselves, the bell notification disappears for all other members of the role.

Tune performance by managing Splunk SOAR (On-premises) features

An administrator can tune performance of their Splunk SOAR (On-premises) deployment by toggling the **Indicators** feature or removing audit logs from the deployment after they have been downloaded.

Enable or disable the indicators feature

Prior to 4.8, retrieval of indicator records did not scale in some large deployments with hundreds of thousands of indicator records. Improvements have been made to enhance performance, but some administrators may wish to disable the feature entirely.

An administrator can toggle the **Indicators** feature of Splunk SOAR (On-premises) by running a script from the *nix shell command line.

Disabling the **Indicators** feature removes it from the **Main Menu**, from the **events** page, and from context menus in the investigations page.

When indicators are disabled, the indicator REST APIs return response 400, with the message body:

```
{
  "failed": true,
  "message": "The indicators feature is not enabled."
}
```

Affected APIs

- /rest/indicator
- /rest/indicator_by_value

- /rest/indicator_artifact
- /rest/indicator_artifact_timeline
- /rest/indicator_stats_indicator_count
- /rest/indicator_stats_top_labels
- /rest/indicator_stats_top_types
- /rest/indicator_stats_top_values
- /rest/ioc
- /rest/indicator_common_container

See REST Indicators.

Toggle the Indicators feature

To disable Indicators:

1. SSH to your Splunk SOAR (On-premises) instance.
SSH <username>@<phantom_hostname>
2. Run the set_preference command.
phenv set_preference --indicators no

To enable Indicators:

1. SSH to your Splunk SOAR (On-premises) instance.
SSH <username>@<phantom_hostname>
2. Run the set_preference command.
phenv set_preference --indicators yes

It can take as much as five minutes for the indicators feature to be hidden or to show from the Splunk SOAR (On-premises) UI after the set_preference command has been run.

Delete audit logs

Downloading Audit logs could take a long time because all the records were loaded into memory before being written to a file. In version 4.8, audit logs have been changed to stream records to a file.

An administrator can remove audit logs after they have been manually downloaded and archived by using the delete_audit_logs.pyc script found in /<PHANTOM_HOME>/phantom/bin.

This script will permanently delete audit records from Splunk SOAR (On-premises). The records cannot be recovered without restoring Splunk SOAR (On-premises) from a backup. Exercise caution when using this script.

delete_audit_logs.pyc arguments

```
# phenv python delete_audit_logs.pyc-h
usage: delete_audit_logs.py [-h] [--before BEFORE_TIMESTAMP]
                             [--after AFTER_TIMESTAMP]
                             [--categories [CATEGORIES [CATEGORIES ...]]]
                             [--dry-run] [--non-interactive]
                             [--log-level {NOTSET,DEBUG,INFO,WARNING,ERROR,CRITICAL}]
```

Argument	Description
-h, --help	Show this help message and exit.
--before <BEFORE_TIMESTAMP>	Records created before this timestamp will be deleted. Records created after this timestamp will not be deleted. The timestamp value must be in yyyy-mm-dd [hh:mm:ss] format.
--after <AFTER_TIMESTAMP>	Records created after this timestamp will be deleted. Records created before this timestamp will not be deleted. The timestamp value must be in yyyy-mm-dd [hh:mm:ss] format.
--categories [CATEGORIES [CATEGORIES ...]]	Only delete records with the given categories. Examples of categories: user, container, playbook, administration, artifact.
--dry-run	Do not run the DELETE queries. Use this argument to test your parameters before running the script for real.
--non-interactive	Do not block on user input. This flag is suitable for running as part of an unsupervised script.
--log-level {NOTSET, DEBUG, INFO, WARNING, ERROR, CRITICAL}	Set the log level. Default level is WARNING.

Examples

Test script parameters by using the --dry-run option first.

Delete all audit logs from before July 2019:

```
sudo phenv python delete_audit_logs.py --before 2019-07-01
```

Delete audit logs between July 1 and December 1 2019:

```
sudo phenv python delete_audit_logs.py --after 2019-07-01 --before 2019-12-01
```

Use data retention strategies to schedule and manage your database cleanup

Manage the records in your Splunk SOAR (On-premises) PostgreSQL database with the `configure_db_maintenance` subcommand of `manage.py`.

Use `configure_db_maintenance` to set options for the `db_maintenance` tool. A set of options is called a strategy. Strategies are applied to models.

Strategy

The set of configurable parameters that define when a record should be deleted, either automatically or when the `db_maintenance` tool runs.

Model

Any PostgreSQL database record or Django object is called a model. Models have characteristics that define what sort of information the model represents.

Model name	Description
container	Containers. See About Splunk SOAR (On-premises).

Model name	Description
indicator	Indicators or Indicators of Compromise. See About Splunk SOAR (On-premises).
container_audit_trail, audit	Audit logs. See Enable and download audit trail logs in Splunk SOAR (On-premises) .
device_profile	Mobile device profiles. See Enable or disable registered mobile devices .
notification	Notifications.
playbook_run	Records of playbook runs.

To use the `configure_db_maintenance.py` tool, follow these steps:

1. SSH to your Splunk SOAR (On-premises) instance.
SSH <username>@<phantom_hostname>
2. Use the following tool to manage data deletion based on your installation.
 1. For an unprivileged installation, use this command:
phenv python /opt/phantom/www/manage.py configure_db_maintenance
 2. For a privileged installation, use this command:
sudo phenv python /opt/phantom/www/manage.py configure_db_maintenance
3. Append your desired argument to the data retention tool command line to schedule, list, enable, or disable data retention actions.

On clustered systems, the `configure_db_maintenance.py` tool can be run from any node, but only the leader node runs the data retention strategy.

Data retention tool arguments

Append the `--help` argument to your tool to get information on the data retention tool arguments;

```
phenv python /opt/phantom/www/manage.py configure_db_maintenance --help
```

Optional arguments

Use these optional arguments to manage your data retention strategy.

Argument	Description
-h, --help	Show this help message and exit.
--schedule	Schedule data retention to execution schedule.
--cron-schedule <CRON_SCHEDULE>	How often to query Data Retention Schedule. Must be a cron schedule expression.
--list	List strategies in data retention strategy.
--target-model <TARGET_MODEL>, -m <TARGET_MODEL>	Name of model to run action on.
-v {0,1,2,3}, --verbosity {0,1,2,3}	Verbosity level; 0=minimal output, 1=normal output, 2=verbose output, 3=very verbose output.

You must specify the target model to add, delete, enable, or disable a model.

Add a model to your data retention strategy

The following arguments are required to successfully add a model to the data retention strategy.

Argument	Description
--add	Add a model strategy to the data retention strategy. You must supply the following sub-arguments: <ul style="list-style-type: none">• <code>-m</code> the name of the model to add; container, indicator, audit, device_profile, notifications, or playbook_runs.• <code>-u</code> unit of time; hours,days,months, or years.• <code>-a</code> number of time units to use
--age-to-keep-time-unit {hours,days,months,years}, -u {hours,days,months,years}	Set the unit of time to use, hours, days, months, or years.
--max-age-to-keep <MAX_AGE_TO_KEEP>, -a <MAX_AGE_TO_KEEP>	How many units of time to keep model.
--disabled	Set the strategy to disabled when it is created.

If you add a data retention strategy for a model that already has one, the new strategy replaces the existing strategy.

Edit a model's entry in your data retention strategy

The following arguments are required to edit a model in the data retention strategy.

Argument	Description
--delete	Delete a model strategy from the data retention strategy. You must supply the <code>-m</code> argument with the name of the model to delete.
--enable	Enable a model strategy in the data retention strategy. You must supply the <code>-m</code> argument with the name of the model to enable.
--disable	Disable a model strategy in the data retention strategy. You must supply the <code>-m</code> argument with the name of the model to disable.

Examples

Delete indicator records after three months:

```
phenv python /opt/phantom/www/manage.py configure_db_maintenance --add -m indicator -u months -a 3
```

Change the schedule on which `configure_db_maintenance` runs:

```
phenv python /opt/phantom/www/manage.py configure_db_maintenance --schedule --cron-schedule "0 * * * *"
```

Configure settings for your Splunk SOAR (On-premises) system's events

Create custom status labels in Splunk SOAR (On-premises)

You can create additional status labels for the events and cases in Splunk SOAR (On-premises) as needed for your business processes.

Statuses are grouped into three categories: New, Open, and Resolved. You can create up to 10 total status labels in Splunk SOAR (On-premises).

Status label rules

Status labels must adhere to the following rules:


- At least one status label must exist for each of the status categories.
- The labels New, Open, and Closed are available upon upgrade. These three labels can be deleted, removing them from the active list. These labels cannot be renamed because they are required for backwards compatibility with apps and playbooks.


To maintain backwards compatibility with apps and existing playbooks, if the status labels New, Open, or Closed have been deleted, ingestion apps and the REST API can still assign the statuses New, Open, and Closed to containers.

Create a status label in Splunk SOAR (On-premises)

To create a status label, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Status**.
3. Click **Add Item** in the status category where you want to create the new status label.
4. Type the new status name. The status label name must adhere to the following conditions:
 - ◆ Only ASCII characters a-z, 0-9, dash (-), or underscores (_) are allowed.
 - ◆ The name cannot exceed 20 characters in length.
5. Click **Add Item**.

To reorder status labels, drag the handle () on the left side of the status label's input box to the desired position.

To delete a status label, click the circled x () to the right of the status label's input box.

To set the status label used as the default label for that status type, select the desired label from the drop-down list in the **Default status** field.

Create custom severity names

Severity defines the impact or importance of an event or case. Different severity names have different assigned service level agreements in the Response page. Splunk SOAR (On-premises) ships with three predefined severity names: High,


Medium, and Low. Your organization might need additional levels of severity to match your business processes. Additional severity names can be defined by a Splunk SOAR (On-premises) administrator.

You can create up to 10 severities in Splunk SOAR (On-premises).

Create a severity in Splunk SOAR (On-premises)


To create a severity, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Severity**.
3. Click **Add Item**.
4. Enter the severity name and select a color from the drop-down list. The severity name must adhere to the following conditions:
 - ◆ Only ASCII characters a-z, 0-9, dash (-), or underscores (_) are allowed.
 - ◆ The name cannot exceed 20 characters in length.
5. Click **Done**.

Severity names cannot be edited. To change a severity name, delete it and recreate the severity name. To reorder severity names, drag the handle () on the left side of the severity name's input box to the desired position.

To set the severity name used as the default severity, select the desired name from the drop-down list.

Delete a severity name in Splunk SOAR (On-premises)

To delete a severity name, click the circled x () to the right of the severity name's input box. Take note of the following Splunk SOAR (On-premises) behaviors before you delete a severity:

- The severity label set as the default severity cannot be removed until a new default is selected.
- Deleting a severity name does not change the severity of a case, event, or artifact. Changing a severity name does not update closed events, cases, or artifacts.
- Deleted severity names appear in search results as strikethrough text.
- Severity names are stored in Splunk SOAR (On-premises)'s internal database. Deleting a severity name from the active severity list does not remove that severity name from the database.
- To maintain backwards compatibility with apps and existing playbooks, if the severity names High, Medium, or Low have been deleted, ingestion apps and the REST API can still assign the severity High, Medium, and Low to events, containers, or artifacts.

Create custom fields to filter Splunk SOAR (On-premises) events

Create custom fields that can be added to containers in Splunk SOAR (On-premises). You can use custom fields to match your business processes, or to help filter containers, events, or cases for extra attention. For example, you might add a custom field named **Department** and assign it a list of values for each department in your organization (for example, IT Ops, Sales, and Business).

Custom fields are searchable. For more information on using the search feature, see Search within Splunk SOAR (On-premises) in *Use Splunk SOAR (On-premises)*.

Using custom fields in playbooks requires [special coding](#). Custom field names described here require additional special handling, so plan your naming convention carefully.

- names containing characters other than letters, numbers, or underscores (_)
- names starting with a space

Create a custom field

To create a custom field, follow these steps:

1. From the **Home** menu select, **Administration**.
2. Select **Event Settings > Custom Fields**.
3. Click **Add Field**.
4. Enter a field name.
5. Select a field type. If you choose **select**, provide additional values in the **Values** field. These values are presented to the user in a drop-down list when working in a container.
6. (Optional) Select **Require on Resolve** to make the field required before a container can be closed or resolved.
7. (Optional) Click **Add Field** to add additional fields.
8. Click **Save Changes**.

Edit custom fields

To edit a custom field, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Custom Fields**.
3. Find the item you want to edit and make your changes. In the **Values** field for select types, you can enter an additional value or click the X icon to remove existing values.
4. Check or uncheck **Require on Resolve** as needed.
5. Click **Save Changes**.

Delete a custom field

You can remove a custom field entirely. To remove a custom field, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Custom Fields**.
3. Locate the field you want to remove.
4. Click the circled x (ⓧ) icon at the end of the field's entry.
5. Click **Save Changes**.

Update and read custom field values

To update custom field values in containers, use the following code examples with the container.update API :

Update a custom field value

Example code to update a custom field value from a container.

```
outputs = {}

# Write your custom code here...
container = {"id": container_id}
update = {
```

```

        "custom_fields": {
            field_name: field_value,
        },
    }

# Make the HTTP request
success, message = phantom.update(container, update)

assert success, message

# Return a JSON-serializable object
assert json.dumps(outputs) # Will raise an exception if the :outputs: object is not JSON-serializable
return outputs

```

Read a custom field value

While not technically an update function, reading a custom field value also uses the `container.update` API.

Example code to read (get) a custom field value from a container.

```

outputs = {}

# Write your custom code here...
container = phantom.get_container(container_id)
custom_fields = container.get("custom_fields", {})
outputs["field_found"] = field_name in custom_fields
outputs["field_value"] = custom_fields.get(field_name)

# Return a JSON-serializable object
assert json.dumps(outputs) # Will raise an exception if the :outputs: object is not JSON-serializable
return outputs

```

Filter indicator records in Splunk SOAR (On-premises)

When you first install Splunk SOAR (On-premises), industry-standard indicator records are generated for events coming in. This can result in the generation of a large volume of indicator records many of which might not be necessary for your system. You can filter out certain indicators to decrease the number of indicator records that are generated.

Create a filter

To filter out certain indicators, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Indicators**.
3. To filter out certain indicator records, uncheck the box by the field name of the record you don't want to generate indicators for. If you have created any custom CEF fields, by default those fields don't have indicator records. If you want to create indicators for these fields, make sure to check the box next to the field name.
4. After you have made any changes, click **Save Changes**.
5. (Optional) To sort by data type, click **Data Type** and choose how you would like to sort the fields. You can also search for indicators by data type in the search bar to add them to the filter.
6. (Optional) Click **Field Type** to sort the fields based on default or custom fields.
7. (Optional) Use the search bar to search for specific fields.
8. (Optional) Use the Total Count column to see the number of each type of indicator record across the system.

This filter applies only to events coming in after the filter is set and does not apply to indicator records that were previously created.

Track information about an event or case using HUD cards


Use the head-up display (HUD) in Investigation to quickly track relevant information about an event or case. HUD cards can display a metric from the built-in list or display a custom field. For more information about custom fields, see [Create custom fields to filter Splunk SOAR \(On-premises\) assets](#).


Create a HUD Card

Perform the following tasks to create a HUD card:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > HUD**.
3. Click **+ HUD Card**.
4. Select a HUD card type.
 - ◆ Select **Preset Metrics** to view predefined metrics about your asset, such as remaining tasks, number of failed actions, or tasks exceeding the SLA. Select the desired metric from the drop-down list. and then choose a background color for the HUD card.
 - ◆ Select **Custom Field** to view the information you defined in a custom field. See [Create custom fields to filter Splunk SOAR \(On-premises\) events](#). The fields defined there are available in the drop-down list. Choose a background color for the HUD card.
5. Click **Done**.

Manage HUD Cards

HUD cards display in Investigation in the same order they appear in the list of HUD cards in the settings page. Reorder the cards by dragging the cards by the handle () into the order you want them to be displayed.

Delete a HUD card by clicking the circled x () icon to the right of the HUD card definition.

See HUD cards for more information on using HUD Cards in *Start with Investigation in Splunk SOAR (On-premises)*.

Configure the response times for service level agreements

Service level agreements (SLA) define the number of minutes that is permitted to pass before an action or approval is considered late. SLAs are used for the following purposes in Splunk SOAR (On-premises):

- To track the amount of time a container or case has remaining before it is considered due.
- To track the amount of time an approver has to approve an action before the approval escalates. For more information about the approval and escalation process, see Approve actions before they run in Splunk SOAR (On-premises) in *Use Splunk SOAR (On-premises)*.

Each event or case must have a severity assigned, and each severity has a corresponding SLA. This table lists the default SLA settings in Splunk SOAR (On-premises):

Severity name	SLA in minutes
High	60
Medium	720
Low	1440

The SLA time starts when a case or container is created. An action or approval is considered late if the SLA time is reached before the case or container is closed.

Set service level agreement times

You can set the SLA for any default or custom severity name in Splunk SOAR (On-premises). Custom severities follow the same escalation process that the default severities follow. To set an SLA time for a severity, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Response**.
3. In each severity level, type a number of minutes permitted to elapse before an action or approval must be escalated.
4. (Optional) Check **Automatic self-approval** if you want actions activated by a user who can approve them to be approved automatically.
5. (Optional) Add executive approvers by selecting them from the drop-down list in the **Executive approvers** field. When all of the SLA escalations have expired without being acted on, the executive approvers receive an SLA breach notification.
6. Click **Save Changes**.

Configure how events are resolved

Set any tags needed before an event can be marked as resolved. Setting a custom field as a required tag updates the settings for the custom field.

To configure how an event is resolved, follow these steps:

1. From the **Home** menu, select **Administration**.
2. Select **Event Settings > Resolution**.
3. Check the **Require the Following Tags on Resolve** checkbox.
4. Type the names of any tags needed before an event or container can be marked as resolved. Tags can be removed by clicking the x next to the tag name.
5. Set the action Splunk SOAR (On-premises) takes when artifacts are added to a resolved event. Select an action from the drop-down list that matches your business process.
 - ◆ Select **Keep Event Resolved** to keep events resolved when new artifacts are added.
 - ◆ Select **Reopen Event** to reopen any event that has a new artifact added.
 - ◆ Select **Duplicate Event** to create a duplicate event, and then add the new artifact to the new event.
6. Click **Save Changes**.

Configure labels to apply to containers

Labels are a property applied to containers. A label applied to a container enables Splunk SOAR (On-premises) to run playbooks and other automation against containers.

Splunk SOAR (On-premises) ships with one label defined: events. More labels can be added to suit your workflow or organizational needs. Labels can have additional custom fields, be used as the basis of a HUD Card, or have tags required before the label's container can be set to a closed or resolved status.

Create a label

Perform the following steps to create a label:

1. From the **Home** menu, select **Administration**.
2. Click **Event Settings > Label Settings**.
3. Click **+ Label**.
4. Type a name for the label.
5. Click **Create**.

Delete or modify a label

Delete a label by clicking the  icon to the right of the label's name.

Perform the following tasks to modify a label:

1. From the **Home** menu, select **Administration**.
2. Click **Event Settings > Label Settings**.
3. Click the label's name in the list.
4. Click either Custom Fields, HUD, or Resolution. Each of these items behaves identically to the top-level settings of the same name.
 - ◆ For Custom Fields settings, see [Create custom fields for containers](#).
 - ◆ For HUD settings, see [Track information about an event or case using HUD cards](#).
 - ◆ For Resolution settings, see [Configure how events are resolved](#).

Use authorized users to grant authorized access

Authorized Users are enabled by default. Use this setting to toggle whether the Authorized section is visible in the Investigation screen's HUD.

The Authorized control for managing the Authorized Users appears in the Investigation screen if the authorized users are turned on. The control appears in the HUD, accessed by using the double-down chevron pull-down tab.

Access the HUD and Event Info by doing the following:

1. Click the double-down chevron.
2. Click the right arrow (>) next to **Event Info**.

The Authorized control is located in the **People** section.

This toggle is available for viewing and editing if your role has view and edit permissions for the system settings. See [Manage roles and permissions in Splunk SOAR \(On-premises\)](#) for more information about roles and permissions.

Disable authorized users by doing the following:

1. From the **Home** menu, select **Administration**.

2. Select **Event Settings > Authorized Users**.
3. Click the **Enable Authorized Users** toggle to the Off position.

Once disabled, the Authorized section is no longer visible in Investigation. Reenabling the Authorized Users makes the Authorized section visible in Investigation and also reenables the authorized access that was previously configured.

Authorized access might not be available for every user in the system by default. Authorized access can only be granted to the subset of users who are already assigned to a label that has edit permissions on the container. For example, some teams only want to allow certain people to work on particular types of cases. Not every user assigned to a label needs access to a particular case.

Grant authorized access by doing the following in Investigation:

1. Expand the **Event Info** collapsible section of a container.
2. Click the edit icon in the **Authorized** section.
3. From the **Authorized Users** drop-down list, select the names of the people who need access.

The Authorized section is visible if you have basic permissions for events with view selected. The Authorized Users drop-down list is editable if you have label permissions for events with view and edit selected.

Administrators always have access to all containers. Normally, you don't need to authorize them. However, if you want to restrict a container to administrators only, set Administrators in the Authorized Users list. Setting specific user names will enable the specific users and administrators.

Manage your Splunk SOAR (On-premises) users and accounts

Manage Splunk SOAR (On-premises) users

View the **Users** page to see the users configured on your Splunk SOAR (On-premises) instance, add new users, or edit existing users.

Perform the following steps to access the Users page:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.

