TippingPoint™

SSL Inspection

User Guide

Privacy and Personal Data Collection Disclosure

Certain features available in Trend Micro products collect and send feedback regarding product usage and detection information to Trend Micro. Some of this data is considered personal in certain jurisdictions and under certain regulations. If you do not want Trend Micro to collect personal data, you must ensure that you disable the related features.

The following link outlines the types of data that the Security Management System collects and provides detailed instructions on how to disable the specific features that feedback the information.

https://success.trendmicro.com/data-collection-disclosure

Data collected by Trend Micro is subject to the conditions stated in the Trend Micro Privacy Policy:

https://www.trendmicro.com/en_us/about/legal/privacy-policy-product.html

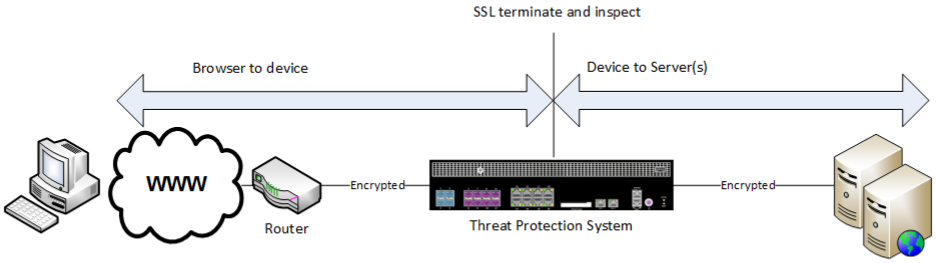
Legal Notice

Publication: September 2020

Overview

The Threat Protection System (TPS) security device provides in-line, real-time threat protection for inbound SSL traffic to your web servers and outbound SSL traffic from your clients.

With access to your server certificate and private key, SSL server inspection receives and decrypts SSL data, inspects it using the Threat Suppression Engine, and then encrypts it before sending it to the actual destination. The TPS manages its own private keys and certificates from the servers it is securing; you can either store them on the device or access them at run-time using the Security Management System (SMS).



SSL client inspection decrypts and inspects SSL sessions from your internal clients to the server. The SSL client proxy can use a signing certificate (a CA certificate) to authenticate the SSL session with your client.

SSL sessions involve a negotiation of shared information as in the following sequence:

1. A client browser requests access to a secure site.
2. The secure site's server responds with a public key in addition to an SSL certificate with information about the owner of that site.
3. The client inspects the certificate to ensure its validity (generated by a certificate authority, not expired, and so on).
4. The client generates a random symmetric key, which it uses to encrypt private information such as its IP address.
5. With the public key it received previously, the client encrypts this symmetric key and sends it back to the secure site.
6. The secure site's server decrypts the symmetric key with its own private key to get the client's IP address and other private data.
7. The secure site's server then sends to the client the encrypted page along with the symmetric key to decrypt it.
8. The client and the secure site use symmetric key pairs to encrypt and decrypt all subsequent communications.

Considerations

When deploying SSL inspection, consider these points:

CONSIDERATION	DESCRIPTION
IPv4 traffic only	The TPS inspects inbound IPv4 traffic—including HTTP and HTTPS traffic—to your secure servers, and outbound client IPv4 traffic to the Internet. SSL inspection does not support IPv6 traffic, including IPv4 over IPv6 tunneling.

Consideration	Description
Tunneled traffic	Supported SSL encapsulations: • GRE (Generic Routing Encapsulation)* • IPv4 (IP-in-IP) • One layer of tunneling only for both GRE and IPv4-in-IPv4. SSL inspection does not include support for GTP or IPv6 encapsulations. *GRE support includes the mandatory GRE fields. Optional GRE key configuration is also supported but requires the key to be the same value for both directions. SSL inspection does not support other optional GRE fields, such as GRE sequence number.
Asymmetric mode	SSL inspection cannot occur when asymmetric mode is enabled.
Quarantine hosts and redirecting HTTP traffic to another site	When configuring an Action Set to quarantine hosts, if you also configure the response to redirect traffic to an HTTP server, the device redirects the HTTP traffic from the quarantined host but does not redirect the HTTPS traffic.
Filter Precedence	The TPS processes filters in the following order of precedence: 1. Inspection Bypass Rules 2. Traffic Management Filters 3. RepDV 4. Quarantine 5. Digital Vaccine Filters When encrypted traffic is routed through the device and SSL inspection is configured, the TPS order of precedence applies to the decrypted traffic. The TPS does not quarantine or apply Digital Vaccine filters to traffic without first decrypting the traffic. If SSL inspection is not configured, the device applies Inspection Bypass, Traffic Management, RepDV, and quarantine filtering against the encrypted traffic. The device applies Digital Vaccine filters, but they do not match against encrypted payload.
Server Name Indication (SNI)	The TPS forwards any SNI sent by the client to the server during the TLS handshake. This enables the support of HTTPS websites that host multiple sites with possibly different TLS/SSL certificates under the same IP address. You can configure multiple certificates in a server proxy for use with SSL server inspection. To avoid decrypting domains that are not intended to be decrypted, the default behavior is to trust SNI when it matches a domain that is configured *not* to be decrypted rather than using the server certificate to make this decision. This avoids issues when accessing a site for the first time and when a site has many IP addresses and subdomains. Contact support for information on how to change this default behavior.
Non-encrypted traffic when SSL is configured	• The TPS device drops non-encrypted traffic flows that match a configured SSL server tuple (destination port and destination IP address) in the SSL profile but send cleartext traffic before starting an SSL handshake (as some protocols allow via STARTTLS). • The TPS device drops non-encrypted traffic flows that match a configured SSL server tuple (destination port and destination IP address) in the SSL profile due to the lack of an SSL handshake.

Consideration	Description
Conditions when encrypted traffic will not be decrypted	When you configure SSL, encypted traffic will be decrypted except when the system or SSL logs indicate the following conditions: • **Decrypted but not inspected** – Initial decryption of flow determined that no subsequent flows to this domain require decryption or inspection. • **Not decrypted** – Flow not decrypted because the domain name matches a category or domain exception. • **Bypassed** – Flow not decrypted because SSL requirements were not met. *Learn more* about SSL requirements. • **Blocked** – Flow that should be decrypted has been blocked.
Security properties for an end-to-end SSL connection	SSL negotiation on the client side of the TPS proxy is done independently from the SSL negotiation on the server side of the TPS proxy. Consequently, supported versions and cipher suites between the two sides of the TPS proxy might not be the same. Each side can enforce only the rules on its half of the negotiation. Administrators can check details of the negotiation by enabling logging and examining the SSL logs.
Traffic Management filters Trust action	The TPS device continues to proxy the SSL session between the client and the server when HTTPS traffic matches a traffic management filter that is set to Trust (incoming traffic is trusted and not inspected).
Packet trace	Packet Trace as an action includes the decrypted traffic.
Traffic capture	Traffic capture by **`tcpdump`** does not include the decrypted contents.
L2FB/ZPHA	The TPS device will not clear proxied SSL sessions when the device enters Intrinsic HA Layer-2 Fallback or Zero Power High Availability (ZPHA). To clear proxied SSL sessions, a `debug` command is required in conjunction with support.
Encrypt-then-MAC	This extension is disabled by default. You can enable this extension using the INI file, but performance of the TPS will be affected.

