Splunk® SOAR (On-premises)
Install and Upgrade Splunk SOAR (On-premises) 5.4.0

Generated: 11/03/2022 11:54 am

Table of Contents

Get Splunk SOAR (On-premises)

How can Splunk SOAR (On-premises) be installed?

Installing Splunk SOAR (On-premises) is the first step to realizing gains from security orchestration and automation. Splunk SOAR (On-premises) allows you to combine security infrastructure orchestration, playbook automation, and case management capabilities to integrate your team, processes, and tools together.

If you are a system administrator who is responsible for setting up Splunk SOAR (On-premises), this guide will help you understand how to get Splunk SOAR (On-premises), the system requirements for installing Splunk SOAR (On-premises), how to install Splunk SOAR (On-premises), as well as clustering, advanced setup, and upgrading Splunk SOAR (On-premises).

You can install Splunk SOAR (On-premises) in the following ways:

- Install Splunk SOAR (On-premises) using the Amazon Marketplace Image
- Install Splunk SOAR (On-premises) as an unprivileged user

There are several options to choose from when you deploy Splunk SOAR (On-premises). Work with your Splunk SOAR (On-premises) Delivery Team representative to choose the right method and options for your organization.

The following table lists your options with links to relevant portions of this manual:

Installation option	Additional information
Clustered, unprivileged	Start with the clustering overview in About Splunk SOAR (On-premises) clusters. Then follow the instructions in Create a Splunk SOAR (On-premises) cluster using an unprivileged installation
SOAR instance with one or more external services	Your Splunk SOAR (On-premises) deployment can externalize services such as the PostgreSQL database, file shares, a load balancer, or a Splunk Enterprise or Splunk Cloud deployment. For each of the options, see the relevant documentation. • Set up an external PostgreSQL server • Set up external file shares using GlusterFS • Set up a load balancer with an HAProxy server • Set up Splunk Enterprise
SOAR instance with Federal Information Processing Standard (FIPS) support	A new, unprivileged deployment of Splunk SOAR (On-premises) can be created in a FIPS-compliant mode. The underlying operating system kernel must be in FIPS mode. Information about setting up RHEL 7.x or CentOS 7.x in Federal Information Processing Standard (FIPS) mode can be found on the Red Hat Security Guide in Chapter 9. Federal Standards and Regulations. You can only deploy a new, unprivileged instance of Splunk SOAR (On-premises) in FIPS-compliant mode. This choice cannot be undone and cannot be changed after deployment.

Installation option	Additional information

See also

- Administer Splunk SOAR (On-premises) in the *Administer Splunk SOAR (On-premises)* manual to learn more about settings and user management.
- About Splunk SOAR (On-premises) in *Use Splunk SOAR (On-premises)*.

Get Splunk SOAR (On-premises)

To get Splunk SOAR (On-premises), you must do one of the following:

- Register and create a Splunk SOAR (On-premises) community account.
- Purchase an AWS Marketplace machine image

Register and create a Splunk SOAR (On-premises) community account

Visit the Splunk SOAR (On-premises) community website to register and create an account. After your account is approved, you can download installation packages from the **Product** link.

If you don't see the installation package you need, contact your sales or delivery team representative.

After you download the Splunk SOAR (On-premises) software, install Splunk SOAR (On-premises) by following the linked instructions.

- Download the unprivileged tarball to install Splunk SOAR (On-premises). See Install Splunk SOAR (On-premises) as an unprivileged user.

Purchase an AWS Marketplace machine image

Install Splunk SOAR (On-premises) for AWS from the AWS Marketplace in the security category. See Install Splunk SOAR (On-premises) using the Amazon Marketplace Image.

Uninstall Splunk SOAR (On-premises)

You can uninstall Splunk SOAR (On-premises) or remove and replace it with a new instance.

Uninstall the application

To uninstall Splunk SOAR (On-premises), run this command:

```
$./soar-uninstall
```

If the soar-install script and the home directory for Splunk SOAR (On-premises) are in different locations, you must use the --phantom-home argument to tell the script where the home directory is.

If you specify `--no-prompt`, the script doesn't warn you before uninstalling Splunk SOAR (On-premises).

Although you can uninstall Splunk SOAR (On-premises), this process won't remove nodes from clusters.

Remove the application

To remove your existing Splunk SOAR (On-premises) installation or replace your existing installation with a new instance, perform these tasks:

1. Create a fresh version of your operating system.
2. Install Splunk SOAR (On-premises). Begin with How can Splunk SOAR (On-premises) be installed?

System Requirements

General system requirements

Splunk SOAR (On-premises) requires certain minimum system requirements. Your environment must meet or exceed these requirements. This section details operating systems, web browsers, system storage, Linux file systems, and other requirements for operating Splunk SOAR (On-premises).

Supported operating systems

Splunk SOAR (On-premises) supports these operating systems and versions:

- Red Hat Enterprise Linux 7.6 through 7.9
- CentOS 7.6 through 7.9

As a rule of thumb, the most recent minor version of a Red Hat Enterprise Linux or CentOS 7 operating system is supported.

Federal Information Processing Standard (FIPS) support

Splunk SOAR (On-premises) can be deployed in a FIPS compliant mode, if the operating system kernel is in FIPS mode.

- Your operating system, either RHEL or CentOS must be in FIPS mode.
- You must create a new, unprivileged deployment of Splunk SOAR (On-premises), either as a single instance or as a cluster.

Information about setting up RHEL 7.x or CentOS 7.x in Federal Information Processing Standard (FIPS) mode can be found in the Red Hat Security Guide in Chapter 9.

Supported browsers

Splunk SOAR (On-premises) requires a web browser that supports HTML 5, SVG graphics, and TLS.

Use the latest, fully patched version of one of the following browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Apple Safari

Operating system accounts

On unprivileged deployments, only a single operating system user account, phantom, is created and used.

Supported file systems and required directories

Splunk SOAR (On-premises) supports any file system where the user account running the application can be given write permissions.

In a clustered environment, Splunk SOAR (On-premises) implements GlusterFS for its file shares. If your organization requires a different file system for your Splunk SOAR (On-premises) cluster, make sure that the user account running Splunk SOAR (On-premises) has write permissions to the required directories.

Required directories for an installation as an unprivileged user:

- <phantom_install_dir>/apps
- <phantom_install_dir>/local_data/app_states
- <phantom_install_dir>/scm
- <phantom_install_dir>/vault
- <phantom_install_dir>/tmp/shared

File permissions

Splunk SOAR (On-premises) is installed in the following environments:

- On an unprivileged AMI deployment - /opt/phantom, also called <PHANTOM_HOME>.
- On an unprivileged deployment - the home directory of the user account that will run Splunk SOAR (On-premises), also called <PHANTOM_HOME>.

The installer expects a umask of 0022 during installation. Applying a different umask may lead to unexpected behavior.

In general, you should not modify file permissions for Splunk SOAR (On-premises). Changing the file permissions can cause errors, or prevent Splunk SOAR (On-premises) from working.

You can check to see if an access control list has been applied using the Linux `getfacl` command, clear any access control list which is incorrectly being applied using the `setfacl -b` command, or apply correct permissions to a file with the `chmod` command. If you have changed file permissions, you will need to restart Splunk SOAR (On-premises).

Directory	Permissions (symbolic)	Permissions (numeric)	Owner	Group	Notes
/opt/phantom	drwxr-xr-x	755	phantom	phantom	This is the default Splunk SOAR (On-premises) 'root' directory. On an unprivileged deployment, it changes to be the user account that runs Splunk SOAR (On-premises). Referred to as <PHANTOM_HOME> in the documentation.
/opt/phantom/apps	drwxrwxr-x	775	phantom	phantom	Required to allow the web-based UI to install apps. Apps installed by the web-based UI will be owned by nginx in the phantom group.
/opt/phantom/local_data	drwxrwxr-x	775	phantom	phantom	
/opt/phantom/local_data/app_states	drwxrwxr-x	775	phantom	phantom	
/opt/phantom/scm	drwxrwx---	770	phantom	phantom	Allows for non-nginx users of to have write access to playbooks.
/opt/phantom/spool	drwxrwxr-x	775	phantom	phantom	Allows the nginx user of the phantom group to have access to create items, such as the uwsgi sub-directory.

Directory	Permissions (symbolic)	Permissions (numeric)	Owner	Group	Notes
/opt/phantom/tmp	drwxrwx---	770	phantom	phantom	Allows non-root users of the phantom group to have write access.
/opt/phantom/vault	drwxrwxr-x	775	phantom	phantom	Allow non-phantom user of phantom group, such as the nginx user, to have the write access to add the file to vault, to create reports, and so on.
/opt/phantom/var/log	drwxr-xr-x	755	phantom	phantom	Allows the web-based UI and other tools to create and write log files for Splunk SOAR (On-premises) actions. You should not modify the permissions for this directory. If logs cannot be written, app installation or other actions may fail. On a privileged deployment, logging is done in /var/log/phantom/.

/opt/phantom/var/log/

phantom/app_install.log

-rw-rw-r--664phantomphantomAllows the web-based UI to write to the app_install.log and other tools to read it. You should not modify the permissions for this file. If this log cannot be written to, the Splunk SOAR (On-premises) web-based UI displays the error message "internal server error."

On a privileged deployment, logging is done in /var/log/phantom/.

/opt/phantom/var/log/

phantom/app_interface.log

-rw-rw----660phantomphantomContains logs from the app-interface module, REST handlers, and apps that provide custom views.

On a privileged deployment, logging is done in /var/log/phantom/.

System requirements for evaluation use

Your evaluation system must meet or exceed the listed requirements:

System Area	Requirement
Hypervisor for virtual machine images.	VMware Fusion, VMware Workstation, VMware Player, Oracle VirtualBox
Operating system	Red Hat Enterprise Linux 7.6 through 7.9, CentOS 7.6 through 7.9
Processor	1 CPU with a minimum of 4 cores
Memory	Minimum 8GB RAM, recommended 16GB
Storage	Minimum 500GB of disk space. Disk space requirements vary based on the volume of data consumed and the size of your evaluation environment.
Network	1 network interface

System Area	Requirement
System utilities	• cron ♦ On unprivileged deployments, the user account that runs Splunk SOAR (On-premises) must have permission to create cron jobs. • ntp or chrony

System requirements for production use

Systems for production must meet or exceed the listed requirements:

System Area	Requirement
Hypervisor for virtual machine images	VMware vSphere ESX/ESXi 5 or higher
Operating system	Red Hat Enterprise Linux 7.6 through 7.9, CentOS 7.6 through 7.9
Processor	1 server-class CPU, 4 to 8 cores
Memory	Minimum of 16GB RAM, 32GB recommended
Storage	Splunk SOAR (On-premises) needs storage for multiple volumes: • Splunk SOAR (On-premises) home directory also known as <PHANTOM_HOME>: 500GiB ♦ mounted as either /opt/phantom/ or as <PHANTOM_HOME> • PostgreSQL database: 500GiB ♦ mounted as either /opt/phantom/data/db or <PHANTOM_HOME>/data/db • Embedded Splunk Enterprise: 500GiB ♦ mounted as /opt/phantom/data/splunk or <PHANTOM_HOME>/data/splunk • File share volumes: 500GiB ♦ mounted as /opt/phantom/vault or <PHANTOM_HOME>/vault Disk space requirements vary depending on the volume of data ingested and the size of your production environment.
Network	A one-gigabit network interface
System utilities	• cron ♦ On unprivileged deployments, the user account that runs Splunk SOAR (On-premises) must have permission to create cron jobs. • ntp or chrony

If you use the Files feature to store virtual machine snapshots or other large-format data, it is recommended you use a larger volume for storage.

Splunk SOAR (On-premises) ports and endpoints

These tables list the ports which must be open to inbound traffic and internet endpoints which must be accessible to use Splunk SOAR (On-premises). Use these tables to design the firewall rules for your deployment.

Endpoints for all Splunk SOAR (On-premises) deployments

This table shows a list of the internet endpoints that a Splunk SOAR (On-premises) deployment uses. This list is not exhaustive.

Endpoint	Required?	Description
splunkbase.splunk.com	Required	Required for app installation and app upgrades.
Splunk Cloud	Conditional	If your deployment uses a Splunk Cloud deployment instead of the embedded Splunk Enterprise instance, Splunk SOAR (On-premises) must be able to reach your Splunk Cloud deployment.
grpc.prod1-cloudgateway.spl.mobi	Conditional	If you use Splunk Mobile to access Splunk SOAR (On-premises) on mobile devices, your Splunk SOAR (On-premises) deployment must be able to reach grpc.prod1-cloudgateway.spl.mobi
https://e1345286.api.splkmobile.com/1.0/e1345286	Required	Splunk SOAR (On-premises) telemetry
*.pool.ntp.org	Required	Used for system clock synchronization.
CentOS and RHEL mirrors	Required	Required to run YUM updates for operating system components and installed software packages. If your organization prefers, you can use a satellite server instead. See the Red Hat Knowledgebase article https://access.redhat.com/solutions/29269.
github.com	Required	Used to access the community playbook repository.
Other source control system	Conditional	Access is required if your deployment uses an alternative repository for playbooks.
Google Maps embed API	Required	Used by the MaxMind app to add visualizations for IP address geolocation results.
pypi.org	Required	Used by some apps to update or install their PIP dependencies.
App specific endpoints	Conditional	Apps might need to reach specific endpoints in order to provide their functions. Consult the app's documentation for details.

Ports for a standalone Splunk SOAR (On-premises) deployment

On a single instance deployment of Splunk SOAR (On-premises) where all services are contained on the same host, open these ports in addition to allowing the Endpoints for all Splunk SOAR (On-premises) deployments.

Port	Required?	Description
TCP 22	Required	SSH port. Used for administering the operating system.
TCP 80	Required	Port for requests sent over HTTP. Splunk SOAR (On-premises) redirects all HTTP requests to HTTPS.
TCP 443	Required	HTTPS port for the web interface and REST API. This port must be exposed to access Splunk SOAR (On-premises) services. In an unprivileged deployment the HTTPS port is specified during installation and is a port greater than 1023. In an AMI-based deployment, the HTTPS port is set to 9999.
TCP 8443	Required, Configurable	HTTPS port for the web interface and REST API. This port must be exposed to access Splunk SOAR (On-premises) services. In an unprivileged deployment the HTTPS port is specified during installation and is a port greater than 1023. During upgrades, Splunk SOAR (On-premises) will set firewalld rules to forward TCP 443 to TCP 8443.

Ports for externalized services

If you opt to deploy services such as Splunk Enterprise or Splunk Cloud, PostgreSQL, or a file share separately from your Splunk SOAR (On-premises) deployment, you need to make sure that Splunk SOAR (On-premises) can reach those services on your network.

In a clustered deployment, all services are external to Splunk SOAR (On-premises), and an added load balancer. See Example: Splunk SOAR (On-premises) cluster for a diagram of a Splunk SOAR (On-premises) cluster.

Required ports for embedded Splunk Enterprise

Open these ports on each Splunk SOAR (On-premises) node for embedded Splunk cluster configuration.

Port	Purpose
TCP 5121	Splunk Enterprise server HTTP Event Collector (HEC) service. Can be blocked on the Shared Services server if using an alternate Splunk Enterprise server.
TCP 5122	Splunk Enterprise server REST port. Can be blocked on the Shared Services server if using an alternate Splunk Enterprise server.

Required ports for non-embedded Splunk Enterprise

If you are using the non-embedded version of Splunk Enterprise, open these ports on each Splunk SOAR (On-premises) node.

Port	Purpose
TCP 8088	Used as the HTTP Event Collecter (HEC) and provides searching capabilities.
TCP 8089	Used for the REST endpoint to send information to the Splunk instances.
TCP 9996-9997	Used for the universal forwarder to either forward or direct the indexers.

PostgreSQL database

A single instance deployment of Splunk SOAR (On-premises) uses a local instance of a PostgreSQL database. If you choose to use an external PostgreSQL database instead, you must make sure that Splunk SOAR (On-premises) can reach the database on your network.

In a clustered Splunk SOAR (On-premises) deployment, each Splunk SOAR (On-premises) node must be able to reach the PostgreSQL database. See About Splunk SOAR (On-premises) clusters in *Install and Upgrade Splunk SOAR (On-premises)*.

Port	Description
TCP 5432	PostgreSQL service. This port is also used by warm standby configurations for PostgreSQL streaming replication.
TCP 6432	Used by PgBouncer to interact with the PostgreSQL database.

File Shares

A single instance deployment of Splunk SOAR (On-premises) uses the local file system to store files for the vault. You can choose to expand storage capacity by using an external file share.

You can use any file system that meets your organization's security and performance requirements for your external file shares. You need to configure any required mounts and permissions. See Supported file systems and required

directories.

