

SSL Decryption

Some protocols, such as HTTPS, use Secure Sockets Layer (SSL) or its follow-on version, Transport Layer Security (TLS), to encrypt traffic for secure transmissions. Because the system cannot inspect encrypted connections, you must decrypt them if you want to apply access rules that consider higher-layer traffic characteristics to make access decisions.

- About SSL Decryption, on page 1
- License Requirements for SSL Decryption, on page 4
- Guidelines for SSL Decryption, on page 5
- How to Implement and Maintain the SSL Decryption Policy, on page 5
- Configuring SSL Decryption Policies, on page 7
- Example: Blocking Older SSL/TLS Versions from the Network, on page 19
- Monitoring and Troubleshooting SSL Decryption, on page 21

About SSL Decryption

Normally, connections go through the access control policy to determine if they are allowed or blocked. However, if you enable the SSL decryption policy, encrypted connections are first sent through the SSL decryption policy to determine if they should be decrypted or blocked. Any unblocked connections, whether or not decrypted, then go through the access control policy for a final allow/block decision.

Note You must enable the SSL decryption policy in order to implement active authentication rules in the identity policy. If you enable SSL decryption to enable identity policies, but do not otherwise want to implement SSL decryption, select Do Not Decrypt for the default action and do not create additional SSL decryption rules. The identity policy automatically generates whatever rules it needs.

The following topics explain encrypted traffic flow management and decryption in more detail.

Why Implement SSL Decryption?

Encrypted traffic, such as HTTPS connections, cannot be inspected.

Many connections are legitimately encrypted, such as connections to banks and other financial institutions. Many web sites use encryption to protect privacy or sensitive data. For example, your connection to the Firewall Device Manager is encrypted.

However, users can also hide undesirable traffic within encrypted connections.

By implementing SSL decryption, you can decrypt connections, inspect them to ensure they do not contain threats or other undesirable traffic, and then re-encrypt them before allowing the connection to proceed. (The decrypted traffic goes through your access control policy and matches rules based on inspected characteristics of the decrypted connection, not on the encrypted characteristics.) This balances your need to apply access control policies with the user's need to protect sensitive information.

You can also configure SSL decryption rules to block encrypted traffic of types you know you do not want on your network.

Keep in mind that decrypting and then re-encrypting traffic adds a processing load on the device, which will reduce overall system performance.

Actions You Can Apply to Encrypted Traffic

When configuring SSL decryption rules, you can apply the actions described in the following topics. These actions are also available for the default action, which applies to any traffic that does not match an explicit rule.

Note Any traffic that passes through the SSL decryption policy must then pass through the access control policy. Except for traffic you drop in the SSL decryption policy, the ultimate allow or drop decision rests with the access control policy.

Decrypt Re-Sign

If you elect to decrypt and re-sign traffic, the system acts as a man-in-the-middle.

For example, the user types in https://www.cisco.com in a browser. The traffic reaches the Firewall Threat Defense device, the device then negotiates with the user using the CA certificate specified in the rule and builds an SSL tunnel between the user and the Firewall Threat Defense device. At the same time the device connects to https://www.cisco.com and creates an SSL tunnel between the server and the Firewall Threat Defense device.

Thus, the user sees the CA certificate configured for the SSL decryption rule instead of the certificate from www.cisco.com. The user must trust the certificate to complete the connection. The Firewall Threat Defense device then performs decryption/re-encryption in both directions for traffic between the user and destination server.



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

If you configure a rule with the Decrypt Re-Sign action, the rule matches traffic based on the referenced internal CA certificate's signature algorithm type, in addition to any configured rule conditions. Because you can select a single re-sign certificate for the SSL decryption policy, this can limit traffic matching for resign rules.

For example, outgoing traffic encrypted with an elliptic curve (EC) algorithm matches a Decrypt Re-Sign rule only if the re-sign certificate is an EC-based CA certificate. Similarly, traffic encrypted with an RSA algorithm matches Decrypt Re-Sign rules only if the global re-sign certificate is RSA; outgoing traffic encrypted with an EC algorithm does not match the rule, even if all other configured rule conditions match.

Decrypt Known Key

If you own the destination server, you can implement decryption with a known key. In this case, when the user opens a connection to https://www.cisco.com, the user sees the actual certificate for www.cisco.com, even though it is the Firewall Threat Defense device that is presenting the certificate.



Your organization must be the owner of the domain and certificate. For the example of cisco.com the only possible way to have the end user see Cisco's certificate would be if you actually own the domain cisco.com (i.e. you are Cisco Systems) and have ownership of the cisco.com certificate signed by a public CA. You can only decrypt with known keys for sites that your organization owns.

The main purpose of decrypting with a known key is to decrypt traffic heading to your HTTPS server to protect your servers from external attacks. For inspecting client side traffic to external HTTPS sites, you must use decrypt re-sign as you do not own the servers.

Note To use known key decryption, you must upload the server's certificate and key as an internal identity certificate, and then add it to the list of known-key certificates in the SSL decryption policy settings. Then, you can write the rule for known-key decryption with the server's address as the destination address. For information on adding the certificate to the SSL decryption policy, see Configure Certificates for Known Key and Re-Sign Decryption, on page 17.

Do Not Decrypt

If you elect to bypass decryption for certain types of traffic, no processing is done on the traffic. The encrypted traffic proceeds to the access control policy, where it is allowed or dropped based on the access control rule it matches.

Block

You can simply block encrypted traffic that matches an SSL decryption rule. Blocking in the SSL decryption policy prevents the connection from reaching the access control policy.

When you block an HTTPS connection, the user does not see the system default block response page. Instead, the user sees the browser's default page for a secure connection failure. The error message does not indicate the site was blocked due to policy. Instead, errors might indicate that there are no common encryption algorithms. It will not be obvious from this message that you blocked the connection on purpose.

Automatically Generated SSL Decryption Rules

Whether you enable the SSL decryption policy, the system automatically generates Decrypt Re-sign rules for each identity policy rule that implements active authentication. This is required to enable active authentication for HTTPS connections.