Requirements

Make sure your environment meets the following requirements:

- Device support – the following TPS devices support SSL inspection:
  - TX Series (5500TX, 8200TX, and 8400TX devices)
  - 2200T device
  - Virtual Threat Protection System (vTPS) (performance mode only, with RDRAND instruction recommended) security devices. For information about how to deploy the vTPS virtual appliance for SSL inspection, see the *vTPS User Guide*.

> **Note**
>
> If you need to configure SSL v3.0 with TLS, always configure all TLS protocols, including TLS v1.0, TLS v1.1, TLS v1.2, and TLS v1.3. You must enable all SSL protocols between the configured lowest-strength and highest-strength SSL protocols.

- Licensing – SSL Inspection is licensed separately. To request an SSL Inspection license, contact your sales representative.

- Cipher suite support – The SMS is capable of configuring the following ciphers if your TOS supports them. Older TOS versions might have limited cipher support. Profile distribution extended status alerts you to any errors.
  - Protocols:
    - TLS v1.3, v1.2, v1.1, and v1.0 (enabled by default)
    - SSL v3.0 (disabled by default)

    Note

    TLS Heartbeat Extension (https://tools.ietf.org/html/rfc6520) is not supported.
  - Key exchange algorithms:
    - Ephemeral Elliptic Curve Diffie-Hellman with RSA signatures (ECDHE-RSA).

      The ECDHE-RSA cipher suite extends SSL inspection capability to Perfect Forward Secrecy (PFS). ECDHE-RSA is enabled by default.
    - RSA (enabled by default)
  - Authentication algorithm:
    - RSA (enabled by default)
  - Bulk encryption algorithms:
    - AES256 and AES128 (enabled by default)
    - 3DES (enabled by default)
    - GCM (enabled by default)
    - DES (disabled by default)
  - Message Authentication Code (MAC) algorithms:
    - SHA384, SHA256, and SHA1 (enabled by default)
    - CHACHA20 (enabled by default)
    - POLY1305 (enabled by default)
- VLAN translation – SSL inspection requires that you do not configure VLAN translation on the device.
- Asymmetric Network mode – SSL inspection requires that you do not enable Asymmetric Network mode on the device.

License the device

Update your license package to assign an available SSL inspection license to any supported TPS security device. The SSL inspection license applies to both SSL client inspection and SSL server inspection.

Note

Manage your license package by using the License Manager on the TMC. When you log in to the TMC, the License Manager is under **My Account** > **License Manager**.

You can configure the SMS to automatically download updates from the TMC. The SMS downloads the most recent license package to the device within 30 minutes. If necessary, manually import the license package.

1. In the SMS tools, click **Admin**.
2. Click **Licensing**.
3. In the **Licensing** workspace, expand your device to view its capabilities.
4. Ensure that the SSL Inspection capability is assigned to the device and is Allowed. If necessary, the SMS prompts you to resolve any issues:
   - **Reboot required** – Reboot the device to enable the SSL inspection license.
   - **Deny –** Verify the license package assigns the SSL inspection capability to the device.

Note

If you intend to configure domain categories for SSL inspection, you must also activate a ThreatDV license. Select **Profiles** > **Reputation Database** > **ThreatDV Entries** to view your license status and details.

Enable SSL inspection

For each device, update the device configuration to enable SSL inspection and optionally, to retain private key information. To optimize the available resources on the device, do not enable SSL inspection until you are ready to inspect secure traffic.

Before you begin

If necessary, you can disable SSL inspection on the device until you configure SSL inspection options.

Procedure

1. In the SMS tools, click **Devices**.
2. Click **All Devices** > ***`device_name`*** > **Device Configuration**.
3. In the **Device Configuration** workspace, click **Edit**.
4. In **SSL Inspection** options, view the supported SSL ciphers on the device.
5. Configure the following options:
   - **Client SSL Inspection** – Enables the SSL client proxy to decrypt and inspect your client traffic to the Internet. If the checkbox is grayed, verify the license package has assigned the SSL Inspection capability to the device.
   - **Server SSL Inspection** – Enables the SSL server proxy to decrypt and inspect Internet traffic to your secure servers. If the checkbox is grayed, verify the license package has assigned the SSL Inspection capability to the device.
   - **Persist Private Keys** – Enables private key information to be retained in the system keystore. By default, the device automatically retrieves private key information from the SMS and temporarily retains the information in memory until the device reboots.

Configure SSL client inspection

Configure SSL client inspection to decrypt the SSL session between a client on your local network and a particular website.

The TPS cannot effectively inspect an encrypted payload of SSL client traffic that does not match the SSL client policy. *Learn more* about configuring an SSL client policy.

The SSL client policy uses a signing certificate to authenticate the session between the SSL client proxy and your client. *Learn more* about generating a signing certficate.

You can configure SSL client inspection so that particular SSL traffic will not be decrypted. During your policy configuration, specify categories that you want exempted from inspection, such as private financial account information. A ThreatDV subscription is required for configuring categories. *Learn more.*

Generate a signing certificate

Signing certificates enable the SSL client proxy to authenticate any SSL server certificate to your client proxy.

For full SSL inspection functionality, you must distribute the signing certificate to the endpoints. In Microsoft Windows environments, use Group Policy Objects (GPO), as described here:

https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/distribute-certificates-to-client-computers-by-using-group-policy

For systems that run Mac or Linux, refer to each vendor's documentation for that operating system's process and requirements.

Although a signing certificate must be a CA certificate, you import it to **Admin** > **Certificate Management** > **Certificates** because you also need its private key. The SMS performs basic validation on the status of the certificates itself. If a certificate is not self-signed, remember to import any intermediate CA and root CA certificates in the signing certificate chain to **Admin** > **Certificate Management** > **CA Certificates**.

All browsers used within the managed environment must trust certificates signed by the signing certificate. Typically, your IT department deploys the signing certificate across all workstations in the company. If the root certificate in the signing certificate chain originates from somewhere other than a public CA source, a valid CA certificate (the associated private key is not necessary) must also be distributed to the clients. For signing certificates that are self-signed, the valid CA certificate to distribute is the signing certificate itself; otherwise, the root CA certificate in the signing certificate chain is the CA certificate to distribute.

When an untrusted site has been accessed, the TPS issues a certificate signed by an untrusted CA certificate. When this happens, or when the proper CA certificate fails to get distributed to the clients, the client might display a warning because of an untrusted certificate chain sent by the TPS.

Because procuring a signing certificate from a public CA can be difficult, you can generate a self-signed certificate for client SSL using the SMS. Self-signed certificates are typically used for personal or test websites. They cost nothing because you do not pay for an established certificate authority (CA) to sign them. Because of this, however, all browsers generate an alert when they encounter a self-signed certificate.

