# Table of Contents

About securing Splunk Enterprise

About securing Splunk software................................................................. 1
How to secure and harden your Splunk software installation................... 1

Install Splunk securely................................................................................... 3
Install Splunk Enterprise securely................................................................. 3
Secure your admin account........................................................................... 4
About TLS encryption and cipher suites...................................................... 6
Securing Splunk Enterprise with FIPS......................................................... 9
About default certificate authentication..................................................... 10
Secure Splunk Enterprise on your network............................................... 10
Disable unnecessary Splunk Enterprise components................................. 11
Secure your service accounts...................................................................... 11
Deploy secure passwords across multiple servers................................... 11
Harden your KV store port........................................................................... 12
Some best practices for your servers and operating system..................... 12

Users and role-based access control......................................................... 14
Use access control to secure Splunk data.................................................. 14
About user authentication.......................................................................... 14
About configuring role-based user access................................................. 15
About defining roles with capabilities....................................................... 16
Add and edit roles with Splunk Web......................................................... 21
Add and edit roles with authorize.conf.................................................... 23
Setting access to manager consoles and apps......................................... 26
Find existing users and roles..................................................................... 28
Delete all user accounts............................................................................ 28
Secure access for Splunk knowledge objects.......................................... 28
Use Access Control Lists........................................................................... 29

Native Splunk Enterprise authentication.................................................. 31
Set up Splunk authentication..................................................................... 31
Configure users with Splunk Web.............................................................. 31
Configure users with the CLI................................................................. 32
Add a user to a role with Splunk Web....................................................... 32

Authentication with LDAP.......................................................................... 33
Set up user authentication with LDAP..................................................... 33
Manage Splunk user roles with LDAP.................................................... 34
LDAP prerequisites and considerations.................................................. 34
Secure LDAP with TLS certificates......................................................... 35
How Splunk Enterprise works with multiple LDAP servers..................... 36
Configure LDAP with Splunk Web............................................................ 37
Map LDAP groups to Splunk roles in Splunk Web................................... 41
Configure LDAP with the configuration file.............................................. 42
Map LDAP groups and users to Splunk roles in the configuration files..... 44
Test your LDAP configuration................................................................. 45
# Table of Contents

## Authentication with LDAP
- Convert to LDAP from Splunk authentication........................................................................................................45
- Best practice for removing an LDAP user..................................................................................................................46

## Multi-factor authentication.................................................................................................................................................47
- About multifactor authentication with Duo Security..................................................................................................47
- Configure Splunk Enterprise to use Duo Security multifactor authentication..........................................................47
- Configure Duo multifactor authentication for Splunk Enterprise in the configuration file......................................48

## Authentication using single sign-on with SAML.............................................................................................................50
- Configure single sign-on with SAML.............................................................................................................................50
- Configure SSO with PingIdentity as your identity provider.......................................................................................51
- Configure SSO with Okta as your identity provider..................................................................................................53
- Configure SSO with AzureAD or AD FS as your Identity Provider...........................................................................54
- Configure SSO with OneLogin as your identity provider...........................................................................................56
- Configure SSO with Optimal as your identity provider............................................................................................58
- Configure SSO in CA siteminder.................................................................................................................................60
- Secure SSO with TLS certificates.............................................................................................................................61
- Configuring SAML in a search head cluster................................................................................................................62
- Configure Ping Identity with leaf or intermediate SSL certificate chains...............................................................63
- Configure SAML SSO for other IdPs..........................................................................................................................64
- Configure advanced settings for SSO........................................................................................................................66
- Map SAML groups to roles...........................................................................................................................................66
- Modify or remove role mappings................................................................................................................................67
- Configure SAML SSO in the configuration files........................................................................................................68
- Troubleshoot SAML SSO............................................................................................................................................71

## Authentication using Proxy SSO.........................................................................................................................................75
- About ProxySSO............................................................................................................................................................75
- Configure ProxySSO.....................................................................................................................................................75
- Troubleshoot Proxy SSO.............................................................................................................................................76

## Authentication using single sign-on with reverse proxy............................................................................................78
- About Single Sign-On using reverse proxy................................................................................................................78
- Configure Single Sign-On with reverse proxy...........................................................................................................80
- Troubleshoot reverse-proxy SSO................................................................................................................................82

## Scripted authentication.......................................................................................................................................................83
- Set up user authentication with external systems......................................................................................................83
- Create the authentication script..................................................................................................................................84
- Edit authentication.conf...............................................................................................................................................86
- Use PAM authentication..............................................................................................................................................87
- Use the getSearchFilter function to filter at search time..........................................................................................88

## Securing Splunk Enterprise communications with SSL..................................................................................................89
- About securing Splunk Enterprise with SSL................................................................................................................89
- About using SSL tools on Windows and Linux.............................................................................................................91
# Table of Contents

## Securing Splunk Enterprise communications with SSL
- Configure allowed and restricted SSL versions...

## Certificates for Splunk
- About creating certificates for Splunk...
- Things to know about your certificates...
- About cipher suites and TLS encryption...
- How to prepare your signed certificates for Splunk authentication...
- Determine your cipher suite...
- Working with multiple intermediate certificates

## Secure browser to Splunk Web communication
- About securing Splunk Web...
- Turn on encryption (https) with Splunk Web...
- Turn on encryption (https) using web.conf...
- Secure Splunk Web with your own certificate...
- Troubleshoot your Splunk Web authentication...

## Secure Splunk forwarder to indexer communication
- About securing data from forwarders...
- Configure Splunk forwarding to use the default certificate...
- Configure Splunk forwarding to use your own certificates...
- Validate your configuration...
- Troubleshoot your forwarder to indexer authentication...

## Secure distributed environments
- About securing inter-Splunk communication...
- Securing distributed search heads and peers...
- Secure your deployment server and clients using certificate authentication...
- Secure your clusters with pass4SymmKey...

## Audit Splunk Enterprise activity
- Use Splunk Enterprise to audit your system activity...
- Audit Splunk activity...
- Use audit events to secure Splunk Enterprise...
- Manage data integrity...

## Best practices for Splunk Enterprise security
- Safeguards for risky commands...
- Splunk server tokens...
- Avoid malicious CSV files in searches...

## Appendix A: How to get SSL certificates
- How to self-sign certificates...
- How to get certificates signed by a third-party...
- Self-sign certificates for Splunk Web...
- Get certificates signed by a third-party for Splunk Web...
About securing Splunk Enterprise

About securing Splunk software

During and after your Splunk Enterprise installation, you must take steps to secure both your configuration and your data. Securing your Splunk Enterprise installation reduces its attack surface and mitigates the risk and impact of most vulnerabilities. Some procedures are simple, such as confirming that your servers are physically secure and that your properly manage your credentials. Others, such as configuring encryption, are more complex, but are equally as important to the integrity of your data.

Read this manual to learn about all of the areas of security that you should consider:

- Secure installation of Splunk Enterprise
- Management of users and role-based access control using your chosen form of authentication
- Usage of certificates to secure indexers, forwarders, and Splunk Web, where data is most vulnerable
- Usage of encryption to secure your configuration information
- Usage of auditing to keep track of activity in your system.

Use the How to secure and harden your Splunk software installation as a checklist and roadmap to ensure that you make your configuration and data as secure as possible.

How to secure and harden your Splunk software installation

Use this topic as a checklist and a roadmap for this manual to help you take all of the steps necessary to secure your Splunk software configuration and protect your data.

Set up authenticated users and manage user access

- Secure your Admin password and use it only for administration tasks.
- Use Access Control Lists to restrict user access.
- Set up users and configure Roles and capabilities to control user access.
- Configure user authentication with one of the following methods:
  - Splunk's own built-in system, described in Set up user authentication with Splunk's built-in system.
  - LDAP, described in Set up user authentication with LDAP.
  - A scripted authentication API for use with an external authentication system, such as PAM or RADIUS, described in Set up user authentication with external systems.
- Use one of the following to to create secure one-step login for users:
  - Single Sign on with SAML
  - Multi-factor authentication
  - ProxySSO
  - Reverse-proxy SSO with Apache

Use certificates and encryption to secure communications for your Splunk software configuration

Splunk software comes with a set of default certificates and keys that demonstrate encryption. Splunk recommends deploying your own certificates and configuring them to secure communications. See About securing Splunk with SSL in
Harden your Splunk software instances to reduce vulnerability and risk

- Secure your indexer clusters and search head clusters.
- Set passwords across multiple servers to ensure consistent authentication.
- Secure your service accounts.
- Harden your KV store port.

Audit your system regularly

Audit events contain information that shows you what changed in your Splunk configuration. It gives you the where and when, as well as the identity of the actor who implemented the change. Leveraging audit events provides better security and other benefits:

- Audit your system regularly to monitor user and admin access, as well as other activities that could tip you off to unsafe practices or security breaches.
- Keep an eye on activities within Splunk (such as searches or configuration changes). You can use this information for compliance reporting, troubleshooting, and attribution during incident response.
- Audit events are especially useful in distributed Splunk configurations for detecting configuration and access control changes across many Splunk Servers.
  
  To learn more, see Audit Splunk Enterprise activity in this manual.

- Use the file system-based monitoring available out of the box on most Splunk-supported operating systems.
  For more information about monitoring, see Monitor Files and Directories in the Getting Data In Manual.
Install Splunk securely

Install Splunk Enterprise securely

To install Splunk Enterprise securely, you must have an installation package that you have confirmed is authentic and has not been modified in any way since Splunk created it. Splunk provides a Message Digest 5 (MD5) secure hash for every package it generates. You can download this hash to quickly verify that the package you downloaded is authentic and has not been changed since its creation.

You can also compare the Secure Hash Algorithm-512 (SHA-512) hashes by opening a case with Splunk Support.

Prerequisites for verifying package integrity

You must have the following to verify the contents of packages you download:

- The `md5sum` program, which prints the hash of the file that you supply, and comes with most versions of Linux. On Windows, you can use the `certutil` tool to verify MD5 hashes.
- Alternatively, the `sha512sum` program prints SHA512 hashes for the file that you supply.
- The MD5 or SHA512 hashes, in text format, which Splunk provides
- Access to a shell prompt

Verify installation package integrity

After you download the Splunk Enterprise package, verify it by using a trusted version of the OpenSSL suite to compare the MD5 or SHA-512 hashes to the hash of the installation package. If the hash of the package you downloaded matches the hash that Splunk provides, then you have downloaded a valid, secure installation package and can proceed with installation.

Download Splunk Enterprise installation package and MD5 hash

Confirm that you download the MD5 hash that exactly matches the version of installation package that you downloaded. Downloading a different file results in the hashes not matching.

1. Go to the Splunk.com download page.
2. Click Splunk Enterprise.
3. Click the tab for the operating system that you want to download Splunk software.
4. Click the Download Now link for the OS version and installation package type that you want to install with.
5. On the next page that loads, read the Splunk Software License Agreement.
6. Click the I have read, understood, and hereby accept the Splunk Software License Agreement checkbox.
7. Click Start your download now. The page refreshes and the download begins.
8. On the next page that loads, in the Useful tools box, click MD5 to verify. A second file, the MD5 hash, begins to download.
9. After both downloads finish, complete the "Verify hashes" procedure.

Download Splunk Enterprise installation package and request SHA512 hash from Splunk Support

1. Complete Steps 1 through 7 of the "Download Splunk Enterprise installation package and MD5 hash" procedure.
2. Open a case with Splunk Support to receive the SHA512 hash. When you open the case, provide a link to the version, operating system, and type of installation package you downloaded.
3. After you receive a link to the hash, follow the link to download it.
4. After the package and SHA512 hash downloads finish, complete the “Verify hashes” procedure.

**Verify hashes**

After you download the package, verify it by running either the `md5sum` or `sha512sum` utilities:

1. Open a shell prompt.
2. Change to the directory where you downloaded the installation package and the MD5 hash.
3. Print the contents of the hash file that you downloaded:

   ```
   cat splunk-xxxx-release.tgz.md5
   cat splunk-xxxx-release.tgz.sha512
   ```

4. Run the `md5sum` or `sha512sum` tool on the installation package that you downloaded:

   ```
   md5sum splunk-xxxx-release.tgz.md5
   sha512sum splunk-xxxx-release.tgz.sha512
   ```

5. Compare the output from the MD5 or SHA512 hash file against the result from the `md5sum` or `sha512sum` utilities.
6. If the hashes match exactly, then the package you downloaded is authentic and has not been modified. You can continue with installation.
7. If the hash does not match, then either the package you downloaded has been modified. Retry the download.

**Verify Signatures**

You can verify the authenticity of the downloaded RPM package using the Splunk GnuPG Public key as follows

1. Download the GnuPG Public key file (yes, this link is over TLS).
2. Install the GnuPG public key:
   ```
   rpm --import <filename>
   ```
3. Verify the package signature using:
   ```
   rpm -K <filename>
   ```

**Proceed with installation from your authenticated installation package**

After you have successfully verified your installation package as authentic, you can proceed with installation.

- Installation instructions in the *Installation Manual*

**Secure your admin account**

Splunk with an Enterprise license has a default administration account and password, admin/changeme. Splunk recommends strongly that you change the default in order to keep your system secure. Your password should be complex and follow general password best practices:

- Use a combination of words, numbers, symbols, and both upper- and lower-case letters.
- Complexity is important, but length is vital. We recommend a minimum of 10 characters.
• Do not choose passwords based upon details that may not be as confidential as you’d expect, such as your birth date, your Social Security or phone number, or names of family members.
• Do not use words that can be found in the dictionary.
• Don’t use a password you use or have used elsewhere.

Use Splunk Web

To change the admin default password:

1. Log into Splunk Web as the admin user.
2. Click **Settings** in the top-right of the interface.
3. Click **Access controls** in the Users and Authentication section of the screen.
4. Click **Users**.
5. Click the **admin** user.
6. Update the password, and click **Save**.

Use Splunk CLI

The Splunk CLI command is:

```
splunk edit user
```

**Important:** You must authenticate with the existing password before you can change it. Log into Splunk via the CLI or use the `-auth` parameter. For example, this command changes the admin password from `changeme` to `foo`:

```
splunk edit user admin -password foo -role admin -auth admin:changeme
```

**Note:** On *nix operating systems, the shell interprets some special characters as command directives. You must either escape these characters by preceding them with `\` individually, or enclose the password in single quotes ('). For example:

```
splunk edit user admin -password 'FFL14io!23ur$' -role admin -auth admin:changeme
```

Or

```
splunk edit user admin -password FFL14io!23ur$ -role admin -auth admin:changeme
```

On Windows, use the caret (^) to escape reserved shell characters, or enclose the password in double-quotes ("). For example:

```
splunk edit user admin -password "FFL14io!23ur>" -role admin -auth admin:changeme
```

Or

```
splunk edit user admin -password FFL14io!23ur^> -role admin -auth admin:changeme
```

**Note:** You can also reset all of your passwords across servers at once. See "Deploy secure passwords across multiple servers for the procedure."
About TLS encryption and cipher suites

As of version 6.6., Splunk provides the following default cipher suites and TLS encryption.

**alert_actions.conf**

```conf
sslVersions = tls1.2

cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256

ecdhCurves = prime256v1, secp384r1, secp521r1
```

This configuration does not support Windows Server 2008 R2. To add support for Windows Server 2008 R2:

1. Set `sslVersions` to `tls`
2. Add the following ciphers to the end of the existing `cipherSuite`:

   ```
   ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:
   ECDHE-RSA-AES128-SHA
   ```

To enable TLS 1.2 support on Windows Server 2008 R2:

1. Add key to the registry:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server

2. In the TLS 1.2\Server key, create the following:

   DWORD (32-bit) Value ? DisabledByDefault; set to 0
   DWORD (32-bit) Value ? Enabled; set to 1
3. Restart Windows


**inputs.conf**

```conf
sslVersions = tls1.2

cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256

ecdhCurves = prime256v1, secp384r1, secp521r1
```

This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set `sslVersions` to `tls`
2. Add the following ciphers to the end of the existing `cipherSuite`:
DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA

outputs.conf

sslVersions = tls1.2
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256
ecdhCurves = prime256v1, secp384r1, secp521r1
This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions to tls
2. Add the following ciphers to the end of the existing cipherSuite:

DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA

server.conf

sslVersions = tls1.2
sslVersionsForClient = tls1.2
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ecdhCurves = prime256v1, secp384r1, secp521r1
This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions = tls
2. Set sslVersionsForClient = tls
3. Append AES256-SHA to the existing cipherSuite.

applicationsManagement

sslVersions = tls1.2
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256
ecdhCurves = prime256v1, secp384r1, secp521r1
This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions = tls
2. Add the following ciphers to the end of the existing cipherSuite:

DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA

web.conf

sslVersions = tls1.2

cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:
ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256

ecdhCurves = prime256v1, secp384r1, secp521r1

This configuration does not support Windows Vista. To add support for Windows Vista:

1. Set sslVersions = tls

2. Add the following ciphers to the existing cipherSuite:

ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:
ECDHE-RSA-AES128-SHA

ldap.conf

TLS_PROTOCOL_MIN: 3.1 for TLSv1.0, 3.2 for TLSv1.1, 3.3 for TLSv1.2.

TLS_PROTOCOL_MIN 3.3

TLS_CIPHER_SUITE ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:
ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256

This configuration does not support Windows Server 2008 R2. To add support for Windows Server 2008 R2:

1. Set TLS_PROTOCOL_MIN = TLS1.0/SSL3.1

2. Add the following ciphers to the existing TLS_CIPHER_SUITE:

ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:
ECDHE-RSA-AES128-SHA

To enable TLS 1.2 support on Windows Server 2008 R2:

1. Add key:

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server

2. In the TLS 1.2\Server key, create the following:

DWORD (32-bit) Value ? DisabledByDefault; set to 0

DWORD (32-bit) Value ? Enabled; set to 1

3. Restart Windows. See:
Securing Splunk Enterprise with FIPS

The Federal Information Processing Standard (FIPS) uses government-certified versions of some algorithms to meet regulatory guidelines. It should not be considered a security enhancement by itself, and might potentially reduce performance on your system. Enable FIPS if it is a regulatory requirement for your environment.

Splunk Enterprise and the Universal Forwarder use an embedded FIPS 140-2-validated cryptographic module (Certificate #3126 Module Version fips-2.0.12) running on various platforms per FIPS 140-2 Implementation Guidance section G.5 guidelines.

- The certificate is listed on the National Institute of Standards and Technology (NIST) site. See Certificate Detail | Cryptographic Module Validation Program
- See the consolidated validation certificate also on the NIST website.

Key points to enabling FIPS

There are several things that you must understand when you enable FIPS on Splunk Enterprise:

- You must enable FIPS mode before you start Splunk Enterprise. FIPS mode is disabled except when it runs on a Linux machine that runs a kernel in FIPS mode.
- FIPS is automatically enabled if you run Splunk software on a Linux machine that runs a kernel in FIPS mode.
- The FIPS module disables the use of some cryptographic algorithms in the instance of Python that Splunk software uses to run apps (such as Message Digest 5 (MD5) and Rivest Cipher 4 (RC4).
- Any Splunk apps you want to run on a FIPS-enabled instance must be certified to run in FIPS mode and cannot have dependencies on algorithms like MD5 or RC4.

Enable FIPS

Always enable FIPS mode upon initial Splunk software installation. If you install the software without FIPS mode enabled, you cannot later upgrade it to a FIPS version, and must either reinstall, or install a new version.

1. Before you start Splunk Enterprise for the first time, use a text editor to edit the $SPLUNK_HOME/etc/splunk-launch.conf configuration file.
2. Add the following line to the file:
   SPLUNK_FIPS=1
3. Start Splunk software. It enables FIPS mode during the installation.

Use indexes with FIPS enabled

Running Splunk in FIPS mode does not alter indexed data in any way. You can copy indexes between FIPS and non-FIPS indexers.

Troubleshoot FIPS

- If you are in FIPS mode and your usual RSA encrypted private keys do not work, they might be incompatible with FIPS. To mitigate this issue, you can convert your Privacy Enhanced Mail (PEM) private key to PKCS#8 format to make them compatible.
After you install Splunk software without FIPS mode enabled, you cannot enable FIPS mode. If you require FIPS compliance, confirm that your initial Splunk installation is FIPS-enabled. To change to a version running FIPS mode, reinstall Splunk software and use the procedure in this topic to enable FIPS.

If you have problems running a Splunk app, confirm that it is certified to run in FIPS mode and does not have dependencies on cryptographic algorithms that FIPS disables (such as MD5 and RC4).

About default certificate authentication

Splunk Enterprise 6.6 and higher comes with default certificates that are signed with Secure Hash Algorithm (SHA)-256 using a 2048-bit key. These certificates are part of a fresh installation.

When you upgrade from a previous release, Splunk Enterprise replaces the existing cacert.pem.default and ca.pem.default Privacy Enhanced Mail (PEM) files. Existing certificates, for example cacert.pem and ca.pem, are not affected.

Because of the new default PEM, you must upgrade all certificates and PEM files to SHA-256 using a 2048-bit key to avoid validation errors. For example, your indexers and forwarders might require updates to meet the same standards as your Splunk Enterprise instance. You might also want to check the certificate for your license manager. If all certificates and PEM files are not updated, Splunk Enterprise logs the following error in splunkd.log when it attempts to connect to another instance over SSL:

```plaintext
ERROR TcpOutputFd - Connection to host=10.140.130.102:9997 failed. sock_error = 0.
SSL Error = error:04091077:rsa routines:INT_RSA_VERIFY:wrong signature length
```

Secure Splunk Enterprise on your network

Under certain conditions, Splunk Enterprise ports can become susceptible to attacks. Prevent access by shielding your Splunk Enterprise configuration from the Internet.

If possible, use a host-based firewall to restrict access to Splunkweb, management, and data ports. Keep Splunk Enterprise within a host-based firewall. Have your remote users access Splunk Enterprise on a Virtual Private Network.

You also can protect Splunk Enterprise from attacks in the following ways:

- Restrict CLI security by restricting this port to local calls only, from behind a host firewall.
- Unless necessary, do not allow access to forwarders on any port.
- Install Splunk Enterprise on an isolated network segment that only trustworthy machines can access.
- Limit port accessibility to only necessary connections. The necessary connections are:
  - End users and administrators must access Splunkweb (TCP port 8000 by default).
  - Search heads must access search peers on the management port (TCP port 8089 by default).
  - Deployment clients must access the deployment server on the management port (TCP port 8089 by default).
  - Forwarders must access the index server data port (TCP port 9997 by default).
  - Remote CLI calls use the management port.
- Restrict access to the KV store port on the search head. (The KV store port, by default, is 8191, and by default that port is open to the network.) On each search head cluster member, allow access to the KV store port only for the other members, so that the cluster can replicate KV store data.
Disable unnecessary Splunk Enterprise components

For single-server Splunk Enterprise deployments:

- Forwarders should not run Splunkweb and should not be configured to receive data on TCP or UDP ports or from other Splunk Enterprise instances.

For multiserver Splunk Enterprise deployments:

- Search heads should not receive data on TCP or UDP ports or from other Splunk Enterprise instances.
- If users are not logging in to Splunkweb on indexers in a distributed environment, Splunkweb should be disabled on the indexers.

Secure your service accounts

Practice the principle of least privilege by running Splunk software as an unprivileged user rather than using a privileged account such as root or Administrator.

- On Unix or Linux, use the "splunk" user created with the PKG or RPM packages, or create your own user that only has privilege and ownership over $SPLUNK_HOME.
- On Windows, the local system context is often the best choice. However, if you require communication using a windows communication channel, such as WMI, use a restricted access account.

Deploy secure passwords across multiple servers

At initial startup, Splunk Enterprise creates a file $SPLUNK_HOME/etc/auth/splunk.secret. This file contains a key used to encrypt some of your authentication information in configuration files. Each of the following files can be encrypted across a deployment using splunk.secret. Note that the passwords and encryption methods used for each file are not necessarily interchangeable.

- web.conf: Your SSL passwords on every instance.
- authentication.conf: Your LDAP passwords, if you have any.
- inputs.conf: Your SSL passwords, if you use splunktcp-ssl to set up ssl for data distribution.
- outputs.conf: Your SSL passwords, if you use splunktcp-ssl to configure splunktcp-ssl in inputs.conf to set up ssl for data distribution.
- server.conf: pass4symmkey, if you have one.
- passwords.conf: Your password for a given app.

When Splunk software starts, if it detects a clear-text password, in one of these settings, it will create or overwrite the configuration in the equivalent local folder with the encrypted password.

Note: If the pass4symmkey or SSLPassword is specified in a default apps file, the password is obfuscated in the local version of the file upon restart. The default version of the file remains in clear text. However, if the file is listed using curl or a splunkd endpoint, the passwords appear encrypted.

When you deploy Splunk software on multiple servers, you must encrypt the passwords and ensure that they are consistent across your deployment. Splunk recommends that you use an encryption tool such as Vault's "secret/" to secure your passwords as you distribute them.
You should perform these steps at initial deployment and also any time you need to deploy a new password for your instances:

1. Using your encryption tool, deploy the passwords to all servers.

2. On each server, place the password, in clear text, in the relevant file.

3. Immediately start/restart to encrypt all the passwords under the server's unique secret.

In a search head cluster, the captain replicates its `splunk.secret` file to all other cluster members during initial deployment of the cluster, so you do not need to copy it manually. As part of its normal operation, the cluster also automatically replicates any credentials that are stored by apps for their own use.

**Harden your KV store port**

We recommend that you secure your environment by restricting KV store access to your port. By default, port 8191 is opened to the network. We recommend that you restrict this port when possible.

For search head clustering, you should open the port only to other members of the cluster so that other members can replicate KV store data.

For more information about working with KV store, see About the app key value store

**Some best practices for your servers and operating system**

**Operating System**

To maximize security, harden the operating system on all computers where you run Splunk software.

- If your organization does not have internal hardening standards, consult the CIS hardening benchmarks.
- As a minimum, limit shell/command line access to your Splunk servers.

**Splunk**

- Configure redundant Splunk instances, both indexing a copy of the same data.
- Backup Splunk data and configurations, regularly.
- Execute a periodic recovery test by attempting to restore Splunk Enterprise from backup.
- Verify your Splunk download using a hash function such as MD5 to compare the hashes. For example:

```
./openssl dgst md5 <filename-splunk-downloaded.zip>
```

**Client browser**

- Use a current version of a supported browser, such as Firefox or Internet Explorer.
- Use a client-side JavaScript blocker such as noscript on Firefox or Internet Explorer 8 Filters to help protect against XSS, XSRF, and similar exploits.
- Ensure that users have the latest Flash version installed.
Physical security

- Secure physical access to all Splunk servers.
- Ensure that Splunk end users practice sound physical and endpoint security.
  ♦ Set a short time-out for Splunk Web user sessions. See Configure timeouts for more information.

More opportunities to secure your configuration

- Use a configuration management tool, such as subversion, to provide version control for Splunk configurations.
- Integrate Splunk configuration changes into your existing change management framework.
- Configure Splunk Enterprise to monitor its own configuration files and alert on changes.
Users and role-based access control

Use access control to secure Splunk data

Role-based access control provides flexible and effective tools that you can use to protect Splunk data.

Splunk Enterprise masks data to the user much like the way a relational database manages role-based access control. In some cases total segmentation of data may be necessary. In other cases, controlling the searches and results at the presentation layer (something you can do with many of our Splunk Apps) may meet your security needs.

Consider your use cases when deciding how to set up your configurations and whether role-based access might fit your needs. For example:

• For extremely sensitive data, where even allowing access to a system that might have sensitive data incurs legal risk, consider installing and configuring more than one instance of Splunk Enterprise, and then configuring each instance with the data for the appropriate audience.

• When intentionally or unintentionally exposing sensitive data to the wrong user might incur legal ramifications, then consider creating indexes specifically for privileged and non-privileged accounts and assigning them to roles created for each level of access.

• When there are security concerns but not so much legal risk, you can restrict access using Apps. For example, you can create an App with static dashboards and assign roles with lower clearance to those dashboards, limiting the type of information the user assigned to the role may access.

• Field encryption (optional feature), search exclusions, and field aliasing to redacted data are also great ways to tighten up a limited search case. If you have a limited search case and only able to search some specific data from a shared index, you can restrict shared reports to restrict ad hoc searches and funneling summary indexing into a index that is secured.

About user authentication

Splunk Enterprise authentication allows you to add users, assign them to roles, and give those roles custom permissions as needed for your organization.

Options for authentication systems are as follows:

• Splunk authentication: Provides Admin, Power and User by default, and you can define your own roles using a list of capabilities. If you have an Enterprise license, Splunk authentication is enabled by default. See Set up user authentication with Splunk's built-in system for more information.

• LDAP: Splunk Enterprise supports authentication with its internal authentication services or your existing LDAP server. See Set up user authentication with LDAP for more information.

• Scripted authentication API: Use scripted authentication to integrate Splunk authentication with an external authentication system, such as RADIUS or PAM. See Set up user authentication with external systems for more information.
Note: Authentication, including native authentication, LDAP, and scripted authentication, is not available in Splunk Free.

You can create and assign users to flexible roles either in Splunk Web or by editing authorize.conf. For more information about roles and capabilities, read About role-based user access.

Important: Splunk authentication takes precedence over any external systems. Users are authenticated in the following order:

1. Splunk authentication
2. LDAP or scripted authentication (if enabled)

About configuring role-based user access

If you’re running Splunk Enterprise, you can create users with passwords and assign them to roles. Roles determine the access and permissions of any user assigned to that role.

For more information about users, see About user authentication.

Predefined roles:

- admin: this role is intended for administrators who will manage all or most of the users, objects, and configuration and comes predefined with the most assigned capabilities.
- power: this role can edit all shared objects (saved searches, etc) and alerts, tag events, and other similar tasks.
- user: this role can create and edit its own saved searches, run searches, edit its own preferences, create and edit event types, and other similar tasks.
- can_delete: This role allows the user to delete by keyword. This capability is necessary when using the delete search operator.
- sc_admin (Cloud only): This role allows users to create users and roles but does not grant any other admin capabilities.

