USING DECRYPTION TO BLOCK THREATS IN ENCRYPTED TRAFFIC

YOU CAN'T ANALYZE WHAT YOU CAN'T SEE

- SSL/TLS review
- Certificate management
- SSL/TLS decryption
- SSH decryption
- Other decryption methods and features

EDU-210 Version A
PAN-OS® 10.2

paloalto NETWORKS

Learning Objectives

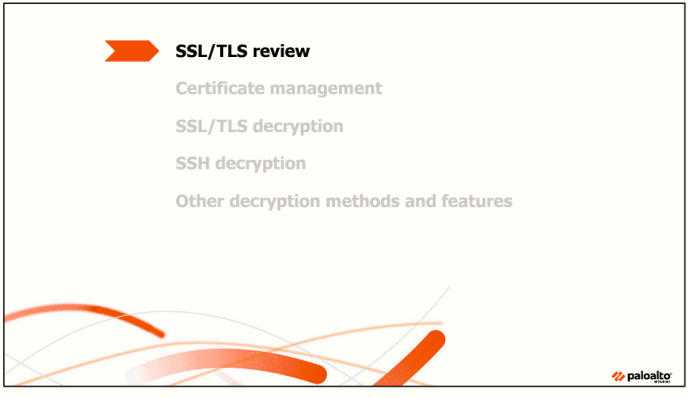
After you complete this module,
you should be able to:

- Review fundamental SSL concepts and operation
- Create and manage certificates using the web interface
- Configure SSL/TLS forward proxy decryption
- Configure SSL/TLS inbound inspection decryption
- Prevent decryption for specific traffic
- View information and troubleshoot SSL/TLS issues using the CLI and logs
- Identify decryption configuration considerations
- Configure SSH decryption
- List other available decryption methods

This module covers the topics listed here. Read through the list before proceeding.

After you complete this module, you should be able to:

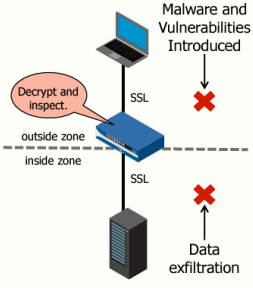
- Review fundamental SSL concepts and operation
- Create and manage certificates using the web interface
- Configure SSL/TLS forward proxy decryption
- Configure SSL/TLS inbound inspection decryption
- Prevent decryption for specific traffic
- View information and troubleshoot SSL/TLS issues using the CLI and logs
- Identify decryption configuration considerations
- Configure SSH decryption
- List other available decryption methods



This section provides a review of SSL/TLS benefits and operation.

Importance of SSL/TLS

- SSL/TLS secures network communication across a shared network:
  - Encrypts for data privacy
  - Uses hashes for data integrity
  - Uses certificates for authentication
- SSL/TLS decryption helps to prevent:
  - Malware introduction
  - Data exfiltration



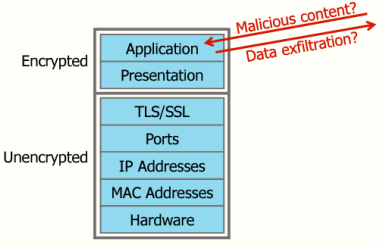
For a Tech Doc about this topic, log into Live and search for "Decryption Overview"

The Secure Sockets Layer (SSL), Transport Layer Security (TLS), and Secure Shell (SSH) encryption protocols secure traffic between two entities, such as a web server and a client. SSL/TLS, and SSH encapsulate traffic, encrypting data so that it is meaningless to entities other than the client and server with the certificates to affirm trust between the devices and the keys to decode the data. SSL/TLS secure communications between network nodes by encrypting cleartext data for privacy before it is sent across the network. SSL also uses hashes to maintain data integrity and digital certificates to authenticate the communication end nodes.

SSL decryption on the firewall helps to prevent the introduction of malware. The traffic is decrypted so that it can be identified and inspected for malware by App-ID and Content-ID. SSL decryption on the firewall also helps to prevent the exfiltration of sensitive and valuable information. The traffic is decrypted to enable the firewall to inspect data for sensitive and valuable information. SSL decryption by the firewall is a primary feature used to block the cyberattack lifecycle.

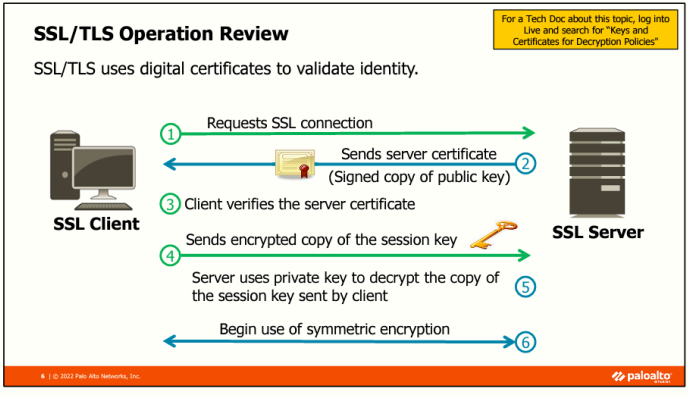
Why Decrypt Network Traffic?

- Most web traffic is encrypted.
- Palo Alto Networks firewalls can decrypt:
  - SSL/TLS inbound and outbound traffic
  - SSHv2



Each year more web traffic is encrypted. In fact, as of 2020, most web traffic is encrypted. Palo Alto Networks firewalls provide the capability to decrypt and inspect network traffic for visibility, control, and granular security. Palo Alto Networks firewalls can decrypt and inspect traffic to provide visibility into threats and control protocols, certificate verification, and failure handling. Decryption can enforce policies on encrypted traffic so that the firewall handles encrypted traffic according to your configured security settings. You create policy rules to decrypt traffic preventing malicious encrypted content from entering your network and sensitive content from leaving your network concealed as encrypted traffic. Enabling decryption can include preparing the keys and certificates required for decryption, creating decryption profiles and policies, and configuring decryption port mirroring.

A Palo Alto Networks firewall can decrypt SSHv2 and SSL/TLS inbound and outbound network traffic.



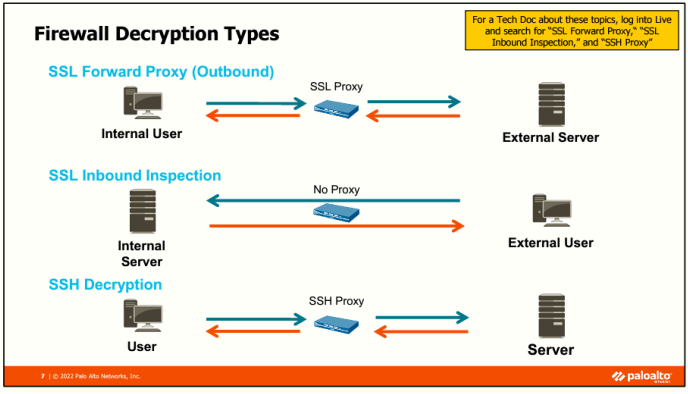
SSL/TLS uses a digital certificate to validate the identity of a communication partner.

The initiation of an SSL session can be summarized as follows:

1. A client requests an SSL connection.
2. The server responds with its certificate, which contains its identity and public key.
3. The client uses the public key infrastructure, PKI, to validate the server certificate and server public key.
4. If the certificate is valid, the client uses the server's public key to encrypt a symmetric session key and send it to the server.
5. The server uses its private key to decrypt the copy of the session key that the client sent to the server.
6. Both sides use the session key to encrypt communications for privacy.

Supplemental Notes

The communication partners periodically might need to establish a new session and rekey the communication. The process of rekeying new or existing sessions is known as Perfect Forward Secrecy, or PFS. It provides assurances that if a private key is compromised, any recorded former sessions cannot be decrypted. PFS support for SSL Forward Proxy was added in PAN-OS® 7.1. PFS support for SSL Inbound Inspection was added in PAN-OS 8.0.



The firewall provides three types of Decryption policy rules: SSL Forward Proxy to control outbound SSL traffic, SSL Inbound Inspection to control inbound SSL traffic, and SSH Proxy to control tunneled SSH traffic. SSL decryption (both forward proxy and inbound inspection) requires certificates to establish the firewall as a trusted third-party and establish trust between a client and a server to secure an SSL/TLS connection. You can also use certificates when excluding servers from SSL decryption for technical reasons, like when the site breaks decryption for reasons such as certificate pinning, unsupported ciphers, or mutual authentication. SSH decryption does not require certificates.

As an example of the usefulness of SSL Forward Proxy decryption, consider a scenario where an internal user will connect via an encrypted connection to Facebook. The company policy allows employees to read Facebook but prevents facebook-chat and facebook-posting. If SSL decryption is enabled for the Facebook application, the firewall can implement company policy quickly. If SSL decryption is not enabled, the firewall cannot identify which application is inside the SSL connection, nor can it recognize that application shifts occur within the connection.

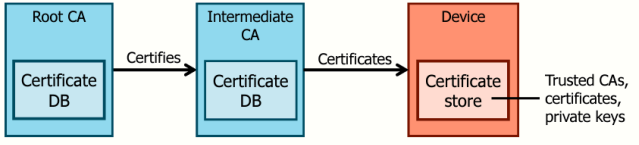
You can also configure SSL Inbound Inspection decryption on the firewall, which decrypts SSL traffic from external users to internal servers. To configure SSL Inbound Inspection, you must access the server's private key and certificate. In this configuration, the firewall does not act as an SSL proxy. An SSL connection is formed directly between the external user and the internal server. The firewall decrypts and inspects only the traffic flowing through it. The firewall can apply Security policy and Security Profiles to the SSL connection and block disallowed traffic.