These following tables uses NFS and GlusterFS as an example for file shares. In a clustered Splunk SOAR (On-premises) deployment, these ports must be opened on each Splunk SOAR (On-premises) node, and in the case of GlusterFS, on each member of the GlusterFS server cluster.

Port	Description
TCP 445	CIFS protocol.
UDP 111	RPC portmapper service for GlusterFS and NFS.
TCP 111	RPC portmapper service for GlusterFS and NFS.
TCP 2049	GlusterFS and NFS for NFS exports. Used by the nfsd process.
TCP 38465	NFS mount protocol.
TCP 38466	NFS mount protocol.
TCP 38468	NFS Lock Manager, NLM.
TCP 38469	NFS ACL support.
TCP 24007	glusterd management port.
TCP 24008	glusterd management port.
TCP 49152+	For GlusterFS brick mounts. The total number of ports required to be open depends on the total number of bricks exported on the server. In most cases, 10 bricks is sufficient. You might need to open additional ports later if you add additional bricks.

Ports for connecting mobile devices to Splunk SOAR (On-premises) using Splunk Connected Experience apps

Open these ports to enable registration of mobile apps, such as Splunk Mobile for iOS or Splunk Mobile for Android. In a clustered deployment, these ports must be opened on each Splunk SOAR (On-premises) node.

When the **Enable Mobile App** toggle is in the ON position, Splunk SOAR (On-premises) launches a new daemon, ProxyD. ProxyD connects to the Splunk Cloud Gateway automatically at grpc.prod1-cloudgateway.spl.mobi, on port 443 using the gRPC protocol.

Splunk SOAR (On-premises) uses the gRPC protocol to communicate to mobile apps through the Splunk Cloud Gateway.

For more information on Splunk Cloud Gateway, its encryption, and the data that is sent and received, see About the Splunk Cloud Gateway security process in *Install and Administer Splunk Cloud Gateway*.

Port	Description
TCP 15505	Port 15505 is used by ProxyD to listen for inter-process communication from other Splunk SOAR (On-premises) daemons on the same instance.
TCP 443	Port 443 is the inbound port to Splunk SOAR (On-premises)'s REST endpoints from ProxyD. REST requests from connected mobile devices received from Splunk Cloud Gateway are sent to and received from other Splunk SOAR (On-premises) daemons by ProxyD on port 443.

Port	Description

For other ports you might need to open, see Prerequisites and Requirements in the *Install and Administer Splunk Cloud Gateway* manual.

Ports for clustered deployments of Splunk SOAR (On-premises)

Splunk SOAR (On-premises) can be deployed as a cluster of nodes connected to a server or set of servers providing a PostgreSQL database, file shares, a Splunk platform deployment, and a load balancer. A cluster can be deployed on-premises or in Amazon Web Services. See About Splunk SOAR (On-premises) clusters in *Install and Upgrade Splunk SOAR (On-premises)*.

This table lists the ports required by Splunk SOAR (On-premises) nodes for inter-node communication and access to Splunk SOAR (On-premises) services on the cluster.

Port	Description
TCP 22	SSH port. Used for administering the operating system of the cluster node. Also used by SSHD for GlusterFS.
TCP 80	Port for requests sent over HTTP. Splunk SOAR (On-premises) redirects all HTTP requests to HTTPS.
TCP 443	HTTPS interface for the web interface, load balancer, and the REST API. This port must be exposed to access Splunk SOAR (On-premises) services. In an unprivileged deployment, the HTTPS port is specified during installation and is a port greater than 1023. In an AMI-based deployment, the HTTPS port is set to 9999.

TCP 8443HTTPS interface for the web interface, load balancer, and the REST API. This port must be exposed to access Splunk SOAR (On-premises) services.

In an unprivileged deployment, the HTTPS port is specified during installation and is a port greater than 1023. In an AMI-based deployment, the HTTPS port is set to 9999.

TCP 4369RabbitMQ port mapper. All cluster nodes must be able to communicate with each other on this port.TCP 5100 - TCP 5120Daemon inter-process communication ports.TCP 5671RabbitMQ service. All cluster nodes must be able to communicate with each other on this port.TCP 8300Consul RPC services. All cluster nodes must be able to communicate with each other on this port.TCP 8301Consul internode communication. All cluster nodes must be able to communicate with each other on this port.TCP 8302Consul internode communication. All cluster nodes must be able to communicate with each other on this port.TCP 8888WebSocket server.TCP 15672RabbitMQ admin UI and HTTP API service.

The RabbitMQ admin UI is disabled by default. Unless you want to use the admin UI, you can block this port. If you choose to activate the RabbitMQ HTTP API and web UI, all cluster nodes must be able to communicate with each other on this port.

TCP 25672RabbitMQ internode communications. All cluster nodes must be able to communicate with each other on this port.

For information on RabbitMQ ports, see "Networking" on the RabbitMQ documentation. For more information on Consul's required ports, see "Ports" in the Consul documentation on the HashiCorp website.

Example: Default firewalld settings for an unprivileged cluster

Here is an example of the default settings for firewalld when Splunk SOAR (On-premises) is deployed as an unprivileged cluster. Splunk Connected experiences apps access is not enabled in this example.

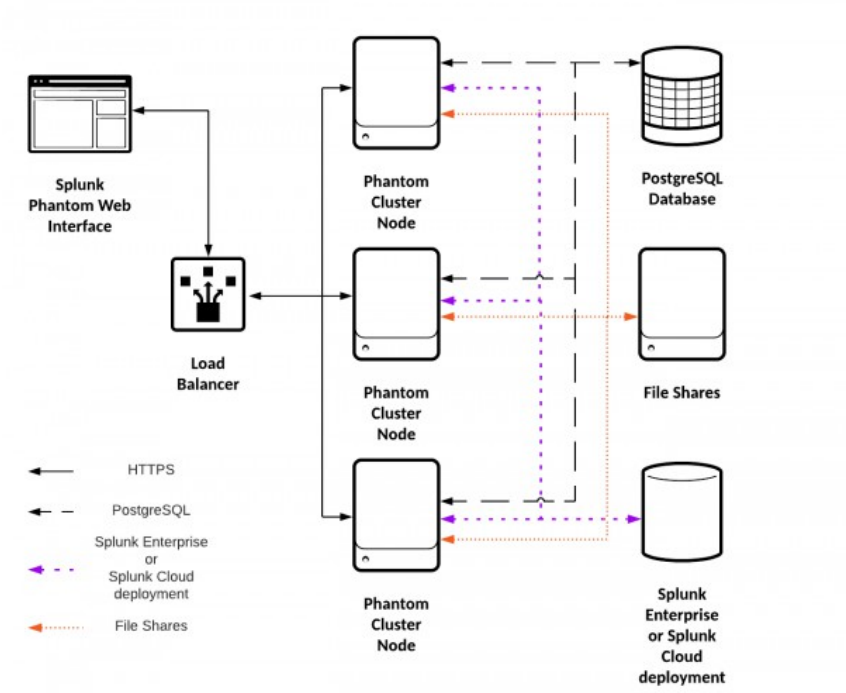
```
[phantom@phantom ~]$ sudo firewall-cmd --list-all
```

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: dhcpv6-client http https ssh
  ports: 9999/tcp 27100-27200/tcp 5121/tcp 5122/tcp 8300/tcp 8301/tcp 8302/tcp 4369/tcp 5671/tcp 25672/tcp
15672/tcp 443/tcp
  protocols:
  masquerade: no
  forward-ports: port=443:proto=tcp:toport=9999:toaddr=
  source-ports:
  icmp-blocks:
  rich rules:
```

Example: Splunk SOAR (On-premises) cluster

A Splunk SOAR (On-premises) cluster consists of a load balancer, three or more Splunk SOAR (On-premises) nodes, a PostgreSQL database, file shares, and either a Splunk Enterprise or Splunk Cloud deployment.

This diagram shows an example of a Splunk SOAR (On-premises) cluster, with the connections marked.



FIPS compliance

With the 5.2.1 and higher releases, Splunk SOAR (On-premises) can be deployed in a Federal Information Processing Standard (FIPS) compliant mode.

In order for a security application such as Splunk SOAR (On-premises) to be considered FIPS compliant it must meet the standards specified by the National Institute of Standards and Technology (NIST) in the standard FIPS 140-2.

Splunk SOAR (On-premises) differences for FIPS

When deployed in FIPS compliant mode, there are differences in Splunk SOAR (On-premises) 5.2.1 from earlier releases.

- Support for MD5 hashing is disabled.
- Python 2 support is disabled.
- Splunk SOAR (On-premises) in FIPS compliant mode uses OpenSSL 1.1
- Splunk SOAR (On-premises) uses a FIPS compliant version of Python 3 which does not support disallowed hashing methods.

Prerequisites for deploying Splunk SOAR (On-premises) in FIPS compliant mode

If you need to adhere to the FIPS standard, you must prepare your environment for FIPS compliance before deploying Splunk SOAR (On-premises).

Operating System

You must use a supported operating system in FIPS mode:

- Red Hat Enterprise Linux 7.6 through 7.9
- CentOS 7.6 through 7.9

Information about setting up RHEL 7.x or CentOS 7.x in Federal Information Processing Standard (FIPS) mode can be found on the Red Hat Security Guide in Chapter 9.

Clustering and external services

When you deploy either a cluster or a Splunk SOAR (On-premises) instance with external services:

- Each cluster node or external service must be deployed on a FIPS compliant operating system.
- Each external service, such as PostgreSQL, Splunk Enterprise, your load balancer, and file share file system must be in FIPS compliant mode.

Limitations

Deploying Splunk SOAR (On-premises) in FIPS compliant mode has the following limitations:

- Only new deployments can be created. Upgrades from non-FIPS deployments to FIPS deployments is not possible.
- Only unprivileged deployments are supported.
- You can not disable FIPS mode. Once deployed in FIPS compliant mode, the choice cannot be undone nor can the deployment be downgraded to a non-FIPS mode.

Apps

Not all apps have been validated for FIPS compliance.

When you attempt to install a new app, or configure an asset for an installed app that is not validated as FIPS compliant, a warning message will be displayed. You may still install apps, but their actions may fail for FIPS related constraints such as disallowed TLS certificate signing or hashing algorithms, or unsupported Python versions.

Updated apps are released on Splunkbase and the Phantom Community Portal. You can always check to see if an app has been updated for FIPS compliance.

How to determine if Splunk SOAR (On-premises) is in FIPS compliant mode

In order to determine if your Splunk SOAR (On-premises) deployment is in FIPS compliant mode, you can either check the user interface, or use a REST API.

Check FIPS compliant status in the user interface

Use the user interface to check FIPS status.

1. From the **Home** menu, select **Administration**.
2. Select **About**.

If the deployment is in FIPS compliant mode, the **FIPS enabled** line will read "Yes".

Check FIPS compliant status with the REST API

Use the REST API to determine whether or not a deployment is in FIPS compliant mode.

Send query using the `/rest/system_settings?sections=["fips"]` API. The response is a JSON body of the `["fips"]` section of the system settings. If the `"enabled"` is true, then FIPS compliant mode is enabled.

