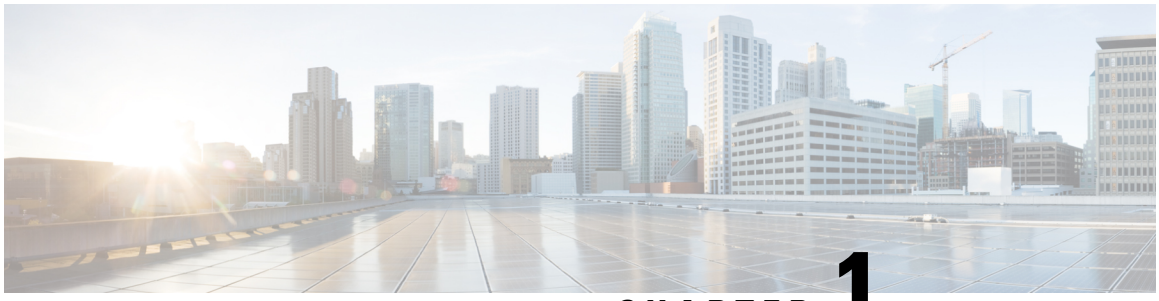
Cisco Secure Firewall ASA Unified Communications Guide

Last Modified: 2022-05-31

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CHAPTER 1

Information About the Secure Firewall ASA in Cisco Unified Communications

This chapter describes how to configure the Secure Firewall ASA for Cisco Unified Communications Proxy features.

This chapter includes the following sections:

Information About the ASA in Cisco Unified Communications

This section describes the Cisco UC Proxy features. The purpose of a proxy is to terminate and reoriginate connections between a client and server. The proxy delivers a range of security functions such as traffic inspection, protocol conformance, and policy control to ensure security for the internal network. An increasingly popular function of a proxy is to terminate encrypted connections in order to apply security policies while maintaining confidentiality of connections. The ASA is a strategic platform to provide proxy functions for unified communications deployments.

The Cisco UC Proxy includes the following solutions:

TLS Proxy: Decryption and inspection of Cisco Unified Communications encrypted signaling

End-to-end encryption often leaves network security appliances “blind” to media and signaling traffic, which can compromise access control and threat prevention security functions. This lack of visibility can result in a lack of interoperability between the firewall functions and the encrypted voice, leaving businesses unable to satisfy both of their key security requirements.

The ASA is able to intercept and decrypt encrypted signaling from Cisco encrypted endpoints to the Cisco Unified Communications Manager (Cisco UCM), and apply the required threat protection and access control. It can also ensure confidentiality by re-encrypting the traffic onto the Cisco UCM servers.

Typically, the ASA TLS Proxy functionality is deployed in campus unified communications network. This solution is ideal for deployments that utilize end to end encryption and firewalls to protect Unified Communications Manager servers.

Mobility Proxy: Secure connectivity between Cisco Unified Mobility Advantage server and Cisco Unified Mobile Communicator clients

Cisco Unified Mobility solutions include the Cisco Unified Mobile Communicator (Cisco UMC), an easy-to-use software application for mobile handsets that extends enterprise communications applications and services to mobile phones and the Cisco Unified Mobility Advantage (Cisco UMA) server. The Cisco Unified Mobility solution streamlines the communication experience, enabling single number reach and integration of mobile endpoints into the Unified Communications infrastructure.

The security appliance acts as a proxy, terminating and reoriginating the TLS signaling between the Cisco UMC and Cisco UMA. As part of the proxy security functionality, inspection is enabled for the Cisco UMA Mobile Multiplexing Protocol (MMP), the protocol between Cisco UMC and Cisco UMA.

Presence Federation Proxy: Secure connectivity between Cisco Unified Presence servers and Cisco/Microsoft Presence servers

Cisco Unified Presence solution collects information about the availability and status of users, such as whether they are using communication devices, such as IP phones at particular times. It also collects information regarding their communications capabilities, such as whether web collaboration or video conferencing is enabled. Using user information captured by Cisco Unified Presence, applications such as Cisco Unified Personal Communicator and Cisco UCM can improve productivity by helping users connect with colleagues more efficiently through determining the most effective way for collaborative communication.

Using the ASA as a secure presence federation proxy, businesses can securely connect their Cisco Unified Presence (Cisco UP) servers to other Cisco or Microsoft Presence servers, enabling intra-enterprise communications. The security appliance terminates the TLS connectivity between the servers, and can inspect and apply policies for the SIP communications between the servers.

Deprecated UC Proxies

Prior to ASA 9.4(1), and ASDM 7.4(1), you could configure the following proxies. These are no longer supported. We recommend that you do not configure them, even in older software releases. Instead, configure SIP inspection with TLS Proxy.

- Phone Proxy, which replaced Cisco Unified Phone Proxy.
- Cisco Unified Communications Intercompany Media Engine.

TLS Proxy Applications in Cisco Unified Communications

The following table shows the Cisco Unified Communications applications that utilize the TLS proxy on the ASA.