1. In the SMS tools, select **Admin** > **Certificate Management** > **Certificates**.
2. In the Certificates workspace, click **New Signing Certificate**.
3. In the Create Signing Certificate dialog, enter values into the following required fields:
   - **Name** – Name of the Signing Certificate you are creating.
   - **Common Name (CN)** – The common name of the signing certificate.

Optionally, you can also configure values for the following options:

- **Key Size** – Size, in bits, of the key. A minimum size of 2048 bits is recommended.
- **Valid for** – Specify the duration, in days, that the signing certificate is valid.
- **Private key can be exported from SMS** – If you intend to use the private key for other purposes, select this option to enable the key to be exported from the SMS.
- **Email** – Email address of the signing certificate.
- **Organizational Unit (OU)** – Division within the organization. For example, `Research and Development`.
- **Organization** – Legal, registered name of the organization. For example, `National Football League`.
- **Locality** – City in which the organization is located or registered. Do not abbreviate. For example, `Saint Louis`.
- **State** – State or province in which the organization is operating. Indicate the state or province in which the protected server is operating. Do not abbreviate. For example, `Missouri`.
- **Country** – Two-character ISO format country code of the organization. For example, `US`.
- **DNS (DNS Name)** – Additional domain name of the organization to be covered by the certificate. For example, `login.company.com`.
- **User (RFC822 Name)** – E-mail address of specific user in the organization. For example, `susan@company.com`.

Generate a certificate signing request for client SSL

A certificate signing request (CSR) is a certificate prototype that you generate using the SMS. It can be used to get a signing certificate for client SSL. After you configure all the certificate options, you export it to an established certificate authority (CA) who signs it. After it is signed, the CA sends you back the certificate. You then import the signed certificate into your client's certificate repository. This is required for your clients to trust the spoofed server certificates signed by the SSL client proxy.

If your signing request includes a certificate with multiple domains, specify the additional domains in the Subject Alternative Name **DNS (DNS Name)** field of the procedure that follows.

Procedure

1. In the SMS tools, select **Admin** > **Certificate Management** > **Signing Requests**.
2. In the Signing Requests workspace, click **New**.
3. In the New Signing Request dialog, enter values into the following required fields:
   - **Request Name** – Name of the Signing Certificate you are creating.
   - **Common Name (CN)** – The common name for the certificate authority the signing certificate will act as.

   You can also configure values for the following options:

   - **Key Size** – Size, in bits, of the key. A minimum size of 2048 bits is recommended.
   - **Make it a Signing Certificate** – For client SSL, select this option to establish the certificate as a signing certificate.

- **Private key can be exported from SMS** – If you intend to use the private key for other purposes, select this option to enable the key to be exported from the SMS after the certificate is signed.
- **Email** – Email address of the requester.
- **Organizational Unit (OU)** – Division within the organization. For example, `Research and Development`.
- **Organization** – Legal, registered name of the organization. For example, `National Football League`.
- **Locality** – City in which the organization is located or registered. Do not abbreviate. For example, `Saint Louis`.
- **State** – State or province in which the organization is operating. Indicate the state or province in which the protected server is operating. Do not abbreviate. For example, `Missouri`.
- **Country** – Two-character ISO format country code of the organization. For example, `US`.
- **DNS (DNS Name)** – Additional domain name of the organization to be covered by the certificate. For example, `login.company.com`.
- **User (RFC822 Name)** – User email address for the certificate authority organization the signing cert will be acting as.

What to do next

After you create the request, *use the SMS to export* the signing request. Then send the request *without its private key* to a certificate authority to sign and issue the certificate. You will then import the signed certificate that the CA sends you into the SMS as a non-CA certificate so that the private key from the signing request will also be imported.

Export a signing request

After you generate your signing request, export the request to a file so that you can send it to a Certificate Authority (CA) for signing.

Procedure

1. In the SMS tools, select **Admin** > **Certificate Management** > **Signing Requests**.
2. In the Signing Requests workspace, select the Signing Request that you created and click **Export**.
3. In the Export signing request dialog, select the directory location to export the file.
4. Send the file to the CA, or upload it to the CA's portal.

What to do next

The CA signs and issues the certificate. After you receive the signed certificate from the CA, save the certificate file and then *import it* into the SMS as a non-CA certificate so that the private key from the signing request will also be imported. The certificate must be installed on the same SMS that issued the request.

Import a certificate

Import to the SMS the end-point certificate (for server SSL) or the signing certificate (for client SSL) that was signed by a CA from a signing request, or import any existing signing certificates and private keys that browsers on your network already trust.

Procedure

1. In the SMS tools, click **Admin** > **Certificate Management** > **Certificates**.
2. In the Certificates workspace, click **Import**.
3. Provide values for the fields in the Import Certificates dialog. Note the following guidelines:

   Certificate Name

   This name must be unique. If you are replacing an existing certificate, you cannot edit this field. If you are importing multiple certificates using a PKCS12 file, this field indicates the base name for all the certificates. The final certificate name of each certificate in the PKCS12 file is created by appending the alias of the certificate to the base name.

   Certificate File

   Click **Browse** to navigate to the certificate file on your system.

   Certificate Format

   Select the certificate format:

   - **PEM/DER** – Base64/binary encoded certificate. Always import the private key in a separate file. If the private key is encrypted, always enter the appropriate password in the Password box.
   - **PKCS12** – Potentially holds multiple certificates. Enter the appropriate password to import the certificate/key pair. When using a certificate chain, always include any intermediate certificates in your import file. Only one file can be imported with the certificate/private key pair and any intermediate certificates.

   Note

   If the certificates are not self-signed, be sure to also import any intermediate CA and root CA certificates in the certificate chain to **Admin** > **Certificate Management** > **CA Certificates**.

Import CA certificates

Trusted certificates authenticate the SSL session. Both intermediate and root CA certificates are required for the certificate chain of end-point certificates and signing certificates used with SSL inspection that are not self-signed. When you add a root CA to your Trust Store, you can then specify CAs that you want to trust other than the ones that are in the default trusted CA package. This topic explains how to add your own trusted CA certificates.

Procedure

1. In the SMS tools, click **Admin**.
2. Click **Certificate Management** > **CA Certificates**.
3. Click the **CA Certificates** tab to import, export, replace, or delete a CA certificate. If you are importing, specify the following information in the Import Certificate dialog:

a. Enter a certificate name for the certificate being imported.

> **Note**
>
> For the CA certificate name, use alphanumeric (uppercase and lowercase) characters. You can also use hyphens, underscores, and spaces. If the name has a space in it, enclose the entire name in quotes (for example `"cert 34"`). You cannot use any other special characters. Naming the certificate `ca` is not allowed.

b. Click **Browse** to locate the certificate file.

c. Select the certificate format:

- **PEM/DER** – Base64/binary encoded certificate.
- **PKCS12** – Potentially holds multiple certificates, and private keys are ignored. Enter the appropriate password to import the certificate. When using a certificate chain, always include any intermediate CA certificates in your import file. Only one file can be imported with the certificate and any intermediate CA certificates.

