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 8
- Enabling TLS and Certificate Verification on Delivery, on page 11
- DNS-based Authentication of Named Entities, on page 14
- Mail Transfer Agent Strict Transport Security, on page 16
- Enabling TLS for Delivery with DANE and MTA-STS Support, on page 20
- Managing Lists of Certificate Authorities, on page 22
- Enabling a Certificate for HTTPS, on page 25

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 email gateway. 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

	Do This	More Info
Step 1	Obtain an X.509 certificate and private key from a recognized certificate authority.	Working with Certificates, on page 2
Step 2	Install the certificate on the email gateway	Install a certificate by either: • Creating a Self-Signed Certificate , on page 4 • Importing a Certificate , on page 6
Step 3	Enable TLS for receiving messages, delivering messages, or both	• Enabling TLS on a Listener's HAT, on page 8 • Enabling TLS and Certificate Verification on Delivery, on page 11
Step 4	(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.	Managing Lists of Certificate Authorities, on page 22
Step 5	(Optional) Configure the email gateway to send an alert when it's unable to deliver messages to a domain that requires a TLS connection.	Sending Alerts When a Required TLS Connection Fails, on page 13

Working with Certificates

To use TLS, the email gateway 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 email gateway 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 gateway and the other machine, for example because that machine is not in your domain. Your corporate security department may have other requirements.

	Do This	More Info
Step 1	If you are deploying in a cluster, follow instructions.	Certificates and Centralized Management, on page 4
Step 2	Generate a self-signed certificate and Certificate Signing Request (CSR).	Creating a Self-Signed Certificate , on page 4
Step 3	Send the generated certificate to a recognized Certificate Authority for signing.	About Sending a Certificate Signing Request (CSR) to a Certificate Authority , on page 5
Step 4	Upload the signed certificate.	Uploading a Certificate Signed by a Certificate Authority , on page 6
Step 5	Ensure that the certificate authority that signed the certificate is on the list of trusted authorities.	Managing Lists of Certificate Authorities, on page 22
Step 6	If applicable, use an intermediate certificate.	Intermediate Certificates, on page 4

Deploying Self-Signed Certificates

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

	Do This	More Info
Step 1	If you are deploying in a cluster, follow instructions.	Certificates and Centralized Management, on page 4
Step 2	Generate a self-signed certificate from the email gateway.	Creating a Self-Signed Certificate , on page 4
Step 3	Export the self-signed certificate.	Exporting a Certificate , on page 7
Step 4	Import the self-signed certificate to the machine with which the email gateway will communicate.	See the documentation for the other machine.
Step 5	Generate and export a self-signed certificate from the other machine.	See the documentation for the other machine.

	Do This	More Info
Step 6	Import the self-signed certificate from the other machine into the email gateway.	Importing a Certificate , on page 6 or See the chapter in this guide for configuring communication with that machine. For example, to configure secure communications with a Cisco Secure Malware Analytics (Threat Grid) Appliance, see instructions for configuring Advanced settings in Configuring an On-Premises File Analysis Server.

Certificates and Centralized Management

A certificate usually uses the local machine's hostname for the certificate's common name. If your email gateways 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 email gateway 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 email gateway 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 email gateway and a Cisco Secure Malware Analytics (Threat Grid) appliance.

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

Procedure

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:

Common Name	The fully qualified domain name.
Organization	The exact legal name of the organization.
Organizational Unit	Section of the organization.
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.
Signature Algorithm	The signature algorithm to be used in the certificate.
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. **Note** This option is not available if you have chosen the *ecdsa-with-sha256* Signature Algorithm.

Step 5 Click **Next**.

Step 6 Select the **FQDN Validation** check box to allow the email gateway to check whether the 'Common Name present in the certificate is in the FQDN format.

Step 7 Enter a name for the certificate. By default, AsyncOS assigns the common name previously entered.

Step 8 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 9 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 gateway 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 certconfig 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 email gateway.

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.

Procedure

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 email gateway. (Tools for doing this are included with OpenSSL, free software from http://www.openssl.org.)

Step 2 Upload the signed certificate to the email gateway:

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 email gateway.

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 6 .

