

Understanding Traffic Decryption

By default, the ASA FirePOWER module cannot inspect traffic encrypted with the Secure Socket Layer (SSL) protocol or its successor, the Transport Layer Security (TLS) protocol. As part of access control, the *SSL inspection* feature allows you to either block encrypted traffic without inspecting it, or inspect encrypted or decrypted traffic with access control. As the module handles encrypted sessions, it logs details about the traffic. The combination of inspecting encrypted traffic and analyzing encrypted session data allows greater awareness and control of the encrypted applications and traffic in your network.

SSL inspection is a policy-based feature. In the Firepower System, an access control policy is a master configuration that invokes subpolicies and other configurations, including an SSL policy. If you associate an SSL policy with access control, the system uses that SSL policy to handle encrypted sessions before it evaluates them with access control rules. If you do not configure SSL inspection, or your devices do not support it, access control rules handle all encrypted traffic.

Note that access control rules also handle encrypted traffic when your SSL inspection configuration allows it to pass. However, some access control rule conditions require unencrypted traffic, so encrypted traffic may match fewer rules. Also, by default, the system disables intrusion and file inspection of encrypted payloads. This helps reduce false positives and improve performance when an encrypted connection matches an access control rule that has intrusion and file inspection configured. For more information, see Creating and Editing Access Control Rules, page 6-2 and Using the SSL Preprocessor, page 22-71.

If the module detects an SSL or TLS handshake over a TCP connection, it determines whether it can decrypt the detected traffic. If it cannot, it applies a configured action:

- block the encrypted traffic, and optionally reset the TCP connection
- not decrypt the encrypted traffic

If the module can decrypt the traffic, it blocks the traffic without further inspection, evaluates undecrypted traffic with access control, or decrypts it using one of the following methods:

- Decrypt with a known private key. When an external host initiates an SSL handshake with a server on your network, the system matches the exchanged server certificate with a server certificate previously uploaded to the appliance. It then uses the uploaded private key to decrypt the traffic.
- Decrypt by re-signing the server certificate. When a host on your network initiates an SSL handshake with an external server, the system re-signs the exchanged server certificate with a previously uploaded certificate authority (CA) certificate. It then uses the uploaded private key to decrypt the traffic.

Decrypted traffic is subject to the same traffic handling and analysis as originally unencrypted traffic: network, reputation, and user-based access control; intrusion detection and prevention; and advanced malware protection. If the system does not block the decrypted traffic post-analysis, it reencrypts the traffic before passing it to the destination host.

Note Certain SSL inspection actions, such as blocking traffic and decrypting outgoing traffic, modify the flow of traffic. ASA FirePOWER modules deployed inline can perform these actions. ASA FirePOWER modules deployed passively cannot affect the flow of traffic. However, these devices can still decrypt incoming traffic; see Example: Decrypting Traffic in a Passive Deployment, page 14-8 for more information.

For more information, see the following sections:

- SSL Handshake Processing, page 14-2
- SSL Inspection Requirements, page 14-5
- Analyzing SSL Inspection Appliance Deployments, page 14-7

SSL Handshake Processing

In this documentation, the term *SSL handshake* represents the two-way handshake that initiates encrypted sessions in both the SSL protocol and its successor protocol, TLS.

In a passive deployment, the Firepower System observes a copy of the handshake, but does not process the actual handshake. In an inline deployment, the Firepower System processes the SSL handshake, potentially modifying the ClientHello message and acting as a TCP proxy server for the session.

After the client establishes a TCP connection with the server (after it successfully completes the TCP three-way handshake), the managed device monitors the TCP session for any attempt to initiate an encrypted session. The SSL handshake establishes an encrypted session via the exchange of specialized packets between client and server. In the SSL and TLS protocols, these specialized packets are called *handshake messages*. The handshake messages communicate which encryption attributes both the client and server support:

- ClientHello—The client specifies multiple supported values for each encryption attribute.
- ServerHello—The server specifies a single supported value for each encryption attribute, which determines which encryption method the system uses during the secure session.

Although the data transmitted in the session is encrypted, the handshake messages are not.

After an SSL handshake completes, the managed device caches encrypted session data, which allows session resumption without requiring the full handshake. The managed device also caches server certificate data, which allows faster handshake processing in subsequent sessions.

