Encrypting Communication with Other MTAs

This chapter contains the following sections:

• Overview of Encrypting Communication with Other MTAs, on page 1
• Working with Certificates, on page 2
• Enabling TLS on a Listener’s HAT, on page 7
• Enabling TLS and Certificate Verification on Delivery, on page 10
• Managing Lists of Certificate Authorities, on page 13
• Enabling a Certificate for HTTPS, on page 15

Overview of Encrypting Communication with Other MTAs

Enterprise Gateways (or Message Transfer Agents, i.e. MTAs) normally communicate “in the clear” over the Internet. That is, the communications are not encrypted. In several scenarios, malicious agents can intercept this communication without the knowledge of the sender or the receiver. Communications can be monitored and even altered by a third party.

Transport Layer Security (TLS) is an improved version of the Secure Socket Layer (SSL) technology. It is a widely used mechanism for encrypting SMTP conversations over the Internet. AsyncOS supports the STARTTLS extension to SMTP (Secure SMTP over TLS), described in RFC 3207 (which obsoletes RFC 2487).

The TLS implementation in AsyncOS provides privacy through encryption. It allows you to import an X.509 certificate and private key from a certificate authority service or create a self-signed certificate to use on the appliance. AsyncOS supports separate TLS certificates for public and private listeners, secure HTTP (HTTPS) management access on an interface, the LDAP interface, and all outgoing TLS connections.

Related Topics

• How to Encrypt SMTP Conversations using TLS, on page 1

How to Encrypt SMTP Conversations using TLS

How to Encrypt SMTP Conversations using TLS
<table>
<thead>
<tr>
<th>Do This</th>
<th>More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Obtain an X.509 certificate and private key from a recognized certificate authority.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Install the certificate on the Email Security appliance</td>
</tr>
<tr>
<td>Step 3</td>
<td>Enable TLS for receiving messages, delivering messages, or both</td>
</tr>
<tr>
<td>Step 4</td>
<td>(Optional) Customize the list of trusted certificate authorities that the appliance uses to verify a certificate from a remote domain to establish the domain’s credentials.</td>
</tr>
<tr>
<td>Step 5</td>
<td>(Optional) Configure the Email Security appliance to send an alert when it’s unable to deliver messages to a domain that requires a TLS connection.</td>
</tr>
</tbody>
</table>

### Working with Certificates

To use TLS, the Email Security appliance must have an X.509 certificate and matching private key for receiving and delivery. You may use the same certificate for both SMTP receiving and delivery and different certificates for HTTPS services on an interface, the LDAP interface, and all outgoing TLS connections to destination domains, or use one certificate for all of them.

You can view the entire list of certificates on the Network > Certificates page in the web interface and in the CLI by using the print command after you configure the certificates using certconfig . Note that the print command does not display intermediate certificates.

⚠️ **Caution**

Your appliance ships with a demonstration certificate to test the TLS and HTTPS functionality, but enabling either service with the demonstration certificate is not secure and is not recommended for general use. When you enable either service with the default demonstration certificate, a warning message is printed in the CLI.

**Related Topics**

- Deploying a Signed Certificate , on page 3
- Deploying Self-Signed Certificates , on page 3
Deploying a Signed Certificate

Use a signed certificate when you cannot exchange self-signed certificates between the Email Security appliance and the other machine, for example because that machine is not in your domain. Your corporate security department may have other requirements.