Procedure

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 Select the **FQDN Validation** check box to allow the email gateway to check whether the 'Common Name,' 'SAN: DNS Name' fields, or both present in the certificate, are in the FQDN format.

Step 8 Enter a name for the certificate.

AsyncOS assigns the common name by default.

Step 9 Submit and commit your changes.

Note If you import a non-compliant X.509 certificate, a warning message is displayed. Despite the warning, the certificate will still be uploaded.

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.

Procedure

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

Step 2 Click **Export Certificate**.

Step 3 Select the certificate you want to export.

Step 4 Enter the file name for the certificate.

Step 5 Enter and confirm the passphrase for the certificate file.

Step 6 Click **Export**.

Step 7 Save the file to a local or network machine.

Step 8 You can export additional certificates or click **Cancel** to return to the Network > Certificates page.

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

TLS Setting	Meaning
`1. No`	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 email gateway.
`2. Preferred`	TLS is allowed for incoming connections to the listener from MTAs.
`3. Required`	TLS is allowed for incoming connections to the listener from MTAs, and until a `STARTTLS` command is received, the email gateway responds with an error message to every command other than `NOOP , EHLO , or QUIT` . 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 email gateway before it is sent, thereby preventing it from be transmitted in the clear.

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 9
- Assigning a Certificate to a Public or Private Listener for TLS Connections Using the CLI, on page 9
- Logging, on page 14
- GUI Example: Changing the TLS setting for Listeners HAT, on page 9
- CLI Example: Changing the TLS Setting for Listeners HAT, on page 10

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

Procedure

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

Procedure

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

Procedure

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

Procedure

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 email gateway. 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.

```
Name: Inboundmail

Type: Public

Interface: PublicNet (192.168.2.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 1000 (TCP Queue: 50)

Domain map: disabled

TLS: Required
```

Step 5 Issue the `commit` command to enable the change

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 four 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.
- [Optional - Only if FQDN validation enabled in SSL Configuration settings]: Check whether the 'Common Name,' 'SAN: DNS Name' fields, or both present in the server certificate, are in the FQDN format.
- [Optional - Only if X 509 validation enabled in SSL Configuration settings]: Check for the signature algorithm of the server certificate.
- [Optional - Only if X 509 validation enabled in SSL Configuration settings]: Check whether the server name is present in the 'Common Name,' or 'SAN: DNS Name' fields in the server certificate.
- [Optional - Only if X 509 validation enabled in SSL Configuration settings]: Check the server certificate version.

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 gateway 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 email gateway 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

TLS Setting	Meaning
Default	The default TLS setting set using the Destination Controls page or the `destconfig -> default` 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?”
1. No	TLS is not negotiated for outgoing connections from the interface to the MTA for the domain.
2. Preferred	TLS is negotiated from the email gateway interface to the MTA(s) for the domain. However, if the TLS negotiation fails (prior to receiving a 220 response), the SMTP transaction does not fall back to clear text. No attempt is made to verify if the certificate originates from a trusted certificate authority. If an error occurs and the TLS negotiation fails after the 220 response is received, the SMTP transaction will continue "in the clear" (not encrypted).
3. Required	TLS is negotiated from the email gateway 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.
4. Preferred (Verify)	TLS is negotiated from the email gateway to the MTA(s) for the domain. The email gateway 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 email gateway 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 Cerfificate Authority (CA). The mail is not delivered. • A TLS connection is not negotiated. The mail is not delivered.

TLS Setting	Meaning
6. Required - Verify Hosted Domains	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`.

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 13
- Logging, on page 14
- Managing Lists of Certificate Authorities, on page 22

Sending Alerts When a Required TLS Connection Fails

You can specify whether the email gateway 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 gateway 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 13

Enabling TLS Connection Alerts

Procedure

Step 1 Navigate to the Mail Policies Destination Controls page.

Step 2 Click **Edit Global Settings**.

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 email gateway attempted to deliver, use the Monitor > Message Tracking page or the mail logs.

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 gateway 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 gateway).
- 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 gateway sent the STARTTLS command.

