



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 3](#)
- [Decryption policy default actions, on page 15](#)
- [Default handling options for undecryptable traffic, on page 15](#)
- [Decryption policy advanced options, on page 17](#)

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.

Create a decryption policy using the wizard

You can use a wizard to create the following types of decryption policies:

- Outbound protection (**Decrypt - Resign** rule action). If traffic matches this rule, the system re-signs the server certificate with the CA certificate, then acts as a man-in-the-middle.

Three rules with a **Do Not Decrypt** action are added to the policy at the same time, saving you the trouble of doing it later. These rules correspond to any decryption exclusions you configure when you create the policy (for example, you can choose to bypass decryption for applications that are known to use certificate pinning).

For more information, see [Create a decryption policy with outbound connection protection, on page 3](#).

- Inbound protection (**Decrypt - Known Key** rule action). You can associate one or more server certificates and paired private keys with the action. If traffic matches the rule, and the certificate used to encrypt the traffic matches the certificate associated with the action, the system uses the appropriate private key to obtain the session encryption and decryption keys.

Three rules with a **Do Not Decrypt** action are added to the policy at the same time but these rules are disabled by default. These rules correspond to any decryption exclusions you configure when you create

the policy (for example, you can choose to bypass decryption for applications that are known to use certificate pinning).

For more information, see [Create a decryption policy with inbound connection protection, on page 6](#).

- Any other decryption rule action (such as blocking or monitoring).

For more information, see [Create a decryption policy with other rule actions, on page 13](#).

The wizard automatically creates a separate rule for each certificate you specify. For example, an inbound protection rule might specify one certificate for traffic coming to the Finance department internal network and a different certificate for traffic coming to the Engineering network.

The wizard creates additional rules for outbound and inbound protection policies as follows:

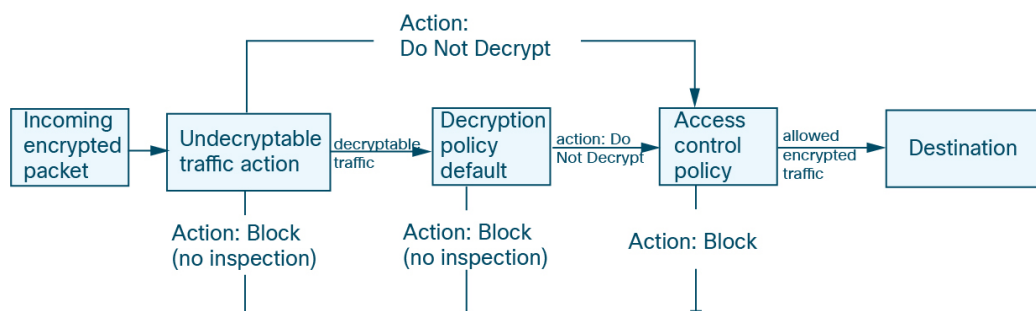
- Outbound protection (**Decrypt - Resign** rule action): The wizard creates Do Not Decrypt rules for traffic that matches exceptions you specify in the wizard. For example, you can choose to not decrypt traffic from undecryptable applications, typically those using certificate pinning.

The Do Not Decrypt rules are placed first in the decryption policy so that traffic is passed through the firewall with minimal processing.

- Inbound protection (**Decrypt - Known Key** rule action): The wizard does not allow you to choose any exceptions but it adds the Do Not Decrypt rules to the policy and disables them; that way, you can enable those exceptions later if you need to.

Do Not Decrypt policy example

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.

To get started, see [Create a decryption policy , on page 3](#)

Requirements and Prerequisites for Decryption Policies

Supported domains

Any

User roles

- Admin
- Access Admin
- Network Admin

Create a decryption policy

You can create any of the following types of decryption policies:

- *Outbound protection* policy with rules that protect outbound connections; that is, the destination server is outside your protected network. This type of rule has a **Decrypt - Resign** rule action. We also create additional rules with a **Do Not Decrypt** action that excludes traffic you specify (such as traffic that uses certificate pinning).

