Decryption Rules and Policy Example

This chapter builds on concepts discussed in this guide to provide a specific example of an SSL policy with decryption rules that follow our best practices and recommendations. You should be able to apply this example to your situation, adapting it to the needs of your organization.

In short:

- For trusted traffic (such as transferring a large compressed server backup), bypass inspection entirely, using prefiltering and flow offload.
- Put *first* any decryption rules that can be evaluated quickly, such as those that apply to specific IP addresses.
- Put *last* any decryption rules that require processing, **Decrypt - Resign**, and rules that block unsecure protocol versions and cipher suites.

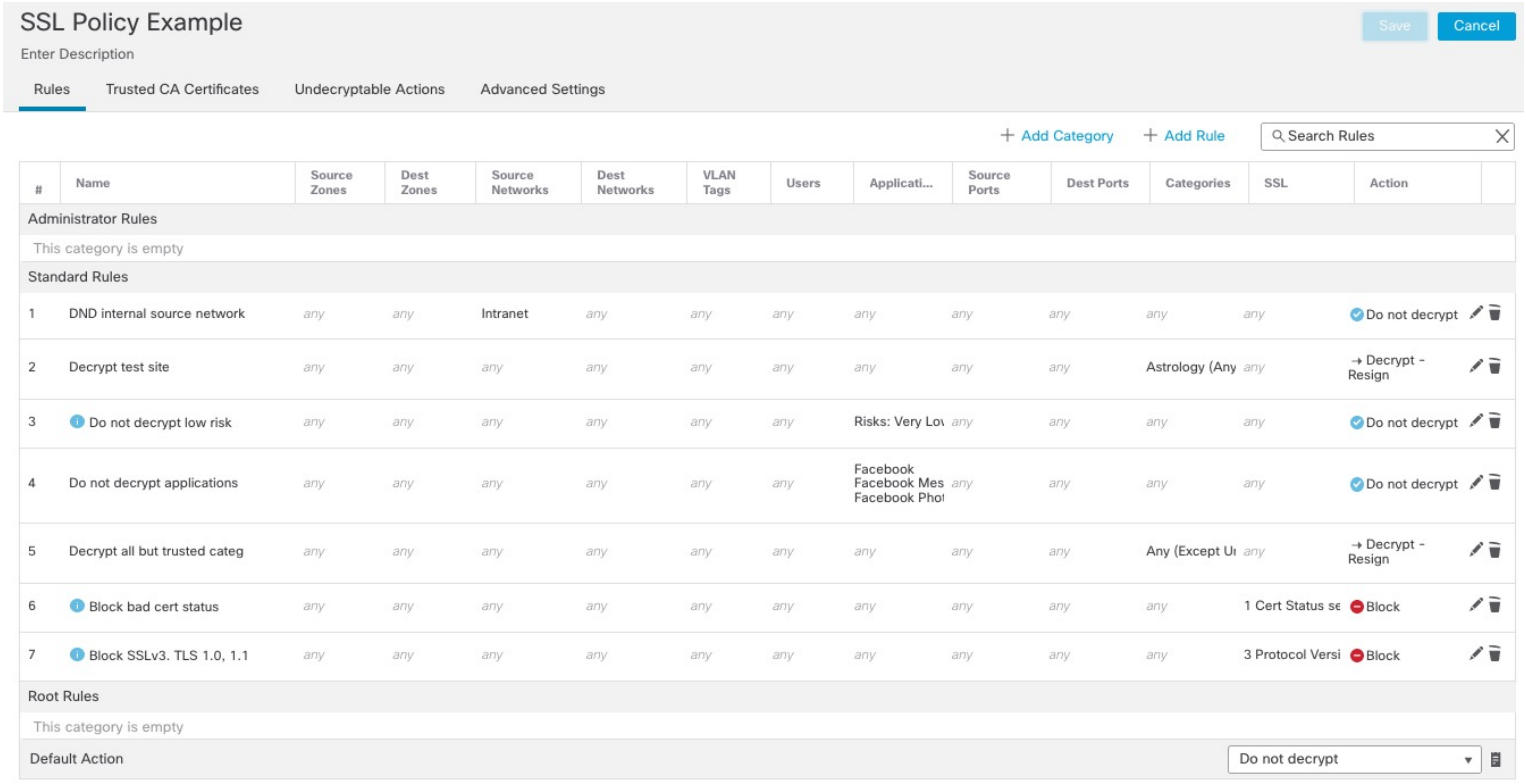
Decryption Rules Best Practices

This chapter provides an example decryption policy with decryption rules that illustrates our best practices and recommendations. First we'll discuss settings for the decryption policies and access control policies and then walk through all the rules and why we recommend they be ordered in a particular way.

Some general guidelines:

- Decrypting traffic requires processing and memory; decrypting too much traffic can impact performance. Before you set up decryption policies and rules, see When to Decrypt Traffic, When Not to Decrypt.
- Among the types of traffic you should exclude from decryption is traffic that is by nature undecryptable; typically, undecryptable traffic uses TLS/SSL certificate pinning. .

Following are the decryption rules we'll discuss in this chapter.



Bypass Inspection with Prefilter and Flow Offload

Prefiltering is the first phase of access control, before the system performs more resource-intensive evaluation. Prefiltering is simple, fast, and early. Prefiltering uses limited outer-header criteria to quickly handle traffic. Compare this to subsequent evaluation, which uses inner headers and has more robust inspection capabilities.

Configure prefiltering to:

- Improve performance— The sooner you exclude traffic that does not require inspection, the better. You can fastpath or block certain types of plaintext, passthrough tunnels based on their outer encapsulation headers, without inspecting their encapsulated connections. You can also fastpath or block any other connections that benefit from early handling.
- Tailor deep inspection to encapsulated traffic—You can rezone certain types of tunnels so that you can later handle their encapsulated connections using the same inspection criteria. Rezoning is necessary because after prefiltering, access control uses inner headers.

If you have a Firepower 4100/9300 or Secure Firewall 3100/4200 available, you can use *large flow offload*, a technique where trusted traffic can bypass the inspection engine for better performance. You can use it, for example, in a data center to transfer server backups.

Related Topics

Large Flow Offloads
Prefiltering vs Access Control
Best Practices for Fastpath Prefiltering

Do Not Decrypt Best Practices

Log traffic during evaluation period

Do Not Decrypt rules generally should disable logging but if you're not sure what traffic matches your rules, you can temporarily enable logging. After you confirm the correct traffic is being matched, disable logging for those rules.

Guidelines for undecryptable traffic

We can determine that certain traffic is not decryptable either because the website itself is not decryptable or because the website uses TLS/SSL pinning, which effectively prevents users from accessing a decrypted site without errors in their browser.

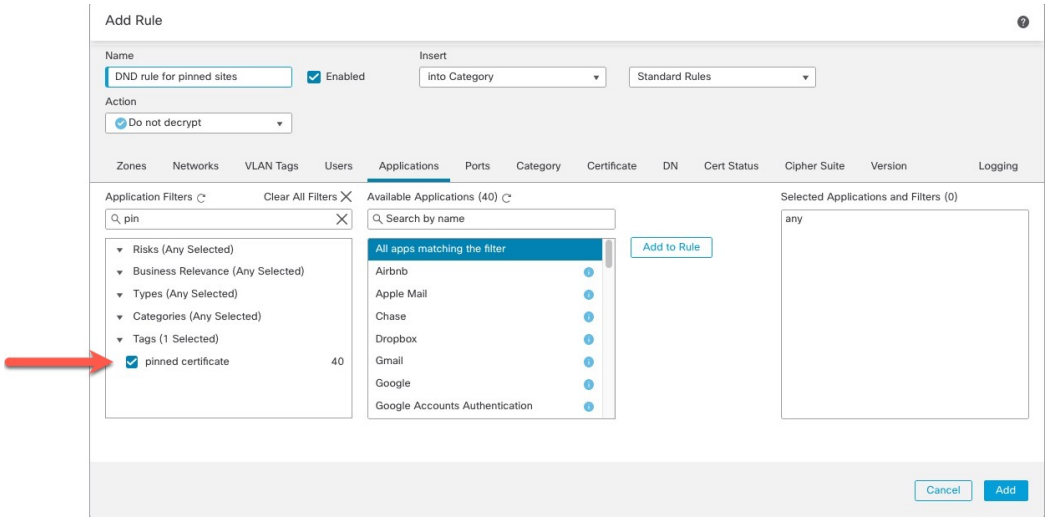
For more information about certificate pinning, see About TLS/SSL Pinning.

