



Decryption Policies

The following topics provide an overview of decryption policy creation, configuration, management, and logging.

- [About Decryption Policies, on page 1](#)
- [Requirements and Prerequisites for Decryption Policies, on page 2](#)
- [Create a Decryption Policy, on page 2](#)
- [Decryption Policy Default Actions, on page 10](#)
- [Default Handling Options for Undecryptable Traffic, on page 11](#)
- [Decryption Policy Advanced Options, on page 12](#)
- [Manage Decryption Policies, on page 16](#)

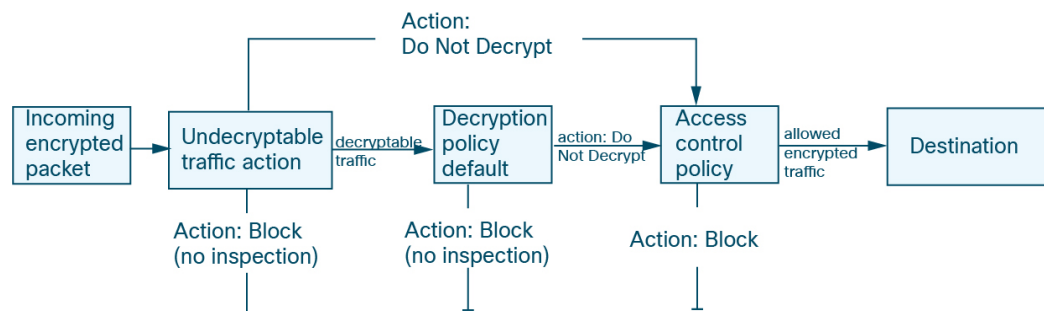
About Decryption Policies

A decryption policy determines how the system handles encrypted traffic on your network. You can configure one or more decryption policies, associate a decryption policy with an access control policy, then deploy the access control policy to a managed device. When the device detects a TCP handshake, the access control policy first handles and inspects the traffic. If it subsequently identifies a TLS/SSL-encrypted session over the TCP connection, the decryption policy takes over, handling and decrypting the encrypted traffic.

You can create multiple rules at the same time, including rules for decrypting incoming traffic (**Decrypt - Known Key** rule action) and outgoing traffic (**Decrypt - Resign** rule action). To create a rule with a **Do Not Decrypt** or other rule action (such as **Block** or **Monitor**), create an empty decryption policy and add the rule afterward.

To get started, see [Create a Decryption Policy, on page 2](#).

Following is an example decryption policy with a **Do Not Decrypt** rule action:



The simplest decryption policy, as shown in the following diagram, directs the device where it is deployed to handle encrypted traffic with a single default action. You can set the default action to block decryptable traffic without further inspection, or to inspect undecrypted decryptable traffic with access control. The system can then either allow or block the encrypted traffic. If the device detects undecryptable traffic, it either blocks the traffic without further inspection or does not decrypt it, inspecting it with access control.

Requirements and Prerequisites for Decryption Policies

Supported Domains

Any

User Roles

- Admin
- Access Admin
- Network Admin

Create a Decryption Policy

This topic discusses how to create a decryption policy and optionally one or more rules to protect internal or external servers. You can also create a decryption policy without rules and add the rules later. Creating an empty policy is a good choice to create rules with a **Do Not Decrypt**, **Block**, **Block With Reset**, or **Monitor** rule actions.

Before you begin

Review your needs for decryption:

- Decryption is a way to expose network traffic to deep inspection; however, there are times you should *not* decrypt traffic: [When to Decrypt Traffic](#), [When Not to Decrypt](#).
- To protect *internal* servers by decrypting and optionally inspecting traffic, you must have the internal certificate for your internal server: [PKI](#).
- To protect *external* servers by decrypting and optionally inspecting traffic, you must upload an internal CA object that will be used to decrypt and resign the traffic: [PKI](#).

Procedure

- Step 1** Click **Policies** > **Access Control** > **Decryption**.
- Step 2** Click **New Policy**.
- Step 3** Enter a name for the policy in the **Name** field and an optional description in the **Description** field.

Create Decryption Policy
? ✕

i A Decryption policy is not required only to perform application or URL discovery; instead, you can use TLS 1.3 Server Identity Discovery on the Access Control policy.