4. Click the **Default CA Certificates** tab to import the Default CA Certificates package.

Before you can import the certificates, manually check for updates and then download the package from the TMC (**Releases** > **ThreatDV** > **Default CA Package**).

Configure an SSL trust store

Configure an SSL trust store to establish which certificates to trust in your profile's SSL client inspection policy.

Before you begin

Import to the SMS your user-defined root CA certificates that you want to be trusted. In the SMS tools, click select **Admin** > **Certificate Management** > **CA Certificates** and click **Import**.

> **Note**
>
> If the status of the import states that there is no trust anchor, click **Refresh Status** to ensure that the certificate you imported shows as `Valid` in the Status column.

The trust store establishes how CA certificates are trusted. Only one SSL trust store can be configured per policy. After a trust store is configured in SSL, it cannot be deleted.

> **Note**
>
> If the SMS manages a device that has an existing client trust store, the next profile distribution will overwrite that trust store.

The only way a client knows it can trust a server CA certificate is based on whether the TPS sends the client a trusted signing certificate or an untrusted signing certificate.

- If the TPS sends a trusted signing certificate, the client must have the associated CA certificate in its trust store (or, if the CA is self-signed, the trust store must have the signing certificate itself).
- If the TPS sends an untrusted signing certificate, the client will be informed not to trust that CA certificate.

Procedure

1. In the SMS tools, select **Profles** > **Shared Settings** > **SSL** > **Client** > **Proxy Trust Store**.
2. In the SSL Client Proxy Trust Stores workspace, click **New**.
3. In the Create SSL Client Proxy Trust Store dialog, specify a unique name (required) and a description (optional) for the trust store.
   Limit the name to 64 characters, which can include uppercase and lowercase letters, digits, underscores, hyphens, and spaces. This name must be unique. Limit the description to 512 characters.
4. Click **Add** to specify which of the user-defined root CA certificates that you imported at the beginning of this topic you want your client policy to trust.
   When the **Include Default CA Certifcates** checkbox is selected, all the CA Certificates in the Default CA Certificate package will be used. You can disable the checkbox and, optionally, add or remove other CA certificates to the default list, but you cannot pick or choose individual CA certificates in the Default CA Certificate package.

Add an SSL client proxy

An SSL proxy acts as an intermediary that performs SSL encryption and decryption between your client and the SSL server. You must define an SSL client proxy before you configure an SSL policy.

For secure HTTP, IMAP, and POP3 traffic, create a separate SSL client proxy to enable DV filtering on the decrypted SSL service. For example, if the traffic on TCP port 2000 is expected to be HTTPS, add an SSL client proxy with a Detection Port of 2000 and a Decrypted Service of HTTPS to decrypt HTTPS.

For other SSL services, such as SMTPS, create an SSL client proxy with a Decrypted Service of Other. The TPS device applies DV filters to the outgoing traffic, but does not apply them to the decrypted SSL service. In the case of SMTPS, for example, any non-SMTPS filters in the DV would be applied to the outgoing decrypted traffic, but any SMTPS-specific filters would not be applied to that decrypted traffic.

Procedure

1. In the SMS tools, click **Profiles**.
2. Click **Profiles** > **Shared Settings** > **SSL** > **Client** > **Proxies**.
3. In the **SSL Client Proxies** workspace, click **New**.
4. In the **Proxy Settings** tab of the Create SSL Client Proxy dialog, specify the following settings:
   - **Name** – Enter an SSL client proxy name that corresponds to the client traffic that you want to decrypt. The name is limited to 128 characters and may include uppercase and lowercase letters, digits, underscores, dashes, and spaces.
   - **Description** – (Optional) Enter a description for the Client Proxy. Limit of 512 characters.
   - **Signing certificate** – From the list, select a signing certificate with a private key to decrypt the client-side SSL traffic, or click **Import** to find another signing certificate. One signing certificate must be specified.
   - **Decrypted Service** – Select the SSL service that is accepted on the SSL Detection Port to enable filtering for that particular service. If the SSL service you want is not listed, select **Other**.
   - **Minimum key length** – If you select this option, the TPS considers any certificate received during SSL negotiation invalid if it has a key size that is lower than the minimum.

- **Detection Ports** – Specify which ports on the client proxy handle the SSL traffic. For example, if your internal clients accept HTTPS traffic on port 2000, specify `2000`.

  Note

  If two SSL client policies within the same profile use the same port or port range for their proxies, the device will not be able to determine which policy to enforce. For example, if one client policy specifies a port range of 400–500 for its proxy, you cannot specify port 443 for the proxy of a second SSL client policy in the same profile.

- **Enable logging** – Select this option to collect detailed log information about any SSL traffic that has been decrypted to the external user disk (CFast or SSD). Use this option only for troubleshooting purposes. For example, enable this option if, after you set up SSL inspection, the TPS device does not see SSL session activity. By default, this option is disabled.

- **Send TCP reset to client for blocked sessions** – Select this option to send a TCP reset to the client whenever the TPS blocks an SSL session. This option overrides the TCP reset action set, if enabled, on a DV filter.

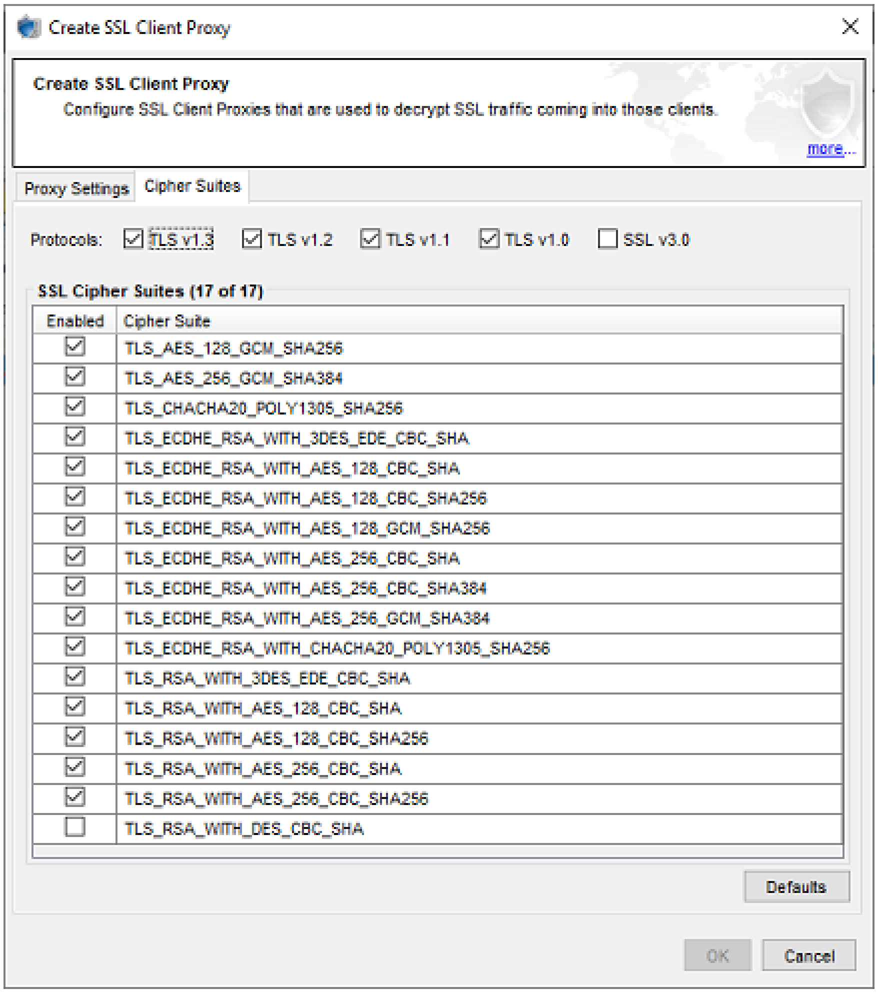
  Enable this option so that protected clients can release network resources on the client quickly if flows are blocked. When this option is disabled, the TCP reset action, if enabled on a DV filter, still applies.