Default users and types of users

On a new Splunk SOAR (On-premises) instance, the following default users are available:

- **Admin:** This is the default admin account and cannot be disabled or deleted. The admin user is not counted towards the seat count of a seat-based license.
- **Automation:** The automation user is not counted towards the seat count of a seat-based license.

An information card is shown for each user. For a local user the information card displays:

- The user's full name
- username
- last access date and time
- roles
- an icon showing the user's initials or custom icon

For automation users, the information card displays a colored ribbon on the left side of the card indicating the user type.

The automation user is a default internal service account used by Splunk SOAR (On-premises) for running automated playbooks and asset actions, such as data ingestion. The automation user and any other users with the automation type do not have passwords and can't log into the Splunk SOAR (On-premises) web interface. However they do provide REST authentication tokens that can be used to read and write data to the REST API. For information on how to use the REST API and authentication tokens, see Using the Splunk SOAR (On-premises) REST API reference in the *Splunk SOAR (On-premises) REST API Reference*.

Customize what you see on the Users page

Customize the information you see on the **Users** page:

- Click the drop-down list in the **Show** field to view more or fewer user cards at a time. By default, 24 user cards are shown.
- Use the filter in the **View by** field to sort the users by first name, last name, username, last accessed, and last created.
- Click on the ellipsis (...) icon in the upper-right corner of each user card for additional options, such as viewing the user's effective permissions, editing the user, or deleting the user.

Configure user permissions

All user permissions in Splunk SOAR (On-premises) are derived from the user's role. To grant permissions to a user, you assign a role with the desired permission. Only the default admin user can have special, hard-coded permissions outside of any roles.

Perform the following steps to view the permissions for a user:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.
3. Click on a user card and review the roles assigned to this user in the **Roles** field.

Users with multiple roles have the sum of all the permissions allowed by those roles.

See [Manage roles and permissions in Splunk SOAR \(On-premises\)](#) for more information about Splunk SOAR (On-premises) roles and the permissions provided by each role.

Add users to Splunk SOAR (On-premises)

You can add users to Splunk SOAR (On-premises) from the Splunk SOAR (On-premises) web interface. The user can be authenticated locally by Splunk SOAR (On-premises), or by using SAML2. In the case of SAML2, the user account can be created in Splunk SOAR (On-premises) or created automatically during the user's initial login. In order for accounts to be automatically created, a group mapping to a Splunk SOAR (On-premises) role must be configured. See [Configuring single sign-on authentication for Splunk SOAR \(On-premises\)](#).

Create a local Splunk SOAR (On-premises) user

Perform the following tasks to add a local Splunk SOAR (On-premises) user. The user is authenticated by the Splunk SOAR (On-premises) instance.

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.
3. Click **+ User**.
4. Verify that the **User type** is set to **Local**.
5. Enter a username in the **Username** field.
6. Enter a password in the **Password** field.
7. (Optional) Complete the other fields on the screen, such as first and last name, email address, title, time zone, and location. If two factor authentication is enabled, also provide the Duo username. See [Secure Splunk SOAR \(On-premises\) using two factor authentication](#).
8. Click **Create**.

Create a SAML2 Splunk SOAR (On-premises) user

Perform the following steps to add a user who is authenticated using single sign-on (SSO). Before you do this, make sure you have single sign-on enabled. See [Configuring single sign-on authentication for Splunk SOAR \(On-premises\)](#).

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.
3. Click **+ User**.
4. In the **User type** field, select the SSO provider. Only the configured and enabled SSO providers are available to choose from.

5. Enter the username in the **Username** field.
6. (Optional) Complete the other fields on the screen, such as time zone and roles. If two factor authentication is enabled, also provide the Duo username. See [Secure Splunk SOAR \(On-premises\) using two factor authentication](#).
7. Click **Create**.

Create an automation user in Splunk SOAR (On-premises)

Perform the following steps to add an automation user in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.
3. Click **+ User**.
4. In the **User type** field, select **Automation**.
5. Enter the username in the **Username** field.
6. (Optional) In the **Allowed IPs** field, specify the IP addresses allowed to connect as this user. You can specify individual IP addresses, CIDR ranges, or **any** to allow all IP addresses.
7. (Optional) Enter a default label for this user. Any containers that get created by this user use this label if another label is not specified.
8. (Optional) If multi-tenancy is enabled, select the default tenant in the **Default Tenant** field.
9. (Optional) The **Automation** role is provided to automation users by default. See [Manage roles and permissions in Splunk SOAR \(On-premises\)](#) for more information about the permissions granted by each role.
10. Click **Create**.

Edit an automation user to view the REST API authorization token and associated assets

Click an existing automation user on the **Users** page to view the following information:

- The REST API authorization token, which is used to authenticate the user for access to the REST API. See [Using the Splunk SOAR \(On-premises\) REST API reference in the Splunk SOAR \(On-premises\) REST API Reference manual](#).
- The assets associated with this user.
 - ◆ The automation user is used to test connectivity with the listed assets, and also for ingesting data. Use the automation user configuration to set the permissions of the asset when the asset is running on its own.
 - ◆ When the asset is not performing test connectivity or data ingestion, it is running with the permissions of the user performing the action. If the asset is being run from a playbook, the asset has the permissions of the playbook user.
 - ◆ You can assign assets to an automation user during asset configuration. If you assigned an automation user to an asset, the asset appears in the automation user's card. See [Configure automation users for a Splunk SOAR \(On-premises\) asset](#).

Disable an existing Splunk SOAR (On-premises) user

Disable a user in Splunk SOAR (On-premises) to prevent that user from logging in or accessing the system. Disabling a user does not delete the user account.

To disable an existing Splunk SOAR (On-premises) user, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Users**.

3. Click the ellipsis (...) icon for the user you want to disable, and select **Edit**.
4. Click the **Disabled** checkbox.
5. Click **Save**.

Manage roles and permissions in Splunk SOAR (On-premises)

Roles in Splunk SOAR (On-premises) serve the following purposes:

- Grant users permission to access system functionality, or restrict access to parts of the system.
- Act as a mechanism for grouping users for approvals. See Approve actions before they run in Splunk SOAR (On-premises) in the *Use Splunk SOAR (On-premises)* manual.

View your Splunk SOAR (On-premises) roles

To view the roles configured in your Splunk SOAR (On-premises) instance, perform the following steps to access the Roles page:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Roles & Permissions**.

Splunk SOAR (On-premises) includes the following default roles that can't be edited or deleted:

Role	Description
Administrator	<p>Users with this role have view, edit, and delete privileges to and can access all Splunk SOAR (On-premises) functions and settings:</p> <ul style="list-style-type: none"> • View, edit, and delete permissions for everything • Manage users and accounts • Change any and all Splunk SOAR (On-premises) settings • Install or remove apps or connectors • Create, edit, and delete Assets • Create, edit, and delete workbooks • Create, edit, run, and delete playbooks
Asset Owner	<p>Users with this role can:</p> <ul style="list-style-type: none"> • Create, edit, and delete assets • View apps or connectors, events, custom lists, playbooks, system settings, and users and roles.
Automation	<p>This is a service account role used for automated tasks including REST API operations, playbook execution, and ingestion.</p>
Automation Engineer	<p>Users with this role can:</p> <ul style="list-style-type: none"> • View, run, edit playbooks, and can edit playbook code • View apps, assets, custom lists, events, system settings, and users and roles
Incident Commander	<p>Users with this role can:</p> <ul style="list-style-type: none"> • Create, edit, and delete cases • Create, edit, delete, run, or edit the code for playbooks • View and edit events • Create, edit, and delete workbooks • View apps, assets, system settings, and users and roles

Role	Description
Observer	Users with this role can view everything except workbooks, but cannot edit or run anything.
OnPrem Broker	This is a service account that allows the Automation Broker to view apps.

Users granted multiple roles have the cumulative privileges of all the roles. You can also restrict access to specific named objects. See [Named object permissions](#).

Add a role to Splunk SOAR (On-premises)

Perform the following steps to add a new role in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Roles & Permissions**.
3. Click **+ Role**.
4. Enter a name for the role.
5. (Optional) Enter a description for the role.
6. Select the **Basic Permissions** provided by this role.

Component	Permission and Description
Apps	<ul style="list-style-type: none"> ◆ Select Edit to allow the user to add or delete apps, or edit settings on individual apps. ◆ Select View to allow the user to view the list of installed apps, and view the settings for individual apps.
Assets	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete assets. Note that the user will also need view assets in order to see the asset before they can edit it. ◆ Select Edit to allow the user add and edit assets. ◆ Select View to allow the user the ability to look at the list of assets and individual asset configurations.
Cases	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete cases. ◆ Select Edit to allow the user to create and edit cases. ◆ Select View to allow the user to view cases.
Events	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete events. ◆ Select Edit to allow the user to modify events. This includes data about the event itself (assigned owner, SLA) as well as being able to add items to artifacts and files. ◆ Select View to allow the user to view events. This includes both the list of events, as well as the contents of individual events.
Custom Lists	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete custom lists. ◆ Select Edit to allow the user to create and edit custom lists. ◆ Select View to allow the user to view custom lists.
Playbooks	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete playbooks. ◆ Select Edit to allow the user to edit playbooks, including modifying the playbook settings such as logging, active, safe mode, and draft mode. For more information on playbook settings, see Manage settings for a playbook in Splunk SOAR (On-premises) in the <i>Build Playbooks with the Visual Editor</i> manual. ◆ Select View to allow the user to view playbooks. ◆ Select Execute to allow the user to execute playbooks on events. ◆ Select Edit Code to allow playbook authors to manually edit Python code and customize code blocks. Authors without this permission can only use the visual block editor.
Workbooks	<ul style="list-style-type: none"> ◆ Select Delete to allow the user to delete workbooks. Note that the user will also need view workbooks in order to see a workbook before they can edit it. ◆ Select Edit to allow the user add and edit workbooks. ◆ Select View to allow the user the ability to look at the list of workbooks.

Component	Permission and Description
System Settings	<p>◆ Select Edit to allow the user to change System Settings.</p> <p>The System Settings include authentication servers. Users with edit system settings have the ability to perform a privilege escalation attack.</p>

7. Select **View** to allow the user to view system settings.

Users and Roles

- ◆ Select **Edit** to allow the user to edit, delete and add users and roles. Security note: a user with **Edit** permission can grant themselves all other privileges. They should be considered equivalent to an administrator.
- ◆ Select **View** to allow the user to view users and roles, including what role each user has, email addresses, and last login time.

8. Click **Label Permissions** to configure label permissions for this role. The labels you see in the table depend on the labels you have defined on your Splunk SOAR (On-premises) instance. See [Create additional custom status labels in Splunk SOAR \(On-premises\)](#). The following permissions can be configured:

Permission	Description
Delete	The user can delete any object in Splunk SOAR (On-premises) that has this label. Clicking this automatically grants the Edit and View permissions.
Edit	The user can edit any object in Splunk SOAR (On-premises) that has this label. Clicking this automatically grants the View permission.
View	The user can view any object in Splunk SOAR (On-premises) with this label, but cannot modify or delete any such objects.

9. Click **Repository Permissions** to configure repository permissions for this role. The repositories you see in the table depend on the repositories configured on your Splunk SOAR (On-premises) instance. See [Configure a source control repository for your Splunk SOAR \(On-premises\) playbooks](#). The following permissions can be configured:

Permission	Description
Delete	The user can delete any playbook in this repository. Clicking this automatically grants the Edit and View permissions.
Edit	The user can edit any playbook in this repository. Clicking this automatically grants the View permission.
View	The user can view any playbook in this repository, but cannot modify or delete any playbooks.
Execute	The user can run any playbook in this repository.

10. Click **Create Role**.

Add users to a role in Splunk SOAR (On-premises)

Perform the following steps to add users to a role in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Roles & Permissions**.
3. Click the role you want to edit and add users to.
4. Click **Add Users**.
5. Select a user from the drop-down list, or start typing a username to filter the users that are displayed.
6. Click **Add**.
7. Repeat and continue adding users as desired. Each time a user is added, the user card appears in the **Users** field in the role.

Edit a role in Splunk SOAR (On-premises)

Perform the following steps to edit a Splunk SOAR (On-premises) role:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Roles & Permissions**.
3. Select a custom role you want to modify. You can modify any of the permissions in a custom role, add users or remove users. When editing a system role, you can only add or remove users.
 - ◆ Users added to a role have their permissions saved in real time, before you click **Save Changes**.
 - ◆ Permission changes to roles are applied in real time to the users who are granted the updated permissions, before you click **Save Changes**.
 - ◆ Users inheriting roles from an SSO provider must log out and log back in to Splunk SOAR (On-premises) to see their updated permissions.
4. Click **Save Changes**.

Delete a role in Splunk SOAR (On-premises)

Perform the following tasks to delete a role in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Roles & Permissions**.
3. Click the role you want to delete.
4. Click **Delete Role**.
5. Click **Delete** to confirm that you want to delete the role.

Configure password requirements and timeout intervals to secure your Splunk SOAR (On-premises) accounts

You can configure password requirements and set timeout intervals for inactivity to secure your local Splunk SOAR (On-premises) accounts. Accounts that authenticate using single sign-on have their password requirements set by the individual service provider.

Perform the following steps to configure account security:

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Account Security**.
3. Configure the desired timeout settings for all local Splunk SOAR (On-premises) accounts.

Setting	Description
Inactivity Timeout	The number of minutes with no activity between the user's browser and the web server before the user is logged out.
Absolute Timeout	The number of minutes after which a local user is logged out, regardless of activity. Some pages, such as the home page and Investigation have constant activity in the form of widgets and dashboards that are updated automatically without user intervention. Setting an absolute timeout is a security precaution to make sure that only authorized users are accessing your Splunk SOAR (On-premises) system.

4. Configure the password requirements for your local Splunk SOAR (On-premises) accounts.

Setting	Description
Length	The minimum required length for any user password. This length can be overridden based on other password configurations. For example, if you set the Length to 8 characters, but also require 5 capital letters and 5 digits, then

Setting	Description
	the minimum length of the password is 10 characters.
Digits	The number of unique digits 0-9 required in the password.
Special Characters	The number of unique special characters required in the password.
Capital Letters	The number of unique capital letters required in the password.

Configure single sign-on authentication for Splunk SOAR (On-premises)

Splunk SOAR (On-premises) supports using Single sign-on (SSO) to authenticate Splunk SOAR (On-premises) users.

Single sign-on (SSO) systems allows users to be authenticated once, then use multiple, distinct services or applications without having to reauthenticate for each application or service. Single sign-on systems rely on an identity provider, such as LDAP, to authenticate the user, then provide an authentication token which applications, such as Splunk SOAR (On-premises), then use to log the user in. For an overview of single sign-on, see the Single sign-on article on Wikipedia.

Splunk SOAR (On-premises) supports any combination of local users and SSO users for your deployment, any combination of SSO providers, and multiple instances of any provider type.

You can configure SSO in Splunk SOAR (On-premises) with the following identity providers:

- LDAP
- OpenID
- SAML2

Configure SSO authentication using LDAP

To configure SSO authentication using LDAP as the identity provider, do the following steps:

1. From the Main Menu, select **Administration**.
2. Select **Users Management**.
3. Select **Authentication**.
4. LDAP is selected by default. Toggle the switch in the LDAP field to **ON** to enable LDAP configuration.
5. Complete the fields to configure SSO authentication using LDAP:

Field	Description
Active	<p>Use this checkbox in conjunction with Add Another at the bottom of the page. You can have multiple LDAP servers and the Active checkbox determines which ones are used by Splunk SOAR (On-premises) for authentication. The toggle button in the LDAP field enables LDAP authentication for all servers which are marked Active.</p> <p>If there are multiple LDAP servers, Splunk SOAR (On-premises) searches each server in a random order to find a match for the username. If the same username exists on multiple servers, the first one matched is used. If this match happens to be for a different user and not the user who is attempting to login, then authentication fails.</p>
Require TLS/SSL encryption	Determines whether secure LDAP connections are required. Enable TLS/SSL encryption to check the server certificate against the Splunk SOAR (On-premises) certificate store. See Manage Splunk SOAR (On-premises)'s certificate store .

Field	Description
Provider Name	The name of the SSO provider. Specify a unique name to easily identify this provider.
Server	The DNS name or IP address for your AD/LDAP Server, without <code>http://</code> or <code>https://</code> . If you plan to use SSL, you must supply a DNS name that matches the certificate.
Domain	The domain name of your organization such as <code>corp.yourorganization.com</code> , used to generate DNS. This field is used as part of the LDAP query.
Bind Username	The username for authenticating to the LDAP server. It will ideally be a service account specifically set up for this purpose, not one belonging to a human user.) This will allow you to grant the account the minimal permissions necessary, set account expiration off, and other protective measures to track how the account is used. If the account is set to expire or requires a password change, do these tasks manually and also update the Splunk SOAR (On-premises) system settings to reflect the same. The account will need to be able to query LDAP users and their properties.
Password	The password for the username to authenticate to the LDAP server.
Test User	The username of an active user who would typically log in to Splunk SOAR (On-premises). Use this to verify that user search is working correctly.
Test Group	The name of a group of which the Test User is a member. Use this to confirm that the group mapping will work. Leave this field blank if you are not using group mapping.
Manage password using Thycotic Secret Server	Manage user credentials using Thycotic Secret Server. If this is checked, you must also provide the Folder , Key , and Thycotic FieldName values. See Manage your organization's credentials with a password vault .

6. Click **Test Authentication** to test that Splunk SOAR (On-premises) can communicate with and query the LDAP server. Your LDAP settings will automatically be saved if the result is success. Or you can click **Save Changes** to save the settings without testing them.

Some LDAP provider specific things to watch for:

- On Microsoft Active Directory LDAP servers, the user authentication uses the email-like form of the username, like `ldap-client@splunk.com`. The username is appended with the domain name.
- You may need to enter **Advanced** settings non-Microsoft LDAP servers. Consult the manual for the LDAP software your organization uses.

If you need additional assistance, contact Phantom Support. See [Where to get help](#).

LDAP provider names must be unique. Using multiple LDAP providers with the same name is not supported.

Configure group mappings for LDAP SSO authentication

Configure a group mapping to map an LDAP group such as Incident Response to a Splunk SOAR (On-premises) role such as Automation Engineer. Doing so enables you to automatically use your LDAP groups to determine who can log into Splunk SOAR (On-premises) and which actions each user is able to perform after they log in. Click **Add Mappings** to create a new mapping. You can configure multiple mappings.

Each LDAP user must be mapped to at least one group to enable that user to login to Splunk SOAR (On-premises) without manually creating the user account in Splunk SOAR (On-premises).

Role mapping is done at login time, meaning that if the Splunk SOAR (On-premises) administrator changes a role mapping that would affect a logged-in user, then that user will retain the old role(s) until they log out and log back in again.

Configure external attribute mapping for LDAP SSO authentication

In some cases you may need to specifically call out external attributes which should be mapped to Splunk SOAR (On-premises) user attributes. Click **Add Mapping** to select a Splunk SOAR (On-premises) user attribute to map, then use the text field to enter the name of the attribute found in your LDAP user's profile.

Configure SSO authentication using SAML2

To configure SSO authentication using SAML2 as the identity provider, perform the following tasks:

1. From the Main Menu, select **Administration**.
2. Select **Users > Authentication**.
3. Click **SAML2**.
4. Click the toggle in the SAML2 field to enable SAML2 configuration.
5. Complete the fields to configure SSO authentication using SAML2:

Field	Description
Active	<p>Use this checkbox in conjunction with Add Another at the bottom of the page. You can have multiple SAML2 servers and the Active checkbox determines which ones are used by Splunk SOAR (On-premises) for authentication. The toggle button in the SAML2 field enables SAML2 authentication for all servers which are marked Active.</p> <p>If there are multiple SAML2 servers, Splunk SOAR (On-premises) searches each server in a random order to find a match for the username. If the same username exists on multiple servers, the first one matched is used. If this match happens to be for a different user and not the user who is attempting to login, then authentication fails.</p>
Require TLS/SSL encryption	Determines whether encrypted connections are required. Enable TLS/SSL encryption to check the server certificate against the Splunk SOAR (On-premises) certificate store. See Manage Splunk SOAR (On-premises)'s certificate store .
Provider Name	The name of the SSO provider. Specify a unique name to easily identify this provider.
Single sign-on URL	The URL that users are directed to for logging in.
Issuer ID	The unique identifier provided by the identity provider.
Metadata URL	The URL hosted by your identity provider containing information about the provider configuration. If you specify a valid Metadata URL, do not leave the Metadata XML field blank.
Metadata XML	XML code containing information about the provider configuration. If you specify valid XML in this field, you can leave the Metadata URL field blank.
Phantom Base URL	The URL used to redirect users back to Splunk SOAR (On-premises). This URL must be reachable by users trying to log in.
Advanced Settings	<p>Click Advanced to configure the following advanced settings:</p> <ul style="list-style-type: none">◆ Select Response Signed to require a signed response from the identity provider.◆ Select Request Signed to require a signed request from the identity provider.◆ Select Assertion Signed to require a signed assertion containing the user attributes from the identity provider.◆ Type an EntityID/Audience to configure an entity ID for the service provider. This is used when defining the audience restriction on the identity provider. A value for this field must be included.◆ Type a Group Key to identify the group membership data within the attributes passed back from the identity provider. Also specify a Group Delimiter if groups are passed back as a single element with a delimiter, instead of separate attribute values.

Field	Description
	<ul style="list-style-type: none"> ◆ Configure Groups. See Configure group mappings for LDAP SSO authentication for more information about group mapping. ◆ Configure External Attributes. See Configure external attribute mappings for LDAP SSO authentication for more information about external attributes mapping. If user name mapping is not provided in the assertion, Splunk SOAR (On-premises) will default to using the value specified in NameID field.