ClientHello Message Handling

The client sends the ClientHello message to the server that acts as the packet destination if a secure connection can be established. The client sends the message to initiate the SSL handshake or in response to a Hello Request message from the destination server.

If you configure SSL inspection, when a managed device receives a ClientHello message, the system attempts to match the message to SSL rules that have the **Decrypt - Resign** action. The match relies on data from the ClientHello message and from cached server certificate data. Possible data includes:

Table 14-1 Data Availability for SSL Rule Conditions

SSL Rule Condition	Data Present In
Zones	ClientHello
Networks	ClientHello
VLAN Tags	ClientHello
Ports	ClientHello
Users	ClientHello
Applications	ClientHello (Server Name Indicator extension)
Categories	ClientHello (Server Name Indicator extension)
Certificate	server Certificate (potentially cached)
Distinguished Names	server Certificate (potentially cached)
Certificate Status	server Certificate (potentially cached)
Cipher Suites	ServerHello
Versions	ServerHello

If the ClientHello message does not match a Decrypt - Resign rule, the system does not modify the message. It then determines whether the message passes access control evaluation (which can include deep inspection). If the message passes, the system transmits it to the destination server.

If the message matches a Decrypt - Resign rule, the system modifies the ClientHello message as follows:

- Compression methods—Strips the `compression_methods` element, which specifies the compression methods the client supports. The Firepower System cannot decrypt compressed sessions. This modification reduces the Compressed Session type of undecryptable traffic.
- Cipher suites—Strips cipher suites from the `cipher_suites` element if the Firepower System does not support them. If the Firepower System does not support any of the specified cipher suites, the system transmits the original, unmodified element. This modification reduces the Unknown Cipher Suite and Unsupported Cipher Suite types of undecryptable traffic.
- Session identifiers—Strips any value from the `Session Identifier` element and the SessionTicket extension that does not match cached session data. If a ClientHello value matches cached data, an interrupted session can resume without the client and server performing the full SSL handshake. This modification increases the chances of session resumption and reduces the Session Not Cached type of undecryptable traffic.
- Elliptic curves—Strips elliptic curves from the Supported Elliptic Curves extension if the Firepower System does not support them. If the Firepower System does not support any of the specified elliptic curves, the managed device removes the extension and strips any related cipher suites from the `cipher_suites` element.
- ALPN extensions—Strips any value from the Application-Layer Protocol Negotiation (ALPN) extension that is unsupported in the Firepower System (for example, the SPDY and HTTP/2 protocols). This modification only occurs if the message matches an SSL rule associated with content restriction features. For more information, see Access Control Using Content Restriction, page 13-1.

- Other Extensions—Strips the Extended Master Secret, Next Protocol Negotiation (NPN), and TLS Channel IDs extensions.

Note The system performs these ClientHello modifications by default. If your SSL policy is configured correctly, this default behavior results in more frequent decryption of traffic. To tune the default behavior for your individual network, contact Support.

After the system modifies the ClientHello message, it determines whether the message passes access control evaluation (which can include deep inspection). If the message passes, the system transmits it to the destination server.

Direct communication between the client and server is no longer possible during the SSL handshake, because after message modification the Message Authentication Codes (MACs) computed by the client and server no longer match. For all subsequent handshake messages (and for the encrypted session once established), the managed device acts as a man-in-the-middle (MITM). It creates two SSL sessions, one between client and managed device, one between managed device and server. As a result, each session contains different cryptographic session details.

Note The cipher suites that the Firepower System can decrypt are frequently updated and do not correspond directly to the cipher suites you can use in SSL rule conditions. For the current list of decryptable cipher suites, contact Support.

ServerHello and Server Certificate Message Handling

The ServerHello message is the response to a ClientHello message in a successful SSL handshake.

After a managed device processes a ClientHello message and transmits it to the destination server, the server determines whether it supports the decryption attributes the client specified in the message. If it does not support those attributes, the server sends a handshake failure alert to the client. If it supports those attributes, the server sends the ServerHello message. If the agreed-upon key exchange method uses certificates for authentication, the server Certificate message immediately follows the ServerHello message.

When the managed device receives these messsages, it attempts to match them with SSL rules. These messages contain information that was absent from either the ClientHello message or the session data cache. Specifically, the system can potentially match these messages on Distinguished Names, Certificate Status, Cipher Suites, and Versions conditions.

If the messages do not match any SSL rules, the managed device performs the default action for the SSL policy. For more information, see Creating a Basic SSL Policy, page 15-2.