- **Block connections with invalid certificates** – Select this option to block connections with an invalid server certificate. A blocked message is returned in the SSL log. If this setting is disabled, the connection succeeds and inspection begins. A warning message can result after the client attempts to validate the spoofed certificate.

- **Block connections with expired certificates** – Select this option to block connections with an expired server certificate. A blocked message is returned in the SSL log. If this setting is disabled, the connection succeeds and inspection begins. A warning message can result after the client attempts to validate the spoofed certificate.

- **Downgrade HTTP2 protocols to HTTP1** – Select this option to downgrade HTTP/2 to HTTP/1. Currently, only the HTTP1 protocol is supported for full inspection.

5. In the **Cipher Suites** tab, select which protocols and algorithms will be supported by your clients.



Create SSL Client Proxy

Configure SSL Client Proxies that are used to decrypt SSL traffic coming into those clients.

Protocols: ☑ TLS v1.3 ☑ TLS v1.2 ☑ TLS v1.1 ☑ TLS v1.0 □ SSL v3.0

SSL Cipher Suites (17 of 17)

Enabled	Cipher Suite
☑	TLS_AES_128_GCM_SHA256
☑	TLS_AES_256_GCM_SHA384
☑	TLS_CHACHA20_POLY1305_SHA256
☑	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
☑	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
☑	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
☑	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
☑	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
☑	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
☑	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
☑	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
☑	TLS_RSA_WITH_3DES_EDE_CBC_SHA
☑	TLS_RSA_WITH_AES_128_CBC_SHA
☑	TLS_RSA_WITH_AES_128_CBC_SHA256
☑	TLS_RSA_WITH_AES_256_CBC_SHA
☑	TLS_RSA_WITH_AES_256_CBC_SHA256
□	TLS_RSA_WITH_DES_CBC_SHA

You must select at least one checkbox for each cipher suite criteria. The SSL Cipher Suites table automatically updates based on you criteria selections. Deselect any cipher suites that you do not want.

You are now ready to add the SSL client proxy to an SSL inspection policy.

Configure a decryption policy

You can control which SSL traffic is decrypted—including domains, addresses, and server categories—and any exceptions. The Default Client Decryption Policy includes domains that are recommended for use with SSL client inspection to ensure connectivity to critical systems. You can modify the list according to the requirements of your specific environment. You can download updates to this list of recommended domains from the TMC (**Releases > ThreatDV**).

Before you begin

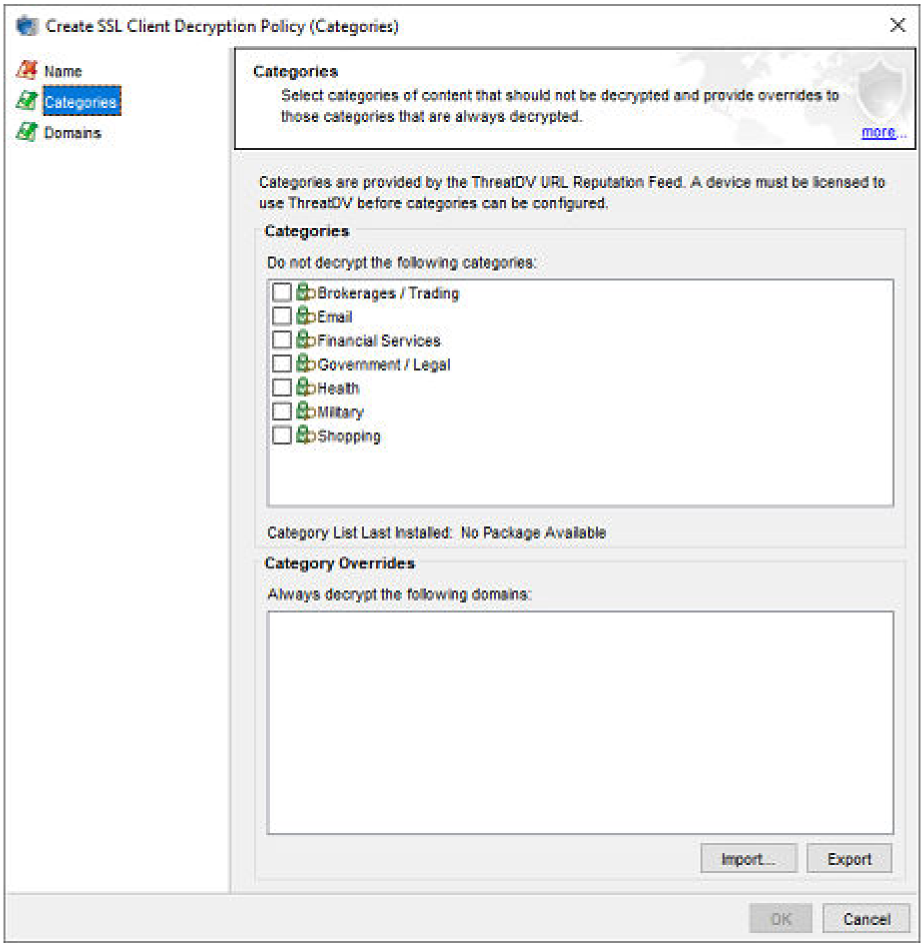
Through category selections, you can exclude certain traffic, such as financial or healthcare traffic, to meet regulatory and privacy requirements. If you make your selections by category, you must have a ThreatDV subscription. You cannot use a pre-TOS v5.4 ThreatDV package on a device that has TOS v5.4 or later installed.

Procedure

1. In the SMS tools, select **Profiles** > **Shared Settings** > **SSL** > **Client** > **Decryption Policies**.
2. In the SSL Client Decryption Policies workspace, click **New**,
3. In the Create SSL Client Decryption Policy dialog:
   - Select the **Name** node and specify a name that corresponds to the client traffic applicable to this policy. You can optionally provide further elaboration in the Description field.
   - Select the **Categories** node and specify which of the sensitive categories should not be decrypted. For a list of categories to be displayed, the SMS must subscribe to a ThreatDV URL Reputation feed and have at least one managed device licensed to use those categories.

