



# Get Started with TLS/SSL Rules

The following topics provide an overview of creating, configuring, managing, and troubleshooting TLS/SSL rules:



**Note** Because TLS and SSL are often used interchangeably, we use the expression *TLS/SSL* to indicate that either protocol is being discussed. The SSL protocol has been deprecated by the IETF in favor of the more secure TLS protocol, so you can usually interpret *TLS/SSL* as referring to TLS only.

The exception is SSL policies. Because the FMC configuration option is **Policies > Access Control > SSL**, we use the term *SSL policies* although these policies are used to define rules for TLS and SSL traffic.

For more information about SSL and TLS protocols, see a resource such as [SSL vs. TLS - What's the Difference?](#).

- [TLS/SSL Rules Overview, on page 1](#)
- [TLS/SSL Rule Guidelines and Limitations, on page 1](#)
- [Requirements and Prerequisites for TLS/SSL Rules, on page 8](#)
- [Encrypted Traffic Inspection Configuration, on page 9](#)
- [Creating and Modifying TLS/SSL Rules, on page 10](#)
- [TLS/SSL Rule Conditions, on page 13](#)
- [TLS/SSL Rule Actions, on page 15](#)
- [TLS/SSL Rules Management, on page 19](#)

## TLS/SSL Rules Overview

*TLS/SSL rules* provide a granular method of handling encrypted traffic across multiple managed devices, whether blocking the traffic without further inspection, not decrypting the traffic and inspecting it with access control, or decrypting the traffic for access control analysis.

## TLS/SSL Rule Guidelines and Limitations

Keep the following points in mind when setting up your TLS/SSL rules. Properly configuring TLS/SSL rules is a complex task, but one that is essential to building an effective deployment that handles encrypted traffic. Many factors influence how you configure rules, including certain application behavior that you cannot control.

In addition, rules can preempt each other, require additional licenses, or contain invalid configurations. Thoughtfully configured rules can also reduce the resources required to process network traffic. Creating overly complex rules and ordering rules the wrong way can adversely affect performance.

For detailed information, see [Best Practices for Access Control Rules](#).

### Related Topics

- [Rule and Other Policy Warnings](#)
- [Best Practices for Access Control Rules](#)
- [Guideline for Using TLS/SSL Decryption, on page 2](#)
- [TLS/SSL Rule Unsupported Features, on page 2](#)
- [TLS/SSL Do Not Decrypt Guidelines, on page 3](#)
- [TLS/SSL Decrypt - Resign Guidelines, on page 3](#)
- [TLS/SSL Decrypt - Known Key Guidelines, on page 5](#)
- [TLS/SSL Block Guidelines, on page 6](#)
- [TLS/SSL Certificate Pinning Guidelines, on page 7](#)
- [TLS/SSL Heartbeat Guidelines, on page 7](#)
- [TLS/SSL Anonymous Cipher Suite Limitation, on page 7](#)
- [TLS/SSL Normalizer Guidelines, on page 7](#)
- [Other TLS/SSL Rule Guidelines, on page 8](#)
- [SSL Rule Order](#)

## Guideline for Using TLS/SSL Decryption

Set up **Decrypt - Resign** or **Decrypt - Known Key** rules *only* if your managed device handles encrypted traffic. Decryption rules require processing overhead that can impact performance.

## TLS/SSL Rule Unsupported Features

### Zone conditions and passive interfaces

If you create a **Decrypt - Resign** rule, and later add a security zone with passive interfaces to a zone condition, the system displays a warning icon next to the rule. Because you cannot decrypt traffic by re-signing a certificate in a passive deployment, the rule has no effect until you remove the passive interfaces from the rule or change the rule action.

### Unsupported characters in rule names

Do not use accented characters (for example, Comunicación) in TLS/SSL rule rule names; doing so prevents the policy from being deployed to managed devices.

### TLS 1.3 not supported

The Firepower System does not currently support TLS version 1.3 encryption or decryption. When users visit a web site that negotiates TLS 1.3 encryption, users might see errors similar to the following in their web browser:

- **ERR\_SSL\_PROTOCOL\_ERROR**
- **SEC\_ERROR\_BAD\_SIGNATURE**
- **ERR\_SSL\_VERSION\_INTERFERENCE**

For more information about how to control this behavior, contact Cisco TAC.