You can also create custom roles and assign your users to those roles. When you create a custom role, you determine the following:

- Allowed searches: you can define the searches that a user assigned to the role is allowed to perform.
- Role inheritance: you can have your role inherit certain properties of one or more existing roles. Role inheritance is discussed later in this topic.
- Assign capabilities: you can specify the allowed actions (change their password, change forwarder settings, etc) of the user assigned to the role. See About defining roles with capabilities for more information.
- Set allowed and default indexes: you can limit access to specific indexes and set the event and metrics indexes that are searched by default.

To create roles in Splunk Web, see Add and edit roles with Splunk Web. To create roles by editing authorize.conf, see Add and edit roles with authorize.conf.

Inheritance

As a rule, members of multiple roles inherit properties from the role with the broadest permissions.
How users inherit search filter restrictions

You can create roles that inherit the characteristics of other roles. Users assigned to multiple roles inherit properties from the assigned roles.

In the case of search filters, if a user is assigned to roles with different search filters, the filters are all combined and thus the restrictions of each role are applied.

For example, by default, the Power and User roles do not have search filters defined to restrict searches. If a user has a combination of these roles and another role with filters defined (for example, `srchFilter=x`), the user will inherit the restrictions of that role, despite the association with roles that have no filter.

How users inherit allowed indexes

In the case of allowed indexes, the user is given the highest level of access granted to any role to which they are assigned.

For example, if a user is assigned to the role "simple user" which limits access to one particular index, and also to a role "advanced user" which has more capabilities and allows access to all indexes, the user will have access to all indexes. If you wanted to grant the capabilities of the "advanced user" but continue to limit their index access to the single index defined for the "simple user", you should create a new role specifically for that user.

How users inherit capabilities

In the case of capabilities, the user is given the highest level of abilities granted to any role to which they are assigned.

For example, if a user is assigned to the role "admin" which has the most capabilities, and also to a role "advanced user" which a different set of capabilities, the user will have the capabilities of both roles.

About defining roles with capabilities

When you create a user in Splunk Web you assign that user to one role. See About role-based user access for more information.

Each role contains a set of capabilities. You can add or edit capabilities for new, existing, and default roles. For example, you might give a role the capability to add inputs or edit saved searches.

To add or change the capabilities to a role in Splunk Web, see Add and edit roles with Splunk Web. To create roles by editing authorize.conf, see Add and edit roles with authorize.conf.

List of capabilities

This list shows the capabilities that you can add to any role, and whether any capabilities are assigned by default to the User, Power, or Admin roles.

Capabilities are subject to change. For the most up-to-date list of capabilities, see authorize.conf.

For the most up-to-date list of capabilities assigned to a role, see the "Imported Capabilities" text box in the "Create a role" page.
<table>
<thead>
<tr>
<th>Capability name</th>
<th>What it lets you do</th>
<th>User</th>
<th>Power</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>accelerate_datamodel</td>
<td>Enable or disable acceleration for data models. Set acceleration to true to enable automatic acceleration of this data model. Additional space is required depending on the number of events, fields, and distinct field values in the data. See the Knowledge Manager Manual for more information.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>accelerate_search</td>
<td>Lets the user enable or disable acceleration for reports. The user must also have the schedule_search capability assigned. Works for searches that use transforming commands. See the Knowledge Manager Manual for more information.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>admin_all_objects</td>
<td>Lets the user access and modify any object in the system regardless of any restrictions set in the objects. For example user objects, search jobs, reports, and knowledge objects. Allows the user to bypasses any ACL restrictions, much the way root access in a *nix environment does.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>change_authentication</td>
<td>Lets the user change authentication settings and reload authentication. See the Securing Splunk Enterprise Manual for more about authentication.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>change_own_password</td>
<td>Lets the user change their own password.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>delete_by_keyword</td>
<td>Lets the user use the &quot;delete&quot; operator. The &quot;delete&quot; command marks all of the events returned by the search as deleted. This masks the data from showing up in search results but does not actually delete the raw data on disk. See the Search Manual for more information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>delete_messages</td>
<td>Lets a user delete system messages that appear in the UI navigation bar.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>dispatch_rest_to_indexers</td>
<td>Lets a user dispatch the REST search command to indexers.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_deployment_client</td>
<td>Lets the user change deployment client settings. See the Managing Indexers and Clusters of Indexers Manual for more about the deployment client.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_deployment_server</td>
<td>Lets the user change deployment server settings. User can change or create remote inputs that are pushed to the forwarders and other deployment clients. See the Managing Indexers and Clusters of Indexers manual for more about the deployment server.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_dist_peer</td>
<td>Lets the user add and edit peers for distributed search. See the Managing Indexers and Clusters of Indexers Manual for more information.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_encryption_key_provider</td>
<td>Lets the user view and edit key provider properties when they use Server-Side Encryption (SSE) for a remote storage volume.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_forwarders</td>
<td>Lets the user change forwarder settings, including settings for SSL, backoff schemes, etc. Also used by TCP and Syslog output admin handlers.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_httpauths</td>
<td>Lets the user edit and end user sessions through the httpauth-tokens endpoint.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_indexer_cluster</td>
<td>Lets the user edit indexer clusters. See the Managing Indexers and Clusters of Indexers Manual for more about indexers.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_indexerdiscovery</td>
<td>Lets the user edit settings for indexer discovery, including settings for master_uri, pass4SymmKey, and so on. Used by Indexer Discovery admin handlers.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_input_defaults</td>
<td>Lets the user use the server settings endpoint to change default hostnames for input data.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Capability name</td>
<td>What it lets you do</td>
<td>User</td>
<td>Power</td>
<td>Admin</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>edit_monitor</td>
<td>Lets the user add inputs and edit settings for monitoring files. Also used by the</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>standard inputs endpoint and the one-shot input endpoint.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_roles</td>
<td>Lets the user edit roles and change user/role mappings. Used by both the user and</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>role endpoint.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_roles_grantable</td>
<td>Lets the user edit roles and change user/role mappings for a limited set of roles.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Can assign any role to other users. To limit this ability, configure grantableRoles</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>in authorize.conf. For example: grantableRoles = role1;role2;role3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_scripted</td>
<td>Lets the user create and edit scripted inputs.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_search_head_clustering</td>
<td>Lets the user edit search head clustering settings.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_search_schedule_priority</td>
<td>Lets the user assign a search a higher-than-normal schedule priority. For</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>information about the search scheduler, see the Knowledge Manager Manual.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_search_schedule_window</td>
<td>Lets the user assign schedule windows to scheduled reports. Requires the schedule_</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>search capability. For more about the search scheduler, see the Knowledge Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manual.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_search_scheduler</td>
<td>Lets the user enable and disable the search scheduler. See the Knowledge Manager</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Manual.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_search_server</td>
<td>Lets the user edit general distributed search settings like timeouts,</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>heartbeats, and blacklists.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_server</td>
<td>Lets the user edit general server settings like server name, log levels, etc.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_server_crl</td>
<td>Lets the user edit general server settings like server name, log levels, etc.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Inherits the ability to read general server and introspection settings.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_source_types</td>
<td>Lets the user edit sourcetypes. See the Knowledge Manager manual for more</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>information about sourcetypes.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_splunktcp</td>
<td>Lets the user change settings for receiving TCP inputs from another Splunk instance.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_splunktcp_ssl</td>
<td>Lets the user view or edit any SSL-specific settings for Splunk TCP input.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_splunktcp_token</td>
<td>Lets the user edit the Splunktcp token.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_tcp</td>
<td>Lets the user change settings for receiving general TCP inputs.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_tcp_token</td>
<td>Lets the user change TCP tokens. This is an admin capability and should only be</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>assigned to system administrators.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_telemetry_settings</td>
<td>Opt in or out of product instrumentation. See Share data in Splunk Enterprise in</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>the Admin Manual.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_token_http</td>
<td>Lets the user create, edit, display, and remove settings for HTTP token input.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Also enables the HTTP Event Collector feature.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_udp</td>
<td>Lets the user change settings for UDP inputs.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>edit_user</td>
<td>Lets the user create, edit, or remove users. A role with the edit_user capability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>can assign any role to other users. To limit this ability, configure grantableRoles</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>in authorize.conf. For example: grantableRoles = role1;role2;role3. Also lets a user manage certificates for distributed search.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_view_html</td>
<td>Lets the user create, edit, or modify HTML-based views.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Capability name</td>
<td>What it lets you do</td>
<td>User</td>
<td>Power</td>
<td>Admin</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>edit_web_settings</td>
<td>Lets the user change settings for web.conf through the system settings endpoint.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>embed_report</td>
<td>Lets the user embed reports and disable embedding for embedded reports.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_results_is_visible</td>
<td>Lets the user display or hide the Export Results button in Splunk Web. The default value is to display the button.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>get_diag</td>
<td>Lets the user get a remote diag from a Splunk instance using the /streams/diag endpoint.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>get_metadata</td>
<td>Lets the user use the &quot;metadata&quot; search processor.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>get_typeahead</td>
<td>Lets the user use typeahead in the endpoint and the typeahead search field.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>indexes_edit</td>
<td>Lets the user change any index settings such as file size and memory limits.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>input_file</td>
<td>Lets the user add a file as an input through inputcsv (except for dispatch=t mode) and inputlookup.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>license_edit</td>
<td>Lets the user edit the license.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>license_tab</td>
<td>Lets the user access and change the license. This attribute is deprecated.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>license_view_warnings</td>
<td>Lets the user see a warning message when they are exceeding data limits or reaching the expiration date of their license. These warnings appear on the system banner.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_accelerate_search</td>
<td>Lets the user view accelerated reports. User cannot accelerate reports.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_deployment_client</td>
<td>Lets the user view deployment client settings.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_deployment_server</td>
<td>View deployment server settings.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_forwarders</td>
<td>Lets a user list and view settings for data forwarding. Can be used by TCP and Syslog output admin handlers.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_httpauths</td>
<td>Lets the user view user sessions through the htpauth-tokens endpoint.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_indexer_cluster</td>
<td>Lets the user view the list of indexer clusters as well as indexer cluster objects such as buckets, peers, etc.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_indexerdiscovery</td>
<td>Lets the user view settings for indexer discovery. Also used by indexer discovery handlers.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_inputs</td>
<td>Lets the user view lists of various inputs, including input from files, TCP, UDP, scripts, etc.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>list_introspection</td>
<td>Lets the user read introspection settings and statistics for indexers, search, processors, queues, etc.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_metrics_catalog</td>
<td>Lets the user query for lists of metrics catalog information such as metric names, dimensions, and dimension values.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>list_search_head_clustering</td>
<td>Lets the user list and view search head clustering objects like artifacts, delegated jobs, members, captain, etc.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_search_scheduler</td>
<td>Lets the user view lists of search scheduler jobs.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list_settings</td>
<td>Lets the user list and view server and introspection settings such as the server name, log levels, etc.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability name</td>
<td>What it lets you do</td>
<td>User</td>
<td>Power</td>
<td>Admin</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>list_storage_passwords</td>
<td>Lets the user list and view the /storage/passwords endpoint, lets the user perform GETs. The admin_all_objects capability must be added to the role for the user to perform POSTs to the /storage/passwords endpoint.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>output_file</td>
<td>Lets the user create file outputs, including outputcsv (except for dispatch=t mode) and outputlookup.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>pattern_detect</td>
<td>Lets the user see and use the Patterns tab in the Search view.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>request_remote_tok</td>
<td>Lets the user obtain a remote authentication token, which lets the user perform some distributed peer management and bundle replication and distribute searches to old 4.0.x Splunk instances.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>rest_apps_management</td>
<td>Lets the user edit settings for entries and categories in the python remote apps handler. See restmap.conf for more information.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>rest_apps_view</td>
<td>Lets the user list and view various properties in the Python remote apps handler. See restmap.conf for more information.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>rest_properties_get</td>
<td>Lets the user get information from the services/properties endpoint.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>rest_properties_set</td>
<td>Lets the user edit the services/properties endpoint.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>restart_splunkd</td>
<td>Lets the user restart Splunk Enterprise through the server control handler.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>rtsearch</td>
<td>Lets the user run real-time searches.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>run_debug_commands</td>
<td>Lets the user run debug commands. For example &quot;Summarize&quot;.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>run_multi_phased_searches</td>
<td>Lets the user run searches with the redistribute command, which invokes parallel reduce search processing in distributed search environments. This capability is not assigned to any role by default.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schedule_search</td>
<td>Lets the user schedule saved searches, create and update alerts, and review triggered alert information.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>search</td>
<td>Lets the user run a search. See the Search Manual for more information.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>search_process_config_refresh</td>
<td>Lets the user use the &quot;refresh search-process-config&quot; CLI command to manually flush idle search processes.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>srchFilter</td>
<td>Lets the user manage search filters. See the Search Manual for more information.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>srchIndexesAllowed</td>
<td>Lets the user run search indexes. See the Search Manual for more information.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>srchIndexesDefault</td>
<td>Lets the user set default search indexes.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>srchJobsQuota</td>
<td>Lets the user set search job quotas.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>srchMaxTime</td>
<td>Lets the user set the maximum time for a search.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>use_file_operator</td>
<td>Lets the user use the &quot;file&quot; search operator. The &quot;file&quot; search operator is deprecated.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>web_debug</td>
<td>Lets the user debug Web files.</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Windows-specific capabilities**

If you are running Splunk Enterprise on Windows, additional capabilities are provided to facilitate monitoring.
<table>
<thead>
<tr>
<th>Capability name</th>
<th>What it lets you do</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edit_modinput_admon</code></td>
<td>Edit modular inputs in admon.conf.</td>
</tr>
<tr>
<td><code>edit_modinput_perfmon</code></td>
<td>Edit modular inputs in perfmon.conf.</td>
</tr>
<tr>
<td><code>edit_modinput_winhostmon</code></td>
<td>Add and edit inputs for monitoring Windows host data.</td>
</tr>
<tr>
<td><code>edit_modinput_winnetmon</code></td>
<td>Add and edit inputs for monitoring Windows network data.</td>
</tr>
<tr>
<td><code>edit_modinput_winprintmon</code></td>
<td>Required to add and edit inputs for monitoring Windows printer data.</td>
</tr>
<tr>
<td><code>edit_win_admon</code></td>
<td>(Deprecated)</td>
</tr>
<tr>
<td><code>edit_win_eventlogs</code></td>
<td>Edit windows eventlogs.</td>
</tr>
<tr>
<td><code>edit_win_perfmon</code></td>
<td>(Deprecated)</td>
</tr>
<tr>
<td><code>edit_win_regmon</code></td>
<td>(Deprecated)</td>
</tr>
<tr>
<td><code>edit_win_wmiconf</code></td>
<td>Edit wmi.conf.</td>
</tr>
<tr>
<td><code>list_pdfserver</code></td>
<td>View PDF server files.</td>
</tr>
<tr>
<td><code>list_win_localavailablelogs</code></td>
<td>List all local Windows event logs.</td>
</tr>
<tr>
<td><code>srchTimeWin</code></td>
<td>Set search time limits.</td>
</tr>
<tr>
<td><code>write_pdfserver</code></td>
<td>Write to PDF server files.</td>
</tr>
</tbody>
</table>

### Add and edit roles with Splunk Web

When you create users, you can assign roles that determine the level of access that users have to Splunk Enterprise, and the tasks that they can perform. Splunk Enterprise comes with a set of default roles that you can use. You can also create your own.

For information about roles and how capabilities and permissions are inherited, see About role-based user access.

Custom roles that inherit from Admin or Power users do not automatically inherit management access. For information about granting management access to custom roles, see Add access controls to custom roles.

### Add or edit a role

Create or edit roles for your Splunk platform implementation on the Roles page in Settings.

1. Click **Settings > Access Controls**.
2. Click **Access controls** page click **Roles**.
3. Click **New** or select and edit an existing role. Role names must use lowercase characters only. They cannot contain spaces, colons, or forward slashes.
4. In the **Restrict search terms** section, you can restrict the scope of the searches that users with the role can run.
   You can restrict the search terms they can use, set limits on search time, and set both user-level and role-level concurrent search limits.
   Search term restrictions offer limited security. A user can override some search term restrictions if they create a calculated field that references a field name listed here as a restricted term.
5. In the **Inheritance** section, identify other roles that your role can inherit properties and capabilities from.
   A user assigned to multiple roles inherits properties from the role with the broadest permissions. See Role inheritance in the About role-based user access topic for more information.

21
6. In the **Capabilities** section, choose any individual capabilities you want to provide to this role. See About defining roles with capabilities for more information.

7. In **Indexes searched by default**, specify the indexes that this role will automatically search if no index is specified in the search. You can specify both event and metric indexes. If a user with the role runs an event search without a specified index, the search pulls events from the default event indexes assigned to the role. If a user with the role runs a metrics search without a specified index, the search pulls events from the default metrics indexes assigned to the role.

8. In **Indexes**, select event and metric indexes the user is allowed to search. If you add at least one index, a user with this role will only be able to conduct searches on the index or indexes selected. If you do not specify any indexes at all, the user assigned to the role is able to search all indexes.

9. Click **Save**.

Updates to the search term restrictions for a role do not take effect until you restart your Splunk platform implementation. If you do not restart Splunk, your Splunk platform will not enforce your search term restriction updates.

For more information about restarting Splunk, see Start and Stop Splunk Enterprise in the *Admin Manual*.

**Search filter format**

The "Restrict search terms" field can include any of the following search terms:

- `source::`
- `host::`
- `index::`
- `sourcetype::`
- `eventtype=`
- `search fields`

When you specify search term restrictions for event data, use the `key::value` syntax, when possible, to restrict search terms to indexed fields. Normal field values can be overwritten with user knowledge objects. The `key::value` syntax only applies to indexed fields.

When you specify search term restrictions for metric data, use the `key=value` comparison syntax. Metrics search cannot utilize the `key::value` syntax.

You can use wildcards. Use `OR` to allow multiple terms, or `AND` to make the filter more restrictive.

The search terms cannot include:

- `saved searches`
- `time operators`
- `regular expressions`
- `any fields or modifiers that you can override from the Splunk Web search bar`

**Special syntax rules for search filters of metric data**

Search filters that limit user access to metric data do not follow the same rules as search filters for event data.

Search filters for metric data must use the `key=value` comparison syntax. Metrics search cannot utilize the `key::value` syntax.
Search filters for metric data can filter dimensions and metric names. For example, you can set up search filters for dimensions like os=linux or server=athens and metric names like metric_name=mem.free.

Add and edit roles with authorize.conf

You can add or modify roles by editing authorize.conf. Users are assigned to roles that determine their level of access and the tasks that they can perform. For more information about roles and capabilities, read About role-based user access.

Never edit or delete roles in $SPLUNK_HOME/etc/system/default/authorize.conf. This could break your admin capabilities. Instead edit the local version at $SPLUNK_HOME/etc/system/local/, or your own custom application directory in $SPLUNK_HOME/etc/apps/.

You must reload authentication or restart Splunk Enterprise after making changes to authorize.conf. Otherwise, your new roles will not appear in the Role list. See “Refresh the authentication scheme” in this topic for instructions on how to refresh authentication without restarting the Splunk platform.

For more information, see

- authorize.conf
- About configuration files in the Admin Manual.

Note: Distributed search configurations have slightly different authorization needs. When you use search head clustering, you must make sure that the search heads and the search peers all use the same set of authorize.conf file(s). To make sure your authorization is properly set up for search pooling, see How authorization works in distributed searches.

Add roles

Here’s the syntax for adding roles through $SPLUNK_HOME/etc/system/local/authorize.conf:

```
[role_<roleName>]
<setting> = <value>
<setting> = <value>
...
```

The <roleName> in the stanza header is the name you want to give your role. For example: security, compliance, ninja.

Role names must use lowercase characters only. They cannot contain spaces, colons, semicolons, or forward slashes.

You can include these settings in the role stanza:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Definition</th>
<th>Default</th>
<th>For more information</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;capability&gt; = enabled</td>
<td>You can add any number of capabilities to a role. To add a capability to a role, just set that capability to &quot;enabled&quot;.</td>
<td>disabled</td>
<td>See About defining roles with capabilities.</td>
</tr>
<tr>
<td>importRoles = &lt;role&gt;;&lt;role&gt;;...</td>
<td>When set, the current role inherits the capabilities from &lt;role&gt;. Members assigned to multiple roles inherit properties from the role with the broadest permissions. If you enter multiple roles, separate them with semicolons.</td>
<td>No roles set.</td>
<td>See Role inheritance.</td>
</tr>
<tr>
<td>Setting</td>
<td>Definition</td>
<td>Default</td>
<td>For more information</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------------------</td>
</tr>
<tr>
<td>srchFilter - &lt;search_string&gt;</td>
<td>This setting lets you define detailed data access controls. Users with this role will have their searches filtered by this expression.</td>
<td>No filters are set.</td>
<td>See Search filter format.</td>
</tr>
<tr>
<td>srchTimeWin - &lt;string&gt;</td>
<td>The maximum time span in seconds allowed for a search executed by a user in this role.</td>
<td>Not set. Search times are not limited.</td>
<td></td>
</tr>
<tr>
<td>srchDiskQuota - &lt;int&gt;</td>
<td>The maximum amount of disk space (MB) that can be used by search jobs performed by a user assigned to this role.</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>cumulativeSrchJobsQuota - &lt;number&gt;</td>
<td>The maximum number of concurrently running historical searches that all members of this role can have. For this setting to apply, you must also set enable_cumulative_quota = true in limits.conf.</td>
<td>Not set.</td>
<td></td>
</tr>
<tr>
<td>cumulativeRTSrchJobsQuota - &lt;number&gt;</td>
<td>The maximum number of concurrently running real-time searches that all members of this role can have. For this setting to apply, you must also set enable_cumulative_quota = true in limits.conf.</td>
<td>Not set.</td>
<td></td>
</tr>
<tr>
<td>srchJobsQuota - &lt;int&gt;</td>
<td>The maximum number of concurrently running searches a member of this role can have.</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>rtSrchJobsQuota - &lt;number&gt;</td>
<td>The maximum number of concurrently running real-time searches a member of this role can have.</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>srchIndexesDefault - &lt;string&gt;</td>
<td>A semicolon-delimited list of default indexes to search when no index is specified. The list can include both event indexes and metric indexes. When the user runs an event search that does not specify an index, the search runs over the default event indexes. When the user runs a metrics search that does not specify an index, the search runs over the default metrics indexes. You can wildcard your entries. However, the wildcard '<em>' does not match internal indexes. To match internal indexes, start with '<em>':. All internal indexes are represented by '</em></em>'.</td>
<td>Not set.</td>
<td></td>
</tr>
</tbody>
</table>

24
Setting | Definition | Default | For more information
--- | --- | --- | ---
srchIndexesAllowed = <string> | A semicolon-delimited list of indexes this role is allowed to search. The list can include both event and metrics indexes. You can wildcard your entries. However the wildcard '*' will not match internal indexes. To match internal indexes, start with '_'. All internal indexes are represented by '_*'. | Not set. | 

Search filter format

The `srchFilter` field can include any of the following search terms:

- `source=`
- `host= `and host tags
- `index= `and index names
- `eventtype= `and event type tags
- `sourcetype=`
- `search fields`
- `wildcards`
- `use OR to use multiple terms, or AND to make searches more restrictive.`

The search terms cannot include:

- `saved searches`
- `time operators`
- `regular expressions`
- `any fields or modifiers Splunk Web can overwrite`

**Example of creating a role in `authorize.conf`**

This example creates the role "ninja", which inherits capabilities from the default "user" role. ninja has almost the same capabilities as the default "power" role, except it cannot schedule searches. In addition:

- The search filter limits ninja to searching on `host=foo`.
- ninja is allowed to search all public indexes (those that do not start with underscore) and will search the indexes `mail` and `main` if no index is specified in the search.
- `ninja` is allowed to run 8 search jobs and 8 real-time search jobs concurrently. (These counts are independent.)
- `ninja` is allowed to occupy up to 500MB total space on disk for all its jobs.

```
[role_ninja]
rtsearch = enabled
importRoles = user
srchFilter = host=foo
srchIndexesAllowed = *
srchIndexesDefault = mail;main
srchJobsQuota = 8
rtSrchJobsQuota = 8
srchDiskQuota = 500
```
Refresh the authentication scheme

After you make changes to `authentication.conf`, you must refresh the authentication scheme to have the changes take effect. You can do this with either Splunk Web or the CLI. Refreshing the authentication scheme does not log users off of the system.

A restart of the Splunk platform also reloads authentication, but it also logs users off of the system, and there is no requirement to restart the platform to refresh authentication.

Refresh the authentication scheme using Splunk Web

1. From the system bar, click **Settings > Authentication Methods**.
2. Click **Reload authentication configuration**. This refreshes the authentication caches but does not boot current users from the system.

Refresh the authentication scheme using the CLI

1. On the Splunk platform instance where you want to reload authentication, open a shell prompt or PowerShell window.
2. Switch to the `$SPLUNK_HOME/bin` directory.
3. Use the CLI command `. /splunk reload auth`.

```
./splunk reload auth
```

The authentication scheme cache resets immediately.

Setting access to manager consoles and apps

The `local.meta` file is handy for allowing you to grant and restrict access to certain parts of your Splunk instance. For example, you can:

- Restrict users in custom roles to a specific app
- Give users in custom roles the ability to access admin level features

Granting admin roles to users

Some management abilities that belong to the Admin role are unique to that specific label. These abilities are not automatically inherited from the Admin role when you configure a role in Splunk Web or `authorize.conf`.

For example, say you want to create a custom role that inherits all of the Admin abilities but has limited access to your search jobs. To do this, you would create a new role called "SpecialAdmin" and set it to inherit all of the capabilities of an Admin as described in About defining roles with capabilities then set your search limits About configuring role-based user access.

Restricting access to specific apps

The `local.meta` file can also be used to restrict access.

For example, say you want to allow a user access to only one dashboard view. To accomplish this, you could create an app for that view and assign the user's role to that app. You should use local.meta to permit the role to view that app.
How to add and remove access via local.meta files

You can give or restrict access by editing the local.meta file to add the new role wherever you want it.

1. Locate the local.meta file. If you are editing access for the main search page (ie, the manager controls), look in $SPLUNK_HOME/etc/system/metadata/. If you want to edit access to a particular app, look in $SPLUNK_HOME/etc/apps/<app_name>/metadata/. If the directory for the desired location does not contain the file, you can copy the default version default.meta and rename it.

   **Note:** Do NOT edit the default.meta file directly, you may need the default values in that file at a future time.

2. In the local.meta file, add the name of the new role to the stanza that corresponds with the desired access.

<table>
<thead>
<tr>
<th>Default stanza</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>[manager/accesscontrols]</td>
<td>Allow all users to read this app's contents, or access functions in the Splunk Manager page, depending on the directory you are in. Unless overridden by other metadata, allows only admin and power users to share objects into this app.</td>
</tr>
<tr>
<td>access = read : [ * ], write : [ admin, power ]</td>
<td></td>
</tr>
<tr>
<td>[views] [manager/accesscontrols] access</td>
<td>Determines the access controls for the Manager page access.</td>
</tr>
<tr>
<td>= read : [ * ], write : [ admin ]</td>
<td></td>
</tr>
</tbody>
</table>

3. When you have made all of your changes, restart Splunk Enterprise.

**Examples**

**Example 1:** A new role called "usermanager" only inherits capabilities from a user and has no searches or indexes inherited. The intent is to create a role that has no access to data and is solely used to create and manage user accounts.

To create this role you would edit the following stanza:

```plaintext
[manager/accesscontrols]
access = read : [ admin ], write : [ admin ]
```

To include the following:

```plaintext
[manager/accesscontrols]
access = read : [ admin, usermanager ], write : [ admin, usermanager ]
```

You have just given "usermanager" the ability to see and edit stuff in the "Access controls" pages in Manager.

**Example 2:** To enable the role "userview," to access but not edit the pages, only add the role to the read value:

```plaintext
[manager/accesscontrols]
access = read : [ admin, userview, usermanager ], write : [ admin, usermanager ]
```

You can also grant access to read the manager pages to EVERY role using the wildcard:

```plaintext
[manager/accesscontrols]
access = read : [ * ], write : [ admin ]
```

**Example 3:** You want to have a subset of users who can only read sales data that you specify. To accomplish this you can create an app for the dashboard and then create a new role "salesusers."

In the local.meta file in your app directory (remember that you can create one from the default.meta file), you then edit the following stanza:
as access = read : [ * ], write : [ * ]

to read:

as access = read : [ salesusers ], write : [ admin ]

Find existing users and roles

To locate the existing user or role in Splunk Web:

1. In the main menu click System > Access Controls.
2. Click Users or Roles to choose which entity you want to search.

By default, all fields are searched for the specified string. To search a particular field, specify the name of the field. Note that Splunk Search supports wildcards. For example:

- To search only email addresses:
  "email=<email address or address fragment>:
- To search only the "Full name" field:
  "realname=<name or name fragment>.
- To search for users in a given role:
  "roles=".

Delete all user accounts

Remove all the user data (user accounts) from your Splunk Enterprise installation by typing ./splunk clean followed by the userdata argument. This deletes all the user accounts other than the default user accounts (admin, power, user).

Removing user data is irreversible. If you accidentally delete user data, you must re-add the accounts manually.

To remove all of the user accounts in the system:

./splunk clean userdata

To remove the user accounts in the system and skip the confirmation prompt:

./splunk clean userdata -f

Secure access for Splunk knowledge objects

As you use Splunk Enterprise, you create a variety of knowledge objects such as event types, tags, lookups, field extractions, workflow actions, and saved searches. Splunk Web lets you restrict and expand access to knowledge objects within your Splunk implementation. You can use it to:

- Make an object available to users of all apps.
- Make an object available to users of a particular app.
- Restrict object access by user role.
- Disable or delete objects.
• Allow users to share or delete objects they do not own.