Name*

Description

Outbound Connections (User Protection)
Inbound Connections (Server Protection)

How Outbound Protection Works

Outbound protection matches traffic based on the referenced internal CA certificate's signature algorithm type, in addition to any configured rule conditions.

```

graph LR
    SOURCE((SOURCE)) --> DECRYPT[DECRYPT RE-SIGN]
    DECRYPT --> DESTINATION((DESTINATION))
    DECRYPT --> EXCLUSIONS[DECRIPTION EXCLUSIONS]
    EXCLUSIONS --> SOURCE
    EXCLUSIONS --> DESTINATION
  
```

Internal CA Download

A rule will be auto-created for the selected certificate authority.

No networks/ports associated

[See how to configure](#)

Cancel
Save

The **Outbound Connections** tab page enables you to create **Decrypt - Resign** rules. These rules require an internal certificate that you can either create beforehand (using **Objects > Object Management > PKI > Internal CAs**) or you can create them as part of the outbound connection rule.

the **Inbound Connections** tab page enables you to create **Decrypt - Known Key** rules. These rules require an internal certificate that you can either create beforehand (using **Objects > Object Management > PKI > Internal Certs**) or you can create them as part of the inbound connection rule.

Step 4 Associate the decryption rule with an access control rule as discussed in [Associating Other Policies with Access Control](#).

Step 5

Step 6 Continue with one of the following sections.

What To Do Next

- [Create a Decryption Policy with Outbound Connection Protection, on page 4 \(Decrypt - Resign\)](#)
- [Create a Decryption Policy with Inbound Connection Protection, on page 7 \(Decrypt - Known Key\)](#)
- [Create a Decryption Policy with Other Rule Actions, on page 9](#)

Create a Decryption Policy with Outbound Connection Protection

This task discusses how to create a decryption policy with a rule that protects outbound connections; that is, the destination server is outside your protected network. This type of rule has a **Decrypt - Resign** rule action.

When you create a decryption policy, you can create multiple rules at the same time, including multiple **Decrypt - Known Key** rules and multiple **Decrypt - Resign** rules.

Threat Defense Feature History:

7.4—This feature is added.

Before you begin

You must upload an internal certificate authority (CA) for your outbound server before you can create a decryption policy that protects outbound connections. You can do this in any of the following ways:

- Create an internal CA object by going to **Objects > Object Management > PKI > Internal CAs** and referring to [PKI](#).
- At the time you create this decryption policy.

Procedure

- Step 1** Click **Policies > Access Control > Decryption**.
- Step 2** Click **New Policy**.
- Step 3** Give the policy a unique **Name** and, optionally, a **Description**.
- Step 4** Click the **Outbound Connections** tab.

Create Decryption Policy
?
×

1 A Decryption policy is not required only to perform application or URL discovery; instead, you can use TLS 1.3 Server Identity Discovery on the Access Control policy.

Name*

Description

Outbound Connections (User Protection)
Inbound Connections (Server Protection)

How Outbound Protection Works

Outbound protection matches traffic based on the referenced internal CA certificate's signature algorithm type, in addition to any configured rule conditions.

The diagram illustrates the decryption process. It shows a flow from a SOURCE (represented by a laptop icon) to a DESTINATION (represented by a cloud icon). In the middle, there is a DECRYPT RE-SIGN step (represented by a green padlock icon). Arrows indicate the direction of traffic. Above the flow, there is a label 'DECRYPTION EXCLUSIONS' with a padlock icon, and arrows pointing to the SOURCE and DESTINATION, indicating that traffic to these destinations is excluded from decryption.

Internal CA [Download](#)

A rule will be auto-created for the selected certificate authority.

Associated: 2 Networks, 0 Ports

[See how to configure](#)

Cancel Save

Step 5 Upload or choose certificates for the rules.
The system creates one rule per certificate.

Step 6 (Optional.) Choose networks and ports.
For more information:

- [Decryption Rule Conditions](#)
- [Network Rule Conditions](#)
- [Port Rule Conditions](#)

Step 7 Click **Save**.

What to do next

- Add rule conditions: [Decryption Rule Conditions](#)
- Add a default policy action: [Decryption Policy Default Actions, on page 10](#)

- Configure logging options for the default action.
- Set advanced policy properties: [Decryption Policy Advanced Options, on page 12](#).
- Associate the decryption policy with an access control policy as described in [Associating Other Policies with Access Control](#).
- Deploy configuration changes.

Upload an Internal CA for Outbound Protection

This task discusses how to upload an internal certificate authority when you create a decryption rule that protects outbound connections. You can also upload the internal CA using **Objects > Object Management** as discussed in [Importing a CA Certificate and Private Key](#).

Before you begin

Make sure you have an internal certificate authority in one of the formats discussed in [Internal Certificate Authority Objects](#).

