Troubleshoot TLS/SSL Rules

You can diagnose whether or not applications on your network use TLS/SSL pinning with connection events. If applications are using TLS/SSL pinning, see TLS/SSL Rule Guidelines and Limitations.

- About TLS/SSL Pinning, on page 1
- Verify TLS/SSL Cipher Suites, on page 5

About TLS/SSL Pinning

Some applications use a technique referred to as *TLS/SSL pinning* or *certificate pinning*, which embeds the fingerprint of the original server certificate in the application itself. As a result, if you configured a TLS/SSL rule with a **Decrypt - Resign** action, when the application receives a resigned certificate from a managed device, validation fails and the connection is aborted.

To confirm that TLS/SSL pinning is occurring, attempt to log in to a mobile application like Facebook. If a network connection error is displayed, log in using a web browser. (For example, you *cannot* log in to a Facebook mobile application but *can* log in to Facebook using Safari or Chrome.) You can use Firepower Management Center connection events as further proof of TLS/SSL pinning

Note TLS/SSL pinning is not limited to mobile applications.

Related Topics

Troubleshoot TLS/SSL Pinning, on page 1

Troubleshoot TLS/SSL Pinning

You can view connection events to determine whether or not the devices are experiencing SSL pinning. You must add at least the **SSL Flow Flags** and **SSL Flow Messages** columns to the table view of connection events.

Before you begin

- Enable logging for your TLS/SSL rules as discussed in Logging Decryptable Connections with SSL Rules.

- Log in to a mobile application like Facebook; if a network connection error displays, log in to Facebook using Chrome or Safari. If you *can* log in using a web browser but not the native application, SSL pinning is likely occurring.

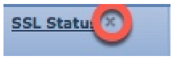
Procedure

Step 1 If you haven't done so already, log in to the Firepower Management Center.

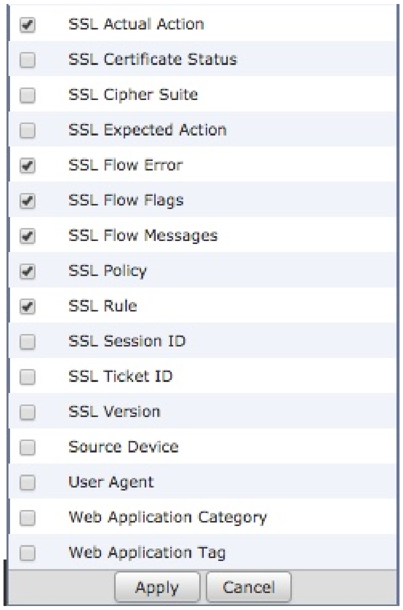
Step 2 Click **Analysis** > **Connections** > **Events**.

Step 3 Click **Table View of Connection Events**.

Step 4 Click **x** on any column in the connection events table to add additional columns for at least **SSL Flow Flags** and **SSL Flow Messages**.



The following example shows adding the **SSL Actual Action**, **SSL Flow Error**, **SSL Flow Flags**, **SSL Flow Messages**, **SSL Policy**, and **SSL Rule** columns to the table of connection events.



☑ SSL Actual Action
☐ SSL Certificate Status
☐ SSL Cipher Suite
☐ SSL Expected Action
☑ SSL Flow Error
☑ SSL Flow Flags
☑ SSL Flow Messages
☑ SSL Policy
☑ SSL Rule
☐ SSL Session ID
☐ SSL Ticket ID
☐ SSL Version
☐ Source Device
☐ User Agent
☐ Web Application Category
☐ Web Application Tag

The columns are added in the order discussed in Connection and Security Intelligence Event Fields.

Step 5 Click **Apply**.

Step 6 The following paragraphs discuss how you can identify SSL pinning behavior.

Step 7 If you determine that applications in your network use SSL pinning, see TLS/SSL Rule Guidelines and Limitations.

What to do next

You can use TLS/SSL connection events to confirm TLS/SSL pinning is occurring by looking for any of the following:

- Applications that send an SSL ALERT Message as soon as the client receives the SERVER_HELLO, SERVER_CERTIFICATE, SERVER_HELLO_DONE message from the server, followed by a TCP Reset, exhibit the following symptoms. (The alert, `Unknown CA (48)`, can be viewed using a packet capture.)
  - The SSL Flow Flags column displays `ALERT_SEEN` but *not* `APP_DATA_C2S` or `APP_DATA_S2C`.
  - The SSL Flow Messages column typically displays: `CLIENT_HELLO`, `SERVER_HELLO`, `SERVER_CERTIFICATE`, `SERVER_KEY_EXCHANGE`, `SERVER_HELLO_DONE`.
  - `Success` is displayed in the SSL Flow Error column.
- Applications that send no alerts but instead send TCP Reset after the SSL handshake is finished exhibit the following symptoms:
  - The SSL Flow Flags column does *not* display `ALERT_SEEN`, `APP_DATA_C2S`, or `APP_DATA_S2C`.
  - The SSL Flow Messages column typically displays: `CLIENT_HELLO`, `SERVER_HELLO`, `SERVER_CERTIFICATE`, `SERVER_KEY_EXCHANGE`, `SERVER_HELLO_DONE`, `CLIENT_KEY_EXCHANGE`, `CLIENT_CHANGE_CIPHER_SPEC`, `CLIENT_FINISHED`,`SERVER_CHANGE_CIPHER_SPEC`, `SERVER_FINISHED`.
  - `Success` is displayed in the SSL Flow Error column.

