



# Decryption Tuning Using TLS/SSL Rules

The following topics provide an overview of how to configure TLS/SSL rule conditions:

- [TLS/SSL Rule Conditions Overview, on page 1](#)
- [Requirements and Prerequisites for Decryption Tuning, on page 2](#)
- [Server Certificate-Based TLS/SSL Rule Conditions, on page 2](#)

## TLS/SSL Rule Conditions Overview

A basic TLS/SSL rule applies its rule action to all encrypted traffic inspected by the device. To better control and decrypt encrypted traffic, you can configure rule conditions to handle and log specific types of traffic. Each TLS/SSL rule can contain 0, 1, or more rule conditions; a rule matches traffic only if the traffic matches every condition in that TLS/SSL rule.



**Note** When traffic matches a rule, the device applies the configured rule action to the traffic. When the connection ends, the device logs the traffic if configured to do so.

Each rule condition allows you to specify one or more properties of traffic you want to match against; these properties include details of:

- The flow of traffic, including the security zone through which it travels, IP address and port, country of origin or destination, and origin or destination VLAN.
- The user associated with a detected IP address.
- The traffic payload, including the application detected in the traffic.
- The connection encryption, including the TLS/SSL protocol version and cipher suite and server certificate used to encrypt the connection.
- The category and reputation of the URL specified in the server certificate's distinguished name..

### Related Topics

- [Interface Conditions](#)
- [Network Conditions](#)
- [VLAN Conditions](#)
- [User, Realm, and ISE Attribute Conditions \(User Control\)](#)

[Application Conditions \(Application Control\)](#)  
[Port and ICMP Code Conditions](#)  
[Filtering HTTPS Traffic](#)  
[Server Certificate-Based TLS/SSL Rule Conditions](#), on page 2  
[Certificate Distinguished Name TLS/SSL Rule Conditions](#)  
[Certificate Status TLS/SSL Rule Conditions](#), on page 9  
[Cipher Suite TLS/SSL Rule Conditions](#), on page 15  
[Encryption Protocol Version TLS/SSL Rule Conditions](#), on page 18  
[ClientHello Message Handling](#)

## Requirements and Prerequisites for Decryption Tuning

### Model Support

Any except NGIPSv.

### Supported Domains

Any

### User Roles

- Admin
- Access Admin
- Network Admin

## Server Certificate-Based TLS/SSL Rule Conditions

TLS/SSL rules can handle and decrypt encrypted traffic based on server certificate characteristics. You can configure TLS/SSL rules based on the following server certificate attributes:

- Distinguished name conditions allow you to handle and inspect encrypted traffic based on the CA that issued a server certificate, or the certificate holder. Based on the issuer distinguished name, you can handle traffic based on the CA that issued a site's server certificate.
- Certificate conditions in TLS/SSL rules allow you to handle and inspect encrypted traffic based on the server certificate used to encrypt that traffic. You can configure a condition with one or more certificates; traffic matches the rule if the certificate matches any of the condition's certificates.
- Certificate status conditions in TLS/SSL rules allow you to handle and inspect encrypted traffic based on the status of the server certificate used to encrypt the traffic, including whether a certificate is valid, revoked, expired, not yet valid, self-signed, signed by a trusted CA, whether the Certificate Revocation List (CRL) is valid; whether the Server Name Indication (SNI) in the certificate matches the server in the request.
- Cipher suite conditions in TLS/SSL rules allow you to handle and inspect encrypted traffic based on the cipher suite used to negotiate the encrypted session.

- Session conditions in TLS/SSL rules allow you to inspect encrypted traffic based on the SSL or TLS version used to encrypt the traffic.

To detect multiple cipher suites in a rule, the certificate issuer, or the certificate holder, you can create reusable cipher suite list and distinguished name objects and add them to your rule. To detect the server certificate and certain certificate statuses, you must create external certificate and external CA objects for the rule.

## Distinguished Name (DN) Rule Conditions

This topic discusses how to use distinguished name conditions in a TLS/SSL rule. If you're not sure, you can find a certificate's [Subject Alternative Name \(SAN\)](#) and Common Name using a web browser, then you can add those values to a TLS/SSL rule as distinguished name conditions.

For detailed information about SANs, see [RFC 528, section 4.2.1.6](#).