## TLS/SSL Do Not Decrypt Guidelines

You should not decrypt traffic if doing so is forbidden by:

- Law; for example, some jurisdictions forbid decrypting financial information
- Company policy; for example, your company might forbid decrypting privileged communications
- Privacy regulations
- Traffic that uses certificate pinning (also referred to as *TLS/SSL pinning*) must remain encrypted to prevent breaking the connection

If you elect to bypass decryption for certain types of traffic, no processing is done on the traffic. The encrypted traffic is first evaluated by SSL policy and then proceeds to the access control policy, where a final allow or block decision is made. Encrypted traffic can be allowed or blocked on any TLS/SSL rule condition, including, but not limited to:

- Certificate status (for example, expired or invalid certificate)
- Protocol (for example, the nonsecure SSL protocol)
- Network (security zone, IP address, VLAN tag, and so on)
- Exact URL or URL category
- Port
- User group

## TLS/SSL Decrypt - Resign Guidelines

You can associate one internal Certificate Authority (CA) certificate and private key with the **Decrypt - Resign** action. If traffic matches this rule, the system re-signs the server certificate with the CA certificate, then acts as a man-in-the-middle. It creates two TLS/SSL sessions, one between client and managed device, one between managed device and server. Each session contains different cryptographic session details, and allows the system to decrypt and reencrypt traffic.

### Best practices

We recommend the following:

- Use the **Decrypt - Resign** rule action for decrypting *outgoing* traffic, as opposed to incoming traffic for which we recommend the **Decrypt - Known Key** rule action.

For more information about **Decrypt - Known Key**, see [TLS/SSL Decrypt - Known Key Guidelines, on page 5](#).

- Always check the **Replace Key Only** check box when you set up a **Decrypt - Resign** rule action.

When a user browses to a web site that uses a *self-signed* certificate, the user sees a security warning in the web browser and is aware that they are communicating with an unsecure site.

When a user browses to a web site that uses a trusted certificate, the user does not see a security warning.

## Details

If you configure a rule with the **Decrypt - Resign** action, the rule matches traffic based on the referenced internal CA certificate's signature algorithm type, in addition to any configured rule conditions. Because you associate one CA certificate with a **Decrypt - Resign** action, you cannot create a TLS/SSL rule that decrypts multiple types of outgoing traffic encrypted with different signature algorithms. In addition, any external certificate objects and cipher suites you add to the rule must match the associated CA certificate encryption algorithm type.

For example, outgoing traffic encrypted with an elliptic curve (EC) algorithm matches a **Decrypt - Resign** rule only if the action references an EC-based CA certificate; you must add EC-based external certificates and cipher suites to the rule to create certificate and cipher suite rule conditions.

Similarly, a **Decrypt - Resign** rule that references an RSA-based CA certificate matches only outgoing traffic encrypted with an RSA algorithm; outgoing traffic encrypted with an EC algorithm does not match the rule, even if all other configured rule conditions match.

## Guidelines and limitations

Also note the following:

### Anonymous cipher suite unsupported

By nature, anonymous cipher suites are not used for authentication and do not use key exchanges. There are limited uses for anonymous cipher suites; for more information, see [RFC 5246, appendix F.1.1.1](#). (Replaced for TLS 1.3 by [RFC 8446 appendix C.5](#).)

You cannot use the **Decrypt - Resign** or **Decrypt - Known Key** action in the rule because anonymous cipher suites are not used for authentication.

### Inline or tap mode limitation

You cannot use the **Decrypt - Resign** action in a passive or inline (tap mode) deployment because the device does not directly inspect traffic. If you create a rule with the **Decrypt - Resign** action that contains passive or inline (tap mode) interfaces within a security zone, the policy editor displays a **Warning** (⚠) next to the rule.

If your SSL policy targets a device with passive or inline (tap mode) interfaces, and contains a **Decrypt - Resign** rule, the system displays an **Information** ⓘ next to the rule. If you later add a zone condition to the TLS/SSL rule that contains passive or inline (tap mode) interfaces, the system displays a **Warning** (⚠). If you deploy an SSL policy that contains a **Decrypt - Resign** rule to a device with passive or inline (tap mode) interfaces, any TLS/SSL sessions that match the rule fail.

### Decrypt - Resign rule action and a Certificate Signing Request

To use a **Decrypt - Resign** rule action, you should create a Certificate Signing Request (CSR) and have it signed by a trusted certificate authority. (You can use the FMC to create a CSR: **Objects > Object Management > PKI > Internal CAs**.)

To be used in a **Decrypt - Resign** rule, your certificate authority (CA) must have at least one of the following extensions:

- **CA: TRUE**

For more information, see the discussion of Basic Constraints in [RFC3280, section 4.2.1.10](#).

- **KeyUsage=CertSign**

For more information see [RFC 5280, section 4.2.1.3](#).