We maintain the list of these sites as follows:

- A Distinguished Name (DN) group named **Cisco-Undecryptable-Sites**
- The **pinned certificate** or **undecryptable** application filter

If you are decrypting traffic and you do not want users to see errors in their browsers when going to these sites, we recommend you set up a **Do Not Decrypt** rule toward the bottom of your decryption rules.

An example of setting up a **pinned certificate** application filter follows.



Decrypt - Resign and Decrypt - Known Key Best Practices

This topic discusses best practices for **Decrypt - Resign** and **Decrypt - Known Key** decryption rule.

Do not use Version or Cipher Suite rule conditions

Important *Never* use either **Cipher Suite** or **Version** rule conditions in a rule with a **Decrypt - Resign** or **Decrypt - Known Key** rule action. The use of these conditions in rules with other rule actions can interfere with the system's ClientHello processing, resulting in unpredictable performance.

Decrypt - Resign best practices with certificate pinning

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 decryption 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. We recommend adding a Do Not Decrypt rule before the **Decrypt - Resign** rule so pinning traffic is excluded from being decrypted.

For more information about certificate pinning, see About TLS/SSL Pinning.

Decrypt - Known Key best practices

Because a **Decrypt - Known Key** rule action is intended to be used for traffic going to an internal server, you should always add either a destination network to the TBD rule rules (**Networks** rule condition) or add a security zone to the access control rule (**Zones** tab page). That way the traffic goes directly to the network or interface on which the server is located, thereby reducing traffic on the network.

Decryption Rules to Put First

Put first any rules that can be matched by the first part of the packet; an example is a rule that references IP addresses (**Networks** rule condition).

Decryption Rules to Put Last

Rules with the following rule conditions should be ordered immediately be last because those rules require traffic to be examined for the longest amount of time by the system:

- Applications
- Category
- Certificate
- Distinguished Name (DN)
- Cert Status
- Cipher Suite
- Version

Recommended Policy and Rule Settings

We recommend the following policy settings:

- Decryption policy:
  - Default action **Do Not Decrypt**.
  - Enable logging.
  - Set **Undecryptable Actions** to **Block** for both **SSL v2 Session** and **Compressed Session**.
  - Enable TLS 1.3 decryption in the policy's advanced settings.
- Decryption rules: Enable logging for every rule except those with a **Do Not Decrypt** rule action. (It's up to you; if you want to see information about traffic that isn't decrypted, enable logging for those rules also.)
- Access control policy:
  - Associate your decryption policy with an access control policy. (If you fail to do this, your decryption policy and rules have no effect.)
  - Set the default policy action to **Intrusion Prevention: Balanced Security and Connectivity**.
  - Enable logging.

Related Topics

Decryption Policy Settings, on page 6
Decryption Rule Settings, on page 26
Access Control Policy Settings, on page 8

Recommended Policy and Rule Settings

We recommend the following policy settings:

- Decryption policy:
  - Default action **Do Not Decrypt**.
  - Enable logging.
  - Set **Undecryptable Actions** to **Block** for both **SSL v2 Session** and **Compressed Session**.
  - Enable TLS 1.3 decryption in the policy's advanced settings.
- Decryption rules: Enable logging for every rule except those with a **Do Not Decrypt** rule action. (It's up to you; if you want to see information about traffic that isn't decrypted, enable logging for those rules also.)
- Access control policy:
  - Associate your decryption policy with an access control policy. (If you fail to do this, your decryption policy and rules have no effect.)

- Set the default policy action to **Intrusion Prevention: Balanced Security and Connectivity**.
- Enable logging.

Related Topics

Decryption Policy Settings, on page 6
Decryption Rule Settings, on page 26
Access Control Policy Settings, on page 8

Decryption Policy Settings

How to configure recommended the following best practice settings for your decryption policy:

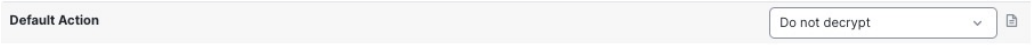
- Default action **Do Not Decrypt**.
- Enable logging.
- Set **Undecryptable Actions** to **Block** for both **SSL v2 Session** and **Compressed Session**.
- Enable TLS 1.3 decryption in the policy's advanced settings.

Procedure

Step 1 Click **Policies** > **Access Control** > **Decryption**.

Step 2 Click **Edit** ( ) next to your decryption policy.

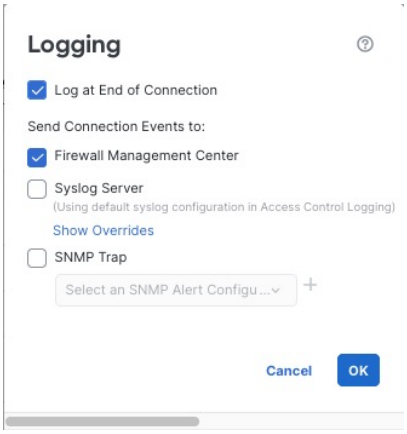
Step 3 From the **Default Action** list at the bottom of the page, click **Do Not Decrypt**.
The following figure shows an example.



Step 4 At the end of the row, click **Logging** ( ).

Step 5 Select the **Log at End of Connection** check box.

The following figure shows an example.



Step 6 Click **OK**.

Step 7 Click **Save**.

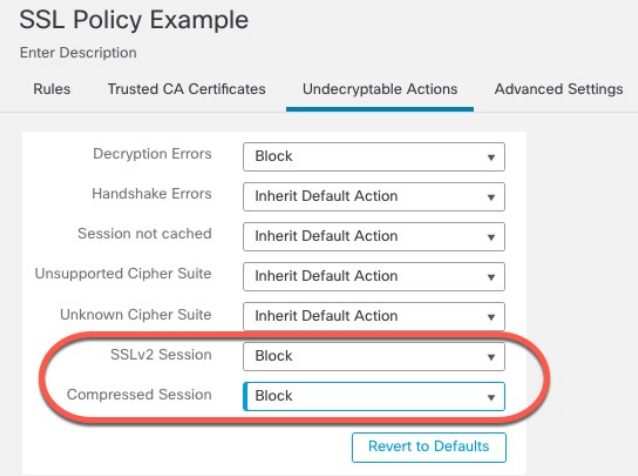
Step 8 Click the **Undecryptable Actions** tab.

Step 9 We recommend setting the action for **SSLv2 Session** and **Compressed Session** to **Block**.

You shouldn't allow SSL v2 on your network and compressed TLS/SSL traffic is not supported so you should block that traffic as well.

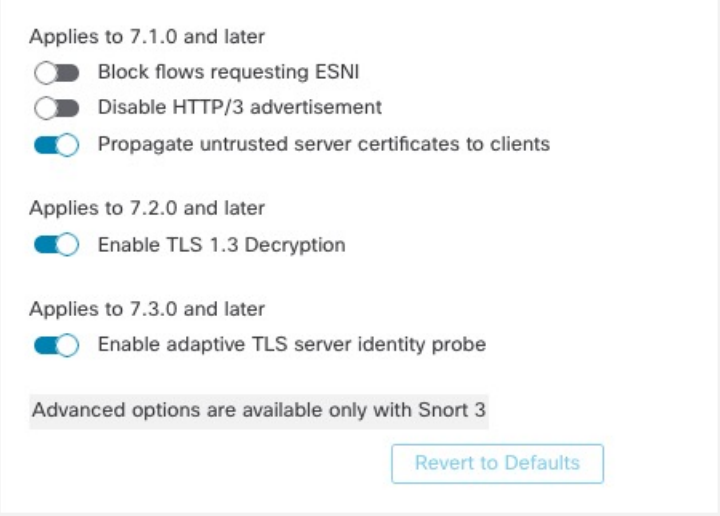
See Default Handling Options for Undecryptable Traffic for more information about setting each option.

The following figure shows an example.



Step 10 Click the **Advanced Settings** tab page.

Step 11 Select the **Enable TLS 1.3 Decryption** check box. For more information about the other options, see Decryption Policy Advanced Options.



Step 12 At the top of the page, click **Save**.

What to do next

Configure decryption rules and set each one as discussed in Decryption Rule Settings, on page 26.

Access Control Policy Settings

How to configure recommended the following best practice settings for your access control policy:

- Associate your decryption policy with an access control policy. (If you fail to do this, your decryption policy and rules have no effect.)
- Set the default policy action to **Intrusion Prevention: Balanced Security and Connectivity**.
- Enable logging.