See [Create a decryption policy with outbound connection protection, on page 3](#)

- *Inbound protection* 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. We also create additional rules with a **Do Not Decrypt** action that excludes traffic you specify (such as traffic that uses certificate pinning.) These rules are disabled initially but you can modify and enable them later if you wish.

See [Create a decryption policy with inbound connection protection, on page 6](#)

- Other actions (including **Do Not Decrypt**, **Block**, and **Block with Reset**).

See [Create a decryption policy with other rule actions, on page 13](#)

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.

If you enabled Change Management, you must create and assign a ticket before you can create a decryption policy. Before the decryption policy can be used, the ticket and all associated objects (like certificate authorities) must be approved. For more information, see [Creating change management tickets](#) and [Policies and objects that support change management](#).

Before you begin

You can optionally must upload or generate an internal CA certificate for your managed device 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 certificate 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** Log in to Secure Firewall Management Center if you haven't already done so.
- Step 2** Click **Policies > Access Control heading > Decryption**.
- Step 3** Click **Create Decryption Policy**.
- Step 4** Give the policy a unique **Name** and, optionally, a **Description**.

The following characters are not supported in decryption policy names:

- Leading period
- #, ; , { , } , = , \$, < , >

- Step 5** Click the **Outbound Connections** tab.

Create Decryption Policy

1 Policy Details
Enter name, description, choose policy type and certificates.

2 Decryption Exclusions
(Optional) Configure exclusions for outbound connections.

i A decryption policy is not required to only 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))
    EXCLUSIONS[DECRYPTION EXCLUSIONS] -.-> DECRYPT
  
```

Internal CA

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

Associated: 1 Network, 1 Port

[See how to configure](#)

[Download](#)

[Cancel](#) [Next](#)

- Step 6** Click the **Outbound Connections** tab.

Create Decryption Policy ?

1 Policy Details
 Enter name, description, choose policy type and certificates.

2 Blocking
 (Optional) Configure blocking based on TLS version and certificate status

3 Decryption Exclusions
 (Optional) Configure exclusions for outbound connections.

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

Name *

Outbound example

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.

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

IntCA Associated: 1 Network, 1 Port

[See how to configure](#)

[Download](#)

[Cancel](#) [Skip](#) [Next](#)

Step 7 From the **Internal CA** list, upload or choose certificates for the rules.

For more information about internal certificates, see [Generate an internal CA for outbound protection, on page 11](#) and [Upload an internal CA for outbound protection, on page 12](#).

Step 8 (Optional.) Choose networks and ports.

For more information:

- [Network rule conditions](#)
- [Port rule conditions](#)

Step 9 Click **Next**.

Step 10 Continue with [Decryption policy exclusions, on page 8](#).

What to do next

- Add rule conditions: [Decryption Rule conditions](#)
- Add a default policy action: [Decryption policy default actions, on page 15](#)
- Configure logging options for the default action as described in *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#).
- Set advanced policy properties: [Decryption policy advanced options, on page 17](#).
- Associate the decryption policy with an access control policy as described in [Associating other policies with access control](#).
- Deploy configuration changes; see [Deploy Configuration Changes](#).

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.

Before you begin

You can optionally 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.

If you enabled Change Management, you must create and assign a ticket before you can create a decryption policy. Before the decryption policy can be used, the ticket and all associated objects (like certificate authorities) must be approved. For more information, see [Creating change management tickets](#) and [Policies and objects that support change management](#).

Procedure

-
- Step 1** Log in to Secure Firewall Management Center if you haven't already done so.
- Step 2** Click **Policies > Access Control heading > Decryption**.
- Step 3** Click **Create Decryption Policy**.
- Step 4** Give the policy a unique **Name** and, optionally, a **Description**.

The following characters are not supported in decryption policy names:

- Leading period
- #, ; , { , } , = , \$, < , >