To verify your CSR or CA has at least one of the preceding extensions, you can use the **openssl** command as discussed in a reference such as the [openssl documentation](#).

This is necessary because for **Decrypt - Resign** inspection to work, the certificate that used in the TLS/SSL policy generates certificates on-the-fly and signs them so as to act as man-in-the middle and proxy all TLS/SSL connections.

#### Non-matching cipher suite

The following error is displayed if you attempt to save a TLS/SSL rule with a cipher suite that does not match the certificate. To resolve the issue, see [Verify TLS/SSL Cipher Suites](#).

Traffic cannot match this rule; none of your selected cipher suites contain a signature algorithm that the resigning CA's signature algorithm

#### Untrusted Certificate Authority

If the client does not trust the Certificate Authority (CA) used to re-sign the server certificate, it warns the user that the certificate should not be trusted. To prevent this, import the CA certificate into the client trusted CA store. Alternatively, if your organization has a private PKI, you can issue an intermediate CA certificate signed by the root CA which is automatically trusted by all clients in the organization, then upload that CA certificate to the device.

#### HTTP proxy limitation

The system cannot decrypt traffic if an HTTP proxy is positioned between a client and your managed device, and the client and server establish a tunneled TLS/SSL connection using the CONNECT HTTP method. The **Handshake Errors** undecryptable action determines how the system handles this traffic.

#### Upload signed CA

If you create an internal CA object and choose to generate a certificate signing request (CSR), you cannot use this CA for a **Decrypt - Resign** action until you upload the signed certificate to the object.

#### Use of certain rule conditions



**Note** Use the **Cipher Suite** and **Version** rule conditions *only* in rules with either the **Block** or **Block with reset** rule actions. The use of these conditions in rules with other rule actions can interfere with the system's ClientHello processing, resulting in unpredictable performance.

## TLS/SSL Decrypt - Known Key Guidelines

When you configure the **Decrypt - Known Key** action, you can associate one or more server certificates and paired private keys with the action. If traffic matches the rule, and the certificate used to encrypt the traffic matches the certificate associated with the action, the system uses the appropriate private key to obtain the session encryption and decryption keys. Because you must have access to the private key, this action is best suited to decrypt traffic incoming to servers your organization controls.

Also note the following:

#### Limitation for DHE and ECDHE key exchanges

You cannot use the **Decrypt - Known Key** action in a passive deployment if the cipher suite used to establish the TLS/SSL connection applies either the Diffie-Hellman ephemeral (DHE) or the elliptic

curve Diffie-Hellman ephemeral (ECDHE) key exchange algorithm. If your SSL policy targets a device with passive or inline (tap mode) interfaces, and contains a **Decrypt - Known Key** rule with a cipher suite condition containing either a DHE or an ECDHE cipher suite, the system displays an **Information** ( ⓘ ) next to the rule. If you later add a zone condition to the TLS/SSL rule that contains passive or inline (tap mode) interfaces, the system displays a **Warning** ( ⚠ ).

#### Anonymous cipher suite unsupported

By nature, anonymous cipher suites are not used for authentication and do not use key exchanges. There are limited uses for anonymous cipher suites; for more information, see [RFC 5246, appendix F.1.1.1](#). (Replaced for TLS 1.3 by [RFC 8446 appendix C.5](#).)

You cannot use the **Decrypt - Resign** or **Decrypt - Known Key** action in the rule because anonymous cipher suites are not used for authentication.

#### Cannot match on Distinguished Name or Certificate

You cannot match on **Distinguished Name** or **Certificate** conditions when creating a TLS/SSL rule with a **Decrypt - Known Key** action. The assumption is that if this rule matches traffic, the certificate, subject DN, and issuer DN already match the certificate associated with the rule.

#### Mismatched signature algorithm

If you configure a rule with the **Decrypt - Resign** action, and mismatch signature algorithm type for one or more external certificate objects or cipher suites, the policy editor displays an **Information** ( ⓘ ) next to the rule. If you mismatch signature algorithm type for all external certificate objects, or all cipher suites, the policy displays a warning icon **Warning** ( ⚠ ) next to the rule, and you cannot deploy the access control policy associated with the SSL policy.

#### Certificate pinning

If the customer's browser uses certificate pinning to verify a server certificate, you cannot decrypt this traffic by re-signing the server certificate. To allow this traffic, configure a TLS/SSL rule with the **Do not decrypt** action to match the server certificate common name or distinguished name.