For more information about securing your knowledge objects, see Manage knowledge object permissions and Disable or delete knowledge objects in the Knowledge Manager Manual.

Use Access Control Lists

To help secure your Splunk configuration, use the Splunk Enterprise Access Control Lists (ACLs) to limit the IP addresses that can access various parts of your networks.

To configure ACLs, you edit server.conf and inputs.conf to specify the IP addresses that will be accepted or rejected for various communications.

How to set up ACLs

The addresses are separated by commas or spaces. You can provide the addresses in the following formats:

- A single IPv4 or IPv6 address. For example: 10.1.2.3, fe80::4a3.
- A CIDR block of addresses. For example: 10/8, fe80:1234/32.
- A DNS name, possibly with an * used as a wildcard, for example: myhost.example.com, *.splunk.com.
- A single * which matches anything (this is the default value).

To add addresses that you wish to include, you add the addresses in one of the formats described below. To exclude an address you prefix the address with '!'.

Rules are applied in order, and the first one to match is used. For example, !10.1/16, * will allow connections from everywhere except the 10.1.*.* network.

Where to set up ACLs

You can secure IP addresses for the following connections by editing the [Accept from] value:

- To instruct a node to only accept replicated data from other nodes with specific IPs, edit the httpServer stanza in server.conf. If you set this attribute, you must make sure that you include the IP addresses of all other peers in the cluster. For more information about clusters, see "About clusters and index replication" For more information about editing server.conf, see server.conf.

- To restrict TCP communications to specific IP addresses, edit the tcp stanza in inputs.conf. Be careful, as this will overwrite the output values in server.conf if the information conflicts.

- To restrict TCP communications that use SSL to specific IP addresses, edit the tcp-ssl stanza in inputs.conf.

- To restrict your indexer to accept data only from forwarders with specific IP addresses, edit the splunktcp stanza in inputs.conf. This prevents someone from spoofing your forwarders and possibly corrupting your data.

- If your forwarder to indexer communications are secured with SSL, edit the splunktcp-ssl stanza in inputs.conf to restrict your indexer to only accept data from forwarders with specific IP addresses.

- To restrict UDP communications to specific IP addresses, edit the UDP stanza in inputs.conf.

29
For more information about editing inputs.conf, see inputs.conf
Native Splunk Enterprise authentication

Set up Splunk authentication

Splunk authentication allows you to easily set up users within your system. Splunk authentication always takes precedence over any external authentication systems. This is the order in which users are authenticated:

1. Splunk authentication.

2. LDAP or Scripted authentication (if enabled). For more information, see "Set up user authentication with LDAP" and "Set up user authentication with external systems."

Note: LDAP and scripted authentication cannot be used together.

You can create new users and assign those users to roles with a role-based access control system in two ways:

- Use Splunk Web to create users and assign roles. For more information see "Configure users with Splunk Web."
- Use the CLI to create users and then assign them to roles with Splunk Web. For more information see "Configure users with the CLI."

Important naming guidelines when creating users and roles

Usernames stored in native authentication cannot contain spaces, colons, or forward slashes. Names are case-insensitive, for example: "Jacque", "jacque", "JacQue" are all the same to Splunk authentication.

Role names must use lowercase characters only. They cannot contain spaces, colons, or forward slashes.

Configure users with Splunk Web

To configure users and roles In Splunk Web:

1. Navigate to Settings > Users and Authentication > Access controls.
2. Click Users.
3. Click New or select an existing user to edit.
4. Specify or change the information for the user. You can specify the user's:
   - full name.
   - email address.
   - time zone. This allows users to view events and other information in their own time zone.
   - default app. This overrides the default app inherited from the user's role.
   - password.
5. Assign the user to an existing role or roles and click Save.

You can also create a role specifically for a user, defining exactly what access that user has to Splunk Enterprise. You can then assign the user to that role. For information about roles, read "About role-based user access."

For information about managing user settings, see the Splunk Enterprise Administration Guide.
Configure users with the CLI

In the CLI, use the \texttt{add user} command. Here are some examples:

- To add a new administrator user with the password "changeme2":

  \texttt{./splunk add user admin2 -password changeme2 -role admin -auth admin:changeme}

- To change an existing user's password to "fflanda":

  \texttt{./splunk edit user admin -password fflanda -role admin -auth admin:changeme}

\textbf{Important}: Passwords with special characters that would be interpreted by the shell (for example '$' or '!') must be either escaped or single-quoted. For example:

\texttt{./splunk edit user admin -password ‘fflanda$’ -role admin -auth admin:changeme}

or

\texttt{./splunk edit user admin -password fflanda\$ -role admin -auth admin:changeme}

\textbf{Add a user to a role with Splunk Web}

You can add a user to a default role or to a custom role you create yourself. For more information, see "About role-based user access."

To add a user or users to a role with Splunk Web:

1. Click \textbf{Settings > Access Control > Access Controls} in the main menu.

2. Click \textbf{Users}.

3. Edit an existing user or create a new one.

4. Choose which role to map to from the \textbf{Role} list. Any custom roles you have created in authorize.conf will be listed here.
Authentication with LDAP

Set up user authentication with LDAP

Splunk Enterprise supports three types of authentication systems:

- Splunk authentication described in "Set up user authentication with Splunk's built-in system."
- LDAP, described in the topic you're now reading.
- A scripted authentication API for use with an external authentication system, such as PAM or RADIUS, described in "Set up user authentication with external systems."

About configuring LDAP authentication for Splunk Enterprise

Splunk Enterprise allows user and role configuration for LDAP users and groups. You can configure one or many LDAP servers and map users and user groups from your servers to Splunk roles.

For more information about configuring multiple LDAP servers, see "How Splunk works with multiple LDAP servers."

Before you configure LDAP, take a look at "LDAP prerequisites and considerations."

How to configure LDAP authentication

These are the main steps to configure Splunk Enterprise to work with LDAP:

1. Configure one or more LDAP strategies (typically, you configure one strategy per LDAP server).

2. Map your LDAP groups to one or more Splunk roles.

3. If you have multiple LDAP servers, specify the connection order of their servers.

You can perform these steps in Splunk Web or by editing the configuration file. See "Configure LDAP with Splunk Web" or "Configure LDAP with the configuration file" for more information.

Authentication precedence

Splunk authentication takes precedence over any external systems. This is the order in which Splunk Enterprise authenticates a user:

1. Splunk authentication

2. LDAP or scripted authentication, if one of these methods is enabled. For more information about scripted authentication see "Set up user authentication with external systems."

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has around LDAP authentication with Splunk.
Manage Splunk user roles with LDAP

To configure Splunk Enterprise to use LDAP authentication, first create a Splunk strategy for each LDAP server and then map Splunk roles to that server's groups. When a user attempts to log in, Splunk Enterprise queries the server(s) to find the user. It grants the user permissions based on any roles associated with the LDAP groups the user is a member of.

When it comes to changing a user’s permissions, you have several options:

- To change the permissions for a group of users, you can remap the LDAP group to a different Splunk role. You can also update the role itself to specify a different set of permissions for it. You do this on Splunk Enterprise.
- To change the permissions for an individual user, you can move the user to an LDAP group mapped to a different Splunk role. You do this on the LDAP server.

Here are some other user management activities:

- To add a user to a Splunk role: First, on Splunk Web, make sure that you’ve mapped the Splunk role to an LDAP group. Then, on your LDAP server, add the user to that LDAP group.
- To remove a user from a Splunk role: On your LDAP server, remove the user from the corresponding LDAP group.

A user can have membership in several roles. In that case, the user has access to all the capabilities available for any of those roles. For example, if the user is a member of both the docs and eng groups, and docs is mapped to "user" and eng is mapped to "admin", the user obtains all permissions assigned to both the "user" or "admin" roles.

Note: Splunk Enterprise checks LDAP membership information when a user attempts to log in. You do not need to reload the authentication configuration when adding or removing users.

LDAP prerequisites and considerations

Before configuring LDAP for authentication with Splunk, make the preparations described in this topic.

Determine your User and Group Base DN

Before you map your LDAP settings to Splunk settings, figure out your user and group base DN, or distinguished name. The DN is the location in the directory where authentication information is stored.

If group membership information for users is kept in a separate entry, enter a separate DN identifying the subtree in the directory where the group information is stored. Users and groups will be searched recursively on all the subnodes under this DN. If your LDAP tree does not have group entries, you can set the group base DN to the same as the user base DN to treat users as their own group. This requires further configuration, described later.

If you are unable to get this information, contact your LDAP Administrator for assistance.

Note: For best results when integrating Splunk Enterprise with Active Directory, place your Group Base DN in a separate hierarchy than the User Base DN.

Additional considerations

When configuring Splunk Enterprise to work with LDAP, note the following:
Entries in Splunk Web and authentication.conf are case sensitive.

Any user created locally through Splunk native authentication will have precedence over an LDAP user of the same name. For example, if the LDAP server has a user with a username attribute (for instance, cn or uid) of 'admin' and the default Splunk user of the same name is present, the Splunk user will win. Only the local password will be accepted, and upon login the roles mapped to the local user will be in effect.

The number of LDAP groups Splunk Web can display for mapping to roles is limited to the number your LDAP server can return in a query. You can use the Search request size limit and Search request time limit settings to configure this.

- To prevent Splunk from listing unnecessary groups, use the groupBaseFilter. For example:
  ```
groupBaseFilter = (|(cn=SplunkAdmins)(cn=SplunkPowerUsers)(cn=Help Desk))
```

- If you must role map more than the maximum number of groups, you can edit authentication.conf directly. In this example, "roleMap_AD" specifies the name of the Splunk strategy. Each attribute/value pair maps a Splunk role to one or more LDAP groups:

  ```
  [roleMap_AD]
  admin = SplunkAdmins1;SplunkAdmins2
  power = SplunkPowerUsers
  user = SplunkUsers
  ```

Splunk always uses LDAP protocol version 3, aka v3.

Secure LDAP with TLS certificates

Splunk uses OpenLDAP and OpenSSL. You can leverage both tools to secure your LDAP authentication with certificates. For more information on creating and managing certificates, see the OpenSSL documentation.

The following examples are certificate configurations for LDAP. For more information about ways you can configure certificates in LDAP, see the OpenLDAP documentation at http://www.openldap.org/doc/admin24/tls.html:

**LDAP server configuration**

```text
TLSCACertificateFile <filename>: the PEM-format file containing certificates for the CA's that slapd will trust, including the certificate for the CA that signed the server certificate. Multiple certificates can be appended to the file in no particular order.

TLSCertificateKeyFile <filename>: the file that contains the private key that matches the certificate stored in the TLSCertificateFile file.

TLSCipherSuite <cipher-suite-spec>: ciphers will be accepted and the preference order. <cipher-suite-spec> should be a cipher specification for OpenSSL. Use "openssl ciphers -v ALL" for a list of available cipher specifications.

TLSRandFile <filename>: the file to obtain random bits from when /dev/urandom is not available. If the system provides /dev/urandom then this option is not needed, otherwise a source of random data must be configured.

TLSephemeralDHParamFile <filename>: the file that contains parameters for Diffie-Hellman ephemeral key exchange.

TLSVerifyClient { never | allow | try | demand }: specifies what checks to perform on client certificates in an incoming TLS session, if any. This option is set to never by default, in which case the server never asks the client for a certificate.
```
**LDAP client configuration**

This directive specifies the file that contains the client certificate. This is a user-only directive and can only be specified in a user's .ldaprc file.

**TLS_KEY <filename>** specifies the file that contains the private key that matches the certificate stored in the TLS_CERT file. The same constraints mentioned for TLSCertificateKeyFile apply here. This is also a user-only directive.

**TLS RandFile <filename>** the same as the server's TLSRandFile option.

**TLS_REQCERT { never | allow | try | demand }**

Note that if you host two or more LDAP servers, you may not want to use self-signed certificates, since each client will have to be configured to work with each certificate. In such a case it would be easier to create a certificate authority to sign your server certificates.

**How Splunk Enterprise works with multiple LDAP servers**

Splunk Enterprise can search against multiple LDAP servers when authenticating users. To configure multiple LDAP servers, you set up multiple LDAP "strategies," one for each LDAP server.

After you create your strategies, you can specify the order in which you want Splunk Enterprise to query the strategies when searching for LDAP users. If you do not specify a search order, Splunk Enterprise assigns a default "connection order" based on the order in which the strategies are created.

For more about the steps to configure LDAP strategies, see Configure LDAP with Splunk Web or Configure LDAP with the configuration file for more information.

**How connection order works during a search**

During authentication, Splunk Enterprise searches based on the strategies you created for your servers in the specified connection order. After Splunk Enterprise locates the user on a server, it stops searching and takes those credentials. If the user also has credentials on a server later in the search order, those credentials are ignored.

For example, assume that you configure and enable three strategies in this order: A, B, C. Splunk Enterprise will search the servers in that same order: A, B, C. If it finds the user on A, it stops looking. Even if the user also exists on B and C, Splunk Enterprise will only use A's credentials for that user. If Splunk Enterprise does not find the user on A, it searches the remaining servers: first B, then C.

If you later disable strategy A, Splunk Enterprise will search the remaining strategies in the order: B, C.

You can change the connection order at any time by editing the strategies' properties in Splunk Web or by changing the order of the strategies in the authSettings attribute, as described in the authentication.conf spec file. For more information about editing this file for LDAP, see Edit authentication.conf.

**Important:** Any user created locally through Splunk authentication has precedence over an LDAP user of the same name. See About user authentication, for details.
Configure LDAP with Splunk Web

This section describes how to configure LDAP through Splunk Web. If you want to configure LDAP by directly editing `authentication.conf`, see Configure LDAP with the configuration file.

There are three main steps to configuring LDAP with Splunk Web:

1. Create an LDAP strategy.
2. Map LDAP groups to Splunk roles.
3. Specify the connection order (for multiple LDAP servers only)

Create an LDAP strategy

To create an LDAP strategy:

1. Click Settings > Users and authentication > Authentication Methods.
2. Check LDAP.
3. Click Configure Splunk to use LDAP. This takes you to the LDAP strategies page.
4. Click New. This takes you to the Add new page.
5. Enter an LDAP strategy name for your configuration.
6. Enter the Host name of your LDAP server. Be sure that your Splunk Server can resolve the host name. Note: At this time, IPv6 address formats for Windows are not supported.
7. Enter the Port that Splunk Enterprise will use to connect to your LDAP server.
   - By default LDAP servers listen on TCP port 389.
   - LDAPS (LDAP with SSL) defaults to port 636.
8. To turn on SSL, check SSL enabled.
   - This setting is recommended for security.
   - You must also have SSL enabled on your LDAP server.
9. Enter the Bind DN.
   - This is the distinguished name used to bind to the LDAP server.
   - This is typically, but not necessarily, the administrator. This user needs to have read access to all LDAP user and group entries you want to retrieve.
   - Leave blank if anonymous bind is sufficient.
10. Enter and confirm the Bind DN password for the binding user.
11. Specify the User base DN. You can specify multiple user base DN entries by separating them with semicolons.
• Splunk Enterprise uses this attribute to locate user information.
• You must set this attribute for authentication to work.

12. Enter the **User base filter** for the object class you want to filter your users on.

   • This is recommended to return only applicable users. For example: (department=IT).
   • Default value is empty, meaning no user entry filtering.

13. Enter the **User name attribute** that contains the user name.

   • The username attribute cannot contain white spaces.
   • In Active Directory, this is typically sAMAccountName, but you can also authenticate on other attributes, like cn.
   • The value uid should work for most other configurations.

14. Enter the **Real name attribute** (common name) of the user.

   • Typical values are displayName or cn (common name).

15. Enter an **Email attribute**

16. Enter the **Group mapping attribute**.

   • This is the user attribute that group entries use to define their members.
   • The default is dn for active directory; set this attribute only if groups are mapped using some other attribute besides user DN.
   • For example, a typical attribute used to map users to groups is dn.

17. Enter the **Group base DN**. You can specify multiple group base DN entries by separating them with semicolons.

   • This is the location of the user groups in LDAP.
   • If your LDAP environment does not have group entries, you can treat each user as its own group:
     ♦ Set groupBaseDN to the same value as userBaseDN. This means you will search for groups in the same place as users.
     ♦ Next, set the groupMemberAttribute and groupMappingAttribute to the same attribute as userNameAttribute. This means the entry, when treated as a group, will use the username value as its only member.
     ♦ For clarity, you should probably also set groupNameAttribute to the same value as userNameAttribute.

**Note:** For best results when integrating Active Directory, place your Group Base DN in a separate hierarchy than the User Base DN.

18. Enter the **Static group search filter** for the object class you want to filter your static groups on.

   • This is recommended to return only applicable groups. For example:
     (|(objectclass=groupofNames)(objectclass=groupofUniqueNames))
   • Default value is empty, meaning no static group entry filtering.

19. Enter the **Group name attribute**.

   • This is the group entry attribute whose value stores the group name.
   • This is usually cn.
20. Enter the Static member attribute.

- This is the group attribute whose values are the group's members.
- This is typically `member`, `uniqueMember`, or `memberUid`.

21. To expand nested groups, check Nested groups.

- This controls whether Splunk Enterprise will expand nested groups using the 'memberof' attribute. Only check this if you have nested groups that leverage the 'memberof' attribute to resolve their members. On OpenLDAP, you need to explicitly enable the 'memberof' overlay.

22. Enter the Dynamic group search filter to retrieve dynamic groups, if any.

- This must match the object class of your dynamic groups definition to ensure that those groups get returned to Splunk. For example: `(objectclass=groupOfURLs)`
- Default value is empty, meaning Splunk Enterprise will not look for dynamic group entries during authentication and authorization.

23. Enter the Dynamic member attribute.

- This is the group attribute that uses the form of an LDAP search URL (such as `ldap:///o=Acme,c=US??sub?(objectclass=person)`) to define its members.
- This is typically `memberURL`.

24. If you check Advanced settings, there are several additional options you can set:

- Enable referrals with anonymous bind only.
  - This setting is on by default. Turn this off if you have no need for referrals.
  - Splunk can chase referrals with anonymous bind only. You must also have anonymous search enabled on your LDAP server.
  - If you are seeing long LDAP search timeouts (likely in Active Directory) and "Operations error" in `splunkd.log` for ScopedLDAPConnection, the issues might be related to referrals.
- Search request size limit
  - To avoid performance-related issues, you can set the search request size limit. Splunk Enterprise will then request that the LDAP server return the specified maximum number of entries in response to a search request. In a large deployment with millions of users, setting this limit to a high value could result in a long response, depending on the search filter set in the LDAP strategy configuration. If this limit is reached, `splunkd.log` should contain a size limit exceeded message.
  - You should set the `search request time limit` and `search request size limit` values in conjunction with the `splunkweb` timeout property, described in Configure user session timeouts. If you have a group that is not showing up in the Splunk console, it was likely excluded due to one of these limits. Tune these properties as needed.
  - To set the request size limit higher than 1000, you must also edit `max_users_to_precache in limits.conf` to accommodate the number of users you set for your request size limit.
- Search request time limit
  - To avoid performance-related issues, you can set the search request time limit. Splunk Enterprise will then request that the LDAP server complete its search within the specified number of seconds. In a large deployment with millions of users, setting this limit to a high value could cause Splunk Web to timeout. If this limit is reached, `splunkd.log` should contain a time limit exceeded message.
  - You should set the `search request time limit` and `search request size limit` values in conjunction with the `splunkweb` timeout property, described in Configure user session timeouts. If you have a group that is not showing up in the Splunk console, it was likely excluded due to one of these limits. Tune these
properties as needed.

- **Network socket timeout**
  - This property is used to break the loop in the authentication chain when one of the LDAP servers in a multiple strategy configuration is unreachable due to network congestion or otherwise takes too long to respond. After waiting the specified number of seconds, the authentication process will continue with the next available strategy, if any.
  - When an LDAP strategy is first created, Splunk Enterprise validates the LDAP server/port and other parameters. If the LDAP server is down or one of the parameters cannot be validated at that time, the LDAP strategy does not get created.

25. Click **Save**.

**Map your new LDAP groups to Splunk roles**

Once you have configured Splunk Enterprise to authenticate via your LDAP server, map your LDAP groups to Splunk roles. If you do not use groups, you can map users individually.

**Note:** You can map either users or groups, but not both. If you are using groups, all users must be members of an appropriate group. Groups inherit capabilities from the highest level role they’re a member of.

All users are visible in the **Users** page in Splunk Manager. To assign roles to groups in Splunk Web:

1. From the main menu, select **System > Users and Authentication > Access Controls**.

2. In the **Access Controls** page, click **Authentication method**.

3. Select the **LDAP** radio button then click **Configure Splunk to use LDAP and map groups**. This takes you to the **LDAP strategies** page.

4. Click **Map groups** in the Actions column for a specific strategy. This takes you to the **LDAP Groups** page. You can use the search field in the upper right corner of the page to qualify the list of groups; for example, to search for groups containing specific users.

5. Click on a group name. This takes you the mapping page, which includes a list of available roles and a list of LDAP users for that group.

6. To map a role to a group, click the arrow to the left of a role in the “Available Roles” list. This moves the group into the “Selected Roles” list. You can map multiple roles to the group.

7. Click **Save**. This takes you back to the **LDAP Groups** page.

8. Repeat the process for each group that you want to assign Splunk roles to.

**Specify the server connection order**

If you have enabled multiple LDAP strategies, you can specify the order in which Splunk Enterprise searches their servers to find a user, as described in How Splunk works with multiple LDAP servers.

By default, Splunk Enterprise searches the servers in the order in which they were enabled. To change the connection (search) order, you need to edit the properties for each strategy individually:
1. From the main menu, select **System > Users and Authentication > Access Controls**.

2. Click **Authentication method**.

3. Select the **LDAP** radio button.

4. Click **Configure Splunk to use LDAP and map groups**. This takes you to the **LDAP strategies** page.

5. Click on the strategy whose connection order you want to specify. This takes you to the properties page for that strategy.

6. Edit the **Connection order** field near the top of the page. This field appears only if multiple strategies are enabled.

   **Note**: The **Connection order** field does not appear when you initially create the strategy. It only appears when you later edit its properties. Also, the field will be grayed out if the strategy has been disabled.

7. Click **Save**.

8. Repeat the process for any other enabled strategy whose connection order you want to change.

### Map LDAP groups to Splunk roles in Splunk Web

If you have configured Splunk Enterprise to authenticate via your LDAP server, you can map your LDAP groups to Splunk roles. If you do not use groups, you can also map LDAP users individually.

For information about setting up LDAP groups in Splunk Web, see "Configure LDAP with Splunk Web" in this manual.

**Note**: You can map either users or groups, but not both. If you are using groups, all users you want to access Splunk Enterprise must be members of an appropriate group. Groups inherit capabilities from the highest level role they’re a member of.

All users are visible in the **Users** page in Splunk Manager. To assign roles to groups in Splunk Web:

1. Click **Settings** in Splunk Web.

2. In the **Users and authentication** section, click **Access controls**.

3. Click **Authentication method**.

4. Select the **LDAP** radio button.

5. Click **Configure Splunk to use LDAP and map groups**. This takes you to the **LDAP strategies** page.

6. Click **Map groups** in the Actions column for a specific strategy. This takes you to the **LDAP Groups** page. You can use the search field in the upper right corner of the page to qualify the list of groups; for example, to search for groups containing specific users.

7. Click on a group name. This takes you the mapping page, which includes a list of available roles and a list of LDAP users for that group.
8. To map a role to a group, click the arrow to the left of a role in the "Available Roles" list. This moves the group into the "Selected Roles" list. You can map multiple roles to the group.

9. Click Save. This takes you back to the LDAP Groups page.

10. Repeat the process for each group that you want to assign Splunk roles to.

Configure LDAP with the configuration file

As an alternative to using Splunk Web to configure LDAP, you can directly edit the authentication.conf file.

This example steps you through the process of setting up authentication.conf. If you prefer to configure LDAP with Splunk Web, see Configure LDAP with Splunk Web.

Note: If you configure LDAP authentication and decide later to return to using the default Splunk authentication, the simplest way is to move the existing authentication.conf file out of the way (for example, by renaming it to authentication.conf.disabled) and restart Splunk Enterprise.

You can see some more examples at the end of the authentication.conf spec file.

Edit authentication.conf in $SPLUNK_HOME/etc/system/local/. For information on configuration files in general, see About configuration files In the Admin Manual.

Set authentication type and strategy name(s)

By default, Splunk Enterprise uses Splunk authentication. Change the type to LDAP in the [authentication] stanza:

[authentication]
authType = LDAP
authSettings = ldaphost1,ldaphost2

Note the following:

• Turn on LDAP by setting authType = LDAP.
• The authSettings attribute identifies one or more LDAP strategies. Each strategy has its own stanza.

Configure LDAP strategy stanzas

Each LDAP strategy needs its own stanza. Map the LDAP values to attribute/value pairs in the strategy's stanza.

Note: Splunk Enterprise does not support IPv6 address formats for Windows.

Here's an example stanza for the "ldaphost1" strategy, specified earlier in the authSettings attribute:

[ldaphost1]
host = ldaphost1.domain.com
port = 389
SSLEnabled = 0
bindDN = cn=bind_user
bindDNpassword = bind_user_password
groupBaseDN = ou=Groups,dc=splunk,dc=com
groupBaseFilter = (objectclass=*)
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
realNameAttribute = displayName
userBaseDN = ou=People,dc=splunk,dc=com
userBaseFilter = (objectclass=*)
userNameAttribute = uid

Note: For best results when integrating Active Directory, place your Group Base DN in a separate hierarchy than the User Base DN.

SSL

If you have enabled SSL for your LDAP strategy, make sure the following minimum settings are present in ldap.conf

TLS_REQCERT demand
TLS_CACERT <path to cert, for example: /opt/splunk/etc/auth/LDAProotcert.crt>
TLS_CIPHER_SUITE <your cipher suite>

Configure multiple LDAP strategies

Splunk Enterprise can search across multiple LDAP servers, as described in How Splunk works with multiple LDAP servers. To configure this, set the authSettings attribute to a comma-separated list of all strategies, in the order in which you want Splunk Enterprise to query them. Then, specify separate stanzas for each strategy.

Map groups to roles

To map Splunk roles to a strategy's LDAP groups, you need to set up a roleMap stanza for that strategy. Each strategy requires its own roleMap stanza. This example maps roles for groups in the "ldaphost1" strategy. The syntax is <Splunk RoleName> = <LDAP group string>:

```
[roleMap_ldaphost1]
admin = SplunkAdmins
itusers = ITAdmins
```

Map users directly to roles

If you need to map users directly to Splunk roles, you can do so by setting the groupBaseDN to the value of userBaseDN. Also, set the attributes for groupMappingAttribute, groupMemberAttribute, and groupNameAttribute to the same attribute as userNameAttribute. For example:

```
[supportLDAP]
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = ########
groupBaseDN = ou=People,dc=splunksupport,dc=com
groupBaseFilter = (objectclass=*)
groupMappingAttribute = uid
groupMemberAttribute = uid
groupNameAttribute = uid
host = supportldap.splunksupport.com
port = 389
realNameAttribute = cn
userBaseDN = ou=People,dc=splunksupport,dc=com
```
userBaseFilter = (objectclass=*)
userNameAttribute = uid

[roleMap_supportLDAP]
admin = rlee;bsmith

Map LDAP groups and users to Splunk roles in the configuration files

Once you’ve set up LDAP authentication and users, you can map your LDAP groups and users to roles in Splunk Web. To set up LDAP for Splunk Enterprise, see Configure LDAP with the configuration file in this manual.

As an alternative to using Splunk Web to map roles, you can directly edit your authentication.conf contained in $SPLUNK_HOME/etc/system/local/. There are further examples at the end of the authentication.conf spec file.

For information on configuration files in general, see About configuration files in the Admin Manual.

Map groups to roles

To map Splunk roles to a strategy’s LDAP groups, you need to set up a roleMap stanza for that strategy. Each strategy requires its own roleMap stanza. This example maps roles for groups in the “ldaphost1” strategy. In your authentication.conf file in $SPLUNK_HOME/etc/system/local/:

[roleMap_ldaphost1]
admin = SplunkAdmins
itusers = ITAdmins

Map users directly to roles

If you need to map users directly to Splunk roles, you can do so by setting the groupBaseDN setting in authentication.conf to the value of userBaseDN.

Also set the following attributes to the same value as userNameAttribute:

- groupMappingAttribute
- groupMemberAttribute
- groupNameAttribute

For example:

[supportLDAP]
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = #######
groupBaseDN = ou=People,dc=splunksupport,dc=com
groupBaseFilter = (objectclass=*)
groupMappingAttribute = MyUserID
groupMemberAttribute = MyUserID
groupNameAttribute = MyUserID
host = supportldap.splunksupport.com
port = 389
realNameAttribute = cn
userBaseDN = ou=People,dc=splunksupport,dc=com
userBaseFilter = (objectclass=*)
userNameAttribute = MyUserID

[roleMap_supportLDAP]
Test your LDAP configuration

If you find that Splunk Enterprise is not able to connect to your LDAP server, try these troubleshooting steps:

1. Check `$SPLUNK_HOME/var/log/splunk/splunkd.log` for any authentication errors. Turn on DEBUG-level logging for `AuthenticationManagerLDAP` to get more information here. This can be done from the Splunk Web UI - Server Settings/Server Logging.

2. Remove any custom values you've added for `userBaseFilter` and `groupByFilter`.

3. In the `*nix command line, you can use `ldapsearch` to confirm that the variables you are specifying will return the expected entries:

   ```
   ldapsearch -x ?h <ldap_host> ?p <ldap_port> ?D "bind_dn" -w "bind_passwd" -b "user_basedn" "userNameAttribute=*"
   ldapsearch -x ?h <ldap_host> ?p <ldap_port> ?D "bind_dn" -w "bind_passwd" ?b "group_basedn" "groupNameAttribute="
   ```

   If these commands return matching entries, then your backend LDAP system is properly configured. Continue to troubleshoot the Splunk LDAP strategy configuration.