You can configure SSH Decryption to decrypt outbound and inbound SSH traffic. If an SSH tunnel (port forwarding) is discovered, the SSH connection is blocked to ensure that SSH is not being used to tunnel disallowed applications and content. You also can apply a Decryption Profile to your Security policy rules to control regular, non-port-forwarded SSH traffic.

This section describes certificate configuration and management using the web interface.

Public Key Infrastructure (PKI)

- Solves the problem of secure identification of public keys
- Uses digital certificates to verify public key owners
- Typical PKI components:

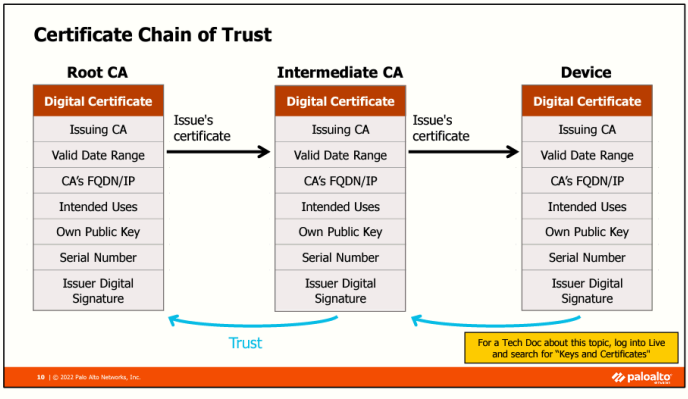


For a Tech Doc about this topic, log into Live and search for "Develop a PKI Rollout Plan"

The public key infrastructure solves the problem of verifying the identity of a public key owner. PKI is the hardware, software, policies, and standards used to create, manage, distribute, and revoke public keys and digital certificates. A PKI digital certificate is a method of packaging and distributing public keys to prove the identity of its owners. Palo Alto Networks firewalls support X.509-format certificates.

A PKI certificate authority (or CA) provides services that authenticate devices, services, and people by issuing certificates that confirm their identity and public key. CAs are arranged in a hierarchical fashion, similar to a file system. The root CAs form the top level of the hierarchy, and intermediate CAs form the second and lower levels. An intermediate CA is certified by a root CA to issue certificates or certify additional lower-level intermediate CAs. Each CA issues and revokes certificates and has a certificate database that stores certificates.

Devices use a certificate store to store their private keys and the certificates they have been issued. They also maintain a list of trusted CAs. This list of trusted CAs can be updated by a user or device software update. If the certificate of the issuing CA is not added to a client's certificate store, the client receives a warning message when browsing to secure sites verified by that CA.

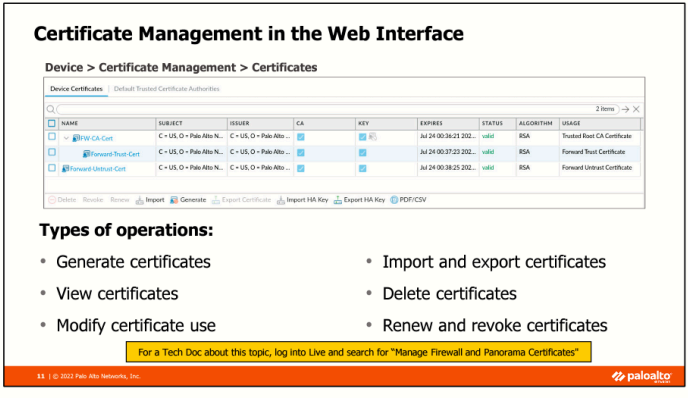


The certificate chain of trust is a hierarchical list of certificates used to authenticate a device, service, or person.

In the example, the chain begins with the device's certificate. Each certificate in the chain is digitally signed by the entity identified by the next-higher certificate in the chain.

The chain terminates with a root CA certificate. The root CA certificate always is self-signed by the root CA itself. A root certificate is a self-signed certificate because the issuing authority is itself. These root CAs form the basis for all PKI deployments.

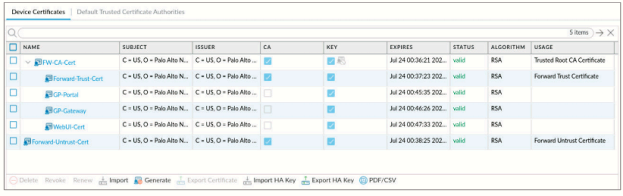
The device can verify the owner of a public key if the device's list of trusted CAs includes a root CA in the chain of trust. For example, a browser can check to determine which authority issued an intermediary certificate, retrieve the intermediary's certificate from that higher authority, and verify the intermediary certificate. This process continues until a root CA is encountered in the chain. In practice, this process is rarely more than two or three hops.



The web interface includes certificate management for the firewall at **Device > Certificate Management > Certificates**. You can use the web interface to generate and view certificates or generate certificate signing requests, known as CSRs. You can import certificates from a third-party or internal CA and export certificates to other devices. You also can modify certificates to meet specific user requirements on the firewall. Certificates issued by the firewall can be renewed or revoked using the web interface.

Certificate Hierarchy

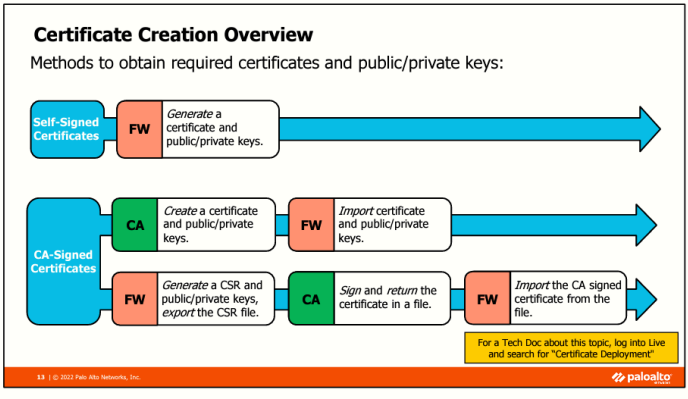
Device > Certificate Management > Certificates



Nested hierarchy visually illustrates the certificate chain of trust.

PAN-OS® software organizes the certificate list hierarchically according to the certificate chain of trust. This format simplifies determining which certificates are related by grouping certificates under their CA certificates on the system.

In the example, FW-CA-Cert is a trusted root CA certificate used to sign and validate subordinate certificates. This CA certificate has signed certificates for the GlobalProtect Portal and Gateway machines, an SSL forward trust certificate to decrypt SSL connections where the server certificate was validated, and a certificate that proves the identity of the firewall's web interface. Notice that the SSL-Fwd-Untrust-Cert certificate is not signed by the FW-CA-Cert certificate and therefore is not indented beneath it. The SSL-Fwd-Untrust-Cert certificate is used to sign and decrypt SSL connections whose certificates could not be validated. This certificate will generate warnings in the user's browser window.



PAN-OS software includes several methods to obtain the digital certificates necessary to decrypt SSL traffic passing through the firewall. You can get certificate authority-signed certificates or generate self-signed certificates. The use of a CA-signed certificate is preferred because it simplifies SSL configuration. If the CA certificate that signs the SSL certificates already is known and trusted by all devices in your organization, then trusted SSL connections are more easily configured between the firewall and those devices.

You can buy a CA-signed certificate and public/private key pair from a public CA. The primary disadvantage is that you must pay for this certificate. The primary limitation is that public CAs typically do not sell signing certificates, which are certificates that can sign other certificates. Signing certificates are required for SSL Forward Proxy decryption.

If one exists, you can also get a CA-signed certificate and public/private key pair from an internal CA. The primary advantage is that these certificates are free of cost. An internal CA can also issue signing certificates required for SSL Forward Proxy decryption.

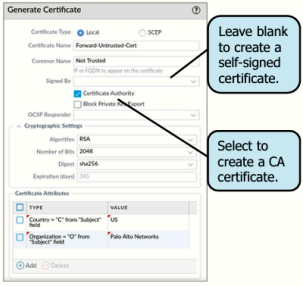
You can create the certificate and public/private key pair on the CA server, regardless of whether you are working with a public or internal CA., then use the web interface to import the signed certificate and key pair into the firewall. A disadvantage of this method is that the private key is transferred over the network, where there is a small risk that it will be stolen.

To avoid a network transfer of a private key, you can use the firewall web interface to generate the public/private key pair along with a certificate signing request, also known as a CSR. You export the CSR file off the firewall to the CA that signs and returns it. Then you use the web interface to import the file that contains the signed certificate into the firewall. The private key never leaves the firewall if you use this method.

You also can use the web interface to create a self-signed certificate and a public/private key pair. The primary advantage is that this certificate is free of cost and can be obtained in minutes. The self-signed certificate can be a signing certificate, too, to be used with SSL Forward Proxy decryption. The primary disadvantage is that a self-signed certificate is, by default, not trusted by the other devices in your organization. You will need to export and install the self-signed certificate to the trusted root certificate stores on the other devices. After you have placed the self-signed certificate on the firewall and all devices, it can be used to establish trust between all the devices and the firewall.

Generate a Self-Signed Certificate

Device > Certificate Management > Certificates > Add



- Configure a self-signed certificate:
  - The **Signed By** field must be blank.
- **Generate** creates a certificate and public/private key pair.