<table>
<thead>
<tr>
<th>Step</th>
<th>Do This</th>
<th>More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>If you are deploying in a cluster, follow instructions.</td>
<td>Certificates and Centralized Management, on page 4</td>
</tr>
<tr>
<td>Step 2</td>
<td>Generate a self-signed certificate and Certificate Signing Request (CSR).</td>
<td>Creating a Self-Signed Certificate, on page 4</td>
</tr>
<tr>
<td>Step 3</td>
<td>Send the generated certificate to a recognized Certificate Authority for signing.</td>
<td>About Sending a Certificate Signing Request (CSR) to a Certificate Authority, on page 5</td>
</tr>
<tr>
<td>Step 4</td>
<td>Upload the signed certificate.</td>
<td>Uploading a Certificate Signed by a Certificate Authority, on page 5</td>
</tr>
<tr>
<td>Step 5</td>
<td>Ensure that the certificate authority that signed the certificate is on the list of trusted authorities.</td>
<td>Managing Lists of Certificate Authorities, on page 13</td>
</tr>
<tr>
<td>Step 6</td>
<td>If applicable, use an intermediate certificate.</td>
<td>Intermediate Certificates, on page 4</td>
</tr>
</tbody>
</table>

Deploying Self-Signed Certificates

You can generally use self-signed certificates for communications between appliances that are behind your corporate firewall. Your corporate security department may have other requirements.

<table>
<thead>
<tr>
<th>Step</th>
<th>Do This</th>
<th>More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>If you are deploying in a cluster, follow instructions.</td>
<td>Certificates and Centralized Management, on page 4</td>
</tr>
<tr>
<td>Step 2</td>
<td>Generate a self-signed certificate from the Email Security appliance.</td>
<td>Creating a Self-Signed Certificate, on page 4</td>
</tr>
<tr>
<td>Step 3</td>
<td>Export the self-signed certificate.</td>
<td>Exporting a Certificate, on page 7</td>
</tr>
<tr>
<td>Step 4</td>
<td>Import the self-signed certificate to the machine with which the Email Security appliance will communicate.</td>
<td>See the documentation for the other machine.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Generate and export a self-signed certificate from the other machine.</td>
<td>See the documentation for the other machine.</td>
</tr>
</tbody>
</table>
Certicates and Centralized Management

A certificate usually uses the local machine’s hostname for the certificate’s common name. If your Email Security appliances are part of a cluster, you will need to import a certificate for each cluster member as the machine level, with the exception of a wild card certificate or a Subject Alternative Name (SAN) certificate that you can install at the cluster level. Each cluster member’s certificate must use the same certificate name so the cluster can refer to it when a member’s listener is communicating with another machine.

Intermediate Certificates

In addition to root certificate verification, AsyncOS supports the use of intermediate certificate verification. Intermediate certificates are certificates issued by a trusted root certificate authority which are then used to create additional certificates - effectively creating a chained line of trust. For example, a certificate may be issued by godaddy.com who, in turn, is granted the rights to issue certificates by a trusted root certificate authority. The certificate issued by godaddy.com must be validated against godaddy.com’s private key as well as the trusted root certificate authority’s private key.

Creating a Self-Signed Certificate

You might want to create a self-signed certificate on the appliance for any of the following reasons:

- To encrypt SMTP conversations with other MTAs using TLS (both inbound and outbound conversations).
- To enable the HTTPS service on the appliance for accessing the GUI using HTTPS.
- Use as a client certificate for LDAPS if the LDAP server asks for a client certificate.
- To allow secure communication between the appliance and a Cisco AMP Threat Grid Appliance.

To create a self-signed certificate using the CLI, use the certconfig command.

| Step 1 | Select Network > Certificates. |
| Step 2 | Click Add Certificate. |
| Step 3 | Select Create Self-Signed Certificate. |
| Step 4 | Enter the following information for the self-signed certificate: |

<table>
<thead>
<tr>
<th>Common Name</th>
<th>The fully qualified domain name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>The exact legal name of the organization.</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>Section of the organization.</td>
</tr>
</tbody>
</table>
**Common Name** | **The fully qualified domain name.**
---|---
City (Locality) | The city where the organization is legally located.
State (Province) | The state, county, or region where the organization is legally located.
Country | The two letter ISO abbreviation of the country where the organization is legally located.
Duration before expiration | The number of days before the certificate expires.
Private Key Size | Size of the private key to generate for the CSR. Only 2048-bit and 1024-bit are supported.