#### Use of certain rule conditions



##### Note

Use the **Cipher Suite** and **Version** rule conditions *only* in rules with either the **Block** or **Block with reset** rule actions. The use of these conditions in rules with other rule actions can interfere with the system's ClientHello processing, resulting in unpredictable performance.

## TLS/SSL Block Guidelines

If decrypted traffic matches an access control rule with an action of **Interactive Block** or **Interactive Block with reset**, the system blocks the matching connection without interaction and the system does *not* display a response page.

Provided you enabled logging in your rule, two connection events are displayed (in **Analysis > Events > Connections**): One event for the interactive block and another event to indicate whether or not the user chose to continue to the site or not.

## TLS/SSL Certificate Pinning Guidelines

Some applications use a technique referred to as *TLS/SSL pinning* or *certificate pinning*, which embeds the fingerprint of the original server certificate in the application itself. As a result, if you configured a TLS/SSL rule with a **Decrypt - Resign** action, when the application receives a resigned certificate from a managed device, validation fails and the connection is aborted.

Because TLS/SSL pinning is used to avoid man-in-the-middle attacks, there is no way to prevent or work around it. You have the following options:

- Create a **Do Not Decrypt** for those applications rule ordered before **Decrypt - Resign** rules.
- Instruct users to access the applications using a web browser.

For more information about rule ordering, see [SSL Rule Order](#).

## TLS/SSL Heartbeat Guidelines

Some applications use the *TLS heartbeat* extension to the Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols defined by [RFC6520](#). TLS heartbeat provides a way to confirm the connection is still alive—either the client or server sends a specified number of bytes of data and requests the other party echo the response. If this is successful, encrypted data is sent.

You can configure a **Max Heartbeat Length** in a Network Analysis Policy (NAP) to determine how to handle TLS heartbeats. For more information, see [The SSL Preprocessor](#).

## TLS/SSL Anonymous Cipher Suite Limitation

By nature, anonymous cipher suites are not used for authentication and do not use key exchanges. There are limited uses for anonymous cipher suites; for more information, see [RFC 5246, appendix F.1.1.1](#). (Replaced for TLS 1.3 by [RFC 8446 appendix C.5](#).)

You cannot use the **Decrypt - Resign** or **Decrypt - Known Key** action in the rule because anonymous cipher suites are not used for authentication.

You can add an anonymous cipher suite to the **Cipher Suite** condition in a TLS/SSL rule, but the system automatically strips anonymous cipher suites during ClientHello processing. For the system to use the rule, you must also configure your TLS/SSL rules in an order that prevents ClientHello processing. For more information, see [SSL Rule Order](#).

## TLS/SSL Normalizer Guidelines

If you enable the **Normalize Excess Payload** option in the inline normalization preprocessor, when the preprocessor normalizes decrypted traffic, it might drop a packet and replace it with a trimmed packet. This does not end the TLS/SSL session. If the traffic is allowed, the trimmed packet is encrypted as part of the TLS/SSL session.

# Other TLS/SSL Rule Guidelines

## Users and groups

If you add a group or user to a rule, then change your realm settings to exclude that group or user, the rule has no effect. (The same applies to disabling the realm.) For more information about realms, see [Create a Realm](#).

## Categories in TLS/SSL rules

If your SSL policy has a **Decrypt - Resign** action but web sites are not being decrypted, check **Category** page on rules associated with that policy.

In some cases, a web site redirects to another site for authentication or other purposes and the redirected site might have a different URL categorization than the site you're trying to decrypt. For example, `gmail.com` (**Web based email** category) redirects to `accounts.gmail.com` (**Internet Portals** category) for authentication. Be sure to include all relevant categories in the SSL rule.




---

**Note** In order to fully process traffic based on URL category, you must also configure URL filtering. See the [URL Filtering](#) chapter.

---

## Query for URLs not in the local database

If you create a **Decrypt - Resign** rule and users browse to a web site whose category and reputation are not in the local database, data might not be decrypted. Some web sites are not categorized in the local database and, if not, data from those web sites is not decrypted by default.

You can control this behavior with the setting **System > Integration > Cisco CSI**, and check **Query Cisco CSI for Unknown URLs**.

For more information about this option, see [Cisco Clouds](#).

# Requirements and Prerequisites for TLS/SSL Rules

## Model Support

Any except NGIPSV.

## Supported Domains

Any

## User Roles

- Admin
- Access Admin
- Network Admin

# Encrypted Traffic Inspection Configuration

You must create reusable public key infrastructure (PKI) objects to control encrypted traffic based on encrypted session characteristics and decrypt encrypted traffic. You can add this information on the fly when uploading trusted certificate authority (CA) certificates to the SSL policy and creating SSL rule conditions, creating the associated object in the process. However, configuring these objects ahead of time reduces the chance of improper object creation.