The following sections discuss:

- [DN rule matching example](#)
- [How the Firepower System uses the SNIs and SANs](#)
- [How to find a certificate's Common Name and Subject Alternative Names](#)
- [How to add a DN rule condition](#)

### DN rule matching example

Following is an example of DN rule conditions in a Do Not Decrypt rule. Suppose you want to make sure to *not* decrypt traffic going to `amp.cisco.com` or to YouTube. You could set up your DN conditions as follows:

The screenshot shows the 'Add Rule' configuration window. The 'Name' field is 'DND', 'Enabled' is checked, and 'Action' is 'Do not decrypt'. The 'DN' tab is selected, showing 'Subject DNs (4)' with entries: CN=\*.amp.cisco.com, CN=\*.amp.cisco.com, CN=\*.youtube.com, and CN=\*.yt.be. The 'Available DNs' list on the left includes various domains like Cisco-Undecryptable-Sites, CN\_api.smarthings.com, CN\_apps.apple.com, CN\_ciscospark.com, CN\_citrixonline.com, CN\_core.windows.net, CN\_data.microsoft.com, and CN\_data.toolbar.yahoo.com. Buttons for 'Add to Subject' and 'Add to Issuer' are visible. At the bottom are 'Cancel' and 'Add' buttons.

The preceding DN rule conditions would match the following URLs and therefore, the traffic would be undecrypted an earlier rule prevented it:

- `www.amp.cisco.com`
- `auth.amp.cisco.com`

- `auth.us.amp.cisco.com`
- `www.youtube.com`
- `kids.youtube.com`
- `www.yt.be`

The preceding DN rule conditions would *not* match any of the following URLs and therefore, the traffic would not match the Do Not Decrypt rule but might match any other TLS/SSL in the same SSL policy.

- `amp.cisco.com`
- `youtube.com`
- `yt.be`

To match any of the preceding host names, add more CNs to the rule (for example, adding `CN=yt.be` would match that URL.)

### How the Firepower System uses the SNI and SANs


The host name portion of the URL in the client request is the [Server Name Indication \(SNI\)](#). The client specifies which hostname they want to connect to (for example, `auth.amp.cisco.com`) using the SNI extension in the TLS handshake. The server then selects the corresponding private key and certificate chain that are required to establish the connection while hosting all certificates on a single IP address.

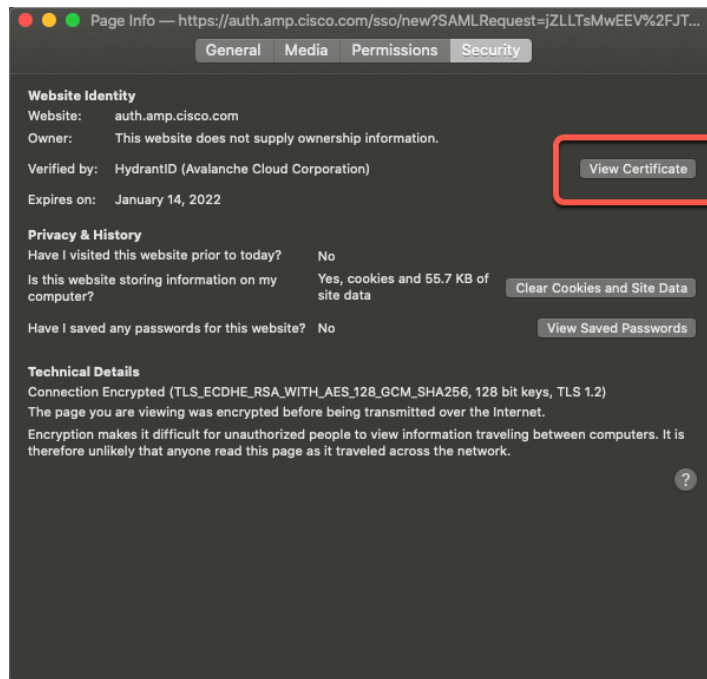
If there's a match between the SNI and the CN or a SAN in the certificate, we use the SNI when comparing against the DNs listed in the rule. If there is no SNI or if it doesn't match the certificate, we use the certificate's CN when comparing against the DNs listed in the rule.