Step 5 From the **Internal Certificates** list, upload or choose certificates for the rules.
For more information about internal CA certificates, see [Internal Certificate Authority Objects](#).

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

- [Network rule conditions](#)
- [Port rule conditions](#)

Step 7 Click the **Inbound Connections** tab.

Create Decryption Policy

1 Policy Details
Enter name, description, choose policy type and certificates.

2 Decryption Exclusions
(Optional) Configure exclusions for outbound connections.

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

Name *
Inbound example

Description

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

How Inbound Protection Works
Protect internal services from external attackers.

Internal Certificates
A rule will be auto-created for each certificate.

+ Drag and drop to order your certificates

| | |
|------------------------|-------------------------------|
| 1. InboundCertFacebook | Associated: 1 Network, 1 Port |
| 2. InternalCert | Associated: 1 Network, 1 Port |

Cancel Next

Step 8 Click **Next**.

Step 9 Continue with [Decryption policy exclusions, on page 8](#).

What to do next

- Add rule conditions: [Decryption Rule conditions](#)
- Add a default policy action: [Decryption policy default actions, on page 15](#)
- Configure logging options for the default action as described in *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#).
- Set advanced policy properties: [Decryption policy advanced options, on page 17](#).
- Associate the decryption policy with an access control policy as described in [Associating other policies with access control](#).

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

Decryption policy exclusions

This task discusses how to exclude certain types of traffic from decryption. We create **Do Not Decrypt** rules in your decryption policy for these although the rules are initially enabled only for an outbound decryption policy (that is, one that uses the **Decrypt - Resign** policy action).

Before you begin

You must upload an internal CA certificate for your managed device 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 certificate 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 Complete the tasks discussed in:

- [Create a decryption policy with outbound connection protection, on page 3](#)
- For more information, see [Create a decryption policy with inbound connection protection, on page 6](#)

Step 2 The exclusions page provides the following options. All options are *enabled* for an outbound protection policy (**Decrypt - Resign** rule action) and *disabled* for all other decryption policy actions.

| Item | Description |
|--|---|
| Bypass decryption for sensitive URL categories | <p>Check the box to not decrypt traffic from the indicated categories. Depending on the laws in your area, decryption certain traffic, such as finance or health-related, might be prohibited. Consult an authority in your area for more information.</p> <p>Click Add to add more categories.</p> <p>Click Delete (✕) to remove categories.</p> |
| Bypass decryption for undecryptable distinguished names | <p>Check the box to not decrypt traffic when re-signing the certificate is likely to cause the connection to fail. Typically, this behavior is associated with <i>certificate pinning</i>, which is discussed in TLS/SSL certificate pinning guidelines.</p> <p>The list of undecryptable distinguished names is maintained by Cisco.</p> |

| Item | Description |
|---|--|
| Bypass decryption for undecryptable applications | <p>Check the box to not decrypt traffic when re-signing the certificate is likely to cause the connection to fail.</p> <p>Typically, this behavior is associated with <i>certificate pinning</i>, which is discussed in TLS/SSL certificate pinning guidelines.</p> <p>Undecryptable applications are updated automatically in the Vulnerability Database (VDB). You can find a list of all applications on the Secure Firewall Application Detectors page; the undecryptable tag identifies applications Cisco determines are undecryptable.</p> <p>The list of undecryptable applications is maintained by Cisco.</p> |

The following figure shows the default options.

Create Decryption Policy ?

1 Policy Details
Enter name, description, choose policy type and certificates.

2 Decryption Exclusions
(Optional) Configure exclusions for outbound connections.

☐ **Bypass decryption for sensitive URL categories**
 In many environments, certain categories of websites are not inspected for regulatory, compliance or privacy reasons. Customize the list below to bypass inspection for designated categories.
 Note: **URL License is Required**
 URL Categories: Finance Online Trading Health and Medicine + Add

☒ **Bypass decryption for undecryptable distinguished names**
 Bypass decryption based on Cisco's list of known undecryptable distinguished names.