Note

If a profile has at least one category enabled and that profile is distributed to a device that is not licensed for ThreatDV, the distribution status notifies you that the target device is not licensed for categories.


Create SSL Client Decryption Policy (Categories)

Categories

Select categories of content that should not be decrypted and provide overrides to those categories that are always decrypted.

Categories are provided by the ThreatDV URL Reputation Feed. A device must be licensed to use ThreatDV before categories can be configured.

Categories

Do not decrypt the following categories:

☐ Brokerages / Trading
☐ Email
☐ Financial Services
☐ Government / Legal
☐ Health
☐ Military
☐ Shopping

Category List Last Installed: No Package Available

Category Overrides

Always decrypt the following domains:

All categories are decrypted by default. After you make your category selections, you can list specific domains within those categories that *should* be decrypted in the **Category Overrides** field. If a Domain entry list is long, you can import them from a file. You also have the option to export a lengthy exceptions list to an external file for easier management of the list.

- Select the **Domains** node to identify which specific domains are not to be decrypted. If a Domain entry list is long, you can import them from a file. To include different subdomains of a primary domain, you can use the `*.` wildcard. For example, specifying `*.company.com` will exclude the following subdomains from decryption:

  - `finance.company.com`
  - `mail.company.com`
  - `legal.compliance.company.com`

  You also have the option to export a lengthy exceptions list to an external file for easier management of the list.

Add an SSL client inspection policy

Configure an SSL client inspection policy to decrypt SSL traffic.

Before you begin

In order to create a client policy, you must first configure a client proxy. *Learn more*.

Procedure

1. In the SMS tools, click **Profiles**.
2. Click **Profiles** > **Inspection Profiles** > ***`profile_name`*** > **SSL Inspection Policies**.
3. In the **Client Policies** workspace, click **New**.
4. In the Create SSL Client Inspection Policy dialog, select **Locked** to prevent an SMS user from changing the SSL inspection policy directly, or as a child instance in another policy. When you select this option, only users with the **Lock SSL Filter** capability can change the SSL inspection policy.
5. Specify a unique policy name that corresponds to your SSL clients.

   Important

   The name you give your client inspection policy cannot match the name you give your server inspection policy. The name must be unique across all SSL client inspection policies in the profile, as well as across any child profiles.

6. Under **Settings**, specify the following options:
   - **State:** Select this option to enable and disable the SSL client policy. When you disable the inspection policy, the SSL client proxy continues to decrypt existing SSL sessions.
   - **Client proxy:** Select the SSL client proxy to decrypt the SSL traffic.
   - **Decryption Policy:** Select an SSL client decryption policy from those you have previously configured, or select **<None>** to not use a decryption policy.
   - **Trust Store:** Select which trust store to use from those you have previously configured. Select only one SSL trust store for the policy.
7. Under **Never Decrypt**, click **Add** to specify source IP addresses that should never be decrypted.
8. Under **Always Decrypt**, click **Add** to specify source IP addresses that should always be decrypted.

You are now ready to distribute your updated inspection profile with the SSL client inspection policy, or, optionally, *configure a decryption policy*.

Configure SSL server inspection

Configure SSL server inspection to specify which SSL sessions you want the TPS device to inspect.

The TPS cannot effectively inspect the encrypted payload of SSL traffic that does not match the SSL inspection profile.

Generate a certificate signing request for server SSL

A certificate signing request (CSR) is a certificate prototype that you generate using the SMS. It can be used to get an endpoint certificate for server SSL inspection. After you configure all the certificate options, you export it to an established certificate authority (CA) who signs it. After it is signed, the CA sends you back the certificate.

Your request can be for wildcard server certificates. These are certificates that have an asterisk-period (`*.`) notation preceding the domain name. For example, `*.domainname.com` ensures that the certificate applies to not just the domain but all its subdomains too, such as `mail.domain.com`. The wildcard must include both the asterisk and the period together, in sequence, and cannot be placed anywhere other than the leftmost position of the domain. Enter the wildcard domain in the Subject Alternative Name **DNS (DNS Name)** field of the procedure that follows.

If your signing request includes a certificate with multiple domains, specify the additional domains in the Subject Alternative Name **DNS (DNS Name)** field of the procedure that follows.

Procedure

1. In the SMS tools, select **Admin** > **Certificate Management** > **Signing Requests**.
2. In the Signing Requests workspace, click **New**.
3. In the New Signing Request dialog, enter values into the following required fields:

   - **Request Name** – Name of the certificate you are creating.
   - **Common Name (CN)** – The common name of the certificate, which is derived from the common name of the server proxy to be protected by the certificate. For example, `company.com`, `www.company.com`, or `*.company.com`.

   You can also configure values for the following options:

   - **Key Size** – Size, in bits, of the key. A minimum size of 2048 bits is recommended.
   - **Make it a Signing Certificate** – Not applicable. Select this option only for client SSL inspection.
   - **Private key can be exported from SMS** – If you intend to use the private key for other purposes, select this option to enable the key to be exported from the SMS after the certificate is signed.
   - **Email** – Email address for the organization.
   - **Organizational Unit (OU)** – Division within the organization. For example, `Research and Development`.
   - **Organization** – Legal, registered name of the organization. For example, `National Football League`.

- **Locality** – City in which the organization is located or registered. Do not abbreviate. For example, `Saint Louis`.
- **State** – State or province in which the organization is operating. Indicate the state or province in which the protected server proxy is operating. Do not abbreviate. For example, `Missouri`.
- **Country** – Two-character ISO format country code of the organization. For example, `US`.
- **DNS (DNS Name)** – Additional domain name of the organization to be covered by the certificate. For example, `login.company.com`.
- **User (RFC822 Name)** – E-mail address of specific user in the protected server proxy's organization. For example, `susan@company.com`.

What to do next

After you create the request, *use the SMS to export* the signing request *without its private key* to a certificate authority who signs and issues the certificate. You will then import the signed certificate that the CA issues you into the SMS as a non-CA certificate so that the private key from the signing request will also be imported.

Import an SSL server certificate and private key

SSL inspection requires both the SSL certificate and private key from your server proxy of interest. Through Server Name Indication (SNI), the SMS accepts multiple certificates and keys from a single SSL server proxy. This enables the server proxy to host multiple secure websites using the same IP address.

The SMS performs basic validation on the status of the certificates itself. If a certificate is not self-signed, remember to import any intermediate CA and root CA certificates in the end-point certificate chain to **Admin** > **Certificate Management** > **CA Certificates**.

Important

Keep your certificates up-to-date. If you see a `No Trust Anchor` status, click **Refresh Status** to confirm the certificate is valid. Whenever you update a certificate on your SSL server proxy, be sure to also replace the certificate on the SMS. *Learn more*.

Procedure

1. In the SMS tools, click **Admin**.
2. Click **Certificate Management** > **Certificates**.
3. In the **Certificates** workspace, click **Import**.
4. In the Import Certificate dialog, enter a certificate name that follows a naming convention so that you can easily assign the correct certificate to the corresponding SSL server proxy.
5. Click **Browse** to locate the certificate file.
6. Select the certificate format, either **Base64 Encoded Certificate (PEM)** or **Encrypted Private Key and Certificate (PKCS12)**.