Convert to LDAP from Splunk authentication

If you move from Splunk authentication to LDAP, it's important to note that Splunk accounts are not automatically disabled and take precedence over LDAP accounts.

If you have converted from Splunk authentication system to LDAP, you might need to delete Splunk users to ensure that you're using LDAP credentials. This is only necessary if usernames are the same in both systems.

Secure local Splunk accounts

If you have configured Splunk Enterprise to use LDAP authentication, it's important to be aware that all local accounts using Splunk authentication are still present and active, including the "admin" account. You need to consider the security implications of this.

To remove all the current local accounts when enabling LDAP authentication:

- Move the `$SPLUNK_HOME/etc/passwd` file to `passwd.bak`.
- Create a blank `$SPLUNK_HOME/etc/passwd` file.
- Restart Splunk Enterprise.

Keep in mind that local Splunk accounts can still be created when Splunk Enterprise is in LDAP authentication mode. Also, any local Splunk accounts that must remain for backup or disaster-recovery purposes should use a very strong password.

When using LDAP, make sure that your LDAP implementation enforces:

- Strong password requirements for length and complexity.
- A low incorrect attempt threshold for password lockout.
Saved searches

If your LDAP usernames are the same as the names you previously used in the built-in system (but then deleted), saved searches should work without any conversion.

If you have existing saved searches created when your system was using Splunk authentication and you’d like to transfer them to an LDAP user of a different name, edit the metadata:

1. Modify `$SPLUNK_HOME/etc/apps/<app_name>/metadata/local.meta` and swap the `owner = <username>` field under each `savedsearch` permission stanza to the corresponding LDAP username and save your changes.

2. Restart Splunk Enterprise for your changes to take effect.

Best practice for removing an LDAP user

If you remove a user from your LDAP directory, Splunk Enterprise does not automatically remove the corresponding Splunk user. Usually this is not an issue, but if the user has global permissions of any sort, LDAP may generate errors.

To more information about working with LDAP users in Splunk Enterprise, see "Set up user authentication with LDAP" in this manual.

Take the following steps to safely remove a Splunk user:

1. First, back up the `$HOME/splunk/etc/users/$userid` folder.

2. Search the files under `$HOME/splunk/etc/apps/` for the user id string to see if the user owns any searches or objects with global permissions.

3. For any searches or objects that the user owns, change the owner. You change it an admin user or maintenance account, or whatever you prefer.

4. Check `splunkd.log` on the search head to make sure there are no further LDAP authentication errors.

5. Once you have redirected any object ownership, you can safely remove the `$HOME/splunk/etc/users/$userid` folder.
Multi-factor authentication

About multifactor authentication with Duo Security

Multifactor authentication allows you to configure a primary and secondary login for your Splunk Enterprise users.

**Note:** Duo Security multifactor authentication secures Splunk Web logins. We recommend that you also secure your users with a firewall.

Splunk Enterprise with Duo Security multifactor authentication requires the user to set up a second authentication method and then use that method for future logins:

1. User logs into their Enterprise Splunk Web homepage using their login credentials. This is the primary login.

2. User then sees a second login page: "Duo Authentication". This is the secondary login.

3. The first time a user logs in, they follow the instructions on the Duo login page to set up their preferred method for accessing their secondary credentials:
   - Login with credentials sent via a push notification on your your smart phone (Duo Security Mobile app required).
   - Login with credentials sent via SMS message sent to your cell phone.
   - Login with credentials sent via a phone call made to your cell phone.
   - Login by entering a one time code generated by the Duo Mobile app.

4. After the initial login and configuration, every time the user arrives at the secondary login, they receive those login credentials using their preferred method.

About setting up Duo Security for multifactor authentication in Splunk

1. Create an account for your Splunk Enterprise configuration on the Duo website. See [https://duo.com](https://duo.com) for more information.

2. Provide Splunk Enterprise with the information from your Duo Security Account. See Configure Splunk to use Duo Security multifactor authentication for more information.

Configure Splunk Enterprise to use Duo Security multifactor authentication

**NOTE:** If you have previously configured Splunk Enterprise to use Duo authentication via [https://duo.com/docs/splunk](https://duo.com/docs/splunk), you must use the task described in this topic to reconfigure multifactor login with Duo Security.

Overview

- Use the Duo Security website to create a Duo Security account for Splunk Enterprise. See [https://duo.com](https://duo.com) for more information.
- Configure Splunk Enterprise to use Duo by providing the following information:
  - Your integration key (i.e. DIXXXXXXXXXXXXXXXXX)
  - Your secret key
  - Your API hostname (i.e. api-XXXXXXXX.duosecurity.com)
When the user logs into Splunk Enterprise and follows the instructions on the Duo login page, they are given secondary login credentials.

**Configure**

1. In the **Menu**, select **Settings > Users and Authentication > Access roles**.

2. Click **Authentication Method**.

3. Under **Multifactor Authentication**, select **Duo Security**.

4. Click the **Configure Duo Security** link.

5. Provide the **Integration Key** from your Duo configuration. You can find this key on your Duo Security configuration page or at **Configuration > Details**.

6. Provide the **Secret Key** from your Duo Security configuration or detail. You can find this key on your Duo Security configuration page or at **Configuration > Details**.

7. Provide the **API Hostname** from your Duo configuration. You can find this key on your Duo Security configuration page or at **Configuration > Details**.

8. Tell Splunk Enterprise how to authenticate users when Duo Security is unavailable:

   - **Let users login** Users who have successfully logged into the Splunk Web (i.e., primary authentication) can access Splunk Enterprise even if Duo authentication (i.e., secondary authentication) fails.
   - **Do not let users login** Users who have successfully logged into the Splunk Web (i.e., primary authentication) cannot access Splunk Enterprise if Duo authentication (i.e., secondary authentication) fails.

9. Provide a time limit, in seconds, for how long authentication is attempted before the connection times out.

10. Save your changes. You do not need to reload authentication for multifactor authentication to take effect.

Once a user logs in, the Duo login page appears, the user is instructed to choose a method to access their secondary login credentials.

**How multifactor authentication works with other forms of authentication**

Note that you cannot use any form of multi-factor authentication with SSO or SAML authentication. Multi-factor authentication works with the following sources of authentication:

- Native authentication
- LDAP
- Scripted authentication

**Configure Duo multifactor authentication for Splunk Enterprise in the configuration file**

In *authentication.conf*, edit the **[2FA stanza name]** stanza as follows:
[authentication]
externalTwoFactorAuthVendor = <Duo>
externalTwoFactorAuthSettings = <2FA stanza name>

[<2FA stanza name>]
integrationKey = <Integration Key as provided by Duo>
secretKey = <Secret Key as provided by Duo>
applicationKey = <Manually generated secret key>
apiHostname = <API Hostname as provided by Duo>
failOpen = True|False (Default : False)
timeout = <in seconds>
Authentication using single sign-on with SAML

Configure single sign-on with SAML

You can configure Splunk software to use SAML authentication for single sign-on (SSO), using information provided by your supported identity provider (IdP).

Splunk software always outputs usernames in lowercase. If your IdP expects Splunk software to preserve uppercase letters in usernames, you can change the username to lowercase in the IdP or configure the IdP to accept the lowercase version of a username.

Note that if the search head is restarted, you must re-enter your credentials to access saved searches.

Prerequisites

- Either:
  - A running version of Splunk software OR
  - A managed deployment of Splunk Cloud. Self-service deployments of Splunk Cloud log in through the Splunk customer portal and cannot independently configure SAML SSO.
- An identity provider configured to provide the role, realName, and mail attributes. The supported identity providers are:
  - Ping Identity
  - Okta
  - Azure AD
  - AD FS
  - OneLogin
  - Optimal
  - CA siteminder
- An admin role (Splunk Enterprise) or sc_admin role (Splunk Cloud) with the change_authentication capability. This permission level lets you enable SAML and edit authentication settings on the Splunk search head.

Other IdPs

Any SAML IdP that is v2 compliant should be configurable, including the following tested IdPs. For assistance with any IdP that is not documented in this chapter, contact Support:

- SecureAuth
- Novell Directory
- G Suite (Formerly Google Apps for Business)

Configure SAML to work with your IdP

SAML does not support encryption, regardless of IdP.

1. Configure SAML SSO with:

- Ping Identity
- Okta
- Azure AD or AD FS
2. Map SAML groups to Splunk Enterprise roles.

**Configure SSO with PingIdentity as your identity provider**

This task describes how to set up SSO for Splunk Enterprise if you have configured PingIdentity as your Identity Provider (IdP). For information about configuring PingIdentity as an IdP, consult your Ping Federate documentation.

Use the following task to configure your Splunk deployment to recognize and use your PingIdentity configuration. Then, map groups of PingIdentity users to Splunk user roles so that those users can log in.

**Prerequisites**

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

As a best practice, the session timeout in splunk and on the IdP should be the same so that the sessions are invalidated simultaneously on Splunk and the IdP. For more information, see Configure session timeouts with Ping Identity.

As a best practice, the session timeout in splunk and on the IdP should be the same so that the sessions are invalidated simultaneously on Splunk and the IdP.

1. In the **Settings** menu, select **Access Controls > Authentication method**.

2. Select **SAML** as your authentication type.

3. Click **Configure Splunk to use SAML**.

4. On the SAML Groups page, click **SAML Configuration**.

5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP's documentation if you are not sure how to get your metadata file.

6. In **General Settings**, provide the following information.

<table>
<thead>
<tr>
<th>Single Sign on URL</th>
<th>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk sends authentication requests. Your users use this URL for SSO login. To access the login page once SAML is enabled, append the full login URL (/account/login) with loginType=Splunk. Users can also log into their local Splunk account by navigating directly to splunkweb:port/en-US/account/login?loginType=Splunk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Log Out URL</td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td>IdP certificate path</td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk software uses that file to validate the SAML response. If you provide a directory, Splunk software looks for all the certificates in that directory and tries to validate the SAML response with each one of them. If any validation fails, the response</td>
</tr>
</tbody>
</table>
is considered invalid.

| IdP certificate chains | If you use a certificate chain, order them as follows:
| | 1. Root
| | 2. Intermediate
| | 3. Leaf
| Replicate certificates | Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head.
| Issuer Id | This is the Entity Id of the IdP. See your IdP documentation if you are not sure where to find this information.
| Entity ID | This field is the entity ID as configured in the SP connection entry in your IdP.
| Sign AuthRequest | Select this option.
| Sign SAML Response | Select this option.

7. In **Attribute Query Requests**, optionally provide the following information so you can create scheduled searches later.

| Attribute Query URL | (Optional) This is the endpoint on the IdP to which queries over SOAP are sent. The format is as follows: <urn:oasis:names:tc:SAML:2.0:attrname-format:uri>
| Sign attribute query request | Verify that this field is selected.
| Sign attribute query response | Verify that this field is selected.
| Username | Enter a user name.
| Provide a password |

8. In the **Alias** section optionally provide the following aliasing information:

| Role Alias | Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes.
| Real Name Alias | You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name.
| Mail Alias | Skip this field.

9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure advanced settings for SSO.

10. Click **Save**.

**Next Step**

Map SAML groups to Splunk Enterprise roles
Configure SSO with Okta as your identity provider

This task describes how to set up SSO for Splunk Enterprise if you have configured Okta as your Identity Provider (IdP). For information about configuring Okta as an IdP, consult your Okta documentation.

Use the following task to configure your Splunk deployment to recognize and use your Okta configuration. Then, map groups from the IdP to Splunk user roles so that those groups can log in.

**Prerequisites**

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

1. In the **Settings** menu, select **Access Controls > Authentication method**.
2. Select **SAML** as your authentication type.
3. Click **Configure Splunk to use SAML**.
4. On the SAML Groups page, click **SAML Configuration**.
5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your Okta documentation if you are not sure how to locate your metadata file.
6. In **General Settings**, provide the following information:

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Sign on URL</strong></td>
<td>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk Enterprise sends authentication requests. To access the login page once SAML is enabled, append the full login URL (/saml/acs) with <code>loginType=Splunk</code>. Users can also log into their local Splunk account by navigating directly to <code>splunkweb:port/en-US/account/login?loginType=Splunk</code>.</td>
</tr>
<tr>
<td><strong>Single Log Out URL</strong></td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td><strong>IdP's certificate path</strong></td>
<td>This value can be a directory or a single file, depending on your IdP requirements. If you provide a file, Splunk Enterprise uses that file to validate authenticity of SAML response. If you provide a directory, Splunk Enterprise looks at all the certificates in the directory and tries to validate SAML response with each one of them. If any validation fails, authentication fails.</td>
</tr>
</tbody>
</table>
| **IdP certificate chains** | If you use a certificate chain, order them as follows:  

1. Root  
2. Intermediate  
3. Leaf |
Replicate certificates

Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head.

Issuer Id

This is the Entity Id of the IdP. See your IdP documentation if you are not sure where to find this information.

Entity ID.

This field is the entity ID as configured in the SP connection entry in your IdP.

Sign AuthRequest.

Select this option.

Sign SAML Response.

Select this option.

If "Request Compression" is set, when you log onto Splunk Web on a Search Head, you are diverted to Okta Applications rather than the Search Head.

7. Skip Attribute Query and go to steps 8 and 9.

8. In the Alias section optionally provide the following aliasing information:

In Alias, provide the following information:

<table>
<thead>
<tr>
<th>Role Alias</th>
<th>Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Name Alias</td>
<td>You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name.</td>
</tr>
<tr>
<td>Mail Alias</td>
<td>Skip this field.</td>
</tr>
</tbody>
</table>

9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure load balancing or SAML bindings

10. Click Save.

Next Step

Note: An error in configuring SAML could result in users and admins being locked out of Splunk Cloud. Use this link to access the local login and revert to None for authentication if you are locked out:

https://<name>.splunkcloud.com/en-US/account/login?loginType=splunk [replace <name> with your account name]

Map SAML groups to Splunk Enterprise roles

Configure SSO with AzureAD or AD FS as your Identity Provider

This task describes how to set up SSO for Splunk deployments if you have configured AzureAD or ADFS as your Identity Provider (IdP).

Prerequisites

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

SAML does not support encryption, regardless of IdP.
When configuring your IdP, note the following suggestions when configuring your groups:

- Before you begin, be sure to consult your IdP documentation and ensure that you have met the IdP configuration requirements.
- For AzureAD the reply URL may require /SAML/acs on the end.
- For AzureAD you may need to change the groupMembershipClaims from null to SecurityGroup.
- For AD FS for Splunk Cloud, you may need to set the Claim Type as "UPN" when configuring your IdP. The Splunk blog post at https://www.splunk.com/blog/2016/09/14/configuring-microsofts-adfs-splunk-cloud.html provides more information about configuring AD FS for Cloud.

Use the following task to configure your Splunk deployment to recognize and use your AzureAD or AD FS configuration. Then, map groups of AzureAD or AD FS users to Splunk user roles so that those users can log in.

**Configure Splunk Software for SAML**

1. In the **Settings** menu, select **Access Controls > Authentication method**.
2. Select **SAML** as your authentication type.
3. Click **Configure Splunk to use SAML**.
4. On the SAML Groups page, click **SAML Configuration**.
5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP's documentation if you are not sure how to get your metadata file.
6. In **General Settings**, provide the following information.

<table>
<thead>
<tr>
<th><strong>Single Sign on URL</strong></th>
<th>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk Enterprise sends authentication requests. Your users also use this URL for SSO login. To access the login page once SAML is enabled, append the full login URL (/account/login) with loginType=Splunk. Users can also log into their local Splunk account by navigating directly to splunkweb:port/en-US/account/login?loginType=Splunk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Log Out URL</strong></td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td><strong>IdP certificate path</strong></td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk software uses that file to validate authenticity of SAML response. If you provide a directory, Splunk looks at all the certificates in the directory and tries to validate SAML response with each one of them. If validation fails, authentication fails.</td>
</tr>
</tbody>
</table>
| **IdP certificate chains** | If you use a certificate chain, order them as follows:  
1. Root  
2. Intermediate  
3. Leaf |
| **Replicate certificates** | Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head. |
7. Skip the **Attribute Query** section and go to steps 8 and 9.

8. In the **Alias** section optionally provide the following aliasing information:

| Role Alias | Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes. |
| Real Name Alias | You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name. |
| Mail Alias | Skip this field. |

9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure load balancing or SAML bindings

10. Click **Save**.

**Next Step**

**Note:** An error in configuring SAML could result in users and admins being locked out of Splunk Cloud. Use this link to access the local login and revert to None for authentication if you are locked out:

https://<name>.splunkcloud.com/en-US/account/login?loginType=splunk [replace <name> with your account name]

**Note 2:** You may experience an issue with Splunk continuously refreshing/re-authenticating if the Splunk Web session timeout is different than your IDP vendor session timeout (i.e. Splunk web session timeout: 1 hour and IDP vendor session timeout: 9 hours). This issue may cause users to lose work. To correct the issue you should set the Splunk web session timeout to be equal to the IDP vendor session timeout.

Map SAML groups to Splunk Enterprise roles

### Configure SSO with OneLogin as your identity provider

This task describes how to set up SSO for Splunk if you have configured OneLogin as your Identity Provider (IdP). For information about configuring OneLogin as an IdP, consult your OneLogin documentation.

Use the following task to configure Splunk to recognize and use your OneLogin configuration. Then, map groups of OneLogin users to Splunk user roles so that those users can log into Splunk.

**Prerequisites**

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.
1. In the **Settings** menu, select **Access Controls > Authentication method**.

2. Select **SAML** as your authentication type.

3. Click **Configure Splunk to use SAML**.

4. On the SAML Groups page, click **SAML Configuration**.

5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP’s documentation if you are not sure how to get your metadata file.

6. In **General Settings**, provide the following information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign on URL</td>
<td>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk sends authentication requests.</td>
</tr>
<tr>
<td></td>
<td>To access the login page once SAML is enabled, append the full login URL (/account/login) with ?loginType=Splunk. Users can also log into their local Splunk account by navigating directly to ?splunkweb:port/en-US/account/login?loginType=Splunk</td>
</tr>
<tr>
<td>Single Log Out URL</td>
<td>OneLogin supports redirect binding for single log out. Set the binding to 'HTTPRedirect'.</td>
</tr>
<tr>
<td>IdP's certificate path</td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk uses that file to validate authenticity of SAML response. If you provide a directory, Splunk looks for all the certificates that are present as children of the directory and tries to validate SAML response with each one of them, if Splunk fails to validate authenticity with all of them, response is not considered authentic.</td>
</tr>
<tr>
<td>IdP certificate chains</td>
<td>If you use a certificate chain, order them as follows:</td>
</tr>
<tr>
<td></td>
<td>1. Root</td>
</tr>
<tr>
<td></td>
<td>2. Intermediate</td>
</tr>
<tr>
<td></td>
<td>3. Leaf</td>
</tr>
<tr>
<td>Replicate certificates</td>
<td>Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head.</td>
</tr>
<tr>
<td>Issuer Id</td>
<td>This is the Entity Id of the IdP. See your IdP documentation if you are not sure where to find this information.</td>
</tr>
<tr>
<td>Entity ID.</td>
<td>This field is the entity ID as configured in the SP connection entry in your IdP.</td>
</tr>
<tr>
<td>Sign AuthRequest.</td>
<td>Set this value to false.</td>
</tr>
<tr>
<td>Sign SAML Response.</td>
<td>Set this value to false.</td>
</tr>
</tbody>
</table>

7. Skip the **Attribute Query** section and go to steps 8 and 9.

8. In the **Alias** section optionally provide the following aliasing information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Alias</td>
<td>Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes.</td>
</tr>
<tr>
<td></td>
<td>You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name.</td>
</tr>
</tbody>
</table>

57
9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure load balancing or SAML bindings

10. Click Save.

Note: An error in configuring SAML could result in users and admins being locked out of Splunk Cloud. Use this link to access the local login and revert to None for authentication if you are locked out:

https://<name>.splunkcloud.com/en-US/account/login?loginType=splunk [replace <name> with your account name]

Next Step

Map SAML groups to Splunk Enterprise roles

Configure SSO with Optimal as your identity provider

This task describes how to set up SSO for Splunk if you have configured Optimal as your Identity Provider (IdP). For information about configuring Optimal as an IdP, consult your Optimal documentation.

Use the following task to configure Splunk to recognize and use your Optimal configuration. Then, map groups of Optimal users to Splunk user roles so that those users can log into Splunk.

Prerequisites

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

1. In the Settings menu, select Access Controls > Authentication method.

2. Select SAML as your authentication type.

3. Click Configure Splunk to use SAML.

4. On the SAML Groups page, click SAML Configuration.

5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP’s documentation if you are not sure how to get your metadata file

6. In General Settings, provide the following information.

<table>
<thead>
<tr>
<th>Single Sign on URL.</th>
<th>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk sends authentication requests. If you are using Splunk Cloud, open a support ticket to have the Splunk Cloud operations team open the port for communicating with the IdP. Your users use this URL for SSO login. To access the login page once SAML is enabled, append the full login URL (/account/login) with loginType=Splunk. Users can also log into their local Splunk account by navigating directly</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Log Out URL</strong></td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td><strong>IdP's certificate path</strong></td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk uses that file to validate authenticity of SAML response. If you provide a directory, Splunk looks for all the certificates that are present as children of the directory and tries to validate SAML response with each one of them, if Splunk fails to validate authenticity with all of them, response is not considered authentic.</td>
</tr>
</tbody>
</table>
| **IdP certificate chains** | If you use a certificate chain, order them as follows:  
1. Root  
2. Intermediate  
3. Leaf |
| **Replicate certificates** | Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head. |
| **Issuer Id** | This is the Entity Id of the IdP. See your IdP documentation if you are not sure where to find this information. |
| **Entity ID** | This field is the entity ID as configured in the SP connection entry in your IdP. |
| **Sign AuthRequest.** | Select this option. |
| **Sign SAML Response.** | Select this option. |

7. Skip the **Attribute Query** section and go to steps 8 and 9.

8. In the **Alias** section optionally provide the following aliasing information:

| **Role Alias** | Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes. |
| **Real Name Alias** | You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name. |
| **Mail Alias** | Skip this field. |

9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure load balancing or SAML bindings

10. Click **Save**.

Note: An error in configuring SAML could result in users and admins being locked out of Splunk Cloud. Use this link to access the local login and revert to None for authentication if you are locked out:

https://<name>.splunkcloud.com/en-US/account/login?loginType=splunk [replace <name> with your account name]

**Next Step**

Map SAML groups to Splunk Enterprise roles
Configure SSO in CA Siteminder

This task describes how to set up SSO for Splunk if you have configured CA as your Identity Provider (IdP). For information about configuring CA as an IdP, consult your CA documentation.

Use the following task to configure Splunk to recognize and use your CA configuration. Then, map groups of CA users to Splunk user roles so that those users can log into Splunk.

Prerequisites

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

1. In the Settings menu, select Access Controls > Authentication method.
2. Select SAML as your authentication type.
3. Click Configure Splunk to use SAML.
4. On the SAML Groups page, click SAML Configuration.
5. Download or browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP's documentation if you are not sure how to get your metadata file.
6. In General Settings, provide the following information.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign on URL</td>
<td>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk sends authentication requests. To access the login page once SAML is enabled, append the full login URL (/account/login) with loginType=Splunk. Users can also log into their local Splunk account by navigating directly to ?splunkweb:port/en-US/account/login?loginType=Splunk</td>
</tr>
<tr>
<td>Single Log Out URL</td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td>IdP's certificate path</td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk uses that file to validate authenticity of SAML response. If you provide a directory, Splunk looks for all the certificates that are present as children of the directory and tries to validate SAML response with each one of them, if Splunk fails to validate authenticity with all of them, response is not considered authentic.</td>
</tr>
<tr>
<td>IdP certificate chains</td>
<td>If you use a certificate chain, order them as follows: 1. Root 2. Intermediate 3. Leaf</td>
</tr>
<tr>
<td>Replicate certificates</td>
<td>Check this to replicate your IdP certificates in a search head cluster. When configuring SAML on a search head cluster, you must use the same certificate for each search head.</td>
</tr>
<tr>
<td>Issuer Id</td>
<td>This is the Entity Id of the IdP. See your IdP documentation if you are not sure where to find this information.</td>
</tr>
<tr>
<td>Entity ID</td>
<td>This field is the entity ID as configured in the SP connection entry in your IdP.</td>
</tr>
</tbody>
</table>
7. Skip the **Attribute Query** and go to steps 8 and 9.

8. In the **Alias** section optionally provide the following aliasing information:

| Role Alias | Use this field to specify a new attribute name on any IdP and then configure an alias in your Splunk deployment for any of the three attributes. |
| Real Name Alias | You may skip this field. For ADFS you can use the displayname for the Attribute Alias Real Name. |
| Mail Alias | Skip this field. |

9. Populate the advanced section only if you need to set up load balancing or change the SAML binding. See Configure load balancing or SAML bindings

10. Click **Save**.

**Note:** An error in configuring SAML could result in users and admins being locked out of Splunk Cloud. Use this link to access the local login and revert to None for authentication if you are locked out:

https://<name>.splunkcloud.com/en-US/account/login?loginType=splunk [replace <name> with your account name]

**Next Step**

Map SAML groups to Splunk Enterprise roles

**Secure SSO with TLS certificates**

Configure the following SSL settings to enable Splunk Enterprise to perform TLS verification between Splunk Instance and the SOAP instance providing **AttributeQuery** service.

Unless noted, values not set default to the setting specified in **server.conf**.

```plaintext
[<saml-authSettings-key>]

sslVersions = <Comma-separated list of SSL versions to support>

sslCommonNameToCheck = <commonName> When populated, and sslVerifyServerCert is "true", splunkd limits most outbound HTTPS connections to hosts which use a cert with this common name.

sslAltNameToCheck = <alternateName1>, <alternateName2>, ...If set, and sslVerifyServerCert' is "true", splunkd can verify certificates with "Subject Alternate Name" that matches any of the is alternate names in this list.

ecdhCurveName = <ECDH curve to use for ECDH key negotiation>
```
serverCert = <Server certificate file> Default certificates, "sever.pem" are auto-generated by splunkd upon starting Splunk, you may replace the default cert with your own PEM format file.

sslPassword = <Server certificate password>

cacertFile = <Public key of the signing authority> The default value is cacert.pem

capath = <Path where all these certs are stored>. Default value is $SPLUNK_HOME/etc/auth

sslVerifyServerCert = [ true | false ] If true, distributed search makes a search request to another server in the search cluster.

blacklistedAutoMappedRoles = <comma separated list of roles> Optionally provide a comma-separated list of Splunk roles that you do not want Splunk to auto-map if received in the IDP Response.

blacklistedUsers = <comma separated list of user names> Optionally provide a comma-separated list of user names that Splunk must reject from the IDP response.

nameIdFormat = <string> Optionally, and If supported by IDP, specify the format of the Subject returned in the SAML Assertion.

ssoBinding = <HTTPPost | HTTPRedirect> Optionally specify the binding to use when making a SP-initiated SAML request. The binding must match the one configured on the IDP.

sloBinding = <HTTPPost | HTTPRedirect> > Optionally specify the binding to use when making a logout request or sending a logout response to complete the logout workflow. The binding must match the one configured on the IDP.

signatureAlgorithm = <RSA-SHA1 | RSA-SHA256> Optionally specify the signature algorithm to use for a SP-initiated SAML request. 'signedAuthnRequest' must be set to true for this setting to take effect. The algorithm applies to both the http post and redirect binding.

replicateCertificates = <boolean> Optionally specify the IdP certificate files to replicate across search head cluster setup. Search head clustering must also be enabled. If certificate replication is not enabled, IdP certificate files must be replicated manually across SHC or verification of SAML signed assertions fails.

Configuring SAML in a search head cluster

You can configure SAML on a search head that does or does not use a load balancer. For authentication requests to be signed (recommended), you must use the same signing certificate on all search head members in the cluster.

Every search head in the cluster must have the public key of the IdP. Splunk uses this key to verify the signature of the SAML authentication response. When you use SplunkWeb to configure SAML, the public key from metadata is automatically set to replicate to Search.

1. Generate a public/private key pair.

2. Concatenate the generated key pair into one pem file. This file is used for signing authentication requests going out from Splunk. Concatenate in the following order:

   - Public key is self signed:
3. Replicate the new certificate file to the location relative to $SPLUNK_HOME on each search head. Make sure to give the certificate the same name on all search heads. For example:

```
$SPLUNK_HOME/etc/auth/samlRequestSigningCerts/samlRequestSigningCert.pem
```

4. Edit the Splunk metadata: In the `<X509Certificate>` file, swap the public key in the metadata with the public key from the new certificate. Then remove the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` tags.

5. Configure your IdP using the Splunk metadata. See the instructions for your IdP.

6. Collect your IdP metadata and use it to configure Splunk. Previous steps created a SAML-related configuration in `$SPLUNK_HOME/etc/system/local/authentication.conf`.

**Note:** To enable seamless Single Logout, we recommend that you configure search head members to all have same `entityID`.

7. Add the path to the `ClientCert` parameter in authentication configuration:

```
clientCert = $SPLUNK_HOME/etc/auth/samlRequestSigningCerts/samlReqCert.pem
```

8. If the private key you created in step 1. is encrypted and you set up a password for the private key `sslPassword = <password for private key>` then you must repeat steps ABC for all search head members.

9. Reload authentication on all search heads to implement your changes.

10. To validate your configuration, log in to each search head individually to ensure all search heads are using the same key for signing authentication requests and that the IdP is configured with the right cert for verifying signature of the request.