When you enable the SSL decryption policy, you see these rules under the Identity Policy Active Authentication Rules heading. These rules are grouped at the top of the SSL decryption policy. The rules are read only. You can change them only by altering your identity policy.

Handling Undecryptable Traffic

There are several characteristics that make a connection undecryptable. If a connection has any of the following characteristics, the default action is applied to the connection regardless of any rule the connection would otherwise match. If you select Block as your default action (rather than Do Not Decrypt), you might run into issues, including excessive drops of legitimate traffic.

- Compressed session—Data compression was applied to the connection.
- SSLv2 session—The minimum supported SSL version is SSLv3.
- Unknown cipher suite—The system does not recognize the cipher suite for the connection.
- Unsupported cipher suite—The system does not support decryption based on the detected cipher suite.
- Session not cached—The 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.
- Handshake errors—An error occurred during the SSL handshake negotiation.
- Decryption errors—An error occurred during the decryption operation.
- Passive interface traffic—All traffic on passive interfaces (passive security zones) is undecryptable.

License Requirements for SSL Decryption

You do not need a special license to use the SSL decryption policy.

However, you do need the **URL** license to create rules that use URL categories and reputations as match criteria. For information on configuring licenses, see Enabling or Disabling Optional Licenses.

Guidelines for SSL Decryption

Keep the following in mind when configuring and monitoring SSL decryption policies:

- The SSL Decryption policy is bypassed for any connections that match access control rules set to trust or block if those rules:
  - Use security zone, network, geolocation, and port only as the traffic matching criteria.
  - Come before any other rules that require inspection, such as rules that match connections based on application or URL, or allow rules that apply intrusion or file inspection.
- When using URL category matching, note that there are cases where the login page for a site is in a different category than the site itself. For example, Gmail is in the "Web based email" category, whereas the login page is in the "Internet Portals" category. To get connections to these sites decrypted, you must include both categories in the rule.
- If a Vulnerability Database (VDB) update removes (deprecates) applications, you must make changes to any SSL decryption rules or application filters that use the application that was deleted. You cannot deploy changes until you fix these rules. In addition, you cannot install system software updates before fixing the issue. On the Application Filters object page, or the Application tab of the rule, these applications say "(Deprecated)" after the application name.
- You cannot disable the SSL decryption policy if you have any active authentication rules. To disable the SSL decryption policy, you must either disable the identity policy, or delete any identity rules that use active authentication.

How to Implement and Maintain the SSL Decryption Policy

You can use SSL decryption policies to turn encrypted traffic into plain text traffic, so that you can then apply URL filtering, intrusion and malware control, and other services that require deep packet inspection. If your policies allow the traffic, the traffic is re-encrypted before it leaves the device.

The SSL decryption policy applies to encrypted traffic only. No unencrypted connections are evaluated against SSL decryption rules.

Unlike some other security policies, you need to monitor and actively maintain the SSL decryption policy, because certificates can expire or even be changed on destination servers. Additionally, changes in client software might alter your ability to decrypt certain connections, because the decrypt re-sign action is indistinguishable from a man-in-the-middle attack.

The following procedure explains the end-to-end process of implementing and maintaining the SSL decryption policy.

Procedure

Step 1 If you will implement Decrypt Re-sign rules, create the required internal CA certificate.

You must use an internal Certificate Authority (CA) certificate. You have the following options. Because users must trust the certificate, either upload a certificate client browsers are already configured to trust, or ensure that the certificate you upload is added to the browser trust stores.

- Create a self-signed internal CA certificate, which is signed by the device itself. See Generating Self-Signed Internal and Internal CA Certificates.
- Upload an internal CA certificate and key signed by an external trusted CA or by a CA inside your organization. See Uploading Internal and Internal CA Certificates.

Step 2 If you will implement Decrypt Known Key rules, collect the certificate and key from each of the internal servers.

You can use Decrypt Known Key only with servers that you control, because you must obtain the certificate and key from the server. Upload these certificates and keys as internal certificates (not internal CA certificates). See Uploading Internal and Internal CA Certificates.

Step 3 Enable the SSL Decryption Policy, on page 8.

When you enable the policy, you also configure some basic settings.

Step 4 Configure the Default SSL Decryption Action, on page 9.

If in doubt, select **Do Not Decrypt** as the default action. Your access control policy can still drop traffic that matches the default SSL decryption rule if appropriate.

Step 5 Configure SSL Decryption Rules, on page 10.

Identify traffic to decrypt and the type of decryption to apply.

Step 6 If you configure known key decryption, edit the SSL decryption policy settings to include those certificates. See Configure Certificates for Known Key and Re-Sign Decryption, on page 17.

Step 7 If necessary, download the CA certificate used for Decrypt Re-sign rules and upload it to the browser on client workstations.

For information on downloading the certificate and distributing it to clients, see Downloading the CA Certificate for Decrypt Re-Sign Rules, on page 18.

Step 8 Periodically, update re-sign and known key certificates.

- Re-sign certificate—Update this certificate before it expires. If you generate the certificate through the Firewall Device Manager, it is valid for 5 years. To check the validity period for a certificate, select **Objects** > **Certificates**, find the certificate in the list, and click the information icon ( ) for it in the Actions column. The information dialog box shows the validity period and some other characteristics. You can also upload a replacement certificate from this page.
- Known-key certificate—For any known-key decryption rules, you need to ensure that you have uploaded the destination server's current certificate and key. Whenever the certificate and key changes on supported servers, you must also upload the new certificate and key (as an internal certificate) and update the SSL decryption settings to use the new certificate.

Step 9 Upload missing trusted CA certificates for external servers.

The system includes a wide range of trusted CA root and intermediate certificates issued by third parties. These are needed when negotiating the connection between the Firewall Threat Defense and the destination servers for decrypt re-sign rules.