Procedure

-
- Step 1** Click **Policies > Access Control > Decryption**.
 - Step 2** Click **New Policy**.
 - Step 3** Enter a name for the policy in the **Name** field and an optional description in the **Description** field.
 - Step 4** Click the **Outbound Connections** tab.
 - Step 5** From the **Internal CA** list, click **Create New > Upload CA**.
 - Step 6** Give the internal CA a **Name**.
 - Step 7** Paste or browse to locate the certificate and its private key in the provided fields.
 - Step 8** If the CA has a password, select the **Encrypted** check box and enter the password in the adjacent field.
-

Generate an Internal CA for Outbound Protection

This task discusses how you can optionally generate an internal certificate authority when you create a decryption rule that protects outbound connections. You can also perform these tasks using **Objects > Object Management** as discussed in [Uploading a Signed Certificate Issued in Response to a CSR](#).

Before you begin

Make sure you understand the requirements for generating an internal certificate authority object as discussed in [Internal Certificate Authority Objects](#).

Procedure

-
- Step 1** Click **Policies > Access Control > Decryption**.
 - Step 2** Click **New Policy**.

Step 3 Enter a name for the policy in the **Name** field and an optional description in the **Description** field.

Step 4 Click the **Outbound Connections** tab.

Step 5 From the **Internal CA** list, click **Create New > Generate CA**.

Step 6 Give the internal CA a **Name** and provide a two-letter **Country Name**.

Step 7 Click **Self-Signed** or **CSR**.

For more information about these options, see [Internal Certificate Authority Objects](#).

Step 8 Enter the requested information in the provided fields.

Step 9 Click **Save**.

Step 10 If you chose **CSR**, after the signing request has been completed, click **Install Certificate** as follows:

- a) Repeat the preceding steps in this procedure.
- b) Edit the CA from the **Internal CA** list as follows.



- c) Click **Install Certificate**.
- d) Follow the prompts on your screen to complete the task.

Create a Decryption Policy with Inbound Connection Protection

This task discusses how to create a decryption policy with a rule that protects inbound connections; that is, the destination server is inside your protected network. This type of rule has a **Decrypt - Known Key** rule action.

When you create a decryption policy, you can create multiple rules at the same time, including multiple **Decrypt - Known Key** rules and multiple **Decrypt - Resign** rules.

Threat Defense Feature History:

7.4—This feature is added.

Before you begin

You must upload an internal certificate for your internal server before you can create a decryption policy that protects inbound connections. You can do this in any of the following ways:

- Create an internal certificate object by going to **Objects > Object Management > PKI > Internal Certs** and referring to [PKI](#).
- At the time you create this decryption policy.

Procedure

- Step 1** Log in to CDO.
- Step 2** Click **Tools & Services > Firewall Management Center > Policies > Policies > Access Control > Decryption**.
- Step 3** Click **New Policy**.
- Step 4** Give the policy a unique **Name** and, optionally, a **Description**.
- Step 5** Click the **Inbound Connections** tab.

Create Decryption Policy
?
×

1 A Decryption policy is not required only to perform application or URL discovery; instead, you can use TLS 1.3 Server Identity Discovery on the Access Control policy.

Name*

Description

Outbound Connections (User Protection)
Inbound Connections (Server Protection)

How Inbound Protection Works

Protect internal services from external attackers.

The diagram illustrates the flow of encrypted traffic. On the left, an 'INTERNAL SERVICE' (represented by a server icon) sends 'Encrypted Traffic' (indicated by a padlock icon) to a central 'DECRYPT KNOWN-KEY' (represented by a green padlock icon). From there, the traffic is sent to a 'SOURCE' (represented by a cloud icon), also labeled as 'Encrypted Traffic'.

Internal Certificates

A rule will be auto-created for each certificate.

+
Drag and drop to order your certificates

1. DecryptOutboundFacebook	Associated: 2 Networks, 0 Ports
2. DecryptOutboundEverythin	Associated: 2 Networks, 0 Ports

Cancel
Save

- Step 6** Upload or choose certificates for the rules.
- The system creates one rule per certificate.
- Step 7** (Optional.) Choose networks and ports.
- For more information:
- [Decryption Rule Conditions](#)
 - [Network Rule Conditions](#)

- [Port Rule Conditions](#)

Step 8 Click **Save**.