Application	TLS Client	TLS Server	ClientAuthentication	Security Appliance Server Role	Security Appliance Client Role
TLS Proxy	IP phone	Cisco UCM	Yes	Proxy certificate, self-signed or by internal CA	Local dynamic certificate signed by the ASA CA
Mobility Proxy	Cisco UMC	Cisco UMA	No	Using the Cisco UMA private key or certificate impersonation	Any static configured certificate

Application	TLS Client	TLS Server	Client Authentication	Security Appliance Server Role	Security Appliance Client Role
Presence Federation Proxy	Cisco UP or MS LCS/OCS	Cisco UP or MS LCS/OCS	Yes	Proxy certificate, self-signed or by internal CA	Using the Cisco UP private key or certificate impersonation

The ASA supports TLS proxy for various voice applications. The TLS proxy running on the ASA has the following key features:

- The ASA forces remote IP phones connecting to the phone proxy through the Internet to be in secured mode even when the Cisco UCM cluster is in non-secure mode.
- The TLS proxy is implemented on the ASA to intercept the TLS signaling from IP phones.
- The TLS proxy decrypts the packets, sends packets to the inspection engine for NAT rewrite and protocol conformance, optionally encrypts packets, and sends them to Cisco UCM or sends them in clear text if the IP phone is configured to be in nonsecure mode on the Cisco UCM.
- The ASA acts as a media terminator as needed and translates between SRTP and RTP media streams.
- The TLS proxy is a transparent proxy that works based on establishing trusted relationship between the TLS client, the proxy (the ASA), and the TLS server.

For the Cisco Unified Mobility solution, the TLS client is a Cisco UMA client and the TLS server is a Cisco UMA server. The ASA is between a Cisco UMA client and a Cisco UMA server. The mobility proxy (implemented as a TLS proxy) for Cisco Unified Mobility allows the use of an imported PKCS-12 certificate for server proxy during the handshake with the client. Cisco UMA clients are not required to present a certificate (no client authentication) during the handshake.

For the Cisco Unified Presence solution, the ASA acts as a TLS proxy between the Cisco UP server and the foreign server. This allows the ASA to proxy TLS messages on behalf of the server that initiates the TLS connection, and route the proxied TLS messages to the client. The ASA stores certificate trustpoints for the server and the client, and presents these certificates on establishment of the TLS session.

Licensing for Cisco Unified Communications Proxy Features

The Cisco Unified Communications proxy features supported by the ASA require a Unified Communications Proxy license. For information about the available licenses for your model, see the licensing document for your ASA version.

The following applications use TLS proxy sessions for their connections. Each TLS proxy session used by these applications (and only these applications) is counted against the UC license limit:

- Presence Federation Proxy
- Encrypted Voice Inspection

Other applications that use TLS proxy sessions do not count towards the UC limit, for example, Mobility Advantage Proxy (which does not require a license).

Some UC applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections, so 2 UC Proxy sessions are used.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command (CLI) or**Configuration** > **Firewall** > **Unified Communications** > **TLS Proxy** pane (ASDM). To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a UC license that is higher than the default TLS proxy limit, the ASA automatically sets the TLS proxy limit to match the UC limit. The TLS proxy limit takes precedence over the UC license limit; if you set the TLS proxy limit to be less than the UC license, then you cannot use all of the sessions in your UC license.

Consider the following:

- For license part numbers ending in "K8" (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in "K9" (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.
- If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the UC license limit, then you see an error message to use the **tls-proxy maximum-sessions** command to raise the limit again (in ASDM, use the **TLS Proxy** pane). If you use failover and enter the write standby command (CLI) or use **File** > **Save Running Configuration to Standby Unit** (ASDM) on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:

- For K8 licenses, SRTP sessions are limited to 250.

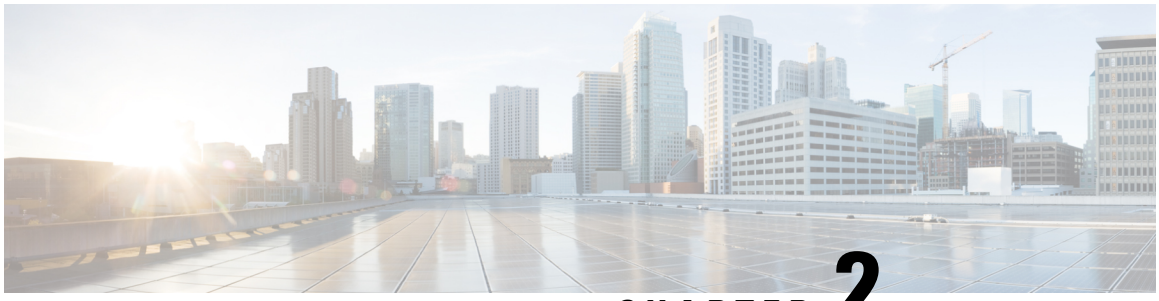
  For K9 licenses, there is not limit.