If the messages match an SSL rule, the managed device continues as appropriate:

Action: Monitor

The SSL handshake continues to completion. The managed device tracks and logs but does not decrypt encrypted traffic.

Action: Block or Block with Reset

The managed device blocks the SSL session. If appropriate, it also resets the TCP connection.

Action: Do Not Decrypt

The SSL handshake contintues to completion. The managed device does not decrypt the application data exchanged during the SSL session.

In rare cases, the system matches a ClientHello message to a Decrypt - Resign rule and modifies the message, but matches the related ServerHello message to a Do Not Decrypt rule. In those cases, the system resets the TCP connection to trigger a refreshed handshake from the client. The refreshed ClientHello message no longer matches the Decrypt - Resign rule, and the SSL session proceeds without decryption.

Action: Decrypt - Known Key

The managed device attempts to match the server Certificate data to a previously uploaded server certificate.

If it matches the certificate to a previously generated certificate, the SSL handshake continues to completion. The managed device uses the uploaded private key to decrypt and reencrypt the application data exchanged during the SSL session.

In rare cases, the system cannot match the server Certificate message to a previously generated certificate. For example, a server might change its certificate between the initial connection with the client and subsequent connections. In this case, the system blocks the SSL connection, so that the client reconnects and the system processes the handshake with the new certificate data.

Action: Decrypt - Resign

The managed device processes the server Certificate message and re-signs the exchanged server certificate with a previously uploaded certificate authority (CA) certificate. The SSL handshake continues to completion. The managed device then uses the uploaded private key to decrypt and reencrypt the application data exchanged during the SSL session.

While processing the ServerHello and Certificate messages, the managed device caches distinguished names and certificate data to allow faster handshake processing in both reestablished and subsequent SSL sessions.

SSL Inspection Requirements

License: feature dependent

How you deploy devices on your network, in addition to your configuration settings and licenses, influences the actions you can take to control and decrypt encrypted traffic.

SSL inspection requires public key certificates and paired private keys for certain features. You must upload certificates and paired private keys to the ASA FirePOWER module to decrypt and control traffic based on encryption session characteristics.

For more information, see the following sections:

- Deploying ASA FirePOWER Modules that Support SSL Inspection, page 14-6
- License Requirements for SSL Inspection, page 14-6
- Collecting Prerequisite Information to Configure SSL Rules, page 14-7

Deploying ASA FirePOWER Modules that Support SSL Inspection

License: Any

ASA FirePOWER modules configured and deployed inline can modify the flow of traffic. These devices can monitor, block, allow, and decrypt incoming and outgoing traffic.

ASA FirePOWER modules configured and deployed passively cannot affect the flow of traffic. They can only monitor, allow, and decrypt incoming traffic. Note that passive deployments do not support decrypting traffic encrypted with the ephemeral Diffie-Hellman (DHE) or the elliptic curve Diffie-Hellman (ECDHE) cipher suites.

Review your list of mapped actions, existing network deployment, and overall requirements to determine whether one or the other type of deployment better suits your organization. See Analyzing SSL Inspection Appliance Deployments, page 14-7 for more information.

License Requirements for SSL Inspection

License: feature dependent

Depending on your licenses, you can use a combination of criteria to determine how to handle encrypted traffic. The ASA FirePOWER module uses warning icons (⚠) and confirmation dialog boxes to designate unsupported features for your deployment. For details, hover your pointer over a warning icon.

You apply an SSL policy as part of an access control policy, and the access control policy inspects traffic decrypted by the SSL policy. See Access Control License and Role Requirements, page 4-2 for more information on access control licensing.

The following table explains the license requirements to apply an SSL policy as part of an access control policy.

Table 14-2 License Requirements for SSL Inspection

To apply an SSL policy that...	Licenses
handles encrypted traffic on the basis of zone, network, port, or SSL-related criteria	Any
handles encrypted traffic using geolocation data	Any
handles encrypted traffic using application or user criteria	Control
filters encrypted traffic using URL category and reputation data	URL Filtering

Collecting Prerequisite Information to Configure SSL Rules

License: feature-dependent

SSL inspection relies on a significant amount of supporting public key infrastructure (PKI) information. Consider your organization's traffic patterns to determine the matching rule conditions you can configure. Collect the information listed in the following table:

Table 14-3 SSL Rule Condition Prerequisites

To match on...	Collect the...
detected server certificates, including self-signed server certificates	server certificate
trusted server certificates	CA certificate
detected server certificate subject or issuer	server certificate subject DN or issuer DN

For more information, see Tuning Traffic Decryption Using SSL Rules, page 17-1.