DNS-based Authentication of Named Entities

- Overview of SMTP DNS-based Authentication of Named Entities, on page 14
- Enabling TLS for Delivery with DANE and MTA-STS Support, on page 20
- Sending Alerts When DANE Fails, on page 22

Overview of SMTP DNS-based Authentication of Named Entities

TLS connections that are authenticated using certificates can become vulnerable to a security breach in any one of the following ways:

- A trusted Certificate Authority (CA) can issue certificates to any domain name.
- An attacker can use a man-in-the-middle (MITM) attack to downgrade a TLS connection to plain text communication.
- If DNSSEC is not configured on the DNS server, an attacker can forge a DNS response with fake DNS MX records and redirect messages to an insecure server which can lead to DNS cache poisoning attacks.
- Self-signed certificates or certificates issued by a private Certificate Authority (CA) can be used when a receiving Mail Transfer Agent (MTA) is not configured with a list of trusted Certificate Authority.

The SMTP DNS-based Authentication of Named Entities (DANE) protocol validates your X.509 certificates with DNS names using a Domain Name System Security (DNSSEC) extension configured on your DNS server and a DNS resource record, also known as a TLSA record.

The TLSA record is added in the cerificate that contains details about either the Certificate Authority (CA), the end entity certificate, or the trust anchor used for the DNS name described in *RFC 6698*. For more information, see Creating TLSA Record, on page 16. The Domain Name System Security (DNSSEC) extensions provide added security on the DNS by addressing vulnerabilities in DNS security. DNSSEC using cryptographic keys and digital signatures ensures that the lookup data is correct and connects to legitimate servers.

The following are the benefits of using SMTP DANE for outgoing TLS connections:

- Provides secure delivery of messages by preventing Man-in-the-Middle (MITM) downgrade attacks, eavesdropping and DNS cache poisoning attacks.
- Provides authenticity of TLS certificates and DNS information, when secured by DNSSEC.

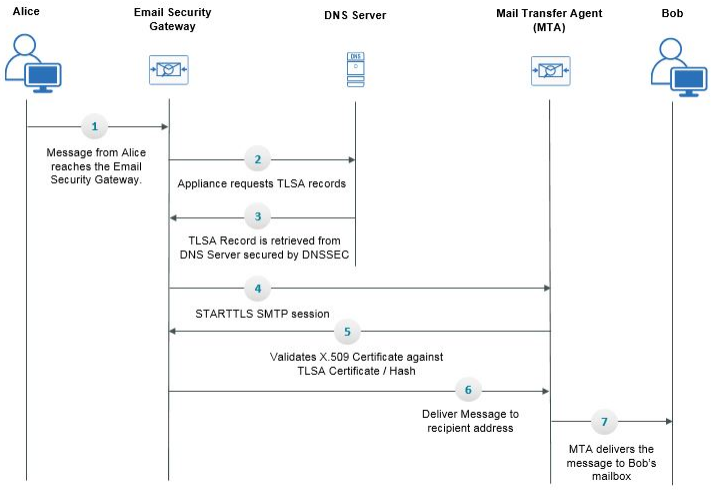
Related Topics

- SMTP DANE Workflow, on page 15
- Creating TLSA Record, on page 16
- Enabling TLS for Delivery with DANE and MTA-STS Support, on page 20
- Sending Alerts When DANE Fails, on page 22

SMTP DANE Workflow

The following figure describes flow of messages using outgoing TLS connection with DANE support:

Figure 1: Message Delivery Using TLS with DANE Support



1. Sender (Alice) sends a message to a recipient (Bob) outside the organization.
2. The message reaches the email gateway.
3. The email gateway requests for a DNS resource record, also known as a TLSA record of the DNS from the DNS server.
4. The certificates and a TLSA record is retrieved from the DNS server, secured by DNSSEC.
5. The email gateway establishes a STARTTLS SMTP session to the recipient's address.
6. The X.509 certificates are validated against the complete TLSA record or hash value of the TLSA record, of the recipient's address. After successful validation, the message is delivered to the recipient's Mail Transfer Agent (MTA). If certificate verification fails, the message is delivered at a later time or the message is bounced.