Upload all certificates within a root CA's chain of trust to the list of trusted CA certificates, including the root CA certificate and all intermediate CA certificates. Otherwise, it is more difficult to detect trusted certificates issued by intermediate CAs. Upload certificates on the **Objects** > **Certificates** page. See Uploading Trusted CA Certificates.

Configuring SSL Decryption Policies

You can use SSL decryption policies to turn encrypted traffic into plain text traffic, so that you can then apply URL filtering, intrusion and malware control, and other services that require deep packet inspection. If your policies allow the traffic, the traffic is re-encrypted before it leaves the device.

The SSL decryption policy applies to encrypted traffic only. No unencrypted connections are evaluated against SSL decryption rules.

Note VPN tunnels are decrypted before the SSL decryption policy is evaluated, so the policy never applies to the tunnel itself. However, any encrypted connections within the tunnel are subject to evaluation by the SSL decryption policy.

The following procedure explains how to configure the SSL decryption policy. For an explanation of the end-to-end process of creating and managing SSL decryption, see How to Implement and Maintain the SSL Decryption Policy, on page 5.

Before you begin

The SSL decryption rules table contains two sections:

- **Identity Policy Active Authentication Rules**—If you enable the identity policy and create rules that use active authentication, the system automatically creates the SSL decryption rules needed to make those policies work. These rules are always evaluated before the SSL decryption rules you create yourself. You can alter these rules only indirectly, by making changes to the identity policy.
- **SSL Native Rules**—These are rules that you have configured. You can add rules to this section only.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

If you have not yet enabled the policy, click **Enable SSL Decryption** and configure policy settings, as described in Enable the SSL Decryption Policy, on page 8.

Step 2 Configure the default action for the policy.

The safest choice is **Do Not Decrypt**. For more information, see Configure the Default SSL Decryption Action, on page 9.

Step 3 Manage the SSL decryption policy.

After you configure SSL decryption settings, this page lists all rules in order. Rules are matched against traffic from top to bottom with the first match determining the action to apply. You can do the following from this page:

- To disable the policy, click the **SSL Decryption Policy** toggle. You can re-enable it by clicking **Enable SSL Decryption**.
- To edit policy settings, including the list of certificates used in the policy, click the **SSL Decryption Settings** button (). You can also download the certificate used with decrypt re-sign rules so that you can distribute it to clients. See the following topics:
  - Configure Certificates for Known Key and Re-Sign Decryption, on page 17
  - Downloading the CA Certificate for Decrypt Re-Sign Rules, on page 18
- To configure rules:
  - To create a new rule, click the + button. See Configure SSL Decryption Rules, on page 10.
  - To edit an existing rule, click the edit icon () for the rule (in the Actions column). You can also selectively edit a rule property by clicking on the property in the table.
  - To delete a rule you no longer need, click the delete icon () for the rule (in the Actions column).
- To move a rule, edit it and select the new location from the **Order** drop-down list.
- If any rules have problems, for example, because of removed or changed URL categories, click the **See Problem Rules** link next to the search box to filter the table to show only those rules. Please edit and correct (or delete) these rules, so that they will provide the service that you require.

Enable the SSL Decryption Policy

Before you can configure SSL decryption rules, you must enable the policy and configure some basic settings. The following procedure explains how to enable the policy directly. You can also enable it when you enable identity policies. Identity policies require that you enable the SSL decryption policy.

Before you begin

If you upgraded from a release that did not have SSL decryption policies, but you had configured the identity policy with active authentication rules, the SSL decryption policy is already enabled. Ensure that you select the Decrypt Re-Sign certificate you want to use, and optionally enable pre-defined rules.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

Step 2 Click **Enable SSL Decryption** to configure the policy settings.

- If this is the first time you enabled the policy, the SSL Decryption Configuration dialog box opens. Proceed with the next steps.

- If you have already configured the policy once and then disabled it, the policy is simply enabled again with your previous settings and rules. You can click the **SSL Decryption Settings** button ( ) and configure settings as described in Configure Certificates for Known Key and Re-Sign Decryption, on page 17.

Step 3 In **Decrypt Re-Sign Certificate**, select the internal CA certificate to use for rules that implement decryption with re-signed certificates.

You can use the pre-defined NGFW-Default-InternalCA certificate, or one that you created or uploaded. If the certificate does not yet exist, click **Create Internal CA** to create it.

If you have not already installed the certificate in client browsers, click the download button ( ) to obtain a copy. See the documentation for each browser for information on how to install the certificate. Also see Downloading the CA Certificate for Decrypt Re-Sign Rules, on page 18.

Step 4 (Optional.) Click **+** under **Trusted CA Certificates** and select the certificates or certificate groups that you want the policy to trust.

The default group, Cisco-Trusted-Authorities, includes all of the system-defined trusted CA certificates. If you uploaded additional certificates, you can add them here, or collect them in your own group and select the group here. You can either replace the Cisco-Trusted-Authorities group, or simply add your group. Users will be prompted to accept the certificate for any site whose certificate's signing authority is not represented in this list: access to the site is not blocked simply because the certificate is not trusted.

If you leave the list empty, or select only empty certificate groups, then the SSL decryption policy will trust any certificate.

Step 5 Select the initial SSL decryption rules.

The system includes the following pre-defined rule that you might find useful:

- **Sensitive_Data**—This rule does not decrypt traffic that matches web sites in the Financial Services or Health and Medicine URL categories, which include banks, healthcare services, and so forth. You must enable the URL license to implement this rule.

Step 6 Click **Enable**.

Configure the Default SSL Decryption Action

If an encrypted connection does not match a specific SSL decryption rule, it is handled by the default action for the SSL decryption policy.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

Step 2 Click anywhere in the **Default Action** field.

Step 3 Select the action to apply to matching traffic.

- **Do Not Decrypt**—Allow the encrypted connection. The access control policy then evaluates the encrypted connection and drops or allows it based on access control rules.

- **Block**—Drop the connection immediately. The connection is not passed on to the access control policy.