```
{
    "fips": {
        "enabled": true
    }
}
```

Install Splunk SOAR (On-premises)

Install Splunk SOAR (On-premises) using the Amazon Marketplace Image

Install Splunk SOAR (On-premises) for AWS from the AWS Marketplace in the security category.

The AMI version of Splunk SOAR (On-premises) is for an unprivileged installation, meaning the the application runs under the phantom user account, and not as root.

- The base installation directory for the unprivileged AMI is /opt/phantom/.
- The custom HTTPS port is 9999, but the Splunk SOAR (On-premises) web interface is still available on port 443.

Prerequisites

Your AWS instance must meet or exceed the requirements for either an evaluation system for evaluation or Proof of Value testing, or a production system for production use, and must include:

- A supported operating system. See Supported operating systems.
- Sufficient storage. See System requirements for production use.

If you need to connect your organization's on-premises infrastructure to an installation of Splunk SOAR (On-premises) hosted in AWS, consult the article Connect Your Data Center to AWS on the AWS web site.

Federal Information Processing Standard (FIPS) support

Splunk SOAR (On-premises) can be deployed in a FIPS compliant mode, if the operating system kernel is in FIPS mode.

- Your operating system, either RHEL or CentOS must be in FIPS mode.
- You must create a new, unprivileged deployment of Splunk SOAR (On-premises), either as a single instance or as a cluster.

Information about setting up RHEL 7.x or CentOS 7.x in Federal Information Processing Standard (FIPS) mode can be found in the Red Hat Security Guide in Chapter 9.

Installation

Perform the following tasks to install Splunk SOAR (On-premises):

1. Log in to your AWS EC2 account.
2. From your EC2 dashboard, select **Launch Instance**.
3. In the AWS Marketplace, search for Splunk SOAR (On-premises).
4. On the Amazon Machine Image entry, click **Select**.
5. Click **Continue**.
6. Select an instance size. The default is m5.xlarge. Splunk SOAR (On-premises) does not support using instances smaller than t2.xlarge.
7. Click **Next: Configure Instance Details**.
8. Configure the instance according to your organization's policies.
9. Click **Next: Add Storage**.

10. Add storage.

    You can increase disk size later, but you cannot decrease disk size.
11. Click **Next: Add Tags**.
12. Add tags to help identify your Splunk SOAR (On-premises) installation in your EC2 dashboard.
13. Click **Next: Configure Security Group**.
14. Configure Security Groups. By default, SSH, HTTP, and HTTPS are permitted from all IP addresses. Increase security by limiting access to your organization's IP addresses.
15. Click **Review and Launch**.
16. Generate or choose SSH keys.

    The SSH user account is phantom. This user account has sudo access for elevating to root.
17. Click **Launch Instances**. The installation typically takes 15 minutes to complete.

Next step: log in to verify the installation

You can log in to the Splunk SOAR (On-premises) web interface after the setup script completes to configure user accounts and additional settings. See Log in to the Splunk SOAR (On-premises) web interface.

Install Splunk SOAR (On-premises) as a privileged user

With the 5.4.0 release of Splunk SOAR (On-premises), you cannot install Splunk SOAR (On-premises) as a privileged user.

You can only install Splunk SOAR (On-premises) as an unprivileged user. See Install Splunk SOAR (On-premises) as an unprivileged user.

Install Splunk SOAR (On-premises) as an unprivileged user

TAR file distributions of Splunk SOAR (On-premises) are available for installations where Splunk SOAR (On-premises) runs as an unprivileged user.

As of version 5.3.0, RPM files are no longer available for Splunk SOAR (On-premises) installations. Follow the updated instructions for privileged and unprivileged installations. Unique tarballs are available for privileged and unprivileged installations.

If you install a stand-alone instance as an unprivileged user, underlying services such as the PostgreSQL database are installed in the user space for that user.

Prerequisites

The following operating systems are supported.

- Red Hat Enterprise Linux 7.6 through 7.9.
- CentOS 7.6 through 7.9.

Federal Information Processing Standard (FIPS) support

Splunk SOAR (On-premises) can be deployed in a FIPS compliant mode, if the operating system kernel is in FIPS mode.

- Your operating system, either RHEL or CentOS must be in FIPS mode.
- You must create a new, unprivileged deployment of Splunk SOAR (On-premises), either as a single instance or as a cluster.

To determine whether your operating system kernel is in FIPS mode, run the following command.

cat /proc/sys/crypto/fips_enabled
If that command returns a 1, the kernel is in FIPS mode. If that command returns a 0, the kernel is not in FIPS mode.

Information about setting up RHEL 7.x or CentOS 7.x in Federal Information Processing Standard (FIPS) mode can be found in the Red Hat Security Guide in Chapter 9.

Install Splunk SOAR (On-premises) from the TAR file

1. On the machine where you want to install Splunk SOAR (On-premises), make sure the operating system is updated.
   sudo yum clean all
   sudo yum update
2. If the machine where you want to install Splunk SOAR (On-premises) required kernel updates, reboot the system before continuing with the installation.
3. Download the unprivileged installer from the Splunk Phantom community website Product Downloads page.
4. If you downloaded the installer onto a local machine and need to copy it to the machine where you want to install Splunk SOAR (On-premises), you can use the following command.
   scp -r ./splunk_soar-unpriv-<version>.tgz <user>@<installation_address>
5. Log in as a user with root privileges to the machine where you want to install Splunk SOAR (On-premises).
6. Extract the Splunk SOAR (On-premises) TAR file.
   tar -xzvf ./splunk_soar-unpriv-<version>.tgz
7. To prepare the system for the unprivileged installation, run a pre-install script using the following command: `sudo ./soar-prepare-system --splunk-soar-home <home_directory> --https-port <port_number>`. The arguments for the command are optional. If left undefined, the --splunk-soar-home argument defaults to the directory that contains the installation script and specifies the home directory for Splunk SOAR (On-premises). That directory must exist and the user meant to run the installation must own that directory. The --https-port argument defaults to port 8443. When you run the pre-install script, it prompts you to configure the system.

   If a configuration requirements is already satisfied in your system, that prompt might not appear.

- `Install pre-requisite RPM packages required by Splunk SOAR (Y/n)`: If prompted, you must answer Y to proceed.
- `GlusterFS is only needed if you are using an external file share. This is common if you're constructing a Splunk SOAR cluster. Do you want to run this step? (Y/n)`: You only need to answer Y if you are setting up certain cluster configurations of Splunk SOAR (On-premises), but you can answer Y even on individual instances.
- `Enable the ntpd service to guarantee clock synchronization. Do you want to run this step? (Y/n)`: Answer Y.
- `Create a non-privileged user for running Splunk SOAR (On-premises). (Y/n)`: If prompted, you must answer Y to proceed.
- `Do you want to set a password for <non-privileged_user> now? (Y/n)`: Answer Y if you created a non-privileged user for running Splunk SOAR (On-premises) in the previous step.

- `Set system resource limits for Splunk SOAR user, particularly file descriptor limits, which are low by default. (Y/n):`

Answer Y.

- If the --splunk-soar-home location differs from the location where you extracted the Splunk SOAR (On-premises) TAR file, follow these steps to move it to the --splunk-soar-home location and then extract it there:

  - Copy the TAR file to the --splunk-soar-home location.
    sudo cp ./splunk_soar-unpriv-<version>.tgz <home_directory>
  - Go to the --splunk-soar-home location.
    cd <home_directory>
  - Log in as the user meant to own the Splunk SOAR (On-premises) installation.
  - Extract the Splunk SOAR (On-premises) TAR file.
    tar -xzvf ./splunk_soar-unpriv-<version>.tgz

- Ensure you are logged in as the user meant to own the Splunk SOAR (On-premises) installation. Do not perform the installation command as the root user.
- Run the soar-install installation script with the same arguments you included in the soar-prepare-system script.
  ./soar-install --splunk-soar-home <home_directory> --https-port <port_number>
- The soar-install installation script displays the installation and path and HTTPS port number, then asks `Do you want to proceed? (y/N)`. If the path and port are correct, answer y.

The --https-port argument specifies what port Splunk SOAR (On-premises) webserver uses to expose the web user interface. If you ran the soar-prepare-system script to forward inbound traffic to port 443, the user interface is visible there, too.

- The soar-install installation script displays the installation and path and HTTPS port number, then asks `Do you want to proceed? (y/N)`. If the path and port are correct, answer y.

Run the sudo ./soar-prepare-system --help and sudo ./soar-install --help commands to see what optional arguments are available.

Log in to the Splunk SOAR (On-premises) web interface

Perform the following tasks to log in to the Splunk SOAR (On-premises) web interface after installation is complete.

1. Using a web browser, go to the IP address you assigned to Splunk SOAR (On-premises).
   - If you installed Splunk SOAR (On-premises) as an unprivileged user, log in to Splunk SOAR (On-premises)'s web interface at the custom HTTPS port. `https://<ip address or hostname>:<your https port>`

     The custom HTTPS port for unprivileged AMI based installations is 9999. However, the UI is still accessible on port 443.

   - If you installed Splunk SOAR (On-premises) from the AWS Marketplace, get the public IP address for the instance from the EC2 Management Console and the full AWS instance ID for the EC2 instance. Log in to the Splunk SOAR (On-premises) web interface by using the public IP address.
2. Log in using the default credentials. Use **admin** as the username and **password** as the password. If you installed Splunk SOAR (On-premises) from the AWS Marketplace, use **admin** as the username and the full AWS instance ID as the password.

AWS requires about 20 minutes to complete the setup. After the setup is complete, you can use your full AWS instance ID as the password. Attempting to login before that will result in a login error.

3. Change the **admin** user's password:
    1. Click the user name **admin**, then select **Account Settings**.
    2. Click the **Change Password** tab.
    3. Type the current password.
    4. Type a new password.
    5. Type a new password a second time to confirm.
    6. Click **Change Password**.

Log in to Splunk SOAR (On-premises) using SSH

To SSH into the Splunk SOAR (On-premises) instance perform the following steps:

1. Open a terminal window.
2. SSH to your Splunk SOAR (On-premises) instance's operating system `ssh phantom@<hostname or IP address of Splunk SOAR (On-premises)>`.

Remote SSH is disabled for the root user. The accounts user and phantom have sudo permissions. You can use the account user to administer the operating system.

In order to log in to the operating system of your AMI-based installation using SSH, use the user id phantom. If you need root access, use sudo su -.

Clustering

About Splunk SOAR (On-premises) clusters

Splunk SOAR (On-premises) supports clustering.

A cluster consists of a minimum of three instances of Splunk SOAR (On-premises) and its supporting external services; file shares, a PostgreSQL database or database cluster, Splunk Enterprise, and at least one load balancer, such as HAProxy.

Splunk SOAR (On-premises) clustering uses additional technologies to support the cluster:

- GlusterFS for file shares. Other file systems, such as NFS can be used instead of GlusterFS.
- Consul to provide action locking as needed.
- RabbitMQ to provide a fast, reliable messaging bus.
- HAProxy as a load balancer. Alternate load balancers can be used instead of HAProxy.

In a cluster, both the PostgreSQL database and the deployment of Splunk Enterprise are externalized from the Splunk SOAR (On-premises) instances. This allows you to scale your database and Splunk Enterprise deployments separately from the Splunk SOAR (On-premises) nodes.

Before creating a cluster, work with your Splunk SOAR (On-premises) delivery team representative to assess your needs and design your cluster.

Why build a Splunk SOAR (On-premises) Cluster?

Clustering addresses several important needs:

- Clustering adds horizontal scaling for Splunk SOAR (On-premises) workloads, allowing for increased capacity.
- Clustering adds redundancy for the Splunk SOAR (On-premises) platform. One or more cluster nodes can fail and you still have a functioning deployment of Splunk SOAR (On-premises).
- Clustering removes system downtime for upgrades or maintenance. You can upgrade individual Splunk SOAR (On-premises) cluster nodes without taking the entire deployment offline.

Building a Splunk SOAR (On-premises) cluster

Clusters can be built from unprivileged installations, where required services are provided by servers external to Splunk SOAR (On-premises). Each Splunk SOAR (On-premises) node is converted from a TAR file installation using the `make_cluster_node.pyc` script. See Create a Splunk SOAR (On-premises) cluster using an unprivileged installation.

Create a Splunk SOAR (On-premises) cluster using a privileged installation

With the 5.4.0 release of Splunk SOAR (On-premises), you cannot install Splunk SOAR (On-premises) as a privileged user.

To build an unprivileged Splunk SOAR (On-premises) cluster, see Create a Splunk SOAR (On-premises) cluster using an unprivileged installation.

Create a Splunk SOAR (On-premises) cluster using an unprivileged installation

Build a cluster, putting each of the services on its own server or group of servers to serve multiple cluster nodes of Splunk SOAR (On-premises).

Set up each of the external services either as the root user or a user with sudo permissions.

Install Splunk SOAR (On-premises) as an unprivileged user. In your cluster, each Splunk SOAR (On-premises) instance must have the same custom username and install directory. See Install Splunk SOAR (On-premises) as an unprivileged user.

Number	Task	Description
1	Create the HAProxy node.	Use the HAProxy server to be a load balancer for the Splunk SOAR (On-premises) nodes in your cluster. See Set up a load balancer with an HAProxy server. There are additional steps to configure your load balancer to handle your custom HTTPS port for unprivileged clusters.
2	Install Splunk SOAR (On-premises) using the tar file method for unprivileged installs.	Do this once for each node you need in your cluster. See Install Splunk SOAR (On-premises) as an unprivileged user.
3	Create the PostgreSQL node.	Establish a PostgreSQL database server or cluster to store Splunk SOAR (On-premises) information. See Set up the external PostreSQL server.
4	Create the file shares node.	Splunk SOAR (On-premises) stores all its shared files on the prepared GlusterFS server. You can use NFS or other network file system. Instructions for that are not included in this document. See Set up external file shares using GlusterFS.
5	Create the Splunk Enterprise node.	Splunk SOAR (On-premises) uses Splunk Enterprise for searches and collect data for indexing using the HTTP Event Collector. See Set up Splunk Enterprise.
6	Prepare Splunk SOAR (On-premises) instances to connect to the GlusterFS file share.	See Prepare an unprivileged Splunk SOAR (On-premises) instance to connect to the GlusterFS file share This task must be completed by a user with root or sudo access.

7Convert Splunk SOAR (On-premises) instances to cluster nodes.Convert the first instance into a cluster node by running `make_cluster_node.pyc`. See Run make_cluster_node.pyc. Repeat on each Splunk SOAR (On-premises) instance that will become a cluster node.

Prepare an unprivileged Splunk SOAR (On-premises) instance to connect to the GlusterFS file share

Before you can convert an unprivileged Splunk SOAR (On-premises) instance into a node in an unprivileged cluster, you must prepare each of the Splunk SOAR (On-premises) instances to connect to the GlusterFS file share.

Do these steps as a user with root or sudo access to the Splunk SOAR (On-premises) instance.

1. Install the GlusterFS client on each Splunk SOAR (On-premises) instance.
   yum install glusterfs-fuse -y
2. Add the required TLS keys for the Gluster FS server and the GlusterFS directory and control file to each Splunk SOAR (On-premises) instance. See Configure Splunk SOAR (On-premises) cluster nodes to connect to the GlusterFS file shares in Set up external file shares using GlusterFS.
3. Edit the cluster member's file system table, `/etc/fstab`, to mount the GlusterFS volumes. Your `fstab` entries must not have line breaks.

```
<glusterfs_hostname>:/apps /<phantom_install_dir>/apps glusterfs defaults,_netdev 0 0
<glusterfs_hostname>:/app_states /<phantom_install_dir>/local_data/app_states glusterfs
defaults,_netdev 0 0
<glusterfs_hostname>:/scm /<phantom_install_dir>/scm glusterfs defaults,_netdev 0 0
<glusterfs_hostname>:/tmp /<phantom_install_dir>/tmp/shared glusterfs defaults,_netdev 0 0
<glusterfs_hostname>:/vault /<phantom_install_dir>/vault glusterfs defaults,_netdev 0 0
```

4. Mount all the volumes to make them available.