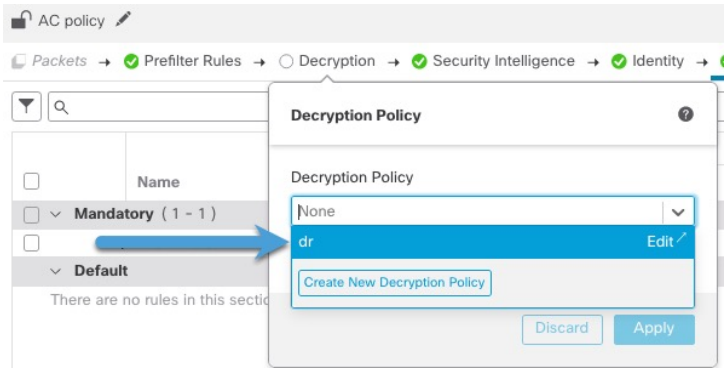
Procedure

Step 1 Click **Policies** > **Access Control**.

Step 2 Click **Edit** ( ) next to your access control policy.

Step 3 (If your decryption policy is not set up yet, you can do this later.)

a) Click the **Decryption** link at the top of the page as the following figure shows.

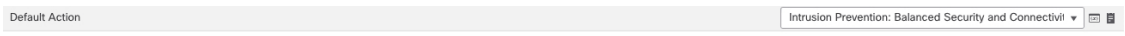


b) From the list, click the name of your decryption policy.

c) Click **Apply**.

d) At the top of the page, click **Save**.

Step 4 From the **Default Action** list at the bottom of the page, click **Intrusion Prevention: Balanced Security and Connectivity**.
The following figure shows an example.



Step 5 Click **Logging** ( ).

Step 6 Select the **Log at End of Connection** check box and click **OK**.

Step 7 Click **Save**.

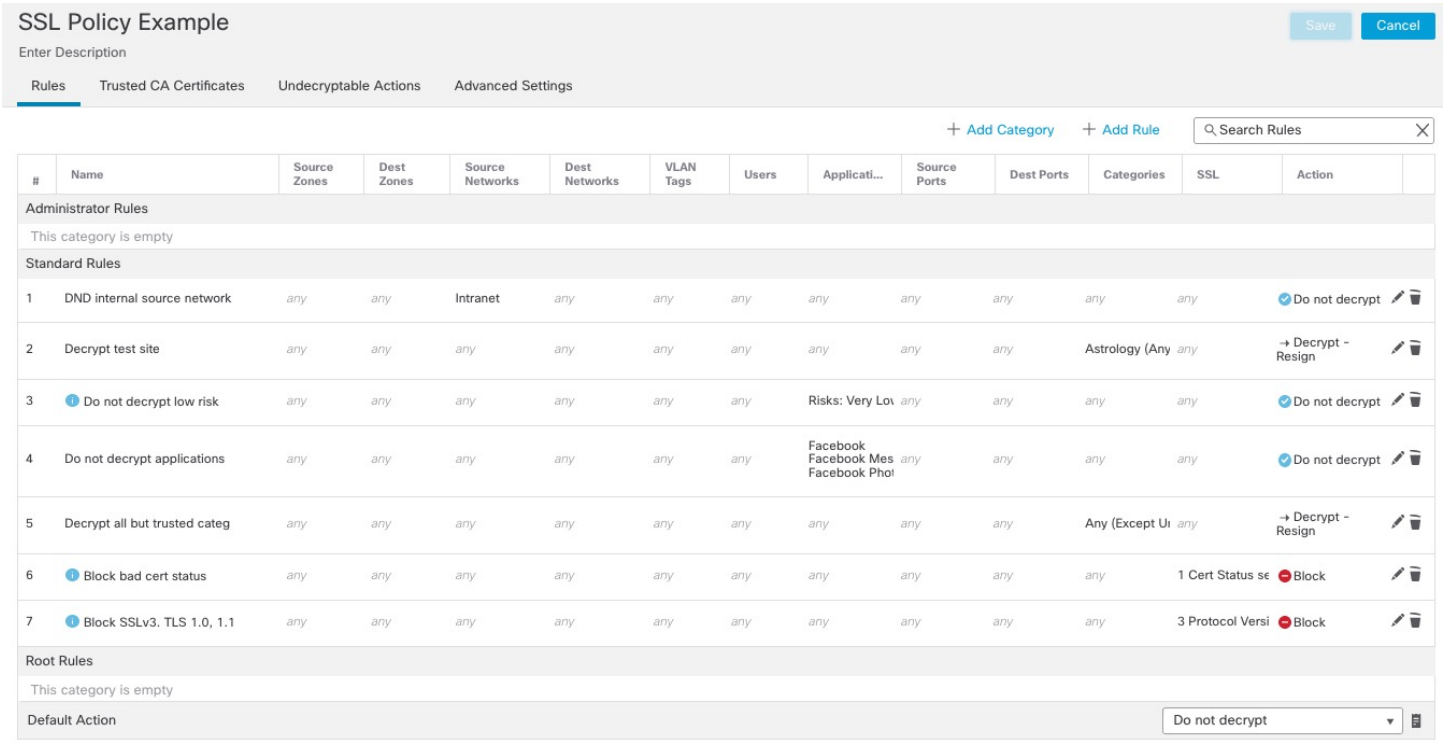
What to do next

See Decryption Rule Examples, on page 13.

Decryption Policy Walkthrough

This chapter provides a step-by-step discussion and walkthrough of how to create a decryption policy using rules that employ our best practices. You'll see a preview of the decryption policy followed by a synopsis of the best practices and finally a discussion of the rules in the policy.

Following is the decryption policy we'll discuss in this chapter.



See one of the following sections for more information.

Related Topics

Recommended Policy and Rule Settings, on page 5
Traffic to Prefilter, on page 13
First Decryption Rule: Do Not Decrypt Specific Traffic, on page 14
Next Decryption Rules: Decrypt Specific Test Traffic, on page 15
Create a Decrypt - Resign Rule for Categories, on page 18
Do Not Decrypt Low-Risk Categories, Reputations, or Applications, on page 16
Decryption Rules: Block or Monitor Certificates and Protocol Versions, on page 19

Recommended Policy and Rule Settings

We recommend the following policy settings:

- Decryption policy:
  - Default action **Do Not Decrypt**.
  - Enable logging.
  - Set **Undecryptable Actions** to **Block** for both **SSL v2 Session** and **Compressed Session**.
  - Enable TLS 1.3 decryption in the policy's advanced settings.
- Decryption rules: Enable logging for every rule except those with a **Do Not Decrypt** rule action. (It's up to you; if you want to see information about traffic that isn't decrypted, enable logging for those rules also.)
- Access control policy:
  - Associate your decryption policy with an access control policy. (If you fail to do this, your decryption policy and rules have no effect.)
  - Set the default policy action to **Intrusion Prevention: Balanced Security and Connectivity**.
  - Enable logging.

Related Topics

Decryption Policy Settings, on page 6
Decryption Rule Settings, on page 26
Access Control Policy Settings, on page 8

Decryption Policy Settings

How to configure recommended the following best practice settings for your decryption policy:

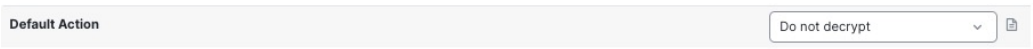
- Default action **Do Not Decrypt**.
- Enable logging.
- Set **Undecryptable Actions** to **Block** for both **SSL v2 Session** and **Compressed Session**.
- Enable TLS 1.3 decryption in the policy's advanced settings.

Procedure

Step 1 Click **Policies** > **Access Control** > **Decryption**.

Step 2 Click **Edit** ( ) next to your decryption policy.

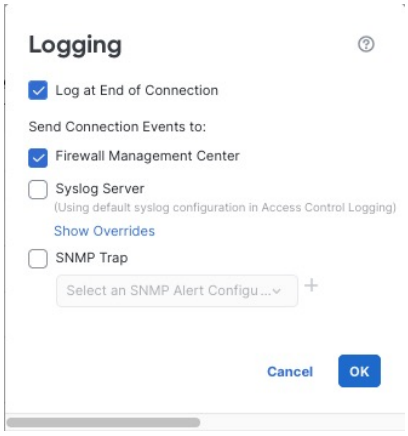
Step 3 From the **Default Action** list at the bottom of the page, click **Do Not Decrypt**.
The following figure shows an example.



Step 4 At the end of the row, click **Logging** ( ).

Step 5 Select the **Log at End of Connection** check box.

The following figure shows an example.



Step 6 Click **OK**.

Step 7 Click **Save**.