**Step 5** Click **Next**.
**Step 6** Enter a name for the certificate. By default, AsyncOS assigns the common name previously entered.
**Step 7** If you will submit this certificate as a Certificate Signing Request (CSR), click **Download Certificate Signing Request** to save the CSR in PEM format to a local or network machine.
**Step 8** Submit and commit your changes.

**What to do next**
See the appropriate next step:
- Deploying a Signed Certificate, on page 3
- Deploying Self-Signed Certificates, on page 3

**About Sending a Certificate Signing Request (CSR) to a Certificate Authority**

A certificate authority is a third-party organization or company that issues digital certificates used to verify identity and distributes public keys. This provides an additional level of assurance that the certificate is issued by a valid and trusted identity. You may purchase certificates and private keys from a recognized certificate authority. Cisco does not recommend one service over another.

The Email Security appliance can create a self-signed certificate and generate a Certificate Signing Request (CSR) to submit to a certificate authority to obtain the public certificate. The certificate authority will return a trusted public certificate signed by a private key. Use the Network > Certificates page in the web interface or the cerfconfig command in the CLI to create the self-signed certificate, generate the CSR, and install the trusted public certificate.

If you are acquiring or creating a certificate for the first time, search the Internet for “certificate authority services SSL Server Certificates,” and choose the service that best meets the needs of your organization. Follow the service’s instructions for obtaining a certificate.

**What To Do Next**
See Deploying a Signed Certificate, on page 3.

**Uploading a Certificate Signed by a Certificate Authority**

When the certificate authority returns the trusted public certificate signed by a private key, upload the certificate to the appliance.
You can use the certificate with a public or private listener, an IP interface’s HTTPS services, the LDAP interface, or all outgoing TLS connections to destination domains.

---

**Step 1**
Make sure that the trusted public certificate that you receive is in PEM format or a format that you can convert to PEM using before uploading to the appliance. (Tools for doing this are included with OpenSSL, free software from http://www.openssl.org.)

**Step 2**
Upload the signed certificate to the appliance:

*Note*  
Uploading the certificate from the certificate authority overwrites the existing self-signed certificate.

a)  
Select **Network > Certificates**.

b)  
Click the name of the certificate that you sent to the Certificate Authority for signing.

c)  
Enter the path to the file on your local machine or network volume.

**Step 3**
You can also upload an intermediate certificate related to the self-signed certificate.

---

**What to do next**

**Related Topics**

- [Deploying a Signed Certificate](#), on page 3

**Importing a Certificate**

AsyncOS also allows you to import certificates from other machines that are saved in the PKCS #12 format to use on your appliance.

To import a certificate using the CLI, use the `certconfig` command.

---

**Note**  
If you are deploying a signed certificate, do not use this procedure to import the signed certificate. Instead, see [Uploading a Certificate Signed by a Certificate Authority](#), on page 5.

---

**Step 1**
Select **Network > Certificates**.

**Step 2**
Click **Add Certificate**.

**Step 3**
Select the **Import Certificate** option.

**Step 4**
Enter the path to the certificate file on your network or local machine.

**Step 5**
Enter the passphrase for the file.

**Step 6**
Click **Next** to view the certificate’s information.

**Step 7**
Enter a name for the certificate.

AsyncOS assigns the common name by default.

**Step 8**
Submit and commit your changes.
What to do next

- If you are deploying self-signed certificates, see Deploying Self-Signed Certificates, on page 3.

Exporting a Certificate

AsyncOS also allows you to export certificates and save them in the PKCS #12 format.

Note

If you are deploying a signed certificate, do not use this procedure to generate a Certificate Signing Request (CSR). Instead, see Deploying a Signed Certificate, on page 3.