```
mount /<phantom_install_dir>/apps
mount /<phantom_install_dir>/local_data/app_states
mount /<phantom_install_dir>/scm
mount /<phantom_install_dir>/tmp/shared
mount /<phantom_install_dir>/vault
```

5. Start Splunk SOAR (On-premises) services on all cluster nodes.
   <$PHANTOM_HOME>/bin/start_phantom.sh

Create a Splunk SOAR (On-premises) cluster in Amazon Web Services

Build a cluster from AMI-based instances of Splunk SOAR (On-premises), building several of the required services using AWS native components: Elastic Load Balancer (ELB), Elastic File System (EFS), and Relational Database System (RDS).

This configuration is built using the Amazon Marketplace Image of Splunk SOAR (On-premises). This release is an unprivileged version of Splunk SOAR (On-premises) which runs under the user account phantom.

Converting an AMI-based installation to a server or cluster node is a one-way operation. It cannot be reverted.

Build a cluster with AWS services

Number	Task	Description
1	Launch and prepare AMI instances of Splunk SOAR (On-premises).	Total number of Splunk SOAR (On-premises) AMI instances = Number of cluster nodes + 1 See Launch and prepare AMI instances of Splunk SOAR (On-premises).
2	Create a load balancer with Elastic Load Balancer (ELB).	See Create a load balancer with Elastic Load Balancer (ELB). 1. Create the ELB. 2. Create the Target Group. 3. Add routing rules.
3	Create the file stores with Elastic File System (EFS).	Create the EFS file store for shared files. See Create the file stores with Elastic File System (EFS).
4	Create the external database with Relational Database System (RDS).	See Create the external database with Relational Database System (RDS). 1. Create the external PostgreSQL database. 2. Create the pgbouncer user.
5	Add the file shares to each Splunk SOAR (On-premises) instance.	Mount the file shares on each Splunk SOAR (On-premises) instance. See Add the file shares to each Splunk SOAR (On-premises) instance.

Number	Task	Description
6	Convert an AMI-based Splunk SOAR (On-premises) Instance into the Splunk Enterprise instance.	Convert one of the Splunk SOAR (On-premises) instances into the Splunk Enterprise instance. This instance will serve as the external search endpoint for the entire cluster. Use the make_server_node.pyc script with the splunk argument. See Convert an AMI-based Splunk SOAR (On-premises) Instance into the Splunk Enterprise instance.
7	Convert the first AMI-based Splunk SOAR (On-premises) instance into a cluster node.	Convert the first Splunk SOAR (On-premises) instance into a cluster node. Creating the first node will use a script option to record all the make_cluster_node.pyc script answers to a file for use on each of your other nodes. See Convert the first AMI-based Splunk SOAR (On-premises) instance into a cluster node.
8	Convert the remaining AMI-based Splunk SOAR (On-premises) instances into cluster nodes.	Convert the remaining Splunk SOAR (On-premises) instances into cluster nodes using make_cluster_node.pyc and the mcn_responses.json file. See Convert the remaining AMI-based Splunk SOAR (On-premises) instances into cluster nodes.

Launch and prepare AMI-based instances of Splunk SOAR (On-premises)

You need a number of AMI-based Splunk SOAR (On-premises) instances equal to the number of Splunk SOAR (On-premises) nodes you want in your cluster plus one. The additional instance will be converted into the externalized Splunk Enterprise instance for your cluster. A Splunk SOAR (On-premises) cluster requires a minimum of three nodes.

Total number of Splunk SOAR (On-premises) AMI instances = Number of cluster nodes + 1

If you already have a Splunk Enterprise deployment that you will use instead, follow the instructions for using an external Splunk Enterprise instance. See Set up Splunk Enterprise.

Installation

1. Log in to your AWS EC2 account.
2. From your EC2 dashboard, select **Launch Instance**.
3. In the AWS Marketplace, search for Splunk SOAR (On-premises).
4. On the Amazon Machine Image entry, click the button **Select**.
5. Click **Continue**.
6. Select an instance size. The default is **m5.xlarge**. Splunk SOAR (On-premises) does not support using instances smaller than **t2.xlarge**.
7. Click **Next: Configure Instance Details**.
8. For **Number of Instances**, type the number of instances you need. Total number of Splunk SOAR (On-premises) AMI instances = Number of cluster nodes + 1
9. Configure the instance according to your organization's policies. See Splunk SOAR (On-premises) required ports for more information.

   Because this is an unprivileged version of Splunk SOAR (On-premises), you will need to be sure to open the custom HTTPS port 9999 for your instances.
10. Click **Next: Add Storage**.
11. Add storage.

    You can increase disk size later, but you cannot decrease disk size.

12. Click **Next: Add Tags**.
13. Add tags to help identify your Splunk SOAR (On-premises) installation in your EC2 dashboard.
14. Click **Next: Configure Security Group**.
15. Configure Security Groups. By default, SSH, HTTP, and HTTPS are permitted from all IP addresses. Increase security by limiting access to your organization's IP addresses.
16. Click **Review and Launch**.
17. Generate or choose SSH keys.
18. Click **Launch Instances**. The installation typically takes 15 minutes to complete.

In order to log in to the operating system of your AMI-based Splunk SOAR (On-premises) install using SSH, use the user account phantom. If you need root access, use sudo su - to elevate to root.

Install SSH keys

During the conversion to Splunk SOAR (On-premises) cluster nodes, each instance will need to SSH as the phantom user into other nodes. Install the client certificate you generated for SSH when the instances were created.

Do this on each of the instances that you will convert to cluster nodes.

1. Copy the .pem file generated earlier to each instance using SCP.
   scp -i <path/to/.pem> <path/to/.pem to transfer> phantom@<instance IP or DNS name>:~/
2. SSH to an AMI-based Splunk SOAR (On-premises) instance as the phantom user.
3. Move the .pem key to the phantom user's .ssh directory.
   mv <name of file>.pem .ssh
4. Set the permissions on the .pem key.
   chmod 600 .ssh/<name of file>.pem
5. Test that you are able to SSH from each instance to the others as the phantom user.

Create a load balancer with Elastic Load Balancer (ELB)

Create a load balancer for your Splunk SOAR (On-premises) cluster. An Elastic Load Balancer will be used instead of HAProxy.

1. Log in to your AWS EC2 account.
2. From the menu on the EC2 dashboard, under the heading **Load Balancing**, choose **Load Balancers**.
3. Click **Create Load Balancer**.
4. Under **Application Load Balancer**, click **Create**.
5. Type a name for your load balancer in the **Name** field.
6. Select a **Scheme**. The scheme will depend on your AWS network configuration. Assuming your load balancer will route on an internal network, select the **internal** radio button.
7. Set the IP address type. This will also depend on your AWS network configuration. In most cases, select **ipv4** from the menu.
8. Under **Listeners, Load Balancer Protocol**, select **HTTPS** from the menu. The **Load Balancer Port** changes to 443.
9. Under Availability Zones, select the VPC and Availability Zones to match your AWS network configuration.
10. Add **Tags** to help organize and identify your load balancer.
11. Click **Next: Configure Security Settings**.
12. Select or create a security group according to your organization's policies. These settings can vary based on factors outside the scope of this document.
13. Click **Next: Configure Routing**.

14. Under **Target group**, choose **New target group** from the menu.
15. Type a name for your target group in the **Name** field.
16. For **Target type**, select the **Instance** radio button.
17. For **Protocol**, select **HTTPS** from the menu. **Port** changes to **443** automatically.

    You must also open Splunk SOAR (On-premises)'s custom HTTPS port 9999 to allow HTTPS traffic for unprivileged processes.

18. Under **Health checks**, set **Protocol** to **HTTPS**.

    Health checks will fail until you have run the make_cluster_node scripts to add your Splunk SOAR (On-premises) instances to your cluster. This is normal and expected.

19. In the **Path** field, type /**check**.
20. Click **Next: Register Targets**.
21. Under **Instances**, find and select the cluster node instances for your Splunk SOAR (On-premises) cluster. You do not need to load balance the external services, such as PostgreSQL, file shares, or Splunk Enterprise.
22. Click **Add to registered**.
23. Click **Next: Review**.
24. Review for and correct any errors.
25. Click **Create**.
26. Select the load balancer by name.
27. From the **Actions** menu, select **Edit attributes**.
28. Set the **Idle timeout** to 120 seconds.
29. Click **Save**.

Create a Target Group for your cluster's websockets traffic

This target group will be used to route websockets traffic for the Splunk SOAR (On-premises) Cluster. See Groups for Your Application Load Balancers in the AWS documentation.

1. In the sidebar on the EC2 dashboard, under **Load Balancing**, select **Target Groups**.
2. Click **Create target group**.
3. Now create the websockets target group. In the **Create target group** dialog:
    1. Type a name in the **Target group name** field.
    2. Select the **Instance** radio button.
    3. Select **HTTPS** from the **Protocol** menu. **Port** will change to **443**.

       You must also open Splunk SOAR (On-premises)'s custom HTTPS port 9999 to allow HTTPS traffic for unprivileged processes.

    4. Select the same **VPC** that your target Splunk SOAR (On-premises) instances are using from the menu.
    5. Under **Health Check settings**, select **HTTPS** from the **Protocol** menu.

       Health checks will fail until you have run the make_cluster_node scripts to add your Splunk SOAR (On-premises) instances to your cluster. This is normal and expected.

    6. In the **Path** field, type /**check**.
    7. Click **Next: Register Targets**.
    8. Under **Instances**, find and select the cluster node instances for your Splunk SOAR (On-premises) cluster. You do not need to load balance the external services, such as PostgreSQL, file shares, or Splunk Enterprise.
    9. Click **Add to registered**.
4. Click **Create**.
5. From the target groups list, select the target group you just created.
6. On the **Description tab**, under **Attributes**, click **Edit attributes**.

7. In the **Edit attributes** dialog:
    1. For **Stickiness**, select the **Enable** check box.
    2. Set **Stickiness duration** by typing **7** and choosing **days** from the menu.
    3. Click **Save**.

Setting the Stickiness duration is important so that websockets can persist. Always use the longest possible duration available. Setting this value too low will result in connections being prematurely closed. Wherever possible, set the idle_timeout.timeout_seconds to a value as high as possible for your Elastic Load Balancer. See Application Load Balancers in the AWS documentation.

Add the routing rules to your load balancer

Here you create rules to route traffic.

- One rule to route all the persistent connections to the websockets listener.
- A second rule to route all other traffic to the other listener.

1. From the EWS menu, under **Load Balancing**, select **Load Balancers**.
2. Select the load balancer you have created for your Splunk SOAR (On-premises) cluster.
3. Click the **Listeners** tab.
4. Under **Rules**, click the **View/edit** rules link.
5. Click the **+** icon to add a new rule.
6. Click the **+ Insert Rule** link to edit the rule.
7. Under **IF (all match)**, click **+ Add condition**.
8. **Select Pathâ ¦**, then type /**websocket** in the text box.
9. Click the checkmark icon.

Create the file stores with Elastic File System (EFS)

Create shared file stores for your Splunk SOAR (On-premises) cluster. Cluster nodes will store files that must be shared by all instances to these shares. See System Requirements for more information.

Only instances in the VPC you select during EFS creation can connect to that file system.

1. Under **Configure file system access**, select the desired **VPC** from the menu.
2. Under **Create mount targets**, select the check boxes for the availability zones you need.
3. Click **Next Step**.
4. Set the security groups as required by your organization's policies.
5. Under **Configure optional settings**, set options as required by your organization's requirements or policies.
6. Click **Next Step**.
7. Review the options selected, then click **Create File System**.

Create the external PostgreSQL database with the Relational Database System (RDS)

Splunk SOAR (On-premises) uses a PostgreSQL 11 database. In many installations, the database runs on the same server as Splunk SOAR (On-premises). For an AWS cluster, it makes sense to set up an external PostgreSQL database using RDS. This database will serve as the primary database for the Splunk SOAR (On-premises) cluster.

You may use any release of PostgreSQL 11.x. See Upgrading for support.

1. From your EC2 dashboard, click **Services** in the menu bar, and under **Database** choose **RDS**.
2. Click **Create database**.
3. Select **Standard Create**.
4. Under **Engine options**, select **PostgreSQL**.
5. For **Version**, select **11.11** from the menu. You may use any PostgreSQL 11.x release.
6. For **Templates**, select either **Production** for production environments or **Dev/Test** for development/testing or Proof of Value environments.
7. Under **Settings**, type a name for your **DB instance identifier**. Make sure that the name is unique across all DB instances owned by your AWS account.
8. Under **Credential Settings**:
    1. **Master username**: postgres
    2. Make sure the **Auto generate a password** checkbox is not selected.
    3. Type and confirm the **Master password** in the fields provided. Record this password. You will need it later.
9. Under **DB instance size**, select the radio button that matches your organization's needs.

    Warning: Instances below db.t2.large may deplete their available connections before installation of your Splunk SOAR (On-premises) cluster is complete.
10. Under **Storage**, select a **Storage type** based on your organization's needs.
    1. For **Allocated storage**, set a number of GiB that matches your organization's needs.

        Databases with less than 500 gigabytes of storage are not supported for production use.
    2. Select the **Enable storage autoscaling** check box.
    3. Set **Maximum storage threshold** to **1000** (GiB).
11. Under **Availability & durability**, select the **Do not create a standby** instance radio button.
12. Under **Connectivity**, select the same **VPC** as you used for your Splunk SOAR (On-premises) instances.
13. Under the **Additional connectivity configuration** section:
    1. Select the correct **Subnet group**. The available groups depend on your **VPC** selection.
    2. Under **Publicly accessible**, select the **No** radio button.
    3. Under **VPC security group**, select **Choose existing**.
    4. Select the appropriate security group from the menu.
    5. Click the **X** icon to remove any unwanted security groups that were added by default.
    6. Make sure the **Database port** is set to **5432**.
14. Under **Additional configuration, Database options**:
    1. Type **phantom** for **Initial database name**.
    2. Make sure the **DB parameter group** is set to **default.postgres11.11**. If you selected a different PostgreSQL version 11 earlier, set the parameter to match.
15. Under **Additional configuration, Backup**, leave everything at the defaults.
16. Click **Create Database**.

Create the pgbouncer user for the RDS

Splunk SOAR (On-premises) interacts with the PostgreSQL database using the pgbouncer user account. This account needs to be created for the database created in RDS.

1. Login to an AMI-based Splunk SOAR (On-premises) instance as the phantom user using SSH.
2. Create the pgbouncer user.
   phenv psql --host <DNS name for RDS instance> --port 5432 --username postgres --echo-all --dbname phantom --command "CREATE ROLE pgbouncer WITH PASSWORD '<pgbouncer password>' login;"

3. Make the pgbouncer user a superuser.
   phenv psql --host <DNS name for RDS instance> --port 5432 --username postgres --echo-all --dbname phantom --command "GRANT rds_superuser TO pgbouncer;"

Add file shares to each Splunk SOAR (On-premises) instance

Set up and mount the needed directories for your Splunk SOAR (On-premises) cluster. Do this in three stages. The first to install the required packages, the second to create the required shared directories in EFS and copy over existing data, the third to mount the directories on all Splunk SOAR (On-premises) instances and make the mounts permanent.

Stage one:

Do this stage on each of your AMI-based Splunk SOAR (On-premises) instances.

1. Login to an AMI-based Splunk SOAR (On-premises) instance as the phantom user using SSH.
2. Elevate to root.
   sudo su -
3. Install the package nfs-utils.
   yum install nfs-utils

Stage two:

Do this stage on only one of your AMI-based Splunk SOAR (On-premises) instances. You will create a temporary directory, mount it to EFS, then use it to copy existing files to EFS.

1. Login to an AMI-based Splunk SOAR (On-premises) instance as the phantom user using SSH.
2. Elevate to root.
   sudo su -
3. Create a local mount on this instance. This mount will be used to replicate the required directory structure on EFS.
   mkdir -p /mnt/external
4. Mount this directory from EFS.
   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/ /mnt/external
5. Now copy the instance's files to EFS with rsync.
   rsync -avz /<PHANTOM_HOME>/apps /mnt/external/
   rsync -avz /<PHANTOM_HOME>/local_data/app_states /mnt/external/
   rsync -avz /<PHANTOM_HOME>/scm /mnt/external/
   rsync -avz /<PHANTOM_HOME>/tmp/shared /mnt/external/
   rsync -avz /<PHANTOM_HOME>/vault /mnt/external/
6. Unmount the temporary mounting.
   umount /mnt/external

Stage three:

Do this stage on each of your AMI-based Splunk SOAR (On-premises) instances. Set the mounts for the shared directories to EFS, then update the file system table to make the directories mount from EFS when the instance starts.

1. Login to an AMI-based Splunk SOAR (On-premises) instance as the phantom user using SSH.
2. Elevate to root.
   sudo su -

3. Mount all the shared directories to EFS.
   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/apps /<PHANTOM_HOME>/apps/

   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/app_states /<PHANTOM_HOME>/local_data/app_states

   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/scm /<PHANTOM_HOME>/scm

   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/shared /<PHANTOM_HOME>/tmp/shared

   mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 <ip address or DNS name for EFS>:/vault /<PHANTOM_HOME>/vault
4. Edit the file system table `/etc/fstab` to make the mounts permanent. Add these entries. You can get the EFS ID from your EFS dashboard.
   vi /etc/fstab
   <EFS ID>:/apps /<PHANTOM_HOME>/apps nfs4 defaults,_netdev 0 0
   <EFS ID>:/app_states /<PHANTOM_HOME>/local_data/app_states nfs4 defaults,_netdev 0 0
   <EFS ID>:/scm /<PHANTOM_HOME>/scm nfs4 defaults,_netdev 0 0
   <EFS ID>:/shared /<PHANTOM_HOME>/tmp/shared nfs4 defaults,_netdev 0 0
   <EFS ID>:/vault /<PHANTOM_HOME>/vault nfs4 defaults,_netdev 0 0

Convert an AMI-based Splunk SOAR (On-premises) instance into the Splunk Enterprise instance

A Splunk SOAR (On-premises) cluster requires either a Splunk Enterprise instance or a distributed Splunk Enterprise deployment as its search endpoint. Convert one of your AMI-based Splunk SOAR (On-premises) instances into the required Splunk Enterprise endpoint.

If you already have a Splunk Enterprise deployment that you will use instead, follow the instructions for using an external Splunk Enterprise instance. See Set up Splunk Enterprise.

Convert Splunk SOAR (On-premises) instance into the Splunk Enterprise instance:

1. Login to an AMI-based Splunk SOAR (On-premises) instance as the phantom user using SSH.
2. Elevate to root.
   sudo su -
3. Run the `make_server_node.pyc` script with the `splunk` argument.
   <PHANTOM_HOME>/bin/phenv python <PHANTOM_HOME>/bin/make_server_node.pyc splunk

The Splunk Enterprise configuration is written to: `<PHANTOM_HOME>/bin/splunk_config.json`

Logs are written to: `<PHANTOM_HOME>/var/log/phantom/make_server_node/make_server_node_<date and time>.log`

Test each Splunk SOAR (On-premises) instance for readiness

Before proceeding, test each instance to make sure it is ready for conversion to a cluster node. Log in to each AMI-based Splunk SOAR (On-premises) instance that will become a cluster node, test that the EFS file shares are mounted and fix any errors.

Make sure that each instance has the EFS file shares mounted.

sudo df -T
You must see entries for shared directories in the table with the `<EFS ID.dns_name>:/` for the directories **apps**, **app_states**, **scm**, **shared**, and **vault**.

Convert the first AMI-Based Splunk SOAR (On-premises) instance into a cluster node

Convert the first instance to a Splunk SOAR (On-premises) cluster node.

> Converting an AMI-based installation to a server or cluster node is a one-way operation. It cannot be reverted.

You will need this information readily available:

- IP or hostname for the RDS Postgres 11.6 or later DB server
- Password for the postgres user
- Password for the pgbouncer user
- IP or hostname of the ELB load balancer
- Username for SSH
- Path to the key file for SSH
- IP or hostname of the Splunk Enterprise instance
- REST API port for Splunk Enterprise: 5122
- User name for Splunk SOAR (On-premises) Search: phantomsearch
- Password for the phantomsearch account
- User name for Splunk SOAR (On-premises) Search: phantomdelete
- Password for the phantomdelete account
- HTTP Event Collector Token
- HTTP Event Collector port: 5121

The information for the Splunk Enterprise instance can be found in the file `<PHANTOM_HOME>/bin/splunk_config.json` on your Splunk Enterprise instance.

> Make a note the AWS instance ID of this instance. You need it later to log in to your Splunk SOAR (On-premises) cluster.

Run make_cluster_node.pyc

1. SSH to the AMI-based Splunk SOAR (On-premises) instance. Log in with the phantom user account.
2. Run the `make_cluster_node.pyc` script with the `--record` argument.
   phenv python <PHANTOM_HOME>/bin/make_cluster_node.pyc --record

The response file is written to: `<PHANTOM_HOME>/bin/response.json`

The log is written to: `<PHANTOM_HOME>/var/log/phantommake_cluster_node/make_cluster_node_<date and time>.log`

The response file can be used with the `make_cluster_node.pyc` script on other nodes to automatically provide the information the script needs.

Convert the remaining AMI-based Splunk SOAR (On-premises) instances into cluster nodes

Convert each of the remaining AMI-based Splunk SOAR (On-premises) instances into cluster nodes by running the make_cluster_node.pyc script.

Run make_cluster_node.pyc

1. SSH to the AMI-based Splunk SOAR (On-premises) instance. Log in with the phantom user account.
2. Run the `make_cluster_node.pyc` script with the `--responses` argument.
   <PHANTOM_HOME>/bin/phenv python <PHANTOM_HOME>/bin/make_cluster_node.pyc --responses <PHANTOM_HOME>/bin/response.json

You don't have to use responses.json. If you do not supply a JSON file, the script prompts you for the information it needs. The mcn_responses.json file contains secrets such as usernames and passwords in plain text. Store it in a secure location or delete it after the cluster configuration is complete.

Log in to the Splunk SOAR (On-premises) web interface

Connect to the web interface of your newly installed Splunk SOAR (On-premises) cluster.

Use the AWS instance ID of the first Splunk SOAR (On-premises) instance where the make_cluster_node script was run for the cluster's initial password.

1. Get the public IP address or DNS name for the elastic load balancer from the EC2 Management Console.
2. Get the full AWS instance ID for the EC2 instance.
3. Using a browser, go to the public IP address or DNS name for the elastic load balancer.
    1. User name: admin
    2. Password: <Full AWS instance ID>
4. Change the admin user's password:
    1. From the User Name menu, select Account Settings.
    2. From the second level of the menu bar, select Change Password.
    3. Type the current password.
    4. Type a new password.
    5. Type a new password a second time to confirm.
    6. Click Change Password.

Convert an existing Splunk SOAR (On-premises) instance into a cluster

If you have an existing Splunk SOAR (On-premises) instance that you want to turn into a clustered Splunk SOAR (On-premises) deployment you need to build a new Splunk SOAR (On-premises) cluster, then restore the PostgreSQL database of your existing instance to your new cluster. The checklist lists the stages and references the topics with detailed instructions.

Number	Task	Description
1	Build a new cluster with external services	Build a cluster using one of the following methods:

Number	Task	Description
		• Create a cluster from an unprivileged installation. See Create a Splunk SOAR (On-premises) cluster using an unprivileged installation. • Create a cluster in Amazon Web Services. See Create a Splunk SOAR (On-premises) cluster in Amazon Web Services.
2	Back up your current Splunk SOAR (On-premises) instance.	See Backup a Splunk SOAR (On-premises) database and restore to an external database in Set up an external PostgreSQL server in *Install and Upgrade Splunk SOAR (On-premises)*.
3	Restore the PostgreSQL database to the PostgreSQL database for the new cluster.	Backup a Splunk SOAR (On-premises) database and restore to an external database in Set up an external PostgreSQL server in *Install and Upgrade Splunk SOAR (On-premises)*.

Run make_server_node.pyc

Use the `make_server_node.pyc` script to convert an install into either a specific service or a Shared Services server for a Splunk SOAR (On-premises) cluster.

Additional configuration steps for unprivileged clusters

Perform the following steps on the load balancer or Shared Services server as **root** or as a user using **sudo** to get elevated permissions.

1. Set SELINUX to allow HAProxy to bind to your custom HTTPS port.

   If SELINUX is disabled, then skip this step.

   semanage port --add --type http_port_t --proto tcp <HTTPS PORT>
   If you receive an error that the port is already defined, use `--modify` instead of `--add`.

   semanage port --modify --type http_port_t --proto tcp <HTTPS PORT>
2. Edit `/etc/haproxy/haproxy.cfg` to remove the comment marker # from the frontend block on the line for your custom HTTPS port. `# bind *:<HTTPS PORT> ssl crt /etc/haproxy/ â ¦`

   Becomes:

   ```
   bind *:<HTTPS PORT> ssl crt /etc/haproxy/ â ¦
   ```
3. Restart HAProxy.
   systemctl restart rh-haproxy18-haproxy

Create a Shared Services server

A single Shared Services server becomes a single point of failure. Any problems on the Shared Services server impact your entire Splunk SOAR (On-premises) cluster. For production use, build a server for each service rather than a single Shared Services server.

A single Shared Services server is not recommended for production use. This mode is primarily intended for Proof of Value or demonstrations.

Create a Shared Services server as root or using sudo:

/opt/phantom/bin/phenv python /opt/phantom/bin/make_server_node.pyc
Making a Shared Services server also generates the `/opt/phantom/bin/mcn_responses.json` file, which can be passed as an argument to `make_cluster_node.pyc` to help set up the first Splunk SOAR (On-premises) node in your cluster.

> The mcn_responses.json file contains secrets such as usernames and passwords in plain text. Store it in a secure location or delete it after the cluster configuration is complete.

Create a specific function server

Create a specific function server, such as an HAProxy load balancer, PostgreSQL database, file share, or Splunk Enterprise as root or using sudo:

/opt/phantom/bin/phenv python /opt/phantom/bin/make_server_node.pyc --<option argument>
Repeat once on separate virtual machine image installations for each server.

Valid arguments:

- fs - sets up a single server GlusterFS for file shares.
- db - sets up the internal PostgreSQL database to be used as an external PostgreSQL database.
- proxy - installs and configures HAProxy to serve as a load balancer for your Splunk SOAR (On-premises) cluster.
- splunk - allows the local Splunk Enterprise to be used as a remote search endpoint.

make_sever_node.pyc prompts and warnings

The `make_server_node.pyc` script issues a warning that you are about to permanently change your Splunk SOAR (On-premises) instance.

The changes are:

- Splunk SOAR (On-premises) is removed from system boot scripts.
- Disabling the internal Splunk SOAR (On-premises) database.
- Configuring file shares.
- Installing HAProxy to act as a load balancer.
- Installing Splunk Enterprise.
- You must respond to the warning with "y" for yes to proceed.

You are prompted to supply information for the TLS certificate.

- Country Code
- State Code
- City
- Organization
- Organization unit
- Hostname (or IP address)
- Email address

The remaining prompts are:

- The subnet on which PostgreSQL will accept connections.
- Set the passwords for the postgres and pgbouncer user accounts.

- Password for the user account.

When the script completes it writes the file `/opt/phantom/bin/mcn_responses.json`.

Logs are written to `/var/log/phantom/make_server_node/make_server_node_<date and time>.log`.

Run make_cluster_node.pyc

Use the `make_cluster_node.pyc` script to configure an installed Splunk SOAR (On-premises) instance into a node of a cluster. This script stores the bulk of required configuration information from the PostgreSQL database.

Before running `make_cluster_node`, make sure that all the required services are working, either as external services or as a Shared Services server.

Collect the required information

You need this information to answer prompts for make_cluster_node.

- IP addresses or hostnames for:
  - PostgreSQL 9.5 server
  - HAProxy server and the port that the HAProxy server uses to accept HTTPS connections
  - GlusterFS server
  - Splunk Enterprise instance REST port
  - Splunk Enterprise instance HTTP Event Collector port
- User names, passwords, tokens, or SSH key information for:
  - pgbouncer PostgreSQL database user
  - postgres PostgreSQL database user
  - login password for the HAProxy server, unless it uses an ssh key
  - Splunk SOAR (On-premises) username and password for the install being converted
  - Splunk Enterprise user with `phantomsearch` permissions
  - Splunk Enterprise user with `phantomdelete` permissions
  - Splunk Enterprise HTTP Event Collector token

Create a Splunk SOAR (On-premises) node

Once you have either a Shared Services server or external services established, you convert installations of Splunk SOAR (On-premises) into cluster nodes.

Privileged installation

On a privileged installation, such as an RPM installation, run the `make_cluster_node.pyc` script as `root` or a user with `sudo` permissions.

1. Run the `make_cluster_node.pyc` script.
   ```
   /opt/phantom/bin/phenv python /opt/phantom/bin/make_cluster_node.pyc --responses
   /path/to/mcn_responses.json
   ```
   You don't have to use mcn_responses.json. If you do not supply a JSON file, the script prompts you for the information it needs. The mcn_responses.json file contains secrets such as usernames and passwords in plain text. Store it in a secure location or delete it after the cluster configuration is complete.

2. For each other node, run the script without arguments.
   ```
   /opt/phantom/bin/phenv python /opt/phantom/bin/make_cluster_node.pyc
   ```

Unprivileged installation

On an unprivileged installation you must first change to the directory where Splunk SOAR (On-premises) is installed.

1. Change to the Splunk SOAR (On-premises) home directory.
   ```
   cd <phantom_install_dir>/bin/
   ```
2. Run `make_cluster_node.pyc` using python.
   ```
   phenv python ./make_cluster_node.pyc --responses /path/to/mcn_responses.json
   ```

   You don't have to use mcn_responses.json. If you do not supply a JSON file, the script prompts you for the information it needs. The mcn_responses.json file contains secrets such as usernames and passwords in plain text. Store it in a secure location or delete it after the cluster configuration is complete.

Run Splunk SOAR (On-premises) Using External Services

Set up an external PostgreSQL server

Splunk SOAR (On-premises) uses a PostgreSQL 11 database. In many installations, the database runs on the same server as Splunk SOAR (On-premises). It is possible to put the database on its own server. For more information about configuring and operating a PostgreSQL database, consult the PostgreSQL website and their documentation.

Limited Support for PostgreSQL 12

If your organization requires PostgreSQL 12, Splunk SOAR (On-premises) can provide limited support. Follow the directions in this topic, substituting PostgreSQL 12.9 for PostgreSQL 11.16.

If you choose to use PostgreSQL 12.9 instead of PostgreSQL 11.16, you must be aware of the following limitations:

- PostgreSQL 12 is only supported for new, unprivileged Splunk SOAR (On-premises) deployments. Do not upgrade an existing external PostgreSQL 11.16 database to PostgreSQL 12.9.
- PostgreSQL 12 is only supported as an external service. Do not upgrade the embedded PostgreSQL 11 database to PostgreSQL 12.
- You cannot use the following features of Splunk SOAR (On-Premises) if you use PostgreSQL 12:
    - Backup and restore. You must use another solution to backup or restore your PostgreSQL 12 database.
    - Warm standby

Install and configure PostgreSQL

If you run the PostgreSQL database on its own server, install and configure PostgreSQL before you install Splunk SOAR (On-premises).These instructions are based on CentOS 7 or Red Hat Enterprise Linux 7. If you choose to to install PostgreSQL on another operating system, consult the documentation on the PostgreSQL website.

1. Install one of the operating systems supported by PostgreSQL 11. PostgreSQL 11.16 is recommended, but you may use any release of PostgreSQL 11.x. Configure the operating system according to your organization's requirements. See Supported Platforms on PostgreSQL.org.
2. Update the kernel semaphore parameters and refresh the system configuration.

   ```
   echo "kernel.sem=250 32000 32 5000" >> /etc/sysctl.conf
   sysctl --system
   ```
3. Configure your firewall to allow access. For a complete list of ports, see Splunk SOAR (On-premises) required ports.
4. Add any additional yum repositories that you need. Use the tool on the Linux downloads (Red Hat family) page to identify the correct repository for your architecture and operating system combination.
   yum install <URL>
5. Install the PostgreSQL server.
   yum install postgresql11-server-11.16
6. Initialize the PostgreSQL database.
   /usr/pgsql-11/bin/postgresql-11-setup initdb
7. Set PostgreSQL to start when the system starts.
   systemctl enable postgresql-11
8. Start the PostgreSQL database.
   systemctl start postgresql-11

9. Change to the postgres user.
   su - postgres
10. Change to the PostgreSQL data directory.
    cd /var/lib/pgsql/11
11. Generate the SSL certificate PostgreSQL uses.
    openssl req -new -x509 -days 3650 -nodes -text -out server.crt -keyout server.key -subj "/CN=postgres.cluster1"

    > You can use an SSL certificate purchased from a Certificate Authority instead of generating a self-signed certificate.

12. Set the permissions on the server.key file.
    chmod og-rwx server.key
13. Run a PostgreSQL shell as the `postgres` user.

    > You should already be the postgres user.

    psql
14. Set the postgres user password, if it has not already been set.
    ALTER USER postgres PASSWORD '<postgrespassword>';
15. Create the pgbouncer user.
    CREATE USER pgbouncer PASSWORD '<pgbouncerpassword>';
16. Set PostgreSQL to use SSL. Provide the keys and cipher level.