### How to find a certificate's Common Name and subject alternative names

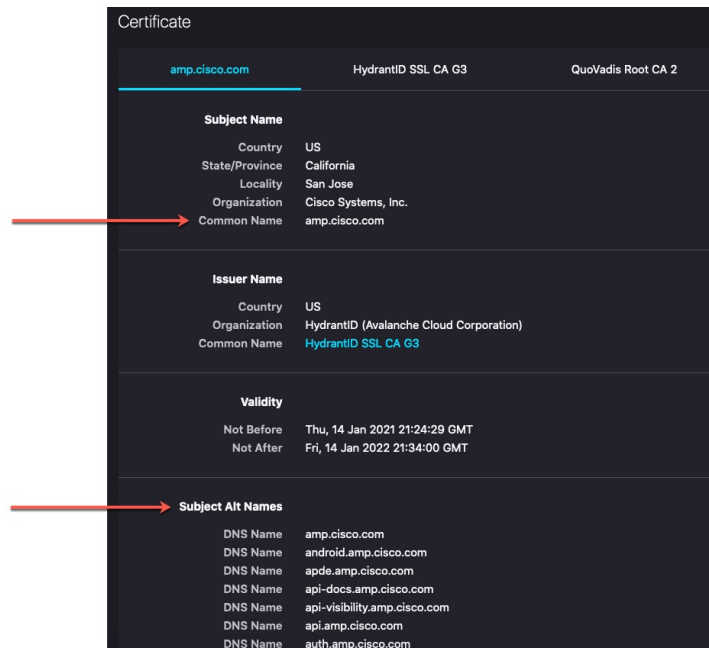
To find any certificate's Common Name, use the following steps. You can even use these steps to find the common name and SANs for a self-signed certificate.

These steps are for Firefox but other browsers are similar. The following procedure uses `amp.cisco.com` as an example.

1. Browse to `amp.cisco.com` in Firefox.
2. In the browser's location bar, to the left of the URL, click .
3. Click **Connection secure > More Information**.  
(For a non-secure or self-signed certificate, click **Connection not secure > More Information**.)
4. On the Page Info dialog box, click **View Certificate**.



5. The next page shows certificate details.



Note the following:

- CN=`auth.amp.cisco.com`, if used as a DN rule condition, would match *only* that host name (that is, SNI). The SNI `amp.cisco.com` would *not* match.
- To match as many domain name fields as possible, use wildcards.

For example, to match `auth.amp.cisco.com`, use `CN=*.amp.cisco.com`. To match `auth.us.amp.cisco.com`, use `CN=*.*.amp.cisco.com`.

A DN like `CN=*.example.com` matches `www.example.com` but *not* `example.com`. To match both SNIs, use two DNs in the rule condition.

- Don't go overboard with wildcards though. For example, a DN object like `CN=*.google.com` matches a very large number of SANs. Instead of `CN=*.google.com`, use a DN object like `CN=*.youtube.com` as the DN object so it matches names like `www.youtube.com`.

You can also use variations of the SNI that match SANs like `CN=*.youtube.com`, `CN=youtu.be`, `CN=*.yt.be`, and so on.

- A self-signed certificate should work the same way. You can confirm it's a self-signed certificate by the fact the issuer DN is the same as the subject DN.

### How to add a DN rule condition

After you know the CN you want to match, edit the TLS/SSL rule in one of the following ways:

- Use an existing DN.

Click the name of a DN and then click either **Add to Subject** or **Add to Issuer**. (**Add to Subject** is much more common.) To view the value of a DN object, hover the mouse pointer over it.)

- Create a new DN object.

Click **Plus** ( **+** ) to the right of Available DNs. The DN object must consist of a name and a value.

- Add the DN directly.

Enter the DN in the field at the bottom of the **Subject DNs** field or the **Issuer DNs** field. (**Subject DNs** is more common.) After you enter the DN, click **Add**.

The screenshot shows the 'Add Rule' dialog with the 'DN' tab selected. The 'Available DNs' list on the left contains the following items: Cisco-Undecryptable-Sites, CN\_apl.smarthings.com, CN\_apps.apple.com, CN\_ciscopark.com, CN\_citrixonline.com, CN\_core.windows.net, CN\_data.microsoft.com, and CN\_data.toolbar.yahoo.com. The 'Subject DNs (0)' and 'Issuer DNs (0)' lists are empty. A red box highlights the 'Add' button next to the 'CN=\*.amp.cisco.com' entry in the 'Subject DNs' list. The 'Action' is set to 'Do not decrypt'. The 'Name' field is empty, and the 'Enabled' checkbox is checked. The 'Insert' dropdown is set to 'into Category', and the 'Standard Rules' dropdown is set to 'Standard Rules'.