Related Topics

Using Connection and Security Intelligence Event Tables
Connection and Security Intelligence Event Fields
Information Available in Connection Event Fields
Event Searches
Troubleshoot Unknown or Bad Certificates or Certificate Authorities, on page 3

Troubleshoot Unknown or Bad Certificates or Certificate Authorities

You can view connection events to determine whether or not the devices are experiencing unknown certificate authorities, bad certificates, or unknown certificates. This procedure can also be used if a TLS/SSL certificate has been pinned. You must add at least the **SSL Flow Flags** and **SSL Flow Messages** columns to the table view of connection events.

Before you begin

- Set up a TLS/SSL decryption rule.
- Enable logging for your TLS/SSL rules as discussed in Logging Decryptable Connections with SSL Rules.

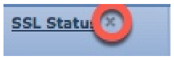
Procedure

Step 1 If you haven't done so already, log in to the Firepower Management Center.

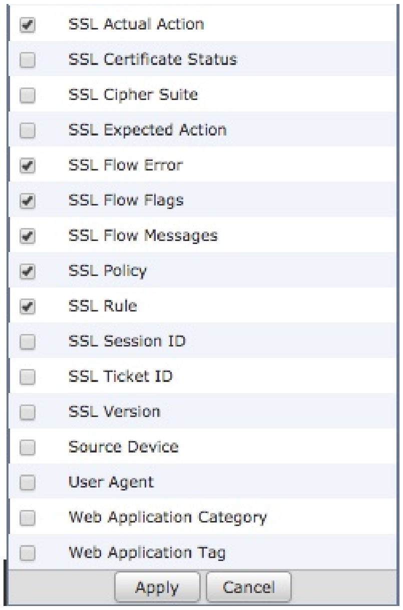
Step 2 Click **Analysis** > **Connections** > **Events**.

Step 3 Click **Table View of Connection Events**.

Step 4 Click **x** on any column in the connection events table to add additional columns for at least **SSL Flow Flags** and **SSL Flow Messages**.



The following example shows adding the **SSL Actual Action**, **SSL Flow Error**, **SSL Flow Flags**, **SSL Flow Messages**, **SSL Policy**, and **SSL Rule** columns to the table of connection events.



☑ SSL Actual Action
☐ SSL Certificate Status
☐ SSL Cipher Suite
☐ SSL Expected Action
☑ SSL Flow Error
☑ SSL Flow Flags
☑ SSL Flow Messages
☑ SSL Policy
☑ SSL Rule
☐ SSL Session ID
☐ SSL Ticket ID
☐ SSL Version
☐ Source Device
☐ User Agent
☐ Web Application Category
☐ Web Application Tag

The columns are added in the order discussed in Connection and Security Intelligence Event Fields.

Step 5 Click **Apply**.

Step 6 The following table discusses how you can determine if a certificate or certificate authority is bad or missing.

SSL flow flag	Meaning
CLIENT_ALERT_SEEN_UNKNOWN_CA	Indicates a valid certificate chain or partial chain was received by an SSL client application, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA. This message always indicates an unrecoverable error.
CLIENT_ALERT_SEEN_BAD_CERTIFICATE	A certificate was corrupt, contained signatures that did not verify correctly, or had other problems.
CLIENT_ALERT_SEEN_CERTIFICATE_UNKNOWN	Some other (unspecified) issue arose in processing the certificate, rendering it unacceptable.

Verify TLS/SSL Cipher Suites

Before you begin

This topic discusses actions you must take if you see the following error when saving a TLS/SSL rule that has cipher suite conditions:

```
Traffic cannot match this rule; none of your selected cipher suites contain a signature
algorithm that the resigning CA's signature algorithm
```

The error indicates that one or more of the cipher suites you chose for the TLS/SSL rule condition are incompatible with the certificate used in the TLS/SSL rule. To resolve the issue, you must have access to the certificate you're using.

Note The tasks in this topic assume knowledge of how TLS/SSL encryption works.

Procedure

Step 1 When you attempt to save an SSL rule with either **Decrypt - Resign** or **Decrypt - Known Key** with specified cipher suites, the following error is displayed:

Example:

```
Traffic cannot match this rule; none of your selected cipher suites contain a
signature algorithm that the resigning CA's signature algorithm
```

Step 2 Locate the certificate you're using to decrypt traffic and, if necessary, copy the certificate to a system that can run openssl commands.

Step 3 Run the following command to display the signature algorithm used by the certificate:

openssl x509 -in *CertificateName* **-text -noout**

The first few lines of output are displayed similar to the following:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4105 (0x1009)
    Signature Algorithm: ecdsa-with-SHA256
```

Step 4 The **Signature algorithm** tells you the following:

- The cryptographic function used (in the preceding example, **ECDSA** means Elliptic Curve Digital Signature Algorithm).
- The hash function used to create a digest of the encrypted message (in the preceding example, **SHA256**).

Step 5 Search a resource such as OpenSSL at University of Utah for cipher suites that match those values. The cipher suite must be in RFC format.
You can also search a variety of other sites, such as Server Side TLS at the Mozilla wiki or Appendix C of RFC 5246. Cipher Suites in TLS/SSL (Schannel SSP) in Microsoft documentation has a detailed explanation of cipher suites.

Step 6 If necessary, translate the OpenSSL name to an RFC name that the Firepower Management System uses. See the RFC mapping list on the on the `https://testssl.sh` site.

Step 7 The previous example, **ecdsa-with-SHA256**, can be found in the Modern Compatibility List on the Mozilla wiki.

a) Choose only cipher suites that have **ECDSA** and **SHA-256** in the name. These cipher suites follow:

```
ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES128-SHA256
```

b) Find the corresponding RFC cipher suite on RFC mapping list. These cipher suites follow:

```
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
```

Step 8 Add the preceding cipher suites to your TLS/SSL rule.