7. The MTA delivers the message to recipient's mailbox.

Creating TLSA Record

You can create a TLSA record of your preferred certificate authority (CA) on the DNS record signed with DNSSEC. Below is a sample TLSA record for a Fully Qualified Domain Name (FQDN) `www.example.com`:

```
_443._tcp.www.example.com. IN TLSA (0 0 1
91751cee0a1ab8414400238a761411daa29643ab4b8243e9a91649e25be53ada)
```

The above example TLSA record has the following fields that are encrypted:

- **Certificate Usage**: Specifies the type of certificate.
  - In the given sample, the first '0' digit specifies the CA certificate that must be matched to the PKIX certification path, as described in RFC 6698.
  - If it is '1', it specifies the end entity certificate that must be matched to the end entity certificate given by the server in TLS.
  - If it is '2', it specifies a certificate that must be used as a trust anchor while validating the end entity certificate given by the server in TLS.
  - If it is '3', it specifies a certificate that must match the end entity certificate given by the server in TLS.
- **Selector Field**: Specifies the part of TLS certificate that is matched with the association data.
  - In the given sample, the second '0' specifies that the full certificate must be matched.
  - If it is '1', it specifies that only the 'SubjectPublicKeyInfo' field must be matched.
- **Matching Type**: Specifies the type of HASH value that is used.
  - In the given sample, the third '1' specifies the SHA-256 hash of the selected content.
  - If it is '0', it specifies the exact match on the selected content.
  - If it is '2' it specifies the SHA-512 hash of the selected content.

Mail Transfer Agent Strict Transport Security

- Overview of MTA-STS, on page 17
- How MTA-STS works, on page 17
- Enabling TLS for Delivery with DANE and MTA-STS Support, on page 20
- DANE and MTA-STS Scenarios, on page 18
- Viewing Logs, on page 19

Overview of MTA-STS

Mail Transfer Agent Strict Transport Security (MTA-STS) is a protocol that enforces the use of secure connections (TLS - Transport Layer Security) when email servers communicate with each other. This helps prevent man-in-the-middle attacks and eavesdropping by ensuring emails are sent over secure, encrypted channels.

MTA-STS enables Secure Email Gateway to determine and act on the TLS policy of a peer Mail Transfer Agent (MTA) for outbound emails, ensuring secure email transmission. Domain owners can publish policies requiring TLS for email transmission, and MTA-STS ensures these policies are followed.

When enabled, the Secure Email Gateway checks the MTA-STS policy for destination domains. It then initiates the MTA-STS process to fetch, validate, and apply the defined policy, ensuring that the connection to the receiving MTA is secure over TLS.

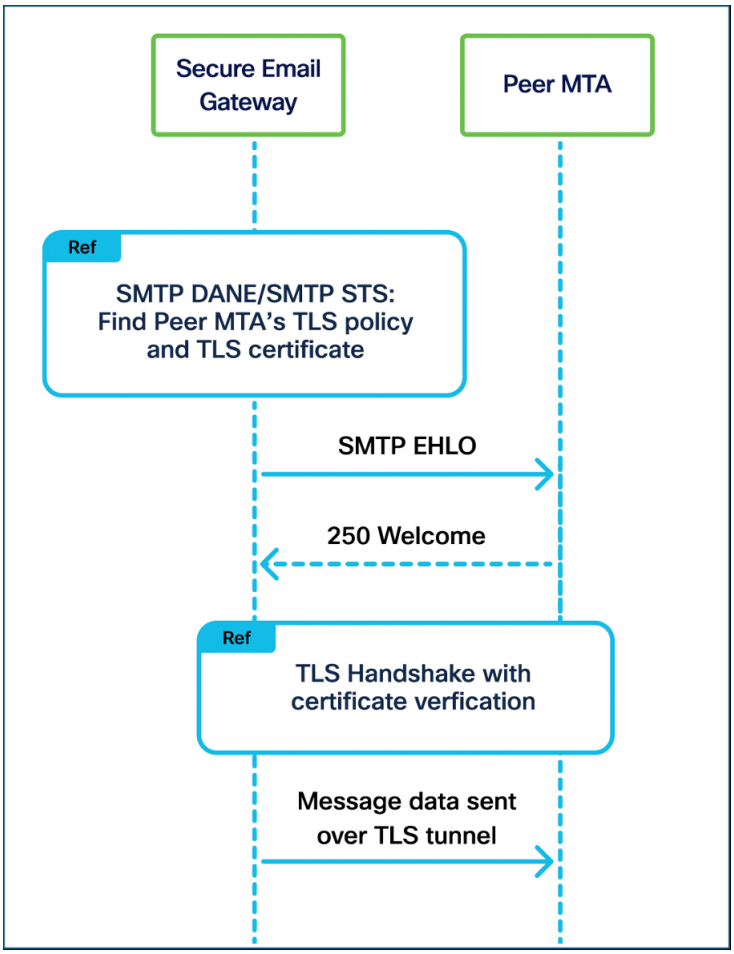
By using MTA-STS, the Secure Email Gateway ensures that all outgoing emails are transmitted securely, according to the TLS policies defined by the receiving MTAs.