What to do next

- Add rule conditions: [Decryption Rule Conditions](#)
- Add a default policy action: [Decryption Policy Default Actions, on page 10](#)
- Configure logging options for the default action .
- Set advanced policy properties: [Decryption Policy Advanced Options, on page 12](#).
- Associate the decryption policy with an access control policy as described in [Associating Other Policies with Access Control](#).
- Deploy configuration changes.

Create a Decryption Policy with Other Rule Actions

To create a decryption rule with a **Do Not Decrypt**, **Block**, **Block With Reset**, or **Monitor** rule action, create a decryption policy and edit the policy to add the rule.

When you create a decryption policy, you can create multiple rules at the same time, including multiple **Decrypt - Known Key** rules and multiple **Decrypt - Resign** rules.

Procedure

- Step 1** Click **Policies > Access Control > Decryption**.
- Step 2** Click **New Policy**.
- Step 3** Give the policy a unique **Name** and, optionally, a **Description**.
- Step 4** Click **Edit** (✎) next to the decryption policy name.
- Step 5** Click **Add Rule**.
- Step 6** Give the rule a Name.
- Step 7** From the **Action** list, click a rule action and see one of the following sections for more information:
- [Decryption Rule Do Not Decrypt Action](#)
 - [Decryption Rule Blocking Actions](#)
 - [Decryption Rule Monitor Action](#)
- Step 8** Click **Save**.
-

What to do next

- Add rule conditions: [Decryption Rule Conditions](#)

- Add a default policy action: [Decryption Policy Default Actions, on page 10](#)
- Configure logging options for the default action.
- Set advanced policy properties: [Decryption Policy Advanced Options, on page 12](#).
- Associate the decryption policy with an access control policy as described in [Associating Other Policies with Access Control](#).
- Deploy configuration changes.

Decryption Policy Default Actions

The default action for a decryption policy determines how the system handles decryptable encrypted traffic that does not match any non-monitor rule in the policy. When you deploy a decryption policy that does not contain any decryption rules, the default action determines how all decryptable traffic on your network is handled. Note that the system does not perform any kind of inspection on encrypted traffic blocked by the default action.

To set the decryption policy default action:

1. Log in to the management center if you haven't already done so.
2. Click **Policies > Access Control > Decryption**.
3. Click **Edit** (✎) next to the name of the decryption policy.
4. In the Default Action row, click one of the following actions from the list.

Table 1: Decryption Policy Default Actions

Default Action	Effect on Encrypted Traffic
Block	Block the TLS/SSL session without further inspection.
Block with reset	Block the TLS/SSL session without further inspection and reset the TCP connection. Choose this option if traffic uses a connectionless protocol like UDP. In that case, the connectionless protocol tries to reestablish the connection until it is reset. This action also displays a connection reset error in the browser so the user is informed that the connection is blocked.
Do not decrypt	Inspect the encrypted traffic with access control.

Related Topics

[Create Basic Decryption Policies](#)

Default Handling Options for Undecryptable Traffic

Table 2: Undecryptable Traffic Types

Type	Description	Default Action	Available Action
Compressed Session	The TLS/SSL session applies a data compression method.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
SSLv2 Session	The session is encrypted with SSL version 2. Note that traffic is decryptable if the ClientHello message is SSL 2.0, and the remainder of the transmitted traffic is SSL 3.0.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
Unknown Cipher Suite	The system does not recognize the cipher suite.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
Unsupported Cipher Suite	The system does not support decryption based on the detected cipher suite.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
Session not cached	The TLS/SSL session has session reuse enabled, the client and server reestablished the session with the session identifier, and the system did not cache that session identifier.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
Handshake Errors	An error occurred during TLS/SSL handshake negotiation.	Inherit default action	Do not decrypt Block Block with reset Inherit default action
Decryption Errors	An error occurred during traffic decryption.	Block	Block Block with Reset

When you first create a decryption policy, logging connections that are handled by the default action is disabled by default. Because the logging settings for the default action also apply to undecryptable traffic handling, logging connections handled by the undecryptable traffic actions is disabled by default.

Note that if your browser uses certificate pinning to verify a server certificate, you cannot decrypt this traffic by re-signing the server certificate. For more information, see [Decryption Rule Guidelines and Limitations](#).

Related Topics

[Set Default Handling for Undecryptable Traffic](#), on page 12

Set Default Handling for Undecryptable Traffic

You can set undecryptable traffic actions at the decryption policy level to handle certain types of encrypted traffic the system cannot decrypt or inspect. When you deploy a decryption policy that contains no decryption rules, the undecryptable traffic actions determine how all undecryptable encrypted traffic on your network is handled.