## Controlling Encrypted Traffic by Certificate Distinguished Name

**Step 1** In the SSL rule editor, select DN.

**Step 2** Find the distinguished names you want to add from the **Available DNs**, as follows:

- To add a distinguished name object on the fly, which you can then add to the condition, click **Add** (+) above the **Available DNs** list.
- To search for distinguished name objects and groups to add, click the **Search by name or value** prompt above the **Available DNs** list, then type either the name of the object, or a value in the object. The list updates as you type to display matching objects.

**Step 3** To select an object, click it. To select all objects, right-click and then select **Select All**.

**Step 4** Click **Add to Subject** or **Add to Issuer**.

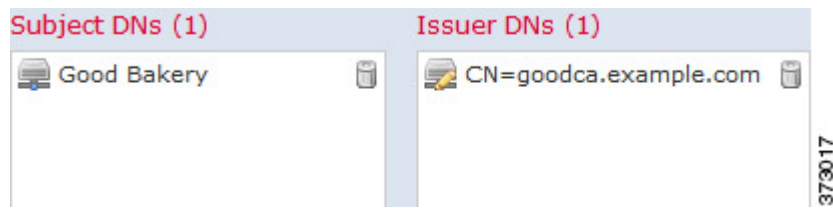
**Tip** You can also drag and drop selected objects.

**Step 5** Add any literal common names or distinguished names that you want to specify manually. Click the **Enter DN or CN** prompt below the **Subject DNs** or **Issuer DNs** list; then type a common name or distinguished name and click **Add**.

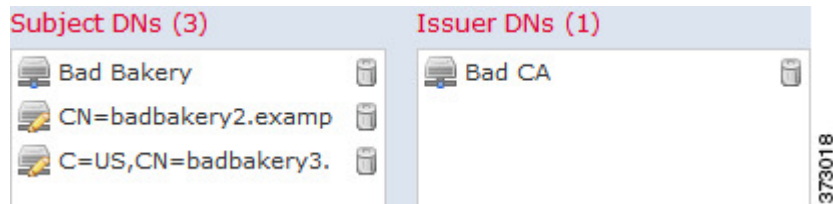
**Step 6** Add or continue editing the rule.

### Example

The following figure shows a distinguished name rule condition searching for certificates issued to goodbakery.example.com or issued by goodca.example.com. Traffic encrypted with these certificates is allowed, subject to access control.



The following figure shows a distinguished name rule condition searching for certificates issued to badbakery.example.com and associated domains, or certificates issued by badca.example.com. Traffic encrypted with these certificates is decrypted using a re-signed certificate.



#### What to do next

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

#### Related Topics

[Distinguished Name Objects](#)

## Certificate TLS/SSL Rule Conditions

When you build a certificate-based TLS/SSL rule condition, you can upload a server certificate; you save the certificate as an external certificate *object*, which is reusable and associates a name with a server certificate. Alternately, you can configure certificate conditions with existing external certificate objects and object groups.

You can search the **Available Certificates** field in the rule condition based for external certificate objects and object groups based on the following certificate distinguished name characteristics:

- Subject or issuer common name (CN)
- Subject or issuer organization (O)
- Subject or issuer organizational unit (OU)

You can choose to match against multiple certificates in a single certificate rule condition; if the certificate used to encrypt the traffic matches any of the uploaded certificates, the encrypted traffic matches the rule.

You can add a maximum of 50 external certificate objects and external certificate object groups to the **Selected Certificates** in a single certificate condition.

Note the following:

- You cannot configure a certificate condition if you also select the **Decrypt - Known Key** action. Because that action requires you to select a server certificate to decrypt traffic, the implication is that the certificate already matches the traffic.



- If you configure a certificate condition with an external certificate object, any cipher suites you add to a cipher suite condition, or internal CA objects you associate with the **Decrypt - Resign** action, must match the external certificate's signature algorithm type. For example, if your rule's certificate condition references an EC-based server certificate, any cipher suites you add, or CA certificates you associate with the **Decrypt - Resign** action, must also be EC-based. If you mismatch signature algorithm types in this case, the policy editor displays a warning next to the rule.
- The first time the system detects an encrypted session to a new server, certificate data is not available for ClientHello processing, which can result in an undecrypted first session. After the initial session, the managed device caches data from the server Certificate message. For subsequent connections from the same client, the system can match the ClientHello message conclusively to rules with certificate conditions and process the message to maximize decryption potential.