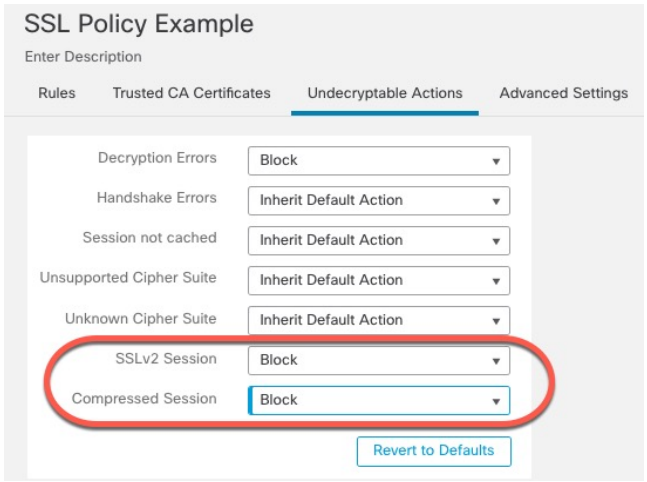
Step 8 Click the **Undecryptable Actions** tab.

Step 9 We recommend setting the action for **SSLv2 Session** and **Compressed Session** to **Block**.

You shouldn't allow SSL v2 on your network and compressed TLS/SSL traffic is not supported so you should block that traffic as well.

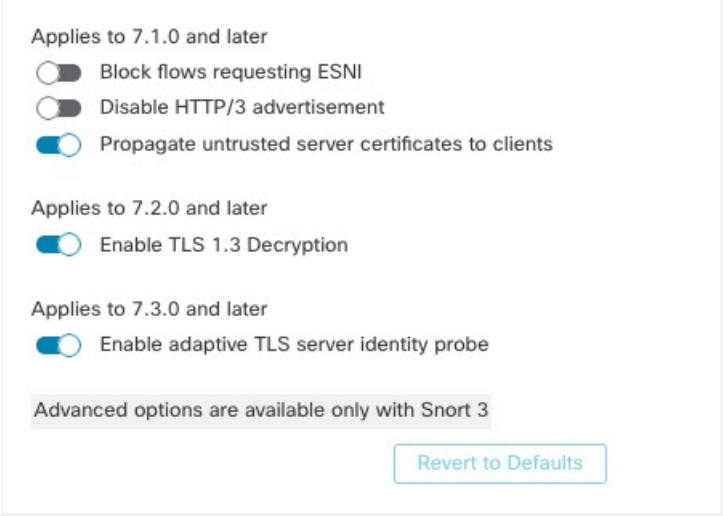
See Default Handling Options for Undecryptable Traffic for more information about setting each option.

The following figure shows an example.



Step 10 Click the **Advanced Settings** tab page.

Step 11 Select the **Enable TLS 1.3 Decryption** check box. For more information about the other options, see Decryption Policy Advanced Options.



Step 12 At the top of the page, click **Save**.

What to do next

Configure decryption rules and set each one as discussed in Decryption Rule Settings, on page 26.

Access Control Policy Settings

How to configure recommended the following best practice settings for your access control policy:

- Associate your decryption policy with an access control policy. (If you fail to do this, your decryption policy and rules have no effect.)
- Set the default policy action to **Intrusion Prevention: Balanced Security and Connectivity**.
- Enable logging.

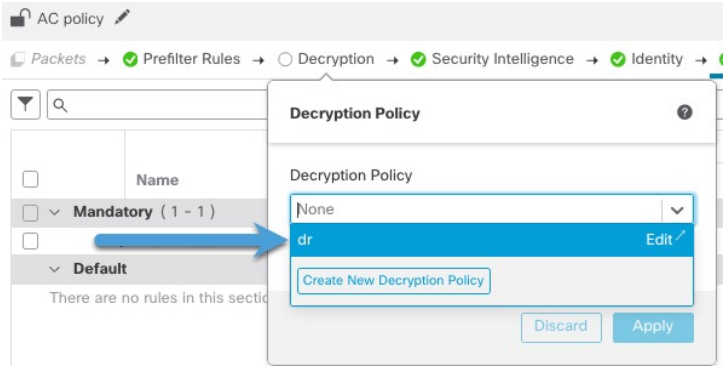
Procedure

Step 1 Click **Policies** > **Access Control**.

Step 2 Click **Edit** (✎) next to your access control policy.

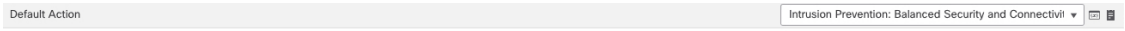
Step 3 (If your decryption policy is not set up yet, you can do this later.)

a) Click the **Decryption** link at the top of the page as the following figure shows.



b) From the list, click the name of your decryption policy.
c) Click **Apply**.
d) At the top of the page, click **Save**.

Step 4 From the **Default Action** list at the bottom of the page, click **Intrusion Prevention: Balanced Security and Connectivity**.
The following figure shows an example.



Step 5 Click **Logging** ( ).

Step 6 Select the **Log at End of Connection** check box and click **OK**.

Step 7 Click **Save**.

What to do next

See Decryption Rule Examples, on page 13.

Decryption Rule Examples

This section provides an example of decryption rule that illustrate our best practices.

See one of the following sections for more information.

Related Topics

Traffic to Prefilter

Prefiltering is the first phase of access control, before the system performs more resource-intensive evaluation. Prefiltering is simple, fast, and early compared to subsequent evaluation, which uses inner headers and has more robust inspection capabilities.

Based on your security needs and traffic profile, you should consider prefiltering and therefore excluding from any policy and inspection the following:

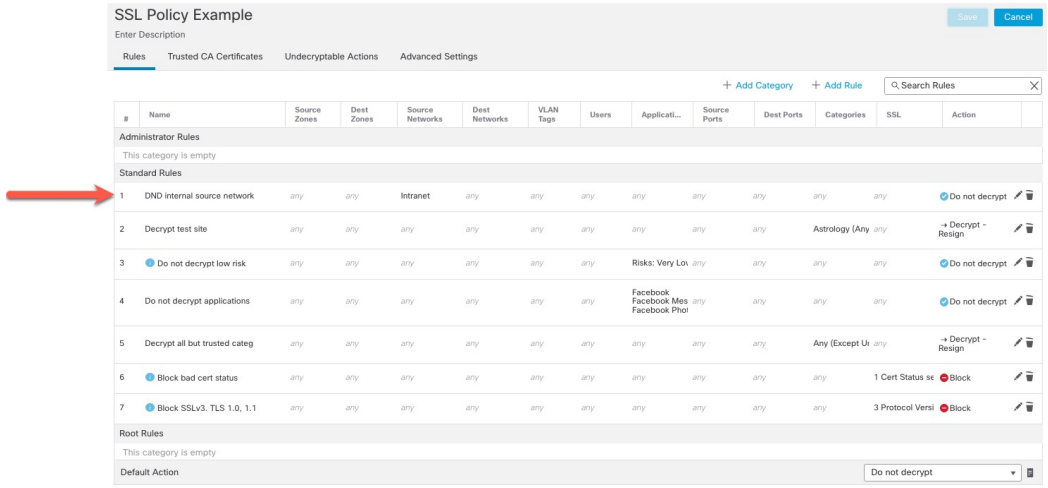
- Common intraoffice applications such as Microsoft Outlook 365
- Elephant flows, such as server backups

Related Topics

Prefiltering vs Access Control
Best Practices for Fastpath Prefiltering

First Decryption Rule: Do Not Decrypt Specific Traffic

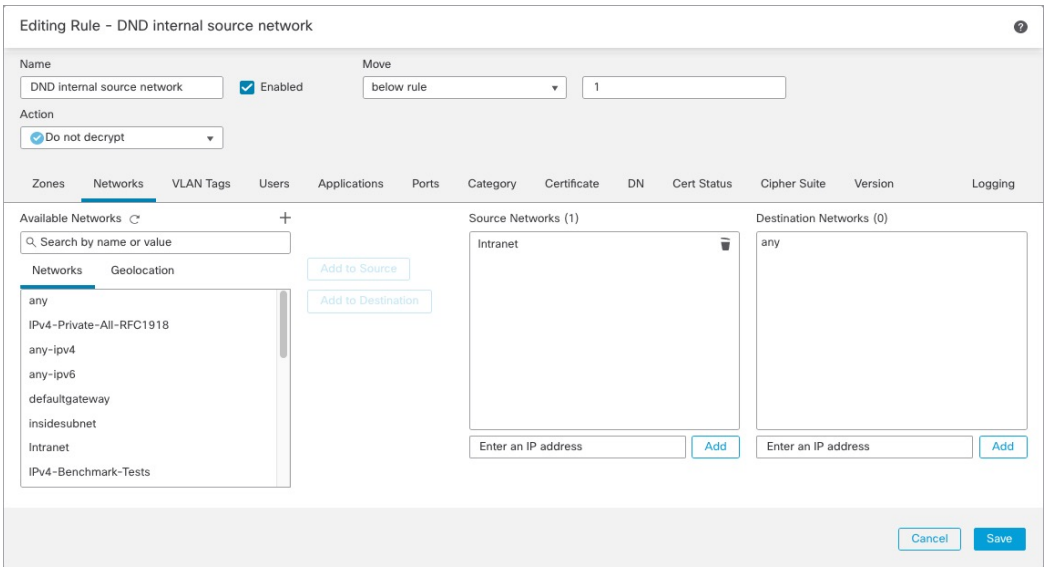
The first decryption rule in the example does not decrypt traffic that goes to an internal network (defined as **intranet**). **Do Not Decrypt** rule actions are matched during ClientHello so they are processed very fast.