Step 4 (Optional.) Configure logging for the default action.

You must enable logging for traffic that matches the default action to be included in dashboard data or Event Viewer. Select from these options:

- **At End of Connection**—Generate an event at the conclusion of the connection.
  - **Send Connection Events To**—If you want to send a copy of the events to an external syslog server, select the server object that defines the syslog server. If the required object does not already exist, click **Create New Syslog Server** and create it. (To disable logging to a syslog server, select **Any** from the server list.)

    Because event storage on the device is limited, sending events to an external syslog server can provide more long term storage and enhance your event analysis.

- **No Logging**—Do not generate any events.

Step 5 Click **Save**.

Configure SSL Decryption Rules

Use SSL decryption rules to determine how to handle encrypted connections. Rules in the SSL decryption policy are evaluated from top to bottom. The rule applied to traffic is the first one where all the traffic criteria are matched.

You can create and edit rules in the SSL Native Rules section only.

Note Traffic for your VPN connections (both site-to-site and remote access) is decrypted before the SSL decryption policy evaluates connections. Thus, SSL decryption rules are never applied to VPN connections, and you do not need to consider VPN connections when creating these rules. However, any use of encrypted connections within a VPN tunnel are evaluated. For example, an HTTPS connection to an internal server through an RA VPN connection is evaluated by SSL decryption rules, even though the RA VPN tunnel itself is not (because it is decrypted already).

Before you begin

If you are creating a decrypt known-key rule, ensure that you upload the certificate and key for the destination server (as an internal certificate) and also edit the SSL decryption policy settings to use the certificate. Known-key rules typically specify the destination server in the destination network criteria of the rule. For more information, see Configure Certificates for Known Key and Re-Sign Decryption, on page 17.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

If you have not configured any SSL decryption rules (other than the ones automatically generated for active authentication identity rules), you can add pre-defined rules by clicking **Add Pre-Defined Rules**. You are prompted to select the rules that you want.

Step 2 Do any of the following:

- To create a new rule, click the + button.
- To edit an existing rule, click the edit icon ( ) for the rule.

To delete a rule you no longer need, click the delete icon ( ) for the rule.

Step 3 In **Order**, select where you want to insert the rule in the ordered list of rules.

You can insert rules into the **SSL Native Rules** section only. The Identity Policy Active Authentication Rules are automatically generated from your identity policy and are read-only.

Rules are applied on a first-match basis, so you must ensure that rules with highly specific traffic matching criteria appear above policies that have more general criteria that would otherwise apply to the matching traffic.

The default is to add the rule to the end of the list. If you want to change a rule's location later, edit this option.

Step 4 In **Title**, enter a name for the rule.

The name cannot contain spaces. You can use alphanumeric characters and these special characters: + . _ -

Step 5 Select the action to apply to matching traffic.

For a detailed discussion of each option, see the following:

- Decrypt Re-Sign, on page 2
- Decrypt Known Key, on page 3
- Do Not Decrypt, on page 3
- Block, on page 4

Step 6 Define the traffic matching criteria using any combination of the following tabs:

- **Source/Destination**—The security zones (interfaces) through which the traffic passes, the IP addresses or the country or continent (geographical location) for the IP address, or the TCP ports used in the traffic. The default is any zone, address, geographical location, and TCP port. See Source/Destination Criteria for SSL Decryption Rules, on page 12.
- **Application**—The application, or a filter that defines applications by type, category, tag, risk, or business relevance. The default is any encrypted application. See Application Criteria for SSL Decryption Rules, on page 13.
- **URL**—The URL category of a web request. The default is that the URL category and reputation are not considered for matching purposes. See URL Criteria for SSL Decryption Rules, on page 14.
- **Users**—The identity source, user or user group. Your identity policies determine whether user and group information is available for traffic matching. You must configure identity policies to use this criteria. See User Criteria for SSL Decryption Rules, on page 15.
- **Advanced**—The characteristics derived from the certificates used in the connection, such as SSL/TLS version and certificate status. See Advanced Criteria for SSL Decryption Rules, on page 16.

To modify a condition, you click the + button within that condition, select the desired object or element, and click **OK** in the popup dialog box. If the criterion requires an object, you can click **Create New *Object*** if the object you require does not exist. Click the **x** for an object or element to remove it from the policy.

When adding conditions to SSL decryption rules, consider the following tips:

- You can configure multiple conditions per rule. Traffic must match all the conditions in the rule for the rule to apply to traffic. For example, you can use a single rule to decrypt traffic based on URL category.
- For each condition in a rule, you can add up to 50 criteria. Traffic that matches any of a condition's criteria satisfies the condition. For example, you can use a single rule to apply application control for up to 50 applications or application filters. Thus, there is an OR relationship among the items in a single condition, but an AND relationship between condition types (for example, between source/destination and application).
- Matching URL category requires the URL filtering license.

Step 7 (Optional.) Configure logging for the rule.

You must enable logging for traffic that matches the rule to be included in dashboard data or Event Viewer. Select from these options:

- **At End of Connection**—Generate an event at the conclusion of the connection.
  - **Send Connection Events To**—If you want to send a copy of the events to an external syslog server, select the server object that defines the syslog server. If the required object does not already exist, click **Create New Syslog Server** and create it. (To disable logging to a syslog server, select Any from the server list.)

    Because event storage on the device is limited, sending events to an external syslog server can provide more long term storage and enhance your event analysis.
- **No Logging**—Do not generate any events.

Step 8 Click **OK**.

Source/Destination Criteria for SSL Decryption Rules

The Source/Destination criteria of an SSL decryption rule define the security zones (interfaces) through which the traffic passes, the IP addresses or the country or continent (geographical location) for the IP address, or the TCP ports used in the traffic. The default is any zone, address, geographical location, and any TCP port. TCP is the only protocol matched to SSL decryption rules.