 Note: **This option is selected by default to allow traffic which cannot be decrypted to remain encrypted. Disabling this option might cause decryption to fail for unsupported distinguished names.**
56 Distinguished names included

☒ **Bypass decryption for undecryptable applications**
 Certain enterprise applications are not supported for decryption due to a variety of reasons (Certificate Pinning, Client Certificate Authentication, etc.). Bypass decryption based on Cisco's list of known undecryptable applications.
 Note: **This option is selected by default to allow traffic which cannot be decrypted to remain encrypted. Disabling this option might cause decryption to fail for unsupported applications.**
55 Applications included

Cancel Back Create Policy

Create Decryption Policy



- 1 **Policy Details**
Enter name, description, choose policy type and certificates.
- 2 **Blocking**
(Optional) Configure blocking based on TLS version and certificate status
- 3 **Decryption Exclusions**
(Optional) Configure exclusions for outbound connections.

Decryption Exclusions

☐ Bypass decryption for sensitive URL categories

In many environments, certain categories of websites are not inspected for regulatory, compliance or privacy reasons. Customize the list below to bypass inspection for designated categories.

Note: **URL License is Required**

URL Categories:

Health and Medicine ×

Online Trading ×

Finance ×

+ Add

☒ Bypass decryption for undecryptable distinguished names

Bypass decryption based on Cisco's list of known undecryptable distinguished names.

Note: **This option is selected by default to allow traffic which cannot be decrypted to remain encrypted. Disabling this option might cause decryption to fail for unsupported distinguished names.**

56 Distinguished names included ▾

☒ Bypass decryption for undecryptable applications

Certain enterprise applications are not supported for decryption due to a variety of reasons (Certificate Pinning, Client Certificate Authentication, etc.). Bypass decryption based on Cisco's list of known undecryptable applications.

Note: **This option is selected by default to allow traffic which cannot be decrypted to remain encrypted. Disabling this option might cause decryption to fail for unsupported applications.**

56 Applications included ▾

Intelligent Decryption Bypass

☐ Bypass decryption for very low-risk connections

New

Bypass decryption for very low-risk clients connecting to trusted servers.

Note: **The access control policy associated with this decryption policy must have the Encrypted Visibility Engine (EVE) enabled. The device to which this policy is deployed must run version 7.7 or later and must have a valid IPS license.**

Cancel

Back

Create Policy

Step 3 Click **Create Policy**.

The following figure shows a sample outbound protection policy.

Outbound example

Enter Description

Save

Cancel

Rules Trusted CA Certificates Undecryptable Actions Advanced Settings

+ Add Category

+ Add Rule

Search Rules

| # | Name | Source Zones | Dest Zones | Source Networks | Dest Networks | VLAN Tags | Users | Applications | Source Ports | Dest Ports | Categories | SSL | Action |
|----------------------------|--|--------------|------------|-----------------|---------------|-----------|-------|-----------------|--------------|------------|---|----------------|---|
| Administrator Rules | | | | | | | | | | | | | |
| This category is empty | | | | | | | | | | | | | |
| Standard Rules | | | | | | | | | | | | | |
| 1 | <input checked="" type="checkbox"/> Auto-Rule-Undecryptable | any | any | any | any | any | any | any | any | any | any | 1 DN selection | <input checked="" type="radio"/> Do not decrypt |
| 2 | <input type="checkbox"/> Auto-Rule-URL-Categories (Disabled) | any | any | any | any | any | any | any | any | any | Finance (Any res), Health and Med, Online Trading (| any | <input checked="" type="radio"/> Do not decrypt |
| 3 | <input type="checkbox"/> Auto-Rule-Undecryptable-A | any | any | any | any | any | any | Tags: undecrypt | any | any | any | any | <input checked="" type="radio"/> Do not decrypt |
| 4 | <input checked="" type="checkbox"/> Auto-Rule-IntCA | any | any | IPv4-Link-Local | any | any | any | any | any | Bittorrent | any | any | <input checked="" type="radio"/> Decrypt - Resign |
| Root Rules | | | | | | | | | | | | | |
| This category is empty | | | | | | | | | | | | | |
| Default Action | | | | | | | | | | | | | Do not decrypt |

In the preceding example, the **Do Not Decrypt** rules corresponding to your choices for rule exclusions are automatically added before the **Decrypt - Resign** rule. The rule for sensitive URL categories is disabled because, by default, that exclusion is disabled. Had you selected the **Bypass decryption for sensitive URL categories** check box, the rule would have been enabled.

Step 4 Click **Create Policy**.

What to do next