### Configure Ping Identity with leaf or intermediate SSL certificate chains

To configure Ping Identity with leaf or intermediate certificates:

1. Verify or create the following directories in Splunk. You can use the following command:

```
/home/build/build-home/galaxy/openssl/ ***(or which every directory /splunk cmd openssl version ?d command returns /opt/splunk/etc/auth/idpCerts/ /opt/splunk/etc/auth/idpCerts/certs/ /)
```

2. Create a link between the two d/opt/splunk/etc/auth/idpCerts/ certificates and

```
/home/build/build-home/galaxy/openssl/(or the directory that is returned by splunk cmd openssl version ?d). You can use the following command:
```
3. Place your certificate chain in `/opt/splunk/etc/auth/idpCerts/` and ensure that they are owned by splunk:splunk:. You can use the following commands:

```bash
ln -s /opt/splunk/etc/auth/idpCerts/ /home/build/build-home/galaxy.openssl/
```

4. Once your root, intermediate, and leaf certificate files are in place, create x.509 hash links from the certificates directory to the certificates in the `idpCerts` directory. You can use the following command:

```bash
ln -s /opt/splunk/etc/auth/idpCerts/aaa_intermediate.pem `openssl x509 -hash -noout -in /opt/splunk/etc/auth/idpCerts/aaa_intermediate.pem`.0
ln -s /opt/splunk/etc/auth/idpCerts/aaa_root.pem `openssl x509 -hash -noout -in /opt/splunk/etc/auth/idpCerts/aaa_root.pem`.0
```

5. In the `authentication.conf` file, point the `idpCertPath` attribute to the intermediate filename (make sure you are pointing to the certificate file, not the symlink). You can use the following entry:

```bash
idpCertPath = /opt/splunk/etc/auth/idpCerts/aaa_intermediate.pem
```


6. Select the following fields:
   - Sign AuthnRequest = checked
   - Sign SAML response = checked

7. Save your changes in Splunk Web.

8. In your Ping Identity configuration, set "Include Certificate in KeyInfo" to "True".

9. Save your changes.

**Configure SAML SSO for other IdPs**

Any identity provider that is compliant with version 2.0 of the Security Assertion Markup Language (SAML) should be configurable with SAML on the Splunk platform. For information about supported and tested IdPs, see How SAML SSO works.

The following topic provides general instructions for configuring an IdP to work with Splunk Enterprise. There might be exceptions or differences in the process of configuring an IdP to work with Splunk Enterprise, depending upon the IdP you use. For assistance with configuration procedures that are not documented in this chapter, contact Splunk Support.

When you configure the Splunk platform to use your SAML authentication system, you can authorize groups on your IdP to log in by mapping them to Splunk user roles.

SAML does not support encryption.
Prerequisites

Verify that your system meets all of the requirements. See Configure single sign-on with SAML.

1. In the Settings menu, select Access Controls > Authentication method.
2. Select SAML as your authentication type.
3. Click Configure Splunk to use SAML.
4. On the SAML Groups page, click SAML Configuration.
5. Browse and select your metadata file, or copy and paste your metadata directly into the text window. Refer to your IdP's documentation if you are not sure how to get your metadata file.
6. In General Settings, provide the following information.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign on URL.</td>
<td>This field is populated automatically by your selected metadata file. It is the protected endpoint on your IdP to which Splunk sends authentication requests. Your users use this URL for SSO login.</td>
</tr>
<tr>
<td></td>
<td>To access the login page once SAML is enabled, append the full login URL (/account/login) with loginType=Splunk. Users can also log into their local Splunk account by navigating directly to splunkweb:port/en-US/account/login?loginType=Splunk</td>
</tr>
<tr>
<td>Single Log Out URL.</td>
<td>This field is populated automatically by the metadata file and is the IdP protocol endpoint. If you do not provide this URL, the user will not be logged out.</td>
</tr>
<tr>
<td>IdP's certificate path</td>
<td>This value can be a directory or a file, depending on your IdP requirements. If you provide a file, Splunk uses that file to validate authenticity of SAML response. If you provide a directory, Splunk looks for all the certificates that are present as children of the directory and tries to validate SAML response with each one of them, if Splunk fails to validate authenticity with all of them, response is not considered authentic.</td>
</tr>
<tr>
<td></td>
<td>When configuring SAML on a search head cluster, make sure you configure the same certificate for each search head.</td>
</tr>
<tr>
<td>Entity ID.</td>
<td>This field is the entity ID as configured in the SP connection entry in your IdP.</td>
</tr>
<tr>
<td>Sign AuthRequest.</td>
<td>Select this option.</td>
</tr>
<tr>
<td>Sign SAML Response.</td>
<td>Select this option.</td>
</tr>
</tbody>
</table>

7. If you use PingIdentity as your IdP, in Attribute Query, provide the following information so you can create scheduled searches later. These fields are not required for creating scheduled searches with Okta, Azure AD, or AD FS.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Query URL.</td>
<td>This field is the endpoint on the IdP to which queries over SOAP are sent. The format is as follows: <a href="">urn:oasis:names:tc:SAML:2.0:attrname-format:uri</a></td>
</tr>
<tr>
<td>Sign attribute query request</td>
<td>Verify that this field is selected.</td>
</tr>
<tr>
<td>Sign attribute query response</td>
<td>Verify that this field is selected.</td>
</tr>
</tbody>
</table>

8. In Advanced settings, provide the following information.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Alias Role</td>
<td>Use this field to specify a new attribute name on any IdP and then configure an alias on the Splunk side for any of the 3 attributes.</td>
</tr>
<tr>
<td></td>
<td>If you have configured the AD FS built-in &quot;Role&quot; attribute to be returned and this has the AD group information, specify <a href="http://schemas.microsoft.com/ws/2008/06/identity/claims/role">http://schemas.microsoft.com/ws/2008/06/identity/claims/role</a>. This value tells Splunk the attribute that contains the role information in the SAML response returned.</td>
</tr>
</tbody>
</table>
If you have configured Azure AD, specify

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Alias</th>
<th>Real Name</th>
<th>Populate this field if you use Azure AD as your IdP. This value tells Splunk Enterprise where to map the real name in the SAML response returned. Enter <a href="http://schemas.microsoft.com/identity/claims/displayname">http://schemas.microsoft.com/identity/claims/displayname</a>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Alias</td>
<td>Mail</td>
<td>Populate this field if you use Azure AD as your IdP. This value maps the alias to the user email addresses in the SAML response returned. Enter <a href="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress">http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress</a>.</td>
</tr>
<tr>
<td>FQDN - Host Name or IP of the load balancer</td>
<td></td>
<td></td>
<td>Set to: <a href="https://sh1.STACKID.splunkcloud.com">https://sh1.STACKID.splunkcloud.com</a>. This setting works for a Splunk deployment with Single Search Head Setup or a Search Head Cluster Setup. You must provide an address if you use load balancing with a search head cluster.</td>
</tr>
<tr>
<td>(Optional) Redirect Port</td>
<td></td>
<td></td>
<td>Provide a redirect port for the load balancer described in the previous field. For Okta set it to &quot;0&quot; (Zero).</td>
</tr>
</tbody>
</table>

9. Click Save.

Next Steps

Map SAML groups to Splunk Enterprise roles

Configure advanced settings for SSO

Populate the advanced section to set up load balancing or change the SAML binding. In Advanced settings, provide the following information:

<table>
<thead>
<tr>
<th>Name Id Format</th>
<th>Provide the Id format provided by your IdP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully qualified domain name or IP of the load balancer</td>
<td>Set to: <a href="https://sh1.STACKID.example.com">https://sh1.STACKID.example.com</a>. This setting works for a Splunk deployment with Single Search Head Setup or a Search Head Cluster Setup. You must provide an address if you use load balancing with a search head cluster.</td>
</tr>
<tr>
<td>(Optional) Redirect port - load balancer port</td>
<td>Provide a redirect port for the load-balancer described in the previous field.</td>
</tr>
<tr>
<td>Redirect to URL after logout</td>
<td>URL to which Splunk should direct a user after logout.</td>
</tr>
<tr>
<td>SSO Binding</td>
<td>The protocol binding to use for the login request sent to the IdP.</td>
</tr>
<tr>
<td>SLO Binding</td>
<td>The protocol binding to use for the logout request sent to the IdP.</td>
</tr>
</tbody>
</table>

Map SAML groups to roles

When you configure your Splunk deployment to use your SAML authentication system, you can authorize groups on your SAML server to log in by mapping them to Splunk user roles. You can map multiple groups to a single user role.

Prerequisites

About SAML SSO
1. In the **Settings** menu, select **Access Controls > Authentication method**.

2. Select SAML as your authentication type.

3. Click **Configure Splunk to use SAML**.

4. On the SAML Groups page, click **New Group** or click **Edit** for a group you want to modify.

5. Provide a name for the group.

6. Determine the roles that you want to assign to this group by moving the desired roles from the left column to the right column.

7. Click **Save**.

After you configure SAML SSO and map groups to role, you can distribute the login URL to your users.

### Modify or remove role mappings

When you configure your Splunk deployment to use your SAML authentication system, you can authorize groups on your SAML server to log in by mapping them to Splunk user roles. You can map multiple groups to a single user role.

This topic describes how to remove roles from existing groups or delete groups entirely. To remove and individual user from a SAML group, consult your IdP documentation.

**Prerequisites**

About SAML SSO

1. In the **Settings** menu, select **Access Controls > Authentication method**.

2. Select SAML as your authentication type.

3. Click **Configure Splunk to use SAML**.

4. To delete an entire group click **Delete** for the group you wish to remove.

5. On the SAML Groups page, click **Edit** for a group you want to modify.

6. Specify the roles that you want to remove from this group by moving the desired roles from the right column to the left column.

7. Click **Save**.

After you configure SAML SSO and map groups to role, you can distribute the login URL to your users.
Configure SAML SSO in the configuration files

This topic explains how to set up SSO for SAML v2 using configuration files:

- Configure `authentication.conf` and `web.conf` in Splunk Enterprise
- Configure your identity provider
- Secure your SAML configuration

Configure `authentication.conf`

Configure the following stanza in `authentication.conf`

```conf
[authentication]
authSettings = saml_settings
authType = SAML

[roleMap_SAML]
admin = Super Admin;
power = Power Admin;
user = <list roles> Admin;Employee;

[saml_settings]
entityId = <entityid>
idpAttributeQueryUrl = <optional path to the Attribute query>
https://your path/idp/attrsvc.ssaml2
idpCertPath = <path to the idp cert in Splunk>/home/user/splunk/saml-install/etc/auth/ping_idp.crt.
idpSSOUrl = <path to the sso url>
https://your path/idp/SSO.saml2.
idpSLOUrl = <Logout url> If not specified, this will be treated as a typical sso and the logout button is disabled.
https://your path/idp/SLO.saml2 #
redirectPort=443
attributeQueryTTL = 3600
signAuthnRequest = true
signedAssertion = true
attributeQueryRequestSigned = <Set to true if using optional idpAttributeQuerySSL>
attributeQueryResponseSigned = <Set to true if using optional idpAttributeQuerySSL>
attributeQuerySoapPassword = <your password>
```
attributeQuerySoapUsername = <your username>

To configure single sign-on with Azure AD or ADFS, add the following additional attributes:

nameIDFormat = (optional) Specify the format of the subject that is returned in the SAML response. AzureAD returns a string to identify the subject and this attribute lets you optionally specify a different format (we recommend email address). This can be useful for auditing and saved searches. To specify email address as the format, use:
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress

role = Populate this field if you use Azure AD for SSO or ADFS. This value tells Splunk Enterprise the attribute that supplies role information in the SAML response returned. For Azure AD, use:
http://schemas.microsoft.com/ws/2008/06/identity/claims/groups

mail = This value maps the alias to the user email addresses in the SAML response returned. For Azure AD, use:
http://schemas.microsoft.com/identity/claims/displayname

realName = This tells Splunk Enterprise where to map the real name in the SAML response returned. For Azure AD use:

Use blacklists to improve security

Splunk supports auto-mapped roles by default, so if Splunk roles are returned in an assertion, Splunk uses them. To turn off auto-mapping for roles, add the list of roles to the blacklistedAutoMappedRoles setting in authentication.conf.

blacklistedAutoMappedRoles = <Comma separated list of splunk roles that should be blacklisted from being auto-mapped by Splunk from the IDP Response>

To prevent blacklisted users from logging in, add the users to blacklistedUsers attribute in authentication.conf. If your Splunk instance is configured to use defaultRolesIfMissing, and role information is missing in the assertion, Splunk uses the defaultRolesIfMissing setting to complete that information.

blacklistedUsers = <comma separated list of user names from the IDP response to be blacklisted by the Splunk Platform>

Do not configure defaultRolesIfMissing with the "Admin" role. The Admin role is temporarily used to send group information in the SAML assertion until the IdP is configured.

Supported blacklist role and group formats

User and group names are not case-sensitive manner. Names are converted to lower case and stored. If the IdP expects the username in the NameId in the exact case that it was sent to Splunk, this might result in an error.

Splunk supports groups information in the following formats:

DN format

<saml:AttributeStatement>
  <saml:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
  </saml:Attribute>
</saml:AttributeStatement>
Strings format

<saml2:AttributeStatement>
<saml2:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
</saml2:Attribute>
</saml2:AttributeStatement>

Semicolon separated lists

<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Name="role">
<saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">role_splunktest;role_admin</saml:AttributeValue>
</saml:Attribute>

Splunk Roles

<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Name="role">
</saml:Attribute>

Configure web.conf and optionally add a failure redirect address

Add the following values to your settings stanza in web.conf

[settings]
appServerPorts = 7065 <make sure this attribute is enabled>
   <this is your custom user redirect for failed logins>

Configure certificate replication for search head clusters

Splunk software uses certificate replication to allow SAML single sign-on in a search head cluster environment. If a search head cluster is set up and certificate replication is turned on, then IdP certificate files are replicated across that search head cluster.

The is enabled by default and if there is not a configured search head cluster, the system ignores the attribute.
In `authentication.conf` turn on certificate replication:

```
replicateCertificates = true / false
```

**Configure your identity provider**

Now you must configure your IdP to import Splunk software metadata. To import Splunk software metadata on your IdP, make sure that the `AuthnRequest signing` and `AttributeQuery request signing` setting is compatible on Splunk software and the IdP:

1. Export the IdP certificate onto a file in your Splunk software instance.

2. Make sure that `authentication.conf` points to this certificate in the SAML configuration stanzas.

3. Import the Splunk software server certificate (`server.pem`) into the IdP for signature verification.

Note that you can export Splunk software metadata using the `/saml/spmetadata` endpoint on Splunk Web. You can also access the `SAML-sp-metadata` endpoint on `splunkd`.

**Secure your SAML configuration**

SAML attributequery service supports all of the standard SSL settings for Splunk Enterprise to perform TLS verification between Splunk Instance and SOAP instance providing AttributeQuery service.

In general, TLS encryption only works with an IdP that supports attribute queries. However, the `sslKeysFile` and `sslKeysFilePassword` attributes will work for any IdPs.

For more information about TLS encryption, see Configure SSO with TLS certificates

**Troubleshoot SAML SSO**

Here are some common issues and how to resolve them.

**Error message: SAML fails to verify assertions**

You see the following error message:

```
Failed to verify the assertion - The 'Audience' field in the saml response from the IdP does not match the configuration
```

**Mitigation**

1. The SAML errors are recorded in the `splunkd.log` on the search head. You can see the complete error message by running a search on that search head:

   - `index=_internal sourcetype=splunkd SAML error`

   You should see the following:

   09-18-2017 14:58:06.939 +0000 ERROR Saml - Failed to verify the assertion - The 'Audience' field in the saml response from the IdP does not match the configuration, Error details=Expected=https://<instance_name>.com, found=https://<wrong_instance_name>.com/
2. Modify `authentication.conf` with the entityId found in the error message in step 1.

```bash
[saml]
entityId= https://<instance_name>.com/ (found from ERROR message)
```

3. Reload `authentication.conf` from Splunk Web at `Settings > Access Controls > Authentication Method > Reload Authentication configuration`

**Error message: Leaf certificate does not match**

You receive the following message:

No leaf certificate matched one from the assertion
This error occurs when the signature certificate on Splunk does not match the certificate that the IdP uses to sign SAML messages.

**Mitigation**

If your signature verification certificate is a self-signed certificate:

Make sure that the certificate specified in the `idpCertPath` attribute in `authentication.conf` is the same as the certificate the IdP uses to sign SAML messages. You can use OpenSSL to determine the details of the certificate that Splunk uses for signature verification.

For example, the following command:

```bash
openssl x509 -in etc/auth/idpCerts/idpCert.pem -text -noout | grep 'Serial|Issuer:|Subject:'
```

Should produce information similar to this:

```
Serial Number: 1478287046063 (0x15830c635af)
Issuer: C=US, ST=CA, L=San Francisco, O=Splunk, OU=Splunk Service, CN=5165ffd1bf1a0363c8a5cd8062337fb4
Subject: C=US, ST=CA, L=San Francisco, O=Splunk, OU=Splunk Service, CN=5165ffd1bf1a0363c8a5cd8062337fb4
```

If the signature verification certificate is part of a certificate chain

Make sure that the signing certificates match and are consistently named. For example, a simple chain would have three files in the following order:

- the root CA, for example: "cert_1.pem"
- the intermediate certificate, for example: "cert_2.pem"
- the leaf certificate or the signing certificate, for example: "cert_3.pem"

In this example, make sure that the "cert_3.pem" (the leaf) is the same certificate that the IdP uses to sign responses.

If you have multiple chains, or chains with more than one intermediate CA

In most cases, the certificate chain consist of a single root certificate, a single intermediate certificate, and a single signing certificate. However, you may have multiple chains configured, or more than one intermediate CA.
If you have multiple chains configured, structure your certificate chain as follows:

$SPLUNK_HOME/etc/auth/idpCerts
idpCertChain_1 idpCertChain_2
$SPLUNK_HOME/etc/auth/idpCerts/idpCertChain_1
cert_1.pem cert_2.pem cert_3.pem
d$SPLUNK_HOME/etc/auth/idpCerts/idpCertChain_2
cert_1.pem cert_2.pem cert_3.pem

If you have more than one intermediate CA

If you have more than one intermediate CA, structure your certificate chain as follows:

$SPLUNK_HOME/etc/auth/idpCerts
idpCertChain_1
$SPLUNK_HOME/etc/auth/idpCerts/idpCertChain_1
cert_1.pem cert_2.pem cert_3.pem cert_4.pem cert_5.pem

Error message: Attribute query request error

Issue: You experience the following message

ERROR AuthenticationManagerSAML - Requesting user info from ID returned an error. Error in Attribute query request, AttributeQueryTransaction err=Cannot resolve hostname, AttributeQueryTransaction descr=Error resolving: Name or service not known, AttributeQueryTransaction statusCode=502

Mitigation

• Make sure that the cipherSuite is specified correctly in the SAML stanza. For example:
  ◆ cipherSuite = TLSv1+MEDIUM:@STRENGTH
  ◆ cipherSuite = ALL:!aNULL:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM

• Make sure all SOAP password requirements are met.

• Make sure your SSL settings for SAML are configured correctly in authentication.conf

Issue: You experience the following message:

ERROR AuthenticationManagerSAML - Attribute query request failed. Status code=urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal, Status msg=No attributes found for requested subject

Mitigation

• Make sure that the role, mail, and realName attributes are mapped to be returned back as part of AuthnRequest and the Attribute Query Request.

Error message: SAML user missing roles

You experience the following message:

ERROR UserManagerPro - user="samluser1" had no roles
Mitigation

Make sure that rolemap_SAML contains the correct role mapping with ";" at the end of each role name.

User cannot login

User cannot log in after successful assertion validation. No valid Splunk role is found in the local mapping or in the assertion.

Mitigation

- Make sure that rolemap_SAML stanza contains proper mapping between roles returned from IdP and the appropriate Splunk role.
- Make sure there are no spaces between, before, or after each role defined in authentication.conf. For example:

  user = User;Employee

User cannot access SAML login page

Authentication is configured as SAML and the settings appear to be correct, but the login screen shows the page for Splunk authentication instead.

Mitigation

- Make sure that in web.conf, appServerPorts is set to a valid port and not ‘0’.
- Make sure web.conf does not contain a value for trustedIP.

Error message: Failed to validate SAML logout response

When you log out of Splunk Enterprise or Splunk Cloud, you see the following error message:

Failed to validate SAML logout response received from IdP

Mitigation

This might be caused by case-sensitive IdPs that expect Splunk software to preserve uppercase letters in usernames. You can change the username to lowercase in the IdP or configure the IdP to accept the lowercase version of a username.

Cannot authenticate users for CLI commands

Unable to authenticate SSO users for CLI commands

Mitigation

You can add the SAML users as native Splunk users.

API and CLI commands cannot be performed by users that are defined only in SAML. This is because the user password is never sent in the SAML assertion.
Authentication using Proxy SSO

About ProxySSO

ProxySSO is an authentication method that lets you configure Single-Sign On (SSO) for Splunk instances through a reverse proxy server. A user logged in using SSO can seamlessly access Splunk Web.

With ProxySSO Single-Sign On, user identity and group information can be passed in HTTP headers to Splunk Enterprise. Splunk Enterprise uses this information to authenticate users and authorize them by mapping groups to appropriate Splunk Enterprise roles.

ProxySSO authentication:

• Combines authentication and authorization into one step for the user, streamlining the login process.
• Reduces configuration steps. No need to configure complex LDAP strategies within Splunk Enterprise.
• Reduces the back and forth messages between Splunk Enterprise and authentication services, making authentication more efficient.
• The external authentication service is not restricted to LDAP as long as the proxy server can pass the required information.

ProxySSO cannot be configured through Splunk Web. Instead you must use the REST API or modify configuration files as described in Configure ProxySSO.

Splunk Cloud does not support ProxySSO.

Prerequisites

To set up ProxySSO, you should already have the following configured:

• A Proxy Server configured to send required HTTP headers.
• A working Splunk Enterprise configuration.

For more information about how to configure these items and set up ProxySSO, see Configure ProxySSO.

How it works

1. The proxy server authenticates against the configured authentication service and creates an HTTP request.
2. Splunk Enterprise receives HTTP headers from the trusted reverse proxy server.
3. Splunk Enterprise checks trustedIP (which is configured in web.conf) for a receiving request from the proxy.

After a successful login, a session cookie is created and the user can seamlessly access Splunk Web.

Configure ProxySSO

Before you configure Splunk Enterprise, configure your proxy server so that it acts as a proxy for Splunk Web, prompts user for credentials, and passes user identity and groups to Splunk Web through HTTP headers:
Configure Splunk Enterprise

1. Configure web.conf

   [settings]
   SSOMode = strict
   trustedIP = 10.1.1.2
   remoteUser = Remote_User
   remoteGroups = Remote_Groups
   remoteGroupsQuoted = true
   allowSsoWithoutChangingServerConf = 1

2. Restart Splunk.

3. In authentication.conf configure the [authentication] stanza:

   [authentication]
   authType = ProxySSO
   authSettings = my_proxy

4. Map groups to Splunk roles in the roleMap_proxySSO stanza.

   [roleMap_proxySSO]
   admin = IT operational admin
   splunk-system-role = IT sub-admin

5. Configure the [my_proxy] stanza for additional settings. If a group mapping is not found, the role configured in defaultRoleIfMissing is assigned:

   [my_proxy]
   defaultRoleIfMissing = user

6. Reload authentication to enable your changes.

Troubleshoot Proxy SSO

You can view the HTTP request headers that proxy server sends to Splunk Web on the below endpoint after you set enableWebDebug=true in web.conf under settings stanza:


This endpoint will help to verify some of the common configuration or setup errors:
• Incoming request IP matches the configured value of trustedIP
• Ensure header attribute names set on proxy server are same as those configured on Splunk
• Make sure group entries are sent and parsed correctly. Especially, when remoteGroupsQuoted = true is set. You can see how groups are parsed by adding category.UiAuth=DEBUG in etc/log.cfg under splunkd stanza.

Once this is verified, check the following configuration:

• Groups parsed have mapping in roleMap_proxySSO
• In some cases, user cannot login because either the user or their roles are blacklisted. Check blacklisted objects under stanza named after value of authSettings

These kind of login events are logged in var/log/splunkd.log along with reason for failure.
Authentication using single sign-on with reverse proxy

About Single Sign-On using reverse proxy

Splunk Single Sign-on (SSO) lets you use a reverse proxy to handle Splunk authentication, meaning that once the user has logged into their proxy, they can seamlessly access Splunk Web (and presumably any other applications configured to your proxy).

The reverse proxy implementation of Splunk Enterprise SSO supports logging into Splunk Enterprise only through Splunk Web. Since the implementation relies on cookies to save authentication information, SSO cannot be used for CLI authentication to Splunk Enterprise. Invoking `https://localhost:8089` (or the assigned management port) still requires independent authentication.

For more information about how to configure these items and set up SSO, see Configure Single Sign-On

How it works

Splunk Enterprise administrators and users invoke Splunk Web via a proxy URL that is deployed with Splunk Web. The proxy authenticates the incoming request against your authentication system. Upon successful authentication the proxy sets a request header with the authenticated identity’s attribute and sends this information to Splunk Enterprise.

Splunk Enterprise accepts the incoming HTTP request from the proxy, and if Splunk Enterprise recognizes the user contained in the header, the user bypasses the login page and is automatically authorized.

For successful single sign-on, all requests from the proxy to Splunk Web must include this authenticated header. If the header is not included in a request, then the user is returned to the login page or an error page, depending on your configuration. Splunk software uses this authenticated header for the duration of the browser session.

How Splunk software processes the proxy request

When the proxy server makes a request to Splunk Web, Splunk Web looks to the `trustedIP` value in `web.conf` to verify that the proxy’s IP is on the trusted IP list.

If the IP is not trusted, the request is rejected and the sign-on attempt fails. If the IP address is trusted, then Splunk Web queries for the identity in the request header and sends splunkd an authorization request containing that header information.

Upon receiving the authorization request from Splunk Web, splunkd verifies whether the incoming IP address of the client (usually Splunk Web) matches the value of the `trustedIP` property of the `server.conf` file.

If the IP addresses are not in the `trustedIP` list the request is rejected and the sign-on attempt fails. The user is either returned to a login page or shown an error page, depending upon your SSO mode configuration in `web.conf`. For more on this attribute and other configuration information, see Configure Splunk Single Sign-On.

If the IP is trusted, then splunkd uses the information contained in the request header and conducts the authorization process.
How Splunk software authorizes the user

Splunk software first checks to see if the given identity and role matches any of your Splunk users. If no match is found, Splunk software looks to see if there are any LDAP matches. (For information about how Splunk software authenticates users, see Set up user authentication with LDAP in this manual.)

If no match is found and the user contained in the header cannot be authorized, then the browser redirects to an error page.

If a match is found, Splunk software authorizes the user and checks to see if an existing session is present. If a session already exists, Splunk software uses that session identifier and creates the necessary cookies to allow the user access to Splunk Web. If a session does not exist, then Splunk software creates a new session as well as the necessary cookies for Splunk Web authorization.

After the cookies are created, Splunk Web resumes its normal flow. Any subsequent access to Splunk Enterprise through the proxy URL does not require re-authorization as long as the request header contains the trusted identity and until the user closes the browser session.

Configure Single Sign-On with reverse proxy

Before you configure reverse proxy-based SSO with Splunk Enterprise, make sure you have the following:

- A Proxy Server (Splunk Enterprise supports IIS or Apache) configured as a reverse proxy to authenticate to external systems.
- An LDAP Server or other external authentication system provisioned with appropriate groups and users for your proxy to authenticate against.
- A working Splunk Enterprise configuration that is either configured to use the same external authentication system as your proxy (usually LDAP) or that has native Splunk Enterprise users that match the user and group IDs contained in your external authentication system.

Configuring SSO with reverse proxy requires the following steps:

1. Edit the properties on your proxy server to authenticate against your external authentication system.
2. Edit the Splunk Enterprise server.conf file.
3. Edit the Splunk Enterprise web.conf file.

Note: For optimal security, any HTTP header-based solutions should be implemented over a TLS/SSL enabled deployment.

Configure server.conf

Edit the trustedIP in the general settings stanza to add the IP address that will make secure authentication requests to splunkd. This is typically Splunk Web and therefore the localhost. You can only enter one IP address per splunkd instance.

trustedIP=127.0.0.1

If no IP addresses are provided in the trustedIP list, Splunk SSO is disabled by default.
Configure web.conf

To enable SSO, configure the following in the [settings] stanza in web.conf ($SPLUNK_HOME/etc/system/local):

SSOMode = strict
trustedIP = 127.0.0.1,10.3.1.61,10.1.8.81
remoteUser = Remote-User
tools.proxy.on = False

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSOMode</td>
<td>no</td>
<td>The SSOMode attribute determines whether the Splunk Web SSO operates in strict or permissive mode. Strict mode restricts authentication to identities that match the IP addresses listed in trustedIP property. If the IP attempting to connect does not match any IP address, an error page appears to the user. Strict mode is recommended for SSO. Permissive mode also restricts authentication to requests from IPs found in the trustedIP list. In permissive mode, if the IP attempting to connect does not match any IP address, a login page is displayed to allow the user to re-authenticate.</td>
</tr>
<tr>
<td>trustedIP</td>
<td>n/a</td>
<td>Set this to the IP address of the authenticating proxy or proxies. Specify a single address or a comma-separated list of addresses; IP ranges and netmask notation are not supported.</td>
</tr>
<tr>
<td>remoteUser</td>
<td>REMOTE_USER</td>
<td>The remoteUser attribute determines the authenticated identity's attribute that is passed by the proxy server via the HTTP request header. This value defaults to REMOTE_USER, but any LDAP attribute can be passed in this request header as long as the proxy sets this attribute properly after authentication. When you configure your remoteUser attribute, you must also configure the RequestHeader property in your proxy configuration to pass the identity's attribute to Splunk software. This process is described in &quot;About Splunk Single Sign-On&quot;. The default Splunk header used is REMOTE_USER, but if your proxy uses a different header, you can change the name of the header here.</td>
</tr>
<tr>
<td>tools.proxy.on</td>
<td>false</td>
<td>For apache 1.x proxy this value should be set to True. For later versions this value should be set to False.</td>
</tr>
</tbody>
</table>

If you host Splunk Web behind a proxy that does not place Splunk Web at the proxy's root, you may also need to configure the root_endpoint setting in $SPLUNK_HOME/etc/system/local/web.conf.