To modify a condition, you click the + button within that condition, select the desired object or element, and click **OK**. If the criterion requires an object, you can click **Create New *Object*** if the object you require does not exist. Click the **x** for an object or element to remove it from the policy.

You can use the following criteria to identify the source and destination to match in the rule.

Source Zones, Destination Zones

The security zone objects that define the interfaces through which the traffic passes. You can define one, both, or neither criteria: any criteria not specified applies to traffic on any interface.

- To match traffic leaving the device from an interface in the zone, add that zone to the **Destination Zones**.

- To match traffic entering the device from an interface in the zone, add that zone to the **Source Zones**.
- If you add both source and destination zone conditions to a rule, matching traffic must originate from one of the specified source zones and egress through one of the destination zones.

Use this criteria when the rule should apply based on where the traffic enters or exits the device. For example, if you want to ensure that all traffic going from outside hosts to inside hosts gets decrypted, you would select your outside zone as the **Source Zones** and your inside zone as the **Destination Zones**.

Source Networks, Destination Networks

The network objects or geographical locations that define the network addresses or locations of the traffic.

- To match traffic from an IP address or geographical location, configure the **Source Networks**.
- To match traffic to an IP address or geographical location, configure the **Destination Networks**.
- If you add both source and destination network conditions to a rule, matching traffic must originate from one of the specified IP addresses and be destined for one of the destination IP addresses.

When you add this criteria, you select from the following tabs:

- **Network**—Select the network objects or groups that define the source or destination IP addresses for the traffic you want to control.



Note For Decrypt Known-Key rules, select an object with the IP address of the destination server that uses the certificate and key you uploaded.

- **Geolocation**—Select the geographical location to control traffic based on its source or destination country or continent. Selecting a continent selects all countries within the continent. Besides selecting geographical location directly in the rule, you can also select a geolocation object that you created to define the location. Using geographical location, you could easily restrict access to a particular country without needing to know all of the potential IP addresses used there.

Source Ports, Destination Ports/Protocols

The port objects that define the protocols used in the traffic. You can specify TCP protocol and ports only for SSL decryption rules.

- To match traffic from a TCP port, configure the **Source Ports**.
- To match traffic to a TCP port, configure the **Destination Ports/Protocols**.
- To match traffic both originating from specific TCP ports and destined for specific TCP ports, configure both. For example, you could target traffic from port TCP/80 to port TCP/8080.

Application Criteria for SSL Decryption Rules

The Application criteria of an SSL decryption rule defines the application used in an IP connection, or a filter that defines applications by type, category, tag, risk, or business relevance. The default is any application that has the SSL Protocol tag. You cannot match SSL decryption rules to any non-encrypted application.

Although you can specify individual applications in the rule, application filters simplify policy creation and administration. For example, you could create an SSL decryption rule that decrypts or blocks all high risk, low business relevance applications. If a user attempts to use one of those applications, the session is decrypted or blocked.

In addition, Cisco frequently updates and adds additional application detectors via system and vulnerability database (VDB) updates. Thus, a rule for high risk applications can automatically apply to new applications without you having to update the rule manually.

You can specify applications and filters directly in the rule, or create application filter objects that define those characteristics. The specifications are equivalent, although using objects can make it easier to stay within the 50-items-per-criteria system limit if you are creating a complex rule.

To modify the application and filters list, you click the **+** button within the condition, select the desired applications or application filter objects, which are listed on separate tabs, and click **OK** in the popup dialog box. On either tab, you can click **Advanced Filter** to select filter criteria or to help you search for specific applications. Click the **x** for an application, filter, or object to remove it from the policy. Click the **Save As Filter** link to save the combined criteria that is not already an object as a new application filter object.

For more information about the application criteria and how to configure advanced filters and select applications, see Configuring Application Filter Objects.

Consider the following tips when using application criteria in SSL decryption rules.

- The system can identify unencrypted applications that become encrypted using StartTLS. This includes such applications as SMTPS, POPS, FTPS, TelnetS, and IMAPS. In addition, it can identify certain encrypted applications based on the Server Name Indication in the TLS ClientHello message, or the server certificate subject distinguished name value.
- The system can identify the application only after the server certificate exchange. If traffic exchanged during the SSL handshake matches all other conditions in an SSL rule containing an application condition but the identification is not complete, the SSL policy allows the packet to pass. This behavior allows the handshake to complete so that applications can be identified. After the system completes its identification, the system applies the SSL rule action to the remaining session traffic that matches its application condition.
- If a selected application was removed by a VDB update, "(Deprecated)" appears after the application name. You must remove these applications from the filter, or subsequent deployments and system software upgrades will be blocked.

URL Criteria for SSL Decryption Rules

The URL criteria of an SSL decryption rule defines the category to which the URL in a web request belongs. You can also specify the relative reputation of sites to decrypt, block, or allow without decryption. The default is to not match connections based on URL categories.

For example, you could block all encrypted Gambling sites, or decrypt untrusted Social Networking sites. If a user attempts to browse to any URL with that category and reputation combination, the session is blocked or decrypted. For more information on URL category matching, see Filtering URLs by Category and Reputation.

Categories Tab

Click **+**, select the desired categories, and click **OK**. Click the **x** for a category to remove it from the policy.

The default is to apply the rule to all URLs in each selected category regardless of reputation. To limit the rule based on reputation, click the down arrow for each category, deselect the **Any** checkbox, and

then use the **Reputation** slider to choose the reputation level. The left of the reputation slider indicates sites that will be allowed without decryption, the right side are sites that will be decrypted or blocked. How reputation is used depends on the rule action:

- If the rule decrypts or blocks connections, selecting a reputation level also selects all reputations more severe than that level. For example, if you configure a rule to decrypt or block **Questionable sites** (level 2), it also automatically decrypts or blocks **Untrusted** (level 1) sites.
- If the rule allows connections without decryption (do not decrypt), selecting a reputation level also selects all reputations less severe than that level. For example, if you configure a rule to not decrypt **Favorable sites** (level 4), it also automatically does not decrypt **Trusted** (level 5) sites.