6. Click **Save Changes**.

Configure SSO authentication using OpenID

To configure SSO authentication using OpenID as the identity provider, perform the following tasks:

1. From the Main Menu, select **Administration**.
2. Select **Users > Authentication**.
3. Click **OpenID**.
4. Click the toggle in the OpenID field to enable OpenID configuration.
5. Complete the fields to configure SSO authentication using OpenID:

Field	Description
Active	<p>Use this checkbox in conjunction with Add Another at the bottom of the page. You can have multiple OpenID servers and the Active checkbox determines which ones are used by Splunk SOAR (On-premises) for authentication. The toggle button in the OpenID field enables OpenID authentication for all servers which are marked Active.</p> <p>If there are multiple OpenID servers, Splunk SOAR (On-premises) searches each server in a random order to find a match for the username. If the same username exists on multiple servers, the first one matched is used. If this match happens to be for a different user and not the user who is attempting to login, then authentication fails.</p>
Require TLS/SSL encryption	Determines whether encrypted connections are required. Enable TLS/SSL encryption to check the server certificate against the Splunk SOAR (On-premises) certificate store. See Manage Splunk SOAR (On-premises)'s certificate store .
Provider Name	The name of the SSO provider. Specify a unique name to easily identify this provider.
Issuer	The base endpoint provided by OpenID. Configuration is based on the discovery document located at <code><endpoint>/.well-known/openid-configuration</code> .
Client ID	Provided by OpenID.
Client Secret	Provided by OpenID.
Phantom Base URL	The URL used to redirect users back to Splunk SOAR (On-premises). This URL must be reachable by users trying to login.
Advanced Settings	<p>Click Advanced to configure the following advanced settings:</p> <ul style="list-style-type: none"> ◆ Enter Scopes to include custom scopes or to limit the scopes requested by Splunk SOAR (On-premises). The <code>openid</code> scope is required. ◆ Set the Token Auth Method to <code>client_secret_post</code> or <code>private_key_jwt</code>, depending on the configuration of your identity provider. ◆ Specify a Resource Identifier if a specific resource other than the default <code>userinfo</code> endpoint is required to obtain user data. ◆ Enter a Group Key to identify the group membership data within the attributes passed back from the identity provider. Also specify a Group Delimiter if groups are passed back as a single element with a delimiter, instead of separate attribute values. ◆ Configure Groups. See Configure group mappings for LDAP SSO authentication for more information about group mapping.

Field	Description
	◆ Configure External Attributes . See Configure external attribute mappings for LDAP SSO authentication for more information about external attributes mapping.

6. Click **Save Changes**.

Secure Splunk SOAR (On-premises) using two factor authentication

Duo is integrated with Splunk SOAR (On-premises) to enable two factor authentication. When enabled, two factor authentication applies to all local Splunk SOAR (On-premises) users. Splunk SOAR (On-premises) sets each user's email address as the Duo username. If an email address is not available, then the username is used.

Perform the following steps to enable two factor authentication in Splunk SOAR (On-premises):

1. Create a web SDK application in the Duo administrative interface. Refer to your Duo documentation for more information.
2. When the web SDK application integration is ready, record the following information to provide to Splunk SOAR (On-premises):
 - ◆ Integration key
 - ◆ Secret key
 - ◆ API hostname
3. In Splunk SOAR (On-premises), from the **Home** menu, select **Administration**.
4. Select **User Management > Two Factor**.
5. Check the **Enable Duo Two Factor Authentication** checkbox.
6. Provide the information you collected in the **Integration Key**, **Secret Key**, and **API Hostname** fields.
7. Click **Test Duo Connectivity** to verify the keys and hostname are correct.
8. Click **Save Changes**.

Disable two factor authentication for the default admin account as a failsafe mechanism so there is at least one account that can log in to administer Duo settings if the integration breaks.

With two factor authentication enabled, two new fields appear in the Edit User page:

- **Two Factor Authentication**. Set this field to **Duo** to enable two factor authentication. Select **None** to disable two factor authentication.
- **Duo Username**. Use this field to make sure the Splunk SOAR (On-premises) and Duo usernames match. For example, a user's Splunk SOAR (On-premises) username is **jsmith** but his Duo username is **jsmith@splunk.com**. In this case, set the Duo username to **jsmith@splunk.com** so the correct Duo user is used when logging in to Splunk SOAR (On-premises).

Configure role based access control inside Splunk apps

Splunk SOAR (On-premises) supports granular asset access control inside of Splunk SOAR (On-premises) apps to ensure that only authorized access to the app is allowed. Asset access control works on an authorized basis, with a default-deny policy.

When granular asset access control is enabled, only users or groups with explicit permissions are able to perform actions in a Splunk SOAR (On-premises) app. Configure user and group permissions on all configured apps before enabling

granular asset access control.

To set up a single user to have access the "lookup domain" action on the Google DNS asset:

1. From the **Home** menu, select **Apps**.
2. Click **1 configured asset** to expand the section.
3. Click **Google DNS** to edit the asset.
4. Click the **Access Control** tab.
5. Click **Edit**.
6. Select **lookup domain** from the **App Action** drop-down list.
7. Select the user desired user name then click the right arrow in order to move the user from the **Users and Roles** list into the **Approved Users and Roles** list.
8. Click **Save**.

Now enable granular asset access control so that the permission set above takes effect.

1. From the **Home** menu, select **Administration**.
2. Select **User Management > Asset Permissions**.
3. Check the **Enable granular Asset Access Control** checkbox.
4. Confirm that you want to change global asset permissions.
5. Click **Save Changes**.

Secure Splunk SOAR (On-premises) by configuring an account password expiration

A common security practice is to set a user account password expiration after a specific period of time, such as every 90 days. Splunk SOAR (On-premises) does not provide the ability to configure an account password expiration. As a system administrator, you need to define, implement, and administer password expiration policies in accordance with your organization's requirements.

Take note of the following if you configure password expiration policies in your environment:

- Do not configure a password expiration for the root account. This can cause issues such as the `crond` daemon stopping, `logrotate` failing to trim logs, data ingestion pausing, or services failing to restart.
- Do not configure a password expiration in AWS environments. By default, AWS instances use key pairs for authentication. If a user account expires, the account is blocked from accessing the AMI unless the user has configured an account password and can provide it when prompted. Key pair authentication doesn't work for expired accounts.

To reset a user's account expiration date, shut down the AWS instance and update user data through the AWS console. For example, to set an account expiration date of January 1, 2023:

```
# cloud-boothook
# !/ bin / bash
# chage -E "Jan 1, 2038" user
```

Specify a date in the future but before Jan 19, 2038. The latest time that can be represented in Unix's signed 32-bit integer time format is 03:14:07 UTC on Tuesday, 19 January 2038.

You can configure the user account to never expire:

```
# chage -m 0 user
# chage -M 99999 user
# chage -l user
Last password change      : Dec 10, 2016
Password expires          : never
Password inactive         : never
Account expires           : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

Manage your registered mobile devices

Enable or disable registered mobile devices

You can allow users to register mobile devices and use Splunk mobile apps with your Splunk SOAR (On-premises) instance.

The **Enable Mobile App** toggle is disabled by default. Toggle this switch so that users see the **Mobile Device Registration** tab in their Account Settings and use it to register mobile devices. This toggle is available for viewing and editing if your role has basic permissions in the system settings with view and edit selected. See [Manage roles and permissions in Splunk SOAR \(On-premises\)](#) for more information about roles and permissions.

Enable mobile device registration with Splunk SOAR (On-premises)

Enable registration of mobile devices by doing the following:

1. From the main menu, select **Administration**.
2. Select **Mobile**.
3. Toggle the **Enable Mobile App** switch to the On position.
4. Click **Confirm**.

Disable mobile device registration with Splunk SOAR (On-premises)

Disable registration of mobile devices by doing the following:

1. From the main menu, select **Administration**.
2. Select **Mobile**.
3. Toggle the **Enable Mobile App** switch to the Off position.
4. Click **Confirm**.

When mobile device registration is disabled, traffic doesn't flow between Splunk SOAR (On-premises) and mobile devices, and the **Mobile Device Registration** tab is no longer visible in Account Settings.

When the mobile feature is re-enabled, previously registered devices can resume communication with Splunk SOAR (On-premises).

View registered mobile devices

When the **Enable Mobile App** toggle is switched on, you can see all the mobile devices that are registered by Splunk SOAR (On-premises) users. You can see them as a list, or you can search for them by device name.

If users remove their own devices from their account settings, then the devices automatically disappear from this list. Users can also remove their registration from the Splunk Mobile app, and the devices also automatically disappear from this list. You can remove your registration only while the **Enable Mobile App** toggle is switched on. If the toggle is switched off, then the removal message is not received.

Remove registered mobile devices

When the **Enable Mobile App** toggle is switched on, you can remove a Splunk SOAR (On-premises) user's registered device.

If you delete a user account, Splunk SOAR (On-premises) removes the registration for all of the devices that belong to the user.

When you no longer want a user to have access to the mobile app from a particular mobile device, do the following to remove the registered device:

1. Locate the device by its name in the table.
2. In the **Action** column, click **remove**.
3. When prompted, confirm by clicking remove.

Monitor your Splunk SOAR (On-premises) system activity

Monitor the health of your Splunk SOAR (On-premises) system

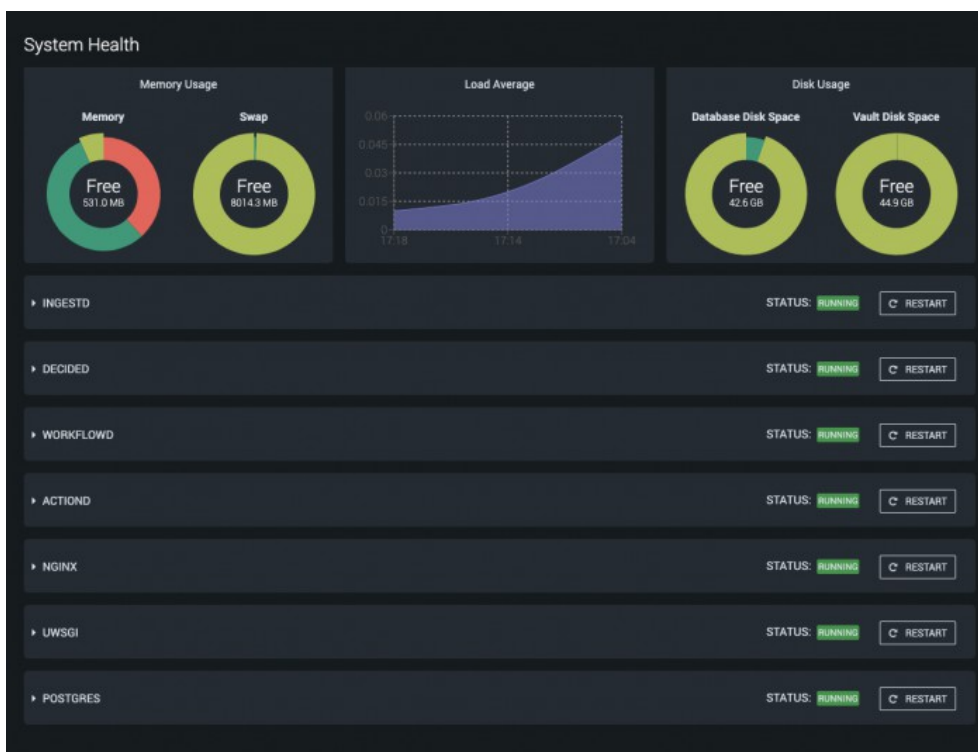
Use the System Health page to view a summary of your Splunk SOAR (On-premises) instance. The System Health page includes the following information:

- Running status of Splunk SOAR (On-premises) processes
- Resource consumption
- Health and status of critical processes

Use the System Health page as a starting point to begin troubleshooting issues. Splunk support might ask for the results of this page to start a troubleshooting investigation.

Perform the following tasks to get to the System Health page:

1. From the main menu, select **Administration**.
2. Select **System Health > System Health**.



The following image shows the System Health page for a standalone, non-clustered Splunk SOAR (On-premises) instance. Additional selections such as a selector for individual nodes and ClusterD statistics are available on the System Health page in a clustered deployment. A clustered deployment doesn't have the Database Disk Space panel since the database in a cluster lives on a different host.

The top row of graphs shows you the status of the following system-wide resources:

- Memory usage
- Load average
- Disk usage

Each row after the top row represents the individual system processes important to Splunk SOAR (On-premises). Verify that each process has a green **Running** status icon. Click **Restart** if you need to restart any one of the individual processes.

Splunk SOAR (On-premises) runs on top of Linux, so these graphs can be interpreted as you might on any Linux system. On a fairly idle Splunk SOAR (On-premises) system, there might be a significant amount of free memory, unused swap, and a lower load compared to the number of allocated CPU cores. There might also be more free disk space for the database and files.

The Splunk SOAR (On-premises) processing daemons `IngestD`, `DecideD`, `WorkflowD`, and `ActionD` perform various scheduling, decision, and management functions as well as critical background functions. All four must be running in order for Splunk SOAR (On-premises) to work properly. Splunk SOAR (On-premises) also relies on `HTTPD` and `Postgres`, which is the database.

If you registered a mobile device and **Enable Mobile App** is on, you can see the following behaviors in Splunk SOAR (On-premises):

- The `ProxyD` daemon starts automatically. The `WatchdogD` daemon keeps track of the toggle switch position and adds or removes the `ProxyD` daemon from the system startup list depending on the status.
- The System Health page also includes usage statistics for the `ProxyD` daemon. See [Enable or disable registered mobile devices](#) for information about the Enable Mobile App toggle.

View how much data is ingested in Splunk SOAR (On-premises) using ingestion summary

The ingestion summary page provides a summary of container ingestion over time and currently scheduled periodic ingestions. Use the Ingestion Summary page to get a broad view of how much data is coming into Splunk SOAR (On-premises) and how that amount is trending over time.

Perform the following steps to view ingestion summary details:

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Ingestion Summary**.

The Ingestion Summary table shows a line chart with the total number of successful and failed container ingestions across all Data Sources and ingestion methods. Use the drop-down list to change the time range of the chart. You can select one of the following time ranges:

- Last 24 hours

- Last 7 days
- Last 30 days

The Scheduled Ingestion table lets you track the configuration of all Data Sources that currently have scheduled polling enabled:

- Time shows the date and time when that Data Source was last set to enable scheduled polling.
- Interval shows how often that Data Source is scheduled to poll.
- Container shows the label that will be applied to containers ingested from that Data Source.
- Asset shows the name of the Data Source asset.
- App shows the name of the Data Source app.
- Action shows the name of the action that will be used to ingest data.

View ingested container statistics using Ingestion Status

Use the Ingestion Status page to see high-level statistics about ingested containers.

To view ingestion status details, perform the following steps:

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Ingestion Status**.

The Ingestion Stats table shows one row for each unique combination of ingestion status, container label, asset, and action. These rows allow you to get a better sense of how many containers are being ingested through each ingestion mechanism. Some containers don't come from an asset because they are manually added by a user, which results in a row with an action such as "User add container".

The Ingestion Errors table lists any failed ingestions. Use the information in the start time, end time, asset, app, and action fields to start debugging the failure.

Configure the logging levels for Splunk SOAR (On-premises) daemons

You can adjust logging levels for each daemon running in Splunk SOAR (On-premises) to help debug or troubleshoot issues.

Splunk SOAR (On-premises) daemons

The following daemons in Splunk SOAR (On-premises) work to control collection and scheduling tasks in the background independently from the Splunk SOAR (On-premises) web interface:

Daemon	Description
Action daemon	<p>Responsible for launching actions by putting into effect the appropriate app on the specified asset. Also responsible for the debug log that says what version of Python is being used. The debug log for Python 3 shows <code>Running executable: spawn3</code>.</p> <p>The following key actions are logged by this daemon:</p> <ul style="list-style-type: none"> • Manual actions run against any configured asset

Daemon	Description
	<ul style="list-style-type: none"> Scheduled actions against any configured asset
Cluster daemon	Responsible for communicating with cluster nodes. Available only in clustered environments.
Decide daemon	<p>Responsible for operating on incoming data.</p> <p>The following key actions are logged by this daemon:</p> <ul style="list-style-type: none"> Launching active playbooks against new containers for associated Operates On types Playbook validation Playbook loading and use Custom automation or playbook APIs Prompting the action daemon for action and app use Prompting the workflow daemon for the approval workflow process Processing approval response and results from the workflow daemon Matching app execution to specific action results Playbook debugging Counting licensed action uses
Ingest daemon	<p>Responsible for ingesting data into the product.</p> <p>The following key actions are logged by this daemon:</p> <ul style="list-style-type: none"> Ingestions from data sources that use polling to get new data Asset health reporting, also known as, connectivity checking for the Asset Health dashboard component Configuration changes to any assets or any app-specific configuration Manual Test Connectivity actions launched directly from any asset
Proxy daemon	Responsible for communicating with Splunk mobile apps to register devices and send notifications to mobile users. This daemon is available only when the mobile app feature is enabled.
Watchdog daemon	<p>Responsible for tracking the status of other daemons and adding or removing them in the system startup list.</p> <p>The following key actions are logged by the watchdog daemon:</p> <ul style="list-style-type: none"> Installation of new apps Health monitoring the Splunk SOAR (On-premises) deployment Maintenance of other Splunk SOAR (On-premises) platform daemons and components Restarts of the Splunk SOAR (On-premises) platform daemons and components
Workflow daemon	<p>Responsible for managing approval requests to action reviewers and asset owners.</p> <p>The following key actions are logged by the workflow daemon:</p> <ul style="list-style-type: none"> Processing and launching approval processes and managing approval escalations Sending user email notifications for container assignment, expiry, manual action requests, and other email templates

Configure the logging level for each Splunk SOAR (On-premises) daemon

Adjust the logging levels as needed to assist Splunk SOAR (On-premises) Support with troubleshooting any issues you might experience.

- From the main menu, select **Administration**.
- Select **System Health > Debugging**.
- Select a logging level for each daemon you want to change. The log levels determine the message types that are written to each daemon's corresponding log file. The Debug level is the most verbose level of logging and is useful for troubleshooting.

4. (Optional) Click **Download Logs** to download a copy of the current log files for manual investigation or a submission to support.
A zipped TAR archive of the logs is downloaded to `/var/log/phantom`.
5. Click **Save Changes**.

At the operating system level, log files are in `/var/log/phantom`. In a non-root installation, the path is `${phantom_home}/var/log/phantom`. Adjust logging levels only at the direction of Splunk SOAR (On-premises) Support.

Configure the logging format

You can configure the logging format for JSON or plaintext. Configure the format through the CLI as the root user. For example: `â [root@phantom]# phenv set_preference â logging-format <format>`
For NRI, you don't need to run the command as root user.

The valid formats are `â plainâ` and `â jsonâ`.

Take note of the following items:

- Running the command restarts Splunk SOAR (on-prem).
- If you are using clusters, you need to run the command on each node.
- You can reconfigure the plaintext format at any time.
- If you're ingesting the logs into Splunk, the ITSI app doesn't support the json format yet.

Example plaintext log structure

See the following sample of a common log format:

```
Oct  5 22:55:18 localhost DECIDED[7177]: TID:7422 : WARNING: DECIDED : rules_engine.cpp : 1503 :
DECIDED_CMD_PROCESS_CONTAINERS : All rules FAILED t
```

This table summarizes the structure of the example log message.

Log message content	Description
Oct 5 22:55:18	Timestamp of when the log message was generated.
localhost	Name of the host where the log message was generated.
DECIDED[7177]:	Name of the component and process ID (PID) generating the message.
TID:7422:	Threat ID (TID) of the message.
WARNING:	Log level or class of the message.
DECIDED:	Functional component that generated the log message.
rules_engine.cpp:	Source file applicable to the log message.
1503:	Line number in the source file that caused this log message to be

Log message content	Description
	generated.
DECIDED_CMD_PROCESS_CONTAINERS: All rules FAILED to process the container: 2964. Error: Playbook 'local/test11 (version: 1, id: 711)' cannot be executed since it is: NOT ACTIVE, ENABLED and VALID	The log message.

Monitor Splunk SOAR (On-premises) logs using an external SIEM

You can use an external SIEM such as Splunk Enterprise to monitor the status of your Splunk SOAR (On-premises) instance using the log files from Splunk SOAR (On-premises). Monitor errors in actions execution through the ingestion of the actiond.log file, and you can find platform information in the watchdogd.log file.

These files are constantly updated by the components of the Splunk SOAR (On-premises) platform. Use a service to monitor the log file regularly, or copy and forward the log files on a regular basis.

Rotate your Splunk SOAR (On-premises) logs when they get too large

The Splunk SOAR (On-premises) logs in `/var/log/phantom` are automatically rotated when they reach 50MB in size. Splunk SOAR (On-premises) keeps a running archive of the last 10 rotated logs. The oldest log is deleted when a new log is rotated in. You can configure log rotation settings for your organization's IT requirements by modifying the `phantom_logrotate.conf` file and restarting `rsyslog` for the changes to take effect.

Perform the following tasks to configure log rotation on Splunk SOAR (On-premises):

1. Run the `logrotate --version` command to find your Logrotate version. For example:

```
[root@phantom]# logrotate --version
logrotate 3.8.6
```

2. Configure logrotate using the instructions based on your Logrotate version.

- ◆ If you are running Logrotate version 3.8.x, use a configuration with the `su` directive. By default, the configuration file is located in `/etc/logrotate.d/phantom_logrotate.conf`. For example:

```
# Note: If you change size to anything approaching or greater than 100MB
# you need to updated the rsyslog.conf as well as he has a maxsize
# of 100MB which triggers logrotate.
/var/log/phantom/actiond.log
/var/log/phantom/clusterd.log
/var/log/phantom/decided.log
/var/log/phantom/ingestd.log
/var/log/phantom/proxyd.log
/var/log/phantom/watchdogd.log
/var/log/phantom/workflowd.log
/var/log/phantom/spawn.log {
    su phantom phantom
    rotate 10
    size 50M
    start 1
    missingok
    create 0660 phantom phantom
    postrotate
        kill -HUP $(cat /var/run/syslogd.pid)
    endscript
}

# wsgi log is generated by wsgi itself.
# use copytruncate to avoid needing to restart uwsgi
#
/var/log/phantom/wsgi.log {
```

```

    su nginx phantom
    copytruncate
    rotate 10
    size 50M
    start 1
    create 0660 nginx phantom
}

# supervisord generated logs. these applications
# log to stdout/stderr which supervisord directs to files.
#
/var/log/phantom/supervisord.log
/var/log/phantom/-stderr.log
/var/log/phantom/-stdout.log {
    su phantom phantom
    copytruncate
    missingok
    rotate 10
    size 50M
    start 1
    create 0660 phantom phantom
}

# rsync logs are typically generated during warm-standby operation
# on the primary system. they are generated by the rsync tool itself.
/var/log/phantom/rsync*.log {
    su root root
    rotate 10
    size 10M
    start 1
    missingok
    create 0644 root root
}

```

- ◆ If you are running a Logrotate version older than 3.8.x, the default location and content of the Logrotate configuration file is `/etc/logrotate.d/phantom_logrotate.conf`. For example:

```

[root@phantom]# more /etc/logrotate.d/phantom_logrotate.conf
/var/log/phantom/*.log {
    rotate 10
    size 10M
    start 1
    create 0660 root phantom
    postrotate
        kill -HUP $(cat /var/run/syslogd.pid)
    endscript
}

```

3. After configuring Logrotate in the configuration file, restart `rsyslog` using the `/opt/phantom/bin/phsvc restart rsyslog` command. On a privileged instance of Splunk SOAR (On-premises):

```

[root@phantom]# /opt/phantom/bin/phsvc restart rsyslog
Shutting down system logger:          [ OK ]
Starting system logger:                [ OK ]

```

On an unprivileged instance of Splunk SOAR (On-premises):

```

service rsyslog restart
Redirecting to /bin/systemctl restart rsyslog.service

```

Enable and download audit trail logs in Splunk SOAR (On-premises)

Enable audit trail logging to help you track the activities of various components in Splunk SOAR (On-premises). Once enabled, audit trail logs can be downloaded and included as evidence in an investigation, or analyzed when troubleshooting an issue.

Enable audit trail tracking

By default, all audit tracking in Splunk SOAR (On-premises) is disabled. Perform the following tasks to enable audit trail tracking in Splunk SOAR (On-premises):

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Audit Trail**.
3. Click **Manage Audit Trail**.
4. Select the product areas for which you want to enable audit tracking.
5. Click **Save**.

Splunk SOAR (On-premises) immediately starts tracking audit events for the selected items.

Even when the audit categories are disabled, events such as action and playbook runs are automatically tracked and logged as audit events.

Export audit logs

To export audit logs for a particular product, make sure you enabled audit tracking for that product area.

After you enable audit logging, use the rest of the **Audit Trail** to configure the audit logs you want to download as a CSV file. Perform the following steps to export audit events to a CSV file for download. This example shows you how to configure audit logging for containers and download a CSV file.

First, enable audit logging for containers:

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Audit Trail**.
3. Click **Manage Audit Trail**.
4. Click the **Container** toggle to enable audit tracking for containers.
5. Click **Save**.

Next, export a CSV file. This example exports the CSV file for a specific container.

1. From the Audit Trail page in the Audit Type section, click **Custom**.
2. Click **Containers**.
3. In the drop-down list for Containers, select **Custom**.
4. Specify the container ID, such as 123456. Only the audit trail for this specific container is downloaded.
5. By default, the audit trail from the last 30 days is downloaded. Click **Custom** in the Audit Range Time Frame field to configure a specific date range.
6. Click **Download** to download the CSV file.

Export audit logs for multiple users

Exporting audit logs for multiple users adds a new input field where you can specify a container to report on. When you download the audit logs, you receive only audit events for the container specified instead of all containers. Other categories might let you pick from a list, such as Users.

You can download audit logs for multiple users. Use `%1E` as the separator. For example, if you want to specify `user1` and `user2`:

```
user1%1Euser2
```

Export audit logs for roles

Roles return two types of events. First, creating a role or changing permissions in it shows up as audit events for that role. Second, the logs show audit events for users currently in that group. In other words, the logs treat the role like a user group, and shows events for those users in it. See [Accessing Audit Data](#) in the REST API Reference for more information.

Required privileges for enabling audit trail

In order to access the Audit Trail page, users must have a role with the View System Settings privilege. If they want to view or change anything under the Manage Audit Trail, then they also need the Edit System Settings privilege.

With only the View System Settings privilege, the user can't access all audit items. Attempting to download with the Audit Type section set to All results in an error.

A user with only some of the required privileges can switch to Custom and select only the items they have the rights to access. The privileges for each of the items are as follows:

Audit Trail Area	Required privileges
Authentication	View Users and Roles
Administration	View System Settings
User	View Users and Roles
Role	View Users and Roles
Playbooks	View Playbooks
Containers	View Containers

Enable the audit trail for individual objects

Users can access audit information in two places: on the page for a playbook and on the Investigation page for a container.

Download a playbook's audit trail

Perform the following steps to download an audit trail for a playbook:

1. Open the playbook.
2. Click **Playbook Settings**.
3. Click **Audit Trail** to download a CSV file containing the audit information for this playbook.

Download a container's audit trail

Perform the following steps to download an audit trail for a container:

1. Click the container to view the container.
2. Click the ... icon, and then select **Audit**.

A CSV file is downloaded containing the audit information related to this container.

Locate long-running playbooks for debugging or troubleshooting in Splunk SOAR (On-premises)

Use the Automation page to locate playbooks that have been running for a long time.

As an example, suppose your system health indicators show heavy utilization, but you are not aware of any process that must be running for a long period of time. You can start on the Automation page to see if any playbooks might be running intensive applications or experiencing other problems.

Perform the following tasks to access the Automation page:

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Automation**.

View the playbook run history in Splunk SOAR (On-premises)

You can view the history of playbook runs on your Splunk SOAR (On-premises) instance.

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Playbook Run History**.

The Playbook Run History page displays a sortable table of playbook runs. Each column except for Git Commit is sortable. The table displays the following columns:


Column title	Description
Name	The name of the playbook that was run.
Run ID	The numeric ID of the Playbook Run.
Event ID	The numeric ID of the event the playbook was run against.
Label	The label the playbook was run against, such as event.
Start Time	The time the playbook was started.
End Time	The time the playbook run finished.
Status	Whether the playbook run succeeded or failed.
Git Commit	The Git commit ID from when the playbook version was committed to the included Git source control module.
Run By	The name of the user who ran the playbook.

View Playbook Run Statistics

Playbook Run Statistics are available in playbooks created in the Visual Playbook Editor, starting in Splunk SOAR (On-premises) version 5.3.3.

Learn how your playbooks are performing, and troubleshoot potential issues, by viewing Playbook Run Statistics. You can view statistics for specific playbook runs at the playbook level and for each block and custom function within a playbook after the playbook has run. Compare statistics for this playbook to other specific runs or to run averages for that playbook to detect differences.

To view the Playbook Run Statistics for one of your playbooks, complete these steps:

1. Open the playbook within the Visual Playbook Editor.
2. In the top right corner of the screen, click the more icon , then click **View Run Statistics**. The Playbook Run Statistics dialog box displays.
3. In the first column in the Playbook Run Statistics dialog box, use the Playbook Run field to specify the playbook run you want to investigate. Use the filter to search for a specific run, if needed.
4. (Optional) In the second column, choose whether you want to compare against the average runs for that playbook for the last 24 hours, 7 days, or all runs for that playbook.
5. (Optional) Use the third column to add another comparison for the same playbook.
6. (Optional) Add a fourth column by clicking the plus icon.

To show all available playbook run statistics, expand each of the sections in the dialog box.

To view information for a specific block in playbook, click that individual playbook block. In the left panel, click the **Stats** tab.

Playbook Run Statistics and older playbooks

Playbook Run Statistics are available in playbooks created in the Visual Playbook Editor, starting in Splunk SOAR (On-premises) version 5.3.3. To view these statistics with playbooks you created with the Visual Playbook Editor in versions lower than 5.3.3, you must first save the playbook to automatically add the code required for statistics. The added statistics code does not affect your custom code.

You can view statistics for these updated playbooks for runs performed after you have updated the code.

Description of Playbook Run Statistics

This table describes Playbook Run Statistics for playbook blocks.

Field	Description
DB Queries	Number of requests to the database made by this block
DB Query Latency	Average amount of time for the block request to reach the database, in seconds
Duration	Amount of time for the block to execute, in seconds
HTTP Bytes in Requests	Number of bytes transmitted by the block through HTTP requests
HTTP Bytes In Response	Number of bytes received by the block through HTTP requests
HTTP Latency	Average amount of time between HTTP requests and responses experienced by the block, in seconds
HTTP Requests	Number of HTTP requests initiated by the block

Field	Description
Times Called	Number of times the block was called
Times Succeeded	Number of times the block completed successfully

View the action run history

You can view the history of actions run on your Splunk SOAR (On-premises) instance.

1. From the **Home** menu, select **Administration**.
2. Select **System Health > Action Run History**.

The Action Run History page displays a sortable list of action runs. Each column except for View Results is sortable. The table displays the following columns:

Column name	Description
Name	The name of the action that was run.
Run ID	The numeric ID of the action that was run.
Event ID	The numeric ID of the event the action was run against.
Start Time	The time the action started.
End Time	The time the action finished.
Status	Whether the action succeeded or failed.
Prompted	If the action taken was a prompt or manual task action, the ID of the user assigned the action appears here.
Run By	The name of the user who ran the action.
View Results	A hyperlink to the action results in Investigation. For prompt or manual task actions, the link opens a window containing the prompt or task results.

Use ITSI to monitor the health of your Splunk SOAR (On-premises) deployment

Splunk IT Service Intelligence (ITSI) is a scalable IT monitoring and analytics solution that provides actionable insight into the performance and behavior of your IT operations. ITSI is built on the Splunk operational intelligence platform and uses the search and correlation capabilities of the platform to help you collect, monitor, and report on data from IT devices, systems, and applications.

As a Splunk SOAR (On-premises) administrator, you can use ITSI to monitor the health of your Splunk SOAR (On-premises) instance or cluster. For more information, see *About the Content Pack for Monitoring Phantom as a Service* in *Splunk ITSI Content Packs*.

Use Python scripts and the REST API to manage your Splunk SOAR (On-premises) deployment

Administrators can use scripts and the Splunk SOAR (On-premises) REST API to manage their Splunk SOAR (On-premises) deployment.

For example, this script uses the Splunk SOAR (On-premises) REST API to send an email alert when containers with the specified label and tag combination reach a predefined percentage of the total containers.

```
import requests
import urllib
import time
import json

try:
    requests.packages.urllib3.disable_warnings()
except:
    from requests.packages.urllib3.exceptions import InsecureRequestWarning
    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

config = {
    'url': 'https://127.0.0.1',
    'token': '', # unnecessary for localhost
    'label': '',
    'tag': '',
    'threshold': .1,
    'email': '',
    'email_asset': 'smtp'
}

headers = {}
token = config['token']
if token:
    headers = {'ph-auth-token': token}

total_url = '{0}/rest/container?_filter_status="closed"&_filter_label="{1}"'.format(config['url'],
config['label'])
response = requests.get(total_url, headers=headers, verify=False)
resp_json = response.json()
total = resp_json['count']

url = '{0}/rest/container?_filter_status="closed"&_filter_label="{1}"&_filter_tags_
_contains="{2}"'.format(config['url'], config['label'], config['tag'])
response = requests.get(url, headers=headers, verify=False)
resp_json = response.json()
count = resp_json['count']

if float(count) / total < config['threshold']:
    print "Threshold not hit. Taking no action."
    exit(0)

print "Hit threshold. Notifying {0}".format(config['email'])

if config.get('app_id'):
    appid = config['app_id']

else:
    build_action_url = '{0}/rest/build_action'.format(config['url'])
    response = requests.get(build_action_url, headers=headers, verify=False)
```

```

build_json = response.json()
assets = build_json['assets']
for asset in assets:
    if asset['name'] == config['email_asset']:
        appid = asset['apps'][0]

action_body = {
    'action': 'send email',
    'container_id': resp_json['data'][0]['id'],
    'name': 'notification email',
    'targets': [
        {
            'assets': [config['email_asset']],
            'parameters': [
                {
                    'to': config['email'],
                    'from': 'Phantom notifications',
                    'subject': 'You are opening a lot of alerts',
                    'body': 'Please consider opening fewer alerts. See
{0}/browse/{1}'.format(config['url'], urllib.quote(config['label']))
                }
            ],
            'app_id': appid
        }
    ],
    'type': 'generic'
}

action_url = '{0}/rest/action_run'.format(config['url'])
response = requests.post(action_url, data=json.dumps(action_body), headers=headers, verify=False)
print response.json()

```

Set the desired values in the `config` dictionary. This table defines the expected values.

Dictionary entry	Values
url	URL of the Splunk SOAR (On-premises) instance. Use the loopback address (127.0.0.1) if the script is run on the localhost.
token	Splunk SOAR (On-premises) API token for a remote connection. If the script is run on the localhost, you don't need to supply the API token.
label	The label name to check.
tag	The tag name to check for items with the required label.
threshold	A percentage, expressed as a decimal, of containers with the given label and tag that will trigger the alert.
email	The email address that receives the alert.
email_asset	The SMTP asset name from which the email server configuration is obtained.

Sample "config"

```

config = {
    'url': 'https://127.0.0.1',
    'token': '', # unnecessary for localhost
    'label': 'soc_alert',
    'tag': 'red_alert',
    'threshold': .1,
}

```

```
'email': 'soc@contoso.com',  
'email_asset': 'smtp'  
}
```

This script is provided as an example of ways Splunk SOAR (On-premises) administrators can use Python and the REST API to manage their on-premises deployment of Phantom.

On Splunk SOAR (On-premises) 5.0.1 or later releases you must use Python 3 to write your management scripts.

Manage Splunk SOAR (On-premises) Certificate Store

Splunk SOAR (On-premises) certificate store overview

Splunk SOAR (On-premises) has a certificate store used to validate certificates when forming connections to other servers. The certificates in the store are trusted certificate authority (CA) certificates from `mkcert.org` and are updated periodically. In almost all cases, Splunk SOAR (On-premises) can use its certificate store to validate any certificate issued by a commercial certificate authority (CA).

The default certificate store cannot be used to validate self-signed certificates, or certificates issued by an internal CA. You must add these custom certificates to the Splunk SOAR (On-premises) certificate store.

Important information about the Splunk SOAR (On-premises) certificate store:

- Certificates are stored in `<$PHANTOM_HOME>/etc/certs/`
- You add certificates to the `<$PHANTOM_HOME>/etc/cacerts.pem` file using the `import_cert.py` tool, located in `<$PHANTOM_HOME>/bin/`. See [Add or remove certificates from the Splunk SOAR \(On-premises\) certificate store](#).
- For more information about how to change the TLS certificate on the platform, see [Provide a valid SSL certificate for the connection between Splunk Phantom and Splunk Enterprise in the *Use the Splunk Phantom App for Splunk to Forward Events* manual](#).

Add or remove certificates from the Splunk SOAR (On-premises) certificate store

To add a custom certificate to the certificate store:

```
phenv python3 /opt/phantom/bin/import_cert.py -i /tmp/ca.crt  
/opt/phantom/bin/phsvc restart uwsgi
```

In this example, the `import_cert.py` script is copying the certificate file `ca.crt` to the `/opt/phantom/etc/certs/` directory, then consolidating all the files in that directory to the `/opt/phantom/etc/cacerts.pem` file. The `cacerts.pem` file is used by Splunk SOAR (On-premises) to verify all server certificates.

The `/opt/phantom/bin/phsvc restart uwsgi` restarts the web server so the updated `cacerts.pem` file is reloaded.

If you need to remove a certificate that you have previously installed, perform the following tasks:

1. Delete the file for that certificate from `/opt/phantom/etc/certs/`.
2. Run the `import_cert.py` script with no parameters.
3. Restart the web server.

Troubleshooting certificate issues

Even after importing the correct certificate, you might notice that the server still reports connectivity issues, which could be related to the certificate. In addition to the certificate being available for validation, it is important to remember some key points about certificate validation:

- The OpenSSL library used must validate a full certificate chain. This means that you cannot just install the end certificate, such as the one on the web server. If it was signed by a parent certificate, then the parent certificate is the one that must be installed. Though, if it's a true self-signed certificate, where it is signed by itself, and has no other parent, then install that certificate.
- Any required intermediate certificates must be present. Many CAs have a root certificate, and then one or more levels of intermediate, issuer, certificates, and then the actual server certificate. It's customary that the server be configured to serve both its own certificate as well as the intermediates, and that the client has the root to complete the chain. However, if the server is not configured to serve the intermediates, then the intermediates must also be installed in the certificate store.
- Certificates must be within their date range. That is, it must be after the **valid from** date and before the **expiration date** in the certificate.
- Certificates must use a valid Common Name (CN) or Subject Alternate Name (SAN) field and Splunk SOAR (On-premises) must be configured to use the resource by that name. Wildcard certificates will also work as expected. For example, you might have a server known as server.example.com at IP address 10.1.1.1. In order for the SSL/TLS connection to it to succeed, Splunk SOAR (On-premises) must be configured to use the full name, server.example.com. Using a short name of "server" or using the IP address 10.1.1.1 does not work.

Backup or restore your Splunk SOAR (On-premises) instance

Splunk SOAR (On-premises) backup and restore overview

Splunk SOAR (On-premises) includes a tool, `ibackup.pyc`, to back up and restore your Splunk SOAR (On-premises) data.

Regularly back up your Splunk SOAR (On-premises) deployment to safeguard your data in these cases:

- To restore your Splunk SOAR (On-premises) deployment in the event of a disaster
 - ◆ Restore a lost or failed file share
 - ◆ Restore a lost or failed PostgreSQL database
- To restore data from your Splunk SOAR (On-premises) deployment to another
 - ◆ Restore data from a Splunk SOAR (On-premises) instance or cluster to a new instance or cluster
 - ◆ Restore data from a standalone instance to a newly deployed cluster
 - ◆ Restore data from a clustered deployment to a standalone Splunk SOAR (On-premises) instance
 - ◆ Rebuild a Splunk SOAR (On-premises) cluster where the Splunk SOAR (On-premises) nodes have failed

Save your backups in a safe place, such as one that is not on the same disk, partition, or virtual machine as your Splunk SOAR (On-premises) instance.

Supported configurations

You can backup a Splunk SOAR (On-premises) deployment using any of these configurations:

- A privileged, standalone instance
- A privileged instance, external PostgreSQL database
- A privileged instance, external file shares
- An unprivileged, standalone instance
- An unprivileged instance, external PostgreSQL database
- An unprivileged instance, external file shares
- A privileged cluster
- An unprivileged cluster

You can build Splunk SOAR (On-premises) deployments from any supported installation method. See *Install Splunk SOAR (On-premises)* using the Amazon Marketplace Image in *Install and Upgrade Splunk SOAR (On-premises)*.

You must use the same operating system for a backup of a Splunk SOAR (On-premises) deployment as for the deployment itself.

If your deployment uses an external PostgreSQL 12.9 database, you cannot use the backup and restore feature. You must use another solution to backup or restore your Splunk SOAR (On-Premises) deployment.

Backup types

The `ibackup.pyc` tool is based on the open source `pgBackRest` project, and it supports full and incremental backups. Differential backups are not supported.

- A full backup includes all the file sets included in the What is in a full backup section on this page.
- Incremental backups contain the changes made to your deployment's PostgreSQL database and files since the last full or incremental backup was made, as well as metadata about previous backups.
- A configuration only backup, which makes a backup of all the Splunk SOAR (On-premises) configurations. This type of backup requires downtime.

For incremental backups, the metadata collected about previous incremental backups accrues cumulatively. To reset the collection of metadata, you must perform a full backup.

Backup levels and groups

Backups are created in `<PHANTOM_HOME>/phantom/data/backup/`. Each backup is part of a group, based on a full backup that is the base of the group, and is labeled by level.

For example, if the first backup on a Splunk SOAR (On-premises) instance is named "phantom_backup_group_0_level_0.tar". Then, the first incremental backup made is then named "phantom_backup_group_0_level_1.tar".

Each subsequent incremental backup in the same group increases by a level of one.

Additional full backups create a new group, and incremental backups based on that full backup start incrementing the level number.

When you begin a new group based on a new full backup, earlier groups stop incrementing. Further incremental backups belong to the new group.

You can control the number of backup groups by using the `--set-full-backup-limit` argument with `ibackup.pyc`.

You can change the number of backup groups by running the `ibackup.pyc` script with a new `--set-full-backup-limit` argument. If you set a lower limit, backup groups are deleted, starting with the oldest group.

For example, a Splunk SOAR (On-premises) deployment is configured for backup and restore using `ibackup.pyc --set-full-backup-limit 3`. After several weeks, three backup groups exist, each with a full backup and several incremental backups:

Group 0

- phantom_backup_group_0_level_0.tar
 - ◆ phantom_backup_group_0_level_1.tar
 - ◆ phantom_backup_group_0_level_2.tar
 - ◆ phantom_backup_group_0_level_3.tar

Group 1

- phantom_backup_group_1_level_0.tar
 - ◆ phantom_backup_group_1_level_1.tar
 - ◆ phantom_backup_group_1_level_2.tar
 - ◆ phantom_backup_group_1_level_3.tar

Group 2

- phantom_backup_group_2_level_0.tar
 - ◆ phantom_backup_group_2_level_1.tar
 - ◆ phantom_backup_group_2_level_2.tar
 - ◆ phantom_backup_group_2_level_3.tar

A decision is made that the oldest backups are no longer required, so an administrator runs `ibackup.py --set-full-backup-limit 3`. When the next backup runs after the new full backup limit is set, Group 0 is deleted.

What is in a full backup?

A full backup of a Splunk SOAR (On-premises) instance contains the following file sets:

File set	Files	Backup path
misc_files	Miscellaneous files used by Splunk SOAR (On-premises).	<PHANTOM_HOME>/keystore/private_key.pem <PHANTOM_HOME>/www/phantom_ui/secret_key.py <PHANTOM_HOME>/www/phantom_ui/secret_key.pyc <PHANTOM_ETC>/nginx/conf.d/default.conf <PHANTOM_HOME>/etc/cacerts.pem <PHANTOM_HOME>/splunk/etc/apps/splunk_httpinput/local/inputs.conf <PHANTOM_HOME>/etc/enable <PHANTOM_HOME>/www/phantom_ui/auth_backends/saml2_xml
apps	All Splunk SOAR (On-premises) apps, excluding app_states.	<PHANTOM_HOME>/apps
ssl	All Splunk SOAR (On-premises) etc/ssl contents.	<PHANTOM_HOME>/etc/ssl
certs	All Splunk SOAR (On-premises) certificates.	<PHANTOM_HOME>/etc/certs
playbooks	All playbooks, excluding playbook states.	<PHANTOM_HOME>/scm
nginx_keys	The NGINX SSH keys.	<PHANTOM_VAR>/cache/nginx/.ssh
vault	All vault contents, excluding files that are still streaming to storage.	<PHANTOM_HOME>/vault
app_states	All Splunk SOAR (On-premises) app states, excluding apps.	<PHANTOM_HOME>/local_data/app_states
playbook_states	All Splunk SOAR (On-premises) playbook states, excluding playbooks.	<PHANTOM_HOME>/tmp

The directory <PHANTOM_HOME> represents:

- On a privileged deployment the directory /opt/phantom.
- On an unprivileged deployment using a virtual machine image or an Amazon Marketplace Image, the directory /opt/phantom.
- On an unprivileged deployment using the installation TAR file the directory for the user account that runs Splunk SOAR (On-premises).

How an incremental backup differs from a full backup

Incremental backups contain only changes to your Splunk SOAR (On-premises) deployment since the last backup was made.

Incremental backups are based on a group of backup files that begin with a full backup, then the backup files in sequence. The new file contains changes that were made since the previous backup.

An incremental backup cannot be used to restore a system on its own. It must be used with the related full backup and any intermediate incremental backups. For example:

```
phantom_backup_group_0_level_0.tar    phantom_backup_group_0_level_1.tar
phantom_backup_group_0_level_2.tar    phantom_backup_group_0_level_3.tar
```

In this example, `phantom_backup_group_0_level_0.tar` is the full backup that forms the base of the backup group. The files `phantom_backup_group_0_level_1.tar`, `phantom_backup_group_0_level_2.tar`, and `phantom_backup_group_0_level_3.tar` are incremental backups that depend on the earlier files.

An administrator can restore a Splunk SOAR (On-premises) deployment from any point in the group as long as the earlier files in the group are present.

Supported restore configurations

This table presents possible destinations for restoring a backup.

The origin of a backup can be any supported Splunk SOAR (On-premises) deployment, such as a virtual machine image, RPM-based installation, or Amazon Marketplace Image.

- The Splunk SOAR (On-premises) deployments, the origin of the backup, and the destination for the restore must be running the same version of Splunk SOAR (On-premises).
- You cannot restore a backup from a privileged instance of Splunk SOAR (On-premises) to an unprivileged instance or from an unprivileged instance to a privileged instance.

Backup origin	Possible backup destinations
Standalone, privileged instance	<ul style="list-style-type: none">• Standalone, privileged instance• Privileged cluster• Single privileged instance, external PostgreSQL database• Single privileged instance, external file shares• Privileged instance, external PostgreSQL database and file shares
Single privileged instance, external PostgreSQL database	<ul style="list-style-type: none">• Standalone, privileged instance• Privileged cluster• Single privileged instance, external PostgreSQL database• Single privileged instance, external file shares• Privileged instance, external PostgreSQL database and file shares
Single privileged instance, external file shares	<ul style="list-style-type: none">• Standalone, privileged instance• Privileged cluster• Single privileged instance, external PostgreSQL database• Single privileged instance, external file shares

Backup origin	Possible backup destinations
	<ul style="list-style-type: none"> Privileged instance, external PostgreSQL database and file shares
Privileged instance, external PostgreSQL database and file shares	<ul style="list-style-type: none"> Standalone, privileged instance Privileged cluster Single privileged instance, external PostgreSQL database Single privileged instance, external file shares Privileged instance, external PostgreSQL database and file shares
Standalone, unprivileged instance	<ul style="list-style-type: none"> Standalone, unprivileged instance Unprivileged cluster Single unprivileged instance, external PostgreSQL database Single unprivileged instance, external file shares Unprivileged instance, external PostgreSQL database and file shares
Unprivileged instance, external PostgreSQL database	<ul style="list-style-type: none"> Standalone, unprivileged instance Unprivileged cluster Single unprivileged instance, external PostgreSQL database Single unprivileged instance, external file shares Unprivileged instance, external PostgreSQL database and file shares
Unprivileged instance, external file shares	<ul style="list-style-type: none"> Standalone, unprivileged instance Unprivileged cluster Single unprivileged instance, external PostgreSQL database Single unprivileged instance, external file shares Unprivileged instance, external PostgreSQL database and file shares
Unprivileged instance, external PostgreSQL database and file shares	<ul style="list-style-type: none"> Standalone, unprivileged instance Unprivileged cluster Single unprivileged instance, external PostgreSQL database Single unprivileged instance, external file shares Unprivileged instance, external PostgreSQL database and file shares

Back up a Splunk SOAR (On-premises) deployment

Your Splunk SOAR (On-premises) deployment is backed up using command-line tools. You do not need to take Splunk SOAR (On-premises) offline to do a backup.

Before you begin, check each of the following things.

- For privileged deployments, make sure that you have root or sudo permissions on the Splunk SOAR (On-premises) deployment.
- Make sure you have completed either the section *Prepare Splunk SOAR (On-premises) for a back up* or *Prepare a Splunk SOAR (On-premises) cluster or deployment with an external PostgreSQL database*, depending on your Splunk SOAR (On-premises) deployment's configuration.

Deployments of Splunk SOAR (On-premises) in AWS that use RDS for their PostgreSQL database should not use `ibackup` to create backups of the Splunk SOAR (On-premises) database. Use the `backup.pyc` tool instead. See [Create a full backup for deployments with an external PostgreSQL database in RDS](#).

Prepare Splunk SOAR (On-premises) for a backup

Before you can create backups, you must prepare Splunk SOAR (On-premises) by running the `ibackup` command with the `--setup` option.

You must prepare Splunk SOAR (On-premises) by running the `ibackup` command with the `--setup` option after any upgrade of Splunk SOAR (On-premises).

During setup, `ibackup` temporarily stops Splunk SOAR (On-premises) services in order to make changes to the PostgreSQL database configuration. Services restart before setup is complete. This causes the following error to be displayed during the setup process, which can be ignored:

```
psql: ERROR:  pgbouncer cannot connect to server
```

If your Splunk SOAR (On-premises) deployment uses a Warm Standby or if you intend to use a Warm Standby, you must configure Warm Standby before setting up backups.

1. From the command line, SSH to your Splunk SOAR (On-premises) instance.
`ssh <username>@<phantom_hostname>`
2. Prepare the system for a backup.
`/opt/phantom/bin/phenv ibackup --setup`

The command output looks like this:

```
[phantom@phantom bin]$ phenv ibackup --setup
Logs for this script run will be located at
/opt/phantom/var/log/phantom/ibackup/ibackup_2022-02-17-22-45-24.log
Attempting to connect to Postgresql, connection errors are expected while bootstrapping ...
Setup will temporarily stop phantom
If you wish to continue, enter yes to proceed: yes
Setup complete
Logs for this script run will be located at
/opt/phantom/var/log/phantom/ibackup/ibackup_2022-02-17-22-45-24.log
```

If the Splunk SOAR (On-premises) instance or cluster has been setup for backups, the tool prompts you to confirm that you wish to run setup again:

```
You appear to have already run setup. Re-run again ? Enter yes to proceed:
```

Enter **yes** to run the setup process again.

Running the set up again after a restore resets the state file used by `ibackup` and archives all existing backup files. Running it a second time before any restore actions does nothing.

Prepare a Splunk SOAR (On-premises) cluster or deployment with an external PostgreSQL database for a backup

For Splunk SOAR (On-premises) deployments where the PostgreSQL database is external to the Splunk SOAR (On-premises) instance or in clustered deployments, you must take additional steps to prepare the deployment for backup.

In clustered deployments, all backup commands must be issued from the same cluster node.

1. From the command line, SSH to one cluster node of your Splunk SOAR (On-premises) deployment.
`ssh <username>@<phantom_hostname>`
2. Run `ibackup`. This step generates the `db_bootstrap.tgz` file.
`phenv ibackup --setup`
3. Use SCP to copy `db_bootstrap.tgz` to your PostgreSQL database host.
`scp db_bootstrap.tgz <username>@<postgresql_hostname>:/<directory>`
4. SSH to the PostgreSQL database host.
`ssh <username>@<postgresql_hostname>`
5. Extract the `db_bootstrap.tgz` file.
`tar -xvzf db_bootstrap.tgz`
6. Change directory to the `setup` directory created when `db_bootstrap.tgz` was extracted.
`cd setup`
7. Run the `extdb_backup_bootstrap` script with the parameter `--pgdata </path/to/postgresql/db>`.
For Splunk SOAR (On-premises) version 4.8:
`python extdb_backup_bootstrap.pyc --pgdata </path/to/postgresql/db>`
For Splunk SOAR (On-premises) version 4.9 and later:
`python extdb_backup_bootstrap.pyc --pgdata </path/to/postgresql/db>`
8. SSH to the cluster node where you ran `ibackup`.
`ssh <username>@<phantom_hostname>`
9. Run `ibackup` to complete the setup.
`phenv ibackup --setup`

Create a full backup

Your Splunk SOAR (On-premises) instance is backed up using command-line tools. You do not need to take Splunk SOAR (On-premises) offline to create a backup.

In clustered deployments, all backup commands must be issued from the same cluster node.

1. From the command line, SSH to your Splunk SOAR (On-premises) instance.
`ssh <username>@<phantom_hostname>`
2. Run the following command to perform the backup.
`phenv ibackup --backup --backup-type full`
If no backups exist, a full backup is created. When a full backup exists, a new incremental backup is added to the group. If you already have a backup group and want to create a new full backup, use the `--backup-type full` argument.

The following sample output shows a backup being run:

```
[11/Dec/2019 21:24:28] INFO: Running ibackup - details will be logged to
/var/log/phantom/backup/backup_2019-12-11T21:24:28.706665Z.log
[11/Dec/2019 21:24:28] INFO: Attempting to connect to Postgresql ...
[11/Dec/2019 21:24:30] INFO: First backup. Performing full backup
[11/Dec/2019 21:24:30] INFO: Backing up files
[11/Dec/2019 21:25:03] INFO: Backing up database
[11/Dec/2019 21:25:17] WARNING: no prior backup exists, incr backup has been changed to full

[11/Dec/2019 21:25:58] INFO: Backup created at: /opt/phantom/data/backup/phantom_backup_group_0_level_0.tar
[11/Dec/2019 21:25:58] INFO: You should ensure this tarball is kept safe. It will be required for restore
```

If you receive an error that postgresql.conf is owned by the root user, you will need to use the `chown` and `chgrp` commands to set the ownership of `<PHANTOM_HOME>/phantom/data/db/postgresql.conf` to the postgres user. For clustered deployments, do this on the database node of the deployment.

Create a full backup for deployments with an external PostgreSQL database in RDS

Amazon Web Services RDS provides automatic backups of hosted PostgreSQL databases which are managed and restored using the management console. See [Backing up and restoring an Amazon RDS DB instance in the AWS documentation](#).

To back up other Splunk SOAR (On-premises) components, use the older `backup.pyc` tool.

1. From the command line, SSH to your Splunk SOAR (On-premises) instance.
`ssh <username>@<phantom_hostname>`
2. Change the directory to `<PHANTOM_HOME>/bin`.
`cd <PHANTOM_HOME>/bin`
3. Perform the backup.
`/opt/phantom/bin/phenv python backup.pyc --all`
 In this case, you can safely ignore the deprecation warning.

The command output looks like this: `[root@phantom bin]# phenv python backup.pyc --all`

```
[pid: 8548] [09/Sep/2020 22:49:17] backup.py:609 WARNING: The --all option of the backup.pyc script has
been deprecated. Please use the ibackup.pyc script to perform backups and restores. Documentation for the
new script can be found at
https://docs.splunk.com/Documentation/SOARonprem/5.0.1/Admin/BackupOrRestoreOverview. The --config option of
the backup.pyc script will continue to work with ibackup.pyc.
```

```
[2020-09-09 22:49:17] The --all option of the backup.pyc script has been deprecated. Please use the
ibackup.pyc script to perform backups and restores. Documentation for the new script can be found at
https://docs.splunk.com/Documentation/SOARonprem/5.0.1/Admin/BackupOrRestoreOverview. The --config option of
the backup.pyc script will continue to work with ibackup.pyc.
```

```
backup.pyc 2.0.0
[2020-09-09 22:49:17] Stopping all Phantom services except PostgreSQL
Stopping Supervisor daemon manager...[ OK ]
/opt/phantom/proxy/bin/splunk_proxyd is already stopped
Stopping phantom_actiond: [ OK ]
Stopping phantom_workflowd: [ OK ]
Stopping phantom_ingestd: [ OK ]
Stopping phantom_decided: [ OK ]
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
[ OK ]
```

```

Stopping splunk helpers...
[ OK ]
Done.
[2020-09-09 22:49:28] Backing up Phantom DB
[=====] 100%
[2020-09-09 22:49:29] Generating CSVs for configuration tables
[=====] 100%
[2020-09-09 22:49:31] Validating CSVs
[2020-09-09 22:49:31] Backing up required Phantom directories
[=====] 100%
[2020-09-09 22:50:37] Backing up Phantom-specific files
[=====] 100%
[2020-09-09 22:50:37] Compressing backup (this may take a while)
[2020-09-09 22:50:47] Running /opt/phantom/bin/start_phantom.sh
Starting all Phantom services
Phantom startup successful
logs recorded in /opt/phantom/data/phantom_backups/phantom_backup_2020-09-09-22-49-17.log
Backup located at /opt/phantom/data/phantom_backups/phantom_backup_2020-09-09-22-49-17.tgz

```

Create an incremental backup

The `ibackup` tool checks whether a full backup exists. If a full backup doesn't exist, a full backup is created. If a full backup exists, an incremental backup is created and added to the backup chain. You do not need to take Splunk SOAR (On-premises) offline to create a backup. If your deployment's PostgreSQL database is hosted in Amazon's RDS, you cannot use incremental backups for the database.

In clustered deployments, all backup commands must be issued from the same cluster node.

1. From the command line, SSH to your Splunk SOAR (On-premises) instance.
`ssh <username>@<phantom_hostname>`
2. Change the directory to `<PHANTOM_HOME>/bin`.
`cd <PHANTOM_HOME>/bin`
3. Perform the backup.
`phenv ibackup --backup`

The command output looks like this:

```

[11/Dec/2019 21:29:47] INFO: Running ibackup - details will be logged to
/var/log/phantom/backup/backup_2019-12-11T21:29:47.549233Z.log
[11/Dec/2019 21:29:47] INFO: Attempting to connect to Postgresql ...
[11/Dec/2019 21:29:53] INFO: Backing up files
[11/Dec/2019 21:29:54] INFO: Backing up database
[11/Dec/2019 21:30:03] INFO: Backup created at: /opt/phantom/data/backup/phantom_backup_group_0_level_1.tar
[11/Dec/2019 21:30:03] INFO: You should ensure this tarball is kept safe. It will be required for restore

```

You can override the default behavior by using this command-line option:

```
phenv ibackup --backup --backup-type incr
```

Save your backups in a safe place, such as one that is not on the same disk, partition, or virtual machine as your Splunk SOAR (On-premises) instance.

Restore Splunk SOAR (On-premises) from a backup

Restoring a backup requires root permissions.

You must restore a backup to an instance with the same privilege level. You can restore a backup from a privileged instance of Splunk SOAR (On-premises) to another privileged instance, or from an unprivileged instance to another unprivileged instance. You *cannot* restore a backup from a privileged instance to an unprivileged instance or from an unprivileged instance to a privileged instance.

You can use backups in conjunction with the Splunk SOAR (On-premises) Warm Standby feature for additional protection against system failure.

In clustered deployments, you must issue all backup and restore commands from the same cluster node.

Prepare your system for restore

Before you can perform a restore in your Splunk SOAR (On-premises) deployment, you must prepare your system. This preparation is especially important if you are restoring data from one Splunk SOAR (On-premises) deployment to another deployment.

You don't need to perform these steps when restoring a backup to the same deployment; backup creation includes the setup step.

To prepare your deployment before restoring, perform the following steps:

1. From the command line, SSH to your Splunk SOAR (On-premises) instance or Splunk SOAR (On-premises) cluster node.
SSH <username>@<phantom_hostname>
2. Change the directory to <PHANTOM_HOME>/bin.
cd <PHANTOM_HOME>/bin
3. Prepare the system for a restore.
sudo phenv python ibackup.pyc --setup

Restore your deployment from a full backup

To restore your deployment from a full backup, follow these steps:

1. From the command line, SSH to your Splunk SOAR (On-premises) instance or Splunk SOAR (On-premises) cluster node.
SSH <username>@<phantom_hostname>
2. Change the directory to <PHANTOM_HOME>/bin.
cd <PHANTOM_HOME>/bin
3. Prepare the system for a restore.
sudo phenv python ibackup.pyc --setup
4. Copy your <number>_phantom_backup.tar from storage to the instance or cluster node you are restoring.
5. Perform the restore. See note below.
sudo phenv python ibackup.pyc --restore <path/to/<number>_phantom_backup.tar>

For deployments of Splunk SOAR (On-premises) in AWS that use RDS for their PostgreSQL database: Do not use `ibackup.pyc`. Create backups using the `backup.pyc` tool and perform restores using the `restore.pyc` tool, as described in the next section.

Restore a full backup for deployments with an external PostgreSQL database in RDS

If your deployment uses Amazon Web Services (AWS) RDS to host Splunk SOAR (On-premises)'s PostgreSQL database, you cannot use `ibackup.pyc` to back up or restore the database. Instead, use a combination of the automatic backups in RDS and the older `backup.pyc` and `restore.pyc` tools. For information on using the back up and restore features of RDS, see [Backing up and restoring an Amazon RDS DB instance in the AWS documentation](#). To perform a restore using the `restore.pyc` tool, follow these steps:

1. From the command line, SSH to your Splunk SOAR (On-premises) instance.
SSH <username>@<phantom_hostname>
2. Change the directory to <PHANTOM_HOME>/bin.
cd <PHANTOM_HOME>/bin
3. Perform the backup.
sudo phenv python restore.pyc --file <PATH/TO/BACKUP/FILE>
For this use, you can safely ignore the deprecation warning.

The command output looks like this: [root@phantom bin]# phenv python restore.pyc --file /opt/phantom/data/phantom_backups/phantom_backup_2020-09-09-22-49-17.tgz
[pid: 10562] [09/Sep/2020 23:03:06] restore.py:692 WARNING: The --all option of the backup.pyc script has been deprecated. Please use the ibackup.pyc script to perform backups and restores. Documentation for the new script can be found at <https://docs.splunk.com/Documentation/SOARonprem/5.0.1/Admin/BackupOrRestoreOverview>. The --config option of the backup.pyc script will continue to work with ibackup.pyc.

[2020-09-09 23:03:06] The --all option of the backup.pyc script has been deprecated. Please use the ibackup.pyc script to perform backups and restores. Documentation for the new script can be found at <https://docs.splunk.com/Documentation/SOARonprem/5.0.1/Admin/BackupOrRestoreOverview>. The --config option of the backup.pyc script will continue to work with ibackup.pyc.