Note If you have traffic going from internal DNS servers to internal DNS resolvers (such as Cisco Umbrella Virtual Appliances), you can add **Do Not Decrypt** rules for them as well. You can even add those to prefiltering policies if the internal DNS servers do their own logging.

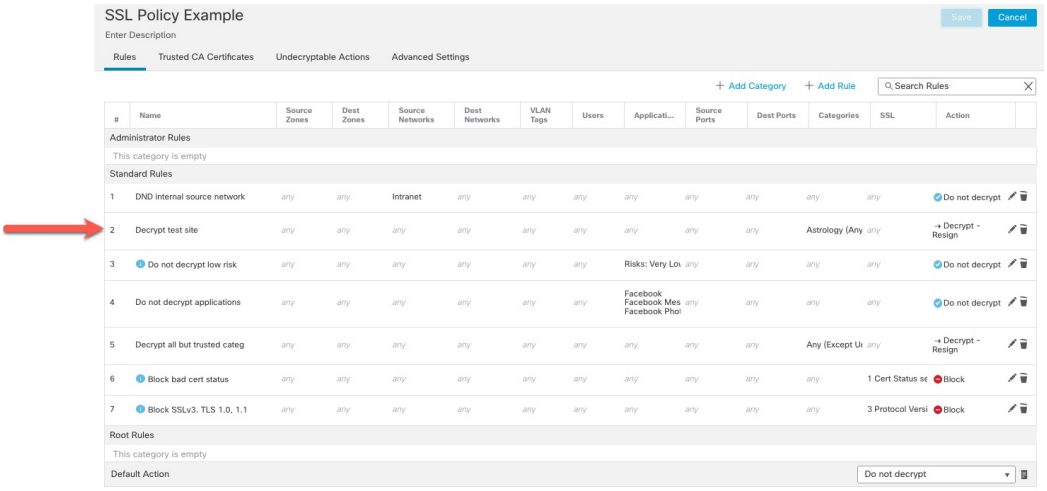
However, we strongly recommend you *do not* use **Do Not Decrypt** rules or prefiltering for DNS traffic that goes to the internet, such as internet root servers (for example, Microsoft internal DNS resolvers built into Active Directory). In those cases, you should fully inspect the traffic or even consider blocking it.

Rule detail:

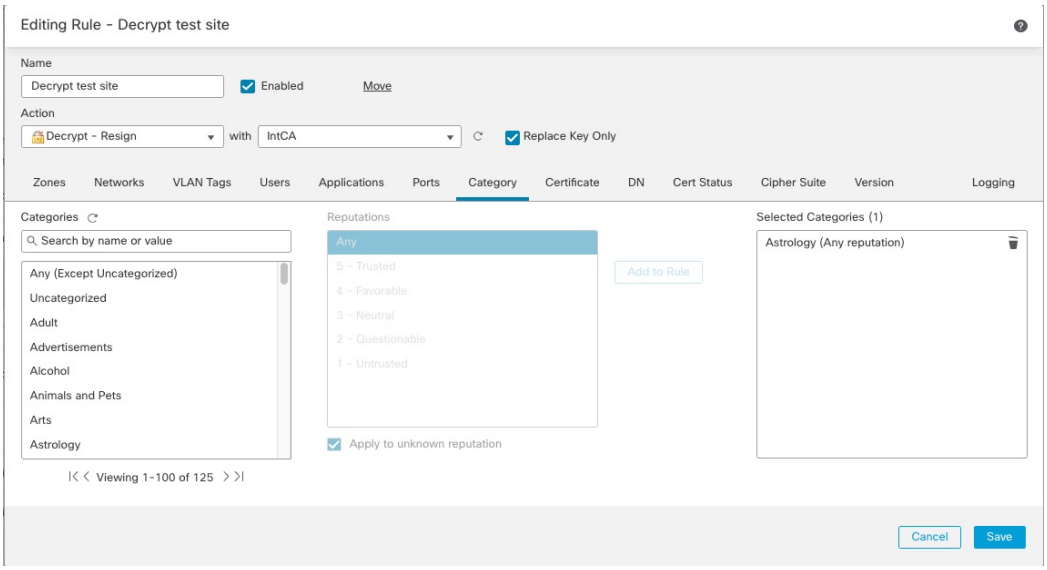


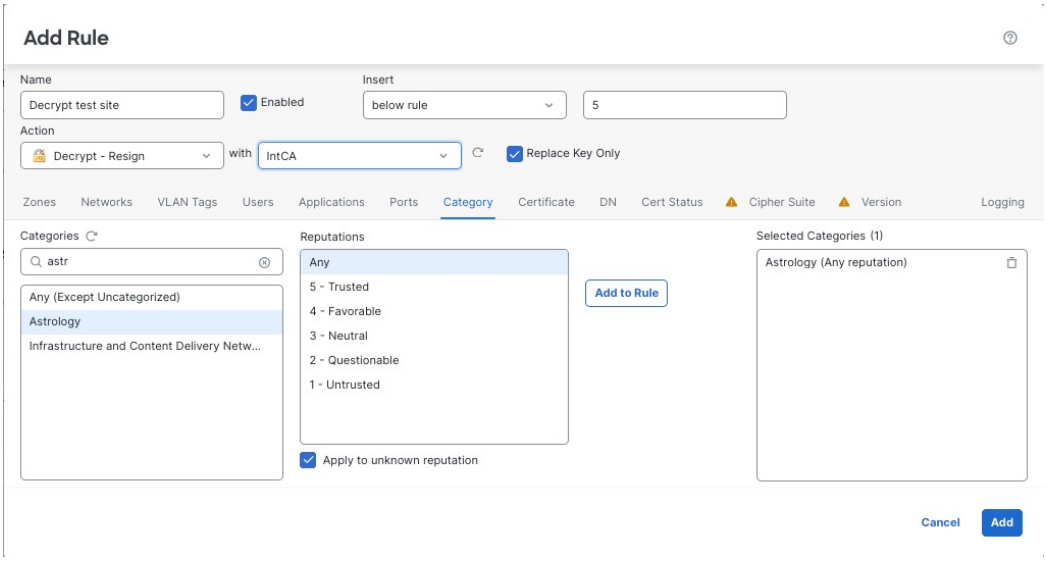
Next Decryption Rules: Decrypt Specific Test Traffic

The next rule is *optional* in the example; use it to decrypt and monitor limited types of traffic before determining whether or not to allow it on your network.



Rule detail:

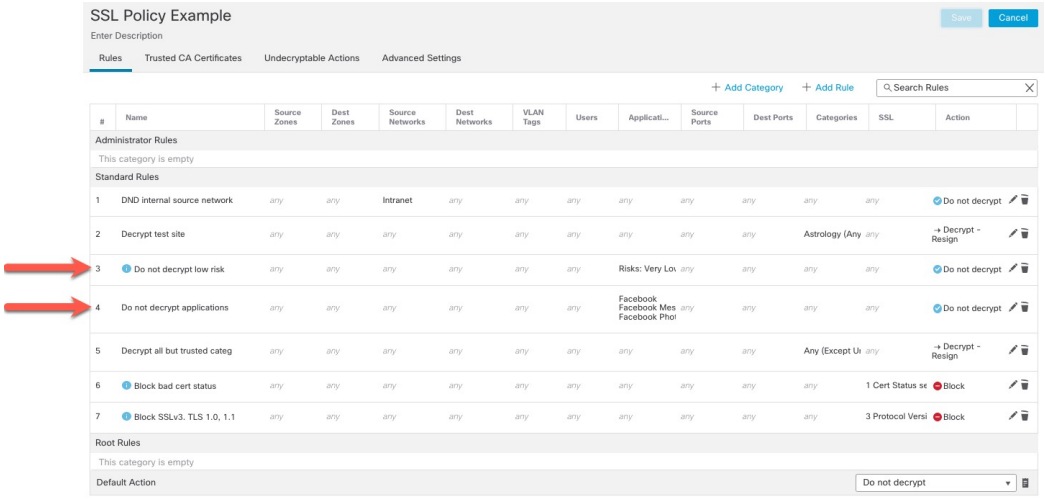




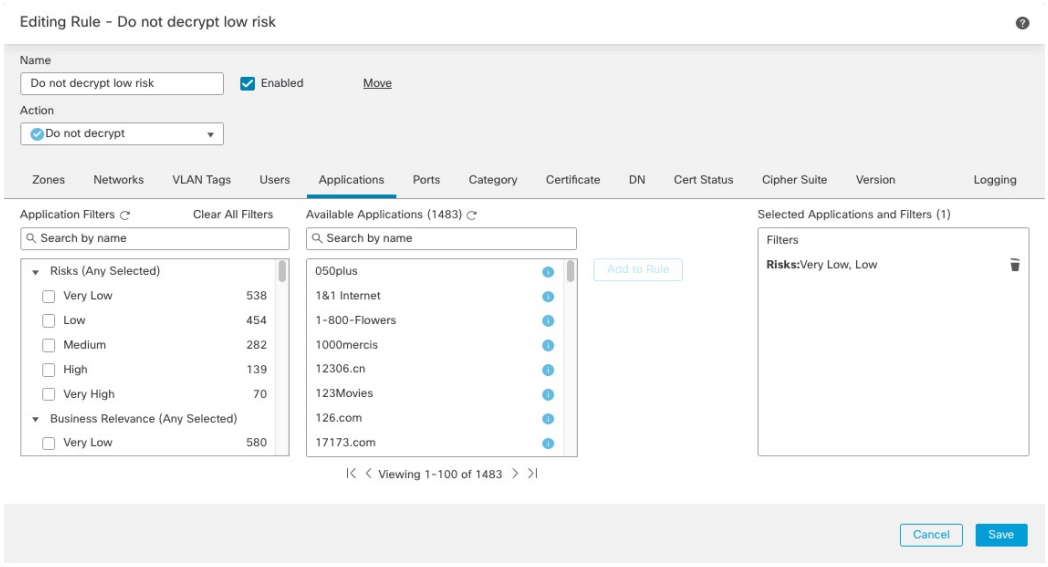
Do Not Decrypt Low-Risk Categories, Reputations, or Applications

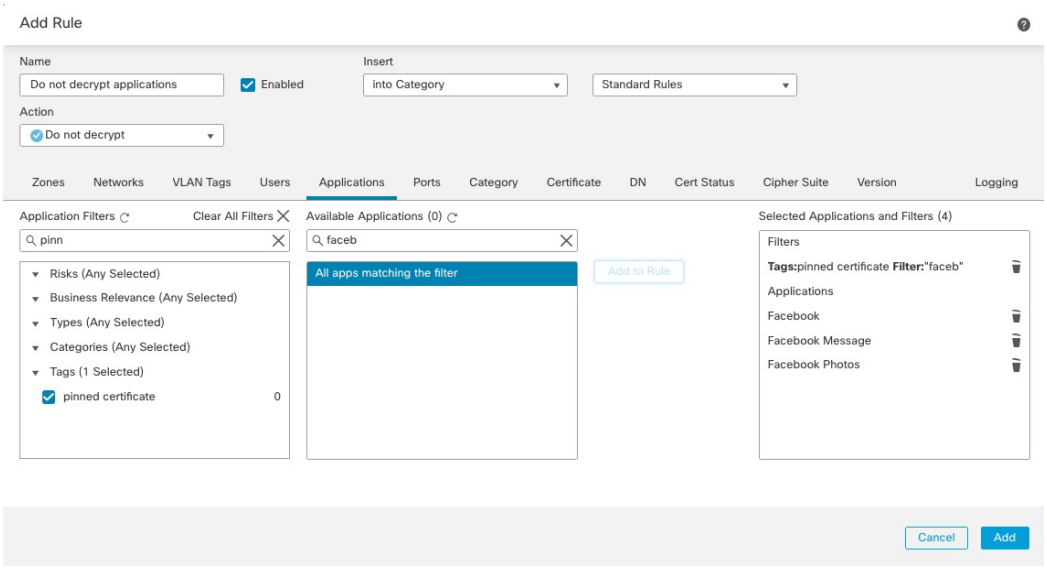
Evaluate the traffic on your network to determine which would match low-risk categories, reputations, or applications, and add those rules with a **Do Not Decrypt** action. Put these rules *after* other more specific **Do Not Decrypt** rules because the system needs more time to process the traffic.

Following is the example.



Rule details:





Related Topics

Best Practices for Configuring Application Control
Recommendations for Application Control

Create a Decrypt - Resign Rule for Categories

This topic shows an example of creating a decryption rule with a **Decrypt - Resign** action for all but uncategorized sites. The rule uses the optional **Replace Key Only** option, which we always recommend with a **Decrypt-Resign** rule action.

Replace Key Only causes the user to see a security warning in the web browser when they browse to a site that uses a self-signed certificate, making the user aware that they are communicating with an unsecure site.

By putting this rule near the bottom, you get the best of both worlds: you can decrypt and optionally inspect traffic while not affecting performance as much as if you had put the rule earlier in the policy.

Procedure

Step 1 If you haven't already done so, upload an internal certificate authority (CA) to the Secure Firewall Management Center (**Objects** > **Object Management**, then **PKI** > **Internal CAs**).

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

Step 3 Click **Edit** ( ) next to your decryption policy.

Step 4 Click **Add Rule**.

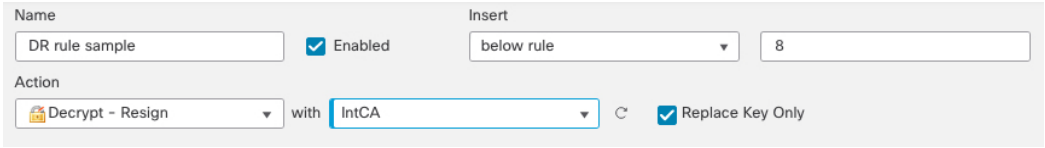
Step 5 In the **Name** field, enter a name to identify the rule.

Step 6 From the **Action** list, click **Decrypt - Resign**.

Step 7 From the **with** list, click the name of your internal CA.

Step 8 Check the **Replace Key Only** box.

The following figure shows an example.



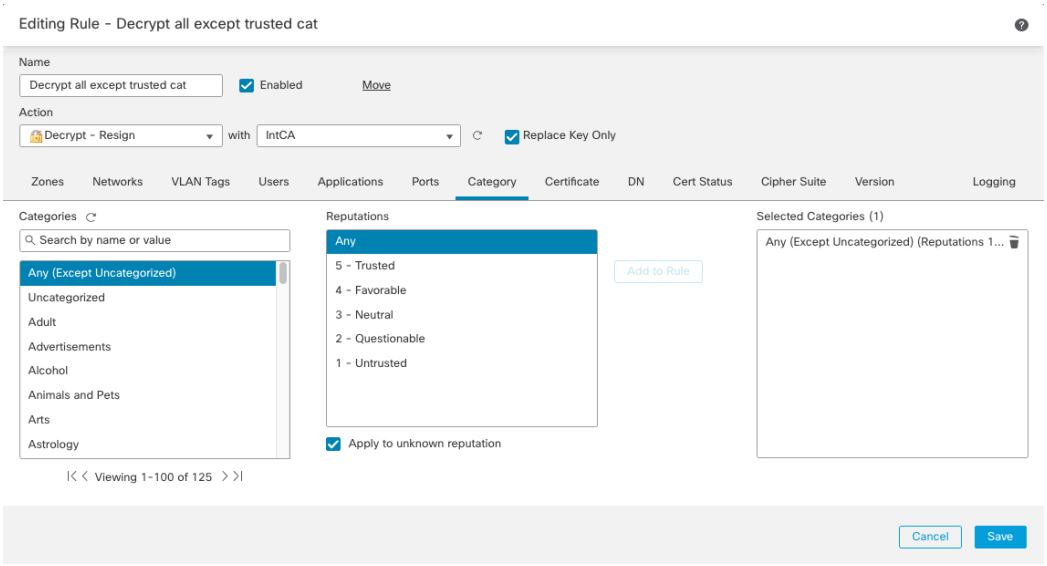
Step 9 Click the **Category** tab page.

Step 10 From the top of the **Categories** list, click **Any (Except Uncategorized)**.

Step 11 From the **Reputations** list, click **Any**.

Step 12 Click **Add to Rule**.

The following figure shows an example.