How MTA-STS works

The email domain owner publishes an MTA-STS policy via DNS and HTTPS (Hypertext Transfer Protocol Secure). This policy specifies that the domain requires secure connections for email transfer. When a Secure Email Gateway (sending MTA) wants to send an email to another domain, it first checks for the recipient domain's MTA-STS policy. If the policy requires TLS, the sending email server will attempt to establish a secure, encrypted connection to the receiving email server (receiving MTA). If a secure connection cannot be established (e.g., if the TLS certificate is invalid or the connection is downgraded to an insecure protocol), the email will not be delivered, and the sender will be notified of the failure.

MTA-STS policies enable mail service providers to declare their ability to receive secure SMTP connections using TLS. They also specify whether sending SMTP servers should refuse to deliver emails to MX hosts that do not offer TLS with a trusted server certificate.

MTA-STS uses DNS TXT records for policy discovery. It fetches the MTA-STS policy from an HTTPS host. During the TLS handshake, which is initiated to fetch a new or updated policy from the Policy Host, the HTTPS server must present a valid X.509 certificate for the "mta-sts" DNS-ID.



DANE and MTA-STS Scenarios

DANE and MTA-STS can coexist for the same destination domain, but DANE takes precedence. If a DANE failure occurs, the email is dropped, even if an MTA-STS policy exists. DANE and MTA-STS together enhance email security. The domain owner publishes TLSA records via DNSSEC to authenticate TLS certificates (DANE) and an MTA-STS policy via DNS and HTTPS to mandate TLS-encrypted transmission.

The following scenarios describe various outcomes of DANE and MTA-STS configuration.

When sending an email, the Mail Transfer Agent (MTA) retrieves both DANE TLSA records and the MTA-STS policy. The MTA first verifies the TLS certificates' authenticity (DANE) and attempts to establish a TLS-encrypted connection. DANE ensures the certificates match the TLSA records, confirming authenticity. If a secure connection cannot be established with DANE, because of insecure or non-existent DNS records, ESA validates the MTA-STS policy for the domain. If a secure connection cannot be established with MTA-STS, the email isn't sent, preventing downgrade attacks. After these checks, the email is securely transmitted, ensuring robust and trusted communication.

DANE is set as None and MTA STS is enabled:

DANE	MTA-STS	Email Delivery
N/A	Pass	Yes, TLS Required

N/A	Fail – DNS MX Record Does Not Match Policy MX Record	No
N/A	Fail – Receiving MTA Certificate validation failure	No
N/A	Fail – Any Other Reason	Yes – TLS mode configured for the domain.

DANE is set as Opportunistic and MTA STS is enabled:

DANE	MTA-STS	Email Delivery
Pass	N/A	Yes, TLS Required
Fail - Destination Domain Cert Verification Fails OR BOGUS MX Record or A Record or TLSA	N/A	No MX/TLSA records are insecure or TLSA record is NXDomain
Fail - Any Other Reason	Pass	Yes, TLS Required
Fail - Any Other Reason	Fail - DNS MX Record Does Not Match Policy MX Record	No
Fail - Any Other Reason	Fail - Any Other Reason	Yes - TLS mode configured for the domain

DANE is set as Mandatory and MTA STS is enabled:

DANE	MTA-STS	Email Delivery
Pass	N/A	Yes, TLS Required
Fail	N/A	No

Configuration for delivery domains hosted on a different domain:

For delivery domains hosted on a different domain, use the following configuration. For example, if your domain is served by the office.com MX server, use:

- **TLS Support**: TLS – Required Verify Hosted Domains
- **MTA-STS**: ON

Viewing Logs

The MTA-STS information is posted to the Mail Logs. Most information is at the Info or Debug level.