## Controlling Encrypted Traffic by Certificate

**Step 1** In the SSL rule editor, select Certificate.

**Step 2** Find the server certificates you want to add from the **Available Certificates**, as follows;

- To add an external certificate object on the fly, which you can then add to the condition, click **Add** (🟢) above the **Available Certificates** list.
- To search for certificate objects and groups to add, click the **Search by name or value** prompt above the **Available Certificates** list, then type either the name of the object, or a value in the object. The list updates as you type to display matching objects.

**Step 3** To select an object, click it. To select all objects, right-click and then select **Select All**.

**Step 4** Click **Add to Rule**.

**Tip** You can also drag and drop selected objects.

**Step 5** Add or continue editing the rule.

### What to do next

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

### Related Topics

[External Certificate Objects](#)

## Certificate Status TLS/SSL Rule Conditions

For each certificate status TLS/SSL rule condition you configure, you can match traffic against the presence or absence of a given status. You can select several statuses in one rule condition; if the certificate matches any of the selected statuses, the rule matches the traffic.

You can choose to match against the presence or absence of multiple certificate statuses in a single certificate status rule condition; the certificate needs to match only one of the criteria to match the rule.

You should consider, when setting this parameter, whether you're configuring a decrypt rule or a block rule. Typically, you should click **Yes** for a block rule and **No** for a decrypt rule. Examples:

- If you're configuring a **Decrypt - Resign** rule, the default behavior is to decrypt traffic with an expired certificate. To change that behavior, click **No** for **Expired** so traffic with an expired certificate is not decrypted and resigned.
- If you're configuring a **Block** rule, the default behavior is to allow traffic with an expired certificate. To change that behavior click **Yes** for **Expired** so traffic with an expired certificate is blocked.

The following table describes how the system evaluates encrypted traffic based on the encrypting server certificate's status.

**Table 1: Certificate Status Rule Condition Criteria**

Status Check	Status Set to Yes	Status Set to No
Revoked	The policy trusts the CA that issued the server certificate, and the CA certificate uploaded to the policy contains a CRL that revokes the server certificate.	The policy trusts the CA that issued the server certificate, and the CA certificate uploaded to the policy does not contain a CRL that revokes the server certificate.
Self-signed	The detected server certificate contains the same subject and issuer distinguished name.	The detected server certificate contains different subject and issuer distinguished names.
Valid	All of the following are true: <ul style="list-style-type: none"> <li>• The policy trusts the CA that issued the certificate.</li> <li>• The signature is valid.</li> <li>• The issuer is valid.</li> <li>• None of the policy's trusted CAs revoked the certificate.</li> <li>• The current date is between the certificate Valid From and Valid To date.</li> </ul>	At least one of the following is true: <ul style="list-style-type: none"> <li>• The policy does not trust the CA that issued the certificate.</li> <li>• The signature is invalid.</li> <li>• The issuer is invalid.</li> <li>• A trusted CA in the policy revoked the certificate.</li> <li>• The current date is before the certificate Valid From date.</li> <li>• The current date is after the certificate Valid To date.</li> </ul>
Invalid signature	The certificate's signature cannot be properly validated against the certificate's content.	The certificate's signature is properly validated against the certificate's content.
Invalid issuer	The issuer CA certificate is not stored in the policy's list of trusted CA certificates.	The issuer CA certificate is stored in the policy's list of trusted CA certificates.
Expired	The current date is after the certificate Valid To date.	The current date is before or on the certificate Valid To date.
Not yet valid	The current date is before the certificate Valid From date.	The current date is after or on the certificate Valid From date.