```
restore.pyc 2.0.0
[2020-09-09 23:03:06] Stopping all Phantom services except PostgreSQL
Stopping Supervisor daemon manager...[ OK ]
/opt/phantom/proxy/bin/splunk_proxycd is already stopped
Stopping phantom_actiond: [ OK ]
Stopping phantom_workflowd: [ OK ]
Stopping phantom_ingestd: [ OK ]
Stopping phantom_decided: [ OK ]
[2020-09-09 23:03:12] Extracting backup tarball (this may take a while)
The backup file appears to be a full backup. This will overwrite any existing data upon restore. Proceed?
[y/N]y
[2020-09-09 23:03:21] Loading tables from backup into database
[=====] 100%
[2020-09-09 23:03:24] Deleting existing Phantom database records
[=====] 100%
[2020-09-09 23:03:31] Inserting backup rows into main tables
[=====] 100%
[2020-09-09 23:03:31] Dropping temporary archive tables used for backup
[=====] 100%
[2020-09-09 23:03:32] Finalizing transaction
[2020-09-09 23:03:32] Updating the main menu
[2020-09-09 23:03:32] Restoring specific Phantom file backups
[=====] 100%
```



```
[2020-09-09 23:03:32] Configuring NGINX SSL certificates (this may take a while)
[2020-09-09 23:03:44] Restoring Phantom subdirectories
[=====] 100%
[2020-09-09 23:04:05] Running /opt/phantom/bin/start_phantom.sh
Starting all Phantom services
Phantom startup successful
[2020-09-09 23:04:09] Resetting passwords for Splunk users
[2020-09-09 23:04:17] Done resetting Splunk user passwords
Stopping Supervisor daemon manager...[ OK ]
Starting Supervisor daemon manager...[ OK ]
Logs recorded in /opt/phantom/data/phantom_backups/phantom_backup_restore_2020-09-09-23-03-06.log
```

Restore your system from an incremental backup

You must prepare the system before restoring your system from an incremental backup. See [Prepare your system for restore](#) earlier in this topic.

Incremental backups contain only the changes made to your Splunk SOAR (On-premises) instance since the last full backup or previous incremental backup. An incremental backup is not sufficient to restore a system on its own. It must be used with the related full backup and any intermediate backups.

Here is a sample sequence of restoring your system from an incremental backup. The sequence is important, but there can be varying increments of time between the steps.

1. Create a full backup called `phantom_backup_group_0_level_0.tar`.
2. Create an incremental backup called `phantom_backup_group_0_level_1.tar`, which is based on `phantom_backup_group_0_level_0.tar`.
3. Create a second incremental backup called `phantom_backup_group_0_level_2.tar`, which is based on `phantom_backup_group_0_level_1.tar` and `phantom_backup_group_0_level_0.tar`.

Remember these important points when restoring your system from the sequential files:

- You can restore `phantom_backup_group_0_level_0.tar` alone.
- You cannot restore `phantom_backup_group_0_level_1.tar` without `phantom_backup_group_0_level_0.tar`.
- You cannot restore `phantom_backup_group_0_level_2.tar` without `phantom_backup_group_0_level_0.tar` and `phantom_backup_group_0_level_1.tar`.

Restore the incremental backup

To restore the incremental backup, follow these steps:

1. From the command line, SSH to your Splunk SOAR (On-premises) instance or cluster node.
SSH <username>@<phantom_hostname>
2. Change the directory to <PHANTOM_HOME>/bin.
cd <PHANTOM_HOME>/bin
3. Prepare the system for a restore.
sudo phenv python ibackup.pyc --setup
4. Copy the full backup TAR file and any incremental-level TAR files from storage to the instance or cluster node you are restoring.
5. Perform the restore. Enter the file name of the last incremental backup file you want to restore.
sudo phenv python ibackup.pyc --restore < phantom_backup_group_<#>_level_<#>.tar >

Determine whether the system restore was successful

If the restore is successful, it writes information to the console. Here is an example of console output from a successful restore:

```
[root@phantom bin]# phenv python ibackup.pyc --restore
/opt/phantom/data/backup/phantom_backup_group_0_level_0.tar
[06/Feb/2020 20:10:15] INFO: Running ibackup.pyc - details will be logged to
/var/log/phantom/backup/ibackup_2020-02-06T20:10:15.089127Z.log
[06/Feb/2020 20:10:15] INFO: Attempting to connect to Postgresql ...
[06/Feb/2020 20:10:17] INFO: Checking filesystem backup state at /opt/phantom/data/ibackup/repo/fs
[06/Feb/2020 20:10:17] INFO: Restoring this backup requires utilizing 9.11334507138% of the total volume
capacity
[06/Feb/2020 20:10:17] INFO: Available: 45901836288 , Required: 2008317952.0
[06/Feb/2020 20:10:21] INFO: Attempting to connect to Postgresql ...
psql: ERROR: pgbouncer cannot connect to server
[06/Feb/2020 20:10:21] INFO: Retrying ...
[06/Feb/2020 20:10:22] INFO: Attempting to connect to Postgresql ...
psql: ERROR: pgbouncer cannot connect to server
[06/Feb/2020 20:10:22] INFO: Retrying ...
[06/Feb/2020 20:10:24] INFO: Attempting to connect to Postgresql ...
psql: ERROR: pgbouncer cannot connect to server
[06/Feb/2020 20:10:24] INFO: Retrying ...
[06/Feb/2020 20:10:28] INFO: Attempting to connect to Postgresql ...
psql: ERROR: pgbouncer cannot connect to server
[06/Feb/2020 20:10:28] INFO: Retrying ...
[06/Feb/2020 20:10:36] INFO: Attempting to connect to Postgresql ...
[06/Feb/2020 20:10:38] INFO: Extracting backup file
/opt/phantom/data/backup/phantom_backup_group_0_level_0.tar
[06/Feb/2020 20:11:08] INFO: Restoring files to filesystem
[06/Feb/2020 20:11:17] INFO: Attempting to connect to Postgresql ...
[06/Feb/2020 20:11:27] INFO: Restore complete
```

Prepare for subsequent backups

After restoring your system, you must run `sudo phenv python ibackup.pyc --setup` again before you can make new backups. See [Prepare your system for restore](#) earlier in this topic.

Splunk SOAR (On-premises) backup tools

Use the `ibackup.pyc` tool to create, manage, and restore backups.

Logs for each run of the tool are written to `/var/log/phantom/backup/backup.log`.

Completed backups are stored in `<PHANTOM_HOME>/phantom/data/backup`.

If you are using an unprivileged installation, the logs are written to `<PHANTOM_HOME>/var/log/phantom/backup/backup.log`.

You can find a repository of staging files for the PostgreSQL database backup in `<PHANTOM_HOME>/data/ibackup/repo/pg`.

ibackup.pyc arguments

The following table shows the ibackup.pyc arguments:

Argument	Description
-h, --help	Shows the ibackup.pyc tool help message and exits.
--setup	Prepares the instance or cluster for backup and restore.
--max-cores <value>	Specifies the maximum number of processing cores allowed for database backup and restore operations. The default value is two cores.
--backup	Performs a backup.
--ignore-size-check	Use this argument to skip the check for available disk space before performing a backup or restore. <ul style="list-style-type: none">• If you don't specify this argument and ibackup does not detect enough free space, you are prompted to either continue or to cancel the backup or restore operation.• Use this argument for unattended backup operations.
--restore <path/to/backup/>	Performs a restore. You must provide a path to the the last backup tar file to perform a restore.
--set-pgbackrest-repo <path to repository>	Sets the path of the pgbackrest repository.
--backup-components	Selectively backs up specific Splunk SOAR (On-premises) components. The default is all components. You must specify the same components for <code>--restore-components</code> when you restore using a backup created this way. See <code>--restore-components</code> for a complete list. For example: <code>--backup-components db,playbooks,keys</code>
--config-only	Backups include only configuration data. This always creates a full backup of configuration data. Incremental backup of configuration data is not supported. <div>Using the <code>--config-only</code> argument requires Splunk SOAR (On-premises) to shutdown in order to create the configuration backup.</div>

`--restore-components <components>`Selectively restores specific Splunk SOAR (On-premises) components. The default is all components.

The following components are valid components:

- db: the PostgreSQL database
- configuration: the Splunk SOAR (On-premises) instance or cluster configuration information
- apps: The apps installed for Splunk SOAR (On-premises)
- app_states: The state of each app at the time of the backup
- playbooks: the current playbooks in the scm
- playbooks_states: the current state of each playbook at the time of the backup
- vault: the Splunk SOAR (On-premises) vault

For example: `--restore-components db,playbooks,keys`

`--list-backups`Lists existing backups and their state. Use with `--verbose` for more detailed output.`--delete-all`Deletes all backups.

This action is irreversible.

`--delete-backup-group <group number>`Deletes a full backup group. Takes an integer that represents the backup group to delete.`--version`Shows the ibackup.pyc tool version number and exit.`--backup-path <path/to/store/backups>`Overrides the default backup path `<PHANTOM_HOME>/phantom/data/backup`. Takes a directory path for the directory where backups will be stored.`--backup-type`

<full,incr>Backup type. Using "full" creates a new full backup. Using "incr" creates an incremental on top of the current full backup.

If no full backup is taken and "incr" is given, the backup type defaults to "full". The default option if none is specified is "incr".

--set-full-backup-limit <value>Sets the maximum number of full backups allowed at once. Automatically rotates once the limit is reached.--verboseWrites debug-level log information to the console.--list-settingsLists the current settings for ibackup.--no-promptAutomatically responds with "yes" to all prompts from ibackup.**The following option has been removed.**--force-pg-stop-backupRuns pg_stop_backup against the current PostgreSQL database.

Use ibackup.pyc with warm standby

The warm standby and the backup and restore features require careful planning to use together.

Warm standby and ibackup features of Splunk SOAR (On-premises) use the Write Ahead Logging feature in PostgreSQL. When you configure a Splunk SOAR (On-premises) deployment to use both warm standby and ibackup, you must configure warm standby first. After restoring a deployment with ibackup, you must update the warm standby configuration.

Configuring warm standby after configuring ibackup archives all existing backups. Archiving all of the backups prevents new backups from being generated or existing backups from being used in a restore. You can generate new backups once you run ibackup with the --setup option.

Restore a system configured for warm standby

In a warm standby configuration, when the primary Splunk SOAR (On-premises) instance is restored from a backup, you must update the warm standby configuration.

Prerequisites

You need the following information to update your warm standby configuration:

- Password for the Splunk SOAR (On-premises) user on the secondary Splunk SOAR (On-premises) instance. If the Splunk SOAR (On-premises) user does not have a password, you must set one.
- Password for the PostgreSQL database replication user.
- Configuration information for creating the SSL certificate:
 - ◆ Country code
 - ◆ State code
 - ◆ Organization
 - ◆ Organization unit
 - ◆ Domain
 - ◆ Email

Restore a backup from a warm standby primary to the same Splunk SOAR (On-premises) instance

When you restore a backup of a Splunk SOAR (On-premises) warm standby primary to the same instance, the warm standby configuration must be updated.

To update the warm standby configuration, perform the following steps:

1. Open a terminal session for both the primary and secondary Splunk SOAR (On-premises) instances. Keep these sessions open until you complete these steps.
 1. From the command line, SSH to your primary Splunk SOAR (On-premises) instance.

- SSH <username>@<primary_phantom_hostname>
2. SSH to your secondary and warm standby Splunk SOAR (On-premises) instance.
SSH <username>@<warm_standby_phantom_hostname>
3. In both sessions, elevate to root.
sudo su -
2. On the primary instance of Splunk SOAR (On-premises), perform the restore. See [Restore Splunk SOAR \(On-premises\) from a backup](#).
3. On the primary instance of Splunk SOAR (On-premises), disable warm standby.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --off
4. On the secondary instance of Splunk SOAR (On-premises), disable warm standby.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --off
5. On the secondary instance of Splunk SOAR (On-premises), stop all Splunk SOAR (On-premises) services.
/<PHANTOM_HOME>/bin/stop_phantom.sh

Failing to stop these services on the secondary instance results in two active instances operating independently, polling for data and executing automated actions. This can result in data loss or other undesired results.

6. On the primary instance of Splunk SOAR (On-premises), configure it to be the primary instance for warm standby. You are prompted to give passwords for the Splunk SOAR (On-premises) user, the PostgreSQL database replication user, and the information for creating a self-signed SSL certificate.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>
7. On the secondary instance, configure it to be the warm standby instance.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>
8. On the both instances of Splunk SOAR (On-premises), verify that warm standby is replicating on each Splunk SOAR (On-premises) instance.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --status

Example output from Splunk SOAR (On-premises) primary:

```
===== Processed Params =====
Instance looks like Primary
DB replication configured with Standby set to: <warm_standby_ip>/32
DB replication currently streaming
Vault sync configured
===== Script Done =====
```

Example output from Splunk SOAR (On-premises) secondary or warm standby:

```
===== Processed Params =====
Instance looks like Standby
DB replication configured
rsync configured
===== Script Done =====
```

Restore a backup from a warm standby primary to a new Splunk SOAR (On-premises) instance

When you restore a backup of a Splunk SOAR (On-premises) warm standby primary to a new instance that you want to become the new primary, you must update the warm standby configuration and move several keys to the secondary instance.

To update the warm standby configuration, perform the following steps:

1. Open a terminal session for both the primary and secondary Splunk SOAR (On-premises) instances. Keep these sessions open until you complete these steps.

1. From the command line, SSH to your primary Splunk SOAR (On-premises) instance.

SSH <username>@<primary_phantom_hostname>

2. SSH to your secondary and warm standby Splunk SOAR (On-premises) instance.

SSH <username>@<warm_standby_phantom_hostname>

3. In both sessions, elevate to root.

sudo su -

2. On the primary instance of Splunk SOAR (On-premises), perform the restore. See [Restore Splunk SOAR \(On-premises\) from a backup](#).

3. On the primary instance of Splunk SOAR (On-premises), disable warm standby.

phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --off

4. On the secondary instance of Splunk SOAR (On-premises), disable warm standby.

phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --off

5. On the secondary instance of Splunk SOAR (On-premises), stop all Splunk SOAR (On-premises) services.

/<PHANTOM_HOME>/bin/stop_phantom.sh

Failing to stop these services on the secondary instance results in two active instances operating independently, polling for data and executing automated actions. This can result in data loss or other undesired results.

6. Copy these files from the new primary instance of Splunk SOAR (On-premises) to the secondary:

1. /<PHANTOM_HOME>/keystore/private_key.pem

2. /<PHANTOM_HOME>/www/phantom_ui/secret_key.py

7. On the secondary instance of Splunk SOAR (On-premises), set the permissions, ownership, and SELinux security contexts for the files you copied to the secondary.

1. chmod 0640 /<PHANTOM_HOME>/keystore/private_key.pem

/<PHANTOM_HOME>/phantom/www/phantom_ui/secret_key.py

2. chown root:phantom /<PHANTOM_HOME>/keystore/private_key.pem

3. chown phantom:phantom /<PHANTOM_HOME>/www/phantom_ui/secret_key.py

4. restorecon /<PHANTOM_HOME>/keystore/private_key.pem

/<PHANTOM_HOME>/www/phantom_ui/secret_key.py

8. On the primary instance of Splunk SOAR (On-premises), configure it to be the primary for warm standby. You are prompted to give passwords for the Splunk SOAR (On-premises) user, the PostgreSQL database replication user, and the information for creating a self-signed SSL certificate.

phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --configure --primary-ip

<primary_ip> --standby-ip <standby_ip>

9. On the secondary instance, configure it to be the warm standby instance.

phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --configure --primary-ip

<primary_ip> --standby-ip <standby_ip>

10. On both instances of Splunk SOAR (On-premises), verify that the warm standby instance is replicating on each Splunk SOAR (On-premises) instance.

phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --status

Example output from Splunk SOAR (On-premises) primary:

```
===== Processed Params =====
```

```
Instance looks like Primary
```

```
DB replication configured with Standby set to: <warm_standby_ip>/32
```

```
DB replication currently streaming
```

```
Vault sync configured
```

```
===== Script Done =====
```

Example output from Splunk SOAR (On-premises) secondary or warm standby:

```
===== Processed Params =====  
Instance looks like Standby  
DB replication configured  
rsync configured  
===== Script Done =====
```

Create and manage a warm standby

Warm standby feature overview

Warm standby is a strategy for high availability that regularly copies data from a primary instance of Splunk SOAR (On-premises) to a secondary instance. In the event of a failure on the primary, a systems administrator can quickly put the secondary into service as a new primary with minimal downtime or data loss.

Splunk SOAR (On-premises)'s warm standby is implemented using PostgreSQL's streaming replication for the internal database and cron-based rsync of file system directories.

Warm standby is not a substitute for regular backups or other disaster recovery preparations.

Warm standby is configured in the same way, and works the same way on both privileged and unprivileged deployments.

Supported configurations

Splunk SOAR (On-premises) systems administrators can configure two identical, standalone Splunk SOAR (On-premises) instances to use the warm standby strategy.

- Only individual, standalone Splunk SOAR (On-premises) instances can be configured to use warm standby.
- Splunk SOAR (On-premises) instances with an external PostgreSQL database cannot use warm standby.
- Splunk SOAR (On-premises) clusters cannot use warm standby.

If your deployment uses an external PostgreSQL 12.9 database, you cannot use the warm standby feature.

How to check the status of warm standby

You may need to check the status of warm standby before performing other actions, such as maintenance, upgrades, or troubleshooting. Splunk SOAR (On-premises) provides a command for checking the current status of warm standby.

To check the status of warm standby use the following command at the shell prompt.

Privileged instance

```
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --status
```

Unprivileged instance

```
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --status
```

The output will tell you which state your warm standby configuration is in. It will also tell you when the last WAL segment came in, and how many WAL segments are being held by the standby.

Warm standby is not configured on the instance

```
===== Processed Params =====
Warm Standby not yet configured
===== Script Done =====
```


Warm standby is configured as the primary and operating normally on the instance

```
===== Processed Params =====
Current Phantom Version: <phantom_version>
Instance looks like Primary
DB replication configured with Standby set to: <warm_standby_ip>/32
DB replication currently streaming
WAL segments configured: 32
Vault sync configured
===== Script Done =====
```

Warm standby is configured as the secondary and operating normally on the instance