Note Only calls that require encryption/decryption for media are counted towards the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count towards the limit.

Guidelines and Limitations

Consider the following the guidelines and limitations for this feature.

- For all Unified Communications proxies to function correctly, you must synchronize the clock on the ASA and all servers associated with each proxy, such as the Cisco Unified Communication Manager server, the Cisco Mobility Advantage server, and the Cisco Unified Presence server.
- If the ASA on which you configure the Cisco Mobility Advantage Proxy and the Cisco Presence Federation Proxy is located behind another firewall, you must ensure that the public IP addresses for the Cisco Mobility Advantage server and the Cisco Unified Presence server are accessible from the Internet.
- If you use the Unified Communication Wizard in ASDM to create to the Presence Federation Proxy, you might be required to adjust the configuration of the ACLs created automatically by the wizard.



CHAPTER 2

TLS Proxy for Encrypted Voice Inspection

This chapter describes how to configure the TLS Proxy so that the system can inspect encrypted voice connections.

This chapter includes the following sections:

Information About the TLS Proxy for Encrypted Voice Inspection

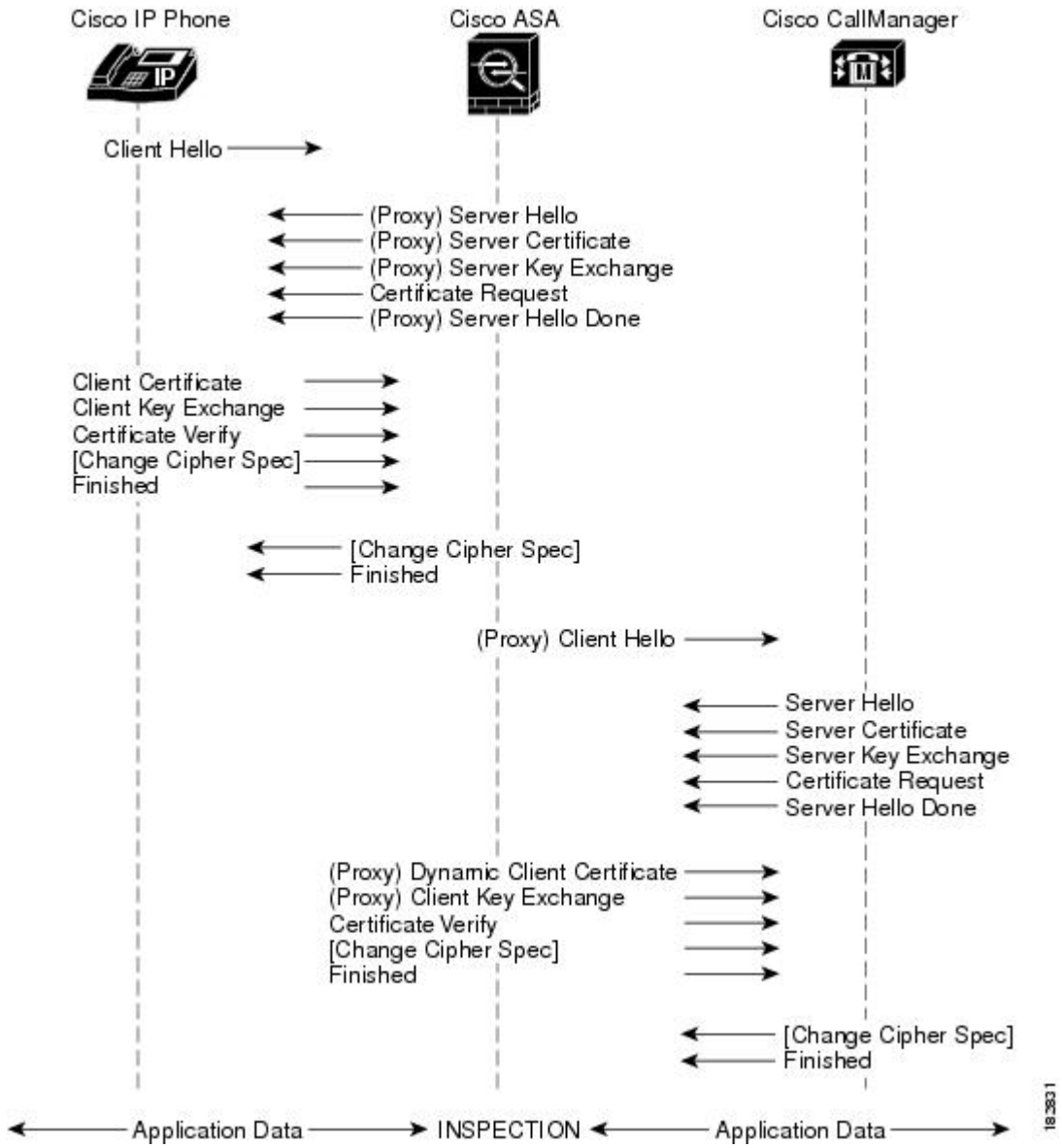
End-to-end encryption often leaves network security appliances “blind” to media and signaling traffic, which can compromise access control and threat prevention security functions. This lack of visibility can result in a lack of interoperability between the firewall functions and the encrypted voice, leaving businesses unable to satisfy both of their key security requirements.