- Add rule conditions: [Decryption Rule conditions](#)
- Add a default policy action: [Decryption policy default actions, on page 15](#)
- Configure logging options for the default action as described in *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#).
- Set advanced policy properties: [Decryption policy advanced options, on page 17](#).
- Associate the decryption policy with an access control policy as described in [Associating other policies with access control](#).
- Deploy configuration changes; see [Deploy Configuration Changes](#).

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** From the **Internal CA** list, click **Create New > Generate CA**.
- Step 2** Give the internal CA a **Name** and provide a two-letter **Country Name**.
- Step 3** Click **Self-Signed** or **CSR**.
For more information about these options, see [Internal Certificate Authority Objects](#).
- Step 4** Enter the requested information in the provided fields.
- Step 5** Click **Save**.
- Step 6** If you chose **CSR**, after the signing request has been completed, click **Install Certificate** as follows:
- Repeat the preceding steps in this procedure.
 - Edit the CA from the **Internal CA** list as follows.



- Click **Install Certificate**.
 - Follow the prompts on your screen to complete the task.
- Step 7** Continue creating the policy as discussed in [Create a decryption policy with outbound connection protection, on page 3](#).
-

Upload an internal CA for outbound protection

This task discusses how you can optionally upload 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** From the **Internal CA** list, click **Create New > Upload CA**.
- Step 2** Give the internal CA a **Name**.
- Step 3** Paste or browse to locate the certificate and its private key in the provided fields.
- Step 4** If the CA has a password, select the **Encrypted** check box and enter the password in the adjacent field.

- Step 5** Continue creating the policy as discussed in [Create a decryption policy with outbound connection protection, on page 3](#).
-

Upload an internal certificate for inbound protection

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

Before you begin

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

Procedure

- Step 1** Log in to Secure Firewall Management Center if you haven't already done so.
- Step 2** Click **Policies > Access Control heading > Decryption**.
- Step 3** Click **Create Decryption Policy**.
- Step 4** Enter a name for the policy in the **Name** field and an optional description in the **Description** field.
- Step 5** Click the **Inbound Connections** tab.
- Step 6** From the **Internal Certificates** list, click **Add (+)**.
- Step 7** If an internal certificate object exists, click its name.
- Step 8** Otherwise, click **Upload**.
- Step 9** Enter the required information.
See [Adding Internal Certificate Objects](#).
- Step 10** Continue creating the decryption policy as discussed in [Create a decryption policy with inbound connection protection, on page 6](#).
-

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.

If you enabled Change Management, you must create and assign a ticket before you can create a decryption policy. Before the decryption policy can be used, the ticket and all associated objects (like certificate authorities) must be approved. For more information, see [Creating change management tickets](#) and [Policies and objects that support change management](#).

Procedure

-
- Step 1** Log in to Secure Firewall Management Center if you haven't already done so.
- Step 2** Click **Policies > Access Control heading > Decryption**.
- Step 3** Give the policy a unique **Name** and, optionally, a **Description**.
- The following characters are not supported in decryption policy names:
- Leading period
 - #, ; , { , } , = , \$, < , >
- Step 4** Click **Next**.
- Step 5** The bypass page is provided for your information only; you cannot bypass traffic for other types of decryption (such as **Block**).
- Step 6** Click **Create Policy**.
- Step 7** Wait for the policy to be created.
- Step 8** Click **Edit** (✎) next to the decryption policy name.
- Step 9** Click **Add Rule**.
- Step 10** Give the rule a **Name**.
- Step 11** 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 12** Click **Save**.
-

What to do next

- Add rule conditions: [Decryption Rule conditions](#)
- Add a default policy action: [Decryption policy default actions, on page 15](#)
- Configure logging options for the default action as described in *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#) .
- Set advanced policy properties: [Decryption policy advanced options, on page 17](#).
- Associate the decryption policy with an access control policy as described in [Associating other policies with access control](#).
- Deploy configuration changes; see [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 policy default action:

1. Log in to the Secure Firewall Management Center if you haven't already done so.
2. Click **Policies > Access Control heading > 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. |

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 |

| Type | Description | Default Action | Available 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 Rules guidelines and limitations](#).

If a decryption policy is associated with an access control policy that uses TCP state bypass, matching traffic is acted on based on the policy's configured **Undecryptable Actions** for **Handshake Errors**.

For example, if a decryption policy's **Handshake Errors** is set to **Block**, traffic matching the rule is blocked and the connection event's action is reported as a handshake error.

For more information about TCP state bypass, see:

- [Configure TCP state bypass](#)

- [Bypass TCP state checks for asymmetrical routing \(TCP state bypass\)](#)

Related Topics

[Set default handling for undecryptable traffic](#), on page 17

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 | Log in to Secure Firewall Management Center if you haven't already done so. |
| Step 2 | Click Policies > Access Control heading > Decryption . |
| Step 3 | Click Edit (✎) next to the name of the decryption policy. |
| Step 4 | In the decryption policy editor, click Undecryptable Actions . |
| Step 5 | 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 15 and Decryption policy default actions, on page 15 for more information. |
| Step 6 | Click Save to save the policy. |
-

What to do next

- Configure default logging for connections handled by the undecryptable traffic actions; see *Logging Connections with a Policy Default Action* in the [Cisco Secure Firewall Management Center Administration Guide](#).
- Deploy configuration changes; see [Deploy Configuration Changes](#).

Decryption policy advanced options

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

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

- A version earlier than 7.1
- Snort 2

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. HTTP/3 is part of the QUIC transport protocol, not the TCP transport protocol. Blocking clients from advertising HTTP/3 provides protection against attacks and evasion attempts potentially buried within QUIC connections.

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

Whether to apply decryption rules to TLS 1.3 connections. If you do not enable this option, the decryption rules apply to TLS 1.2 or lower traffic only. See [TLS 1.3 decryption best practices, on page 20](#).

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 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.

Enable QUIC Decryption

Whether to apply decryption rules to connections that use the HTTP/3 over the QUIC protocol. When you decrypt QUIC connections, the system can inspect the contents of the sessions for intrusions, malware, or other issues. You can also apply granular control and filtering of decrypted QUIC connections based on specific criteria in the access control policy. QUIC support is in line with RFC 9000, 9001, 9002, 9114, 9204.

Consider the following when implementing QUIC decryption:

- On high availability or clustered devices, QUIC decryption works only if the connection remains on the same node. If the connection fails over, or is forwarded to another node, the connection drops and must be re-established. Multi-instance is supported without restrictions.
- Rules that apply to QUIC traffic would include the UDP protocol with destination port 443.
- Access control rules that apply to QUIC traffic would include the HTTP/3 or QUIC protocols, either explicitly or by implication.

The following limitations apply to QUIC decryption:

- QUIC decryption applies to Firewall Threat Defense 7.6+ only. Devices running a lower release cannot decrypt QUIC connections.
- Connections from browsers using the Chromium stack (Google Chrome, Opera, Edge) cannot be decrypted for outbound traffic. But inbound traffic from the same browsers can be decrypted.
- Connection Migration as described in RFC 9000 is not supported. The concept of Connection ID in QUIC allows endpoints to retain the same connection in the event of address change.
- Key update, session resumption, and QUIC version 2 are not supported.
- Interactive Block and Interactive Block with Reset (in access control rules) is not supported. These actions will work as Block and Block with Reset.
- The active connection-ID per connection is limited to 5. If necessary, you can modify these limits using the **system support quic-tuning** and **system support quic-tuning-reset** commands in the device CLI.

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 17 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 Firewall 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](#).