Status Check	Status Set to Yes	Status Set to No
Invalid certificate	<p>The certificate is not valid. At least one of the following is true:</p> <ul style="list-style-type: none"> <li>Invalid or inconsistent certificate extension; that is, a certificate extension had an invalid value (for example, an incorrect encoding) or some value inconsistent with other extensions.</li> <li>The certificate cannot be used for the specified purpose.</li> <li>The Basic Constraints path length parameter has been exceeded.</li> </ul> <p>For more information, see <a href="#">RFC 5280, section 4.2.1.9</a>.</p> <ul style="list-style-type: none"> <li>The certificate's value for Not Before or Not After is invalid. These dates can be encoded as UTCTime or GeneralizedTime</li> </ul> <p>For more information, see <a href="#">RFC 5280 section 4.1.2.5</a>.</p> <ul style="list-style-type: none"> <li>The format of the name constraint is not recognized; for example, an email address format of a form not mentioned in <a href="#">RFC 5280, section 4.2.1.10</a>. This could be caused by an improper extension or some new feature not currently supported.</li> </ul> <p>An unsupported name constraint type was encountered. OpenSSL currently supports only directory name, DNS name, email, and URI types.</p> <ul style="list-style-type: none"> <li>The root certificate authority is not trusted for the specified purpose.</li> <li>The root certificate authority rejects the specified purpose.</li> </ul>	<p>The certificate is valid. All of the following are true:</p> <ul style="list-style-type: none"> <li>Valid certificate extension.</li> <li>The certificate can be used for the specified purpose.</li> <li>Valid Basic Constraints path length.</li> <li>Valid values for Not Before and Not After.</li> <li>Valid name constraint.</li> <li>The root certificate is trusted for the specified purpose.</li> <li>The root certificate accepts the specified purpose.</li> </ul>

Status Check	Status Set to Yes	Status Set to No
Invalid CRL	<p>The <a href="#">Certificate Revocation List (CRL)</a> digital signature is not valid. At least one of the following is true:</p> <ul style="list-style-type: none"> <li>• The value of the CRL's Next Update or Last Update field is invalid.</li> <li>• The CRL is not yet valid.</li> <li>• The CRL has expired.</li> <li>• An error occurred when attempting to verify the CRL path. This error occurs only if extended CRL checking is enabled.</li> <li>• CRL could not be found.</li> <li>• The only CRLs that could be found did not match the scope of the certificate.</li> </ul>	<p>The CRL is valid. All of the following are true:</p> <ul style="list-style-type: none"> <li>• Next Update and Last Update fields are valid.</li> <li>• The CRL's date is valid.</li> <li>• The path is valid.</li> <li>• The CRL was found.</li> <li>• The CRL matches the certificate's scope.</li> </ul>
Server mismatch	<p>The server name does not match the server's <a href="#">Server Name Indication (SNI)</a> name, which could indicate an attempt to spoof the server name.</p>	<p>The server name matches the SNI name of the server to which the client is requesting access.</p>

Note that even though a certificate might match more than one status, the rule causes an action to be taken on the traffic only once.

Checking whether a CA issued or revoked a certificate requires uploading root and intermediate CA certificates and associated CRLs as objects. You then add these trusted CA objects to an SSL policy's list of trusted CA certificates.

## Trusting External Certificate Authorities

You can trust CAs by adding root and intermediate CA certificates to your SSL policy, then use these trusted CAs to verify server certificates used to encrypt traffic.

If a trusted CA certificate contains an uploaded certificate revocation list (CRL), you can also verify whether a trusted CA revoked the encryption certificate.

**Step 1** In the SSL rule editor, select **Trusted CA Certificates**.

**Step 2** Find the trusted CAs you want to add from the **Available Trusted CAs**, as follows:

- To add a trusted CA object on the fly, which you can then add to the condition, click **Add (+)** above the **Available Trusted CAs** list.
- To search for trusted CA objects and groups to add, click the **Search by name or value** prompt above the **Available Trusted CAs** list, then type either the name of the object, or a value in the object. The list updates as you type to display matching objects.

**Step 3** To select an object, click it. To select all objects, right-click and then select **Select All**.

**Step 4** Click **Add to Rule**.

**Tip** You can also drag and drop selected objects.

**Step 5** Add or continue editing the rule.

### What to do next

- Add a certificate status TLS/SSL rule condition to your SSL rule. See [Matching Traffic on Certificate Status, on page 13](#) for more information.
- Deploy configuration changes; see [Deploy Configuration Changes](#).

### Related Topics

[Trusted Certificate Authority Objects](#)

## Trusted External Certificate Authority Configuration

Verified server certificates include certificates signed by trusted CAs. After you add trusted CA certificates to the SSL policy, you can configure a TLS/SSL rule with certificate status conditions to match against this traffic.