```
===== Processed Params =====
Current Phantom Version: <phantom_version>
Instance looks like Standby
DB replication configured
DB last updated Wednesday <timestamp_last_updated>
WAL segments in use: 2
WAL segments configured: 32
rsync configured
===== Script Done =====
```

IP addresses

Communication between the primary Splunk SOAR (On-premises) instance and the warm standby instance is handled using IP addresses, not hostnames. If the IP address changes for either instance, you must reestablish the warm standby pairing. See [Create a warm standby](#).

PostgreSQL streaming replication

PostgreSQL streams data from its Write Ahead Log (WAL) from the primary database to the secondary database. This keeps the secondary synchronized with the primary database. See Streaming Replication on the PostgreSQL documentation site.

If network communication between the primary and secondary databases fails for any significant period of time, the WAL segments needed to update the secondary may have already been recycled by the primary. If this happens, you will need to recreate the warm standby relationship with the primary.

If your Splunk SOAR (On-premises) instances process high volumes of events or there is significant latency between the primary and the warm standby instance of Splunk SOAR (On-premises), consider increasing the value of `wal_keep_segments` in `pg_hba.conf` to a value large enough to prevent the primary from recycling WAL segments too quickly for your environment. This will use additional disk space on the primary Splunk SOAR (On-premises) instance, but can increase the reliability of the database synchronization.

Rsync

Rsync is used to synchronize file system data between the primary Splunk SOAR (On-premises) instance and the warm standby. When the `setup_warm_standby.pyc` script is run to set up warm standby, a cron job is created to run rsync to keep the warm standby up to date with the primary.

What is synchronized between the primary and the warm standby

File system directories synced from the primary to the warm standby.

Item to be synced	Directory or notes
PostgreSQL database	Synchronized using PostgreSQL streaming replication.
Files or Vault	/<PHANTOM_HOME>/vault
Certificates and system files	/<PHANTOM_HOME>/etc
Playbooks	/<PHANTOM_HOME>/scm/git/
App states	/<PHANTOM_HOME>/local_data/app_states
Playbook data state	/<PHANTOM_HOME>/tmp
SSL keys and certificates	/<PHANTOM_HOME>/etc/ssl
	/<PHANTOM_HOME>/etc/ssl/certs
Reports	/<PHANTOM_HOME>/vault/reports
SAML configuration	/<PHANTOM_HOME>/www/phantom_ui/auth_backends/saml2_xml

PIP and RPM packages are also synchronized. You can disable syncing PIP and RPM packages by using the `--ignore-package-updates` option with `setup_warm_standby.pyc`.

Relevant log files Each of these log files can be useful when troubleshooting or maintaining your warm standby configuration.

Log file name	Path
Setup and command output	
warm_standby.log	/var/log/phantom/warm_standby.log
RSYNC state logs	
rsync_opt.log	/var/log/phantom/rsync_opt.log
rsync_vault.log	/var/log/phantom/rsync_vault.log
rsync_apps.log	/var/log/phantom/rsync_apps.log
rsync_app_states.log	/var/log/phantom/rsync_app_states.log
rsync_etc_ssl_certs	/var/log/phantom/rsync_etc_ssl_certs
rsync_playbooks.log	/var/log/phantom/rsync_playbooks.log
rsync_reports.log	/var/log/phantom/rsync_reports.log
rsync_pip_req.log	/var/log/phantom/rsync_pip_req.log
rsync_rpm_req.log	/var/log/phantom/rsync_rpm_req.log
Package installation logs	
rpm_packages_primary.txt	/var/log/phantom/rpm_packages_primary.txt
pip_packages_primary.txt	/var/log/phantom/pip_packages_primary.txt

Log file name	Path

Create a warm standby

You will need two identical instances of Splunk SOAR (On-premises), one to serve as your primary Splunk SOAR (On-premises) instance, and the second to serve as the warm standby.

Do these steps to create your warm standby.

1. Complete the prerequisites.
2. Create a second Splunk SOAR (On-premises) instance to be the warm standby.
3. Setup SSH access between the primary Splunk SOAR (On-premises) instance and the new warm standby.
4. Configure warm standby using the `setup_warm_standby.pyc` script.

Creating a warm standby will restart Splunk SOAR (On-premises). You should schedule setting up warm standby for a change window or other scheduled downtime.

Prerequisites

There are some tasks that need to be completed before you can set up warm standby.

1. Create a full backup or a virtual machine snapshot of the Splunk SOAR (On-premises) instance that will be your primary.
2. Create a DNS A record for a hostname for your Splunk SOAR (On-premises) instance. You may need to work with other teams who manage DNS to accomplish this. Establish an appropriate Time To Live (TTL) value for this record since you will update the DNS A record in the event of a failover.
3. Set the **Base URL for Splunk SOAR (On-premises) Appliance** with the the hostname from the DNS A record in **Main Menu > Administration > Company Settings**. Example: `https://phantom.example.com`
4. Open the following ports on the primary Splunk SOAR (On-premises) instance's firewall TCP 22 for SSH, TCP 443 (HTTPS), and TCP 5432 for PostgreSQL operations.
5. Set up SSH between the primary Splunk SOAR (On-premises) instance and the warm standby.

Create a second Splunk SOAR (On-premises) instance to be the warm standby

You can either:

- clone the virtual machine that is your primary Splunk SOAR (On-premises) instance, or
- create an entirely new instance of Splunk SOAR (On-premises) to serve as the warm standby.

Create a Clone of your primary Splunk SOAR (On-premises) instance

You can create a clone of your primary Splunk SOAR (On-premises) instance. This clone will serve as the warm standby.

Consult the documentation for your virtualization software or the operating system software for how to clone and deploy the cloned instance of Splunk SOAR (On-premises).

Your clone will need to have its own IP and MAC addresses.

Before you clone the Splunk SOAR (On-premises) instance check to see if it is already being used as part of a warm standby pair. If the instance is part of a warm standby pairing, warm standby must be disabled before cloning the instance. See [Disable warm standby](#).

1. Clone your Splunk SOAR (On-premises) instance as described by your virtualization or operating system documentation.
2. Change the MAC and IP addresses for the new clone copy of Splunk SOAR (On-premises).
3. On the clone copy and primary instance of Splunk SOAR (On-premises), set a password for the Splunk SOAR (On-premises) user account. This password will be used later during configuration.
`passwd phantom`
4. On the clone of Splunk SOAR (On-premises), disable cron to prevent any jobs from making changes during setup and configuration.
`sudo systemctl stop crond.service`
5. On the clone of Splunk SOAR (On-premises), make sure that the port used for PostgreSQL 5432 is allowed through your firewalls.
 1. Check your firewall rules.
`firewall-cmd --list-all`
 2. (Conditional) If the port 5432 is not permitted through the firewall, add an entry to the firewall rules for it.
`firewall-cmd --zone=public --add-port=5432/tcp`
 3. (Conditional) If you needed to add port 5432 to your firewalld configuration, make the entry from the previous step permanent.
`firewall-cmd --zone=public --add-port=5432/tcp --permanent`

Create a new Splunk SOAR (On-premises) instance

If using a clone of your primary Splunk SOAR (On-premises) instance is not feasible or is otherwise unwanted, you can install a new instance of Splunk SOAR (On-premises) to serve as your warm standby.

Do these steps as either the root user or a user with sudo access.

1. Install Splunk SOAR (On-premises). See [How can Splunk SOAR \(On-premises\) be installed?](#) in *Install and Upgrade Splunk SOAR (On-premises)*.
2. SSH to your warm standby Splunk SOAR (On-premises) instance.
`ssh <username>@<warm_standby_phantom_hostname>`
3. Stop Splunk SOAR (On-premises) services on the standby.
`sudo /<PHANTOM_HOME>/bin/stop_phantom.sh`
4. Copy these files from the primary instance of Splunk SOAR (On-premises) to the new warm standby instance.
 1. `/<PHANTOM_HOME>/keystore/private_key.pem`
 2. `/<PHANTOM_HOME>/www/phantom_ui/secret_key.py`
5. On the warm standby instance of Splunk SOAR (On-premises), set the permissions, ownership, and SELinux security contexts for the files you copied to it.
 1. `chmod 0640 /<PHANTOM_HOME>/keystore/private_key.pem`
`/<PHANTOM_HOME>/www/phantom_ui/secret_key.py`
 2. `chown phantom:phantom /<PHANTOM_HOME>/keystore/private_key.pem`
 3. `chown phantom:phantom /<PHANTOM_HOME>/www/phantom_ui/secret_key.py`
 4. `restorecon /<PHANTOM_HOME>/keystore/private_key.pem`
`/<PHANTOM_HOME>/www/phantom_ui/secret_key.py`
6. On both the new warm standby instance and the primary instance of Splunk SOAR (On-premises), set a password for the phantom user account. This password will be used later during configuration.
`passwd phantom`

7. On both the new warm standby instance and the primary instance of Splunk SOAR (On-premises), make sure that the port used for PostgreSQL 5432 is allowed through your firewalls.
 1. Check your firewall rules.
`firewall-cmd --list-all`
 2. (Conditional) If the port 5432 is not permitted through the firewall, add an entry to the firewall rules for it.
`firewall-cmd --zone=public --add-port=5432/tcp`
8. On the new warm standby instance of Splunk SOAR (On-premises), disable cron to prevent any jobs from making changes during setup and configuration.
`sudo systemctl stop crond.service`

If you have installed and configured CyberArk AIM on your primary, you will need to install and configure CyberArk AIM on your warm standby.

Setup SSH between the primary and the new warm standby

During setup the primary instance of Splunk SOAR (On-premises) will need to connect to the warm standby instance of Splunk SOAR (On-premises) using SSH.

If password authentication is disabled, it must be enabled in order to proceed and can be disabled once set up is complete.

Configure warm standby using the `setup_warm_standby.pyc` script

Once both your primary and warm standby instances are ready, you can configure warm standby using the `setup_warm_standby.pyc` script.

If you do not know if one or both of the instances are already part of a warm standby configuration, check warm standby status before proceeding. See [How to check the status of warm standby in the Warm standby feature overview](#). Warm standby must be disabled before reconfiguring warm standby to use different instances. See [Disable warm standby](#).

Do these steps as either the root user or a user with sudo permissions.

1. On the primary Splunk SOAR (On-premises) instance, make sure that Splunk SOAR (On-premises) is running.
`/opt/phantom/bin/start_phantom.sh`
2. On the warm standby Splunk SOAR (On-premises) instance, make sure that Splunk SOAR (On-premises) is running.
`/opt/phantom/bin/start_phantom.sh`
3. On the primary Splunk SOAR (On-premises) instance, run the `setup_warm_standby.pyc` script.
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --configure --primary-ip <IP address of the primary> --standby-ip <IP address of the warm standby>`
 You will be prompted for:
 - ◆ The password for the user account Splunk SOAR (On-premises) on the warm standby. This password was set when the warm standby instance was created earlier.
 - ◆ Create a password for the database replication user. This password will be used to configure PostgreSQL database replication.
 - ◆ Configuration information to create the SSL certificate file used for communication between the primary and warm standby Splunk SOAR (On-premises) instances.

Example:

Country Code: US
State Code: CA
City: Palo Alto
Organization: Example
Organization Unit: Security
Domain: phantom.soc.example.com
Email: soc@example.com

4. On the warm standby Splunk SOAR (On-premises) instance, run the `setup_warm_standby.pyc` script.
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --configure --primary-ip <IP address of the primary> --standby-ip <IP address of the warm standby>`
5. On the warm standby re-enable the cron service.
`sudo systemctl start crond.service`

Failover to the warm standby

Failing over to the warm standby is a manual process.

- You can failover to the warm standby in the event of a systems failure with the primary instance of Splunk SOAR (On-premises).
- You may wish to failover even if the primary instance of Splunk SOAR (On-premises) is healthy in order to perform system maintenance or upgrades without significant downtime.

Failover procedure

Do these steps as the root user or a user with sudo permissions.

1. If the primary instance of Splunk SOAR (On-premises) is online, you must stop all Splunk SOAR (On-premises) services. The warm standby will not take over if it detects that the primary instance is still operating.
`/<PHANTOM_HOME>/bin/stop_phantom.sh`
2. SSH to your warm standby Splunk SOAR (On-premises) instance.
`SSH <username>@<warm_standby_phantom_hostname>`
3. Run the `setup_warm_standby.pyc` script to convert the standby to the primary.
On Splunk SOAR (On-premises) instances version 4.6.19142 or newer:
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --convert-to-primary --ignore-package-updates`
On Splunk SOAR (On-premises) instances version 4.6.18265 or earlier:
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --convert-to-primary`
4. Update DNS to resolve the hostname of your Splunk SOAR (On-premises) instance to the IP address of the new primary.
5. If you are ingesting from external services, you will need to update their configurations to use the new primary. Elasticsearch users will need to manually reindex in **Main Menu > Administration > Administration Settings > Search Settings**.

After the failover procedure, the warm standby is now the primary instance of Splunk SOAR (On-premises). The previous primary should be offline.

Do not reboot or restart Splunk SOAR (On-premises) services on the decommissioned primary. It can lead to two standalone instances of Splunk SOAR (On-premises) polling the same assets, and lead to data loss or other unwanted behavior.

Disable warm standby for Splunk SOAR (On-premises)

Disable warm standby to perform the following tasks:

- Perform system maintenance
- Configure a backup or restore your system
- Upgrade Splunk SOAR (On-premises)

If you want to enable warm standby again after disabling it, you must recreate it. See [Create a warm standby](#).

To disable warm standby, you must run commands on both the primary Splunk SOAR (On-premises) system and the warm standby system.

1. Log in to the Splunk SOAR (On-premises) primary system from the command line as either the root user or as a user with sudo permissions.
2. On the Splunk SOAR (On-premises) primary system, run the following command to turn off warm standby.
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --off`
3. Log in to the warm standby system from the command line as either the root user or as a user with sudo permissions.
4. On the warm standby system, run the following command to turn off warm standby.
`phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --off`
5. Continuing on the warm standby system, run the following command to stop all Splunk SOAR (On-premises) services.
`/<PHANTOM_HOME>/bin/stop_phantom.sh`

Warm standby is now disabled, and cron jobs are removed to prevent rsync jobs from running.

See also:

To perform tasks while warm standby is disabled, refer to the following resources:

- [Splunk SOAR \(On-premises\) backup and restore overview](#)
- [Splunk SOAR \(On-premises\) upgrade overview and prerequisites](#)
- [System maintenance and updates](#) in Splunk SOAR (On-premises) security information

Recreate warm standby after a failover

After a failover, the previous warm standby is now a standalone primary instance of Splunk SOAR (On-premises) and the previous primary is offline or otherwise unavailable. A Splunk SOAR (On-premises) administrator can reconfigure these two instances into a new warm standby pair.

For the rest of this topic the two Splunk SOAR (On-premises) instances will be referred to as either instance A or instance B.

Instance A

The original primary Splunk SOAR (On-premises) instance.

Instance B

The original warm standby instance of Splunk SOAR (On-premises).

Configure instance B as the primary and instance A as the warm standby

This is the easiest way to reconfigure the instances for warm standby after a failover.

The initial states for your instances must be:

- Instance A, the original primary is online but Splunk SOAR (On-premises) services are not running.
- Instance B, the former warm standby is now a stand alone Splunk SOAR (On-premises) instance.

If the Splunk SOAR (On-premises) instances are not in these states, stop. Evaluate if another option is more appropriate for your needs.

Do these steps as either the root user or a user with sudo permissions.

1. SSH to instance A.
SSH <username>@<instance_A_hostname>
 1. Start PostgreSQL.
/<PHANTOM_HOME>/bin/phsvc start postgresql-11
 2. Start pgbouncer.
/<PHANTOM_HOME>/bin/phsvc start pgbouncer
 3. Turn off primary mode
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --off
2. SSH to instance B.
SSH <username>@<instance_B_hostname>
 1. Configure instance B as the new primary.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>
3. SSH to instance A.
SSH <username>@<instance_A_hostname>
 1. Configure instance A as the new warm standby.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>

Now the two instances are configured for warm standby. Instance B is now the primary and instance A is now the warm standby.

Configure instance A as the primary and instance B as the warm standby

This option returns the instances to the same roles they served before the failover. This can be done after you have configured the instance B as the primary using the steps in the earlier section.

Each time warm standby is configured the database on the standby instance is erased and the entire Splunk SOAR (On-premises) PostgreSQL database has to be streamed from the primary.

The initial states for your instances must be:

- Instance B, the former warm standby is now a stand alone Splunk SOAR (On-premises) instance. All Splunk SOAR (On-premises) services are running.
- Instance A, the original primary is configured as the warm standby. All Splunk SOAR (On-premises) services are running.

If the Splunk SOAR (On-premises) instances are not in these states, stop. Evaluate if another option is more appropriate for your needs.

1. SSH to instance B.
SSH <username>@<instance_B_hostname>
1. Stop Splunk SOAR (On-premises) services.
/<PHANTOM_HOME>/bin/stop_phantom.sh
2. SSH to instance A.
SSH <username>@<instance_A_hostname>
1. Configure instance A as the primary.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --convert-to-primary
Warm standby is disabled. Instance A is a standalone Splunk SOAR (On-premises) instance, while instance B is idle and all Splunk SOAR (On-premises) services have been shut down.

If the Splunk SOAR (On-premises) instances are not in the described states, stop. Check for and do any steps which have been missed before proceeding.

3. SSH to instance B.
SSH <username>@<instance_B_hostname>
1. Start PostgreSQL.
/<PHANTOM_HOME>/bin/phsvc start postgresql-11
2. Start pgbouncer.
/<PHANTOM_HOME>/bin/phsvc start pgbouncer
3. Turn off primary mode
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --off
4. SSH to instance A.
SSH <username>@<instance_A_hostname>
1. Configure instance A as primary.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --primary-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>
5. SSH to instance B.
SSH <username>@<instance_B_hostname>
1. Configure instance B as the warm standby.
phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc --standby-mode --configure --primary-ip <primary_ip> --standby-ip <standby_ip>

Instance A and B are configured as a warm standby pair. Instance A is the primary, and instance B is the warm standby.

Upgrade or maintain warm standby instances

In order to perform system maintenance or to upgrade Splunk SOAR (On-premises) on a warm standby pair, warm standby must be disabled.

1. [Disable warm standby](#) on the primary.
2. [Disable warm standby](#) on the warm standby.

3. Perform system maintenance or upgrade Splunk SOAR (On-premises) for both instances. See Splunk SOAR (On-premises) upgrade overview and prerequisites in *Install and Upgrade Splunk SOAR (On-premises)*.
4. Create a warm standby pair. See [Create a warm standby](#).

Warm standby tools

Use the `phenv python /<PHANTOM_HOME>/bin/setup_warm_standby.pyc` script to manage warm standby.

Warm standby script arguments

Argument	Description
<code>-h, --help</code>	Show this help message and exit.
<code>--primary-mode</code>	Run the instance as the primary in the warm standby pairing.
<code>--standby-mode</code>	Run the instance as the warm standby in the warm standby pairing.
<code>--version</code>	Show the program's version number and exit.
<code>--status</code>	Show the status of the current Splunk SOAR (On-premises) instance.
<code>--configure</code>	Configure warm standby. Additional arguments are required.
<code>--off</code>	Turn warm standby off on the current instance based on which mode the instance is in.
<code>--convert-to-primary</code>	Convert a standby to primary valid only in case of <code>--standby-mode</code>
<code>--primary-ip <PRIMARY_IP></code>	IP address of the primary.
<code>--standby-ip <STANDBY_IP></code>	IP address of the warm standby.
<code>-d, --ignore-database</code>	Ignore the PostgreSQL database. Ignores the Postgres database during setup. Only backs up system files.
<code>-t, --ignore-vault</code>	Ignore vault. Ignores the vault from setup. Only backs up various contents from <code>/<PHANTOM_HOME>/</code> .
<code>-l <RECOVERY_DATABASE_LOCATION>, --recovery-database-location <RECOVERY_DATABASE_LOCATION></code>	When setting up the standby, copy the original database to this location for recovery in the event of a script failure.
<code>--primary-phantom-version <PRIMARY_PHANTOM_VERSION></code>	Version of the primary Splunk SOAR (On-premises) instance. Only valid for <code>--standby-mode</code> . If passed, validates against the current version.
<code>-r <REMOTE_USER>, --remote-user <REMOTE_USER></code>	The username of the remote user.
<code>-x, --relax_verification</code>	Relax user verification requirements for non-root installations. Setting this option is not recommended.
<code>-p <SSH_PORT>, --ssh-port <SSH_PORT></code>	Port used to be used by all SSH commands.
<code>--no-modify-ciphers</code>	Don't overwrite <code>ssl_cipher</code> in PostgreSQL configurations.
<code>-u, --ignore-package-updates</code>	Skip updating packages. Skips re-installing rpm and pip packages.
<code>--no-cron-install</code>	Set but don't install the warm standby crontab.
<code>--recreate-local-db</code>	Purge current database and generate a blank Splunk SOAR (On-premises) instance when turning off your standby instance.

Argument	Description
	This will delete all of your data.

-w <WAL_KEEP_SEGMENTS>, --wal-keep-segments <WAL_KEEP_SEGMENTS>The number of wal segments retained on the primary instance. Increase the wal segments to allow greater network latency between the primary instance and standby instance. Increasing wal segments will take up additional disk space in your DB directory, specifically 16 MB per segment.--replicator-password <REPLICATOR_PASSWORD>Password for the postgres replicator role. It can also be provided via the "PHANTOM_WARM_STANBY_REPLICATOR_PASSWORD" environment variable.--ssh-password <SSH_PASSWORD>Password for the remote user. Can also be provided via the "PHANTOM_WARM_STANBY_SSH_PASSWORD" environment variable.

SSL certificate information

The following arguments are options for the data required to generate an SSL certificate while configuring warm standby.

Argument	Description
--ssl-country <SSL_COUNTRY>	Value for a SSL certificate with the country code subject line.
--ssl-state <SSL_STATE>	Value for a SSL certificate with the state code subject line.
--ssl-city <SSL_CITY>	Value for a SSL certificate with the city subject line.
--ssl-org <SSL_ORG>	Value for a SSL certificate with the organization subject line.
--ssl-unit <SSL_UNIT>	Value for a SSL certificate with the organization unit subject line.
--ssl-domain <SSL_DOMAIN>	Value for a SSL certificate with the domain subject line.
--ssl-email <SSL_EMAIL>	Value for a SSL certificate with the email subject line.

Warm standby API

The API /rest/warm_standby_check can be used to determine if a Splunk SOAR (On-premises) instance is the standby in a warm standby pair. See REST Warm standby.

The API returns the same 500 result if used on either a warm standby or a cluster node. Clusters cannot use the warm standby feature.

Manage your Splunk SOAR (On-premises) Apps and Assets

Add and configure apps and assets to provide actions in Splunk SOAR (On-premises)

Splunk SOAR (On-premises) apps expand the capabilities of your Splunk SOAR (On-premises) instance by enabling connections to third party products and services. These third-party products and services provide actions you can run or automate in your Splunk SOAR (On-premises) playbooks. For example, the MaxMind app provides the **geolocate ip** action for your Splunk SOAR (On-premises) deployment.

You can upgrade existing apps or install new apps at any time without having to upgrade the entire Splunk SOAR (On-premises) platform.

Apps have full access to the operating system and there are no security restrictions on any app while it is running.

An asset is a specific configuration, or instance, of an app. An asset is configured with the information required to communicate with the third-party product or service, such as IP address, automation service account, username, and password.

For example, Splunk SOAR (On-premises) ships with a VMware vSphere app enabling Splunk SOAR (On-premises) to get information from and take actions against a vSphere host. You can use Splunk SOAR (On-premises) to start and stop VMs, take snapshots, and download memory snapshots for analysis. In order for the app to be able to communicate with your vSphere servers, you must provide login credentials such as the hostname or IP address. You might have multiple vSphere servers, such as several individual ESXi hosts, or you might have them centralized onto one vCenter server. To tell Splunk SOAR (On-premises) about a given vSphere server, create a vSphere asset and provide the address and credentials needed for that server. You can then create another vSphere asset with a different address and credentials if needed. When taking actions, you specify which asset the action is for.

This table shows how multiple vSphere assets are configured from a vSphere app:

Splunk SOAR (On-premises) app	Configure multiple assets from a single app
VMware vSphere	vSphere 1 <ul style="list-style-type: none">• IP address 192.168.1.1• User admin1, password example1
	vSphere 2 <ul style="list-style-type: none">• IP address 192.168.1.2• User admin2, password example2
	vSphere 3 <ul style="list-style-type: none">• IP address 192.168.1.3• User admin3, password example3

View your Splunk SOAR (On-premises) apps

Splunk SOAR (On-premises) ships with hundreds of apps already installed. You can find more apps on splunkbase, from other users, and even create your own. See Splunk SOAR (On-premises) apps overview in *Develops Apps for Splunk*

SOAR (On-premises).

Perform the following tasks to view the apps provided by Splunk SOAR (On-premises) on the Apps page.

1. From the **Home** menu, select **Apps** to access the Apps page.
2. View the list of configured apps on the **Configured Apps** tab. Any app that has at least one asset configured appears on this page. You can expand each asset to view the configured assets and available actions provided by the app. Click **Configure New Asset** to configure a new asset for the app. See [Add a new Splunk SOAR \(On-premises\) asset](#).
3. (Optional) Click **Unconfigured Apps** to view the list of apps installed on your Splunk SOAR (On-premises) instance that do not have at least one asset configured.
4. (Optional) Click **Orphaned Assets** to review any assets that no longer have a corresponding app installed.

Install, update, or delete apps on Splunk SOAR (On-premises)

Navigate to the Apps page to install, update, or delete Splunk SOAR (On-premises) apps.

Install a new Splunk SOAR (On-premises) app

Perform the following steps to install a new Splunk SOAR (On-premises) app:

1. Obtain the new app or develop a new app. See Splunk SOAR (On-premises) apps overview in *Develops Apps for Splunk SOAR (On-premises)*.
2. From the **Home** menu, select **Apps**.
3. Click **Install App**.
4. Drag and drop a .tar or .rpm archive of the app into the file field, or click in the file field and navigate to the location of the app file on your system.
5. Click **Install**.

The new app is available on the **Unconfigured Apps** tab of the Apps page.

For compatibility needs, you can install multiple versions of the same app. However, only one version of the app can be active at a time.

Switching the active version of an app may have unintended consequences. For example, there might be differences among the actions, parameters, or output depending on the version of the app. Be sure to modify any playbooks as needed to be compatible with the active version of the app.


Update existing Splunk SOAR (On-premises) apps

To update an existing Splunk SOAR (On-premises) app, perform the following steps:

1. From the **Home** menu, select **Apps**.
2. Click **App Updates**.
3. Select any apps with available updates.
4. Click **Update**.

Delete a Splunk SOAR (On-premises) app

Perform the following steps to delete a Splunk SOAR (On-premises) app:

1. From the **Home** menu, select **Apps**.
2. Click the trash can () icon for the app you want to delete.
3. Click **Delete** to confirm you want to delete the app.

You can re-install any app that you deleted by downloading the app and installing the app again.

View your Splunk SOAR (On-premises) assets

Splunk SOAR (On-premises) ships with one asset for the DNS, MaxMind, PhishTank, REST Data Source, and WHOIS apps already configured.

To view configured assets, perform the following tasks:

1. From the **Home** menu, select **Apps**.
2. Verify the **Configure Apps** tab is selected.
3. In any app, click the arrow icon corresponding to **configured assets** to expand the section and view the assets. For example, if an app shows **3 configured assets**, click on the arrow to view the configured assets. You can hover over the asset to edit or delete the asset.

Add, edit, or delete a Splunk SOAR (On-premises) asset

Manage the assets in your Splunk SOAR (On-premises) instance. You can add a new asset, and edit or delete existing assets.

Add a new Splunk SOAR (On-premises) asset

Perform the following steps to create a new Splunk SOAR (On-premises) asset:

1. From the **Home** menu, select **Apps**.
2. Click **Configure New Asset** for the desired app.
3. In the **Asset Name** field, enter a name for the asset such as **firewall**. This name is the one you use when referring to the asset in scripts. Specify the name as a string without spaces or punctuation.
4. (Optional) In the **Asset Description** field, enter a longer and more descriptive name for this asset, such as **Perimeter Firewall for the engineering network**.
5. (Optional) Enter one or more tags for the asset. You can use the same tag for multiple assets to group them together, and then perform actions on all assets with matching tags. See [Add tags to objects in Splunk SOAR \(On-premises\)](#).
6. Click **Save**.

The amount of configuration required for each asset is determined by the app. Some assets require additional configuration. For example, if you configure a QRadar asset, you must also configure settings on the **Asset Settings** and **Ingest Settings** tabs before you can save the configuration.

- Most assets require authentication information so that Splunk SOAR (On-premises) can connect to the desired server or service. You can configure authentication for an asset on the **Asset Settings** tab.
- Data ingestion settings, such as polling intervals and where to put the data once the data is ingested, are configured on the **Ingest Settings** tab. The destination for ingested data is called a container in Splunk SOAR

(On-premises).

Edit a Splunk SOAR (On-premises) asset

Perform the following steps to edit a Splunk SOAR (On-premises) asset:

1. From the **Home** menu, select **Apps**.
2. Make sure the **Configured Apps** tab is selected.
3. Click on the number of configured assets in the app to expand the section.
4. In the table of configured assets, click the asset you want to edit.
5. Click **Edit**, then make any desired changes. You can edit an asset's description, tags, settings, and approval settings. To change the asset name, you must delete the current asset and create a new asset with the desired name.
6. Click **Save**.

Reassign an orphaned Splunk SOAR (On-premises) asset

You can now assign orphaned assets to an App from the user interface.

1. From **Home > Apps > Orphaned Assets** select the orphaned asset.
2. Click **Assign App**.
3. In the dropdown menu, select the App, then click **Assign**.

Delete a Splunk SOAR (On-premises) asset

Perform the following steps to delete a Splunk SOAR (On-premises) asset.

1. From the **Home** menu, select **Apps**.
2. Make sure the **Configured Apps** tab is selected.
3. Click on the number of configured assets in the app to expand the section.
4. In the table of configured assets, click the asset you want to delete.
5. Click **Delete Asset**.
6. Click **Confirm** to confirm that you want to delete the asset.

Configure advanced asset settings

Configure advanced asset settings such as the concurrent action limit, just in time (JIT) credentials, automation users, asset environment variables, and proxies.

Set the concurrent action limit

You can run concurrent actions on an existing asset, or on a new asset by following these steps:

1. From the Splunk SOAR (On-premises) **Home** menu, select **Apps**.
2. Find the app you want to run an action on and click **Configure New Asset**. Or, to run concurrent actions on an existing asset, click on your desired preexisting asset.
3. Click the **Asset Setting** tab > **Advanced**.
4. In the **Concurrent Action Limit** box, enter the number of concurrent actions you want to run on your asset. You can run up to 10 actions at once. Use caution when changing this limit as it can significantly affect performance.
5. Run the actions on an asset; evaluate performance.

For information on setting the global action concurrency limit, see [Set the global action concurrency limit](#).

Disable action lock or action concurrency

Within an action entry, the optional lock key defines a set of parameters that you can set to run actions concurrently.

- A lock is represented by its name.
- Multiple actions locking on the same name will be serialized even if the actions are from different apps.
- In the absence of a lock dictionary, the platform runs the actions concurrently using the asset as the lock name.

To disable the lock for an action, the lock dictionary must be present and the "enabled" key set to false. When "enabled" is set to false, you can run as many concurrent actions as you like.