   Depending on the certificate format, consider the following points:

   - **PEM/DER** format – Always import the private key in a separate file. Be sure to select the **Include a Private Key** checkbox, and then browse to the private key file. If the private key is encrypted, always enter the appropriate password in the Password box.

Note

If this option is unchecked for a certificate that was created from a signing request generated from this SMS, the private key will be retrieved automatically.

- **PKCS12** format – Enter the appropriate password to import the certificate/key pair. When using a certificate chain, always include any intermediate CA certificates in your import file. Only one file can be imported with the certificate/private key pair and any intermediate CA certificates.

Note

If the certificates are not self-signed, be sure to also import any intermediate CA and root CA certificates in the certificate chain to **Admin** > **Certificate Management** > **CA Certificates**.

7. If you intend to use the private key for other purposes, select **Private key can be exported from SMS** so that the key can be exported from the SMS.

You are now ready to add the SSL certificate to an SSL server proxy.

Add an SSL server proxy

Add an SSL server proxy to specify which SSL server configuration to proxy, including the SSL service that is accepted on the SSL detection port.

If you upgrade to TOS v5.4 with an SSL server proxy already configured, the TLSv1.3 protocol and six new cipher suites, including TLSv1.3-specific ciphers, will be automatically added.

For secure HTTP, IMAP, and POP3 traffic, create a separate SSL server proxy to enable DV filtering on the decrypted SSL service. For example, if the web server accepts POP3S traffic on port 2000, add an SSL server proxy with a Detection Port of 2000 and a Decrypted Service of POP3 to enable DV filters for POP3.

For other SSL services, such as SMTPS, create an SSL server proxy with a Decrypted Service of Other. The TPS device applies DV filters to the incoming traffic, but does not apply them to the decrypted SSL service. In the case of SMTPS, for example, any non-SMTPS filters in the DV would be applied to the incoming decrypted traffic, but any SMTPS-specific filters would not be applied to that decrypted traffic.

To inspect more than one decrypted service on a particular SSL server proxy, define the same server IP for each service you want. For example, you can define a server proxy with IP 1.1.1.1 and port 443 (HTTPS), and another server proxy with IP 1.1.1.1 and port 995 (POP3S), and associate them with the same SSL inspection profile.

Procedure

1. In the SMS tools, click **Profiles**.
2. Click **Profiles** > **Shared Settings** > **SSL** > **Server Proxies**.
3. In the **SSL Server Proxies** workspace, click **New**.
4. In the Create SSL Server Proxy dialog, specify the following settings under the **SSL Server Proxy Config** tab:
    - **Name:** Enter an SSL server proxy name that corresponds to your Web server name so that you can easily associate it with your web server.
    - **Destinations:** Specify the SSL server IPv4 address or CIDR range.

- **Detection Ports:** Specify the port range of the encrypted application traffic. For example, if your web server accepts POP3S traffic on port 2000, specify `2000`.
- **Decrypted Service:** Choose the SSL service that is accepted on the SSL Detection Port to enable filtering for that particular service. If the SSL service you want is not listed, choose **Other**.
- **Enable logging:** Select this option to enable the TPS device to write log information about SSL inspection to the external user disk (CFast or SSD). This option collects detailed logging information and should only be enabled for troubleshooting purposes. For example, enable this option if, after you set up SSL inspection, the TPS device does not see SSL session activity. By default, this option is disabled.
- **Send TCP reset to server for blocked sessions:** Select this option to always send a TCP reset to the server proxy whenever the TPS blocks an SSL session. This option overrides the TCP reset action set, if enabled, on a DV filter.

  Enable this option so that protected server proxies can release network resources quickly if flows are blocked. When this option is disabled, the TCP reset action, if enabled on a DV filter, still applies.

5. In the **Certificates** tab, specify the SSL certificates that, along with their private keys, decrypt incoming traffic for your web server proxy. You can import a certificate now, or if you have already imported a certificate into the SMS certificate repository, select which ones you want.

   For TOS v5.3 or later, you can select several system-wide certificates (up to 1000 per device) for a single server profile. The SMS discards any duplicate certificates.

   For any earlier TOS, you can select only one certificate; otherwise, the profile distribution will fail. Click the **Repair** button to re-import the private key of a certificate that has been identified as broken.

6. In the **Cipher Suites** tab, select the protocols that are supported by your web server proxy.

   The Cipher Suite list automatically updates based on your protocol selections. Deselect any cipher suites that you do not want.

7. Click **OK**. You are now ready to add the SSL server proxy to an SSL inspection policy.

Add an SSL server policy

Add an SSL server policy to your inspection profile to specify the server proxy traffic that you want to decrypt, and any client exceptions. Always configure an SSL inspection policy for the inspection profile that is assigned to the virtual segment that requires the SSL inspection.

Before you begin

In order to create a server policy, you must first configure a server proxy. *Learn more*.

Procedure

1. In the SMS tools, click **Profiles** > **Inspection Profiles** > ***`profile_name`*** > **SSL Inspection Policies**.
2. In the **Server Policies** workspace, click **New**.
3. In the Create SSL Server Inspection Policy dialog, select **Locked** to prevent an SMS user from changing the SSL server policy directly, or as a child instance in another policy. When you select this option, only users with the **Lock SSL Filter** capability can change the SSL server policy.
4. Specify a unique policy name that corresponds to the SSL server proxy configuration.

Important

The name you give your server proxy inspection policy cannot match the name you give your client inspection policy.

5. Under Settings, select the SSL server that you want this server policy to proxy.
6. Under Source Exceptions, click **Add** to specify any SSL client IPv4 addresses to exclude from SSL inspection.

You are now ready to distribute your updated inspection profile with the SSL server policy.

Inspection profile distribution

Always distribute the inspection profile with your SSL inspection policy to the virtual segments that require the SSL inspection and to the virtual segments that will receive the SSL client requests.

Important

You do not need to distribute an SSL inspection policy to the virtual segment that receives the SSL server response. The SSL inspection policy enables the device to proxy (and decrypt) the SSL session between both the SSL client and the device, and between the SSL server and the device.

When the inspection profile includes both an SSL server policy and an SSL client policy, and an SSL connection matches both policies, the SSL **server** policy decrypts the SSL session.

Grant permissions to SSL inspection

Grant permissions so that an assigned user group can configure SSL inspection for particular SSL server proxies and SSL client proxies.

Add SSL inspection to the user role

Grant permission to a user role to configure SSL inspection, including:

- SSL inspection profiles
- SSL server proxies
- SSL client proxies
- SSL global settings
- SSL log
- SSL event information

By default, permissions for SSL inspection are granted to the Administrator role.

Procedure

1. In SMS tools, click **Admin**.

2. Click **Authentication and Authorization** > **Roles**.
3. In the **User Roles** workspace, click **New** to create a user role based on an existing role or click **Edit** to change an existing role. You cannot change the default user roles.
4. In **Capabilities** options, configure the **SSL Inspection Management** inspection capability under **Profiles** > **Shared settings management**.