**Tip** Upload all certificates within a root CA's chain of trust to the list of trusted CA certificates, including the root CA certificate and all intermediate CA certificates. Otherwise, it is more difficult to detect trusted certificates issued by intermediate CAs. Also, if you configure certificate status conditions to trust traffic based on the root issuer CA, all traffic within a trusted CA's chain of trust can be allowed without decryption, rather than unnecessarily decrypting it.

When you create an SSL policy, the system populates the Trusted CA Certificates tab with a default Trusted CA object group, Cisco Trusted Authorities.

You can modify individual entries in the group, and choose whether to include this group in your SSL policy. You cannot delete the group. System updates can modify the entries on this list, but user changes are preserved.

## Matching Traffic on Certificate Status

### Before you begin

- Add a trusted CA object or group to your SSL policy. See [Trusting External Certificate Authorities, on page 12](#) for more information.

**Step 1** In the Firepower Management Center, choose **Policies > Access Control > SSL**.

**Step 2** Add a new policy or edit an existing policy.

**Step 3** Add a new TLS/SSL rule or edit an existing rule.

**Step 4** In the Add Rule or Editing Rule dialog box, choose **Cert Status**.

**Step 5** For each certificate status, you have the following options:

- Choose **Yes** to match against the presence of that certificate status.

- Choose **No** to match against the absence of that certificate status.
- Choose **Any** to skip the condition when matching the rule. In other words, choosing **Any** means the rule matches whether the certificate status is present or absent.

## Step 6 Add or continue editing the rule.

### Example

The organization trusts the Verified Authority certificate authority. The organization does not trust the Spammer Authority certificate authority. The system administrator uploads the Verified Authority certificate and an intermediate CA certificate issued by Verified Authority to the system. Because Verified Authority revoked a certificate it previously issued, the system administrator uploads the CRL that Verified Authority provided.

The following figure shows a certificate status rule condition checking for valid certificates, those issued by a Verified Authority, are not on the CRL, and still within the Valid From and Valid To date. Because of the configuration, traffic encrypted with these certificates is not decrypted and inspected with access control.

Revoked:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Self Signed:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Signature:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid Issuer:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Expired:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Not Yet Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Certificate:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid CRL:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Server Mismatch:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any

The following figure shows a certificate status rule condition checking for the absence of a status. In this case, because of the configuration, it matches against traffic encrypted with a certificate that has not expired and monitors that traffic.

Revoked:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Self Signed:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Signature:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid Issuer:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Expired:	<input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Any
Not Yet Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Certificate:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid CRL:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Server Mismatch:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any

The following graphic illustrates a certificate status rule condition that matches on the presence or absence of several statuses. Because of the configuration, if the rule matches incoming traffic encrypted with a certificate issued by an invalid user, self-signed, invalid, or expired, it decrypts the traffic with a known key.

Revoked:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Self Signed:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Signature:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid Issuer:	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Any	Expired:	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Any
Not Yet Valid:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Invalid Certificate:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any
Invalid CRL:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any	Server Mismatch:	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Any

The following graphic illustrates a certificate status rule condition that matches if the SNI of the request matches the server name or if the CRL is not valid. Because of the configuration, if the rule matches either condition, traffic is blocked.

Revoked:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>	Self Signed:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>
Valid:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>	Invalid Signature:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>
Invalid Issuer:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>	Expired:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>
Not Yet Valid:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>	Invalid Certificate:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>
Invalid CRL:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>	Server Mismatch:	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Any"/>

### What to do next

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

## Cipher Suite TLS/SSL Rule Conditions

The system provides predefined cipher suites you can add to a cipher suite rule condition. You can also add cipher suite list objects containing multiple cipher suites.