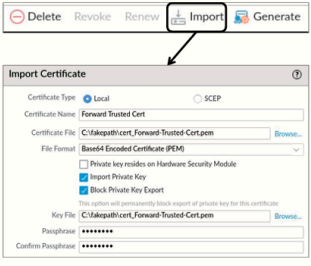
For a Tech Doc about this topic, log into Live and search for "Create a Self-Signed Root CA Certificate"

When you create a self-signed certificate, leave the **Signed By** field blank. Select **Certificate Authority** to create a CA certificate that can sign other certificates. When you click **Generate**, the firewall creates a certificate and a corresponding public/private key pair.

The certificate in the example is a self-signed CA certificate that can sign other certificates. This certificate could be used as the forward untrust certificate in an SSL Forward Proxy decryption configuration.

Import a CA Certificate

- Use an internal CA to create:
  - Firewall CA certificate
  - Public/private key pair
- Use **Device > Certificate Management > Certificates > Import**.
- Complete the form and click **OK**.
- Imports certificate and public/ private keys into the firewall.

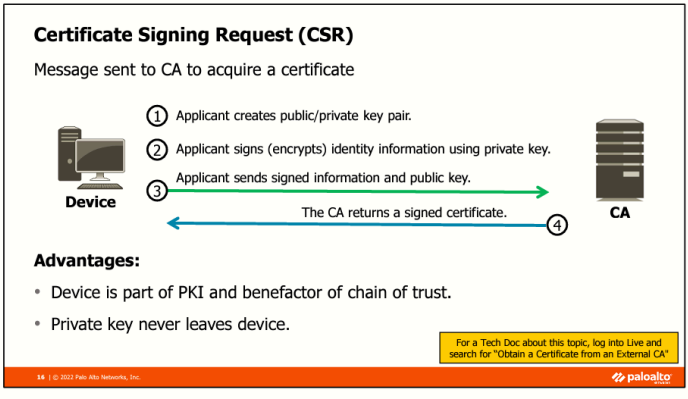


For a Tech Doc about this topic, log into Live and search for "Import a Certificate and Private Key"

A firewall CA certificate and public/private key pair can be created on an internal CA and imported into the firewall. Ensure that you have a secure network because the firewall's private key will be transferred over the network. You can permanently block the export of private keys for certificates that have been generated in or imported into PAN-OS software. Blockage of the export of private keys from your PAN-OS appliances hardens your security posture because it prevents rogue administrators from misusing keys.

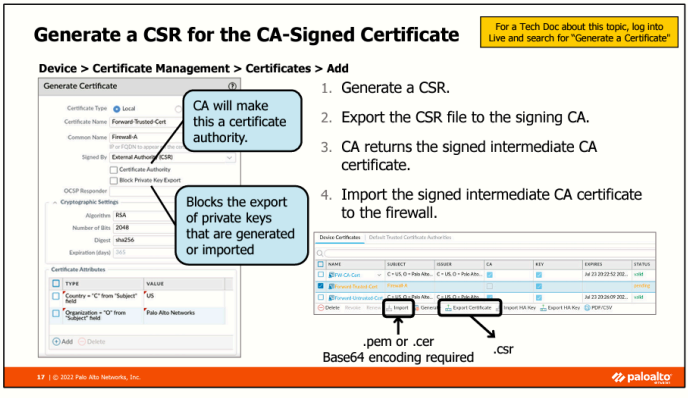
After the internal CA has created the certificate and keys file, use the form shown here to import the files. A PKCS12 file contains both the certificate and private key in the same file. A PEM file containing the certificate will not contain the private key. The private key would have to be transferred in a separate file.

With a CA certificate configured on the firewall, you can use the web interface to create any other required certificates, and the firewall's CA certificate can sign them.



Some devices, including a Palo Alto Networks firewall, can submit a CSR to a CA. To submit a CSR, the device generates a public/private key pair and identity information and then submits the public key and identity information to a CA using a CSR file. The CA utilizes the information in the CSR file to create a certificate signed with the CA's signature. The signed certificate is sent back to the device.

The advantage of using CSRs is that the device becomes part of the existing PKI infrastructure and its certificate chain of trust. Another advantage is that the device's private key never leaves the device. The primary disadvantages are that the device must generate a CSR and that there is some administrative overhead compared to a device using a self-signed certificate.



You can generate a public/private key pair and a certificate for a CA to sign after browsing **Device > Certificate Management > Certificates** and click **Add**. Provide a descriptive name for the certificate. The Common Name typically is the FQDN or IP address of the firewall.

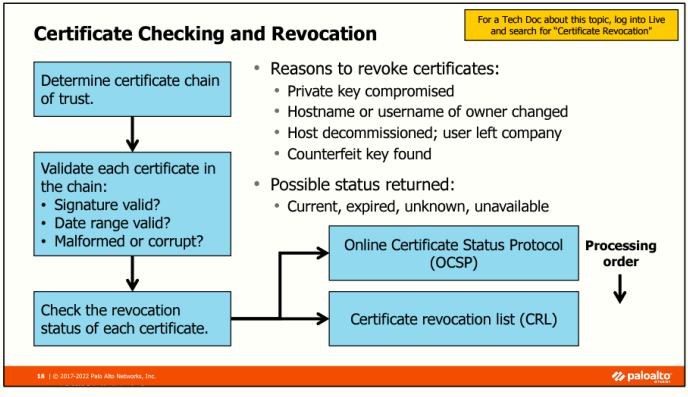
The key to generating a CSR is to select **External Authority (CSR)** in the **Signed By** field. You do not have to select the **Certificate Authority** check box because this certificate will be configured as a subordinate certificate authority by the CA that signs this certificate. Configuring this certificate as a subordinate certificate authority enables the firewall to use this certificate to sign SSL server certificates.

With the release of PAN-OS 10.0, you can permanently block the export of private keys for certificates that have been generated in or imported into PAN-OS. Blocking the export of private keys from your PAN-OS appliances hardens your security posture because it prevents rogue administrators from misusing keys. You cannot block the export of private keys for keys that already exist on the firewall.

You can also change the default cryptographic settings used to generate the certificate and add a variety of Certificate Attributes used to identify the owner further and use for the certificate.

Click the **Generate** button after you have completed the form. A new but unsigned certificate appears in the certificate management window. The firewall also creates and stores a public/private key pair.

Select the unsigned certificate in the web interface and then click **Export** to generate a .csr file. Send this .csr file to your external or internal CA. The CA will sign the certificate and create a .pem file. Use the web interface to **Import** the .pem file into the firewall. After the import, the newly signed certificate is available for use.

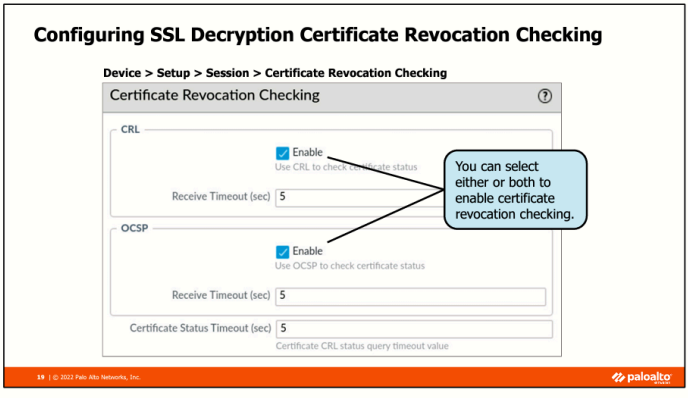


An SSL client must check all certificates in the chain of trust before an SSL connection can be considered secure. Before an SSL client can check a certificate, it must determine the chain of trust. Then the SSL client must validate each certificate in the chain.

If the signatures are valid, then the SSL client must check the revocation status of each certificate in the chain. The firewall acts as an SSL client in an SSL Forward Proxy configuration. The two methods available to check certificate revocation status are OCSP and CRLs. A firewall can use OCSP, CRLs, or both to verify certificate revocation status for SSL decryption. If you configure both methods on a firewall, the firewall first tries OCSP. If the OCSP responder is unavailable, the firewall uses the CRL method.

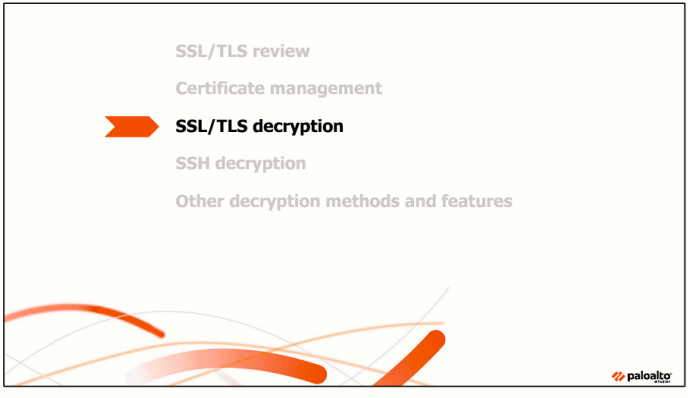
A certificate might need to be invalidated before its expiration date for one of several reasons: the private key of the certificate owner might have been compromised, the hostname or username of the certificate owner might change, a host could be decommissioned, or a user can leave the company, or a counterfeit key might need to be invalidated. CAs store the list of revoked certificates in their certificate database.