The ASA is able to intercept and decrypt encrypted signaling from Cisco encrypted endpoints to the Cisco Unified Communications Manager (Cisco UCM), and apply the required threat protection and access control. It can also ensure confidentiality by re-encrypting the traffic onto the Cisco UCM servers.

Typically, the ASA TLS Proxy functionality is deployed in campus unified communications network. This solution is ideal for deployments that utilize end to end encryption and firewalls to protect Unified Communications Manager servers.

The security appliance in the following figure serves as a proxy for both client and server, with Cisco IP Phone and Cisco UCM interaction.

Figure 1: TLS Proxy Flow



Decryption and Inspection of Unified Communications Encrypted Signaling

With encrypted voice inspection, the security appliance decrypts, inspects and modifies (as needed, for example, performing NAT), and re-encrypts voice signaling traffic while all of the existing VoIP inspection functions for SIP are preserved. Once voice signaling is decrypted, the plain text signaling message is passed to the existing inspection engines.

The security appliance acts as a TLS proxy between the Cisco IP Phone and Cisco UCM. The proxy is transparent for the voice calls between the phone and the Cisco UCM. Cisco IP Phones download a Certificate Trust List from the Cisco UCM before registration which contains identities (certificates) of the devices that the phone should trust, such as TFTP servers and Cisco UCM servers. To support server proxy, the CTL file must contain the certificate that the security appliance creates for the Cisco UCMs.

To proxy calls on behalf of the Cisco IP Phone, the security appliance presents a certificate that the Cisco UCM can verify, which is a Local Dynamic Certificate for the phone, issued by the certificate authority on the security appliance.

TLS proxy is supported by the Cisco Unified CallManager Release 5.1 and later. You should be familiar with the security features of the Cisco UCM. For background and detailed description of Cisco UCM security, see the Cisco Unified CallManager documentation.

Supported Cisco UCM and IP Phones for the TLS Proxy

Cisco Unified Communications Manager

The following releases of the Cisco Unified Communications Manager are supported with the TLS proxy:

- Cisco Unified CallManager Version 5.1
- Cisco Unified Communications Manager 6.1
- Cisco Unified Communications Manager 7.0
- Cisco Unified Communications Manager 8.0
- Cisco Unified Communications Manager 8.6
- Cisco Unified Communications Manager 10.5

Cisco Unified IP Phones

The following IP phones are supported with the TLS proxy:

- Cisco Unified IP Phone 7985
- Cisco Unified IP Phone 7975
- Cisco Unified IP Phone 7971
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7965
- Cisco Unified IP Phone 7962
- Cisco Unified IP Phone 7961
- Cisco Unified IP Phone 7961G-GE
- Cisco Unified IP Phone 7960
- Cisco Unified IP Phone 7945
- Cisco Unified IP Phone 7942
- Cisco Unified IP Phone 7941
- Cisco Unified IP Phone 7941G-GE
- Cisco Unified IP Phone 7940
- Cisco Unified Wireless IP Phone 7921
- Cisco Unified Wireless IP Phone 7925

- Cisco Unified IP Conference Phone 8831
- Cisco IP Communicator (CIPC) for softphones

Incorporating the Firewall into the Unified Communications System

Configuring the ASA is not enough to fully incorporate the firewall into the Cisco Unified Communications system. You must also add the ASA to the Certificate Trust List (CTL) using the Cisco Certificate Trust List Client, which is part of the Unified Communications Manager.

When you configure a firewall in the CTL file, you can secure a ASA firewall as part of a secure Cisco Unified Communications Manager system. The Cisco CTL Client displays the firewall certificate as a "CCM" certificate.

When configured correctly, the ASA receives the CTL file from the CTL provider. However, the ASA does not store the raw CTL file in the flash, rather, it parses the CTL file and installs the appropriate trustpoints.

For detailed information on how to add the ASA as a firewall to the Unified Communications Manager system, look for information on the CTL Client Setup in the *Security Guide for Cisco Unified Communications Manager* for the software version you are using. You can find the documents at http://www.cisco.com/c/en/us/support/unified-communications/unified-communications-manager-call%20manager/products-maintenance-guides-list.html.

Also see the Security Guide for information on installing, exporting, and creating UCM-side certificates. You will need to import the ASA certificate into UCM.

Configuring the TLS Proxy for Encrypted Voice Inspection (CLI)

The following procedure explains the end-to-end process for enabling inspection of encrypted voice traffic.