```
ALTER SYSTEM SET ssl = on;
ALTER SYSTEM SET ssl_cert_file = '/var/lib/pgsql/11/server.crt';
ALTER SYSTEM SET ssl_key_file = '/var/lib/pgsql/11/server.key';
ALTER SYSTEM SET ssl_ciphers = 'HIGH:+3DES:!aNULL';
```

17. Exit the PostgreSQL shell by typing CTRL+D.
18. Change back to the `root` user.
    exit
19. Edit the `pg_hba.conf` file to enable access to the database. Splunk SOAR (On-premises) must be able to connect as both the `postgres` and `pgbouncer` users. In each entry, supply the IP range that will be used by your Splunk SOAR (On-premises) install or cluster.

```
# TYPE  DATABASE    USER        ADDRESS         METHOD
local   all           all         peer
hostssl all         postgres    <IP Range>/<XX> md5
hostssl phantom     pgbouncer   <IP Range>/<XX> md5
```

20. Edit `postgresql.conf`. Set values for `max_connections`, `work_mem`, `shared_buffers`, and `listen_address`.

```
max_connections=2500
work_mem=2796kB
shared_buffers=2GB
listen_addresses = '*' # what IP address(es) to listen on;
```

    > Several factors can influence the amount of memory dedicated to the work_mem setting. Larger, high event volume deployments will want significantly more, while smaller, lower volume deployments may use slightly less. The setting above assumes a medium sized deployment with a moderate event volume.

    For `listen_address` set a value that matches your security requirements. Valid settings are:

    - `*` for all addresses, 0.0.0. for all IPv4 addresses
    - `::` for all IPv6 addresses
    - specific addresses you supply.
21. Restart the PostgreSQL service.
    systemctl restart postgresql-11

Backup a Splunk SOAR (On-premises) database and restore to an external database

To backup a Splunk SOAR (On-premises) database and restore it on an external database, use the ibackup.pyc tool. See Splunk SOAR (On-premises) backup and restore overview in *Administer Splunk SOAR (On-premises).*

Create an external PostgreSQL database in AWS RDS

Some Splunk SOAR (On-premises) deployments, especially those in Amazon Web Services, may want to put the PostgreSQL database on its own host using AWS's Relational Database Service (RDS). These steps can also be used to migrate a standalone Splunk SOAR (On-premises) instance to an instance using a PostgreSQL database hosted in AWS' RDS.

For more information on building a PostgreSQL database host in RDS, see the Amazon Relational Database Service documentation.

Checklist

Number	Task	Description
1	Create a PostgreSQL database in AWS.	Create the external PostgreSQL database with the Relational Database System
2	Create the pgbouncer user account.	Create the pgbouncer user for the RDS
3	Back up the Splunk SOAR (On-premises) instance's PostgreSQL database, then restore it to the AWS RDS instance.	Backup a Splunk SOAR (On-premises) database and restore to an external database

Create the external PostgreSQL database with the Relational Database System (RDS)

Splunk SOAR (On-premises) uses a PostgreSQL 11.16 database. In many installations, the database runs on the same server as Splunk SOAR (On-premises). If you opt to run the PostgreSQL database on its own Amazon Web Services RDS instance, follow the procedures here. These instructions assume you already have an AWS account and VPCs established for your organization's resources.

These instructions use PostgreSQL version 11.16. You may use any PostgreSQL 11.x release. If you do use a different release, you must to use matching parameters for your release where PostgreSQL 11.16 is specified in these instructions.

1. From your EC2 dashboard, click **Services** in the menu bar, and under **Database** choose **RDS**.
2. Click **Create database**.
3. Select **Standard Create**.
4. Under **Engine options**, select **PostgreSQL**.
5. For **Version**, select **11.16** from the menu.
6. For **Templates**, select either **Production** for production environments or **Dev/Test** for development/testing or Proof of Value environments.
7. Under **Settings**, type a name for your **DB instance identifier**. Make sure that the name is unique across all DB instances owned by your AWS account.
8. Under **Credential Settings**:
    1. **Master username**: postgres
    2. Make sure the **Auto generate a password** checkbox is not selected.
    3. Type and confirm the **Master password** in the fields provided. Record this password. You will need it later.
9. Under **DB instance size**, select the radio button that matches your organization's needs.

Warning: Instances below db.t2.large may deplete their available connections before installation of Splunk SOAR (On-premises) is complete.

10. Under **Storage**, select a **Storage type** based on your organization's needs.
    1. For **Allocated storage**, set a number of GiB that matches your organization's needs.

       Databases with less than 500 gigabytes of storage are not supported for production use.

    2. Select the **Enable storage autoscaling** check box.
    3. Set **Maximum storage threshold** to **1000** (GiB).
11. Under **Availability & durability**, select the **Do not create a standby** instance radio button.
12. Under **Connectivity**, select the same **VPC** as you used for your Splunk SOAR (On-premises) instance.
13. Under the **Additional connectivity configuration** section:
    1. Select the correct **Subnet group**. The available groups depend on your **VPC** selection.
    2. Under **Publicly accessible**, select the **No** radio button.
    3. Under **VPC security group**, select **Choose existing**.
    4. Select the appropriate security group from the menu.
    5. Click the **X** icon to remove any unwanted security groups that were added by default.
    6. Make sure the **Database port** is set to **5432**.
14. Under **Additional configuration, Database options**:
    1. Type **phantom** for **Initial database name**.
    2. Make sure the **DB parameter group** is set to **default.postgres11.16**.
15. Under **Additional configuration, Backup**, leave everything at the defaults.
16. Click **Create Database**.

Create the pgbouncer user for the RDS

Splunk SOAR (On-premises) interacts with the PostgreSQL database using the pgbouncer user account. This account needs to be created for the database you built in RDS.

1. SSH to the operating system of your Splunk SOAR (On-premises) instance.
2. Elevate to root.
   sudo su -
3. Create the pgbouncer user.
   psql --host <DNS name for RDS instance> --port 5432 --username postgres --echo-all --dbname phantom --command "CREATE ROLE pgbouncer WITH PASSWORD '<pgbouncer password>' login;"
4. Make the pgbouncer user a superuser.
   psql --host <DNS name for RDS instance> --port 5432 --username postgres --echo-all --dbname phantom --command "GRANT rds_superuser TO pgbouncer;"

Backup the external PostgreSQL database with the Relational Database System (RDS)

To backup an external PostgreSQL database with the RDS, perform the following steps as the **root** user or a user with **sudo** permissions.

You must use identical versions of Splunk SOAR (On-premises) for this procedure. For example, if your PostgreSQL backup is from Splunk SOAR (On-premises) 5.0.1, you must restore it to use with an instance of Splunk SOAR (On-premises) 5.0.1.

1. Backup the database.

   ```
   cd <PHANTOM_HOME>/bin
   ```

```
phenv python backup.pyc --all
```

2. Copy the file path that shows the backup file that was created to use in a future step.