## Decrypting Encrypted Traffic with Certificates and Paired Keys

The system can decrypt incoming encrypted traffic if you configure an internal certificate object by uploading the server certificate and private key used to encrypt the session. If you reference that object in an SSL rule with an action of **Decrypt - Known Key** and traffic matches that rule, the system uses the uploaded private key to decrypt the session.

The system can also decrypt outgoing traffic if you configure an internal CA object by uploading a CA certificate and private key. If you reference that object in a TLS/SSL rule with an action of **Decrypt - Resign** and traffic matches that rule, the system re-signs the server certificate passed to the client browser, then acts as a man-in-the-middle to decrypt the session. You can optionally replace the self-signed certificate key only and not the entire certificate, in which case users see a self-signed certificate key notice in the browser.

## Controlling Traffic Based on Encrypted Session Characteristics

The system can control encrypted traffic based on the cipher suite or server certificate used to negotiate the session. You can configure one of several different reusable objects and reference the object in a TLS/SSL rule condition to match traffic. The following table describes the different types of reusable objects you can configure:

If you configure...	You can control encrypted traffic based on whether...
A cipher suite list containing one or more cipher suites	The cipher suite used to negotiate the encrypted session matches a cipher suite in the cipher suite list
A trusted CA object by uploading a CA certificate your organization trusts	The trusted CA trusts the server certificate used to encrypt the session, whether: <ul style="list-style-type: none"> <li>• The CA issued the certificate directly</li> <li>• The CA issued a certificate to an intermediate CA that issued the server certificate</li> </ul>
An external certificate object by uploading a server certificate	The server certificate used to encrypt the session matches the uploaded server certificate
A distinguished name object containing a certificate subject or issuer distinguished name	The subject or issuer common name, country, organization, or organizational unit on the certificate used to encrypt the session matches the configured distinguished name

## Related Topics

[Cipher Suite Lists](#)

[Distinguished Name Objects](#)

[PKI Objects](#)

# Creating and Modifying TLS/SSL Rules

## Procedure

---

**Step 1** Log in to the Firepower Management Center.

**Step 2** Click **Policies > Access Control > SSL**.

**Step 3** Click **Edit** (  ) next to the SSL policy.

If **View** (  ) appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

**Step 4** You have the following choices:

- To add a new rule, click **Add Rule**.
- To edit an existing rule, click **Edit** (  ).

**Step 5** Enter a **Name** for the rule.

Do not use accented characters (for example, Comunicación) in TLS/SSL rule rule names; doing so prevents the policy from being deployed to managed devices.

**Step 6** Specify whether the rule is **Enabled**.

**Step 7** Specify the rule position; see [TLS/SSL Rule Order Evaluation](#).

**Step 8** Click a rule **Action**; see [Configuring TLS/SSL Rule Actions, on page 17](#).

**Step 9** Configure the rule's conditions; see [TLS/SSL Rule Condition Types, on page 14](#).

**Step 10** Click **Save**.

If the following error displays, see [Verify TLS/SSL Cipher Suites: Traffic cannot match this rule; none of your selected cipher suites contain a signature algorithm that the resigning CA's signature algorithm](#).

---

## What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

# Adding a TLS/SSL Rule to a Rule Category

## Procedure

---

**Step 1** In the SSL rule editor, from the **Insert** drop-down list, select **Into Category**, then select the category you want to use.

**Step 2** Click **Save**.

**Tip** When you save the rule, it is placed last in that category.

---

#### What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

## Positioning a TLS/SSL Rule by Number

#### Procedure

---

**Step 1** In the SSL rule editor, from the **Insert** drop-down list, select **above rule** or **below rule**, then type the appropriate rule number.

**Step 2** Click **Save**.

**Tip** When you save the rule, it is placed where you specified.

---

#### What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

## TLS/SSL Rule Search

You can search the list of TLS/SSL rules for matching values using an alphanumeric string, including spaces and printable, special characters. The search inspects the rule name and any rule condition you have added to the rule. For rule conditions, the search matches any name or value you can add for each condition type (zone, network, application, and so on). This includes individual object names or values, group object names, individual object names or values within a group, and literal values.

You can use complete or partial search strings. The column for matching values is highlighted for each matching rule. For example, if you search on all or part of the string `100Bao`, at a minimum, the Applications column is highlighted for each rule where you have added the `100Bao` application. If you also have a rule named `100Bao`, both the Name and Applications columns are highlighted.