Before you begin

Prerequisites for the TLS Proxy for Encrypted Voice Inspection

Before configuring TLS proxy, the following prerequisites are required:

- You must set clock on the security appliance before configuring TLS proxy. To set the clock manually and display clock, use the **clock set** and **show clock** commands. We recommend that the security appliance use the same NTP server as the Cisco Unified CallManager cluster. TLS handshake may fail due to certificate validation failure if clock is out of sync between the security appliance and the Cisco Unified CallManager server.
- 3DES-AES license is needed to interoperate with the Cisco Unified CallManager. AES is the default cipher used by the Cisco Unified CallManager and Cisco IP Phone.

Step 1 (Optional) Set the maximum number of TLS proxy sessions to be supported by the security appliance. For example:

Example:

```
ciscoasa(config)# tls-proxy maximum-sessions 1200
```

The default and maximum differ by device model. This command controls the memory size reserved for cryptographic applications such as TLS proxy. Crypto memory is reserved at the time of system boot. If you increase the number, you must reboot the system to reserve the additional memory.

Step 2 Create the Proxy Trustpoint for the Unified Call Manager Cluster

Step 3 Create the Internal Local CA to Sign Local Dynamic Certificates for Phones

Step 4 Create a CTL Provider

Step 5 Create the TLS Proxy

Step 6 Enable TLS Proxy for SIP Inspection

Step 7 Export the local CA certificate (ldc_server) and install it as a trusted certificate on the Cisco UCM server.

a. Use the following command to export the certificate if a trust-point with proxy-ldc-issuer is used as the signer of the dynamic certificates, for example:

```
ciscoasa(config)# crypto ca export ldc_server identity-certificate
```

b. For the embedded local CA server LOCAL-CA-SERVER, use the following command to display the certificate, which you can then copy and paste to a text file. For example:

```
ciscoasa(config)# show ca server certificate
```

c. Import the certificate into UCM. For detailed information, see the *Security Guide for Cisco Unified Communications Manager* for the software version you are using. You can find the documents at http://www.cisco.com/c/en/us/support/unified-communications/unified-communications-managercallmanager/products-maintenance-guides-list.html.

Step 8 Run the CTL Client application to add the server proxy certificate (ccm_proxy) to the CTL file and install the CTL file on the security appliance.

For more information, see Incorporating the Firewall into the Unified Communications System.

Create the Proxy Trustpoint for the Unified Call Manager Cluster

The Cisco UCM proxy certificate could be self-signed or issued by a third-party CA. The certificate is exported to the CTL client. The following procedure shows how to create a self-signed proxy.

The ASA uses this proxy when authenticating with the phones. The ASA acts as the server in place of the Call Manager.

Step 1 Create the RSA keypair for the trustpoint.

crypto key generate rsa label *key-pair-label* **modulus** *size*

Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits. The default is 2048, so you need to include the **modulus** keyword.

Example:

```
ciscoasa(config)# crypto key generate rsa label ccm_proxy_key modulus 1024
INFO: The name for the keys will be: ccm_proxy_key
Keypair generation process begin. Please wait...
ciscoasa(config)#
```

Step 2 Create the proxy trustpoint for the Unified Call Manager cluster.

crypto ca trustpoint *trustpoint_name*

You enter trustpoint configuration mode, where you can configure the trustpoint characteristics.

Example:

```
ciscoasa(config)# crypto ca trustpoint ccm_proxy
```

Step 3 Generate a self-signed certificate.

enrollment self

Example:

```
ciscoasa(config-ca-trustpoint)# enrollment self
```

Step 4 Do not to include a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.

fqdn none

Example:

```
ciscoasa(config-ca-trustpoint)# fqdn none
```

Step 5 Specify a subject DN in the certificate during enrollment.

subject-name *X.500_name*

Cisco IP Phones require certain fields from the X.509v3 certificate to be present to validate the certificate by consulting the CTL file. Consequently, the **subject-name** entry must be configured for a proxy certificate trustpoint. The subject name must be composed of the ordered concatenation of the CN, OU and O fields. The CN field is mandatory; the others are optional.

If you use multiple attributes, separate them with a semicolon. Use one of the following forms:

- CN=xxx;OU=yyy;O=zzz
- CN=xxx;OU=yyy
- CN=xxx;O=zzz
- CN=xxx

Example:

```
ciscoasa(config-ca-trustpoint)# subject-name cn=EJW-SV-1-Proxy
```

Step 6 Specify the key pair you created for the trustpoint.

keypair *keyname*

Example:

```
ciscoasa(config-ca-trustpoint)# keypair ccm_proxy_key
```

Step 7 Enroll the trustpoint.

crypto ca enroll *trustpoint_name*

Example:

```
ciscoasa(config-ca-trustpoint)# crypto ca enroll ccm_proxy
WARNING: The certificate enrollment is configured with an fqdn
that differs from the system fqdn. If this certificate will be
used for VPN authentication this may cause connection problems.

Would you like to continue with this enrollment? [yes/no]: yes
```

```
% The fully-qualified domain name will not be included in the certificate

% Include the device serial number in the subject name? [yes/no]: no

Generate Self-Signed Certificate? [yes/no]: yes
ciscoasa(config)#
```

Create the Internal Local CA to Sign Local Dynamic Certificates for Phones

Create an internal local CA to sign the LDC for Cisco IP Phones. This local CA is a regular self-signed trustpoint with **proxy-ldc-issuer** enabled. The ASA presents these certificates to the Call Manager server on behalf of the phones, securing the connection between the ASA and the Call Manager.