Decide whether you want to not decrypt, block, monitor, or decrypt the encrypted traffic you match your rules against. Map these decisions to SSL rule actions, undecryptable traffic actions, and the SSL policy default action. If you want to decrypt traffic, see the following table:

Table 14-4 SSL Decryption Prerequisites

To decrypt...	Collect...
incoming traffic to a server you control	the server's certificate file and paired private key file
outgoing traffic to an external server	a CA certificate file and paired private key file You can also generate a CA certificate and private key.

For more information, see Using Rule Actions to Determine Encrypted Traffic Handling and Inspection, page 16-8.

After you have collected this information, upload it to the system and configure reusable objects. See Managing Reusable Objects, page 3-1 for more information.

Analyzing SSL Inspection Appliance Deployments

License: feature-dependent

This section presents several scenarios in which the Life Insurance Example, Inc. life insurance company (LifeIns) uses SSL inspection on encrypted traffic to help audit their processes. Based on their business processes, LifeIns plans to deploy:

- one ASA FirePOWER device in a passive deployment for the Customer Service department
- one ASA FirePOWER device in an inline deployment for the Underwriting Department

Customer Service Business Processes

LifeIns created a customer-facing website for their customers. LifeIns receives encrypted questions and requests regarding policies from prospective customers through their website and through e-mail. LifeIns's Customer Service department processes them and returns the requested information within 24 hours. Customer Service wants to expand its incoming contact metrics collection. LifeIns has an established internal audit review for Customer Service.

LifeIns also receives encrypted applications online. The Customer Service department processes the applications within 24 hours before sending the case file to the Underwriting department. Customer Service filters out any obvious false applications sent through the online form, which consumes a fair portion of their time.

Underwriting Business Processes

LifeIns's underwriters submit encrypted medical information requests online to the Medical Repository Example, LLC medical data repository (MedRepo). MedRepo reviews the requests and transmits the encrypted records to LifeIns within 72 hours. The underwriters subsequently underwrite an application and submit policy and rate decisions. Underwriting wants to expand its metrics collection.

Lately, an unknown source has been sending spoofed responses to LifeIns. Though LifeIns's underwriters receive training on proper Internet use, LifeIns's IT department first wants to analyze all encrypted traffic that takes the form of medical responses, then wants to block all spoof attempts.

LifeIns places junior underwriters on six-month training periods. Lately, these underwriters have been incorrectly submitting encrypted medical regulation requests to MedRepo's customer service department. MedRepo has submitted multiple complaints to LifeIns in response. LifeIns plans on extending their new underwriter training period to also audit underwriter requests to MedRepo.

For more information, see the following sections:

- Example: Decrypting Traffic in a Passive Deployment, page 14-8
- Example: Decrypting Traffic in an Inline Deployment, page 14-11

Example: Decrypting Traffic in a Passive Deployment

License: feature-dependent

LifeIns's business requirements state that Customer Service must:

- process all requests and applications within 24 hours
- improve its incoming contact metrics collection process
- identify and discard incoming false applications

Customer Service does not require additional audit review.

LifeIns plans to passively deploy a Customer Service device.

Traffic from an external network goes to LifeIns's router. The router routes traffic to the Customer Service department, and mirrors a copy of the traffic to the ASA FirePOWER module for inspection.

On the ASA FirePOWER module, a user in the Access Control and SSL Editor custom role configures SSL inspection to:

- log all encrypted traffic sent to the Customer Service department
- decrypt encrypted traffic sent using the online application form to Customer Service
- not decrypt all other encrypted traffic sent to Customer service, including traffic sent using the online request form

The user also configures access control to inspect the decrypted application form traffic for fake application data and log when fake data is detected.

In the following scenarios, the user submits an online form to Customer Service. The user's browser establishes a TCP connection with the server, then initiates an SSL handshake. The ASA FirePOWER module receives a copy of this traffic. The client and server complete the SSL handshake, establishing the encrypted session. Based on handshake and connection details, the system logs the connection and acts upon the copy of the encrypted traffic.