Examples of MTA-STS log entries.

In this example, the log shows that the STS Record is invalid because of multiple entries.

```
Info: MTA-STS record Error
(Invalid MTA-STS TXT records found) for domain(test-sts.net)
```

In this example, the log shows that the STS Record is invalid because of length and syntax.

```
Info: MTA-STS record Error
(Invalid MTA-STS TXT records found) for domain(test-sts.net)
```

In this example, the log shows that the MTA-STS records are not found for Domain.

```
Debug: DNS query: Q(_mta-sts.test-sts.net, 'TXT')
Info: MTA-STS record Error(Invalid MTA-STS TXT records found) for domain(test-sts.net)
```

In this example, the log shows that the MTA-STS valid records are found.

```
Info: Successfully fetched MTA-STS TXT record for domain(test-sts.net)
```

In this example, the log shows failure to get STS Policy.

```
Info: Failure encountered while fetching MTA-STS policy for the domain(test-sts.net)
```

In this example, the log shows that the MTA-STS policy application success.

```
Info: MTA-STS policy for the domain (test-sts.net) Successful.
```

In this example, the log shows that the MTA-STS policy application start.

```
Info: Applying MTA-STS policy for the domain (test-sts.net).
Info: Enforcing TLS for MTA-STS Policy
```

In this example, the log shows that the MTA-STS policy application failure - Certificate validation failed.

```
Info: Applying MTA-STS policy for the domain (test-sts.net).
Info: Applying MTA-STS policy for the domain (test-sts.net) Failed.
```

In this example, the log shows that the MTA-STS policy application failure when receiving MTA does not support TLS.

```
Info: New SMTP DCID 498 interface 10.10.2.83 address 10.10.2.7 port 25
Info: DCID 498 STARTTLS command not supported
Info: DCID 498 TLS failed: TLS required, STARTTLS unavailable, destination is TLS disabled
Info: DCID 498 TLS was required but could not be successfully negotiated
Info: Connection for the domain (auto-sts.net) failed when MTA-STS Policy was enforced.
```

In this example, the log shows repetitive policy fetch failures.

```
Info: Failures encountered while fetching STS policy, successive request is skipped for a
day.
```

In this example, the log shows MTA-STS skipped due to DANE Enforce.

```
Info: Successfully fetched MTA-STS TXT record for domain(pert-sts.net)
Debug: Fetch MTA-STS policy for the domain(pert-sts.net)
Info: Successfully fetched MTA-STS policy for the domain(pert-sts.net)
Debug: MTA-STS Cache entry added for the domain (pert-sts.net).
Info: MID 188 matched all recipients for per-recipient policy DEFAULT in the outbound table
Info: MID 188 queued for delivery
Info: MTA-STS policy will be skipped for the (pert-sts.net) due to DANE being enforced.
```

Enabling TLS for Delivery with DANE and MTA-STS Support

Before you begin

- Ensure that the envelope sender and the TLSA resource record is DNSSEC verified.

- Ensure that you enable TLS to configure DANE on your email gateway. For more information, see Enabling TLS and Certificate Verification on Delivery, on page 11.

Procedure

Step 1 Go to **Mail Policies** > **Destination Controls** page.

Step 2 Click **Add Destination Controls** or modify an existing entry.

Step 3 From the **TLS Support** field, you must select any option other than **None** to enable DANE and MTA-STS support on your email gateway.

Step 4 From the **DANE Support** field, you can specify the following settings for DANE for a given TLS connection.

DANE Setting	Description
Default	The default DANE setting set using the Destination Controls page is used for outgoing TLS connections from the listener to the MTA for the domain. The "Default" DANE setting is inherited from the default TLS settings in Destination Controls. You can override this setting to the custom Destination Control entry.
None	Select "None", if you do not want DANE to be used for negotiating outgoing connections from the interface to the MTA for the domain.
Opportunistic	If you select "Opportunistic" and the remote host does not support DANE, opportunistic TLS is used for encrypting SMTP conversations. If you select "Opportunistic" and the remote host supports DANE, it becomes the preferred mode for encrypting SMTP conversations.
Mandatory	If you select "Mandatory" and the remote host does not support DANE, no connection is established to the destination host. If you select "Mandatory" and the remote host supports DANE, it becomes the preferred mode for encrypting SMTP conversations.