You can navigate to each previous or next matching rule. A status message displays the current match and the total number of matches.

Matches may occur on any page of a multi-page rule list. When the first match is not on the first page, the page where the first match occurs is displayed. Selecting the next match when you are at the last match takes you to the first match, and selecting the previous match when you are at the first match takes you to the last match.

## Searching TLS/SSL Rules

### Procedure

---

**Step 1** In the SSL policy editor, click the **Search Rules** prompt, type a search string, then press Enter.

**Tip** Columns for rules with matching values are highlighted, with differentiated highlighting for the indicated (first) match.

**Step 2** Find the rules you are interested in:

- To navigate between matching rules, click **Next-Match** or **Previous-Match**.
  - To refresh the page and clear the search string and any highlighting, click **Clear** (x).
- 

## Enabling and Disabling TLS/SSL Rules

When you create a TLS/SSL rule, it is enabled by default. If you disable a rule, the system does not use it to evaluate network traffic and stops generating warnings and errors for that rule. When viewing the list of rules in an SSL policy, disabled rules are grayed out, although you can still modify them. Note that you can also enable or disable a TLS/SSL rule using the rule editor.

### Procedure

---

**Step 1** In the SSL policy editor, right-click a rule and choose a rule state.

**Step 2** Click **Save**.

---

### What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

## Moving a TLS/SSL Rule

### Procedure

---

**Step 1** In the SSL policy editor, select the rules by clicking in a blank area for each rule.

**Step 2** Right-click the rule and select **Cut**.

**Step 3** Right-click a blank area for a rule next to where you want to paste the cut rules and select **Paste above** or **Paste below**.

**Tip** You cannot copy and paste TLS/SSL rules between two different SSL policies.

**Step 4** Click Save.

---

#### What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

## Adding a New TLS/SSL Rule Category

You can create custom categories between the Standard Rules and Root Rules categories to further organize your rules without having to create additional policies. You can rename and delete categories that you add. You cannot move these categories, but you can move rules into, within, and out of them.

#### Procedure

---

**Step 1** In the policy editor, click **Add Category**.

**Tip** If your policy already contains rules, you can click a blank area in the row for an existing rule to set the position of the new category before you add it. You can also right-click an existing rule and select **Insert new category**.

**Step 2** Type a **Name**.

**Step 3** You have the following choices:

- Select **above Category** from the first **Insert** drop-down list, then select the category above which you want to position the rule from the second drop-down list.
- Select **below rule** from the drop-down list, then enter an existing rule number. This option is valid only when at least one rule exists in the policy.
- Select **above rule** from the drop-down list, then, enter an existing rule number. This option is valid only when at least one rule exists in the policy.

**Step 4** Click **OK**.

**Tip** Rules in a category you delete are added to the category above.

**Step 5** Click Save.

---

## TLS/SSL Rule Conditions

An SSL rule's conditions identify the type of encrypted traffic the rule handles. Conditions can be simple or complex, and you can specify more than one condition type per rule. Only if traffic meets all the conditions in a rule does the rule apply to the traffic.

If you do not configure a particular condition for a rule, the system does not match traffic based on that criterion. For example, a rule with a certificate condition but no version condition evaluates traffic based on the server certificate used to negotiate the session, regardless of the session SSL or TLS version.

## TLS/SSL Rule Condition Types

Every TLS/SSL rule has an associated action that determines the following for matching encrypted traffic:

- Handling: Most importantly, the rule action governs whether the system will monitor, trust, block, or decrypt encrypted traffic that matches the rule's conditions
- Logging: The rule action determines when and how you can log details about matching encrypted traffic.

Your TLS/SSL inspection configuration handles, inspects, and logs decrypted traffic:

- The SSL policy's undecryptable actions handle traffic that the system cannot decrypt.
- The policy's default action handles traffic that does not meet the condition of any non-Monitor TLS/SSL rule.

You can log a connection event when the system blocks or trusts an encrypted session. You can also force the system to log connections that it decrypts for further evaluation by access control rules, regardless of how the system later handles or inspects the traffic. Connection logs for encrypted sessions contain details about the encryption, such as the certificate used to encrypt that session. You can log only end-of-connection events, however:

- For blocked connections (Block, Block with reset), the system immediately ends the sessions and generates an event
- For trusted connections (Do not decrypt), the system generates an event when the session ends

## TLS/SSL Rule Condition Types

When you add or edit an SSL rule, use the tabs on the left side of the lower portion of the rule editor to add and edit rule conditions.