For more information, see the following:

- Monitoring Encrypted Traffic in a Passive Deployment, page 14-9
- Not Decrypting Encrypted Traffic in a Passive Deployment, page 14-9
- Inspecting Encrypted Traffic with a Private Key in a Passive Deployment, page 14-10

Monitoring Encrypted Traffic in a Passive Deployment

License: Any

For all SSL-encrypted traffic sent to Customer Service, the system logs the connection.

The following steps occur:

1. The user submits the plain text request (`info`). The client encrypts this (`AaBb`) and sends the encrypted traffic to Customer Service.
2. LifeIns's router receives the encrypted traffic and routes it to the Customer Service department server. It also mirrors a copy to the ASA FirePOWER module.
3. The Customer Service department server receives the encrypted information request (`AaBb`) and decrypts it to plain text (`info`).
4. The module does not decrypt the traffic.

   The access control policy continues to process the encrypted traffic and allows it. The module generates a connection event after the session ends.
5. The ASA FirePOWER module receives the connection event.

Not Decrypting Encrypted Traffic in a Passive Deployment

License: Any

For all SSL-encrypted traffic that contains requests about policies, the system allows the traffic without decrypting it and logs the connection.

The following steps occur:

1. The user submits the plain text request (`info`). The client encrypts this (`AaBb`) and sends the encrypted traffic to Customer Service.
2. LifeIns's router receives the encrypted traffic and routes it to the Customer Service department server. It also mirrors a copy to the ASA FirePOWER module.
3. The Customer Service department server receives the encrypted information request (`AaBb`) and decrypts it to plain text (`info`).
4. The ASA FirePOWER module does not decrypt the traffic.

   The access control policy continues to process the encrypted traffic and allows it. The module generates a connection event after the session ends.

5. The ASA FirePOWER module receives the connection event.

Inspecting Encrypted Traffic with a Private Key in a Passive Deployment

License: Any

For all SSL-encrypted traffic that contains application form data, the system decrypts the traffic and logs the connection.

Note In a passive deployment, if traffic is encrypted with either the DHE or ECDHE cipher suite, you cannot decrypt it with a known private key.

For traffic with legitimate application form information, the system logs the connection.

The following steps occur:

1. The user submits the plain text request (`form`). The client encrypts this (`AaBb`) and sends the encrypted traffic to Customer Service.
2. LifeIns's router receives the encrypted traffic and routes it to the Customer Service department server. It also mirrors a copy to the ASA FirePOWER module.
3. The Customer Service department server receives the encrypted information request (`AaBb`) and decrypts it to plain text (`form`).
4. The ASA FirePOWER module uses the session key obtained with the uploaded known private key to decrypt the encrypted traffic to plain text (`form`).

   The access control policy continues to process the decrypted traffic and does not find fake application information. The module generates a connection event after the session ends.
5. The ASA FirePOWER module receives a connection event with information about the encrypted and decrypted traffic.

In contrast, if the decrypted traffic contains fake application data, the system logs the connection and the fake data.

The following steps occur:

1. The user submits the plain text request (`fake`). The client encrypts this (`CcDd`) and sends the encrypted traffic to Customer Service.
2. LifeIns's router receives the encrypted traffic and routes it to the Customer Service department server. It also mirrors a copy to the device.
3. The Customer Service department server receives the encrypted information request (`CcDd`) and decrypts it to plain text (`fake`).
4. The ASA FirePOWER module uses the session key obtained with the uploaded known private key to decrypt the encrypted traffic to plain text (`fake`).

   The access control policy continues to process the decrypted traffic and finds fake application information. The module generates an intrusion event. After the session ends, it generates a connection event.
5. The ASA FirePOWER module receives a connection event with information about the encrypted and decrypted traffic, and an intrusion event for the fake application data.

Example: Decrypting Traffic in an Inline Deployment

License: feature-dependent

LifeIns's business requirements state that Underwriting must:

- audit new and junior underwriters, verifying that their information requests to MedRepo comply with all applicable regulations
- improve its underwriting metrics collection process
- examine all requests that appear to come from MedRepo, then drop any spoofing attempts
- drop all improper regulatory requests to MedRepo's Customer Service department from the Underwriting department
- not audit senior underwriters

LifeIns plans to deploy a device in an inline deployment for the Underwriting department.

Traffic from MedRepo's network goes to MedRepo's router. It routes traffic to LifeIns's network. The device receives the traffic, passes allowed traffic to LifeIns's router, and sends events to the ASA FirePOWER module. LifeIns's router routes traffic to the destination host.