Select the **Include Sites with Unknown Reputation** option to have URLs with unknown reputation included in the reputation match. New sites typically are unrated, and there can be other reasons a site's reputation is unknown or cannot be determined.

Check the Category for a URL

You can check on the category and reputation for a particular URL. Enter the URL in the **URL to Check** box and click **Go**. You will be taken to an external website to see the results. If you disagree with a categorization, click the **Submit a URL Category Dispute** link and let us know.

User Criteria for SSL Decryption Rules

The User criteria of an SSL decryption rule defines the user or user group for an IP connection. You must configure identity policies and the associated directory server to include user or user group criteria in a rule.

Your identity policies determine whether user identity is collected for a particular connection. If identity is established, the IP address of the host is associated with the identified user. Thus, traffic whose source IP address is mapped to a user is considered to be from that user. IP packets themselves do not include user identity information, so this IP-address-to-user mapping is the best approximation available.

Because you can add a maximum of 50 users or groups to a rule, selecting groups usually makes more sense than selecting individual users. For example, you could create a rule that decrypts traffic to the Engineering group that comes from the outside network, and create a separate rule that does not decrypt outgoing traffic from that group. Then, to make the rule apply to new engineers, you only need to add the engineer to the Engineering group in the directory server.

You an also select identity sources to apply to all users within that source. Thus, if you support multiple Active Directory domains, you can provide differential decryption based on the domain.

To modify the users list, you click the **+** button within the condition and select the desired users or user groups using one of the following techniques. Click the **x** for a user or group to remove it from the policy.

- **Identity Sources**—Select an identity source, such as an AD realm or the local user database, to apply the rule to all users obtained from the selected sources. If the realm you need does not yet exist, click **Create New Identity Realm** and create it now.
- **Groups**—Select the desired user groups. Groups are available only if you configure groups in the directory server. If you select a group, the rule applies to any member of the group, including subgroups. If you want to treat a sub-group differently, you need to create a separate access rule for the sub-group and place it above the rule for the parent group in the access control policy.
- **Users**—Select individual users. The user name is prefixed with the identity source, such as Realm\username.

  There are some built-in users under the Special-Identities-Realm:

- **Failed Authentication**—The user was prompted to authenticate, but failed to enter a valid username/password pair within the maximum number of allowed attempts. Failure to authenticate does not itself prevent the user from accessing the network, but you can write an access rule to limit network access for these users.
- **Guest**—Guest users are like Failed Authentication users, except that your identity rule is configured to call these users Guest. Guest users were prompted to authenticate and failed to do so within the maximum number of attempts.
- **No Authentication Required**—The user was not prompted to authentication, because the user's connections matched identity rules that specified no authentication.
- **Unknown**—There is no user mapping for the IP address, and there is no record of failed authentication yet. Typically, this means that no HTTP traffic has yet been seen from that address.

Advanced Criteria for SSL Decryption Rules

The Advanced traffic matching criteria relate to characteristics derived from the certificates used in the connection. You can configure any or all of the following options.

Certificate Properties

Traffic matches the certificate properties option of the rule if it matches any of the selected properties. You can configure the following:

Certificate Status

Whether the certificate is **Valid** or **Invalid**. Select **Any** (the default) if you do not care about certificate status.

A certificate is considered valid if all of the following conditions are met, otherwise it is invalid:

- The policy trusts the CA that issued the certificate.
- The certificate's signature can be properly validated against the certificate's content.
- The issuer CA certificate is stored in the policy's list of trusted CA certificates.
- None of the policy's trusted CAs revoked the certificate.
- The current date is between the certificate Valid From and Valid To dates.

Self-Signed

Whether the server certificate contains the same subject and issuer distinguished name. Select one of the following:

- **Self-Signing**—The server certificate is self-signed.
- **CA-Signing**—The server certificate is signed by a Certificate Authority. That is, the issuer and subject are not the same.
- **Any**—Do not consider whether the certificate is self-signed as a match criteria.

Supported Version

The SSL/TLS version to match. The rule applies to traffic that uses the any of the selected versions only. The default is all versions. Select from: **SSL 3.0**, **TLS 1.0**, **TLS 1.1**, **TLS 1.2**.

For example, if you wanted to permit TLSv1.2 connections only, you could create a block rule for the lower versions.

Traffic that uses any version not listed, such as SSL v2.0, is handled by the default action for the SSL decryption policy.

Configure Certificates for Known Key and Re-Sign Decryption

If you implement decryption, either by re-signing or using known keys, you need to identify the certificates that the SSL decryption rules can use. Ensure that all certificates are valid and unexpired.

Especially for known-key decryption, you need to ensure that the system has the current certificate and key for each destination server whose connections you are decrypting. With a decrypt known key rule, you use the actual certificate and key from the destination server for decryption. Thus, you must ensure that the Firewall Threat Defense device has the current certificate and key at all times, or decryption will be unsuccessful.

Upload a new internal certificate and key whenever you change the certificate or key on the destination server in a known key rule. Upload them as an internal certificate (not an internal CA certificate). You can upload the certificate during the following procedure, or go to the **Objects** > **Certificates** page and upload it there.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

Step 2 Click the **SSL Decryption Settings** button ( ).

Step 3 In **Decrypt Re-Sign Certificate**, select the internal CA certificate to use for rules that implement decryption with re-signed certificates.

You can use the pre-defined NGFW-Default-InternalCA certificate, or one that you created or uploaded. If the certificate does not yet exist, click **Create Internal CA** to create it.

If you have not already installed the certificate in client browsers, click the download button ( ) to obtain a copy. See the documentation for each browser for information on how to install the certificate. Also see Downloading the CA Certificate for Decrypt Re-Sign Rules, on page 18.

Step 4 For each rule that decrypts using a known key, upload the internal certificate and key for the destination server.

a) Click + under **Decrypt Known-Key Certificates**.
b) Select the internal identity certificate, or click **Create New Internal Certificate** to upload it now.
c) Click **OK**.