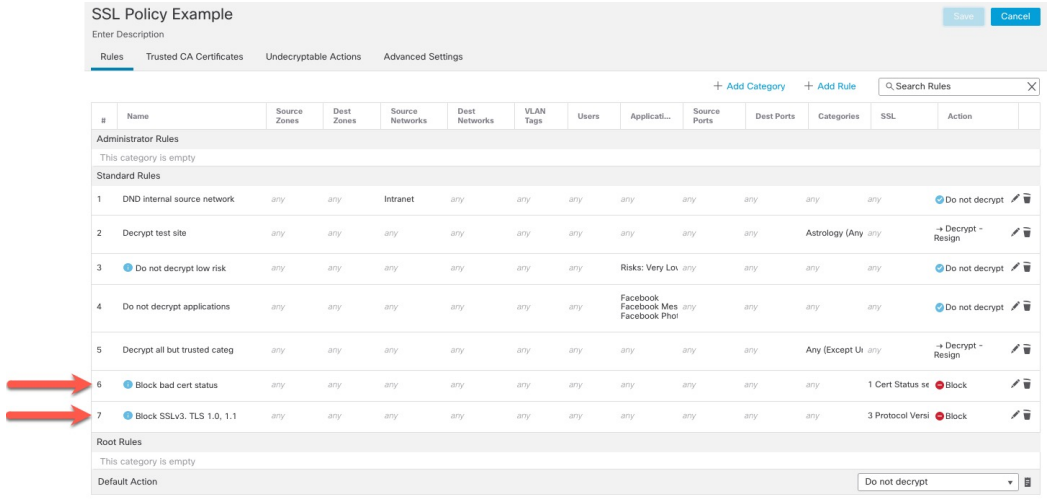


Related Topics

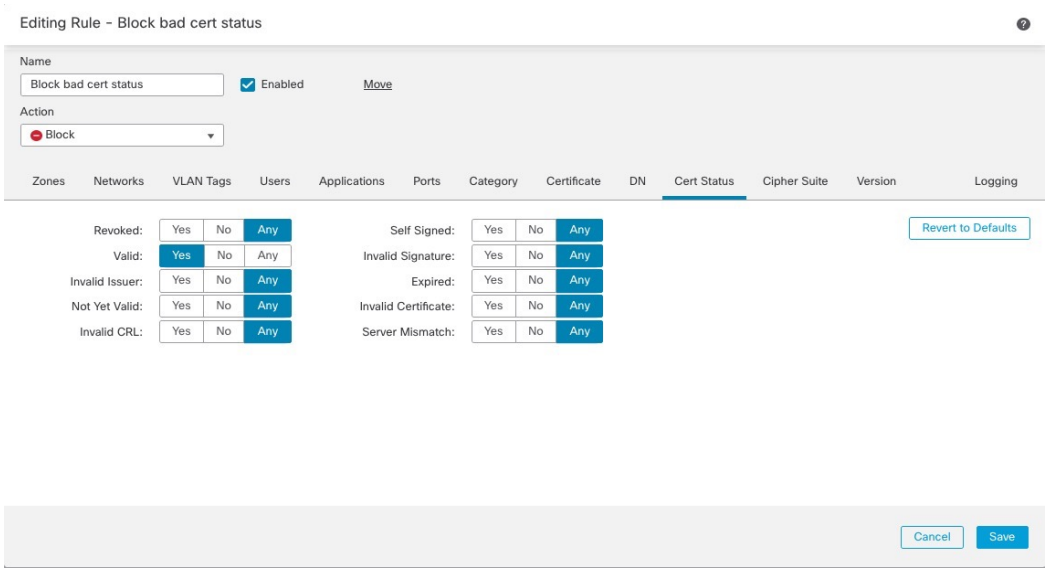
Internal Certificate Authority Objects

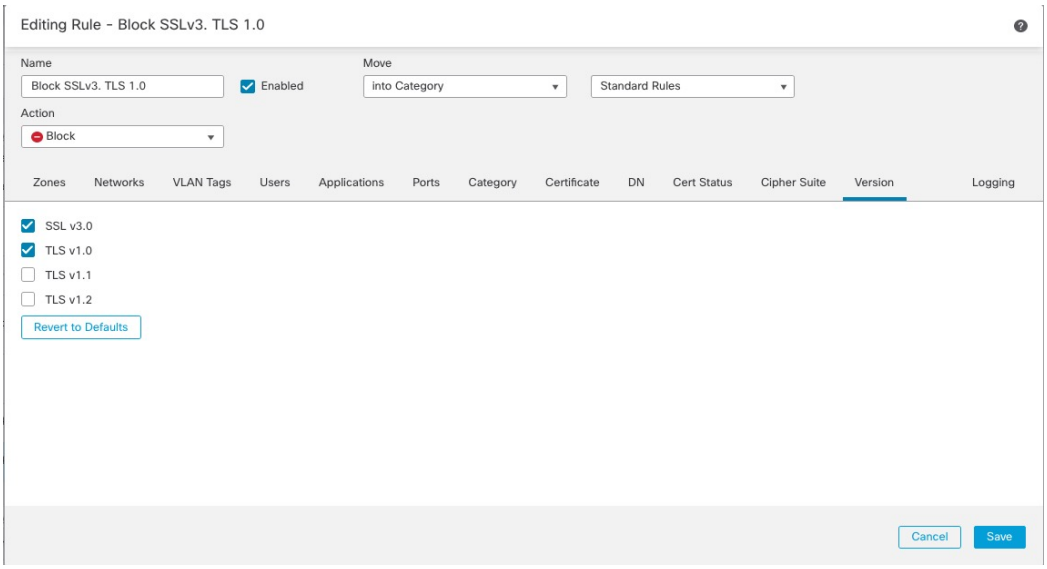
Decryption Rules: Block or Monitor Certificates and Protocol Versions

The last decryption rules, because they are the most specific and require the most processing, are rules that either monitor or block bad certificates and unsecure protocol versions.



Rule details:





Related Topics

Example: Decryption Rule to Monitor or Block Certificate Status, on page 21
Example: Decryption Rule to Monitor or Block Protocol Versions, on page 23
Optional Example: Manual Decryption Rule to Monitor or Block Certificate Distinguished Name, on page 24

Example: Decryption Rule to Monitor or Block Certificate Status

The last decryption rules, because they are the most specific and require the most processing, are rules that either monitor or block bad certificates and unsecure protocol versions. The example in this section shows how to monitor or block traffic by certificate status.

Important 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.

Procedure

Step 1 Log in to the Secure Firewall Management Center if you haven't already done so.

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

Step 3 Click **Edit** (✎) next to your decryption policy.

Step 4 Click **Edit** (✎) next to a decryption rule.

Step 5 Click **Add Rule**.

Step 6 n the Add Rule dialog box, in the **Name** field, enter a name for the rule.

Step 7 Click **Cert Status**.

Step 8 For each certificate status, you have the following options:

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

- Click **No** to match against the *absence* of that certificate status.
- Click **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 9 From the **Action** list, click either **Monitor** to only monitor and log traffic that matches the rule or click **Block** or **Block with Reset** to block the traffic and optionally reset the connection.

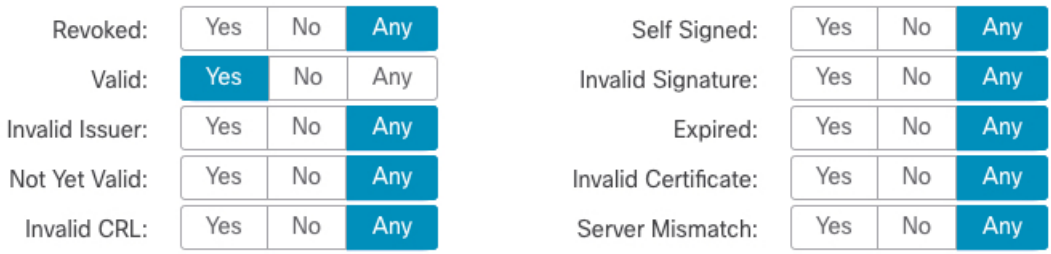
Step 10 To save changes to the rule, at the bottom of the page, click **Add**.

Step 11 To save changes to the policy, at the top of the page, click **Save**.

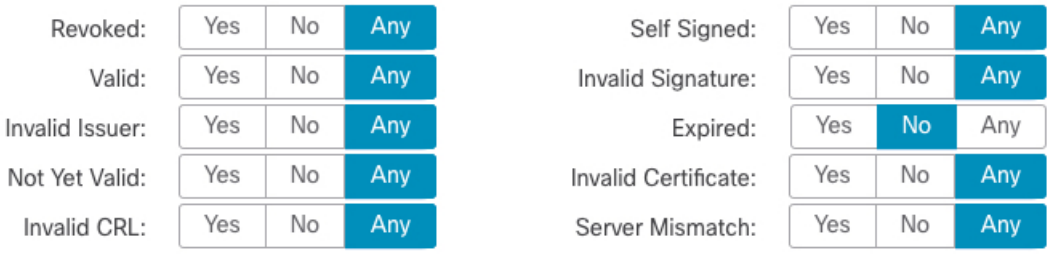
Example

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.