```
All data backed up to
<PHANTOM_HOME>/data/phantom_backups/phantom_backup_2017-07-15-20-47-04.126913.tgz
```

3. Edit the `<PHANTOM_HOME>/etc/pgbouncer/pgbouncer.database.ini` file as shown in the following code. `host` is the IP address or DNS name of the database server.

```
[databases]
phantom = user=pgbouncer password=<pgbouncerpassword> host=<pg server>
postgres = user=postgres password=<postgrespassword> host=<pg server>
[pgbouncer]
server_tls_sslmode = require
```

> Amazon Web Services RDS PostgreSQL databases do not need the server_tls_sslmode = require entry.

4. Stop all Splunk SOAR (On-premises) services.

```
<PHANTOM_HOME>/bin/stop_phantom.sh
```

5. Reload pgbouncer. For all deployments, use the following command:

```
<PHANTOM_HOME>/bin/phsvc restart pgbouncer
```

6. Test the connection to the database server.

```
<PHANTOM_HOME>/bin/phenv psql  -h /tmp -p 6432 -d postgres
```

If connectivity is successful, you will see the following message:

```
psql (11.16)
Type "help" for help.
postgres=#
```

7. Initialize the database to use with Splunk SOAR (On-premises).

```
cd <PHANTOM_HOME>/bin
phenv prepare_db
```

8. Start all Splunk SOAR (On-premises) services.

```
<PHANTOM_HOME>/bin/start_phantom.sh
```

9. Restore the backup using the file name you copied in step 2.

```
cd <PHANTOM_HOME>/bin
phenv python restore.pyc --file
<PHANTOM_HOME>/data/phantom_backups/phantom_backup_2017-07-15-20-47-04.126913.tgz
```

10. Connect to the Splunk SOAR (On-premises) server's web user interface.

Set up external file shares using GlusterFS

Splunk SOAR (On-premises) uses several volumes for storage. Splunk SOAR (On-premises) implements GlusterFS for scalability and security of its file shares. You can put these volumes on their own server, or any server that has adequate storage and bandwidth.

> You can use other file systems to provide shared storage. Any file system that meets your organization's security and performance requirements is sufficient. You need to configure the required mounts and permissions. See Supported file systems and required directories.

You can run GlusterFS as an expandable cluster of servers which provide a single mount point for access. While you can run GlusterFS on a single server, three or more servers provides more options for redundancy and high availability.

These instructions cover only configuring a single server and the required shares. To achieve high availability, data redundancy, and other features of GlusterFS see the GlusterFS Documentation.

Prepare the GlusterFS server

1. Install and configure one of the supported operating systems according to your organization's requirements.
2. Install the prerequisites.
   yum install -y wget curl ntp
3. Synchronize the system clock.
   ntpdate -v -u 0.centos.pool.ntp.org
4. Configure your firewall to allow access for Splunk SOAR (On-premises) nodes and other members of your GlusterFS cluster. For a complete list of ports, see Splunk SOAR (On-premises) required ports.
5. Format and mount the storage partition. This partition must be separate from the operating system partition. The partition must be formatted with a file system that supports extended attributes.

   ```
   mkfs.xfs /dev/<device_name>
   mkdir -p /data/gluster
   echo '/dev/<device_name> /data/gluster xfs defaults 0 0' >> /etc/fstab
   mount -a && mount
   ```
6. Install the phantom-base repository.
   - CentOS or RHEL version 7:
     tar -xvzf <installer>.tgz
7. Update yum.
   yum update
8. Install GlusterFS server.
   yum install -y glusterfs-server-7.5-1.el7
9. Start the GlusterFS daemon and set it to start at boot.

   ```
   systemctl start glusterd
   systemctl enable glusterd
   ```

Prepare TLS certificates

1. Create the TLS certificates for GlusterFS.
   openssl genrsa -out /etc/ssl/glusterfs.key 2048
2. Generate the .pem key for GlusterFS. You can use a certificate from a CA instead of generating a self-signed certificate.
   openssl req -new -x509 -days 3650 -key /etc/ssl/glusterfs.key -subj '/CN=gluster' -out /etc/ssl/glusterfs.pem
3. Copy the glusterfs.pem file to a .ca file.
   cp /etc/ssl/glusterfs.pem /etc/ssl/glusterfs.ca
4. Set ownership, read, write, and execute permissions on the `glusterfs.key` file.

   ```
   chown <user>:<group> /etc/ssl/glusterfs.key
   chmod o-rwx /etc/ssl/glusterfs.key
   ```
5. Create the directory and control file to make GlusterFS use TLS.

   ```
   mkdir -p /var/lib/glusterd/
   touch /var/lib/glusterd/secure-access
   ```
6. Copy the files for the TLS configuration. Store the copies in a safe place.

You will need these files to connect client machines to the file share.

tar -C /etc/ssl -cvzf glusterkeys.tgz glusterfs.ca glusterfs.key glusterfs.pem

Configure the shared volumes

1. Create the shared directories used by Splunk SOAR (On-premises).

   ```
   cd /data/gluster/
   mkdir -p apps app_states scm tmp/shared vault
   ```
2. Create the volumes in GlusterFS from the directories. Repeat for each volume: apps, app_states, scm, tmp, and vault.
   gluster volume create <volume name> transport tcp <GlusterFS hostname>:/data/gluster/<volume name> force
3. Activate SSL/TLS for each volume. Repeat for each volume: apps, app_states, scm, tmp, and vault.

   ```
   gluster volume set <volume name> client.ssl on
   gluster volume set <volume name> server.ssl on
   gluster volume set <volume name> auth.ssl-allow '*'
   ```
4. Start each volume. Repeat for each volume: apps, app_states, scm, tmp, and vault.
   gluster volume start <volume name>

Configure Splunk SOAR (On-premises) cluster nodes to connect to the GlusterFS file shares

Each Splunk SOAR (On-premises) node in your cluster must have the same TLS keys stored in `/etc/ssl/`. Make sure to use the keys generated during GlusterFS installation.

If you are using the Splunk SOAR (On-premises) web interface to add new cluster nodes, you will need to supply the TLS keys in **Administration > Product Settings > Clustering**.

1. Create the directory and control file to make GlusterFS use TLS.

   ```
   mkdir -p /var/lib/glusterd/
   touch /var/lib/glusterd/secure-access
   ```
2. Copy your `glusterkeys.tgz` file to `/etc/ssl/` on the Splunk SOAR (On-premises) instance.
3. Extract the tar file.
   tar xvzf glusterkeys.tgz
4. Delete the `glusterkeys.tgz` file from `/etc/ssl/`.

Sync Splunk SOAR (On-premises) cluster nodes to the shared volumes

Splunk SOAR (On-premises) nodes must sync their local files to your newly shared volumes. The local directories for `apps`, `app_states`, `scm`, `tmp/shared`, and `vault` contain files that need to be preserved for use by your Splunk SOAR (On-premises) instance or cluster.

In a clustered environment, data only needs to be synced from the first node. Syncing data from additional nodes will overwrite data from the first node.

1. Stop Splunk SOAR (On-premises) services on each node of the cluster.
   stop_phantom.sh
2. Mount the local volumes to a temporary directory.