Step 5 (Optional.) Click + under **Trusted CA Certificates** and select the certificates or certificate groups that you want the policy to trust.

The default group, Cisco-Trusted-Authorities, includes all of the system-defined trusted CA certificates. Following are the main cases where you might want to change this setting:

- You want to use trusted CA certificates that are not in the default group. You would then select both the default group and your new group in the SSL decryption policy settings. You might do this if you have uploaded additional trusted CA certificates.
- You want to use a more limited list of trusted CA certificates than what is in the default group. You would then create a group that has a complete list of trusted certificates, not just your delta, and select it as the sole group in the SSL decryption policy settings.

Users will be prompted to accept the certificate for any site whose certificate's signing authority is not represented in this list: access to the site is not blocked simply because the certificate is not trusted.

If you leave the list empty, or select only empty certificate groups, then the SSL decryption policy will trust any certificate.

Step 6 Click **Save**.

Downloading the CA Certificate for Decrypt Re-Sign Rules

If you decide to decrypt traffic, users must have the internal CA certificate that is used in the encryption process defined as a Trusted Root Certificate Authority in their applications that use TLS/SSL. Typically if you generate a certificate, or sometimes even if you import one, the certificate is not already defined as trusted in these applications. By default in most web browsers, when users send HTTPS requests, they will see a warning message from the client application informing them that there is a problem with the web site's security certificate. Usually, the error message says that the web site's security certificate was not issued by a trusted certificate authority or the web site was certified by an unknown authority, but the warning might also suggest there is a possible man-in-the-middle attack in progress. Some other client applications do not show this warning message to users nor allow users to accept the unrecognized certificate.

You have the following options for providing users with the required certificate:

Inform users to accept the root certificate

You can inform the users in your organization what the new policies are at the company and tell them to accept the root certificate supplied by the organization as a trusted source. Users should accept the certificate and save it in the Trusted Root Certificate Authority storage area so that they are not prompted again the next time they access the site.



Note The user needs to accept and trust the CA certificate that created the replacement certificate. If they instead simply trust the replacement server certificate, they will continue to see warnings for each different HTTPS site that they visit.

Add the root certificate to client devices

You can add the root certificate to all client devices on the network as a trusted root certificate authority. This way, the client applications automatically accept transactions with the root certificate.

You can either make the certificate available to users by E-mailing it or placing it on a shared site, or you could incorporate the certificate into your corporate workstation image and use your application update facilities to distribute it automatically to users.

The following procedure explains how to download the internal CA certificate and install it on Windows clients.

Procedure

Step 1 Download the certificate from the Firewall Device Manager.

a) Select **Policies** > **SSL Decryption**.

b) Click the **SSL Decryption Settings** button ().

c) Click the **Download** button ().

d) Select a download location, optionally change the file name (but not the extension), and click **Save**.

You can now cancel out of the SSL Decryption Settings dialog box.

Step 2 Install the certificate in the Trusted Root Certificate Authority storage area in web browsers on client systems, or make it available for clients to install themselves.

The process differs depending on the operating system and type of browser. For example, you can use the following process for Internet Explorer and Chrome running on Windows. (For Firefox, install through the **Tools** > **Options** > **Advanced** page.)

a) From the **Start** menu, select **Control Panel** > **Internet Options**.

b) Select the **Content** tab.

c) Click the **Certificates** button to open the Certificates dialog box.

d) Select the **Trusted Root Certificate Authorities** tab.

e) Click **Import**, and follow the wizard to locate and select the downloaded file (<uuid>_internalCA.crt) and add it to the Trusted Root Certificate Authorities store.

f) Click **Finish**.

Messages should indicate that the import was successful. You might see an intermediate dialog box warning you that Windows could not validate the certificate if you generated a self-signed certificate rather than obtaining one from a well-known third-party Certificate Authority.

You can now close out the Certificate and Internet Options dialog boxes.

Example: Blocking Older SSL/TLS Versions from the Network

Some organizations are required to prevent the use of older versions of SSL or TLS either by government regulation or company policy. You can use the SSL Decryption policy to block traffic that uses an SSL/TLS version that you prohibit. Consider placing this rule at the top of the SSL Decryption policy to ensure that you catch the prohibited traffic immediately.

The following example blocks all SSL 3.0 and TLS 1.0 connections.

Before you begin

This procedure assumes you have already enabled the SSL Decryption policy as explained in Enable the SSL Decryption Policy, on page 8.

Procedure

Step 1 Select **Policies** > **SSL Decryption**.

Step 2 Click the + button to create a new rule.

Step 3 In Order, select **1** to place the rule at the top of the policy, or select the number most suitable for your network.

The default is to add the rule at the end of the policy.

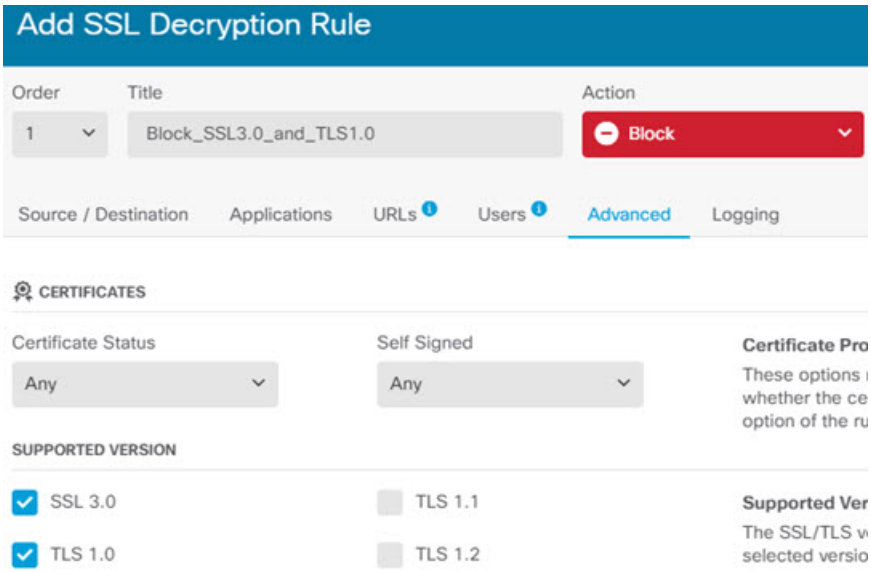
Step 4 In **Title**, enter a name for the rule, for example, Block_SSL3.0_and_TLS1.0.