Depending on the type of undecryptable traffic, you can choose to:

- Block the connection.
- Block the connection, then reset it. This option is preferable for connectionless protocols like UDP, which keep trying to connect until the connection is blocked.
- Inspect the encrypted traffic with access control.
- Inherit the default action from the decryption policy.

Procedure

- Step 1** Click **Policies > Access Control > Decryption**.
 - Step 2** Click **Edit** (✎) next to the name of the decryption policy.
 - Step 3** In the decryption policy editor, click **Undecryptable Actions**.
 - Step 4** For each field, choose either the decryption policy's default action or another action you want to take on the type of undecryptable traffic. See [Default Handling Options for Undecryptable Traffic, on page 11](#) and [Decryption Policy Default Actions, on page 10](#) for more information.
 - Step 5** Click **Save** to save the policy.
-

What to do next

- Configure default logging for connections handled by the undecryptable traffic actions.
- Deploy configuration changes.

Decryption Policy Advanced Options

A decryption policy's **Advanced Settings** tab page has global settings that are applied to all managed devices that are configured for Snort 3 to which the policy is applied.

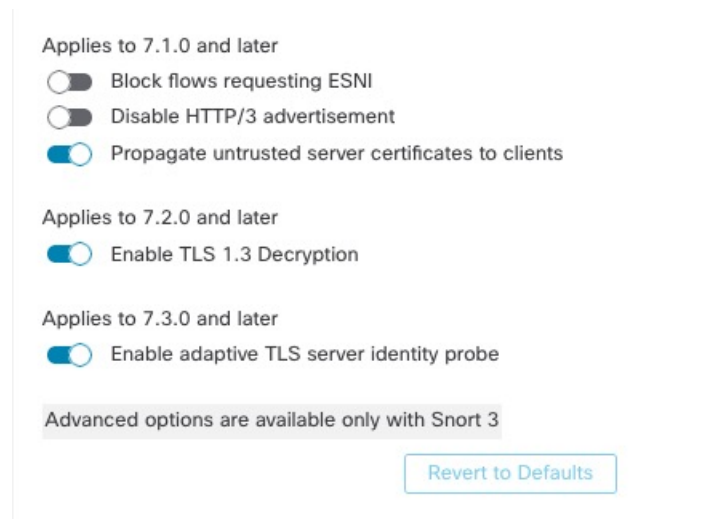
Threat Defense Feature History:

- 7.3—Enable adaptive TLS server identity probe added.
- 7.2—Enable TLS 1.3 decryption added.
- 7.2—Initial options.

A decryption policy advanced settings are all ignored on any managed device that runs:

- A version earlier than 7.1
- Snort 2

Following is an example.

**Block flows requesting ESNI**

Encrypted Server Name Indication (ESNI ([link to draft proposal](#))) is a way for a client to tell a TLS 1.3 server what the client is requesting. Because the SNI is encrypted, you can optionally block these connections because the system cannot determine what the server is.

Disable HTTP/3 advertisement

This option strips HTTP/3 ([RFC 9114](#)) from the ClientHello in TCP connections because:

- As stated in RFC 9114, HTTP/3 is part of the QUIC transport protocol, not the TCP transport protocol
- QUIC is not yet supported by the Firepower system
- Blocking clients from advertising HTTP/3 provides protection against attacks and evasion attempts

Propagate untrusted server certificates to clients

This applies only to traffic matching a **Decrypt - Resign** rule action.

Enable this option to substitute the certificate authority (CA) on the managed device for the server's certificate in cases where the server certificate is untrusted. An *untrusted* server certificate is one that is not listed as a

trusted CA in the Secure Firewall Management Center. (**Objects > Object Management > PKI > Trusted CAs**).

Enable TLS 1.3 Decryption

(Snort 3 only.) Move the slider to enable Threat Defense devices managed by this Management Center to decrypt TLS 1.3 traffic.

If the slider is moved to the disabled position, the system decrypts TLS 1.2 traffic only.

Enable adaptive TLS server identity probe

Automatically enabled when TLS 1.3 decryption is enabled. A *probe* is a partial TLS connection with the server, the purpose of which is to obtain the server certificate and cache it. (If the certificate is already cached, the probe is never established.)

If TLS 1.3 Server Identity Discovery is disabled on the access control policy with which the decryption policy is associated, we attempt to use the Server Name Indication (SNI), which is not as reliable.