```
mkdir -p /tmp/phantom/<volume>
mount -t glusterfs <hostname of external file share>:<glusterfs volume name> /tmp/phantom/<volume>
```
The shared directory should be mounted a little differently.
```
mkdir -p /tmp/phantom/shared
mount -t glusterfs <hostname of external file share>:tmp /tmp/phantom/shared
```

If you get an error message mount: unknown filesystem type 'glusterfs', then you have not installed glusterfs. See Prepare the GlusterFS server

3. Sync local data to the temporary location.
   rsync -ah --progress <path/to/local/volume>/ /tmp/phantom/<volume>/
   The shared directory should be synched using this command.
   rsync -ah --progress <path/to/local/volume>/tmp/shared/ /tmp/phantom/shared/
   Repeat for each volume: `apps`, `app_states`, `scm`, and `shared`.
4. Sync the vault.
   rsync -ah --exclude tmp --exclude chunks --progress <path/to/local/vault>/ /tmp/phantom/vault/
   Sync the vault separately because it often contains very large amounts of data.
5. Unmount the temporary volumes. Repeat for each volume: apps, app_states, scm, tmp/shared, and vault.
   umount /tmp/phantom/<volume>
6. Edit the cluster member's file system table, `/etc/fstab`, to mount the GlusterFS volumes. Your `fstab` entries must not have line breaks.

   ```
   <glusterfs_hostname>:/apps /<phantom_install_dir>/apps glusterfs defaults,_netdev 0 0
   <glusterfs_hostname>:/app_states /<phantom_install_dir>/local_data/app_states glusterfs
   defaults,_netdev 0 0
   <glusterfs_hostname>:/scm /<phantom_install_dir>/scm glusterfs defaults,_netdev 0 0
   <glusterfs_hostname>:/tmp /<phantom_install_dir>/tmp/shared glusterfs defaults,_netdev 0 0
   <glusterfs_hostname>:/vault /<phantom_install_dir>/vault glusterfs defaults,_netdev 0 0
   ```
7. Mount all the volumes to make them available.

   ```
   mount /<phantom_install_dir>/apps
   mount /<phantom_install_dir>/local_data/app_states
   mount /<phantom_install_dir>/scm
   mount /<phantom_install_dir>/tmp/shared
   mount /<phantom_install_dir>/vault
   ```
8. Start Splunk SOAR (On-premises) services on all cluster nodes.
   start_phantom.sh

Set up a load balancer with an HAProxyÂ® server

A Splunk SOAR (On-premises) cluster uses HAProxy as a load balancer to distribute requests between instances.

You can use a different load balancer. Your load balancer must be configured to:

- provide round-robin balancing
- support SSL/TLS
- handle redirection from HTTP to HTTPS services.

The HA Proxy server that serves a Splunk SOAR (On-premises) cluster with the default configuration will encrypt traffic from clients to the proxy, and from the proxy to the Splunk SOAR (On-premises) nodes. The traffic to the Splunk SOAR (On-premises) nodes is sent over port 443, but the certificates of the Splunk SOAR (On-premises) nodes do not require validation.

If you use a different load balancer when creating a Splunk SOAR (On-premises) cluster, see Configuration files in the Reference section for an HAProxy configuration to use as an example.

1. Install and configure one of the supported operating systems according to your organization's requirements.
2. Update SELinux and any firewalls to allow access to the ports for HAProxy, and your Splunk SOAR (On-premises) cluster nodes.
3. Install HAProxy.
   yum install haproxy
4. Add SSL/TLS certificates to /etc/haproxy/certificates. These certificates are used to encrypt communications between the load balancer and clients.

   Do not use a self-signed certificate in a production environment for client communications.
5. Edit `/etc/haproxy/haproxy.cfg`. If the file does not exist, create it. Use the example file HAProxy Configuration as a guide. If you are creating an unprivileged cluster, make sure to include a directive for your custom HTTPS port such as:
   ```
   bind *:443 ssl crt /etc/haproxy/certificates no-sslv3 no-tlsv10 ciphers <ciphers go here>
   # for unprivileged installs, add another declaration
   bind *:<your https port> ssl crt /etc/haproxy/certificates no-sslv3
   ```
6. Set HAProxy to start when the system starts.
   systemctl enable haproxy.service
7. Start HAProxy.
   systemctl start haproxy.service

See also

- For general setup and information on HAProxy, see the HAProxy documentation on the HAProxy.org website.
- For specific information on SSL/TLS certificates, see the section about certs in the HAProxy Configuration Manual.

Set up Splunk Enterprise

If Splunk SOAR (On-premises) is installed as a stand-alone product, it includes a version of Splunk Enterprise as the internal search engine. You can also configure Splunk SOAR (On-premises) to use an external Splunk instance for searching. A Splunk SOAR (On-premises) cluster also requires an external Splunk Enterprise instance.

Review the product compatibility matrix in About the Splunk Phantom Remote Search app in the *Splunk Phantom Remote Search* manual to make sure compatible versions of the Splunk platform and Splunk SOAR (On-premises) are being used.

The Splunk Phantom Remote Search App defines the user roles and indices needed by Splunk SOAR (On-premises) to use Splunk Enterprise for searches.

Install Splunk Enterprise and add-ons

1. Install and configure Splunk Enterprise from the documentation. See the Splunk Enterprise Installation Manual.
2. Configure your firewall to allow access. For a complete list of ports, see Splunk SOAR (On-premises) required ports.
3. Install the Splunk Phantom Remote Search App. See Where to get more apps and add-ons in the *Splunk Enterprise Admin Manual*.

4. Set up the HTTP Event Collector in Splunk. See Set up and use HTTP Event Collector in Splunk Web in the *Splunk Enterprise Getting Data In* manual.

Create required user accounts for Splunk SOAR (On-premises)

Splunk SOAR (On-premises) requires two user accounts with roles added by the Splunk Phantom Remote Search App. The roles are phantomsearch and phantomdelete. You can use any user names you like for these accounts. These instructions use phantomsearch and phantomdelete.

1. Select **Settings > Access Controls**.
2. Click **Users**.
3. Click **New User**.
4. Type **phantomsearch** for Name.
5. Set and confirm a password for this user which complies with your organization's security policies.
6. Under Assigned role(s), in the Selected item(s) box, select **user** to remove that role.
7. Under Assigned role(s), in the Available item(s) box, select **phantomsearch** to add that role.
8. Deselect the **Require password change on first login** check box.
9. Click **Save**.
10. Click **New User**.
11. Type **phantomdelete** for Name.
12. Set and confirm a password for this user which complies with your organization's security policies.
13. Under Assigned role(s), in the Selected item(s) box, select **user** to remove that role.
14. Under Assigned role(s), in the Available item(s) box, select **phantomdelete** to add that role.
15. Deselect the **Require password change on first login** check box.
16. Click **Save**.

Configure Splunk SOAR (On-premises) instances to use external Splunk Enterprise

Once your Splunk SOAR (On-premises) instances have been installed, configure them to use the external Splunk Enterprise.

You need a Splunk Enterprise license to use external Splunk Enterprise for remote search. If you do not already have one, please work with your Delivery Team to purchase one.

You need the host name of your Splunk Enterprise server, the HTTP Event Collector token, and the passwords for the user accounts with the phantomsearch and phantomdelete roles.

1. Log in to Splunk SOAR (On-premises) as an administrative user.
2. From the Main Menu, select **Administration**.
3. Select **Administration Settings > Search Settings**.
4. From Search Endpoint, select the radio button for **External Splunk Enterprise Instance**.
5. Type the host name of your Splunk Enterprise server in the Host field.
6. Type the user name and password for the user account with the phantomsearch role in the Username and Password fields.
7. Type the user name and password for the user account with the phantomdelete role in the Username and Password fields.
8. Type the port number that Splunk Enterprise uses to listen for REST API calls in the REST Port field.
9. Select the **Use SSL for REST** to enable SSL for REST API calls.
10. Select the **Verify Certificate for REST** to validate the SSL certificate used for REST API calls. Requires a trusted certificate configured in your certificate store.

11. Type the port number for the Splunk Enterprise HTTP Event Collector in the HTTP Event Collector Port field.
12. Select the **Use SSL for HTTP Event Collector** check box to enable SSL for the HTTP Event Collector.
13. Paste the HTTP Event Collector token in the HTTP Event Collector Token field.
14. Select the **Verify Certificate for HTTP Event Collector** check box to validate the SSL certificate used by the Event Collector. This requires a trusted certificate configured in your certificate store.
15. Click **Save Changes**.

Upgrade Splunk SOAR (On-premises)

Splunk SOAR (On-premises) upgrade overview and prerequisites

Splunk SOAR (On-premises) requires incremental upgrades from earlier versions. This means, for example, that you need to upgrade from the latest version of 5.0.x to the latest version of 5.1.x to the latest version of 5.2.x to the latest version of 5.3.x. Version numbers are formatted as major.minor.patch. For example, 5.2.1 is major version 5, minor version 2, patch version 1.

The current upgrade path is as follows:

- 4.6.latest version to 4.8.any version
- 4.8.latest version to 4.9.any version
- 4.9.latest version to 4.10.any version
- 4.10.any version to 4.10.any higher version. You cannot go backwards to a lower release.
- 4.10.latest version (4.10.7) to 5.0.1
- 5.0.latest version to 5.1.any higher version
- 5.1.latest version to 5.2.any higher version
- 5.2.latest version to 5.3.any higher version
- 5.3.latest version to 5.3.any higher version
- 5.3.latest version to 5.4.0

Python 3.9 impact on apps: You must upgrade apps to be compatible with with Python 3.9. If you don't, those apps might not run in the Python 3.9 environment.

Refer to the following table for latest build numbers:

Starting Splunk Phantom or Splunk SOAR (On-premises) release	Build number	Upgrade to version	Build number
Splunk Phantom 4.6	4.6.19142	Splunk Phantom 4.8 patch 1	4.8.24304
Splunk Phantom 4.8 patch 1	4.8.24304	Splunk Phantom 4.9 Release 5	4.9.39220
Splunk Phantom 4.9 Release 5	4.9.39220	Splunk Phantom 4.10.7	4.10.7.63984
Splunk Phantom 4.10.7	4.10.7.63984	Splunk SOAR (On-premises) 5.0.1	5.0.1.66250
Splunk SOAR (On-premises) 5.0.1	5.0.1.66250	Splunk SOAR (On-premises) 5.1.0	5.1.0.70187
Splunk SOAR (On-premises) 5.1.0	5.1.0.70187	Splunk SOAR (On-premises) 5.2.1	5.2.1.78411
Splunk SOAR (On-premises) 5.2.1	5.2.1.78411	Splunk SOAR (On-premises) 5.3.5	5.3.5.97812
Splunk SOAR (On-premises) 5.3.3	5.3.3.92213	Splunk SOAR (On-premises) 5.3.5	5.3.5.97812
Splunk SOAR (On-premises) 5.3.4	5.3.4.95226	Splunk SOAR (On-premises) 5.3.5	5.3.5.97812

Starting Splunk Phantom or Splunk SOAR (On-premises) release	Build number	Upgrade to version	Build number
Splunk SOAR (On-premises) 5.3.5	5.3.5.97812	Splunk SOAR (On-premises) 5.4.0	5.4.0.101028

Do not skip any minor versions when upgrading. You can upgrade to a higher patch version without upgrading to the patch versions between your currently deployed minor version and a higher patch version.

Upgrade checklist

Follow these steps to prepare for and upgrade Splunk SOAR (On-premises):

Step	Tasks	Description
1	Make a full backup of your Splunk SOAR (On-premises) deployment	Make a full backup of your Splunk SOAR (On-premises) deployment before upgrading. See Backup or restore your Splunk SOAR (On-premises) instance in *Administer Splunk SOAR (On-premises)*. For single instance deployments running as a virtual machine, you can create a snapshot of the virtual machine instead.
2	Do the prerequisites	See Prerequisites for upgrading Splunk SOAR (On-premises). 1. Obtain logins 2. Make sure the Splunk SOAR (On-premises) instance or cluster nodes have enough available space. 3. If needed, add a local yum repository or create a satellite server for yum updates.
3	Upgrade Splunk SOAR (On-premises)	See Upgrade Splunk SOAR (On-premises)
4	Repair indicator hashes for non-federal information processing standards (FIPS)	If you are upgrading a non-FIPS instance, you must run the following script after running the installation script: `repair_520_indicators.sh`. That script is located in the home directory you specified in the --phantom-home argument. You may optionally pass the batch size as an argument: `repair_520_indicators.sh <batch_size>`. The default batch size is 1000. You can restart the script at any time. The script terminates after execution. • In clustered configurations, run this script on any single Splunk SOAR (On-premises) node. • In configurations using warm standby, run this script only on the primary system.
5	Conditional Rerun the setup command for ibackup	See Prepare Splunk SOAR (On-premises) for a backup in *Administer Splunk SOAR (On-premises)*.

After all the preparation stages are complete, you can upgrade your Splunk SOAR (On-premises) instance or cluster. For clustered deployments, after the preparation stages are complete, upgrading your Splunk SOAR (On-premises) cluster is done in a rolling fashion, one node at a time.

Prerequisites for upgrading Splunk SOAR (On-premises)

You need the following information before beginning your upgrade:

- Logins
  - For privileged deployments, user accounts on the operating system for your Splunk SOAR (On-premises) instance or cluster nodes with **sudo** or **root** access on those systems.
  - For unprivileged deployments, you also need the login credentials for the user account that runs Splunk SOAR (On-premises). For new AMI versions of Splunk SOAR (On-premises), the user account is

phantom.
    - Your Splunk Phantom Community portal login.
- If your Splunk SOAR (On-premises) deployment has restricted internet access, you will need a local yum repository or a satellite server from which to get yum packages.
- A minimum of 5GB of space available in the `/tmp` directory on the Splunk SOAR (On-premises) instance or cluster node.
- Make note of the directory where Splunk SOAR (On-premises) is installed.
    - On a privileged deployment - /opt/phantom
    - On an unprivileged AMI deployment - /opt/phantom, also called <PHANTOM_HOME>.
    - On an unprivileged deployment - the home directory of the user account that will run Splunk SOAR (On-premises), also called <PHANTOM_HOME>.

For deployments with restricted internet access, add local yum repositories for upgrade

If your Splunk SOAR (On-premises) deployment has no access or restricted access to the internet, you must either create a satellite server or local YUM repository for operating system packages and other dependencies. See the Red Hat Knowledgebase article How can we regularly update a disconnected system (A system without internet connection)?

The required upgrade repositories are as follows:

OS version	CentOS	RHEL
7	[base] [updates]	[rhel-7-server-rpms] [rhel-server-rhscl-7-rpms] [rhel-7-server-optional-rpms]

Prepare your Splunk SOAR (On-premises) deployment for upgrade

Before you upgrade Splunk SOAR (On-premises), you will need to prepare your instance or your cluster nodes by updating the operating system, installed packages, and adding the Splunk SOAR (On-premises) repositories and their signing keys.

Migrate a privileged deployment to an unprivileged deployment

The AMI and OVA versions of Splunk SOAR (On-premises) are unprivileged. New AMI and OVA installations run Splunk SOAR (On-premises) as the user account phantom rather than as root.

Update the operating system and installed packages

Follow these steps to update the operating system and otherwise prepare your deployment for the upgrade.

For a clustered deployment, prepare cluster nodes in a rolling fashion, one cluster node at a time.

1. Log in to the Splunk SOAR (On-premises) instance's operating system:
    1. For privileged deployments, log in as the **root** user or a user with **sudo** privileges.
    2. For unprivileged deployments, log in as the user account that runs Splunk SOAR (On-premises).
2. If you use a warm standby or use ibackup.pyc for backups, you must disable those features before proceeding. If you are not using either of those features, you may skip these sub-steps.

    1. On a single instance deployment of Splunk SOAR (On-premises), disable warm standby. See Upgrade or maintain warm standby instances in *Administer Splunk SOAR (On-premises)*.
    2. If you are using automation to run ibackup.pyc to make backups, cancel backups that could run during your upgrade window. For example, if you have configured a cron job to run ibackup.pyc, disable that cron job.

1. Stop all Splunk SOAR (On-premises) services. For example, as the **root** user:
   `/<PHANTOM_HOME>/bin/stop_phantom.sh`
2. Clear the YUM caches. As the **root** user:
   `yum clean all`
3. Update the installed software packages and apply operating system patches. As the **root** user:
   `yum update`
   Systems which cannot access YUM repositories over the internet need a satellite server. See For deployments with restricted internet access, add local yum repositories for upgrade.

   If you are using the EPEL repository some packages may be upgraded to a version higher than supported by Splunk SOAR (On-premises). In this case, you want to use the Official Offline RPMs instead of using YUM to get the required versions of package dependencies for Splunk SOAR (On-premises). See For Splunk Phantom deployments without internet access or unprivileged deployments for instructions.
4. Restart the operating system. As the **root** user:
   `reboot`
5. After the system restarts, log in to the operating system as either the **root** user or a user with **sudo** privileges.
6. The install script requires the ability to create jobs in cron. See System requirements for production use. Check that the cron daemon is running.
   `ps -ef | grep crond`
    1. If the cron daemon is not running, start it.
       `systemctl start crond.service`

Upgrade Splunk SOAR (On-premises)

When you are ready to upgrade Splunk SOAR (On-premises), follow one of these sets of instructions, based on your deployment type:

- Upgrade a single privileged Splunk SOAR (On-premises) instance
- Upgrade a single unprivileged Splunk SOAR (On-premises) instance
- Upgrade a privileged Splunk SOAR (On-premises) cluster
- Upgrade an unprivileged Splunk SOAR (On-premises) cluster

Convert a privileged deployment to an unprivileged deployment

From release 5.3.3 and higher of Splunk SOAR (On-premises), you can convert an privileged deployment of Splunk SOAR (On-premises) to an unprivileged deployment.

When you upgrade to release 5.4.0 of Splunk SOAR (On-premises) the installer automatically converts any privileged deployment of Splunk SOAR (On-premises) to an unprivileged deployment.

Converting a privileged Splunk SOAR (On-premises) deployment to an unprivileged deployment cannot be undone. Make sure you are ready to convert before running the running the conversion tool or upgrading to release 5.4.0.

Before you begin

There are a few steps to perform before you begin the conversion.

1. Make a full backup of your Splunk SOAR (On-premises) deployment. See Splunk SOAR (On-premises) backup and restore overview in *Administer Splunk SOAR (On-premises*.
2. Disable any warm standby. See Disable warm standby for Splunk SOAR (On-premises) in *Administer Splunk SOAR (On-premises*.
3. Disable any cron jobs or other automated processes that might try to make changes to your Splunk SOAR (On-premises) deployment during the conversion process.

Changes to a privileged deployment when converting to an unprivileged deployment

Unprivileged instances of Splunk SOAR (On-premises) run as a user other than the root user.

- New Splunk SOAR (On-premises) OVA or AMI deployments run under the user account phantom.
- Privileged deployments converted to unprivileged deployments run under the user account phantom.
- Manually installed unprivileged deployments run under the user account specified during installation.

These changes are made to a deployment which is converted from privileged to unprivileged.

- RPM dependencies that are replaced with unprivileged versions are uninstalled.
    - pgbouncer
    - nginx
    - postgresql
    - git
- Splunk SOAR (On-premises) RPM files are removed from the RPM database. Existing files are not removed, only the RPM database entries. This largely impacts deployments which were upgraded from Splunk Phantom.
- Change the owner of everything in the <PHANTOM_HOME> directory to the owner phantom:phantom.
- Disable SElinux
- Install the unprivileged versions of dependency items.
    - pgbouncer
    - nginx
    - postgresql
    - git
- Reconfigures auto-boot.
- Modifies logging config setting for all the Splunk SOAR daemons in the phantom database.
- Remove rsyslog configuration.
- Updates the necessary configuration files, mostly for updating logging paths.
- Moves Splunk SOAR (On-premises) logs from /var/log/phantom to <PHANTOM_HOME>/var/log/phantom.
- Ensures that the phantom user has a gecos/full name attribute set.
- Configure a firewall port forward from the custom unprivileged HTTPS port (default is 8443) to HTTPS port 443. This item requires firewalld to be running.

Manually converting a privileged deployment to an unprivileged deployment

Once you have upgraded to the 5.3.3 release of Splunk SOAR (On-premises), you can convert your privileged deployment to unprivileged one at any time. The tool works for single instances or clusters.

Converting a privileged Splunk SOAR (On-premises) deployment to an unprivileged deployment cannot be undone. Make sure you are ready to convert before running the running the conversion tool.

If you want to manually convert a privileged deployment of Splunk SOAR (On-premises) to an unprivileged one, do the following:

1. Make sure that firewalld is active and running. The migration script requires firewalld to be active so it can be configured.
    1. Check the status of firewalld.
       ```
       sudo systemctl status firewalld
       ```
       Example output from an active firewalld:
       ```
       â   firewalld.service - firewalld - dynamic firewall daemon

       Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)


       Active: active (running) since Wed 2022-07-13 19:00:17 GMT; 1 weeks 1 days ago
       ```
    2. (Conditional) If firewalld is not active, enable it, then activate it.
       ```
       sudo systemctl enable firewalldsudo systemctl start firewalld
       ```
2. Change directory to /opt/phantom.
   ```
   cd /opt/phantom
   ```
3. Run the migration tool, and follow the prompts.
   ```
   phenv python migration/migrate.py
   ```
   The migrate.py tool supports two arguments:

    - Use `--no-prompt` or `-y` to run the tool without prompting the user for input.
    - Use `--https-port` or `-p` to specify your custom HTTPS port. If you do not specify port, 8443 is used.
4. (Optional) If you are converting a privileged Splunk SOAR (On-premises) cluster, stop Splunk SOAR on all nodes, then repeat the preceeding steps for each cluster node.

   If you are converting a privileged cluster to an unprivileged one, you will need to configure your load balancer to listen for your custom HTTPS port. If you did not specify a port during the migration, the port 8443 is set for you.

If the script fails to complete the migration, an error message is displayed on stdout that will contain the error encountered and the log file to consult for further troubleshooting.

Upgrade a single privileged Splunk SOAR (On-premises) instance

When you upgrade your Splunk SOAR (On-premises) deployment to release 5.4.0 your privileged deployment will be automatically converted to an unprivileged one. For more information on this conversion, see Convert a privileged deployment to an unprivileged deployment.

Follow these steps to upgrade your Splunk SOAR (On-premises) instance.

The same TAR file is used for install and upgrade processes. The file detects the presence of SOAR and installs or upgrades accordingly.

1. Read Splunk SOAR (On-premises) upgrade overview and prerequisites.

2. Restart the operating system if you did not recently restart it as part of the prerequisites in Step 1. This step is required to ensure that the upgrade completes successfully and efficiently.
   As the **root** user:
   ```
   reboot
   ```
3. After the system restarts, log in to the operating system as either the **root** user or a user with **sudo** privileges.
4. Download the privileged installer from the Splunk Phantom community website Product Downloads page. The installer is packaged with static versions of the product's dependencies when the product is built. The installer is named in the format `splunk_soar-priv-<major>.<minor>.<patch>.<build>-<commit_short_sha>-el7-x86_64.tgz`
5. Extract the TGZ file you downloaded into the /opt/phantom directory using `tar -xf <installer>.tgz` .
6. Change directory to the /opt/phantom/splunk-soar directory. This directory is created when you extract the TGZ file in the previous step.
7. The installer package you extracted created a file called soar-install in the /opt/phantom/splunk-soar directory. Run that as root:
   ```
   sudo ./soar-install --upgrade --with-apps
   ```

Upgrade a single unprivileged Splunk SOAR (On-premises) instance

Follow these steps to upgrade your unprivileged Splunk SOAR (On-premises) instance, or to convert and upgrade your existing, privileged Splunk SOAR (On-premises) instance to an unprivileged instance. Use these steps even if your unprivileged Splunk SOAR (On-premises) instance has limited access to the internet. The installation TAR file contains everything needed to complete this upgrade.

The same TAR file is used for install and upgrade processes. The file detects the presence of SOAR and installs or upgrades accordingly.

1. Read Splunk SOAR (On-premises) upgrade overview and prerequisites.
2. Restart the operating system if you did not recently restart it as part of the prerequisites in Step 1. This step is required to ensure that the upgrade completes successfully and efficiently.
   As the **root** user:
   ```
   reboot
   ```
3. After the system restarts, log in to the operating system as either the **root** user or a user with **sudo** privileges.
4. Download the unprivileged installer from the Splunk Phantom community website Product Downloads page. The unprivileged installer prepackages its dependencies and can be installed on systems that cannot reach out to the internet. The unprivileged installer is named in the format `splunk_soar-unpriv-<major>.<minor>.<patch>.<build>-<commit_short_sha>-el7-x86_64.tgz`.
5. Extract the TGZ file you downloaded into the Splunk SOAR (On-premises) home directory using `tar -xvf <installer>.tgz`. When this finishes, there is a new directory in the Splunk SOAR (On-premises) home directory, <PHANTOM_HOME>/splunk-soar.
6. Log in as the user who owns the Splunk SOAR (On-premises) upgrade. Do not perform the upgrade as the root user.
7. Change directory to the <PHANTOM_HOME>/splunk-soar directory.
8. The installer package you extracted creates a file called soar-install in the <PHANTOM_HOME>/splunk-soar directory. Run that script:
   ```
   ./soar-install --upgrade --with-apps
   ```

You can see the full list of arguments for the soar-install script by using the --help option.

Upgrade a privileged Splunk SOAR (On-premises) cluster

When you upgrade your Splunk SOAR (On-premises) clustered deployment to release 5.4.0 your privileged deployment is automatically converted to an unprivileged one. For more information on this conversion, see Convert a privileged deployment to an unprivileged deployment.

Perform the following tasks to upgrade your Splunk SOAR (On-premises) cluster. These tasks apply to privileged clusters running on local hardware, or privileged clusters running in Amazon Web Services.

Before you begin

Before you begin to upgrade your Splunk SOAR (On-premises) cluster, do the following steps:

1. Read Splunk SOAR (On-premises) upgrade overview and prerequisites.
2. Prepare your cluster's server node or load balancer, if the load balancer is separate from your cluster's server node.
    1. Log in to the operating system of your server node or load balancer as either the **root** user or a user with **sudo** privileges.
    2. Update the firewalld rules to allow the custom HTTPS port for Splunk SOAR (On-premises) .
       ```
       firewall-cmd --permanent --add-port=8443/tcp
       ```
    3. Reload firewalld.
       ```
       fireall-cmd --reload
       ```
    4. Configure haproxy (or other load balancer) to add the custom HTTPS port. On in /etc/haproxy/haproxy.cfg, copy the entry that looks like this:
       ```
       bind *:443 ssl crt /etc/haproxy/certificates no-sslv3 no-tlsv10 ciphers
       ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DHE-RSA-AES256-GCM
       -ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA
       -ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:ECDHE-ECDSA-
       -SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:AES256-GCM-SHA384:AES128-
       -SHA256:AES256-SHA256:AES128-SHA256
       ```
       Add a copy of that entry, edited with your custom HTTPS port:

       ```
       bind *:8443 ssl crt /etc/haproxy/certificates no-sslv3 no-tlsv10 ciphers
       ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DHE-RSA-AES256-GCM
       -ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA
       -ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:ECDHE-ECDSA-
       -SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:AES256-GCM-SHA384:AES128-
       -SHA256:AES256-SHA256:AES128-SHA256
       ```
    5. Reload haproxy.
       ```
       systemctl reload haproxy
       ```
       or
       ```
       systemctl reload rh-haproxy18-haproxy
       ```
       If you use a load balancer other than haproxy, you need to do the equivalent of these steps for that load balancer.

Upgrade the cluster nodes

For each SOAR node, follow the upgrade instructions, one node at a time:

The same TAR file is used for install and upgrade processes. The file detects the presence of SOAR and installs or upgrades accordingly.

1. Restart the operating system if you did not recently restart it as part of the prerequisites.
   This step is required to ensure that the upgrade completes successfully and efficiently.
   As the **root** user:
   ```
   reboot
   ```
2. After the system restarts, log in to the operating system as either the **root** user or a user with **sudo** privileges.
3. Download the privileged installer from the Splunk Phantom community website Product Downloads page. The installer is packaged with static versions of the product's dependencies when the product is built. The installer is named in the format
   `splunk_soar-priv-<major>.<minor>.<patch>.<build>-<commit_short_sha>-el7-x86_64.tgz`.
4. Extract the TGZ file you downloaded using `tar -xf <installer>.tgz` into the /opt/phantom directory.
5. The installer package you extracted creates a file called soar-install in the <PHANTOM_HOME>/splunk-soar directory. Run that as root:
   ```
   sudo <PHANTOM_HOME>/splunk-soar/soar-install --upgrade --with-apps
   ```

You can see the full list of arguments for the soar-install script by using the --help option.

Upgrade an unprivileged Splunk SOAR (On-premises) cluster

Perform the following tasks to upgrade your unprivileged Splunk SOAR (On-premises) cluster.

For each SOAR node, follow the upgrade instructions, one node at a time:

The same TAR file is used for install and upgrade processes. The file detects the presence of SOAR and installs or upgrades accordingly.

1. Read Splunk SOAR (On-premises) upgrade overview and prerequisites.
2. Restart the operating system if you did not recently restart it as part of the prerequisites in Step 1.
   This step is required to ensure that the upgrade completes successfully and efficiently.
   As the **root** user:
   ```
   reboot
   ```
3. After the system restarts, log in to the operating system as either the **root** user or a user with **sudo** privileges.
4. Download the unprivileged installer from the Splunk Phantom community website Product Downloads page. The unprivileged installer prepackages its dependencies and can be installed on systems that cannot reach out to the internet. The unprivileged installer is named in the format
   `splunk_soar-unpriv-<major>.<minor>.<patch>.<build>-<commit_short_sha>-el7-x86_64.tgz`.
5. Extract the TGZ file you downloaded using `tar -xf <installer>.tgz` into the <PHANTOM_HOME> directory. It will create a new directory in your SOAR installation called splunk-soar.
6. As the user that owns the SOAR installation, run the soar-install script provided with the new version. It will automatically detect that you're running an upgrade:
   ```
   <PHANTOM_HOME>/splunk-soar/soar-install --upgrade --with-apps
   ```

You can see the full list of arguments for the soar-install script by using the --help option.

Reference

Splunk SOAR (On-premises) default credentials, script options, and sample configuration files

This section has the default Splunk SOAR (On-premises) credentials, script options and example configuration files.

Default credentials

The default credentials on a new installation of Splunk SOAR (On-premises) are:

Web Interface

- Username: admin
- Password: password

The default credentials of a new AMI installation of Splunk SOAR (On-premises) are:

SSH accounts:

- Username: phantom
- Password: None. You must use the SSH key created when deploying the AMI version of Splunk SOAR (On-premises).

Web Interface

- Username: admin
- Password: <full AWS instance ID>

You should change the default passwords immediately after the installation is complete.

Installation or configuration scripts

This section lists various installation scripts and their command line options.

soar-prepare-system.sh

Use these arguments to prepare your system to install Splunk SOAR (On-premises). This script must be run by the root user or a user with sudo privileges.

General arguments

These options are information or intended to be used for debugging purposes.

Argument	Description
-h, --help	Display the help message then exit.

Argument	Description
Debug options. These are not intended to be used in production systems.	
--no-color	Do not color log output
--dry-run	If this argument is specified, just print the install steps instead of running them.
-v, --verbose, --debug	Output debug-level logging to the console and the log file.

Arguments for install, upgrade, or removal

These options are used for any case; install, upgrade, or removal.

Argument	Description
-y, --no-prompt	If given, do not ask for confirmation before running the installation steps.
--log-format {json, plain, pretty-json}	Default logging format is JSON. Pretty-JSON logging decorates log messages with useful context, whereas plain text logging mirrors the console output more closely.
--no-spinners	If your terminal has problems with showing loading spinners, or you're automating the install and the spinner output is noisy, you can use this flag to disable them.

Arguments for new installations

This set of options is only for installing a new deployment of Splunk SOAR (On-premises).

Argument	Description
--splunk-soar-home <PHANTOM_HOME>	Path that should act as the Splunk SOAR installation directory. Defaults to the directory this script is located in, usually /opt/phantom.
--https-port <PHANTOM_PORT>	TCP port to which Splunk SOAR's webserver will bind for HTTPS. Must be between 1024 and 65535, and defaults to 8443
--splunk-soar-user <PHANTOM_USER>	Name of the user which will own the Splunk SOAR installation
--allow-reboot, --reboot	Allow the script to reboot the system if necessary.

Arguments for running optional steps without prompts

Use these arguments to run specific optional steps without prompting the user.

Argument	Description
--gluster-fs	GlusterFS is only needed if you are using an external file share. This is common if you're constructing a Splunk SOAR cluster.
--ntpd-service	Enable the ntpd service to guarantee clock synchronization
--firewall	Ensure that the required ports are opened in firewalld. Do not use this argument if you are not using firewalld.
--port-forward	Make Splunk SOAR available on the default HTTPS port (443) in addition to the configured port. Do not use this argument if you are not using firewalld or if you are creating a cluster.

soar-install.sh

This script is used to install, upgrade, or remove Splunk SOAR (On-premises).

Optional arguments

These arguments can be used to manually specify which action soar-install.sh should perform, or to turn on debugging options.

Argument	Description
-h, --help	Show the help message then exit.
--upgrade	If a lower version of Splunk SOAR is already installed then upgrade it.
--remove	If a lower version of Splunk SOAR is already installed then remove it.
Debug arguments. These arguments are not intended to be used in production.	
--no-color	Do not color log output.
--dry-run	If this argument is specified, just print the install steps instead of running them.
-v, --verbose, --debug	Output debug-level logging to the log file and the console.
--version <VERSION>	Supply a custom Splunk SOAR version, rather than relying on the .soar file.

Arguments for install, upgrade, or removal

These options are used for any case; install, upgrade, or removal.

Argument	Description
-y, --no-prompt	If given, do not ask for confirmation before running the installation steps.
--log-format {json, plain, pretty-json}	Default logging format is JSON. Pretty-JSON logging decorates log messages with useful context, whereas plain text logging mirrors the console output more closely.
--no-spinners	If your terminal has problems with showing loading spinners, or you're automating the install and the spinner output is noisy, you can use this flag to disable them.

Arguments for new installations

This set of options is only for installing a new deployment of Splunk SOAR (On-premises).

Argument	Description
-c <CONTINUE_FROM>, --continue-from <CONTINUE_FROM>	<CONTINUE_FROM> must be a name for an installation step. Use this option for recovering from a failure. Start running from the named step, and assume that previous steps have already run. Note that removal steps are run in reverse.
-s <STOP_AT>, --stop-at <STOP_AT>	<STOP_AT> must be a name for an installation step. Use this option for recovering from a failure. Stop running before the named step. Note that removal steps are run in reverse.
--ova	Install Splunk SOAR for use in OVA format
--splunk-soar-home <PHANTOM_HOME>, --phantom-home <PHANTOM_HOME>	Path that should act as the Splunk SOAR installation directory. Defaults to the directory this script is located in, usually /opt/phantom.
--https-port <PHANTOM_PORT>,	TCP port to which Splunk SOAR's web server will bind for HTTPS. Must be between 1024 and 65535, and defaults to 8443

Argument	Description
--splunk-soar-port <PHANTOM_PORT>, --phantom-port <PHANTOM_PORT>	
--ignore-warnings	If specified, continue through any non-fatal warnings. This setting is only recommended if you've previously run with warnings enabled and determined, possibly with the assistance of Splunk SOAR support, that the warnings you see can be safely ignored.

--with-appsIf specified, install or upgrade apps at the same time as the platform.

make_server_node.pyc options

Use these options to control the `make_server_node.pyc` command.

Argument	Description
--version	Displays the program's version number.
--help	Display a list and description of arguments.
--no-prompt	Run the program. Do not display the warning prompt.

Configuration files

This section contains example configuration files. Use these as a guide when configuring items for use in your Splunk SOAR (On-premises) deployment.

HAProxy Configuration