Step 5 Select the MTA-STS Support that you want.

- **Default**: Use the MTA-STS setting used in Default Destination Control
- **No**: Disable MTA-STS support
- **Yes**: Enable MTA-STS support

Note Secure Email Gateway secures TLS connections using MTA-STS to fetch, validate, and apply the receiving MTA's policy for the destination domain. If DANE is enabled, the use of MTA-STS also depends on DANE settings and its success. For more information, see DANE and MTA-STS Scenarios, on page 18.

Note It is recommended not to enable MTA-STS in the default destination control as it may affect overall performance.

Step 6 Select address tagging preference for Bounce Verification. For more information, see Bounce Verification

Step 7 Select the Bounce Profile to be used by the destination control policy.

Step 8 Submit and Commit your changes.

Sending Alerts When DANE Fails

You can specify whether the email gateway sends an alert if the DANE verification fails to all MX hosts when delivering messages to a domain that requires a TLS connection with DANE support. The email gateway sends the alert message to all recipients set to receive Warning severity level alerts for System alert types.

Enabling DANE Alerts

Procedure

Step 1 Go to **System Administration > Alerts** page.

Step 2 Select the alert recipient you want to enable the alert.

Step 3 Select the **Message Delivery** check box corresponding to the alert type.

Step 4 Submit and commit your changes.

Managing Lists of Certificate Authorities

The email gateway 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 email gateway to use the following trusted certificate authorities:

- **Pre-installed list.** The email gateway 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 email gateway.

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 23.
- **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 23.

- **Choose whether or not to use a custom certificate authority list.** You can enable the email gateway 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 24.
- **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 24.

Related Topics

- Viewing the Pre-Installed list of Certificate Authorities, on page 23
- Disabling the System Certificate Authority List, on page 23
- Importing a Custom Certificate Authority List, on page 24
- Exporting a Certificate Authorities List, on page 24
- Certificate Updates, on page 24
- Managing Trusted Root Certificates, on page 25

Viewing the Pre-Installed list of Certificate Authorities

Procedure

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 email gateway, but you can enable or disable it. You might want to disable it to allow the email gateway to only use your custom list to verify certificates from remote hosts.

Procedure

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 of list trusted certificate authorities and import it onto the email gateway. The file must be in the PEM format and include certificates for the certificate authorities that you want the email gateway to trust.

Procedure

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 Select the **FQDN Validation** check box to allow the email gateway to check whether the 'Common Name,' 'SAN: DNS Name' fields, or both present in the certificate, are in the FQDN format.

Step 6 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 email gateway as a custom list.

Procedure

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 want to open or save the list as a .txt file.

Certificate Updates

The Updates section under Certificate Lists displays the version and last-updated information for the Cisco trusted-root-certificate (system CA certificate) bundle on the email gateway. The bundle is updated periodically.

Click **Update Now** on the Certificates page to update the existing Cisco trusted-root-certificate (system CA certificate) bundle to the latest available version.

Managing Trusted Root Certificates

You can view the count and details of the following certificates in your email gateway:

- Custom Trusted Root (custom CA) certificates
- Cisco Trusted Root (system CA) certificates

Procedure

Step 1 Navigate to **Network** > **Certificates** page.

Step 2 Click **Manage Trusted Root Certificates** under the Certificate Lists section to view the details of the custom or system CA certificates.

Step 3 Click the required certificate link (for example, 'Admin-Root-CA') under the Cisco Trusted Root Certificate List section to view the certificate details.

Step 4 [Optional] Click the **Download Certificate** link below the certificate (for example, 'Admin-Root-CA') details to download the certificate.

Note You can also click the required certificate link under the Custom Trusted Root Certificates List section to view or download the certificate details.

Note You can also delete a Custom Trusted Root (custom CA) certificate, if required.

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.

Procedure

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 email gateway. 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.