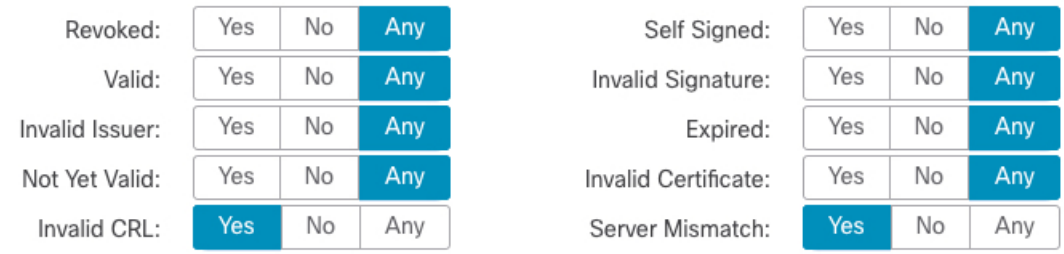


The following figure shows a certificate status rule condition checking for the absence of a status. In this case, because of the configuration, it matches against traffic encrypted with a certificate that has not expired.



In the following example, traffic would match this rule condition if the incoming traffic is using a certificate that has an invalid issuer, is self-signed, expired, and it is an invalid certificate.

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.



Example: Decryption Rule to Monitor or Block Protocol Versions

This example shows how to block TLS and SSL protocols on your network that are no longer considered secure, such as TLS 1.0, TLS 1.1, and SSLv3. It's included to give you a little more detail about how protocol version rules work.

You should exclude nonsecure protocols from your network because they are all exploitable. In this example:

- You can block some protocols using **Version** page on the decryption rule.
- Because the system considers SSLv2 as undecryptable, you can block it using the **Undecryptable Actions** on the decryption policy.
- Similarly, because compressed TLS/SSL is not supported, you should block it as well.

Important 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.

Procedure

Step 1 Click **Policies** > **Access Control** > **Decryption**.

Step 2 Click **Edit** ( ) next to your decryption policy.

Step 3 Click **Edit** ( ) next to a decryption rule.

Step 4 Click **Add Rule**.

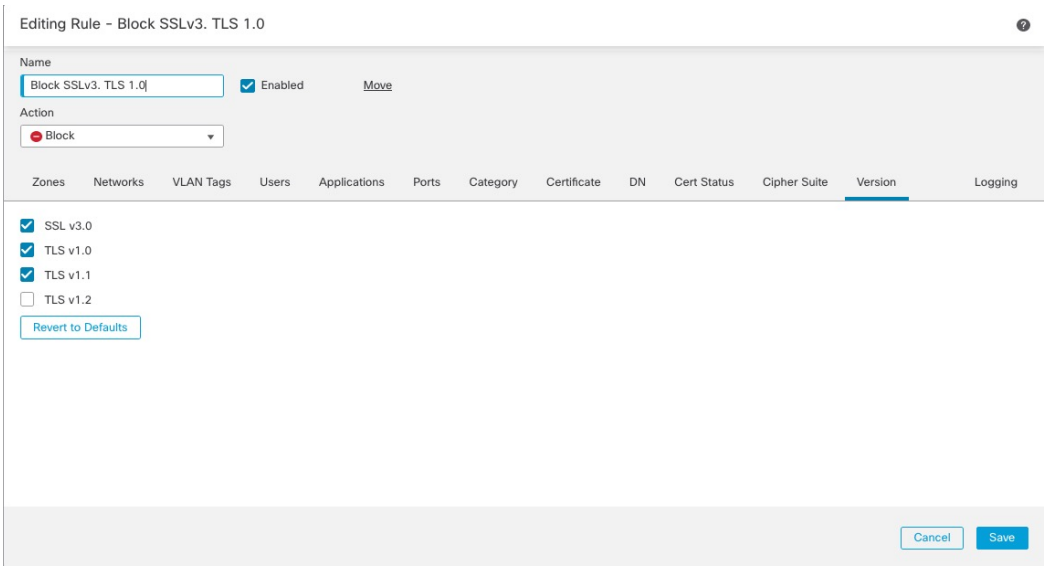
Step 5 In the Add Rule dialog box, in the **Name** field, enter a name for the rule.

Step 6 From the **Action** list, click **Block** or **Block with reset**.

Step 7 Click **Version** page.

Step 8 Check the check boxes for protocols that are no longer secure, such as **SSL v3.0**, **TLS 1.0**, and **TLS 1.1**. Clear the check boxes for any protocols that are still considered secure.

The following figure shows an example.



Step 9 Choose other rule conditions as needed.

Step 10 Click **Add**.

Optional Example: Manual Decryption Rule to Monitor or Block Certificate Distinguished Name

This rule is included to give you an idea about how to monitor or block traffic based on the server certificate's distinguishedname. It's included to give you a little more detail.

The distinguished name can consist of country code, common name, organization, and organizational unit, but typically consists of a common name only. For example, the common name in the certificate for `https://www.cisco.com` is `cisco.com`. (However, it's not always this simple; Distinguished Name (DN) Rule Conditions shows how to find common names.)

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.

Procedure

Step 1 Click **Policies** > **Access Control** > **Decryption**.

Step 2 Click **Edit** ( ) next to your decryption policy.

Step 3 Click **Edit** ( ) next to a decryption rule.

Step 4 Click **Add Rule**.

Step 5 In the Add Rule dialog box, in the **Name** field, enter a name for the rule.

Step 6 From the **Action** list, click **Block** or **Block with reset**.

Step 7 Click **DN**.

Step 8 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 9 To select an object, click it. To select all objects, right-click and then **Select All**.

Step 10 Click **Add to Subject** or **Add to Issuer**.

Tip You can also drag and drop selected objects.

Step 11 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**.

Although you can add a CN or DN to either list, it's more common to add them to the **Subject DNs** list.

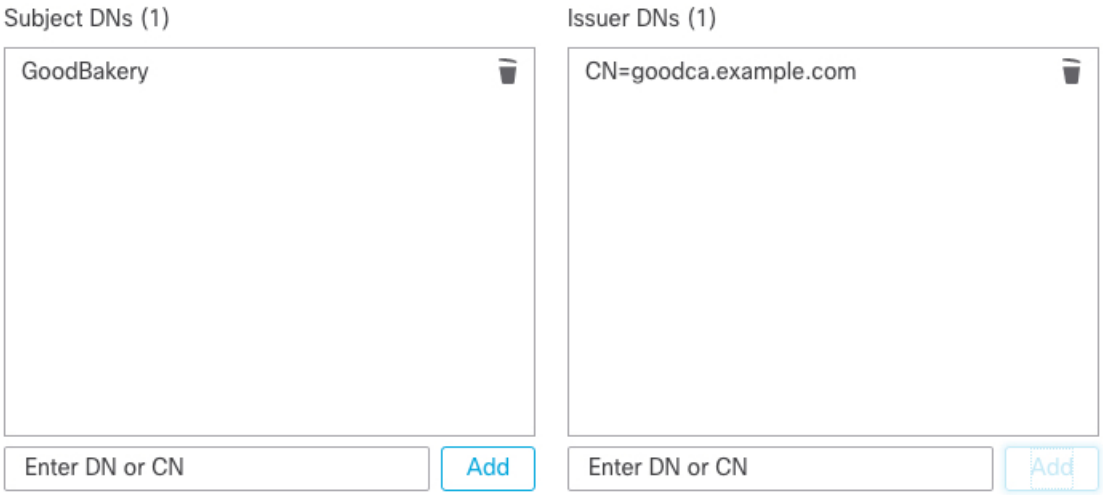
Step 12 Add or continue editing the rule.

Step 13 When you're done, to save changes to the rule, click **Add** at the bottom of the page.

Step 14 To save changes to the policy, click **Save** at the top of the page.

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.



Decryption Rule Settings

How to configure recommended best practice settings for your decryption rules.

Decryption rules: Enable logging for every rule except those with a **Do Not Decrypt** rule action. (It's up to you; if you want to see information about traffic that isn't decrypted, enable logging for those rules also.)

Procedure

Step 1 Click **Policies** > **Access Control** > **Decryption**.

Step 2 Click **Edit** (✎) next to your decryption policy.

Step 3 Click **Edit** (✎) next to a decryption rule.

Step 4 Click the **Logging** tab.

Step 5 Click **Log at End of Connection**.

Step 6 Click **Save**.

Step 7 Click **Save** at the top of the page.