Step 5 In **Action**, select **Block**. This will immediately drop any traffic that matches the rule.

Step 6 Leave the default values for all options on the following tabs: **Source/Destination**, **Applications**, **URLs**, **Users**.

Step 7 Click the **Advanced** tab and under **Supported Versions**, leave SSL 3.0 and TLS 1.0 selected, but uncheck TLS 1.1, TLS 1.2.

The policy should look like the following:



Step 8 (Optional) Click the **Logging** tab and select **At End of Connection** if you want to dashboards and events to reflect blocked connections. You can also select an external syslog server if you are using one.

Step 9 Click **OK**.

You can now deploy the policy. Once deployed, any SSL 3.0 or TLS 1.0 connection that goes through the system will be dropped.

Note
SSL 2.0 connections are handled by the default action for the policy. If you want to ensure these are also dropped, change the default action to Block.

What to do next

If you implement this rule, we have the following recommendations:

- For any type of decrypt rule, leave the default settings on the Advanced tab, where all SSL/TLS options are selected. By applying to all versions, the handshake process is simplified. However, your initial block rule will still prevent SSL 3.0 and TLS 1.0 connections.
- We normally recommend that you use Do Not Decrypt as the default action for the policy. However, because SSL 2.0 connections are always handled by the default action, you might want to use Block

instead. However, if you want to apply Do Not Decrypt as the default action for all decryptable traffic, create a Do Not Decrypt rule at the end of the policy where you accept all default values for traffic matching criteria. This rule would match any supported TLS connection that does not match an earlier rule in the table, and act as the default for those TLS versions.

Monitoring and Troubleshooting SSL Decryption

The following topics explain how to monitor and troubleshoot SSL decryption policies.

Monitoring SSL Decryption

You can view information about decryption in the dashboards and events for traffic that matches rules (or the default action) for which you enabled logging.

SSL Decryption Dashboard

To evaluate overall decryption statistics, view the **Monitoring** > **SSL Decryption** dashboard. The dashboard shows the following information:

- Percentage of encrypted versus plain text traffic.
- How much encrypted traffic is decrypted per SSL rules.

Events

In addition to the dashboard, the event viewer (**Monitoring** > **Events**) includes SSL information for encrypted traffic. Following are some tips in evaluating events:

- For connections that were dropped because they matched an SSL rule (or default action) that blocked matching traffic, the **Action** should be "Block," and the **Reason** should indicate "SSL Block."
- The **SSL Actual Action** field indicates the actual action that the system applied to the connection. This can differ from the **SSL Expected Action**, which indicates the action defined on the matching rule. For example, a connection might match a rule that applies decryption, but could not be decrypted for some reason.

Handling Web Sites Where Decrypt Re-sign Works for a Browser but not an App (SSL or Certificate Authority Pinning)

Some apps for smart phones and other devices use a technique called SSL (or Certificate Authority) pinning. The SSL pinning technique embeds the hash of the original server certificate inside the app itself. As a result, when the app receives the resigned certificate from the Firewall Threat Defense device, the hash validation fails and the connection is aborted.

The primary symptom is that users cannot connect to the web site using the site's app, but they can connect using the web browser, even when using the browser on the same device where the app fails. For example, users cannot use the Facebook iOS or Android app, but they can point Safari or Chrome at https://www.facebook.com and make a successful connection.

Because SSL pinning is specifically used to avoid man-in-the-middle attacks, there is no workaround. You must choose between the following options:

- Support app users, in which case you cannot decrypt any traffic to the site. Create a Do Not Decrypt rule for the site's application (on the Application tab for the SSL Decryption rule) and ensure that the rule comes before any Decrypt Re-sign rule that would apply to the connections.
- Force users to use browsers only. If you must decrypt traffic to the site, you will need to inform users that they cannot use the site's app when connecting through your network, that they must use their browsers only.

More Details

If a site works in a browser but not in an app on the same device, you are almost certainly looking at an instance of SSL pinning. However, if you want to delve deeper, you can use connection events to identify SSL pinning in addition to the browser test.

There are two ways an app might deal with hash validation failures:

- Group 1 apps, such as Facebook, send an SSL ALERT Message as soon as it receives the SH, CERT, SHD message from the server. The Alert is usually an "Unknown CA (48)" alert indicating SSL Pinning. A TCP Reset is sent following the Alert message. You should see the following symptoms in the event details:
  - SSL Flow Flags include ALERT_SEEN.
  - SSL Flow Flags do not include APP_DATA_C2S or APP_DATA_S2C.
  - SSL Flow Messages typically are: CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE, SERVER_KEY_EXCHANGE, SERVER_HELLO_DONE.



Note If you enabled SSL hardware acceleration, you would also see the CLIENT_ALERT message.

- Group 2 apps, such as Dropbox, do not send any alerts. Instead they wait until the handshake is done and then send a TCP Reset. You should see the following symptoms in the event:
  - SSL Flow Flags do not include ALERT_SEEN, APP_DATA_C2S, or APP_DATA_S2C.
  - SSL Flow Messages typically are: CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE, SERVER_KEY_EXCHANGE, SERVER_HELLO_DONE, CLIENT_KEY_EXCHANGE, CLIENT_CHANGE_CIPHER_SPEC, CLIENT_FINISHED, SERVER_CHANGE_CIPHER_SPEC, SERVER_FINISHED.