When the validity of a certificate is checked, OCSP and CRL can return a current status, which means the certificate is valid. A return status of *expired* means that the SSL client cannot trust the server's identity. A return status of *unknown* means that OCSP or CRL was consulted, but the certificate's validity could not be established. If OCSP or CRL cannot be contacted, the status is *unavailable*.

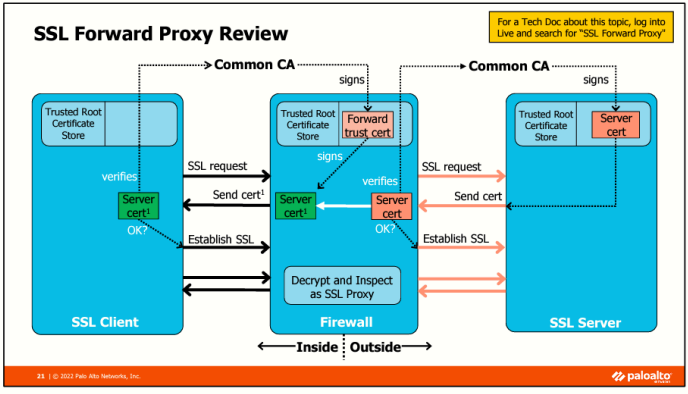


You can configure the firewall to verify the revocation status of certificates used for decryption. Certificate revocation status checking for SSL decryption is not enabled by default because, for some websites, the additional time required to perform the checks might cause SSL connection failure.

To configure the firewall to perform certificate revocation status checking for SSL decryption certificates, browse **Device > Setup > Session** and click **Certificate Revocation Checking**. Select the checkboxes to enable either CRL, OCSP, or both. The online web interface help pages describe how the timeout values interact, depending on whether you have configured OCSP, CRL, or both. If you configure OCSP and CRL, CRL is used only if the OCSP responder is unavailable.



This section describes the operation and configuration of SSL/TLS decryption on the firewall.



The firewall uses SSL Forward Proxy to decrypt and inspect SSL traffic when you *do not* have access to the private key of the SSL server. Before configuring SSL Forward Proxy, you must deploy the certificates that SSL uses to confirm the identity of an endpoint. The firewall and SSL server must share a common CA for the firewall to validate the server's identity. The SSL client and the firewall must have access to a common CA for the client to validate the identity of the firewall.

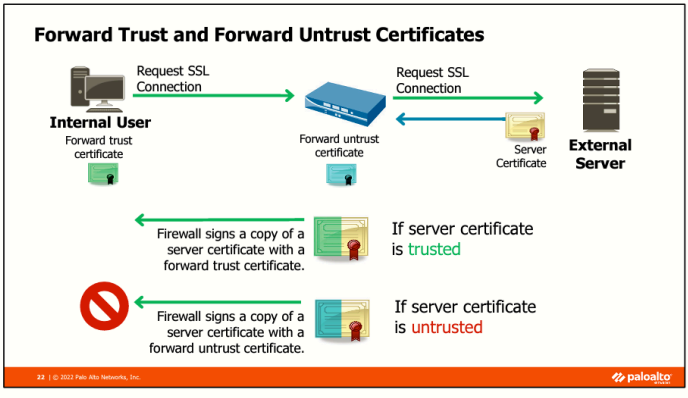
The SSL client establishes a session with the server by initiating an SSL handshake. The firewall intercepts the handshake request and re-issues it to the server. Then the SSL server responds with its certificate, which is signed by a CA common to the server and the firewall. The firewall uses the CA to validate the certificate and the server's identity.

Then the firewall copies the server certificate and signs it with its forward trust certificate and public key. A forward trust certificate indicates that the firewall has verified and trusts the server certificate to the SSL client. The firewall then forwards the client's newly copied and signed server certificate.

The client then uses the forward trust certificate of the firewall to validate the firewall identity, using a CA common to the client and the firewall.

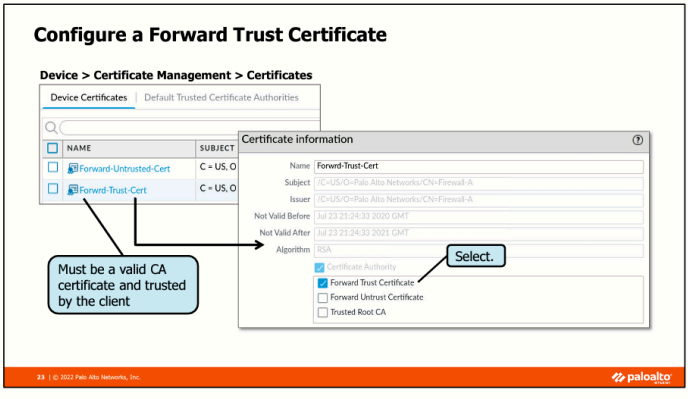
At this point, two SSL tunnels have been established: one between the client and the firewall and another between the firewall and the server. The firewall acts as an SSL proxy between the client and server and decrypts and inspect data flowing between the client and server.

A firewall that cannot verify the server certificate signs the server certificate with a forward untrust certificate that indicates that the SSL server certificate could not be verified.



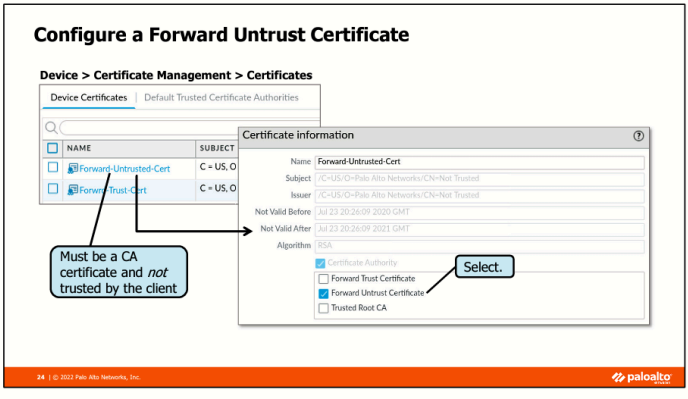
With SSL Forward Proxy decryption, the firewall resides between the internal SSL client and the external web server. As a trusted third party, the firewall uses its forward trust or forward untrust certificates to inform the SSL client whether the firewall has verified the validity of the web server's certificate.

When an SSL client initiates a session with an external server, the firewall intercepts the SSL request and forwards its SSL request to the server. The server's certificate is sent to the firewall. If a CA signs the server's certificate that the firewall trusts, the firewall creates a copy of the server's certificate signed by the firewall's *forward trust* certificate. If a CA signs the server's certificate that the firewall does not trust, it creates a copy of the server's certificate and signs it with its *forward untrust* certificate. The firewall sends the signed certificate to the SSL client in either case. If a forward untrust certificate was used, the SSL client sees a block page warning that the website it is trying to connect to is not trusted by the firewall acting as an SSL Proxy. The user can choose to proceed or terminate the session.



The first step to configure SSL Forward Proxy decryption is to configure a forward trust certificate on the firewall. The forward trust certificate can be signed by a public or an internal CA. Alternatively; you can create a firewall self-signed forward trust certificate. However, every SSL client will need to have this certificate installed in their certificate store. Otherwise, they will get certificate warning errors. The forward trust certificate must be a CA certificate to sign other certificates.

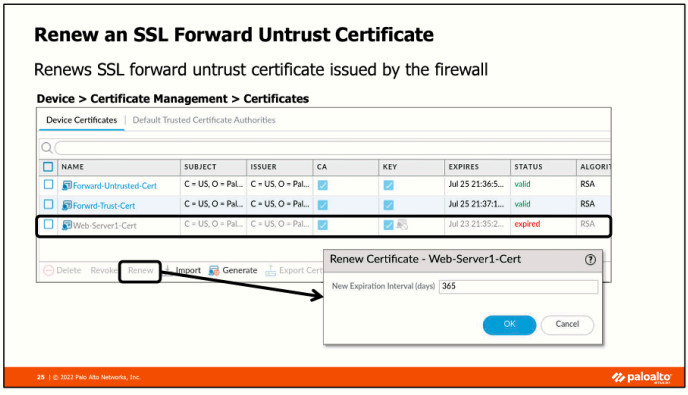
Use the web interface to create a CA certificate as a forward trust certificate. After the certificate has been created, click to open it and select the **Forward Trust Certificate** check box, enabling the firewall to use the certificate as its forward trust certificate during SSL Forward Proxy decryption.



Palo Alto Networks recommends that you configure a separate forward untrust certificate, which ensures that an SSL client will receive a browser block page when the firewall does not trust the certificate of the SSL server to which the client is attempting to connect. An SSL client will *not* receive a browser block page when it connects to an untrusted server if the SLL client trusts the forward untrust certificate of the firewall. To ensure that clients do not trust this certificate, the certificate should not be issued by a trusted CA, nor should it be copied to the certificate store of an SSL client.

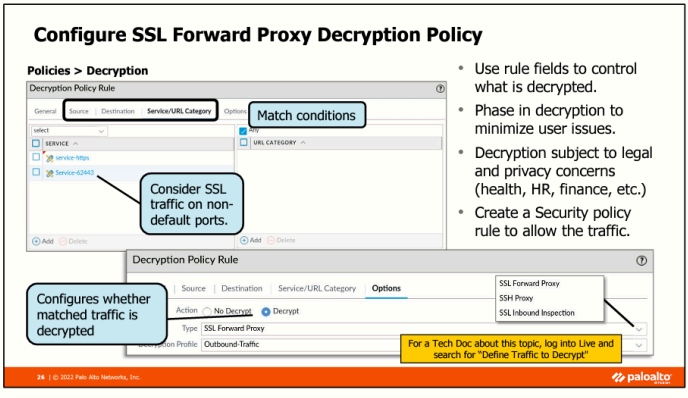
To create a forward untrust certificate, generate a new CA certificate not signed by any SSL client-recognized CA. Also, do not copy this certificate to the certificate store of the SSL client.