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Navigate to the Network &gt; Certificates page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Click Export Certificate.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Select the certificate you want to export.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Enter the file name for the certificate.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Enter and confirm the passphrase for the certificate file.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Click Export.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Save the file to a local or network machine.</td>
</tr>
<tr>
<td>Step 8</td>
<td>You can export additional certificates or click Cancel to return to the Network &gt; Certificates page.</td>
</tr>
</tbody>
</table>

What to do next

- If you are deploying self-signed certificates, see Deploying Self-Signed Certificates, on page 3.

Enabling TLS on a Listener’s HAT

You must enable TLS for any listeners where you require encryption. You may want to enable TLS on listeners facing the Internet (that is, public listeners), but not for listeners for internal systems (that is, private listeners). Or, you may want to enable encryption for all listeners.

You can specify the following settings for TLS on a listener.

Table 1: TLS Settings for a Listener

<table>
<thead>
<tr>
<th>TLS Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. No</td>
<td>TLS is not allowed for incoming connections. No connections to the listener will require encrypted SMTP conversations. This is the default setting for all listeners you configure on the appliance.</td>
</tr>
<tr>
<td>2. Preferred</td>
<td>TLS is allowed for incoming connections to the listener from MTAs.</td>
</tr>
</tbody>
</table>
Meaning of TLS Setting

<table>
<thead>
<tr>
<th>TLS Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Required</td>
<td>TLS is allowed for incoming connections to the listener from MTAs, and until a <code>STARTTLS</code> command is received, the appliance responds with an error message to every command other than <code>NOOP</code>, <code>EHLO</code>, or <code>QUIT</code>. This behavior is specified by RFC 3207, which defines the SMTP Service Extension for Secure SMTP over Transport Layer Security. “Requiring” TLS means that email which the sender is not willing to encrypt with TLS will be refused by the appliance before it is sent, thereby preventing it from being transmitted in the clear.</td>
</tr>
</tbody>
</table>

By default, neither private nor public listeners allow TLS connections. You must enable TLS in a listener’s HAT to enable TLS for either inbound (receiving) or outbound (sending) email. In addition, all default mail flow policy settings for private and public listeners have the `tls` setting set to “off.”

You can assign a specific certificate for TLS connections to individual public listeners when creating a listener. For more information, see Listening for Connection Requests by Creating a Listener Using Web Interface.

Related Topics

- Assigning a Certificate to a Public or Private Listener for TLS Connections Using the GUI, on page 8
- Assigning a Certificate to a Public or Private Listener for TLS Connections Using the CLI, on page 8
- Logging, on page 13
- GUI Example: Changing the TLS setting for Listeners HAT, on page 9
- CLI Example: Changing the TLS Setting for Listeners HAT, on page 9

Assigning a Certificate to a Public or Private Listener for TLS Connections Using the GUI

**Step 1**
Navigate to the Network > Listeners page.

**Step 2**
Click the name of the Listener to edit.

**Step 3**
In the Certificate field, choose a certificate.

**Step 4**
Submit and commit your changes.

Assigning a Certificate to a Public or Private Listener for TLS Connections Using the CLI

**Step 1**
Use the `listenerconfig -> edit` command to choose a listener you want to configure.

**Step 2**
Use the `certificate` command to see the available certificates.

**Step 3**
Choose the certificate you want to assign to the listener when prompted.

**Step 4**
When you are finished configuring the listener, issue the `commit` command to enable the change.
Logging

The Email Security appliance will note in the mail logs instances when TLS is required but could not be used by the listener. The mail logs will be updated when the following conditions are met:

- TLS is set to “required” for a listener.
- The Email Security appliance has sent a “Must issue a STARTTLS command first” command.
- The connection is closed without having received any successful recipients.

Information on why the TLS connection failed will be included in the mail logs.

GUI Example: Changing the TLS setting for Listeners HAT

Step 1
Navigate to the Mail Policies > Mail Flow Policies page.

Step 2
Choose a listener whose policies you want to modify, and then click the link for the name of policy to edit. (You can also edit the Default Policy Parameters.)