```
"lock": {  
  "enabled": false,  
  "data_path": "parameters.hash",  
  "timeout": 600  
}
```

Parameter	Required?	Description
<i>enabled</i>	Required	Boolean value that specifies if the lock is enabled or not for this action.
<i>data_path</i>	Optional	The name of the lock. Only valid if lock is enabled. This value is either a datapath that points to a parameter of the action with <code>parameters.hash</code> where <code>hash</code> is one of the parameters of the action, or a datapath that points to a configuration parameter for something like <code>configuration.server</code> . At runtime, the platform will read the values stored in these data paths and use it as the name of the lock. You can also use a constant string, for example, any string that does not start with <code>configuration.</code> or <code>parameters.</code> The platform will use this value as is. In case the <code>data_path</code> is not specified, the asset will be used as the lock name.
<i>timeout</i>	Optional	Specifies the number of seconds to wait to acquire the lock, before an error condition is reported.

If you have multiple actions with the lock enabled that are scheduled to run on an asset, you may want to exclude only some of them from running concurrently. To exclude a certain action from running concurrently, set concurrency to false in the app JSON. When both "enabled" and "concurrency" are set to true, you can run multiple actions concurrently up to the concurrent action limit. When "enabled" is set to true and "concurrency" is set to false, you can only run a single action.

```
"lock": {  
  "enabled": true,  
  "concurrency": false  
}
```

Parameter	Required?	Description
<i>enabled</i>	Required	Boolean value that specifies if the lock is enabled or not for this action.
<i>concurrency</i>	Optional	By default concurrency is set to <code>true</code> to allow concurrent actions to run on an app. Set concurrency to <code>false</code> to opt out of concurrent actions running on an app.

If the lock is enabled on an action, but concurrency is set to false in the app.json, the action will not be counted in the concurrent action limit you set in Asset Settings.

Configure Just In Time Credentials for a Splunk SOAR (On-premises) asset

Some assets can be configured to use just in time (JIT) credentials, which require a Splunk SOAR (On-premises) user to type in credentials before any further action is taken. Use JIT credentials if your organization has policies against providing credentials in an automated manner, or if you are using one-time passwords.

To configure JIT credentials, perform the following steps:

1. Navigate to the asset configuration page.
2. Click the **Asset Settings** tab.
3. Click **Advanced** to expand the section.
4. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
5. In the **Enable Just in Time credentials for** field, select the fields for which you want to enable JIT authentication.
For example, select **username** and **password** to enable JIT for login credentials.
6. Click **Save**.

Once enabled, JIT uses the asset's approval settings to determine the set of users that must supply the credentials to complete the action. See [Configure approval settings for a Splunk SOAR \(On-premises\) asset](#).

To use JIT, you must have at least one approver set up for the asset. If you have selected multiple users that require a quorum to approve, then the last user (the one that would cast the final vote that causes the action to run) must be the one who supplies correct credentials. Earlier users can supply credentials, but the last user supplies the set that is actually used. Anything entered before that user is overwritten by the last user. Note that even if you have "Automatic self-approval" configured in Splunk SOAR (On-premises) for your own approval vote, you still receive a JIT prompt when credentials are required.

Configure automation users for a Splunk SOAR (On-premises) asset

Define the automation user to specify the service account Splunk SOAR (On-premises) uses to run the asset. The default account is the **automation** account provided by Splunk SOAR (On-premises).

Perform the following tasks to create a custom automation user in Splunk SOAR (On-premises):

1. Navigate to the asset configuration page.
2. Click the **Asset Settings** tab.
3. Click on **Advanced** to expand the section.
4. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
5. In the **Select a user on behalf of which automated actions can be executed (e.g. test connectivity, ingestion)** field, select the desired automation user.
6. Click **Save**.

Configure environment variables for a Splunk SOAR (On-premises) asset

Environment variables configured in an asset take precedence over any global environment variables. Perform the following tasks to set environment variables for a Splunk SOAR (On-premises) asset:

1. Navigate to the asset configuration page.
2. Click the **Asset Settings** tab.
3. Click on **Advanced** to expand the section.
4. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
5. Click **+ Variable** to add a new environment variable.
6. Enter the name and value of the variable.
7. (Optional) Click **Secret** to encrypt the value so that it is not displayed in the Splunk SOAR (On-premises) web interface.
8. (Optional) Click **+ Variable** to add more variables as needed.
9. Click **Save**.

See [Configure proxies for a Splunk SOAR \(On-premises\) asset](#) for information on how to set environment variables so that the asset can use a proxy.

Configure proxies for a Splunk SOAR (On-premises) asset

Perform the following steps to configure the environment variables needed for the app to communicate with a proxy:

1. Navigate to the asset configuration page.
2. Click the **Asset Settings** tab.
3. Click **Advanced** to expand the section.
4. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
5. Click **+ Variable** to add a new environment variable.
6. Configure the **HTTP_PROXY**, **HTTPS_PROXY**, or **NO_PROXY** variables depending on the type of proxy connection.
 - ◆ For **HTTP** and **HTTPS** proxy configurations, include the protocol, hostname or IP address, and the port of the proxy server. For example: `<Protocol>://<Hostname/IP>:<Port>`
 - ◆ For **NO_PROXY** configurations, include the IP address, hostname, or domain of the asset.
7. (Optional) Click **Secret** to encrypt the value so that it is not displayed in the Splunk SOAR (On-premises) web interface.
8. Click **Save**.

The table shows an example of how to configure HTTP, HTTPS, and no proxy for a Splunk SOAR (On-premises) asset. For apps that use requests, configuring both HTTPS and HTTP environment variables directs all app traffic through the proxy server.

Proxy Name	Proxy Value
HTTP_PROXY	http://192.168.13.1:80
HTTPS_PROXY	https://192.168.13.100:8800
NO_PROXY	example.com

Configure ingest settings for a Splunk SOAR (On-premises) asset

Data ingestion settings are available for assets such as QRadar, Splunk, and IMAP. Perform the following steps to configure ingestion settings for a Splunk SOAR (On-premises) asset:

1. Navigate to the Asset Configuration page.
2. Click the **Ingest Settings** tab.
3. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
4. In the **Label to apply to objects from this source** field, select a container label you want to apply to objects from this source. You can also type in a new label name.
5. (Optional) Configure a polling interval for the asset to ingest data.
 - ◆ Select **Interval** to configure the number of minutes between polls.
 - ◆ Select **Scheduled** to view additional options and intervals.
6. (Optional) Some assets have a **Process Missed Jobs** checkbox. Check this box if you want Splunk SOAR (On-premises) to process any missed jobs. Jobs can be missed in cases where Splunk SOAR (On-premises) is not running, or one poll didn't complete before the next one started.
7. Click **Save**.

Configure approval settings for a Splunk SOAR (On-premises) asset

Assets created with no approvers run immediately. It is usually an acceptable company policy for an asset providing a whois lookup action. For assets such as firewalls, company policies usually restrict access to the ability to change firewall settings. Any actions performed on a firewall asset must go through the approval process.

Configure the approval settings for a Splunk SOAR (On-premises) asset to determine who must approve the actions taken against the asset. See Approve actions before they run in Splunk SOAR (On-premises) in the *Use Splunk SOAR (On-premises)* manual.

To configure approval settings for an asset, perform the following steps:

1. Navigate to the asset configuration page.
2. Click the **Approval Settings** tab.
3. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
4. Select the users and roles you want to configure as primary approvers. Click the arrow keys to add or remove users and roles to the **Primary Approvers** field.
5. Select the number of required primary approvers from the drop-down list in the **Required primary approvers** field.
6. Select the users and roles you want to configure as secondary approvers. Click the arrow keys to add or remove users and roles to the **Secondary Approvers** field.
7. Select the number of required secondary approvers from the drop-down list in the **Required secondary approvers** field.
8. Click **Save**.

Configure the tenant assigned to a Splunk SOAR (On-premises) asset

Assign a tenant to an asset to separate data and make sure that the asset is only used with the container with the same tenant. You can only assign tenants to an asset if multi-tenancy is configured and enabled in Splunk SOAR (On-premises). See [Configure multiple tenants on your Splunk SOAR \(On-premises\) instance](#).

Perform the following steps to assign a tenant to a Splunk SOAR (On-premises) asset:

1. Make sure multi-tenancy is enabled on your Splunk SOAR (On-premises) instance.
2. Navigate to the asset configuration page.
3. Click the **Tenants** tab.
4. Click **Edit** if you are editing an existing asset. You don't need to do this if you are configuring a new asset.
5. Select the desired tenants from the **Available Tenants** box and click the arrows to move them to the **Mapped to Asset** box.
 - ◆ Non-ingestion assets that do not have a tenant assigned are available to all tenants. You can assign multiple tenants to a non-ingestion asset.
 - ◆ Ingestion assets must have one tenant assigned. You can't assign multiple tenants. If no tenant is selected in the asset configuration, the default system tenant is assigned to the asset and any containers created by the asset.
6. Click **Save**.

Splunk SOAR (On-premises) telemetry

Share data from Splunk SOAR (On-premises)

When Splunk SOAR (On-premises) is deployed, the platform sends anonymized usage data to Splunk Inc. ("Splunk") to help improve Splunk SOAR (On-premises) in future releases.

How data is collected

Splunk SOAR (On-premises) uses Splunk Web Analytics (swa.js) to collect anonymous usage data. These analytics run in the background. Collecting data affects the Splunk SOAR (On-premises) web interface loading in a minimal way.

Share data from Splunk SOAR (On-premises)

When Splunk SOAR (On-premises) is deployed, the platform sends anonymized usage data to Splunk Inc. ("Splunk") to help improve Splunk SOAR (On-premises) in future releases. You can opt in or opt out of sharing telemetry data.

Enable telemetry by doing the following:

1. From the main menu, select **Administration**.
2. Expand the **Product Settings** drop-down list.
3. Click **Telemetry**.
4. Toggle the switch to the **On** position.
5. Click **Confirm**.

Disable telemetry by doing the following:

1. From the main menu, select **Administration**.
2. Expand the **Product Settings** drop-down list.
3. Click **Telemetry**.
4. Toggle the switch to the **Off** position.
5. Click **Confirm**.

How data is collected

Splunk SOAR (On-premises) uses Splunk Web Analytics (swa.js) to collect anonymous usage data. These analytics run in the background regardless of whether you opt in to sending usage data to Splunk. Collecting data affects the Splunk SOAR (On-premises) UI loading in a minimal way. Performance numbers are currently being gathered to compare with a baseline Splunk SOAR (On-premises) system with no telemetry.

What data is collected

Data is collected to measure metrics of the product, assess performance for optimizations, evaluate engagement for roadmaps, and discover client-side errors to inform UI fixes. The metrics do not contain any user-provided values such as username, email, or any URL parameters that are user or customer identifiable. Splunk SOAR (On-premises) collects the following basic usage information:

Name	Description	Example
------	-------------	---------

app.session.session_start	Reports the browser and OS, along with their versions.	<pre> data: { app: UNKNOWN_APP browser: Chrome browserVersion: 78.0.3904.97 device: MacIntel locale: en-US os: Mac OS X osVersion: 10. page: UNKNOWN_PAGE splunkVersion: not available } eventID: d9ca862c-d48d-83a1-d1bb-f0f25f4b5af8 experienceID: 6c2c534b-e750-e1a0-95fd-fcadala50be0 optInRequired: 3 timestamp: 1574213029 visibility: anonymous </pre>
app.session.phantom.pageview	Reports which pages are visited by users.	<pre> data: { app: phantom page: admin.company_settings.info phantomDeploymentID: phantom-a2a983de-38ec-42d7-a179-30087b0ca8ca phantomUserID: 5d900c28b8d1555745c09908ef386860 } eventID: 0db11144-7c14-88f7-b3e9-3a999102bfc6 experienceID: 20d4d671-7d18-f74a-c72f-9811b5bee20d optInRequired: 3 timestamp: 1574210581565 visibility: anonymous </pre>
app.session.phantom.error	Reports uncaught errors of front-end Splunk Phantom scripts.	<pre> data: { app: phantom errorMsg: Uncaught ReferenceError: helloworld is not defined file: /inc/swa/swa_enabled.js page: admin.product_settings.telemetry position: 74:1 phantomDeploymentID: phantom-a2a983de-38ec-42d7-a179-30087b0ca8ca phantomUserID: 5d900c28b8d1555745c09908ef386860 } eventID: 94efce66-ab89-33ae-f894-1cceb8f68f78 experienceID: 239facf6-261d-dd96-be08-33870c7d3750 optInRequired: 3 timestamp: 1574294947704 visibility: anonymous </pre>
app.session.phantom.apiTime	Reports roundtrip time consumption for each API request.	<pre> data: { app: phantom endpoint: /rest/ph_user/3/permissions method: get page: UNKNOWN_PAGE </pre>

		5d900c28b8d1555745c09908ef386860 } eventID: 5854bede-18d9-5a88-d023-e698dab1afaf experienceID: 31a418cc-1371-c58a-a0b8-dc87638b126f optInRequired: 3 timestamp: 1575656115189 visibility: anonymous
Name	Description	Example
app.session.phantom.systemSettings	Reports the feature on/off settings and product version.	<pre> component: app.session.phantom.systemSettings data: { app: phantom isClusteringEnabled: false isMultiTenantEnabled: false numOfClusterNodes: 0 page: UNKNOWN_PAGE productVersion: 10900.0.5 nodeGUID: dca36837-3e10-4cbd-bf14-b49097b84347 searchConfig: { isElasticSearchEnabled: false searchLocation: local searchType: standalone } phantomDeploymentID: phantom-a2a983de-38ec-42d7-a179-30087b0ca8ca phantomUserID: 5d900c28b8d1555745c09908ef386860 } eventID: d4b331e7-3ce3-91b6-7724-bc4d7235bca9 experienceID: 21febb16-c3f6-cbd5-ffac-905f1466c830 optInRequired: 3 timestamp: 1576695256840 visibility: anonymous </pre>
app.session.phantom.vpe	Reports: <ul style="list-style-type: none"> • VPE version (Classic or Modern) • The types of blocks in a playbook • The number of blocks in a playbook • Which hotkey shortcuts were used while editing a playbook • Specific SOAR features used in a playbook 	<pre> data: { app: soar jsonSchemaVersion:"5.0.3" page: UNKNOWN_PAGE blocks: { totalCount: 14 blockTypes: { action: 2 playbook: 1 code: 1 utility: 1 filter: 1 decision: 1 format: 6 prompt: 1 } } customCodeBlockCount: 3 customCodeBlockTypeCounts: { start: 0 end: 1 action: 2 playbook: 0 code: 0 utility: 0 filter: 0 decision: 0 format: 0 prompt: 0 } actions: ["geolocate ip", "whois domain"] } hotkeys: { totalCount: 14 </pre>

		<pre> "mac address": 0 "port": 0 "process name": 0 "url": 0 "user name": 0 } dedupeCount: 0 </pre>
Name	Description	Example
		<pre> playbookType: automation playbookName: 5d900c28b8d1555745c09908ef133337 soarDeploymentID: soar-a2a983de-38ec-42d7-a179-30087b0ca8ca soarUserID: 5d900c28b8d1555745c09908ef386860 } deploymentID: soar-a2a983de-38ec-42d7-a179-30087b0ca8ca eventID: d4b331e7-3ce3-91b6-7724-bc4d7235bca9 experienceID: 21febb16-c3f6-cbd5-ffac-905f1466c830 optInRequired: 3 timestamp: 1576695256840 visibility: anonymous </pre>
app.session.phantom.vpeTime	Reports the time in milliseconds it took for the VPE to load in the browser.	<pre> data: { app: soar pageLoadTime: 10298 } deploymentID: soar-a2a983de-38ec-42d7-a179-30087b0ca8ca eventID: d4b331e7-3ce3-91b6-7724-bc4d7235bca9 experienceID: 21febb16-c3f6-cbd5-ffac-905f1466c830 optInRequired: 3 timestamp: 1576695256840 visibility: anonymous </pre>