Grant the user group access to SSL configuration

Grant a user group permission to configure your SSL server proxies and client proxies. By default, a user group has access to all SSL server and client proxies, including new ones that have yet to be defined.

Procedure

1. In the SMS tools, click **Admin**.
2. Click **Authentication and Authorization** > **Groups**.
3. In the **User Groups** workspace, click **New** to create a group or **Edit** to change an existing group.
4. From the **SSL Servers** node, deselect an SSL server to deny access.
5. From the **SSL Client Proxies** node, deselect an SSL client proxy to deny access.
6. From the **SSL Client Decryption Rules** node, deselect a decryption rule to deny access.
7. From the **Profiles** node, deselect an SSL inspection profile to deny access.
8. From the **Devices** node, deselect a device to deny access.

Troubleshoot SSL inspection

If SSL clients cannot reach the server, check Traffic Management and Reputation filters to verify the sessions of interest are not blocked. Traffic Management and Reputation filters are applied before SSL inspection.

View event information

Check the IPS Block and Alert logs, and the Quarantine log for event information about SSL traffic. These logs include an **SSL Inspect** column to report on SSL sessions.

Only URL Reputation gets SSL inspected sessions reported in the Reputation Block and Alert logs. IP and DNS Reputation do not report on SSL sessions because they are analyzed prior to SSL Inspection.

Note

If you see unexpected alerts on SSL traffic, the inspection profile might be inspecting the decrypted server response. *Learn more*.

1. In the SMS tools, click **Devices**.
2. Click **All Devices** > ***`device_name`***.

3. Click **Events** > **Traffic** to view traffic graph for SSL traffic.

TRAFFIC GRAPH	INFORMATION
SSL Decrypted Traffic	Displays the overall SSL traffic seen and amount inspected.

View log information

The SSL Inspection log displays SSL session information, including information about SSL sessions that failed to negotiate SSL parameters. By default, when you add an SSL server proxy or client proxy, logging is disabled.

If you do not see SSL sessions for a particular server proxy or client proxy, edit the SSL proxy to enable logging and then review this log for useful troubleshooting information. When you finish troubleshooting, disable logging on the proxy.

The SSL inspection log does *not* contain:

- SSL system errors – Check the System log for SSL-related system errors.
- SSL sessions that are Blocked or Quarantined because of a filter match – Check the IPS Block and Alert logs, and the Quarantine log. These logs include an **SSL Inspect** column to report on SSL sessions.

1. In the SMS tools, click **Events**.
2. Click **SSL Inspection Logs**.
3. In the **SSL Log Entries** workspace, view and query the SSL Inspection log.

Note

When you delete an SSL inspection profile or policy, the device continues to inspect the corresponding SSL connections until the SSL connection closes, but the SSL inspection log incorrectly indicates that the SSL connections have an unknown profile or policy. You can disregard these entries. The device stops logging these connections after the SSL connections close.

Basic troubleshooting

If SSL clients are reaching your server but the TPS device is not inspecting some or all of the encrypted sessions of interest, check the following items:

- System log – Determine whether the TPS device is bypassing SSL sessions.
- SSL server configuration – Verify the SSL server IP and ports.
- SSL client configuration – Verify the SSL client port.
- SSL inspection profile – Confirm any SSL inspection policies do not include a source IP exception that would bypass SSL inspection.
- Virtual segment – consider the following:
  - If the virtual segment designates a segment, confirm that it is the correct segment. For example, confirm that interface 1A receives the SSL client request to your SSL server.
  - If the virtual segment defines VLANs, confirm that the VLANs are correct for your SSL server.
  - If the virtual segment defines source IP addresses, confirm that the SSL client requests correspond to those IP addresses.

- If the virtual segment defines destination IP addresses, confirm that your SSL server corresponds to one of those addresses.

To verify	Do this
The TPS is not bypassing SSL sessions	On the device, check the System log for an entry similar to the following: `SSL Inspection reached Critical threshold of Max Concurrent Connections. Action: Allow but bypass Inspection` If the number of concurrent SSL sessions exceeds the maximum threshold as specified by the entry in the System log, the TPS device does not inspect them. If necessary, reconfigure SSL inspection to reduce the number of concurrent SSL connections. For information about configuring SSL inspection to block SSL sessions that exceed the maximum threshold, contact customer support.
SSL inspection license is installed and valid	Verify the license package.
SSL inspection is enabled	*Enable SSL inspection*.
The correct certificate and key are installed	For SSL server inspection, import the SSL server certificate and private key. For SSL client inspection, import the SSL signing certificate and private key.
The SSL server matches the correct IP address and port	Edit the SSL server.
The profile is applied to the correct virtual segments	Distribute the inspection profile.
The virtual segment includes the desired SSL server IP addresses and ports	Verify the SSL clients are reaching the SSL server.

Advanced troubleshooting

If the basic troubleshooting does not resolve your issue, troubleshoot the TPS device.

Procedure

1. Run the **`debug np stats show npSslInspStats`** command to check the connection counters. If they are all zero, then it is likely that you have a configuration issue. If there are refused connections, it is also a configuration issue, but there are likely incompatible ciphers or the connection is trying to use compression when the profile does not support it. *Learn more*.

2. Run the **`debug np stats show npSslInspProtocolStats`** command and consider these points:

   - Non-zero entries in the **`clientCipherOther`** counter – Indicate a possibly unsupported cipher. Use the other error counters to narrow the source of the problem to at least the server or the client.

   - Server connection failures – Indicate a possibly unsupported cipher, but with the added chance that the server might be asking for a client certificate, which the proxy does not support.

3. Run the **`debug np stats show npTcpProxyStats`** command to confirm whether the profile and server are configured to correctly match traffic. If the results are all zero, then there is no matching traffic to inspect. If there is any TCP traffic that matches a profile, the results are non-zero.

Replace an SSL server certificate

Replace an SSL certificate before it expires. If a certificate expires, the System log generates an error.

When you replace a certificate, consider these points:

- The SMS replaces the certificate on any applicable devices. Replacing the certificate preserves any existing references to the certificate in the device configuration.
- Changes to the device configuration, including certificate replacement, require you to have the **Device X509 Certification Configuration** capability on the device.
- Certificate replacement requires you to have the **Admin X509 Certificate Management** capability.

Important

When configuring SSL server inspection, make sure that you update both the device certificate and the SSL webserver certificate before they expire. When configuring SSL client inspection, make sure that you update the signing certificate before it expires. If the certificates expire or become out of sync, the SSL connections might fail.

Procedure

1. In the SMS tools, click **Admin**.
2. Click **Certificate Management > Certificates**.
3. In the **Certificates** workspace, select a certificate from the list and click **Replace**.
    - For certificates with a private key, browse to and open a certificate.
    - For PEM/DER certificates, browse to and open the associated private key.
4. (Optional) Provide a password to decrypt the private key.
5. Click **OK**. The SMS appends `_REPLACED` to the name of the original certificate.

    The SMS certificate repository automatically updates any managed devices with the new certificate. The SMS displays an error message if the certificate replacement fails on a particular device.