Step 3
In the “Encryption and Authentication” section, for the “TLS:” field, choose the level of TLS you want for the listener.

Step 4
Submit and commit your changes.

The mail flow policy for the listener is updated with the TLS setting you chose.

CLI Example: Changing the TLS Setting for Listeners HAT

Step 1
Use the listenerconfig -> edit command to choose a listener you want to configure.

Step 2
Use the hostaccess -> default command to edit the listener’s default HAT settings.

Step 3
Change the TLS setting by entering one of the following choices when you are prompted with the following questions:

Do you want to allow encrypted TLS connections?

1. No
2. Preferred
3. Required

[1]> 3

You have chosen to enable TLS. Please use the 'certconfig' command to ensure that there is a valid certificate configured.

Step 4
Note that this example asks you to use the certconfig command to ensure that there is a valid certificate that can be used with the listener. If you have not created any certificates, the listener uses the demonstration certificate that is pre-installed on the appliance. You may enable TLS with the demonstration certificate for testing purposes, but it is not secure and is not recommended for general use. Use the listenerconfig -> edit -> certificate command to assign a certificate to the listener. Once you have configured TLS, the setting will be reflected in the summary of the listener in the CLI.
Enabling TLS and Certificate Verification on Delivery

You can require that TLS is enabled for email delivery to specific domains using the Destination Controls page or the destconfig command.

In addition to TLS, you can require that the domain’s server certificate is verified. This domain verification is based on a digital certificate used to establish the domain’s credentials. The validation process involves two validation requirements:

- The chain of issuer certificates for the SMTP session ends in a certificate issued by a trusted certificate authority (CA)
- The Common Name (CN) listed on the certificate matches either the receiving machine's DNS name or the message's destination domain.

-or-

The message's destination domain matches one of the DNS names in the certificate's Subject Alternative Name (subjectAltName) extension, as described in RFC 2459. The matching supports wildcards as described in section 3.1 of RFC 2818.

A trusted CA is a third-party organization or company that issues digital certificates used to verify identity and distributes public keys. This provides an additional level of assurance that the certificate is issued by a valid and trusted identity.

You can configure your Email Security appliance to send messages to a domain over a TLS connection as an alternative to envelope encryption. See the “Cisco Email Encryption” chapter for more information.

You can specify a certificate for the appliance to use for all outgoing TLS connections. To specify the certificate, click Edit Global Settings on the Destination Controls page or use destconfig -> setup in the CLI. The certificate is a global setting, not a per-domain setting.

You can specify 5 different settings for TLS for a given domain when you include a domain using the Destination Controls page or the destconfig command. In addition to specifying whether exchanges with a domain are required or preferred to be TLS encoded, you can dictate whether validation of the domain is necessary. See the following table for an explanation of the settings:
### Table 2: TLS Settings for Delivery

<table>
<thead>
<tr>
<th>TLS Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>The default TLS setting set using the Destination Controls page or the <code>destconfig -&gt; default</code> subcommand used for outgoing connections from the listener to the MTA for the domain. The value “Default” is set if you answer “no” to the question: “Do you wish to apply a specific TLS setting for this domain?”</td>
</tr>
<tr>
<td>1. No</td>
<td>TLS is not negotiated for outgoing connections from the interface to the MTA for the domain.</td>
</tr>
<tr>
<td>2. Preferred</td>
<td>TLS is negotiated from the Email Security appliance interface to the MTA(s) for the domain. However, if the TLS negotiation fails (prior to receiving a 220 response), the SMTP transaction will continue “in the clear” (not encrypted). No attempt is made to verify if the certificate originates from a trusted certificate authority. If an error occurs after the 220 response is received the SMTP transaction does not fall back to clear text.</td>
</tr>
<tr>
<td>3. Required</td>
<td>TLS is negotiated from the Email Security appliance interface to MTA(s) for the domain. No attempt is made to verify the domain’s certificate. If the negotiation fails, no email is sent through the connection. If the negotiation succeeds, the mail is delivered via an encrypted session.</td>
</tr>
</tbody>
</table>
| 4. Preferred (Verify) | TLS is negotiated from the Email Security appliance to the MTA(s) for the domain. The appliance attempts to verify the domain’s certificate. Three outcomes are possible:  
  - TLS is negotiated and the certificate is verified. The mail is delivered via an encrypted session.  
  - TLS is negotiated, but the certificate is not verified. The mail is delivered via an encrypted session.  
  - No TLS connection is made and, subsequently the certificate is not verified. The email message is delivered in plain text. |
| 5. Required (Verify) | TLS is negotiated from the appliance to the MTA(s) for the domain. Verification of the domain certificate is required. The following outcomes are possible:  
  - A TLS connection is negotiated and the certificate is verified. The email message is delivered via an encrypted session.  
  - A TLS connection is negotiated, but the certificate is not verified by a trusted Certificate Authority (CA). The mail is not delivered.  
  - A TLS connection is not negotiated. The mail is not delivered. |
### TLS Setting