For example if your proxy hosts Splunk Web at "yourhost.com:9000/splunk", root_endpoint should be set to /splunk.

For example:

root_endpoint=/lzone

In the above example, Splunk Web is accessed via http://splunk.example.com:8000/lzone instead of http://splunk.example.com:8000/.

You would next make it visible to the proxy by mapping it in httpd.conf:

ProxyPass /lzone http://splunkweb.splunk.com:8000/lzone
ProxyPassReverse /lzone http://splunkweb.splunk.com:8000/lzone
Session management

Since there is no simple log out for a session and Splunk Enterprise will preserve a session as long as the correct header information is contained in the proxy header, you should set your proxy’s session timeout value with this in mind.

If you need to end a session before the timeout has occurred, you can use the REST end point along with the session identifier to destroy the session:

curl -s -uadmin:changeme -k -X DELETE
https://localhost:8089/services/authentication/httpauth-tokens/990cb3e61414376554a39e390471fff0

Troubleshoot reverse-proxy SSO

Splunk Web provides an interface that allows you to analyze the environment and the run-time data to help you debug your deployment. This page can be accessed via the proxy or the direct URL. The request headers will not be available if you do not access this page through the proxy server.

+ Splunk recommends that this setting is disabled after troubleshooting is complete.

This URL is located at:

http://YourSplunkServer:8000/debug/sso

Important: This debug page is not available by default. In order to make the page available, two steps must be completed. First, the role that is accessing this end point must have the web_debug capability, which the admin role has by default. Second, in web.conf, the setting enableWebDebug=true must be configured. You should immediately disable this setting after you have finished troubleshooting.

Consider the following when using the troubleshooting page to analyze your deployment:

- Compare the IP provided as the Splunk trusted IP with that of the Host IP. The values must be the same (they should be the IP of your proxy). If they are not the same in the troubleshooting page, you must edit the trustedIP value in server.conf.

- Check the value for Incoming request IP received by splunkweb to make sure that it displays your client’s IP address. If the IP does not match that of your client, you must:
  - Edit web.conf to correct this.
  - Make sure that tools.proxy.on is set to true.

- Make sure that your proxy is providing a header. Check the Authorization field under Other HTTP Headers. If there is no value present, check the http.conf file in your proxy to make sure that the remote header attribute value is properly set. Splunk software is configured to accept the remote header value of REMOTE_USER, which is the default for most proxies. If your proxy's remote header is different, and you wish to keep that value, you can edit the remote header value in web.conf to change the header that Splunk software will accept. See Configure SSO for more information.

- Make sure that Splunk Web is creating a cookie to send to splunkd. Check the Cookie field under Other HTTP headers to make sure that a cookie is set. If a cookie is not set, then check your web.conf file to make sure your file is properly configured. Configure SSO for more information.
Scripted authentication

Set up user authentication with external systems

Your options for user authentication are:

- Splunk authentication
- LDAP
- Single sign-on
- A scripted authentication API for use with an external authentication system, such as PAM or RADIUS, described in this section.

Important: Splunk authentication takes precedence over any external systems.

This is the order in which Splunk software authenticates a user for LDAP:

1. Splunk authentication or SSO.

2. LDAP or scripted authentication (if enabled). For more information about LDAP, see "Set up user authentication with LDAP".

How scripted authentication works

In scripted authentication, a user-generated Python script serves as the middleman between the Splunk server and an external authentication system such as PAM or RADIUS.

The API consists of a few functions that handle communications between Splunk software and the authentication system. You need to create a script with handlers that implement those functions.

To integrate your authentication system with Splunk Enterprise, make sure the authentication system is running and then do the following:

1. Create a Python authentication script. See "Create the authentication script" for the procedure.

2. Enable your script by editing authentication.conf to specify scripted authentication and associated settings. See "Edit authentication.conf" for the procedure.

Examples

Splunk provides several example authentication scripts and associated configuration files, including one set for RADIUS and another for PAM. There is also a simple script called dumbScripted.py, which focuses on the interaction between the script and Splunk deployments.

You can use an example script and configuration file as the starting point for creating your own script. You must modify them for your environment.
You can find these examples in $SPLUNK_HOME/share/splunk/authScriptSamples/. That directory also contains a README file with information on the examples, as well as additional information on setting up the connection between Splunk Enterprise and external systems.

Important: These scripts are provided as examples that you can modify or extend as needed. They are not supported and there is no guarantee that they will fully meet your authentication and security needs.

Create the authentication script

To integrate your authentication system with your Splunk deployment, make sure the authentication system is running and then do the following:

1. Create a Python authentication script. See "Create a Python script" in this topic for the procedure.
2. Test the new script. See "Test the script" in this topic for the procedure.
3. Enable your script by editing authentication.conf to specify scripted authentication and associated settings. See "Edit authentication.conf" for the procedure.

Create a Python script

You must create a Python script that implements these authentication functions:

- userLogin
- getUserInfo
- getUsers

The Splunk server will call these functions as necessary, either to authenticate user login or to obtain information on a user's roles.

The script can optionally also include a handler for this function:

- getSearchFilter

This table summarizes the authentication functions, their arguments, and their return values:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Argument string</th>
<th>Return value string</th>
</tr>
</thead>
</table>
| userLogin      | Login with user credentials.       | --username=<username> --password=<password>                                  | fail (safely passed over stdout)
|                |                                    | (values passed one per line over stdin)                                      |                     |
| getUserInfo    | Return a user's information,       | --username=<username> --userInfo=<userId>;<username>;<realname>;<roles>        |                     |
|                | including name and role(s).        |                                                                              | Note the following:  |
|                |                                    |                                                                              | • userInfo must specify a semicolon-delimited list. |

84
### getUsers

Return information for all Splunk users.

**Argument string:** none

**Return value string:**

```
--status=success
--userInfo=<userId>;<username>;<realname>;<roles>
```

**Note the following:**
- See `getUserInfo` for information on the syntax to use to return each user's information.
- Separate each user's information with a space.
- **<roles>** is required. To return multiple roles, use colons to separate the roles.
  - For example: `admin:power`
- This example returns just the roles for a user named "docsplunk":
  ```
  --status=success
  --userInfo=;docsplunk;;admin:power
  ```

### getSearchFilter

Optional. Returns the filters applied specifically to this user, along with those applied to the user's roles. The filters are OR'd together.

**Argument string:**

```
--username=<username>
```

**Return value string:**

```
--status=success|fail
--userInfo=<userId>;<username>;<realname>;<roles>
--search_filter=<filter>
```

**Note:** User-based search filters are optional and not recommended. A better approach is to assign search filters to roles and then assign users to the appropriate roles.

For more information, see "Use the `getSearchFilter` function to filter at search time"

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Argument string</th>
<th>Return value string</th>
</tr>
</thead>
</table>
| getUsers         | Return information for all Splunk users.         | none                             | --status=success
--userInfo=<userId>;<username>;<realname>;<roles>
--userInfo=<userId>;<username>;<realname>;<roles>
--userInfo=<userId>;<username>;<realname>;<roles>
...                                                                                                                                                                                                 |
| getSearchFilter  | Optional. Returns the filters applied specifically to this user, along with those applied to the user's roles. The filters are OR'd together. | --username=<username>            | --status=success|fail
--userInfo=<userId>;<username>;<realname>;<roles>
--search_filter=<filter>
--search_filter=<filter> ...

---

**Test the script**

Since the communication between your Splunk deployment and the script occurs via stdin and stdout, you can test the script interactively in your command shell. Be sure to send one argument per line and end each function call with an EOF (Ctrl-D).

Test each function individually, using this pattern:

```
> python [script] [function name]
[pass arguments here, one per line]
[send eof, with Ctrl-D]
[output appears here, check that it's correct]
```

The following example shows a debugging session that does some simple testing of a fictional script called "example.py", with two users "alice" and "bob". "alice" is a member of the "admin" and "super" roles, and "bob" is a member of the "user"
> python example.py userLogin
   --username=alice
   --password=correctpassword
<send an EOF>
   --status=success
> python example.py userLogin
   --username=bob
   --password=wrongpassword
<send an EOF>
   --status=fail
> python example.py getUsers
<no arguments for this function, send an EOF>
   --status=success
   --userInfo=bob;bob;bob;user
   --userInfo=alice;alice;alice;admin;super
> python example.py getUserInfo
   --username=bob
<send an EOF>
   --status=success
   --userInfo=bob;bob;bob;user
> python example.py getUserInfo
   --username=userdoesnotexist
<send an EOF>
   --status=fail

Important: This is just an example of how to go about testing a script. It does not attempt to perform exhaustive debugging of any real script.

Edit authentication.conf

To integrate your authentication system with your Splunk deployment, make sure the authentication system is running and then do the following:

1. Create and test a Python authentication script. See "Create the authentication script" for the procedure.

2. Edit authentication.conf to enable your authentication script. See "Enable your script" in this topic.

3. Edit authentication.conf to set your cache duration. See "Set cache durations" in this topic.

Enable your script

Once you create a Python script to implement authentication, you update the authentication.conf in
$SPLUNK_HOME/etc/system/local/ to enable your script. You can also copy and edit a sample authentication.conf from
$SPLUNK_HOME/share/splunk/authScriptSamples/.

Specify Scripted as your authentication type under the [authentication] stanza heading:

[authentication]
authType = Scripted
authSettings = script

Set script variables under the [script] stanza heading. For example:

[script]
Set cache durations

To significantly speed authentication performance when using scripted authentication, enable Splunk authentication caching. You do so by adding the optional [cacheTiming] stanza. Each script function (except getSearchFilter) has a settable cacheTiming attribute, which turns on caching for that function and specifies its cache duration. For example, to specify the cache timing for the getUserInfo function, use the getUserInfoTTL attribute. Caching for a function occurs only if its associated attribute is specified.

The cacheTiming settings specify the frequency at which Splunk software calls your script to communicate with the external authentication system. You can specify time in seconds (s), minutes (m), hours (h), days (d), etc. Typically, you'll limit the cache frequency to seconds or minutes. If a unit is not specified, the value defaults to seconds. So, a value of "5" is equivalent to "5s".

This example shows typical values for the caches:

```
[cacheTiming]
userLoginTTL = 10s
getUserInfoTTL = 1m
getUsersTTL = 2m
```

You'll want to set userLoginTTL to a low value, since this determines how long user login/password validity is cached.

To refresh all caches immediately, use the CLI command reload auth:

```
./splunk reload auth
```

**Note:** This command does not boot current users off the system.

You can also refresh caches in Splunk Web:

1. In the **System** menu, under **Users and authentication** select **Access controls**.

2. Click **Authentication method**.

3. Click **Reload authentication configuration** to refresh the caches.

Each specified function, except getUsers, has a separate cache for each user. So, if you have 10 users logged on and you've specified the getUserInfoTTL attribute, the getUserInfo function will have 10 user-based caches. The getUsers function encompasses all users, so it has a single, global cache.

Use PAM authentication

You can configure Splunk Enterprise to use PAM authentication by following the steps in the example directory's README, which is located at `$/SPLUNK_HOME/share/splunk/authScriptSamples/`.

If you are still unable to authenticate, then edit `/etc/pam.d/pamauth` and add this line:

```
auth sufficient pam_unix.so
```
Use the getSearchFilter function to filter at search time

This function is optional and can be used to implement a user-based filter at search time. When `getSearchFilter` is enabled, Splunk software calls it every time a search is run. A user-based search filter supplements any filters specified for that user’s role(s). The returned filter(s) will be applied to each search, along with any configured at the role level. Caching of the filter does not occur with this function.

**Note:** User-based search filters are optional and not recommended. A better approach is to assign search filters to roles and then assign users to the appropriate roles.

To enable `getSearchFilter` function, set the `scriptSearchFilters` parameter in `authentication.conf`:

```
[script]
scriptPath = $SPLUNK_HOME/bin/python $SPLUNK_HOME/bin/<scriptname.py>
scriptSearchFilters = 1
```

**Note:** In previous releases, `getSearchFilter` could also be used to implement search filters for users who had been authenticated by Splunk software. Starting with 4.2, `getSearchFilter` is called only for users who have been authenticated by scripted authorization.

In addition, if a call to `getSearchFilter` fails, Splunk Enterprise will cancel the user’s search and return an error message, to ensure that users cannot view results from unauthorized searches.
Securing Splunk Enterprise communications with SSL

About securing Splunk Enterprise with SSL

This section describes the types of Splunk configurations that you might want to secure with SSL.

About the default certificates

Splunk software ships with, and is configured to use, a set of default certificates. These certificates discourage casual snoopers but could still leave you vulnerable, because the root certificate is the same in every Splunk download and anyone with the same root certificate can authenticate.

The default certificates are generated and configured at startup and can be found in ${SPLUNK_HOME/etc/auth/}. They are set to expire three years after they are generated and new certificates must be created and configured at that time.

- For information about the default certificate for Splunk Web, see "Turn on encryption (https) with Splunk Web." or "Turn on encryption (https) using web.conf."
- For information about SSL for forwarding with the default certificate, see "Configure Splunk forwarding to use the default certificate."

Ways you can secure Splunk Enterprise

You can apply encryption and/or authentication using your own certificates for:

- Communications between the browser and Splunk Web
- Communication from Splunk forwarders to indexers
- Other types of communication, such as communications between Splunk instances over the management port

The table below describes the most common scenarios and the default SSL settings:

<table>
<thead>
<tr>
<th>Type of exchange</th>
<th>Node A function</th>
<th>Node B function</th>
<th>Encryption</th>
<th>Certificate Authentication</th>
<th>Common Name checking</th>
<th>Type of data exchanged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser to Splunk Web</td>
<td>Browser</td>
<td>Splunk Web</td>
<td>NOT enabled by default</td>
<td>dictated by client (browser)</td>
<td>dictated by client (browser)</td>
<td>configuration and search data</td>
</tr>
<tr>
<td>Splunk Web to search head</td>
<td>Splunk Web</td>
<td>splunkd as a search head</td>
<td>enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default</td>
<td>configuration and search data</td>
</tr>
<tr>
<td>Forwarding</td>
<td>splunkd as a forwarder</td>
<td>splunkd as an indexer</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default</td>
<td>data to be indexed</td>
</tr>
<tr>
<td>Deployment server to deployment clients</td>
<td>splunkd as a deployment client</td>
<td>splunkd as deployment server</td>
<td>enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default</td>
<td>configuration data</td>
</tr>
<tr>
<td>Distributed search</td>
<td>splunkd as search peer</td>
<td>splunkd as a search head</td>
<td>Enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default</td>
<td>search data</td>
</tr>
<tr>
<td>Type of exchange</td>
<td>Node A function</td>
<td>Node B function</td>
<td>Encryption</td>
<td>Certificate Authentication</td>
<td>Common Name checking</td>
<td>Type of data exchanged</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>------------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Search head clusters</td>
<td>splunkd as cluster members</td>
<td>splunkd as cluster members</td>
<td>Not enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default, SSL is not recommended, use Pass4SymmKey instead</td>
<td>cluster data</td>
</tr>
<tr>
<td>Search head cluster deployer</td>
<td>splunkd as cluster members</td>
<td>splunkd as cluster deployer</td>
<td>Enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default, SSL is not recommended, use Pass4SymmKey instead</td>
<td>configuration data</td>
</tr>
<tr>
<td>Indexer cluster peer nodes</td>
<td>splunkd as indexer cluster peer nodes</td>
<td>splunkd as indexer cluster peer nodes</td>
<td>NOT enabled by default. You must use the replication_port-ssl setting in server.conf to enable replication of data over SSL</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default, SSL is not recommended, use Pass4SymmKey instead</td>
<td>replication data</td>
</tr>
<tr>
<td>Indexer cluster master</td>
<td>splunkd as peer and search head nodes</td>
<td>splunkd as a master node</td>
<td>Enabled by default</td>
<td>NOT enabled by default</td>
<td>NOT enabled by default, SSL is not recommended, use Pass4SymmKey instead</td>
<td>cluster data</td>
</tr>
</tbody>
</table>

**Communications between the browser and Splunk Web**

Browser to Splunk Web data most commonly consists of search requests and returned data.

Data encryption (HTTPS) can be easily turned on using Splunk Web, or by editing the configuration files. Keep in mind that encryption with the default certificate protects against casual listening but is not fully secure.

For better security, replace the default certificates with certificates signed by a trusted CA. We strongly recommend using CA certs rather than signing your own in this case. Unless you have the ability to add your CA to the certificate stores in every browser that will access Splunk Web, a self-signed certificate is considered untrusted by users' browsers. For more information, see "About securing Splunk Web."

**Splunk forwarders to indexers**

Data sent from forwarders to indexers is the data that your indexers use for searches and reports. Depending upon your organization and the nature and format of the data being transmitted and Splunk configuration, this data may or may not be readable or sensitive.

Securing sensitive raw data helps to avoid snooping and man-in-the-middle attacks.

You can turn on SSL encryption using the default certificate to provide encryption and compression. However, communication using the default certificate does not provide secure authentication, as the certificate password is supplied with every installation of Splunk software. The default certificates are set to expire three years after initial startup, and forwarder to indexer communications will fail at this point.

For better security, require certificate authentication using a self- or CA-signed certificate. A certificate signed by a known and mutually trusted Certificate Authority is considered more secure by outside parties than a certificate you sign yourself.
For more information about using certificates with Splunk forwarders and indexers, see "About securing data from forwarders."

**Other SSL communications**

Other Splunk communications happen between different instances of Splunk software over the management port, usually but not always in a distributed environment. An example of this is configuration data sent by a deployment server to clients. This type of SSL encryption is enabled by default. For most configurations this is adequate and is the recommended security method. However, if you do need to secure your communications with SSL authentication, we've provided some guidelines to help you in "About securing Splunk to Splunk communication" in this manual.

To learn about more ways to use TLS certificates, see the following topics:

- Secure LDAP with TLS certificates
- Secure SSO with TLS certificates

**Getting your certificates**

If you are experienced with SSL certificates, you can create them as you normally would and go straight to configuring your Splunk instances to use them.

If you need help getting your certificates together, we've provided very simple examples using OpenSSL commands. (OpenSSL ships with Splunk software)

- How to self-sign certificates
- How to get third-party certificates
- How to self-sign certificates for Splunk Web
- How to get third-party certificates for Splunk Web

**What to do when you have your certificates**

The following topics provide more information about configuring Splunk software to use your certificates once you have them:

- Secure Splunk Web with your own certificate
- Configure Splunk forwarding to use your own certificates
- About securing inter-splunk communication

**About using SSL tools on Windows and Linux**

This manual describes how to configure Splunk deployments to use default, self-signed, or Certificate Authority signed certificates. For those who may not have certificates, we also provide simple examples for generating the certificates and keys using the command line and a version of OpenSSL that is packaged with Splunk software.

**Using the OpenSSL command-line examples**

This manual provides a few basic examples for creating certificates using the Splunk version of OpenSSL in the command line. In order to perform these tasks you must have root administrator permissions. If you are working on a remote or
virtual machine, you may have to take an extra step to ensure that you are able to perform all tasks:

- When working on a Windows platform, you may need to open the command line as the administrator: In the Start Menu, right click the .exe application and select **run as administrator**.

- When working on a *nix platform, you might need to use `sudo` to log in as the root administrator.

For more information about the differences between Windows and *nix, see the Administration Guide.

**About SSL tools**

Splunk software ships with a recent version of OpenSSL at `$SPLUNK_HOME/splunk/lib`. For 6.0, Splunk supports OpenSSL with FIPS 140-2 enabled.

A variety of other SSL tools are available for purchase and download that you can use to create and set up certificates. If you do choose to use OpenSSL for certificate configuration, we strongly recommend that you use the version that ships with Splunk to avoid compatibility issues. To make sure that you are using the version provided with Splunk software, set your environment to the version in `$SPLUNK_HOME/splunk/lib` or `$SPLUNK_HOME\splunk\bin` for Windows:

The following is an example of the library path for *nix:

```bash
export LD_LIBRARY_PATH=$SPLUNK_HOME/splunk/lib
```

The following is an example of the path for Windows (using the command prompt):

```command
set PATH = %PATH%;%SPLUNK_HOME%;bin
```

**Configure allowed and restricted SSL versions**

Splunk Enterprise version 6.2 and later provides the `sslVersions` keyword to restrict older versions of protocols. SSLv3 is shipped out of box to support easy upgrades but should be disabled as soon as upgrades are complete. By default, Splunk Enterprise allows communications on SSLv3 and all subsequent versions.

When Splunk Enterprise is secured with FIPS, SSLv2 and SSLv3 are always disabled regardless of any additional configuration.

**CAUTION:** To avoid the v3 "POODLE" vulnerability, remove SSLv3 as upgrades are applied to your environment.

**Configure `web.conf`**

1. In `web.conf`, update the `sslVersions` attribute to list or limit the versions (separated by commas) you want to permit. By defaults this attribute is set to `*-sslv2`, which is any version newer than SSLv2 (not recommended). For 6.2 the allowed SSL versions are:

   - SSLv2 (not recommended)
   - SSLv3 (not recommended)
   - TLS1.0 (not recommended)
   - TLS1.1
   - TLS1.2

   For example:

   ```
   sslVersions = tls1.1, tls1.2
   ```
Syntax options

To select all supported versions use "*":

```plaintext
sslVersions = *
```

To include all versions tls1.0 or newer use "tls":

```plaintext
sslVersions = tls
```

To restrict a particular version prefix it with "-":

```plaintext
sslVersions = *, -ssl3
```

**Note:** When Splunk Enterprise is configured in FIPS mode, SSLv2 and SSLv3 are always disabled regardless of this configuration.

2. In `inputs.conf`, update the `sslVersions` attribute to list or limit the versions (separated by commas) you want Splunk Enterprise to support.

```plaintext
sslVersions = ssl2, tls1.1, tls1.2
```

You can use "*" to select all supported versions:

```plaintext
sslVersions = *
```

Simply use "tls" to include all versions tls1.1 or newer:

```plaintext
sslVersions = tls
```

The prefix a version with "-" to restrict a particular version:

```plaintext
sslVersions = *, -ssl3
```

3. Configure forwarders to be compatible with your indexer. Changing or limiting the SSL versions (and restricting SSLv3) can create compatibility issues with forwarders, particularly those that run earlier versions of Splunk Enterprise. For forwarders running 6.2 you can mitigate compatibility issues by also updating each forwarder's `inputs.conf` and `web.conf` settings in addition to your indexer.

Update any forwarders to 6.2 to be consistent with your indexer and the SSL settings (For purposes of backward compatibility, 6.0 can support up to tls1.1.)

**Configure server.conf**

Configure your `server.conf` file to accept connections with clients. In other words, you would configure `web.conf` by editing the `sslVersions` attribute so that it is the same as your the version configured in your `server.conf` file for your client(s).

For example:

```plaintext
[sslConfig]
sslVersions = tls1.1, tls1.2
```

**Caveats to configuring splunkd with server.conf**

If you have configured Splunk Enterprise deployment clients with SSL, confirm that you also configure Splunk Enterprise deployment servers to listen over SSL. If you do not, then deployment clients cannot connect to those servers, even if you have correctly configured certificates and TLS properly. To ensure that deployment clients can connect to deployment servers, review the `server.conf` configuration file on the deployment servers and confirm that it has at least the following setting:
[sslConfig]
enableSplunkdSSL = true
Certificates for Splunk

About creating certificates for Splunk

There are multiple ways you can create signed certificates, depending upon your organizations policies, your platform, and the tools that you are using. If you have already generated these certificates and key, or if you are experienced in generating certificates, you can skip this task and go directly to the configuration topic "Secure Splunk Web with your own certificate" in this manual.

Since self-signed certificates are signed by your organization, they are not contained in browser certificate stores. As a result, web browsers consider self-signed certificates "untrusted". This produces a warning page to users and may even prevent access for the user.

Things to know about your certificates

Make sure you are using a compatible version of OpenSSL

Make sure that you are using the version of OpenSSL provided with Splunk by setting your environment to the version in $SPLUNK_HOME/lib in *nix or $SPLUNK_HOME/bin in Windows.

Decide between self-signed or third-party certificates

Self-signed certificates are best used for browser to Splunk Web communication that happens within an organization or between known entities where you can add your own CA to all browser stores that will contact Splunk Web. For any other scenario, CA-signed certificates are recommended. See "Get certificates signed by a third party for Splunk Web" for more information.

Remove your password from browser certificates

When you create a new private key for Splunk Web, you must generate a new private key and remove the password. We recommend that you generate a new private key especially for browser to Splunk Web encryption so that you do not remove the password from the keys you use elsewhere.

1. Generate a new private key:

2. When prompted, create a password.

3. Remove the password from your key. (Splunk Web does not currently support password-protected private keys.)

   In *nix:

   $SPLUNK_HOME/bin/splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -out mySplunkWebPrivateKey.key

   In Windows:

   $SPLUNK_HOME\bin\splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -out mySplunkWebPrivateKey.key

You can make sure your password is gone by issuing the following command:

   In *nix:
Create a single PEM file for Splunk

Combine your server certificate and public certificates, in that order, into a single PEM file.

Set up certificate chains for Splunk

To use multiple certificates, append the intermediate certificate to the end of the server’s certificate file in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

About cipher suites and TLS encryption

As of version 6.6, Splunk provides the following default cipher suites and TLS encryption. If you are upgrading from a previous version, you must update your existing certificates to be compatible with later versions.

```
inputs.conf

sslVersions = tls1.2
ecdhCurves = prime256v1, secp384r1, secp521r1
```

This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set `sslVersions = tls`

2. Add the following ciphers to the end of the existing `cipherSuite`:

```
DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA
```
outputs.conf

sslVersions = tls1.2
ecdhCurves = prime256v1, secp384r1, secp521r1

This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions to tls
2. Add the following ciphers to the end of the existing cipherSuite:

DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA

server.conf

sslVersions = tls1.2
sslVersionsForClient = tls1.2
AES256-GCM-SHA384:AES128-GCM-SHA256:AES128-SHA256
ecdhCurves = prime256v1, secp384r1, secp521r1

This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions = tls
2. Set sslVersionsForClient = tls
3. Append AES256-SHA to the existing cipherSuite.

applicationsManagement

sslVersions = tls1.2
ecdhCurves = prime256v1, secp384r1, secp521r1

This configuration does not support Splunk 5.x. To add support for Splunk 5.x:

1. Set sslVersions = tls
2. Add the following ciphers to the end of the existing cipherSuite:

DHE-RSA-AES256-SHA:AES256-SHA:DHE-RSA-AES128-SHA:AES128-SHA:
AES256-SHA:AES128-SHA
web.conf

sslVersions = tls1.2
ecdhCurves = prime256v1, secp384r1, secp521r1

This configuration does not support Windows Vista. To add support for Windows Vista:

1. Set sslVersions = tls

2. Add the following ciphers to the existing cipherSuite:

ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:
ECDHE-RSA-AES128-SHA

ldap.conf

TLS_PROTOCOL_MIN: 3.1 for TLSv1.0, 3.2 for TLSv1.1, 3.3 for TLSv1.2.
TLS_PROTOCOL_MIN 3.3

This configuration does not support Windows Server 2008 R2. To add support for Windows Server 2008 R2:

1. Set TLS_PROTOCOL_MIN = TLS1.0/SSL3.1

2. Add the following ciphers to the existing TLS_CIPHER_SUITE:

ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:
ECDHE-RSA-AES128-SHA

To enable TLS 1.2 support on Windows Server 2008 R2:

1. Add key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server

2. In the TLS 1.2\Server key, create the following:

DWORD (32-bit) Value ? DisabledByDefault; set to 0
DWORD (32-bit) Value ? Enabled; set to 1


How to prepare your signed certificates for Splunk authentication

Once you have your certificates, you must combine the server certificate and your keys into a single file that Splunk software can use.
If you do not have your certificates and need help getting them, we provide some basic examples using OpenSSL in the following topics:

- How to self-sign certificates.
- How to get certificates signed by a third party.

**Note:** Make sure your certificates and public key are in x509 format and that your private key is in RSA format.

**Create a single PEM file**

Combine your server certificate and public certificate, in that order, into a single PEM file.

For the examples here, we are using the file names described in "How to self-sign certificates" and "How to get certificates signed by a third party."

The following is an example for *nix:

```bash
cat myServerCertificate.pem myServerPrivateKey.key myCACertificate.pem > myNewServerCertificate.pem
```

The following is an example for Windows:

```bash
>type myServerCertificate.pem myServerPrivateKey.key myCACertificate.pem > myNewServerCertificate.pem
```

Once created, the contents of the file `myNewServerCertificate.pem` should contain, in the following order:

- The server certificate (`myServerCertificate.pem`)
- The private key (`myServerPrivateKey.key`)
- The certificate authority public key (`myCACertificate.pem`)

Here's an example of a properly concatenated certificate:

```plaintext
-----BEGIN CERTIFICATE-----
MIICUTCCAbocCQQcBkn/xeyITANBgkqhkiG9w0BAQUFADBtMQswCQYDVQQGEwJV...
<Server Certificate>
...
8/PZr3EuXYkic+N5hQIqys5a/HIn
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,CFCECC7976725DE5
S+DpcQ0l2Z1bk7JN3bBqr/nwEXPNDQ4uqtecCd3iGMV3B/WSOWAQxcWzhe9JnIsl...
<Server Private Key ? Passphrase protected>
...
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIICUTCCAbocCQQcBkn/xeyITANBgkqhkiG9w0BAQUFADBtMQswCQYDVQQGEwJV...
<Certificate Authority Public Key>
...
8/PZr3EuXYkic+N5hQIqys5a/HIn
-----END CERTIFICATE-----
```
How to configure certificate chains

To use multiple certificates, append the intermediate certificate to the end of the server’s certificate file. You can add as many certificates you need to in decreasing order of hierarchy, up to the root.