**Table 1: TLS/SSL Rule Condition Types**

This Condition...	Matches Encrypted Traffic...	Details
Zones	Entering or leaving a device via an interface in a specific security zone	A security zone is a logical grouping of one or more interfaces according to your deployment and security policies. Interfaces in a zone may be located across multiple devices.
Networks	By its source or destination IP address, country, or continent	You can explicitly specify IP addresses. The geolocation feature also allows you to control traffic based on its source or destination country or continent.
VLAN Tags	Tagged by VLAN	The system uses the innermost VLAN tag to identify a packet by VLAN.
Ports	By its source or destination port	You can control encrypted traffic based on the TCP port.
Users	By the user involved in the session	You can control encrypted traffic based on the LDAP user logged into a host involved in an encrypted, monitored session. You can control traffic based on individual users or groups retrieved from a Microsoft Active Directory server.

This Condition...	Matches Encrypted Traffic...	Details
Applications	By the application detected in a session	You can control access to individual applications in encrypted sessions, or filter access according to basic characteristics: type, risk, business relevance, and categories.
Categories	By the URL requested in the session, based on the certificate subject distinguished name	You can limit the websites that users on your network can access based on the URL's general classification and risk level.
Distinguished Names	The URL the user enters in the browser matches the Common Name (CN), or the URL is contained in the certificate's <a href="#">Subject Alternative Name (SAN)</a>	You can control encrypted traffic based on the CA that issued a server certificate, or the server certificate holder.
Certificates	By the server certificate used to negotiate the encrypted session	You can control encrypted traffic based on the server certificate passed to the user's browser in order to negotiate the encrypted session.
Certificate Status	By properties of the server certificate used to negotiate the encrypted session	You can control encrypted traffic based on a server certificate's status.
Cipher Suites	By the cipher suite used to negotiate the encrypted session	You can control encrypted traffic based on the cipher suite selected by the server to negotiate the encrypted session.
Versions	By the version of SSL or TLS used to encrypt the session	You can control encrypted traffic based on the version of SSL or TLS used to encrypt the session.

### Related Topics

- [Network-Based TLS/SSL Rule Conditions](#)
- [User-Based TLS/SSL Rule Conditions](#)
- [Reputation-Based URL Blocking in Encrypted Traffic](#)
- [Server Certificate-Based TLS/SSL Rule Conditions](#)

## TLS/SSL Rule Actions

The following sections discuss the actions available with TLS/SSL rules.

### Related Topics

- [TLS/SSL Rule Decrypt Actions, on page 16](#)
- [TLS/SSL Rule Blocking Actions, on page 16](#)
- [TLS/SSL Rule Do Not Decrypt Action, on page 16](#)
- [TLS/SSL Rule Monitor Action, on page 15](#)

## TLS/SSL Rule Monitor Action

The **Monitor** action is not designed to permit or deny traffic. Rather, its primary purpose is to force connection logging, regardless of how matching traffic is eventually handled. Traffic is then matched against additional rules, if present, to determine whether to trust, block, or decrypt it. The first non-Monitor rule matched determines traffic flow and any further inspection. If there are no additional matching rules, the system uses the default action.

Because the primary purpose of Monitor rules is to track network traffic, the system automatically logs end-of-connection events for monitored traffic to the Firepower Management Center database, regardless of the logging configuration of the rule or default action that later handles the connection.

## TLS/SSL Rule Do Not Decrypt Action

The **Do Not Decrypt** action passes encrypted traffic for evaluation by the access control policy's rules and default action. Because some access control rule conditions require unencrypted traffic, this traffic may match fewer rules. The system cannot perform deep inspection on encrypted traffic, such as intrusion or file inspection.

Typical reasons for a **Do Not Decrypt** rule action include:

- When decrypting TLS/SSL traffic is prohibited by law.
- Sites you know you can trust.
- Sites you can disrupt by inspecting traffic (such as Windows Update).
- To view the values of TLS/SSL fields using connection events. (You do not need to decrypt traffic to view connection event fields.) For more information, see [Requirements for Populating Connection Event Fields](#).

For more information, see [Default Handling Options for Undecryptable Traffic](#)

## TLS/SSL Rule Blocking Actions

The Firepower System provides the following TLS/SSL rule actions for traffic you do not want to pass through the system:

- **Block** to terminate the connection, resulting in an error in the client browser.

The error message does not indicate the site was blocked due to policy. Instead, errors might indicate that there are no common encryption algorithms. It is not obvious from this message that you blocked the connection on purpose.

- **Block with reset** to terminate and reset the connection, resulting in an error in the client browser.

The error indicates the connection was reset but does not indicate why.