You can alternatively use the embedded local CA LOCAL-CA-SERVER on the ASA to issue the LDC, but this is not recommended.

Step 1 Create the RSA keypair for the local CA.

crypto key generate rsa label *key-pair-label* **modulus** *size*

Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits. The default is 2048, so you need to include the **modulus** keyword.

Example:

```
ciscoasa(config)# crypto key generate rsa label ldc_signer_key modulus 1024
INFO: The name for the keys will be: ldc_signer_key
Keypair generation process begin. Please wait...
ciscoasa(config)#
```

Step 2 Create the proxy trustpoint for the local Certificate Authority (CA) for signing local dynamic certificates.

crypto ca trustpoint *trustpoint_name*

You enter trustpoint configuration mode, where you can configure the trustpoint characteristics.

Example:

```
ciscoasa(config)# crypto ca trustpoint ldc_server
```

Step 3 Generate a self-signed certificate.

enrollment self

Example:

```
ciscoasa(config-ca-trustpoint)# enrollment self
```

Step 4 Define the CA as one that can issue local dynamic certificates (LDC).

proxy-ldc-issuer

You can configure this option for a local CA only if you also specify **enrollment self**.

The ASA generates a local dynamic certificate for each phone that registers with the Call Manager. When the phone unregisters, the dynamic certificate is automatically deleted. These certificates do not appear in the running configuration; they are created and destroyed as needed.

Example:

```
ciscoasa(config-ca-trustpoint)# proxy-ldc-issuer
```

Step 5 Add a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.

fqdn *name*

Example:

```
ciscoasa(config-ca-trustpoint)# fqdn my-ldc-ca.example.com
```

Step 6 Specify a subject DN in the certificate during enrollment.

subject-name *X.500_name*

The CN field is mandatory; the others are optional. If you use multiple attributes, separate them with a semicolon. Use one of the following forms:

- CN=xxx;OU=yyy;O=zzz
- CN=xxx;OU=yyy
- CN=xxx;O=zzz
- CN=xxx

Example:

```
ciscoasa(config-ca-trustpoint)# subject-name cn=FW_LDC_SIGNER_172_23_45_200
```

Step 7 Specify the key pair you created for the local CA.

keypair *keyname*

Example:

```
ciscoasa(config-ca-trustpoint)# keypair ldc_signer_key
```

Step 8 Enroll the trustpoint.

crypto ca enroll *trustpoint_name*

Example:

```
ciscoasa(config-ca-trustpoint)# crypto ca enroll ldc_server
WARNING: The certificate enrollment is configured with an fqdn
that differs from the system fqdn. If this certificate will be
used for VPN authentication this may cause connection problems.

Would you like to continue with this enrollment? [yes/no]: yes

% The fully-qualified domain name in the certificate will be:
my-ldc-ca.example.com

% Include the device serial number in the subject name? [yes/no]: no

Generate Self-Signed Certificate? [yes/no]: yes
ciscoasa(config)#
```

Create a CTL Provider

Create a CTL Provider in preparation for a connection from the CTL Client.

Step 1 Create the Certificate Trust List (CTL) provider.

ctl-provider *ctl_name*

You enter CTL provider configuration mode, where you can configure the provider characteristics.

Example:

```
ciscoasa(config)# ctl-provider my_ctl
```

Step 2 Specify the addresses of the CTL clients that should be able to connect with the CTL provider on the ASA.

client interface *if_name ipv4_address*

Where *if_name* is the interface through which the client can be reached, and the IPv4 address is the address of the workstation on which the CTL client is installed. Enter this command as many times as needed to identify all CTL clients that you want to allow.

Example:

```
ciscoasa(config-ctl-provider)# client interface inside address 172.23.45.1
```

Step 3 Specify the username and password for client authentication.

client username *name* **password** *password* **[encrypted]**

The username and password must be the username and password for Cisco UCM administration. Specify the optional **encrypted** keyword if the password is encrypted (in which case it must be 16 characters).

Example:

```
ciscoasa(config-ctl-provider)# client username CCMAdministrator
password XXXXXX
```

Step 4 Export the proxy trustpoint you created for the Cisco UCM server to the CTL client.

export certificate *trustpoint_name*

The certificate will be added to the Certificate Trust List file composed by the CTL client and subsequently distributed to all the phones that download the CTL file. Specify the name of the trustpoint you created in Create the Proxy Trustpoint for the Unified Call Manager Cluster. This ensures that each phone has the server certificate the ASA uses to authenticate the connection.