The adaptive TLS server identity probe occurs on any of the following conditions as opposed to on every connection as in earlier releases:

- Certificate Issuer—Matched when the value of **Issuer DNs** in a decryption rule's DN rule condition is matched.
For more information, see [Distinguished Name \(DN\) Rule Conditions](#).
- Certificate Status—Matched when any of the **Cert Status** conditions are matched in a decryption rule.
For more information, see [Certificate Status Decryption Rule Conditions](#).
- Internal/External Certificate—Internal certificates can be matched by the certificate used in **Decrypt - Known Key** rule actions; external certificates can be matched in **Certificates** rule conditions.
For more information, see [Known Key Decryption \(Incoming Traffic\)](#) and [Certificate Decryption Rule Conditions](#).
- Application ID—Can be matched by **Applications** rule conditions in either an access control policy or a decryption policy.
For more information, see [Application Rule Conditions](#).
- URL Category—Can be matched by **URLs** rule conditions in an access control policy.
For more information, see [URL Rule Conditions](#).



Note **Enable adaptive TLS server discovery mode** is not supported on any Secure Firewall Threat Defense Virtual deployed to AWS. If you have any such managed devices managed by the Secure Firewall Management Center, the connection event **PROBE_FLOW_DROP_BYPASS_PROXY** increments every time the device attempts to extract the server certificate.

Related information

[TLS 1.3 Decryption Best Practices, on page 15](#)

TLS 1.3 Decryption Best Practices

Recommendation: When to enable advanced options

Both the decryption policy and the access control policy have advanced options that affect how traffic is handled, whether the traffic is being decrypted or not.

The advanced options are:

- Decryption policy:
 - TLS 1.3 decryption
 - TLS adaptive server identity probe
- Access control policy: TLS 1.3 Server Identity Discovery

The access control policy setting takes precedence over the decryption policy setting.

Use the following table to decide which option to enable:

TLS adaptive server identity probe setting (decryption policy)	TLS 1.3 Server Identity Discovery setting (access control policy)	Result	Recommended when
Enabled	Disabled	Adaptive probe sent if decryption policy contains <i>any</i> rule conditions specified in Decryption Policy Advanced Options, on page 12 and if the server certificate is not cached.	<ul style="list-style-type: none"> • You're not using application or URL conditions in access control rules • You're decrypting traffic
Enabled	Enabled	Probe is always sent if the server certificate is not cached.	Use only if your access control rules have URL or application conditions
Disabled	Enabled	Probe is always sent if the server certificate is not cached.	Not recommended.
Disabled	Disabled	Probe is never sent.	Very limited usefulness; use only if not decrypting traffic and not using application or URL conditions in the access control rule



Note A cached TLS server's certificate is available to all Snort instances on a particular threat defense. The cache can be cleared with a CLI command and is automatically cleared when the device is rebooted.

Reference

For more information, see the discussion of [TLS server identity discovery](#) on [secure.cisco.com](#).

Manage Decryption Policies

In the decryption policy editor, you can:

- Add, edit, delete, enable, disable, and organize decryption rules.
- Add trusted CA certificates.
- Determine the handling for encrypted traffic the system cannot decrypt.
- Log traffic that is handled by the default action and undecryptable traffic actions.
- Set advanced options.






In a multidomain deployment, the system displays policies created in the current domain, which you can edit. It also displays policies created in ancestor domains, which you cannot edit. To view and edit policies created in a lower domain, switch to that domain.

Only one person should edit a policy at a time, using a single browser window. If multiple users save the same policy, the last saved changes are retained. For your convenience, the system displays information on who (if anyone) is currently editing each policy. To protect the privacy of your session, a warning appears after 30 minutes of inactivity on the policy editor. After 60 minutes, the system discards your changes.

Procedure

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

Step 2 Manage decryption policies:

- Compare—Click **Compare Policies**; see [Comparing policies](#).
- Copy—Click **Copy** (.
- Create—Click **New Policy**; see [Create Basic Decryption Policies](#).
- Delete—Click **Delete** (). If the controls are dimmed, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.
- Report—Click **Report** (); see [Generating Current Policy Reports](#).
- Edit—Click **Edit** (). If **View** () appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.
- To add trusted CA certificates to your decryption policy, see [Trusting External Certificate Authorities](#).
- To configure how your decryption policy handles undecryptable traffic, see [Set Default Handling for Undecryptable Traffic, on page 12](#).

- To log connections for undecryptable traffic handling and traffic that does not match SSL rules, see *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#).
 - Deploy—Choose **Deploy** > **Deployment**; see [Deploy Configuration Changes](#).
-