**Tip** You cannot use the **Block** or **Block with reset** action in a passive or inline (tap mode) deployment because the device does not directly inspect the traffic. If you create a rule with the **Block** or **Block with reset** action that contains passive or inline (tap mode) interfaces within a security zone condition, the policy editor displays a warning () next to the rule.

### Related Topics

[About HTTP Response Pages](#)

## TLS/SSL Rule Decrypt Actions

The **Decrypt - Known Key** and **Decrypt - Resign** actions decrypt encrypted traffic. The system inspects decrypted traffic with access control. Access control rules handle decrypted and unencrypted traffic identically

— you can inspect it for discovery data as well as detect and block intrusions, prohibited files, and malware. The system reencrypts allowed traffic before passing it to its destination.

We recommend you use a certificate from a trusted Certificate Authority (CA) to decrypt traffic. This prevents **Invalid Issuer** from being displayed in the SSL Certificate Status column in connection events.

For more information about adding trusted objects, see [Trusted Certificate Authority Objects](#).

## Configuring TLS/SSL Rule Actions

### Before you begin

See:

- [TLS/SSL Rule Blocking Actions, on page 16](#)
- [TLS/SSL Rule Do Not Decrypt Action, on page 16](#)
- [TLS/SSL Rule Monitor Action, on page 15](#)

### Procedure

---

**Step 1** In the SSL policy editor, you have the following options:

- To add a new rule, click **Add Rule**.
- To edit an existing rule, click **Edit** ().

**Step 2** Select a rule action from the **Action** drop-down list.

- To block encrypted traffic, select **Block**.
- To block encrypted traffic and reset the connection, select **Block with reset**.
- To decrypt incoming traffic, see [Configuring a Decrypt - Known Key Action, on page 18](#) for more information.
- To decrypt outgoing traffic, see [Configuring a Decrypt - Resign Action, on page 18](#) for more information.
- To log encrypted traffic, select **Monitor**.
- To not decrypt encrypted traffic, select **Do not decrypt**.

**Step 3** Click **Add**.

---

### What to do next

- Configure rule conditions, as described in [Network-Based TLS/SSL Rule Conditions](#), [User-Based TLS/SSL Rule Conditions](#), [Reputation-Based TLS/SSL Rule Conditions](#), and [Server Certificate-Based TLS/SSL Rule Conditions](#).
- Deploy configuration changes; see [Deploy Configuration Changes](#).

**Before you begin**

See [TLS/SSL Decrypt - Resign Guidelines](#), on page 3.

**Procedure**

**Step 1** In the SSL rule editor, select **Decrypt - Resign** from the **Action** list.

**Step 2** Select an internal CA certificate object from the list.

**Step 3** Check **Replace Key**.

Always check the **Replace Key Only** check box when you set up a **Decrypt - Resign** rule action.

When a user browses to a web site that uses a *self-signed* certificate, the user sees a security warning in the web browser and is aware that they are communicating with an unsecure site.

When a user browses to a web site that uses a trusted certificate, the user does not see a security warning.

**Step 4** Click **Add**.

**Step 5** *Optional.* To use a Trusted CA certificate in your SSL policy so you can avoid **Invalid Issuer** in the SSL Certificate Status column in connection events, add the certificate to the policy:

a) In the SSL policy editor page, click **Trusted CA Certificates**.

b) Add the CA certificate corresponding to your known key to the SSL policy.

**What to do next**

- Deploy configuration changes; see [Deploy Configuration Changes](#).

**Configuring a Decrypt - Known Key Action****Before you begin**

See [TLS/SSL Decrypt - Known Key Guidelines](#), on page 5.

**Procedure**

**Step 1** In the SSL rule editor, select **Decrypt - Known Key** from the **Action** drop-down list.

**Step 2** Click the **Click to select decryption certs** field.

**Step 3** Select one or more internal certificate objects in the **Available Certificates** list, then click **Add to Rule**.

**Step 4** Click **OK**.

**Step 5** Click **Add**.

**Step 6** *Optional.* To use a Trusted CA certificate in your SSL policy so you can avoid **Invalid Issuer** in the SSL Certificate Status column in connection events, add the certificate to the policy:

a) In the SSL policy editor page, click **Trusted CA Certificates**.

- b) Add the CA certificate corresponding to your known key to the SSL policy.
- 

#### What to do next

- Deploy configuration changes; see [Deploy Configuration Changes](#).

## TLS/SSL Rules Management

The **Rules** page of the SSL policy editor allows you to add, edit, search, move, enable, disable, delete, and otherwise manage TLS/SSL rules in your policy.