The certificates should be concatenated in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

In another example, when using Splunk Forwarder to Indexer Certificates that contain a Private Key, the completed certificate file might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
...<Server Private Key ? Passphrase protected>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

Next steps

Now that you have the certificates you need, you must configure Splunk software to find and use them:

- See "Configure Splunk forwarding to use your own certificates" to learn more about configuring certificate authentication for forwarding.
- See "About securing inter-Splunk communication" to learn more about configuring certificate authentication for inter-Splunk communications.

Determine your cipher suite

You can select and specify a cipher suite for inter-Splunk, Splunk Web, and Splunk forwarder to indexer communications. You add your cipher suite by appending a line at the end of your server SSL configuration stanza.
The following is an example of how you would update `inputs.conf` when configuring forwarder to indexer certificate authentication:

```bash
[splunktcp-ssl:9998]
[SSL]
password = password
requireClientCert = false
rootCA = $SPLUNK_HOME/etc/auth/cacert.pem
serverCert = $SPLUNK_HOME/etc/auth/server.pem
cipherSuite = AES256-SHA256:DHE-RSA-AES256-SHA256
```

To see which ciphers are available to you:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl ciphers -v
$SPLUNK_HOME/bin/splunk cmd openssl ciphers -v "TLSv1.2"
$SPLUNK_HOME/bin/splunk cmd openssl ciphers -v "HIGH"
```

Cipher suites are available to you based on your version of OpenSSL. To see which version of OpenSSL you are running:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl version
```

### Working with multiple intermediate certificates

To use multiple certificates, append the intermediate certificate to the end of the server's certificate file. You can include multiple intermediate certificates as a chain. You can add as many certificates as you need to in decreasing order of hierarchy, up to the root.

The certificates should be concatenated in the following order:

```bash
[ server certificate]
[ intermediate certificate]
[ intermediate certificate]
[ intermediate certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```
Secure browser to Splunk Web communication

About securing Splunk Web

Information transmitted to Splunk Web mostly consists of search requests and results.

Note that browser to Splunk Web transmission does not always need to be secured. For example, if your users only access Splunk Web from a local browser behind the same firewall as Splunk Web, security may not be a concern. In this case simple encryption using Splunk's default certificates might be adequate.

- For information about the default certificate for Splunk Web, see Turn on encryption (https) with Splunk Web. or
  Turn on encryption (https) using web.conf.
- For information about SSL for forwarding with the default certificate, see Configure Splunk forwarding to use the default certificate.

To turn on basic encryption, see Turn on encryption (https) with Splunk Web.

On the other hand, if your Splunk configuration lives in a distributed environment where Splunk Web is accessed from browsers outside of firewalls from varied locations, stronger security should be implemented using signed certificates. For information about configuring Splunk Web to use signed certificates, see Secure Splunk Web using your own certificate.

There are several ways you can use signed certificates to improve security for your browser to Splunk Web communications:

- For secured encryption with authentication, you can replace the default certificate with a signed certificate. You replace the default certificate provided by Splunk with one that you request from a trusted Certificate Authority. This is the most secure option and recommended if security is a concern.
  For more information about obtaining CA certificates for Splunk deployments, see Get certificates signed by a third-party for Splunk Web.
  Note that you may also use self-signed certificates to secure authentication, however, because they are signed by you rather than a known and trusted Certificate Authority, browsers will not have you as a CA in their certificate store and as a result will not trust you or your certificates. For self-signed certificates to be effective you would need the ability to add your certificate to the certificate store of every single browser that will access Splunk Web.
  For more information about creating self-signed certificates for Splunk deployments, see Self-sign certificates for Splunk Web.

- When you use a signed certificate, you can further strengthen your SSL configuration by turning on common name checking. Common name checking adds an extra layer of security by requiring that the common name provided in the certificates on each communicating instance are a match. You can enable common name checking when setting up your certificate and configure Splunk Enterprise to check for that common name when authenticating.

For more information about configuring Splunk Enterprise to use certificates and learn more about common name checking, see Secure Splunk Web using your own certificate.
**Turn on encryption (https) with Splunk Web**

This topic explains how to use Splunk Web to enable HTTPS for browser to Splunk Web communication. Splunk software can listen on HTTPS or HTTP, but not both.

The simple encryption that can be turned on in Splunk Web uses the default certificate that is provided in the "out of box" installation. Since every installation provides the same default certificate, this method is not highly secure. If security is a priority, change the default certificate and configure authentication for better security. See Secure Splunk Web with your own certificate for information about replacing the default certificates.

To enable HTTPS with Splunk Web:

1. In Splunk Web, select **Settings > System > Server settings**, and then click **General Settings**.

2. Under **Splunk Web**, for **Enable SSL (HTTPS) in Splunk Web**, select the **Yes** radio button.

   By default, Splunk deployments point to the default certificates when encryption is turned on, so no further action is needed.


   You must now prepend "https://" to the URL you use to access Splunk Web.

**Turn on encryption (https) using web.conf**

You can enable HTTPS through the `web.conf` configuration file. If it is not already present in your local directory, copy the default version of the file from `$SPLUNK_HOME/etc/system/default` to your local directory `$SPLUNK_HOME/etc/system/local/` or your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see About configuration files.

The encryption that can be turned on in the task described here is not secure. If security is a priority, change the default certificate and configure authentication for better security. See Secure Splunk Web with your own certificate for information about replacing the default certificates.

To enable HTTPS through `web.conf`:

1. Set the `enableSplunkWebSSL` attribute to `true`:

   ```
   [settings]
   httpport = <https port number>
   enableSplunkWebSSL = true
   ```

   **Note:** By default, Splunk software points to the default certificates when encryption is turned on.

2. Restart Splunk.

   You must now prepend "https://" to the URL you use to access Splunk Web.
Secure Splunk Web with your own certificate

This example assumes that you have already generated self-signed certificates or purchased third-party certificates. If you have not done this and are unsure how to proceed, we've provided some simple examples:

- Self-sign certificates for Splunk Web.
- Get a certificates signed by a third-party for Splunk Web.

Before you begin: make sure your certificate and key are available from your folder. In this example we are using $SPLUNK_HOME/etc/auth/mycerts/:

- $SPLUNK_HOME/etc/auth/mycerts/mySplunkWebCertificate.pem
- $SPLUNK_HOME/etc/auth/mycerts/mySplunkWebPrivateKey.key

Configure Splunk Web to use the key and certificate files

1. In $SPLUNK_HOME/etc/system/local/web.conf (or any other applicable location, if you are using a deployment server), make the following changes to the [settings] stanza:

The following is an example of an edited settings stanza:

```
[settings]
enableSplunkWebSSL = true
privKeyPath = </home/etc/auth/mycerts/mySplunkWebPrivateKey.key >
Absolute paths may be used. non-absolute paths are relative to $SPLUNK_HOME

serverCert = </home/etc/auth/mycerts/mySplunkWebCertificate.pem >
Absolute paths may be used. non-absolute paths are relative to $SPLUNK_HOME
```

2. Restart Splunk:

```
# $SPLUNK_HOME/bin/splunk restart splunkd
```

Troubleshoot your Splunk Web authentication

If you are unable to verify your certificate configuration, you can use the web_service.log in $SPLUNK_HOME/var/log/splunk to view and troubleshoot any errors that occur upon restart.

Look for SSL configuration warnings. For example, if you provide an incorrect path to the server certificate declared in serverCert, Splunk Web fails to start and the following error appears:

```
```

**Note:** If the private key is provided in privKeyPath is password protected, no error is provided but your browser won't load Splunk Web.

See Self-sign certificates for Splunk Web or Get certificates signed by a third party for Splunk Web for information about removing your password.
Secure Splunk forwarder to indexer communication

About securing data from forwarders

Forwarders send raw data to your indexers. This data can be vulnerable to snooping and corruption. If data is forwarded outside of a closed or co-located network, or if your data is very sensitive you should use SSL certificates to secure your data.

Using the default certificates will discourage casual snoopers but could still leave you vulnerable because the root certificate that ships with Splunk software is the same root certificate in every download, and anyone with the same root certificate can authenticate. The default certificates are generated and configured at startup and can be found in $SPLUNK_HOME/etc/auth/.

Important: If you use the default certificates, keep in mind that they are set to expire three years after they are generated and new certificates must be created and configured at that time using one of the methods described in this manual.

For information about setting up SSL with the default certificate, see Configure Splunk forwarding to use the default certificate.

To ensure that no one can easily snoop on your traffic or send data to your indexers, we recommend that you use new signed certificates that are either self-signed or purchased from a third-party certificate authority. To configure your forwarders and indexers to use certificates, see Configure Splunk forwarding to use your own certificates.

There are several ways you can use self or CA-signed certificates to improve security for your forwarder to indexer:

- **You can replace the default certificates with certificates signed by your own root CA.**
  You replace the default certificate provided by Splunk with one that you generate and sign yourself. For information about generating and self-signing certificates, see How to self-sign certificates.

- **You can replace the default certificates with certificates signed by a trusted certificate authority.**
  See How to get certificates signed by a third-party.

- **You can further strengthen security by configuring common name checking.**
  Common name checking adds an extra layer of security by requiring that the common name provided in the certificates on each indexer match the common name specified in the configuration file on the forwarder. You can also configure multiple certificates with different common names and distribute them to your indexers. You enable common name checking when setting up your certificate. See Configure Splunk forwarding to use your own certificates for more information.

Configure Splunk forwarding to use the default certificate

The default root certificate that ships with Splunk software is the same root certificate in every download. Splunk software uses these certificates by default and you should not need to configure them unless they have been removed or corrupted.

Default certificates are not considered highly secure. Anyone who has downloaded Splunk Enterprise has server certificates signed by the same root certificate and are able to authenticate to your certificates. To ensure that no one can easily snoop on your traffic or wrongfully send data to your indexers, we recommend that you replace them with signed certificates.
Important: The default certificates are set to expire three years after they are generated and new certificates must be created and configured at that time using one of the methods described in this manual.

To configure your forwarders to use certificates signed by your own root CA or a third-party CA see Configure Splunk forwarding to use your own certificates.

In this topic we describe how to:

- Configure the indexer to use the default certificates that ship with Splunk software
- Configure the forwarder to use the default certificates that ship with Splunk software

Set up the indexer to use the default server certificate

1. In $SPLUNK_HOME/etc/system/local/inputs.conf (or the appropriate directory of any app you are using to distribute your forwarding configuration), set up the following stanzas:

   [splunktcp-ssl:9997]
   disabled = 0

   [SSL]
   serverCert = Absolute path to the certificate. The default certificate is $SPLUNK_HOME/etc/auth/server.pem
   sslPassword = password
   requireClientCert= false (There is no need validate the default server certificate)

2. Your server.conf should also have the following (skip this for Windows configurations):

   [sslConfig]
   sslRootCAPath = /opt/splunk/etc/auth/cacert.pem

3. Restart splunkd:

   $SPLUNK_HOME/bin/splunk restart splunkd

Configure the forwarder

Set up your forwarder to use the same default certificates as your indexer and configure the forwarder to send data to the configured listening port.

1. Define the following stanzas in $SPLUNK_HOME/etc/system/local/outputs.conf (or in the appropriate directory of any app you are using to distribute your forwarding configuration):

   [tcpout:group1]
   server = 10.1.12.112:9997
   clientCert = $SPLUNK_HOME/etc/auth/server.pem
sslPassword = password

sslVerifyServerCert = false

2. Your forwarder's server.conf should also have the following:

Linux:

[sslConfig]

sslRootCAPath = /opt/splunkforwarder/etc/auth/cacert.pem

Windows:

cacert.pem

capath = $SPLUNK_HOME\etc\auth

3. Restart splunkd:

# $SPLUNK_HOME/bin/splunk restart splunkd

Next steps

Next, you should check your connection to make sure your configuration works. See "Validate your configuration" for more information.

Configure Splunk forwarding to use your own certificates

You can send data from your forwarders to your indexers using your own certificates. You can self-sign these certificates, or use a third party to sign them. Using your own certificates to secure Splunk communications involves the following procedures:

- Configuring indexers to use a new signed certificate, as described in this topic.
- Configuring forwarders to use a new signed certificate, as described in this topic.

Prerequisites

Before you can secure communications between Splunk indexers and forwarders, you must procure and prepare the certificates. You must satisfy the following conditions:

- The certificates that you procure are Privacy-Enhanced Mail (.pem) files, and that the format of those files conforms with the x.509 public key certificate standard.
- The certificate key must be in RSA security format.

You can also create multiple certificates that are signed by the same Certificate Authority (CA) with different common names, and distribute those to your indexers for added security. When you give the forwarder the CA public key, the forwarder trusts the CA, verifies the certificate of the CA, and matches one of the SSL common names or alternative names (as configured by either the sslCommonNameToCheck or sslAltNameToCheck settings in the forwarder configuration file.

If you need help on creating and preparing your own certificates, see the following topics for more information:

- About securing data from forwarders
 Configure your indexer to use a signed certificates

1. Copy your server certificate and CA public certificate into an accessible folder on the indexer you want to configure. For example, you can use a destination folder of $SPLUNK_HOME/etc/auth/mycerts/

   If you configure inputs.conf or outputs.conf in an app directory, the indexer does not encrypt the password, and the clear-text value remains in the file. You might want to create different certificates (signed by the same root CA) to use when configuring SSL in app directories.

2. Configure inputs.conf on the indexer to use the new server certificate. Add the following stanzas to $SPLUNK_HOME/etc/system/local/inputs.conf (or the appropriate directory of any app you are using to distribute your forwarding configuration), stanzas:

   [splunktcp-ssl:9997]
   disabled=0

   [SSL]
   serverCert = <Absolute path to the certificate. The default certificate can be found at $SPLUNK_HOME/etc/auth/>
   sslPassword = <password associated with the server certificate, if it exists>
   requireClientCert = "true" if you want your indexer to require authentication from the client (which in this case is the forwarder), "false" otherwise
   sslVersions (Optional) = <string of accepted password SSL versions. Default: the recommended setting of "*,-ssl2", which is anything newer than SSLv2.>
   cipherSuite (Optional) = <cipher suite string. If not set, the indexer uses the default cipher string>
   sslCommonNameToCheck (Optional) = <commonName1>, <commonName2>, ...
   If provided, the indexer checks the common name of the client certificate against this list of names. If there is no match the Splunk instance is not authenticated. You must set the 'requireClientCert' setting to "true" to use this setting.
   sslAltNameToCheck (Optional) = <alternateName1>, <alternateName2>, ... If provided, the indexer checks the alternate name of the client certificate against this list of names. If there is no match the Splunk instance is not authenticated. requireClientCert setting must be set to "true" to use this setting.

   When you edit the file in $SPLUNK_HOME/etc/system/local/inputs.conf, the indexer encrypts the password and overwrites the clear-text server certificate password that you provided when you restarted Splunk Enterprise.

3. On indexers that do not run on Windows, configure server.conf and add the following:

   sslRootCAPath = <Absolute path to the CA certificate, for example, the default value is $SPLUNK_HOME/etc/auth/cacert.pem>

4. Restart the splunkd process:

   # $SPLUNK_HOME/bin/splunk restart splunkd
Configure your forwarders to use your certificates

1. Generate a new certificate (for example, client.pem).
2. Copy the new certificate and the CA public certificate myCACertificate.pem into an accessible folder on the forwarders you want to configure. For example, you can use a destination folder of $SPLUNK_HOME/etc/auth/mycerts/.

If you configure inputs.conf or outputs.conf in an app directory, the forwarder does not encrypt the passwords and the clear-text value remains in the file. You might want to create different certificates (signed by the same root CA) to use when configuring SSL in app directories.
3. Define the [SSL] stanza in $SPLUNK_HOME/etc/system/local/outputs.conf (or in the appropriate directory of any app you use to distribute your forwarding configuration):

[tcpout:group1]
server=10.1.1.197:9997

disabled = 0

clientCert = <The full path to the client SSL certificate, in PEM format. If this value is provided, the connection will use SSL.>

useClientSSLCompression = true (Disabling TLS compression can cause bandwidth issues.)

sslPassword = <password for the client certificate>

sslCommonNameToCheck (Optional) = <commonName1>, <commonName2>, ...

sslVerifyServerCert (Optional) = "true" if you want to use SSL common name checking. Default: No common name checking.

sslAltNameToCheck (Optional) = <alternateName1>, <alternateName2>, ...

cipherSuite = (Optional) Splunk uses any specified cipher string for the input processors. If not set, Splunk uses the default cipher string provided by OpenSSL.

When you save the file in $SPLUNK_HOME/etc/system/local/outputs.conf, Splunk Enterprise encrypts and overwrites the clear-text server certificate password on restart.

4. On forwarders that do not run on Windows, configure server.conf and add the following:

[sslConfig]
sslRootCAPath = <absolute path to the CA cert, for example, the default value is $SPLUNK_HOME/etc/auth/cacert.pem>

5. Restart the splunkd process.

Forward data over SSL to more than one indexer

If you need to forward data securely to multiple indexers, complete the following procedure:

1. On the forwarder where you want to send data to multiple indexers, edit outputs.conf on the forwarder.
2. In the target output group definition stanza for the forwarder, add a host:port entry for each indexer to which you want to send data over SSL. Separate multiple entries with commas.
3. Save the outputs.conf file and close it.
4. Restart the forwarder.
The following example outputs.conf file uses the same certificate for the indexer and the forwarders:

```
[tcpout]
[tcpout:group1]
server = 10.1.12.112:9997,10.1.12.111:9999
disabled = 0
clientCert = $SPLUNK_HOME/etc/auth/client.pem
useClientSSLCompression = true Defaults to the value set in the useClientSSLCompression attribute set in server.conf.
sslPassword = <password for the client certificate>
sslCommonNameToCheck = indexercn.example.org
sslVerifyServerCert = true
```

**Forward data to multiple indexers using certificates with different common names**

You can create and configure one server certificate for each indexer by configuring outputs.conf on the forwarder with one server-specific [SSLConfig] stanza for each indexer.

If you have created one server certificate for each indexer and you have set a unique `sslCommonNameToCheck` or `sslAltNameToCheck` in each indexer certificate to be checked by the forwarders, you must configure one `[tcpout-server://host:port]` configuration stanza for each indexer in outputs.conf. This is so that you can specify which name to check for each indexer.

**Next steps**

Check your forwarder-indexer configuration to make sure it works. See Validate your configuration.

**Validate your configuration**

To verify your SSL connections in Splunk Web, try the following command:

```
index=_internal source=*metrics.log* group=tcpin_connections |
dedup hostname | table _time hostname version sourceIp destPort ssl
```

You can also `splunkd.log` to validate and troubleshoot your configuration. Splunkd.log is located on your indexer and forwarder at `$SPLUNK_HOME/var/log/splunk/splunkd.log`.

On the indexer, look for the following or similar messages at the start-up sequence to verify a successful connection:

```
02-06-2011 19:19:01.552 INFO TcpInputProc - using queueSize 1000
02-06-2011 19:19:01.552 INFO TcpInputProc - SSL cipherSuite=ALL:!aNULL:!eNULL:!LOW:!EXP:
RC4+RSA:+HIGH:+MEDIUM
02-06-2011 19:19:01.552 INFO TcpInputProc - supporting SSL v2/v3
02-06-2011 19:19:01.555 INFO TcpInputProc - port 9997 is reserved for splunk 2 splunk (SSL)
02-06-2011 19:19:01.555 INFO TcpInputProc - Port 9997 is compressed
02-06-2011 19:19:01.556 INFO TcpInputProc - Registering metrics callback for: tcpin_connections
```

On the forwarder, look for the following or similar messages at the start-up sequence to verify a successful connection:

```
TcpOutputProc - Retrieving configuration from properties
```
TcpOutputProc - Using SSL for server 10.1.12.112:9997, clientCert=/opt/splunk/etc/auth/server.pem

TcpOutputProc - ALL Connections will use SSL with sslCipher=

TcpOutputProc - initializing single connection with retry strategy for 10.1.12.112:9997

Below is how a successful connection might appear in splunkd.log on the indexer:

TcpInputProc - Connection in cooked mode from 10.1.12.111

TcpInputProc - Valid signature found

TcpInputProc - Connection accepted from 10.1.12.111

Below is how a successful connection might appear in splunkd.log on the forwarder:

TcpOutputProc - attempting to connect to 10.1.12.112:9997...

TcpOutputProc - Connected to 10.1.12.112:9997

You can also check metrics.log for something similar to the following:

```
index=_internal host=heavy hostname=universal | stats last(connectionType) as connectionType
```

For help troubleshooting your configuration issues, see Troubleshoot your forwarder to indexer configuration in this manual.

## Troubleshoot your forwarder to indexer authentication

1. Test your certificates:

   ```
   openssl s_client -connect {server}:{port}
   ```

   Port 8000, 8060, 8089, 9998, etc.

   A good certificate will return the following or something similar:

   ```
   Verify return code: 0 (ok)
   ```

2. Check `$SPLUNK_HOME/var/log/splunk/splunkd.log` (indexer and forwarder) for errors. On the indexer, check for the messages from the TCP input processor `TcpInputProc`. On the forwarder, check the messages from the TCP output processor `TcpOutputProc`.

3. Increase the logging level of the appropriate processors on the indexer and the forwarder in `$SPLUNK_HOME/etc/log.cfg`.


4. Restart Splunk Enterprise for these to take effect and observe the start-up sequence for the pertinent component. Most configuration issues are explicitly revealed by this method.

5. Check the SSL configuration using `btool` as follows:

   On the indexer:

   ```
   $SPLUNK_HOME/bin/splunk cmd btool inputs list --debug
   ```
On the forwarder:

$SPLUNK_HOME/bin/splunk cmd btool outputs list --debug

Common problems

- The path to the server certificate file set as the value of serverCert in inputs.conf is wrong, or the file cannot be read. This will generate the following error:

```
12-16-2010 16:07:30.965 ERROR SSLCommon - Can't read certificate file /opt/splunk/etc/auth/server.pem
errno=33558530 error:02001002:system library:fopen:No such file or directory
```

- The password to the RSA private key contained in the server certificate file is wrong.

```
12-07-2010 07:56:45.663 ERROR SSLCommon - Can't read key file /opt/splunk/etc/auth/server.pem
```

On *nix, you can manually test the password of the RSA key contained in the file with the command:

```
# openssl rsa -in /opt/splunk/etc/auth/server.pem -text
```

On Windows, you can manually test the password of the RSA key using the following command:

```
>openssl.exe rsa -in "c:\Program Files\Splunk\etc\auth\server.pem" -text
```
Secure distributed environments

About securing inter-Splunk communication

This chapter discusses the following “Splunk to Splunk” types of communications and how you can secure them:

- Securing distributed search heads and peers
- Securing deployment server and clients
- About securing clusters

Securing distributed search heads and peers

Distributed search configurations share search information, knowledge objects and app and configuration information over the management port.

Communication between search heads and peers relies on public-key encryption. Upon startup, Splunk software generates a private key and public key on your Splunk installation. When you configure distributed search on the search head, the public keys are distributed by search heads to peers and those keys are used to secure communication. This default configuration provides built-in encryption as well as data compression that improves performance.

It is possible to swap these generated keys out with your own keys, though the existing keys are generally considered adequate for most configurations.

To configure public-key encryption for distributed search setups, you create your keys and distribute them to your search heads and peers. To learn more about distributing key files to distributed search peers, look in the section on configuring distributed search in the Distributed Search manual: “Distribute the key files”.

Secure your deployment server and clients using certificate authentication

Authentication using signed certificates between deployment servers and clients is not recommended, because the configuration data pulled from the deployment server by the deployment client does not generally provide exploitable information. Configuring certificate authentication for a deployment server and clients impacts the rest of your configuration as follows:

- Splunk Web will fail to authenticate unless you also configure it to use the certificate.
- The CLI will be not be able to communicate with the deployment server.

You may find certificate authentication necessary in certain distributed configurations, perhaps where extremely sensitive server configuration data is sent to a variety of locations outside your firewall. You can manually configure each indexer to communicate with your Deployment Server:

**Note:** The deployment server cannot properly push certificates to peers. You must configure each member separately.

1. Create one or more certificates using the same root CA.
2. Distribute the certificates to your deployment server and clients.
3. Edit `server.conf` to provide the location of your certificates:

```bash
[sslConfig]
enableSplunkdSSL = true

sslVersions = Defaults to "*-ssl2" (anything newer than SSLv2). This is the recommended setting.

serverCert = The full path to the PEM format server certificate file. Default certificates
($SPLUNK_HOME/etc/auth/server.pem) are generated by Splunk at start. To secure Splunk,
you should replace the default cert with your own PEM file.

sslPassword = password

sslRootCAPath = absolute path to the operating system's root CA (Certificate Authority) PEM
format file containing one or more root CA. Do not configure this attribute on Windows.
```

4. Edit `server.conf` to authenticate against your certificates by adding the following attribute to the `[sslConfig]` stanza in previous step:

```bash
requireClientCert = true
```

**Important:** This `requireClientCert` is set to "false" by default. If you change it to true to force Splunk to check your client's certificates, Splunk Web and the CLI will also be checked for certificates. Your CLI connection will no longer work because your CLI is unable to present a certificate as a client.

5. Edit `web.conf` to present a certificate signed by the same root CA so that Splunk Web can connect to the server.

The following is an example of an edited settings stanza:

```bash
[settings]
enableSplunkWebSSL = true

privKeyPath = etc/auth/splunkweb/mySplunkWebPrivateKey.key

serverCert = etc/auth/splunkweb/mySplunkWebCertificate.pem

cipherSuite = <your chosen cipher suite (optional)>
```

**Note:** Splunk Web does not support passwords, so you must remove the password from the private key. For more information, see "Get certificates signed by a third party for Splunk Web."

### Secure your clusters with pass4SymmKey

Splunk provides a security key to let your search head or indexer clustering nodes authenticate with each other. When you set up an indexer cluster or search head cluster, you assign the same key to each node in the cluster. You use the `pass4SymmKey` setting in the `server.conf` file. You can set the key through Splunk Web, the configuration file, or the CLI.

`pass4SymmKey` controls authentication between Splunk platform instances. It does not manage user access, nor does it negate the need to use valid SSL/TLS certificates to secure the instances.

#### Configure pass4SymmKey for search head clustering

Configure `pass4SymmKey` when you deploy the search head cluster. See Deploy a search head cluster.

For details on configuring `pass4SymmKey` on a search head cluster, including how to set it post-deployment, see Set a security key for the search head cluster.
Configure pass4SymmKey for indexer clustering

Configure pass4SymmKey when you deploy the indexer cluster, while enabling the master node. See Enable the indexer cluster master node.

For more details on setting pass4SymmKey on an indexer cluster, see Configure the security key.

How apps encrypt pass4SymmKey

When you specify pass4SymmKey in clear-text for an app directory on a Splunk instance (for example: etc/apps/myapp/default/server.conf), the software writes an obfuscated version of the key to the local file (in this example, system/local/server.conf) when you restart the instance. Configuration files in the default directory are generally read-only, and the software writes the information to the local file, which is editable.

Placing a password directly into an app's local directory (for example: etc/apps/myapp/local/server.conf), replaces it with the encrypted version.

When the configuration is listed using curl or a splunkd endpoint, the pass4SymmKey appears encrypted. If the configuration location is read-only, Splunk software likewise writes to local.

Use OpenSSL to generate a random passphrase for pass4SymmKey

You can use the OpenSSL utilities that come with Splunk software to generate a passphrase that you can use with pass4SymmKey.

For the strongest security, select a passphrase that is at least 12 characters long and checks out against a dictionary of known bad passphrases, like abc123, password, qwerty, admin, and so on. The OpenSSL utility that comes with Splunk software lets you randomly generate a passphrase that you can then use to set pass4SymmKey with on all nodes of your Splunk deployment.

1. On a Splunk instance, open a shell prompt.
2. Change to the $SPLUNK_HOME/bin directory.
3. Run the following command to generate a random 12-character passphrase:

   splunk cmd openssl rand -base64 9

4. Copy the output of the command to your clipboard.
5. For all machines that you want to use the new passphrase:
   1. Edit $SPLUNK_HOME/etc/system/local/server.conf.
   2. Set pass4SymmKey = <new passphrase that you just generated>
   3. Save $SPLUNK_HOME/etc/system/local/server.conf.
   4. Restart Splunk software.
Audit Splunk Enterprise activity

Use Splunk Enterprise to audit your system activity

Knowing what is happening in your system is vital to keeping it secure. To make the most of your system and keep it secure, we recommend the following best practices:

• Perform a periodic review of Splunk access and audit logs.
• Perform a periodic review of Splunk server audit and security logs.
• Perform a periodic review of all Splunk users and roles.

Audit Splunk activity

With auditing enabled, Splunk logs distinct events to the audit index (index=_audit). Interactions with Splunk such as searches and configuration changes generate audit events.

What's in an audit event?

• Timestamp:
  ♦ date and time of the event.
• User information:
  ♦ the user who generated the event.
  ♦ If the event contains no user information, Splunk sets the user to whoever is currently logged in.
• Additional information:
  ♦ available event details -- what file, success/denial, etc.

Activities that generate audit events

Audit events are generated from:

• all files in Splunk's configuration directory $SPLUNK_HOME/etc/*
  ♦ files are monitored for add/change/delete using the file system change monitor.
• system start and stop.
• users logging in and out.
• adding / removing a new user.
• changing a user's information (password, role, etc).
• execution of any capability in the system.
  ♦ capabilities are listed in authorize.conf

Audit event storage

Splunk stores audit events locally in the audit index (index=_audit). Audit events are logged in the log file:
$SPLUNK_HOME/var/log/splunk/audit.log.

If you have configured Splunk as a forwarder in a distributed setting, audit events are forwarded like any other event.
Use audit events to secure Splunk Enterprise

Use Splunk to search your audit log to review and alert on administrative access:

1. Audit user access

   index=_audit action=log* action="login attempt"

2. Find the location from which users are accessing Splunk:

   index=_internal | eval timestamp=strftime(_time, "%Y-%m-%d %H:%M:%S.%Q") | table timestamp, user, clientip

3. Consider setting up a real-time alert on administrative user access:

   (index=_audit action=log* action="login attempt") OR (index=_internal) user=admin

   See the Alerting Manual for more information about creating alerts.

4. You can also create dashboards that collect and display your chosen searches, see Build Dashboards in Splunk Web in the Dashboards and Visualizations Manual.

Manage data integrity

The Splunk Enterprise data integrity control feature provides a way to verify the integrity of data that is indexed.

When you enable data integrity control for an index, Splunk Enterprise computes hashes (using SHA 256) on every slice of data and stores those hashes so that you can go back later and verify the integrity of your data.

How it works

When you enable data integrity control, Splunk Enterprise computes hashes on every slice of newly indexed raw data and writes it to a 1lHashes file. When the bucket rolls from hot to warm, Splunk Enterprise computes a hash on the contents of the 1lHashes and stores the computed hash in 12Hash. Both hash files are stored in the rawdata directory for that bucket.

Note that data integrity control hashes newly indexed data, data coming from a forwarder should be secured and encrypted with SSL. For more information, see About securing Splunk with SSL.

Check your hashes to validate your data

To check Splunk Enterprise data, run the following CLI command to verify the integrity of an index or bucket:

```
./splunk check-integrity -bucketPath [ bucket path ] [ -verbose ]
./splunk check-integrity -index [ index name ] [ -verbose ]
```

Configure data integrity control

To configure Data Integrity Control, edit indexes.conf to enable the enableDataIntegrityControl attribute for each index. The default value for all indexes is false (off).