After generating the new self-signed certificate, click the certificate to open it in the web interface. After the certificate opens, select **Forward Untrust Certificate** to configure it as a forward untrust certificate.



You can use the web interface to renew any CA or non-CA certificates issued by the firewall. The SSL forward untrust certificate will need to be renewed periodically. Renewal of a certificate changes its expiration date to a later date. The default expiration date for certificates issued by the firewall is one year. This expiration date typically should be increased to two or more years.

Select the certificate to renew in the web interface and then click **Renew**. In the **Renew Certificate** window that opens, enter the new number of days before the certificate expires and click **OK**.

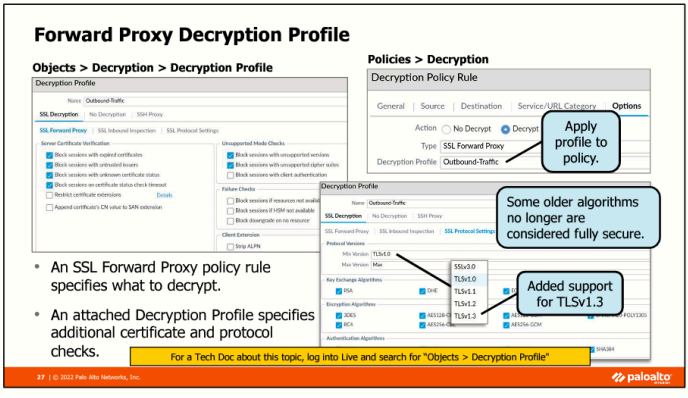


SSL Decryption policy rulesets are similar to the other rulesets in PAN-OS software. The rules are parsed from top to bottom, comparing network packets to each rule. When the packets match a particular rule, the actions defined in that rule are taken, and no further rules are checked.

A Decryption policy should be phased in to minimize end-user issues. Choose your most used or most critical encrypted traffic first. You can use a phased approach to resolve decryption issues one at a time.

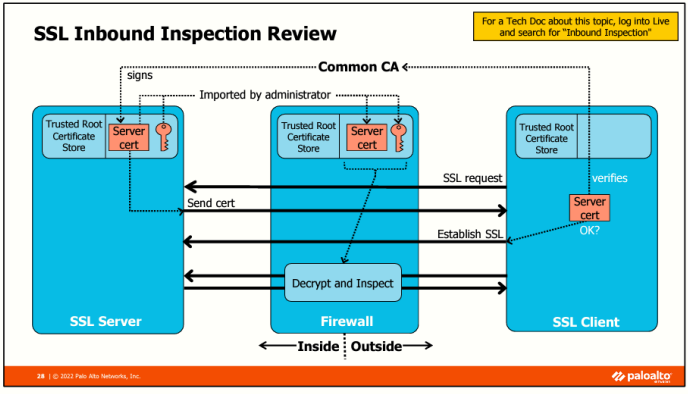
Not all traffic should be decrypted. Depending on local laws and regulations concerning health records, financial records, and other privacy concerns, some traffic cannot legally be decrypted.

Ensure that you also create a Security policy rule that allows the encrypted traffic to pass through the firewall.



A Decryption Profile enables the firewall to perform checks on decrypted traffic. A Decryption Profile allows the firewall to block sessions using unsupported protocols, cipher suites, or sessions that require SSL client authentication. You also can block sessions based on certificate status. For example, a certificate could be expired, be signed by an untrusted CA, have extensions restricting the use of the certificate, or have an unknown status. You also can block sessions if the resources to perform decryption are not available or if a hardware security module (HSM) is not available to sign certificates.

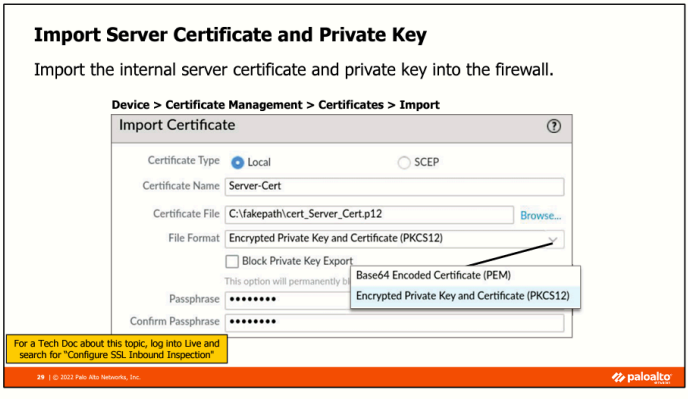
After creating a Decryption Profile, you attach it to a Decryption policy rule. The firewall enforces the Decryption Profile settings on traffic matched to the Decryption policy rule.



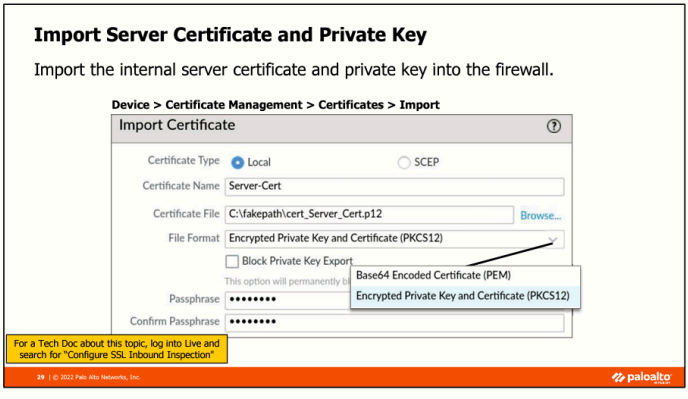
The firewall uses SSL Inbound Inspection to decrypt and inspect SSL traffic when you ~~~~have access to the certificate and private key of the SSL server. Before configuring SSL Inbound Inspection, you must deploy the SSL certificates to confirm an endpoint identity. The SSL server and client must share a common CA for the client to be able to validate the server identity. You also must import the private key and certificate of the server into the firewall. By importing these objects, you enable the firewall to decrypt and inspect the passing traffic.

The SSL client establishes a session with the server by initiating an SSL handshake. The SSL server responds with its certificate signed by a CA common to the server and the client. The client uses the CA to validate the certificate and the server identity.

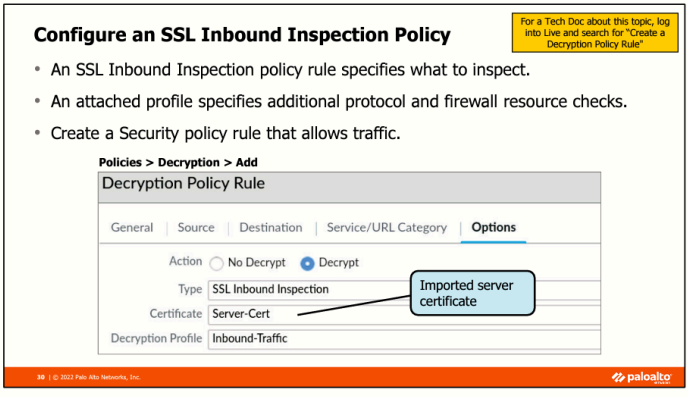
The SSL client has established a secure connection to the SSL server through the firewall. The firewall does not proxy the connection between the client and the server. However, the firewall can use the certificate and the private key of the SSL server to decrypt and inspect data flowing between the client and the server.



Creating an SSL Inbound Inspection policy is a two-step process, with an optional third step. The first step is to import the certificate and private key of the internal server into the firewall, which enables the firewall to decrypt and inspect SSL traffic to and from the internal SSL server.



Creating an SSL Inbound Inspection policy is a two-step process, with an optional third step. The first step is to import the certificate and private key of the internal server into the firewall, which enables the firewall to decrypt and inspect SSL traffic to and from the internal SSL server.



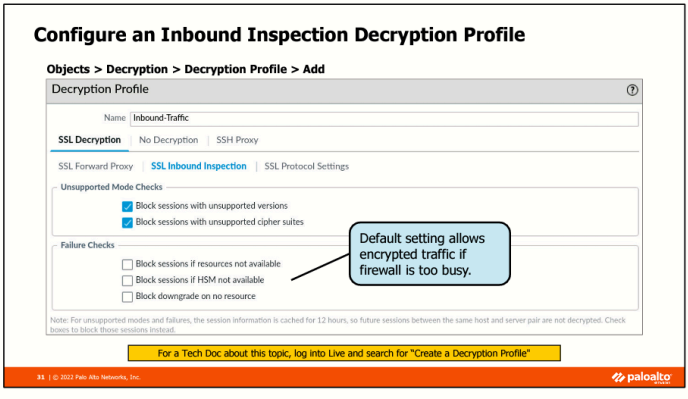
The second step to configure an SSL Inbound Inspection policy is to create the actual Decryption policy rule. You can specify a source zone and IP address, a destination zone, an IP address, protocol, port, and URL category as match conditions. Be sure that the destination IP address includes the IP address of the internal SSL server. On the **Options** tab, select **Decrypt** and select **SSL Inbound Inspection**. For the **Certificate** field, select the name of the internal SSL server certificate that was imported into the firewall.