<table>
<thead>
<tr>
<th>TLS Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Required - Verify Hosted Domains</td>
<td>The difference between TLS Required - Verify and TLS Required - Verify Hosted Domain options lays in identity verification process. The way how the presented identity is processed and what type of reference identifiers are allowed to be used make a difference about a final result. The presented identity is first derived from subjectAltName extension of type dNSName. If there is no match between the dNSName and one of accepted reference identities (REF-ID), the verification fails no matter if CN exist in subject field and could pass further identity verification. The CN derived from subject field is validated only when the certificate does not contain any of subjectAltName extension of type dNSName.</td>
</tr>
</tbody>
</table>

If there is no specific entry for a given recipient domain in the good neighbor table, or if there is a specific entry but there is no specific TLS setting for the entry, then the behavior is whatever is set using the Destination Controls page or the destconfig -> default subcommand (“No,” “Preferred,” “Required,” “Preferred (Verify),” or “Required (Verify)”).

#### Related Topics
- Sending Alerts When a Required TLS Connection Fails, on page 12
- Logging, on page 13
- Managing Lists of Certificate Authorities, on page 13

### Sending Alerts When a Required TLS Connection Fails

You can specify whether the Email Security appliance sends an alert if the TLS negotiation fails when delivering messages to a domain that requires a TLS connection. The alert message contains name of the destination domain for the failed TLS negotiation. The Email Security appliance sends the alert message to all recipients set to receive Warning severity level alerts for System alert types. You can manage alert recipients via the System Administration > Alerts page in the GUI (or via the alertconfig command in the CLI).

#### Related Topics
- Enabling TLS Connection Alerts, on page 12

### Enabling TLS Connection Alerts

1. **Step 1** Navigate to the Mail Policies Destination Controls page.
2. **Step 2** Click **Edit Global Settings**.
3. **Step 3** Click **Enable** for “Send an alert when a required TLS connection fails.”
   This is a global setting, not a per-domain setting. For information on the messages that the appliance attempted to deliver, use the Monitor > Message Tracking page or the mail logs.
4. **Step 4** Submit and commit your changes.
What to do next
You can also configure this in the command-line interface using the destconfig -> setup command to enable TLS connection alerts using the CLI

Logging
The Email Security appliance will note in the mail logs instances when TLS is required for a domain but could not be used. Information on why the TLS connection could not be used will be included. The mail logs will be updated when any of the following conditions are met:

- The remote MTA does not support ESMTP (for example, it did not understand the EHLO command from the Email Security appliance).
- The remote MTA supports ESMTP but “STARTTLS” was not in the list of extensions it advertised in its EHLO response.
- The remote MTA advertised the “STARTTLS” extension but responded with an error when the Email Security appliance sent the STARTTLS command.