```
enableDataIntegrityControl=true
```
Data Integrity in clustered environments

In a clustered environment, the cluster master and all the peers must run Splunk Enterprise 6.3 or later to enable accurate index replication.

Optionally modify the size of your data slice

By default, data slices are set to 128kb, which means that a data slice is created and hashed every 128KB. You can optionally edit `indexes.conf` to specify the size of each slice.

```
rawChunkSizeBytes = 131072
```

Store and secure your data hashes

For optimal security, you can optionally store your hashes outside the system where the data is hosted, such as a different server. To avoid naming conflicts, store your secured hashes in separate directories.

Regenerate hashes

If you lose your hashes for a bucket, Use the following CLI command to re-generate hash files on a bucket or index. This command extracts the hashes embedded in the journal:

```
./splunk generate-hash-files -bucketPath [ bucket path ] [ verbose ]
./splunk generate-hash-files -index [ index name ] [ verbose ]
```
Best practices for Splunk Enterprise security

Safeguards for risky commands

Splunk Enterprise contains built-in search processing language (SPL) safeguards to warn you when you are about to unknowingly run a search that contains commands that might be a security risk. This warning appears when you click a link or type a URL that loads a search that contains risky commands.

The warning does not appear when you create ad hoc searches.

This warning alerts you to the possibility of unauthorized actions by a malicious user. Unauthorized actions include:

- Copying or transferring data (data exfiltration)
- Deleting data
- Overwriting data

A possible scenario when this might occur is when a malicious person creates a search that includes commands that exfiltrate or damage data. The malicious person then sends an unsuspecting user a link to the search. The URL contains a query string (q) and a search identifier (sid), but the sid is expired. The malicious person hopes the user will use the link and the search will run.

Commands that trigger the warning

The commands that trigger this warning are listed here:

- collect
- crawl
- dump
- delete
- input
- outputcsv
- outputlookup
- runshellscript
- script
- sendalert
- sendemail
- tscollect

Actions in the warning dialog box

Instead of running the search immediately, Splunk Enterprise analyzes the search for risky commands. If one or more risky commands are identified, a warning dialog box appears. You have the option to cancel, run, or investigate the search.

Cancel

Closes the warning dialog box. The search does not run and the search is removed from the Search bar. Closing the dialog box, by clicking the Close button (X), is the same as clicking Cancel.

Run

Runs the search.

Investigate
Displays the search in the Search bar so that you can review the SPL. Use this option to copy the syntax of the search. Send a copy of the search, along with any information about the source of the link, to your system administrator.

Turning off the warning

Only users with Write permission can edit the web.conf file to turn off the warning dialog box.

You can turn off the warning for a specific command, or for all of the risky commands.

**Turn off the warning for a specific command**

1. Copy the commands.conf file, which is located in the $SPLUNK_HOME/etc/system/default directory.
2. Paste the copy of the file in the $SPLUNK_HOME/etc/system/local directory.
3. Locate the command and change the setting from is_risky = true to is_risky = false.
4. Restart Splunk Enterprise.

**Turn off the warning for all of the commands**

1. Open the web.conf file. This file is located in the $SPLUNK_HOME/etc/system/default/ directory.
2. Change the enable_risky_command_check parameter to false.
3. Restart Splunk Enterprise.

See also

In the Admin Manual:

- About configuration files
- commands.conf file
- web.conf file

Splunk server tokens

If a forwarder TCP token is corrupt or rejected, the indexer that receives the token generates error messages in its logs. If you do not locate the bad token, that forwarder tries to use it indefinitely.

To locate the bad forwarder token, increase the logging level of the indexer:

1. Open a shell prompt.
2. Using a text editor, edit SPLUNK_HOME/etc/log.cfg as follows:
   
   ```
   category.TcpOutputProc=DEBUG
   category.TcpInputConfig=DEBUG
   category.TcpInputProc=DEBUG
   ```
3. Save the file and close it.
4. Restart the indexer.

When a token that a forwarder sends matches the token that the indexer receives, the following messages are generated:

Indexer:
Avoid malicious CSV files in searches

If you export your search results as a CSV and then open it in Excel/OpenOffice, any fields that start with an '=' character will be executed.

For example:

1. User runs stats count | eval trick="=1+1".
2. User exports the results as a CSV file.
3. User load the new CSV file in Excel.
4. Field in Excel has a value of 2 when it should have a value of '='1+1.

To avoid this, you can do one of the following:

- For any cell that starts with the following characters, add a space to the beginning and remove any tab characters (0x09) in the cell.

  ♦ =
  ♦ -
  ♦ "
  ♦ @
  ♦ +

- Append any cell beginning with the previously listed characters with an apostrophe (').

- Make sure users do not have the "export_results_is_visible" capability (version 6.4 and later only). This capability displays the export results button, without this capability it is not possible to generate CSV files at all.
Appendix A: How to get SSL certificates

How to self-sign certificates

This topic describes one way you can use OpenSSL to self-sign certificates for securing forwarder-to-indexer and Inter-Splunk communication.

If you already possess or know how to generate the needed certificates, you can skip this topic and go directly to the configuration steps, described later in this manual:

- How to prepare your signed certificates for Splunk
- Configure Splunk forwarding to use your own certificates
- About securing inter-Splunk communication

Self-signed certificates are best for data communication that occurs within an organization or between known entities. If you communicate with unknown entities, we recommend CA-signed certificates to secure your data.

Before you begin

In this discussion, $SPLUNK_HOME refers to the Splunk Enterprise installation directory:

- For Windows, Splunk software is installed in C:\Program Files\splunk by default
- For most Unix platforms, the default installation directory is at /opt/splunk
- For Mac OS, it is /Applications/splunk

See the Administration Guide to learn more about working with Windows and *nix.

Create a new directory for your certificates

Create a new directory to work from when creating your certificates. In our example, we are using $SPLUNK_HOME/etc/auth/mycerts:

```bash
# mkdir $SPLUNK_HOME/etc/auth/mycerts
# cd $SPLUNK_HOME/etc/auth/mycerts
```

This ensures you do not overwrite the Splunk-provided certificates that reside in $SPLUNK_HOME/etc/auth.

Create the root certificate

First you create a root certificate that serves as your root certificate authority. You use this root CA to sign the server certificates that you generate and distribute to your Splunk instances.

### Generate a private key for your root certificate

1. Create a key to sign your certificates.

   In *nix:
   ```bash
   $SPLUNK_HOME/bin/splunk cmd openssl genrsa -aes256 -out myCA_privateKey.key 2048
   ```

   In Windows:
   ```bash
   ```

   1. Create a key to sign your certificates.

      In *nix:
      ```bash
      $SPLUNK_HOME/bin/splunk cmd openssl genrsa -aes256 -out myCA_privateKey.key 2048
      ```

      In Windows:
      ```bash
      ```
When the step is completed, the private key `myCAPrivateKey.key` appears in your directory.

**Generate and sign the certificate**

1. Generate a new Certificate Signing Request (CSR):
   
   In *nix:
   
   ```sh
   $SPLUNK_HOME/bin/splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
   
   In Windows:
   
   $SPLUNK_HOME\bin\splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
   ```

   2. When prompted, enter the password you created for the private key in
      `$SPLUNK_HOME/etc/auth/mycerts/myCAPrivateKey.key`.

   3. Provide the requested certificate information, including the common name if you plan to use common name checking in your configuration.

   A new CSR `myCACertificate.csr` appears in your directory.

   4. Use the CSR `myCACertificate.csr` to generate the public certificate:
   
   In *nix:
   
   ```sh
   $SPLUNK_HOME/bin/splunk cmd openssl x509 -req -in myCACertificate.csr -sha512 -signkey myCAPrivateKey.key -CAcreateserial -out myCACertificate.pem -days 1095
   
   In Windows:
   
   $SPLUNK_HOME\bin\splunk cmd openssl x509 -req -in myCACertificate.csr -sha512 -signkey myCAPrivateKey.key -CAcreateserial -out myCACertificate.pem -days 1095
   ```

   5. When prompted, enter the password for the private key `myCAPrivateKey.key`.

   A new file `myCACertificate.pem` appears in your directory. This is the public CA certificate that you will distribute to your Splunk instances.

**Create the server certificate**

Now that you have created a root certificate to serve as your CA, you must create and sign your server certificate.

**A note about common name checking**

This topic shows you how to create a new private key and server certificate.

You can distribute this server certificate to all forwarders, indexers as well your Splunk instances that communicate on the management port. If you plan to use a different common name for each instance, you simply repeat the process described here to create different certificates (each with a different common name) for your Splunk instances.

For example, if configuring multiple forwarders, you can use the following example to create the certificate `myServerCertificate.pem` for your indexer, then create another certificate `myForwarderCertificate.pem` using the same root CA and install that certificate on your forwarder. Note that an indexer will only accept a properly generated and...
configured certificate from a forwarder that is signed by the same root CA.

See Configure Splunk forwarding to use your own certificates for more information about configuring your forwarders and indexers.

**Generate a key for your server certificate**

1. Generate a new RSA private key for your server certificate. In this example we are again using AES encryption and a 2048 bit key length:

   In *nix:
   ```
   $SPLUNK_HOME/bin/splunk cmd openssl genrsa -aes256 -out myServerPrivateKey.key 2048
   ```
   In Windows:
   ```
   $SPLUNK_HOME\bin\splunk cmd openssl genrsa -aes256 -out myServerPrivateKey.key 2048
   ```

2. When prompted, create a new password for your key.

A new key `myServerPrivateKey.key` is created. You will use this key to encrypt the outgoing data on any Splunk Software instance where you install it as part of the server certificate.

**Generate and sign a new server certificate**

1. Use your new server private key `myServerPrivateKey.key` to generate a CSR for your server certificate.

   In *nix:
   ```
   $SPLUNK_HOME/bin/splunk cmd openssl req -new -key myServerPrivateKey.key -out myServerCertificate.csr
   ```
   In Windows:
   ```
   $SPLUNK_HOME\bin\splunk cmd openssl req -new -key myServerPrivateKey.key -out myServerCertificate.csr
   ```

2. When prompted, provide the password to the private key `myServerPrivateKey.key`.

3. Provide the requested information for your certificate, including a Common Name if you plan to configure Splunk Software to authenticate via common-name checking.

   A new CSR `myServerCertificate.csr` appears in your directory.

4. Use the CSR `myServerCertificate.csr` and your CA certificate and private key to generate a server certificate.

   In *nix:
   ```
   $SPLUNK_HOME/bin/splunk cmd openssl x509 -req -in myServerCertificate.csr -SHA256 -CA myCACertificate.pem -CAkey myCAPrivateKey.key -CAcreateserial -out myServerCertificate.pem -days 1095
   ```
   In Windows:
   ```
   $SPLUNK_HOME\bin\splunk cmd openssl x509 -req -in myServerCertificate.csr -SHA256 -CA myCACertificate.pem -CAkey myCAPrivateKey.key -CAcreateserial -out myServerCertificate.pem -days 1095
   ```

5. When prompted, provide the password for the certificate authority private key `myCAPrivateKey.key`. Make sure to sign this with your private key and not the server key you just created.
A new public server certificate `myServerCertificate.pem` appears in your directory.

**Next steps**

You should now have the following files in the directory you created, which is everything you need to configure indexers, forwarders, and Splunk instances that communicate over the management port:

- `myServerCertificate.pem`
- `myServerPrivateKey.key`
- `myCACertificate.pem`

Now that you have the certificates you need, prepare your server certificate (including appending any intermediate certificates), and then configure Splunk to find and use them:

- See How to prepare your signed certificates for Splunk to learn how to set up your certificates to work with Splunk.
- See Configure Splunk forwarding to use your own certificates to learn more about configuring certificate authentication for forwarding.
- See About securing inter-Splunk communication to learn more about configuring certificate authentication for Splunk to Splunk communications.

**How to get certificates signed by a third-party**

This topic describes one way you can use the version of OpenSSL that ships with Splunk Enterprise to obtain third-party certificates that you can use to secure your forwarder-to-indexer and inter-Splunk communication.

To get certificates that you can use to secure for browser-to-Splunk Web communication, see Get certificates signed by a third-party for Splunk Web.

If you already possess or know how to generate the certificates you can, skip this topic and go directly to the configuration steps, which are described later in this manual:

- Configure Splunk forwarding to use your own certificates
- About securing inter-Splunk communication

**Note:** If you plan to use multiple common names in your configurations, you can repeat the steps described here to create a different server certificate using the same root CA for each instance with it’s own common name and then configure your Splunk instances to use them. See Configure Splunk forwarding to use your own certificates for more information about configuring your forwarders and indexers.

**Before you begin**

In this discussion, `$SPLUNK_HOME` refers to the Splunk Enterprise installation directory. We recommend that you follow this convention, but if you do not, you should replace `$SPLUNK_HOME` with your installation directory when using these examples.

For Windows, you might need to set this variable at the command line or in the Environment tab in the System Properties dialog.

Default home directories depend on your platform:
• For Windows, the Splunk Enterprise directory is at `C:\Program Files\Splunk` by default.
• For most *nix platforms, the default installation directory is at `/opt/splunk`.
• For Mac OS, it is `/Applications/splunk`.

See the Administration Guide to learn more about working with Windows and *nix.

Create a new directory for your certificates

Create a new directory for your new certificates. In our example, we are using `$SPLUNK_HOME/etc/auth/mycerts`:

```bash
# mkdir $SPLUNK_HOME/etc/auth/mycerts
# cd $SPLUNK_HOME/etc/auth/mycerts
```

When you make a new folder you protect the existing certificates and keys in `$SPLUNK_HOME/etc/auth`. Working in a new directory protects the default certificates and lets you use them for other Splunk Software components as necessary.

Request your server certificate

Create and sign a Certificate Signing Request (CSR) to send to your Certificate Authority.

**Important:** This example shows you how to create a new private key and request a server certificate. You can distribute this server certificate to all forwarders, indexers as well your Splunk instances that communicate on the management port. If you want to use a different common names for each instance, you simply repeat the process described here to create different certificates (each with a different common name) for your Splunk instances.

For example, when configuring multiple forwarders, you can use the following example to create the certificate `myServerCertificate.pem` for your indexer, then create another certificate `myForwarderCertificate.pem` using the same root CA and install that certificate on your forwarder. An indexer will only accept a properly generated and configured certificate from a forwarder that is signed by the same root CA.

See Configure Splunk forwarding to use your own certificates for more information about configuring your forwarders and indexers.

*Generate a private key for your server certificate*

1. Create a new private key. The following example uses DES3 encryption and a 2048 bit key length. We recommend a key length of 2048 or higher.

   In *nix:
   ```bash
   $SPLUNK_HOME/bin/splunk cmd openssl genrsa -des3 -out myServerPrivateKey.key 2048
   ```

   In Windows:
   ```bash
   $SPLUNK_HOME\bin\splunk cmd openssl genrsa -des3 -out myServerPrivateKey.key 2048
   ```

2. When prompted, create a password for your key.

When you are done, a new private key `myServerPrivateKey.key` is created in your directory. You will use this key to sign your Certificate Signing Request (CSR).

*Generate a new Certificate Signing Request (CSR)*

1. Use your private key `myServerPrivateKey.key` to generate a CSR for your server certificate:
In *nix:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl req -new -key myServerPrivateKey.key -out myServerCertificate.csr
```

In Windows:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl req -new -key myServerPrivateKey.key -out myServerCertificate.csr
```

2. When prompted, provide the password you created for your private key myServerPrivateKey.key.

3. Provide the requested information for your certificate. To use common-name checking, make sure to provide a Common Name when entering your certificate details.

When you are done, a new CSR myServerCertificate.csr appears in your directory.

### Download and verify the server certificate and public key

1. Send your CSR to your Certificate Authority (CA) to request a new server certificate. The request process varies based on the Certificate Authority you use.

2. Download the new server certificate from your Certificate Authority. For the examples in this manual, let's call this myServerCertificate.pem.

3. Also download your Certificate Authority's public CA certificate. For the examples in this manual, let's call this myCACertificate.pem.

If your Certificate Authority does not provide you with certificates in PEM format, you must convert them using the OpenSSL command appropriate to your existing file type, consult your OpenSSL documentation for more information about converting different file types.

4. View the contents to make sure it has everything you need:

   - The "Issuer" entry should refer to your CA's information.
   - The "Subject" entry should show the information (country name, organization name, Common Name, etc) that you entered when creating the CSR earlier.

**Note:** For *nix, you can view the contents your certificate using the following command:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl x509 -in myServerCertificate.pem -text
```

### Next steps

You should now have the following files in the directory you created, which is everything you need to configure indexers, forwarders, and Splunk instances that communicate over the management port:

- myServerCertificate.pem
- myServerPrivateKey.key
- myCACertificate.pem

Now that you have the certificates you need, you must prepare your server certificate (including appending any intermediate certificates), and then configure Splunk software to find and use your certificates:

- See "How to prepare your signed certificates for Splunk" to learn how to set up your certificates to work with Splunk.
• See "Configure Splunk forwarding to use your own certificates" to learn more about configuring certificate authentication for forwarding.
• See "About securing inter-Splunk communication" to learn more about configuring certificate authentication for inter-Splunk communications.

Self-sign certificates for Splunk Web

This topic provides basic examples for creating the self-signed certificates in the command line using the version of OpenSSL included with Splunk software.

There are multiple ways you can create signed certificates, depending upon your organizations policies, your platform, and the tools that you are using. If you have already generated these certificates and key, or if you are experienced in generating certificates, you can skip this task and go directly to the configuration topic Secure Splunk Web with your own certificate in this manual.

Since self-signed certificates are signed by your organization, they are not contained in browser certificate stores. As a result, web browsers consider self-signed certificates "untrusted". This produces a warning page to users and may even prevent access for the user.

Self-signed certificates are best for browser to Splunk Web communication that happens within an organization or between known entities where you can add your own CA to all browser stores that will contact Splunk Web. For any other scenario, CA-signed certificates are recommended. See Get certificates signed by a third party for Splunk Web for more information.

Before you begin

In this discussion, $SPLUNK_HOME refers to the Splunk installation directory.

• For Windows, the default installation directory is C:\Program Files\splunk.
• For most *nix platforms, the default installation directory is /opt/splunk.
• For Mac OS, the default installation directory is /Applications/splunk.

See the Administration Guide to learn more about working with Windows and *nix.

Generate a new root certificate to be your Certificate Authority

1. Create a new directory to host your certificates and keys. For this example we will use $SPLUNK_HOME/etc/auth/mycerts.

We recommend that you place your new certificates in a different directory than $SPLUNK_HOME/etc/auth/splunkweb so that you don’t overwrite the existing certificates. This ensures that you are able to use the certificates that ship with Splunk software in $SPLUNK_HOME/etc/auth/splunkweb for other Splunk components as necessary.

Note: If you created a self-signed certificate as described in How to self-sign certificates, you can copy that root certificate into your directory and skip to the next step: Create a new private key for Splunk Web.

2. Generate a new RSA private key. Splunk Web supports 2048 bit keys, but you can specify larger keys if they are supported by your browser.

$SPLUNK_HOME/bin/splunk cmd openssl genrsa -aes256 -out myCAPrivateKey.key 2048
Note that in Windows you may need to append the location of the `openssl.cnf` file:

```
$SPLUNK_HOME\bin\splunk cmd openssl genrsa -aes256 -out myCAPrivateKey.key 2048
```
Splunk Web supports 2048 bit keys, but you can specify larger keys if they are supported by your browser.

3. When prompted, create a password.

The private key `myCAPrivateKey.key` appears in your directory. This is your root certificate private key.

4. Generate a certificate signing request using the root certificate private key `myCAPrivateKey.key`:

In *nix:

```
$SPLUNK_HOME/bin/splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
```

In Windows:

```
$SPLUNK_HOME\bin\splunk cmd openssl req -new -key myCAPrivateKey.key -out myCACertificate.csr
```

5. Provide the password to the private key `myCAPrivateKey.key`.

A new CSR `myCACertificate.csr` appears in your directory.

6. Use the CSR to generate a new root certificate and sign it with your private key:

In *nix:

```
$SPLUNK_HOME/bin/splunk cmd openssl x509 -req -in myCACertificate.csr 
-signtel myCAPrivateKey.key -out myCACertificate.pem -days 3650
```

In Windows:

```
>$SPLUNK_HOME/bin\splunk cmd openssl x509 -req -in myCACertificate.csr
-signtel myCAPrivateKey.key -out myCACertificate.pem -days 3650
```

7. When prompted, provide for the password to the private key `myCAPrivateKey.key`.

A new certificate `myCACertificate.pem` appears in your directory. This is your public certificate.

### Create a new private key for Splunk Web

1. Generate a new private key:

In *nix:

```
$SPLUNK_HOME/bin/splunk cmd openssl genrsa -aes256 -out mySplunkWebPrivateKey.key 2048
```

In Windows:

```
$SPLUNK_HOME\bin\splunk cmd openssl genrsa -aes256 -out mySplunkWebPrivateKey.key 2048
```

2. When prompted, create a password.

A new key, `mySplunkWebPrivateKey.key` appears in your directory.

3. Remove the password from your key. (Splunk Web does not support password-protected private keys.)
In *nix:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl rsa -in mySplunkWebPrivateKey.key
   -out mySplunkWebPrivateKey.key
```

In Windows:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl rsa -in mySplunkWebPrivateKey.key
   -out mySplunkWebPrivateKey.key
```

You can verify that your password was removed with the following command:

In *nix:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -text
```

In Windows:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -text
```

You should be able to read the contents of your certificate without providing a password.

Create and sign a server certificate

1. Create a new certificate signature request using your private key `mySplunkWebPrivateKey.key`:

In *nix:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl req -new -key mySplunkWebPrivateKey.key
   -out mySplunkWebCert.csr
```

In Windows:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl req -new -key mySplunkWebPrivateKey.key
   -out mySplunkWebCert.csr
```

The CSR `mySplunkWebCert.csr` appears in your directory.

2. Self-sign the CSR with the root certificate private key `myCAPrivateKey.key`:

In *nix:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl x509 -req -in mySplunkWebCert.csr -CA myCACertificate.pem
   -CAkey myCAPrivateKey.key -CAcreateserial -out mySplunkWebCert.pem -days 1095
```

In Windows:

```bash
$SPLUNK_HOME\bin\splunk cmd openssl x509 -req -in mySplunkWebCert.csr -CA myCACertificate.pem
   -CAkey myCAPrivateKey.key -CAcreateserial -out mySplunkWebCert.pem -days 1095
```

3. When prompted, provide the password to the root certificate private key `myCAPrivateKey.key`:

The certificate `mySplunkWebCert.pem` is added to your directory. This is your server certificate.

Create a single PEM file

Combine your server certificate and public certificates, in that order, into a single PEM file.
Here’s an example of how to do this in Linux:

```bash
# cat mySplunkWebCert.pem myCACertificate.pem > mySplunkWebCertificate.pem
```

Here’s an example in Windows:

```bash
# type mySplunkWebCert.pem myCACertificate.pem > mySplunkWebCertificate.pem
```

**Set up certificate chains**

To use multiple certificates, append the intermediate certificate to the end of the server’s certificate file in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

**Next steps**

Now that you have your certificates, you need to distribute them and configure Splunkd and Splunk Web to use them. See Secure Splunk Web with your own certificate in this manual for more information.

**Get certificates signed by a third-party for Splunk Web**

This topic provides basic examples for creating the third-party signed certificates necessary to configure Splunk Web for SSL authentication and encryption.

There are multiple ways you can create these certificates, depending upon your organization's policies, your network structure and the tools that you are using. If you have already generated these certificates and key, or if you are experienced with third-party certificates, you may prefer to skip this step and go directly to the configuration topic in this manual at Secure Splunk Web with your own certificate.

**Before you begin**

In this discussion, $SPLUNK_HOME refers to the Splunk installation directory. On Windows, Splunk software is installed at C:\Program Files\splunk by default. For most Unix platforms, the default installation directory is at /opt/splunk; for Mac OS, it is /Applications/splunk. See the Administration Guide to learn more about working with Windows and *nix.

**Create a new private key for Splunk Web**

1. Create a new directory to host your own certificates and keys. In this example we will use $SPLUNK_HOME/etc/auth/mycerts.
We recommend that you place your new certificates in a different directory than \$SPLUNK_HOME/etc/auth/splunkweb so that you don't overwrite the existing certificates. This ensures that you can use the certificates that ship with Splunk for other Splunk components as necessary.

2. Generate a new private key. Splunk Web supports 2048-bit keys or larger.

   Linux:

   \$SPLUNK_HOME/bin/splunk cmd openssl genrsa -des3 -out mySplunkWebPrivateKey.key 2048

   Windows:

   \$SPLUNK_HOME\bin\splunk cmd openssl genrsa -des3 -out mySplunkWebPrivateKey.key 2048

3. Create a password when prompted to enter the passphrase for the original key.

   A new private key mySplunkWebPrivateKey.key is added to your directory. You can use this key to sign your CSR.

4. Remove the password from the private key. Splunk Web does not support private key passwords.

   Linux:

   \$SPLUNK_HOME/bin/splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -out mySplunkWebPrivateKey.key

   Windows:

   \$SPLUNK_HOME\bin\splunk cmd openssl rsa -in mySplunkWebPrivateKey.key -out mySplunkWebPrivateKey.key

   You can use to following command to make sure that your password was successfully removed:

   # openssl rsa -in mySplunkWebPrivateKey.key -text

   If the password was successfully removed, you can view the certificate contents without providing a password.

Create a Certificate Authority (CA) request and obtain your server certificate

1. Create a new certificate signature request using your private key mySplunkWebPrivateKey.key:

   In *nix:

   \$SPLUNK_HOME/bin/splunk cmd openssl req -new -key mySplunkWebPrivateKey.key -out mySplunkWebCert.csr

   In Windows:

   \$SPLUNK_HOME\bin\splunk cmd openssl req -new -key mySplunkWebPrivateKey.key -out mySplunkWebCert.csr

   Note for Windows platforms: If you see an error similar to this:

   Unable to load config info from c:\build-amd64-5.0.2-20130120-1800\splunk/ssl/openssl.cnf
   Try typing the following in your command prompt then run the openssl command again:

   set OPENSSL_CONF=c:/Program Files/Splunk/openssl.cnf

2. Use the CSR mySplunkWebCert.csr to request a new signed certificate from your Certificate Authority (CA). The process for requesting a signed certificate varies depending on how your Certificate Authority handles a certificate signature request. Contact your CA for more information.
3. Download the server certificate returned by your Certificate Authority. For this example, let's call it "mySplunkWebCert.pem."

4. Download your Certificate Authority's public CA certificate. For this example, let's call it "myCAcert.pem."

5. Make sure that both the server certificate and the public CA certificate are both in PEM format. If the certificates are not in PEM format, convert them using the `openssl` command appropriate to your existing file type. Here's an example of a command that you can use for DER formats:

```bash
$SPLUNK_HOME/bin/splunk cmd openssl x509 -in mySplunkWebCert.crt -inform DER -out mySplunkWebCert.pem -outform PEM
$SPLUNK_HOME\bin\splunk cmd openssl x509 -in myCACert.crt -inform DER -out myCACert.pem -outform PEM
```

6. Check both certificates to make sure they have the necessary information and are not password protected.

```bash
$SPLUNK_HOME/bin/splunk cmd openssl x509 -in myCACert.pem -text
$SPLUNK_HOME/bin/splunk cmd openssl x509 -in mySplunkWebCert.pem -text

$SPLUNK_HOME/bin/splunk cmd openssl x509 -in myCACert.pem -text
$SPLUNK_HOME\bin\splunk cmd openssl x509 -in mySplunkWebCert.pem -text
```

The issuer information for `mySplunkWebCert.pem` should be the subject information for `myCACert.pem` (unless you are using intermediary certificates).

**Combine your certificate and keys into a single file**

Combine your server certificate and public certificate, in that order, into a single PEM file.

**Set up certificate chains**

To use multiple certificates, append the intermediate certificate to the end of the server's certificate file in the following order:

```
[ server certificate]
[ intermediate certificate]
[ root certificate (if required) ]
```

So for example, a certificate chain might look like this:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the intermediate certificate)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA)...
-----END CERTIFICATE-----
```

Note that the root CA that signed the intermediate certificate and all intermediary certificates must be in the browser certificate stores.

**Next steps**

Configure Splunk's `web.conf` file to find and use your certificates for authentication. See Secure Splunk Web with your own certificate for more information.