Example:

```
ciscoasa(config-ctl-provider)# export certificate ccm_proxy
```

Step 5 If necessary, change the port on which the CTL provider listens.

service port *number*

The default port number listened to by the CTL Provider is TCP 2444, which is the default CTL port on the Cisco UCM. If you changed this port in the Cisco UCM, you must specify the port on this command so that the CTL Provider can communicate with the CTL client. The port must be in the range 2000-9999.

Example:

```
ciscoasa(config-ctl-provider)# service port 2445
```

Step 6 Enable the CTL provider to parse the CTL file from the CTL client and install trustpoints for entries from the CTL file.

ctl install

Trustpoints installed by this command have names prefixed with "_internal_CTL_<ctl_name>."

Example:

```
ciscoasa(config-ctl-provider)# ctl install
```

Create the TLS Proxy

Create the TLS proxy to handle the encrypted signaling.

Step 1 Create the RSA keypair for the local CA.

crypto key generate rsa label *key-pair-label* **modulus** *size*

Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits. The default is 2048, so you need to include the **modulus** keyword.

Example:

```
ciscoasa(config)# crypto key generate rsa label phone_common modulus 1024
INFO: The name for the keys will be: phone_common
Keypair generation process begin. Please wait...
ciscoasa(config)#
```

Step 2 Create the TLS proxy.

tls-proxy *name*

You enter TLS proxy configuration mode, where you can configure the proxy characteristics.

Example:

```
ciscoasa(config)# tls-proxy my_proxy
```

Step 3 Specify the proxy trustpoint certificate to present during the TLS handshake with phone clients.

server trust-point *proxy_trustpoint*

Specify the name of the trustpoint you created in Create the Proxy Trustpoint for the Unified Call Manager Cluster.

The **server** command configures the proxy parameters for the original TLS server. In other words, the parameters for the ASA to act as the server during a TLS handshake the TLS clients.

Example:

```
ciscoasa(config-tlsp)# server trust-point ccm_proxy
```

Step 4 Specify the local CA trustpoint to provide the local dynamic certificates.

client ldc issuer *ca_trustpoint_name*

Specify the name of the local CA you created in Create the Internal Local CA to Sign Local Dynamic Certificates for Phones. This trustpoint must include the proxy-ldc-issuer command or be the default local CA server (LOCAL-CA-SERVER).

Example:

```
ciscoasa(config-tlsp)# client ldc issuer ldc_server
```

Step 5 Specify the key pair you created for the local dynamic certificates.

client ldc key-pair *keyname*

Example:

```
ciscoasa(config-tlsp)# client ldc key-pair phone_common
```

Step 6 (Optional.) Configure the cipher suite to use when the proxy acts as a client to the Cisco UCM.

client cipher-suite *cipher_suite*

For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite, or the one defined by the **ssl encryption** command. You can use this command to achieve difference ciphers between the two TLS sessions. You should use AES ciphers with the CallManager server. Separate multiple options with spaces.

Example:

```
ciscoasa(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1
```

Step 7 (Optional.) Configure the cipher suite to use when the proxy acts as a server to the phones.

If you do not define the ciphers the TLS proxy can use, the proxy server uses the global cipher suite defined by the **ssl cipher** command. By default, the global cipher level is medium, which means all ciphers are available except for NULL-SHA, DES-CBC-SHA, and RC4-MD5. Specify the **server cipher-suite** command only if you want to use a different suite than the one generally available on the ASA. Separate multiple options with spaces.

To set the minimum TLS version for all SSL server connections on the ASA, see the **ssl server-version**command. The default is TLS v1.0.

Example:

```
ciscoasa(config-tlsp)# server cipher-suite aes128-sha1 aes256-sha1
```

Enable TLS Proxy for SIP Inspection

Configure the required service policies to enable encrypted voice inspection for SIP.

Because secure protocols use different ports than the regular unencrypted versions, you need to configure unique classes for TLS proxy inspection.

- Secure SIP (SIPS) uses TCP/UDP 5061 (rather than SIP's 5060).

The inspection_default class filters on the unencrypted ports only.

The following procedure explains how to create these classes and add the TLS proxy inspections to the existing global_policy policy map. Alternatively, you can create service policies for specific interfaces.

For more detailed information on how to configure service policies, see the firewall configuration guide.

Before you begin

You can configure inspection policy maps to customize the inspection. If you do not want to use the default settings for the inspections, configure the inspection policy maps before configuring the service policy. For details on customizing SIP, see the firewall configuration guide. The following procedure assumes you are using the default settings.

Step 1 Create the class for secure SIP.