On the ASA FirePOWER module, a user configures SSL inspection to:

- log all encrypted traffic sent to the Underwriting department
- block all encrypted traffic incorrectly sent from LifeIns's underwriting department to MedRepo's customer service department
- decrypt all encrypted traffic sent from MedRepo to LifeIns's underwriting department, and from LifeIns's junior underwriters to MedRepo's requests department
- not decrypt encrypted traffic sent from the senior underwriters

The user also configures access control to inspect decrypted traffic with a custom intrusion policy and:

- block decrypted traffic if it contains a spoof attempt, and log the spoof attempt
- block decrypted traffic that contains information not compliant with regulations, and log the improper information
- allow all other encrypted and decrypted traffic

The system reencrypts allowed decrypted traffic before sending it to the destination host.

In the following scenarios, the user submits information online to a remote server. The user's browser establishes a TCP connection with the server, then initiates an SSL handshake. The module receives this traffic; based on handshake and connection details, the system logs the connection and acts on the traffic. If the system blocks the traffic, it also closes the TCP connection. Otherwise, the client and server complete the SSL handshake, establishing the encrypted session.

For more information, see the following:

- Monitoring Encrypted Traffic in an Inline Deployment, page 14-12
- Allowing Specific Users' Encrypted Traffic in an Inline Deployment, page 14-12
- Blocking Encrypted Traffic in an Inline Deployment, page 14-12
- Inspecting Encrypted Traffic with a Private Key in an Inline Deployment, page 14-13
- Inspecting Specific Users' Encrypted Traffic with a Re-signed Certificate in an Inline Deployment, page 14-14

Monitoring Encrypted Traffic in an Inline Deployment

License: Any

For all SSL-encrypted traffic sent to and from the Underwriting department, the system logs the connection.

The following steps occur:

1. The user submits the plain text request (`help`). The client encrypts this (`AaBb`) and sends the encrypted traffic to MedRepo's Requests department server.
2. LifeIns's router receives the encrypted traffic and routes it to the Requests department server.
3. The ASA FirePOWER module does not decrypt the traffic.

   The access control policy continues to process the encrypted traffic and allows it, then generates a connection event after the session ends.
4. The external router receives the traffic and routes it to the Requests department server.
5. The Underwriting department server receives the encrypted information request (`AaBb`) and decrypts it to plain text (`help`).
6. The ASA FirePOWER module receives the connection event.

Allowing Specific Users' Encrypted Traffic in an Inline Deployment

License: Control

For all SSL-encrypted traffic originating from the senior underwriters, the system allows the traffic without decrypting it and logs the connection.

The following steps occur:

1. The user submits the plain text request (`help`). The client encrypts this (`AaBb`) and sends the encrypted traffic to MedRepo's Requests department server.
2. LifeIns's router receives the encrypted traffic and routes it to the Requests department server.
3. The ASA FirePOWER module does not decrypt this traffic.

   The access control policy continues to process the encrypted traffic and allows it, then generates a connection event after the session ends.
4. The external router receives the traffic and routes it to the Requests department server.
5. The Requests department server receives the encrypted information request (`AaBb`) and decrypts it to plain text (`help`).
6. The ASA FirePOWER module receives the connection event.

Blocking Encrypted Traffic in an Inline Deployment

License: Any

For all SMTPS email traffic improperly sent from LifeIns's underwriting department to MedRepo's Customer Service department, the system blocks the traffic during the SSL handshake without further inspection and logs the connection.

The following steps occur:

1. Having received the request to establish an SSL handshake from a client's browser, the Customer Service department server sends the server certificate (`cert`) as the next step in the SSL handshake to the LifeIns underwriter.
2. MedRepo's router receives the certificate and routes it to the LifeIns underwriter.
3. The ASA FirePOWER module blocks the traffic without further inspection and ends the TCP connection. It generates a connection event.
4. The internal router does not receive the blocked traffic.
5. The underwriter does not receive the blocked traffic.
6. The ASA FirePOWER module receives the connection event.

Inspecting Encrypted Traffic with a Private Key in an Inline Deployment

License: Any

For all SSL-encrypted traffic sent from MedRepo to LifeIns's underwriting department, the system uses an uploaded server private key to obtain session keys, then decrypts the traffic and logs the connection. Legitimate traffic is allowed and reencrypted before being sent to the Underwriting department.