**Note** You cannot add new cipher suites. You can neither modify nor delete predefined cipher suites.

You can add a maximum of 50 cipher suites and cipher suite lists to the **Selected Cipher Suites** in a single cipher suite condition. The system supports adding the following cipher suites to a cipher suite condition:

- SSL\_RSA\_FIPS\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_FIPS\_WITH\_DES\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_DHE\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA256
- TLS\_DHE\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA
- TLS\_DHE\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA256
- TLS\_DHE\_RSA\_WITH\_DES\_CBC\_SHA
- TLS\_DH\_Annon\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_DH\_Annon\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_DH\_Annon\_WITH\_CAMELLIA\_128\_CBC\_SHA
- TLS\_DH\_anon\_WITH\_CAMELLIA\_128\_CBC\_SHA256

- TLS\_DH\_Anon\_WITH\_CAMELLIA\_256\_CBC\_SHA
- TLS\_DH\_anon\_WITH\_CAMELLIA\_256\_CBC\_SHA256
- TLS\_ECDHE\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_ECDSA\_WITH\_NULL\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_RC4\_128\_SHA
- TLS\_ECDHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_NULL\_SHA
- TLS\_ECDHE\_RSA\_WITH\_RC4\_128\_SHA
- TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA
- TLS\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA256
- TLS\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA
- TLS\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA256
- TLS\_RSA\_WITH\_DES\_CBC\_SHA



- TLS\_RSA\_WITH\_NULL\_MD5
- TLS\_RSA\_WITH\_NULL\_SHA
- TLS\_RSA\_WITH\_RC4\_128\_MD5
- TLS\_RSA\_WITH\_RC4\_128\_SHA

Note the following:

- If you add cipher suites not supported for your deployment, you cannot deploy your configuration. For example, passive deployments do not support decrypting traffic with the any of the ephemeral Diffie-Hellman (DHE) or ephemeral elliptic curve Diffie-Hellman (ECDHE) cipher suites. Creating a rule with these cipher suites prevents you from deploying your access control policy.
- If you configure a cipher suite condition with a cipher suite, any external certificate objects you add to a certificate condition, or internal CA objects you associate with the **Decrypt - Resign** action, must match the cipher suite's signature algorithm type. For example, if your rule's cipher suite condition references an EC-based cipher suite, any server certificates you add, or CA certificates you associate with the **Decrypt - Resign** action, must also be EC-based. If you mismatch signature algorithm types in this case, the policy editor displays a warning icon next to the rule.
- You can add an anonymous cipher suite to the **Cipher Suite** condition in an SSL rule, but keep in mind:
  - 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](#).
  - You cannot use either the **Decrypt - Resign** or **Decrypt - Known Key** action in the rule, because the system cannot decrypt traffic encrypted with an anonymous cipher suite.
- When specifying a cipher suite as a rule condition, consider that the rule matches on the negotiated cipher suite in the ServerHello message, rather than on the full list of cipher suites specified in the ClientHello message. During ClientHello processing, the managed device strips unsupported cipher suites from the ClientHello message. However, if this results in all specified cipher suites being stripped, the system retains the original list. If the system retains unsupported cipher suites, subsequent evaluation results in an undecrypted session.

## Controlling Encrypted Traffic by Cipher Suite

**Step 1** In the SSL rule editor, select Cipher Suite.

**Step 2** Find the cipher suites you want to add from the **Available Cipher Suites**, as follows;

- To add a cipher suite list on the fly, which you can then add to the condition, click **Add** (+) above the **Available Cipher Suites** list.
- To search for cipher suites and lists to add, click the **Search by name or value** prompt above the **Available Cipher Suites** list, then type either the name of the cipher suite, or a value in the cipher suite. The list updates as you type to display matching cipher suites.

**Step 3** To select a cipher suite, click it. To select all cipher suites, right-click and then select **Select All**.

**Step 4** Click **Add to Rule**.

**Tip** You can also drag and drop selected cipher suites.

**Step 5** Add or continue editing the rule.

---

**What to do next**

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

**Related Topics**

[Cipher Suite Lists](#)

## Encryption Protocol Version TLS/SSL Rule Conditions

You can choose to match against traffic encrypted with SSL version 3.0, or TLS version 1.0, 1.1, or 1.2. By default, all protocol versions are selected when you create a rule; if you select multiple versions, encrypted traffic that matches any of the selected versions matches the rule. You must select at least one protocol version when saving the rule condition.

You cannot select SSL v2.0 in a version rule condition; the system does not support decrypting traffic encrypted with SSL version 2.0. You can configure an undecryptable action to allow or block this traffic without further inspection.

## Controlling Traffic by Encryption Protocol Version

---

**Step 1** In the SSL rule editor, select Version.

**Step 2** Select the protocol versions you want to match against.

**Step 3** Add or continue editing the rule.

---

**What to do next**

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