Managing Lists of Certificate Authorities
The appliance uses stored trusted certificate authorities that it uses to verify a certificate from a remote domain to establish the domain’s credentials. You can configure the appliance to use the following trusted certificate authorities:

- **Pre-installed list.** The appliance has a pre-installed list of trusted certificate authorities. This is called the system list.
- **User-defined list.** You can customize a list of trusted certificate authorities and then import the list onto the appliance.

You can use either the system list or the customized list, and you can also use both lists to verify certificate from a remote domain.

Manage the lists using the Network > Certificates > Edit Certificate Authorities page in the GUI or the certconfig > certauthority command in the CLI.

On the Network > Certificates > Edit Certificate Authorities page, you can perform the following tasks:

- **View the system list (pre-installed) of certificate authorities.** For more information, see Viewing the Pre-Installed list of Certificate Authorities, on page 14.
- **Choose whether or not to use the system list.** You can enable or disable the system list. For more information, see Disabling the System Certificate Authority List, on page 14.
- **Choose whether or not to use a custom certificate authority list.** You can enable the appliance to use a custom list and then import the list from a text file. For more information, see Importing a Custom Certificate Authority List, on page 14.
- **Export the list of certificate authorities to a file.** You can export either the system or customized list of certificate authorities to a text file. For more information, see Exporting a Certificate Authorities List, on page 14.

Related Topics
- Viewing the Pre-Installed list of Certificate Authorities, on page 14
- Disabling the System Certificate Authority List, on page 14
Viewing the Pre-Installed list of Certificate Authorities

- Step 1: Navigate to the Network > Certificates page.
- Step 2: Click Edit Settings in the Certificate Authorities section.
- Step 3: Click View System Certificate Authorities.

Disabling the System Certificate Authority List

The pre-installed system certificate authorities list cannot be removed from the appliance, but you can enable or disable it. You might want to disable it to allow the appliance to only use your custom list to verify certificates from remote hosts.

- Step 1: Navigate to the Network > Certificates page.
- Step 2: Click Edit Settings in the Certificate Authorities section.
- Step 3: Click Disable for the System List.
- Step 4: Submit and commit your changes.

Importing a Custom Certificate Authority List

You can create a custom list of trusted certificate authorities and import it onto the appliance. The file must be in the PEM format and include certificates for the certificate authorities that you want the appliance to trust.

- Step 1: Navigate to the Network > Certificates page.
- Step 2: Click Edit Settings in the Certificate Authorities section.
- Step 3: Click Enable for the Custom List.
- Step 4: Enter the full path to the custom list on a local or network machine.
- Step 5: Submit and commit your changes.

Exporting a Certificate Authorities List

If you want to use only a subset of the trusted certificate authorities in the system or edit an existing custom list, you can export the list to a .txt file and edit it to add or remove certificate authorities. After you have finished editing the list, import the file back onto the appliance as a custom list.
Step 1: Navigate to the Network > Certificates page.
Step 2: Click Edit Settings in the Certificate Authorities section.
Step 3: Click Export List.
AsyncOS displays the Export Certificate Authority List page.
Step 4: Select the list you want to export.
Step 5: Enter a filename for the list.
Step 6: Click Export.
AsyncOS displays a dialog box asking if you want to open or save the list as a .txt file.

Enabling a Certificate for HTTPS

You can enable a certificate for HTTPS services on an IP interface using either the Network > IP Interfaces page in the GUI or the interfaceconfig command in the CLI.

Step 1: Navigate to the Network > IP Interfaces page.
Step 2: Select the interface you want to enable the HTTPS service.
Step 3: Under Appliance Management, check the HTTPS check box and enter the port number.
Step 4: Submit and commit your changes.

What to do next

Note
The demonstration certificate that is pre-installed on the appliance. You may enable HTTPS services with the demonstration certificate for testing purposes, but it is not secure and is not recommended for general use.
You can enable HTTPS services using the System Setup Wizard in the GUI. For more information, see Setup and Installation.
Enabling a Certificate for HTTPS