An optional third step is to create a Decryption Profile. After creating a Decryption Profile, attach it to a Decryption policy rule. The firewall enforces the Decryption Profile settings on traffic matched to the Decryption policy rule.

Ensure that you also create a Security policy rule that allows the encrypted traffic to pass through the firewall.

Supplemental Notes

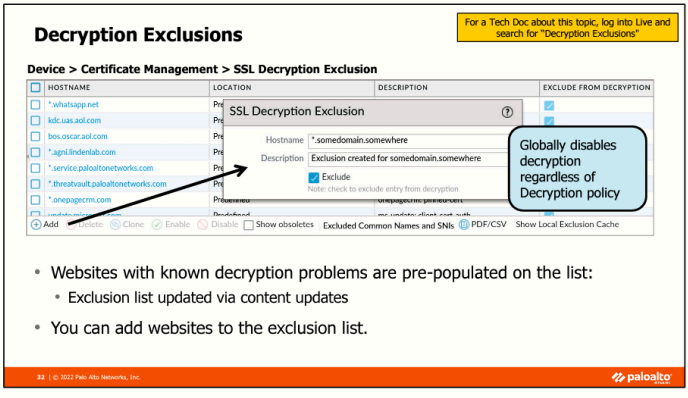
SSL Inbound Inspection uses fewer resources than SSL Forward Proxy, but each firewall model has a supported limit to imported certificates. For information about securing the enterprise, log into Live and search for "Product Selection" or see the documentation at https://www.paloaltonetworks.com/network-security.



To create a Decryption Profile, browse to **Objects > Decryption > Decryption Profile** and click **Add**. A Decryption Profile enables the firewall to check decrypted traffic and traffic you have excluded from decryption. You should block sessions using unsupported versions or cipher suites.

By default, the firewall does not decrypt traffic when the firewall data plane is too busy to perform decryption. Not decrypting traffic could allow an attacker to use encrypted traffic to exfiltrate sensitive data or insert malware. However, the default setting ensures access to applications and services even if the firewall becomes too busy with decrypting traffic. If you prefer a security-first posture rather than an availability-first posture, then select the **Block sessions if resources are not available,** check box.

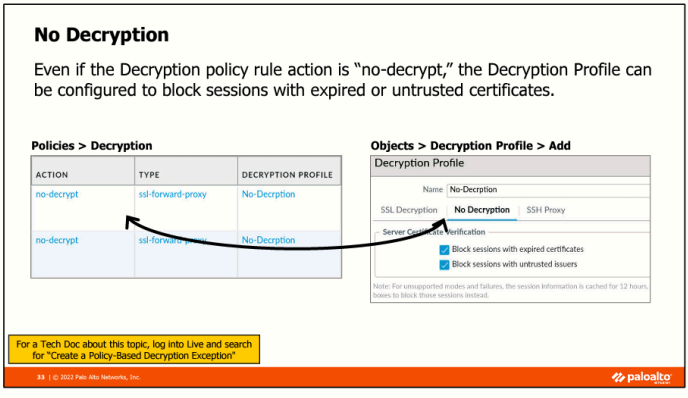
You also can block encrypted traffic if an HSM is not available to sign certificates by selecting the **Block sessions. If HSM not available** check box.



Starting with PAN-OS 8.0, you have centralized management for decryption exclusions. Decryption exclusions prevent the firewall from attempting to decrypt traffic to specific websites. You can view predefined decryption exclusions that identify applications that decryption is known to break. Updates and additions to the predefined decryption exclusion list are delivered to the firewall in content updates and are enabled by default. You also can create custom decryption exclusions based on domain names. Domain names are compared against the Server Name Indication, or SNI, in the SSL client request or against the Common Name, or CN, presented in the server certificate. When the hostname matches either the SNI or the CN, all traffic originating from or destined to that server is exempt from decryption.

To display the list of websites excluded from decryption, browse **Device > Certificate Management > SSL Decryption Exclusion**. To add a website, click **Add**. The **Hostname** field accepts the asterisk wildcard character. In the example, any website in the somedomain.somewhere domain will be excluded from decryption. To disable or enable individual exclusions, deselect or select the check box in the right-side column. To disable or enable multiple exclusions at a time, select multiple exclusion entries using the check box in the left column and then click **Disable** or **Enable**.

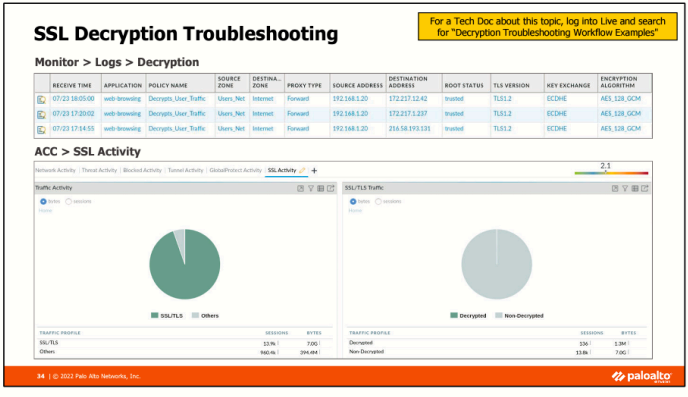
Palo Alto Networks uses content updates to remove decryption exclusions if they become obsolete. However, a predefined decryption exclusion that you disabled is not removed automatically from the list by a content update. Select **Show obsoletes** to see if disabled, predefined exclusions on your list already were removed by Palo Alto Networks via a new content update.



Even if the Decryption policy rule action is "no-decrypt," the Decryption Profile attached to the rule still can be configured to block sessions with expired or untrusted certificates. To ensure that the firewall verifies certificates, click the **No Decryption** tab in the Decryption Profile and select the desired checkboxes.

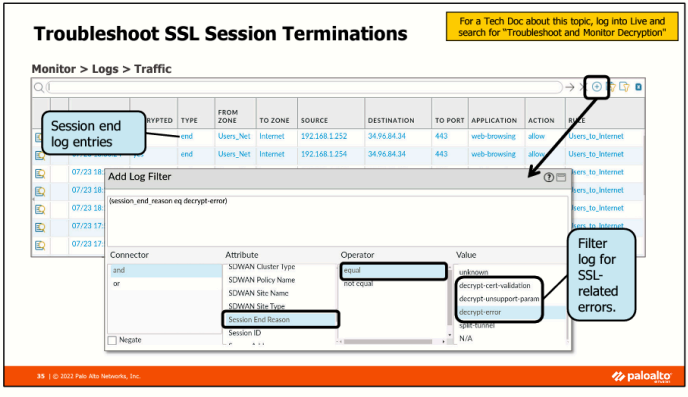
Select **Block sessions with expired certificates** to terminate the SSL connection if the server certificate is expired. This action prevents a user from accepting an expired certificate and continuing with an SSL session. Select **Block sessions with untrusted issuers** to terminate the SSL session if the server certificate issuer is untrusted. The Traffic log records entries for terminated sessions.

The firewall must act as an SSL proxy to perform these certificate checks even though the application data is not decrypted. The Traffic log for the session will include a decrypted flag, but the application will be listed as *ssl* instead of *web-browsing* or an actual application name.



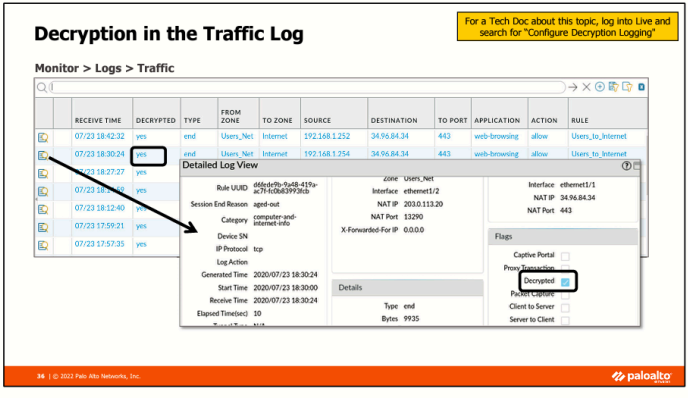
With the release of PAN-OS 10.0, a new Decryption log and new Application Command Center (ACC) widgets provide enhanced visibility into SSL/TLS traffic, enabling you to troubleshoot decryption issues and identify traffic that uses weak algorithms and protocols. The ACC widgets show you details about successful and unsuccessful SSL Decryption activity in your network. The widgets identify applications and SNIs that cause decryption issues and use weak ciphers and algorithms. The ACC widgets include **Traffic Activity**, **Successful TLS Version Activity**, **Decryption Failure Reasons**, **SSL/TLS Traffic**, and **Successful Key Exchange Activity**.

The Decryption log provides comprehensive information about sessions that match a Decryption policy. You can view log information such as application, SNI, Decryption policy name, error-index, TLS version, key exchange version, encryption algorithm, and certificate key types. You can use the Decryption log to drill down into details and gain context about the traffic.