The following steps occur:

1. The user submits the plain text request (`stats`). The client encrypts this (`AaBbC`) and sends the encrypted traffic to the Underwriting department server.
2. The external router receives the traffic and routes it to the Underwriting department server.
3. The ASA FirePOWER module uses the session key obtained with the uploaded known private key to decrypt this traffic to plain text (`stats`).

   The access control policy continues to process the decrypted traffic with the custom intrusion policy and does not find a spoof attempt. The device passes the encrypted traffic (`AaBbC`), then generates a connection event after the session ends.
4. The internal router receives the traffic and routes it to the Underwriting department server.
5. The Underwriting department server receives the encrypted information (`AaBbC`) and decrypts it to plain text (`stats`).
6. The ASA FirePOWER module receives the connection event with information about the encrypted and decrypted traffic.

In contrast, any decrypted traffic that is a spoof attempt is dropped. The system logs the connection and the spoof attempt.

The following steps occur:

1. The user submits the plain text request (`spoof`), altering the traffic to appear to originate from MedRepo, LLC. The client encrypts this (`FfGgH`) and sends the encrypted traffic to the Underwriting department server.
2. The ASA FirePOWER module uses the session key obtained with the uploaded known private key to decrypt this traffic to plain text (`spoof`).

   The access control policy continues to process the decrypted traffic with the custom intrusion policy and finds a spoof attempt. The ASA FirePOWER module blocks the traffic, then generates an intrusion event. It generates a connection event after the session ends.
3. The internal router does not receive the blocked traffic.

4. The Underwriting department server does not receive the blocked traffic.
5. The ASA FirePOWER module receives a connection event with information about the encrypted and decrypted traffic, and an intrusion event for the spoofing attempt.

Inspecting Specific Users' Encrypted Traffic with a Re-signed Certificate in an Inline Deployment

License: Control

For all SSL-encrypted traffic sent from the new and junior underwriters to MedRepo's requests department, the system uses a re-signed server certificate to obtain session keys, then decrypts the traffic and logs the connection. Legitimate traffic is allowed and reencrypted before being sent to MedRepo.



Note When decrypting traffic in an inline deployment by re-signing the server certificate, the ASA FirePOWER module acts as a man-in-the-middle. It creates two SSL sessions, one between client and ASA FirePOWER module, one between ASA FirePOWER module and server. As a result, each session contains different cryptographic session details.

The following steps occur:

1. The user submits the plain text request (`help`). The client encrypts this (`AaBb`) and sends the encrypted traffic to the Requests department server.
2. The internal router receives the traffic and routes it to the Requests department server.
3. The ASA FirePOWER module uses the session key obtained with a re-signed server certificate and private key to decrypt this traffic to plain text (`help`).

   The access control policy continues to process the decrypted traffic with the custom intrusion policy and does not find an improper request. The module reencrypts the traffic (`CcDd`), allowing it to pass. It generates a connection event after the session ends.
4. The external router receives the traffic and routes it to the Requests department server.
5. The Requests department server receives the encrypted information (`CcDd`) and decrypts it to plain text (`help`).
6. The ASA FirePOWER module receives the connection event with information about the encrypted and decrypted traffic.



Note Traffic encrypted with a re-signed server certificate causes client browsers to warn that the certificate is not trusted. To avoid this, add the CA certificate to the organization's domain root trusted certificates store or the client trusted certificates store.

In contrast, any decrypted traffic that contains information that does not meet regulatory requirements is dropped. The system logs the connection and the non-conforming information.

The following steps occur:

1. The user submits the plain text request (`regs`), which does not comply with regulatory requirements. The client encrypts this (`EeFf`) and sends the encrypted traffic to the Requests department server.
2. The internal router receives the traffic and routes it to the Requests department server.

3. The ASA FirePOWER module uses the session key obtained with a re-signed server certificate and private key to decrypt this traffic to plain text (`regs`).

   The access control policy continues to process the decrypted traffic with the custom intrusion policy and finds an improper request. The module blocks the traffic, then generates an intrusion event. It generates a connection event after the session ends.

4. The external router does not receive the blocked traffic.
5. The Requests department server does not receive the blocked traffic.
6. The ASA FirePOWER module receives a connection event with information about the encrypted and decrypted traffic, and an intrusion event for the improper request.