Forcepoint DLP

10.4

Configuring Postfix to Enable Protector TLS Support for Explicit MTA

Revision A

Contents

Configuring Postfix to Enable Protector TLS Support for Explicit MTA

The Forcepoint DLP protector includes a Postfix release compiled with Transport Layer Security (TLS) support. Configure Postfix to enable TLS support. Although TLS is not officially supported, Postfix is available to allow for individual customer configurations.

This document provides a simple example TLS configuration for use as a test case and as a reference for future deployments.

The sample configuration is stored on the protector in the `/etc/postfix/main.cf` file. Every time the file is modified, reload Postfix using one of the following operations:

- With the “postfix reload” command
- With the “postfix-reconf” command
- By clicking Deploy in the Data Security module of the Forcepoint Security Manager

In the protector, Postfix serves as a store-and-forward proxy. This means that it functions as both a server (getting messages from the previous hop) and a client (delivering the non-blocked messages to the next hop).

Because previous and next hops may have different TLS requirements, settings for server and client modes are configured differently.

Client side configuration

Before you begin

This examples assumes that at least some *next* hops require TLS.

Steps

1) On the protector, open the `/etc/postfix/tls_policy` file in a text editor.
   If the file does not exist, create it.

2) Add this line to the file:
`next.hop.domain encrypt`

3) Compile the file using the following command: `postmap hash:/etc/postfix/tls_policy`

4) Open the `/etc/postfix/main.cf` file in a text editor.

5) Add the following lines to the file:

```
relayhost = next.hop.domain ## maintained by management !!!
## certs files
smtp_tls_cert_file=/etc/pki/tls/certs/mydomain.com.cert
smtp_tls_key_file=/etc/pki/tls/private/mydomain.com.key
smtp_tls_CAfile=/etc/pki/tls/cert.pem
## policy map
smtp_tls_policy_maps = hash:/etc/postfix/tls_policy
```

6) Run the following command: `postfix reload`

Note

- The policy map is required for better granularity. Specifically, Postfix also traverses messages internally (for example, to the pamad plugin), and these transfers should be plain.
- Certificates are very customer-dependent. For this example, 3 PEM files were created:
  - mydomain.com.cert

    This is the client certificate, not necessarily required by servers.
  - mydomain.com.key

    This is the client's private key, sometimes included in client certificate.
  - default cert.pem

    This is the Certificate Authority (CA) certificate.

The following command was used to create the certificate files in this example: `genkey -days 265 $(hostname)`

The **genkey** utility is a part of crypto utilities package, which is not installed by default and should be installed manually or using yum. To install the package with yum, use the following command: `yum install crypto-utils`

If package installation is not an option, it is possible to create certificates using the **openssl** command: `openssl req -new -nodes -keyout myhost.com.key -out req.pem`

In this case, sign "req.pem" with a CA and get in return the "myhost.com.cert" and "cert.pem" files.

Server side configuration

Before you begin

The example assumes that at least some *previous* hops require TLS:

Steps

1) Open the `/etc/postfix/main.cf` file in a text editor.

2) Add the following lines to the file:

```
smtpd_tls_cert_file = /opt/websense/PolicyEngine/allcerts.cer
smtpd_tls_key_file = $smtpd_tls_cert_file
smtpd_tls_security_level = may

smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtpd_tls_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtpd_tls_mandatory_ciphers = high
smtpd_tls_ciphers = high
tls_high_cipherlist = ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
tls_preempt_cipherlist = no

smtpd_tls_dh1024_param_file = /etc/postfix/dhparam.pem
```

3) Run the following command:

```
postfix reload && openssl dhparam -out /etc/postfix/dhparam.pem 2048
```

Note

- This sample uses the protector's certificates. Some clients (previous hops) may require this certificate to be trusted by a known CA.
- Optionally, a private key is included in the certificate file.
- In this sample, the security level is set to enable TLS, but not make it mandatory. This can be changed.

For further details, see: http://www.postfix.org/TLS_README.html and http://www.postfix.org/postconf.5.html.

Published 24 September 2025