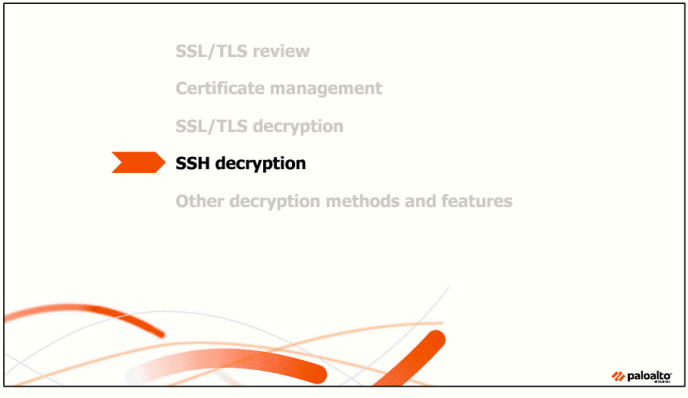
SSL sessions can be terminated for reasons that might include expired server certificates, unsupported ciphers or protocol versions, untrusted certificate issuers, and unknown certificate status and SSL timeout events. SSH decryption can be terminated because of unsupported SSH algorithms.

- The Traffic log records the start and end of each session. The word “end” in the **Type** column indicates a log entry for the end of a session. The **Session End Reason** column records why a session ended. You can use the **Session End Reason** filter values to filter the list of sessions to display only sessions that ended because of problems specific to SSL. SSL sessions that were terminated because of a Decryption Profile “block” action or the reception or transmission of fatal SSL/TLS alert messages are mapped to one of the following **Session End Reason** values:
- **decrypt-cert-validation**: An SSL session is terminated with this end reason attribute under one or more of the following scenarios:
  - Expired server certificate
  - Untrusted issuer
  - Unknown certificate status
  - Certificate status timeout
  - Client authentication
- **decrypt-unsupport-param**: An SSL session is terminated with this end reason attribute under one or more of the following scenarios:
  - Unsupported protocol version
  - Unsupported cipher
  - Unsupported SSH algorithm
- **decrypt-error**: An SSL session is terminated with this end reason attribute under one or more of the following scenarios:
  - Resources unavailable
  - HSM unavailable
  - SSH errors

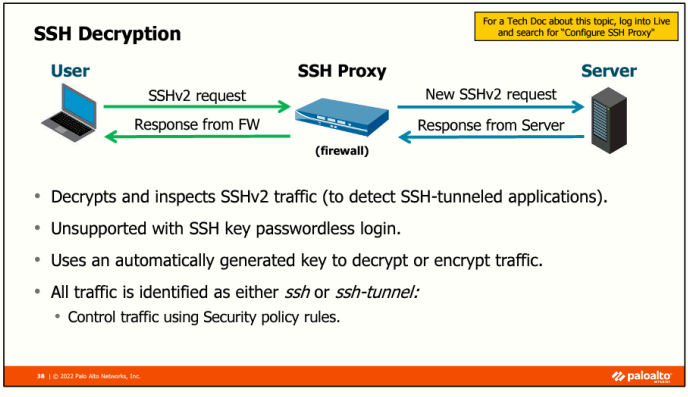


You can use the Traffic log to determine whether SSL sessions are decrypted. If the log entry contains a packet capture, the packet capture will be encrypted because packet capture occurs before decryption.

You can also search for decryption traffic using the log filter (**`flag has proxy`**).



This section describes how to configure SSH decryption on the firewall.



Secure Shell, also called SSH, supports secure remote login. SSH also enables other applications to be carried in encrypted SSH tunnels. SSH tunnels are a common way to subvert firewalls and breach security policies. SSH does not require digital certificates, as SSL does. The firewall can decrypt, inspect, and re-encrypt inbound and outbound SSHv2 connections passing through the firewall. With SSH Proxy, separate SSH sessions are created between the client, the firewall, and the server.

SSH decryption enabled, SSH, SCP, and SFTP are identified as the application *ssh*. After the firewall identifies SSH traffic, it checks for an SSH tunnel. If the firewall identifies and labels traffic as the application *ssh-tunnel*, you can configure a Security policy rule to allow the *ssh-tunnel* application. SSH decryption does not control applications or inspection of threats within the SSH-tunneled application.

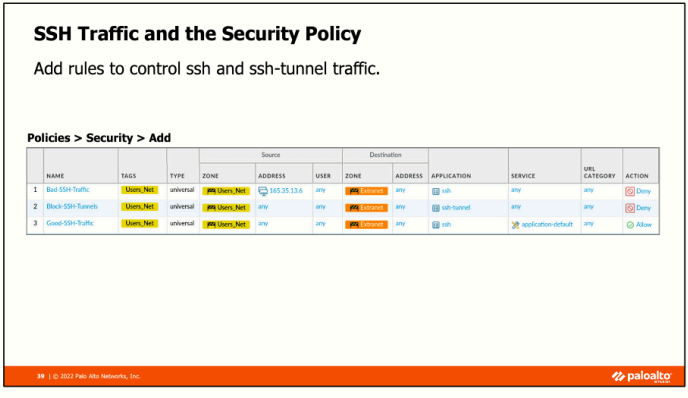
PAN-OS software does not support decryption for SSH passwordless, key-authenticated sessions. However, any password-authenticated SSH connections can be decrypted.

The key used to decrypt SSH sessions is generated automatically on the firewall during bootup. The same key is used to decrypt all SSH sessions across all virtual systems configured on the firewall. The key also is automatically synchronized between HA partner firewalls.

Supplemental Notes

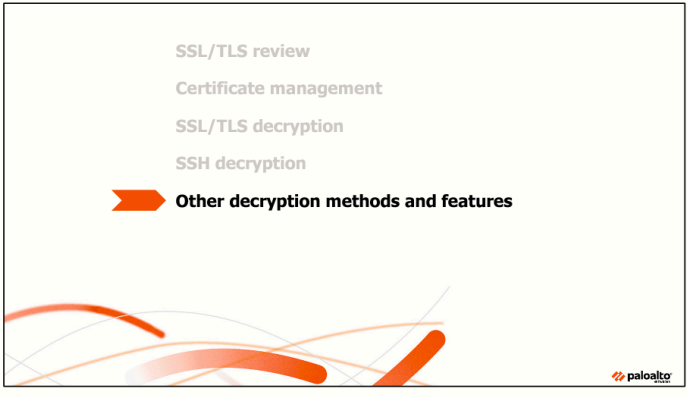
Configure your Decryption policy to exclude the systems that require public key authentication by adding the IP addresses to be excluded as destination IP addresses and then selecting the **Negate** box.

If an SSH client already has cached the server SSH public key but does not require passwordless entry, the client could remove the server entry from the known-hosts file and log in to the SSH server again.

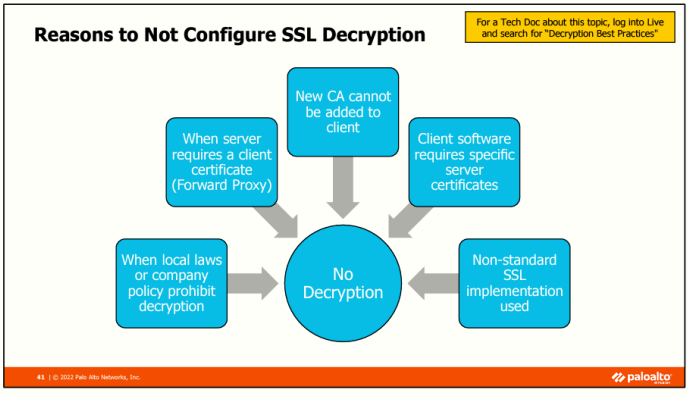


After configuring the SSH Proxy Decryption policy and Decryption Profile, create Security policy rules that control ssh and ssh-tunnel traffic. In the example, the first rule denied SSH traffic from the inside zone to the IP address 165.35.13.6 in the dmz zone. The second rule denies ssh-tunnel traffic from the inside zone to the dmz zone.

The third rule allows any SSH traffic from the other hosts inside the DMZ. This traffic also matches a Decryption policy rule, so it is decrypted. If an SSH tunnel is detected after SSH has been decrypted, then App-ID would identify the traffic as ssh-tunnel rather than ssh, and the traffic no longer would match the third rule.



This section describes other decryption methods.



Decryption is not a good choice when it is not allowed by local laws or company policy governing personal, financial, medical, government, and military information.

SSL Forward Proxy does not work when the SSL server requires SSL client authentication. In this scenario, the firewall has no access to the certificate and private key of the client and, therefore, cannot decrypt and inspect any traffic that has been encrypted by the use of the client's public key.

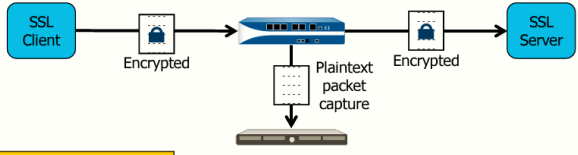
When you configure SSL Forward Proxy, the client and the firewall must share a common CA, or the client cannot trust its SSL connection to the firewall. Suppose the client list of trusted CAs do not include any CAs known and trusted by the firewall, and the client CA list cannot be updated. In that case, SSL Forward Proxy decryption can fail, depending on the **Server Certificate Verification** settings used in the Decryption Profile. These settings determine whether the firewall will accept unverified certificates.

Some browsers and web applications are preconfigured to expect a server to present a specific and limited list of certificates. Because the firewall must modify a server certificate to perform SSL Forward Proxy decryption, the client cannot use the revised certificate to authenticate the server. The modified certificate is not on the preconfigured lists of acceptable certificates. A connection cannot be established unless the browser or web application enables a user to click past the authentication error.