```
    ##---------------------------------------------------------------------
    ## HAPROXY 1.8.7 CONFIGURATION FILE
    ##---------------------------------------------------------------------
    #---------------------------------------------------------------------
    # global settings
    #---------------------------------------------------------------------
    global
        tune.ssl.default-dh-param 2048
        log 127.0.0.1:514 local0

    #---------------------------------------------------------------------
    # common defaults
    #---------------------------------------------------------------------
    defaults
        mode http
        timeout connect 0ms
        timeout client 0ms
        timeout server 0ms
        log global

    #---------------------------------------------------------------------
    # SSL w/ redirect to HTTPS
    #---------------------------------------------------------------------
    frontend localhost
        bind *:80
        bind *:443 ssl crt /etc/haproxy/certificates no-sslv3 no-tlsv10 ciphers <ciphers go here>
    # for unprivileged installs, add another declaration
```

```
#   bind *:<your https port> ssl crt /etc/haproxy/certificates no-sslv3
# no-tlsv10 ciphers <ciphers go here>
    redirect scheme https if !{ ssl_fc }
    mode http
    default_backend nodes

#---------------------------------------------------------------------
# backend (output)
#---------------------------------------------------------------------
backend nodes
    mode http
    balance roundrobin
    option http-keep-alive
    option forwardfor
    cookie SRVNAME insert
    option httpchk GET /check HTTP/1.1\r\nHost:\ www.example.com
    http-check expect status 200
    default-server fastinter 1s downinter 5s
    server <phantom node UUID> <IP Address>:443 cookie <phantom node UUID> check ssl verify none
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
```

Remediate directory changes

As of this release, Splunk SOAR (On-premises) features new methods for installing and upgrading.

The new installation and upgrade process includes changes to the directory structure for Splunk SOAR (On-premises). To determine whether the new structure requires remediation, ensuring your applications and playbooks run correctly, reference the following tables.

Remediate directory changes in privileged and unprivileged installations

Reference the directory schema and check for remediation actions in the following table for both privileged and unprivileged installations of Splunk SOAR (On-premises).

Files
All local conf files for the internal Splunk instance • <PHANTOM_HOME>/splunk/etc/system/local/props.conf • <PHANTOM_HOME>/splunk/etc/system/local/web.conf • <PHANTOM_HOME>/splunk/etc/system/local/telemetry.conf • <PHANTOM_HOME>/splunk/etc/system/local/authorize.conf • <PHANTOM_HOME>/splunk/etc/system/local/server.conf • <PHANTOM_HOME>/splunk/etc/splunk-launch.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_httpinput/local/inputs.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_httpinput/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/search/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/alert_logevent/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/alert_webhook/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/appsbrowser/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/launcher/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/gettingstarted/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/introspection_generator_addon/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/learned/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/legacy/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/sample_app/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_archiver/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/SplunkForwarder/local/app.conf

Files
• <PHANTOM_HOME>/splunk/etc/apps/SplunkLightForwarder/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_monitoring_console/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/user-prefs/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_instrumentation/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/splunk_instrumentation/local/telemetry.conf • <PHANTOM_HOME>/splunk/etc/apps/framework/server/apps/quickstartfx/splunkd/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/framework/server/apps/homefx/splunkd/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/framework/server/splunkdj/app_templates/basic/splunkd/local/app.conf • <PHANTOM_HOME>/splunk/etc/apps/framework/server/splunkdj/app_templates/splunkweb/local/app.conf • <PHANTOM_HOME>/splunk/etc/licenses/fixed-sourcetype_8D5CE731EA83A7D11CF05F4FBA3465C457E53DD68FE64EA2C8196F66F07092A5
<PHANTOM_HOME>/etc/supervisord.conf

Remediate directory changes in privileged installations

Reference the directory schema and check for remediation actions in the following table for privileged installations of Splunk SOAR (On-premises).

Files	Remediation
pgbouncer configuration • /etc/pgbouncer/hba.conf • /etc/pgbouncer/userlist.txt • /etc/pgbouncer/pgbouncer.ini	If you need to customize pgbouncer configuration, create a file at <PHANTOM_HOME>/usr/local/pgbouncer.ini.
PostgeSQL configuration • /opt/phantom/data/db/pg_hba.conf • /opt/phantom/data/db/pg_ident.conf • /opt/phantom/data/db/postgresql.conf	If you need to customize postgresql configuration, create a file at <PHANTOM_HOME>/usr/local/postgresql.conf.
NGINX configuration • /etc/nginx/conf.d/default.conf • /usr/share/nginx/html/502.html • /usr/share/nginx/html/502_phantom.html	NGINX reads all files matching /etc/nginx/conf.d/*.conf.
UWSGI configuration • /etc/nginx/uwsgi.ini • /etc/nginx/uwsgi_log_json.ini	If you need to customize UWSGI configuration, create a file at /etc/nginx/uwsgi_local.ini.
/etc/logrotate.d/phantom_logrotate.conf	If you need to customize the logrotate configuration, create a custom conf file at <PHANTOM_HOME>/usr/local/logrotate.conf.
/usr/lib/tmpfiles.d/phantom.conf	No action required. Splunk SOAR (On-premises) doesn't support modification of this configuration.
/etc/fonts/conf.d/33-phantom-fonts.conf	No action required. Splunk SOAR (On-premises) doesn't support modification of this configuration.
/etc/cron.d/phantom	Use crontab instead.

Remediate directory changes in unprivileged installations

Reference the directory schema and check for remediation actions in the following table for unprivileged installations of Splunk SOAR (On-premises).

Files	Remediation
pgbouncer configuration • <PHANTOM_HOME>/etc/pgbouncer/hba.conf • <PHANTOM_HOME>/etc/pgbouncer/userlist.txt • <PHANTOM_HOME>/etc/pgbouncer/pgbouncer.ini	If you need to customize pgbouncer configuration, create a file at <PHANTOM_HOME>/usr/local/pgbouncer.ini.
PostgeSQL configuration • <PHANTOM_HOME>/data/db/pg_hba.conf • <PHANTOM_HOME>/data/db/postgresql.conf	If you need to customize postgresql configuration, create a file at <PHANTOM_HOME>/usr/local/postgresql.conf
NGINX configuration • <PHANTOM_HOME>/usr/nginx/conf/phantom-nginx-server.conf • <PHANTOM_HOME>/usr/nginx/conf/conf.d/phantom-nginx-server.conf • <PHANTOM_HOME>/usr/nginx/html/502.html • <PHANTOM_HOME>/usr/nginx/html/502_phantom.html	NGINX reads all files matching <PHANTOM_HOME>/usr/nginx/conf/conf.d/*.conf.
UWSGI configuration • <PHANTOM_HOME>/etc/uwsgi.ini • <PHANTOM_HOME>/etc/uwsgi_log_json.ini	If you need to customize UWSGI configuration, create a file at <PHANTOM_HOME>/etc/uwsgi_local.ini.
<PHANTOM_HOME>/etc/logrotate.d/phantom_logrotate.conf	If you need to customize the logrotate configuration, create a custom conf file at <PHANTOM_HOME>/usr/local/logrotate.conf.