Decryption will fail if the client or server requires protocols or cipher suites that are not supported by the firewall. For a list of supported cipher suites log into Live and search for “Supported Cipher Suites” or go to https://www.paloaltonetworks.com/documentation/global/compatibility-matrix/supported-cipher-suites. You can also browse to Objects > Decryption Profile > Add > SSL Decryption > SSL Protocol Settings on the firewall and display supported ciphers and algorithms. To view the cipher suites supported by the client, perform packet capture and display the SSL handshake sequence.

Decryption Port Mirroring

- Exports decrypted flows out of a dedicated interface on the firewall.
- Use cases include data loss prevention (DLP) and network forensics.
- Requires a free license.

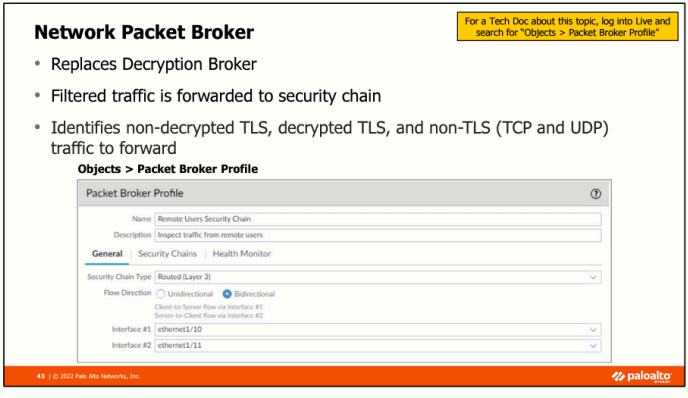


For a Tech Doc about this topic, log into Live and search for "Decryption Mirroring"

The decryption port mirroring feature enables a firewall to forward packet captures of decrypted traffic to a traffic collection tool, such as NetWitness or Solera, for archiving and analysis. This feature is necessary for organizations that require comprehensive data capture for forensic and historical purposes or to enhance data loss prevention functionality.

The decryption port mirroring feature is available on all hardware and VM-Series firewall models but requires downloading and installing a free license. This free license is a perpetual license with no expiration date and can be requested from the Customer Support Portal at support.paloaltonetworks.com.

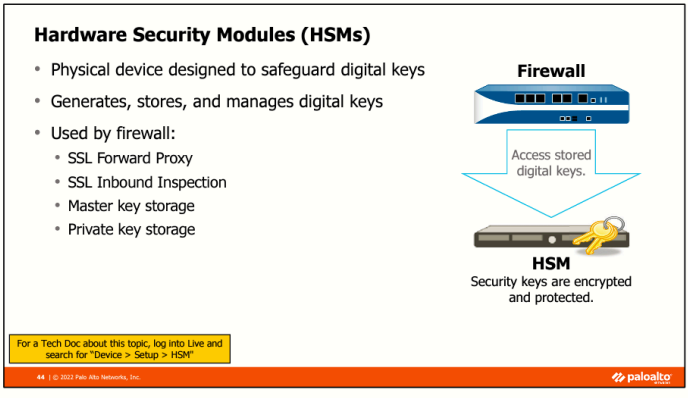
This feature is not supported on the VM-Series firewall on VMware NSX, Amazon Web Services, Microsoft Azure, or Google Cloud Platform public clouds.



With PAN-OS 10.1, the Network Packet Broker feature replaces the Decryption Broker. It expands its capabilities to filter and forward decrypted TLS traffic and non-decrypted TLS and non-TLS traffic to one or more third-party appliances (a security chain). The ability to filter and forward all traffic to a security chain eliminates complications from dedicated decryption devices and security chain management devices, thus simplifying your network and reducing capital and operating costs. The Network Packet Broker checks path health to and from the security chain and filters traffic based on applications, users, devices, IP addresses, and zones. These features are precious in high-security environments like financial and government institutions requiring offloading traffic to external security chains.

Network Packet Broker is supported for PA-7000 Series, PA-5400 Series, PA-5200 Series, PA-3200 Series devices, and VM-300 and VM-700 models. It requires SSL Forward Proxy decryption to be enabled, where the firewall is established as a trusted third party (or man-in-the-middle) to session traffic.

For more information about configuring the Network Packet Broker, log into Live and search for "Network Packet Broker" or see *PAN-OS Administrator's Guide Version 10.2* at https://docs.paloaltonetworks.com/pan-os/10-1/pan-os-networking-admin/network-packet-broker.html.



An HSM is a physical device that generates, stores, and manages digital keys. It provides logical and physical protection of the firewall's private keys from unauthorized use and potential adversaries. Use dedicated HSMs to manage the certificate signing functions for SSL Forward Proxy, SSL Inbound Inspection, and master key storage functions. HSM support generally is required when FIPS 140-2 Level 3 protection for CA keys is needed.

Each firewall maintains a default master key to encrypt its private keys, session keys used in asymmetric encryption, and locally stored passwords. To increase the level of master key security, you change the default master key on each firewall and encrypt the master key with a wrapping key that resides on an HSM.

HSM use is supported on the PA-3200 Series, PA-5200 Series, PA-7000 Series, and VM-Series firewalls. It is also supported on the Panorama M-100 and M-500 appliances and the Panorama VM.

PAN-OS software supports the following HSM devices and versions:

- nCipher nShield Connect:
  - PAN-OS 10.1 and 10.0 support client version 12.40.2. (backward compatible up to client version 11.50 for older appliances).
  - PAN-OS 9.1, 9.0, and 8.1 support client version 12.30.
  - PAN-OS 8.0 and earlier releases support client version 11.62.
- SafeNet Network:
  - PAN-OS 10.1 and 10.0 support SafeNet Network client versions 5.4.2 and 7.2.
  - PAN-OS 9.1 and 9.0 support SafeNet Network client versions 5.4.2 and 6.3.
  - PAN-OS 8.1 supports SafeNet Network client versions 5.4.2 and 6.2.2.
  - PAN-OS 8.0.2 and later PAN-OS 8.0 releases (also, PAN-OS 7.1.10 and later PAN-OS 7.1 releases support SafeNet Network client versions 5.2.1, 5.4.2, and 6.2.2)

For more information about integrating an HSM with your firewall, log into live and search for PAN-OS admin guide or see *PAN-OS Administrator's Guide Version 10.2* at https://docs.paloaltonetworks.com/pan-os/10-2/pan-os-admin.html.

Module Summary

Now that you have completed this module, you should be able to:

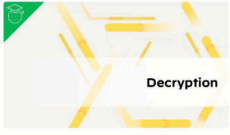
- Review fundamental SSL concepts and operation
- Create and manage certificates using the web interface
- Configure SSL/TLS forward proxy decryption
- Configure SSL/TLS inbound inspection decryption
- Prevent decryption for specific traffic
- View information and troubleshoot SSL/TLS issues using the CLI and logs
- Identify decryption configuration considerations
- Configure SSH decryption
- List other available decryption methods

Now that you have completed the module, you should be able to perform the tasks listed.

Additional Resources

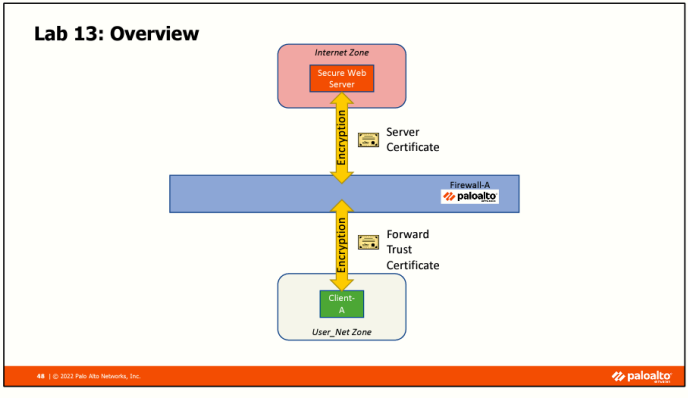
For a digital review of this module, log into Beacon and search for:

"Decryption"



Review Questions

1. Which two types of activities does SSL/TLS decryption by the firewall helps to block? (Choose two.)
   a. malware introduction
   b. denial-of-service attacks
   c. sensitive data exfiltration
   d. protocol-based attacks
2. True or false? CRL is consulted first if OCSP and CRL are configured on a firewall.
   a. true
   b. false
3. Which type of firewall decryption requires the administrator to import a server certificate and a private key into the firewall?
   a. SSH decryption
   b. SSH tunnel decryption
   c. SSL Forward Proxy decryption
   d. SSL Inbound Inspection decryption
4. True or false? The SSL forward untrust certificate should not be trusted by the client but should still be a CA certificate.
   a. true
   b. false
5. True or false? The firewall can still check for expired or untrusted certificates even if the SSL traffic is not decrypted.
   a. true
   b. false



Your instructor can provide an overview of the lab environment and the details needed for this lab. See the following page for a list of tasks that you will carry out in this lab.

Lab 13: Using Decryption to Block Threats in Encrypted Traffic

- Load a lab configuration
- Test the firewall without decryption
- Create a self-signed certificates for trusted connections
- Create A self-signed certificates for untrusted connections
- Create and test a Decryption policy rule for outbound traffic
- Test outbound Decryption policy rule
- Export the firewall certificate and import to Firefox
- Test outbound Decryption policy again
- Review firewall logs
- Exclude URL categories from decryption using a No-Decrypt rule
- Test the No-Decrypt rule

Answers to Review Questions

1. a, c
2. b (false)
3. d
4. a (true)
5. a (true)