Because SIP endpoints can use TCP or UDP, you cannot create a simple port match for the class. Instead, create a service group for TCP/5061 and UDP/5061, then use that object in an ACL.

a. Create a service object group for TCP/UDP 5061.

```
ciscoasa(config)# object-group service sec_sip_ports
ciscoasa(config-service-object-group)# service-object
tcp-udp destination eq 5061
```

b. Create an ACL that matches secure SIP traffic for all addresses.

```
ciscoasa(config)# access-list sec_sip_acl extended permit
object-group sec_sip_ports any any
ciscoasa(config)# show access-list
access-list sec_sip_acl; 2 elements; name hash: 0x46fa3345
access-list sec_sip_acl line 1 extended permit object-group
sec_sip_ports any any (hitcnt=0) 0x04ff39a5
access-list sec_sip_acl line 1 extended permit
tcp any any eq 5061 (hitcnt=0) 0x23e41037
access-list sec_sip_acl line 1 extended permit
udp any any eq 5061 (hitcnt=0) 0x511cfebe
```

c. Create the class map using an ACL match.

```
ciscoasa(config)# class-map sec_sip
ciscoasa(config-cmap)# match access-list sec_sip_acl
```

Step 2 Edit the global_policy policy map.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)#
```

Step 3 Add the Secure SIP class and configure SIP inspection with the TLS proxy.

```
ciscoasa(config-pmap)# class sec_sip
ciscoasa(config-pmap-c)# inspect sip tls-proxy my_proxy
```

Step 4 Verify the global_policy policy map now has the expected content.

In the following output, you can see that the running configuration performs SIP inspection without the TLS proxy on the unencrypted default ports that match the inspection_default class. Then, the TLS proxy inspections appear for the correct sec_sip class for the encrypted ports.

```
ciscoasa(config-pmap-c)# show run policy-map global_policy
!
policy-map global_policy
class inspection_default
inspect dns migrated_dns_map_1
inspect ftp
inspect ip-options
inspect netbios
inspect rsh
inspect esmtp
inspect sqlnet
inspect sunrpc
inspect tftp
inspect xdmcp
inspect sip
inspect skinny
class sec_sip
```

```
inspect sip tls-proxy my_proxy
!
```

Step 5 Enable the global_policy service policy to implement your changes.

```
ciscoasa(config-pmap-c)# service-policy global_policy global
```

Configuring the TLS Proxy for Encrypted Voice Inspection (ASDM)

The following procedure explains the end-to-end process for enabling inspection of encrypted voice traffic using ASDM.

Step 1 Create the Proxy Trustpoint for the Unified Call Manager Cluster (ASDM).

Step 2 Create the Internal Local CA to Sign Local Dynamic Certificates for Phones (ASDM)

Step 3 Create a CTL Provider (ASDM)

Step 4 Create the TLS Proxy (ASDM)

Step 5 Enable TLS Proxy for SIP Inspection

Step 6 Export the local CA certificate (ldc_server) and install it as a trusted certificate on the Cisco UCM server.

a. Choose **Configuration** > **Firewall** > **Advanced** > **Certificate Management** > **Identity Certificates**

b. Select the LDC trustpoint and click **Export**.

c. Specify a file name and click **Export Certificate**.

d. Import the certificate into UCM. For detailed information, see the *Security Guide for Cisco Unified Communications Manager* for the software version you are using. You can find the documents at http://www.cisco.com/c/en/us/support/unified-communications/unified-communications-managercallmanager/products-maintenance-guides-list.html.

Step 7 Run the CTL Client application to add the server proxy certificate (ccm_proxy) to the CTL file and install the CTL file on the security appliance.

For more information, see Incorporating the Firewall into the Unified Communications System.

Create the Proxy Trustpoint for the Unified Call Manager Cluster (ASDM)

The Cisco UCM proxy certificate could be self-signed or issued by a third-party CA. The certificate is exported to the CTL client. The following procedure shows how to create a self-signed proxy.

The ASA uses this proxy when authenticating with the phones. The ASA acts as the server in place of the Call Manager.

Step 1 Choose **Device Management** > **Certificate Management** > **Identity Certificates**then click the **Add** button.

Step 2 Enter a Trustpoint Name. For example, **ccm_proxy**.

Step 3 Choose **Add a New Identity Certificate**.

Step 4 For **Key Pair**, create a new key pair:

a. Click **New**.

b. Select **RSA** as the key type.

c. Select **Enter New Key Pair Name**, then enter a name. For example, **ccm_proxy_key**.

d. For **Size**, select **1024**.

Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits.

e. For **Usage**, select **General Purpose**.

f. Click **Generate Now**.

The key is generated and you are returned to the Add Identity Certificate dialog box with the new key automatically selected.

Step 5 For **Certificate Subject DN**, specify a subject DN.

Cisco IP Phones require certain fields from the X.509v3 certificate to be present to validate the certificate by consulting the CTL file. Consequently, you must configure a subject DN for a proxy certificate trustpoint. The subject name must be composed of the ordered concatenation of the CN, OU and O fields. The CN field is mandatory; the others are optional.

If you use multiple attributes, separate them with a semicolon. Use one of the following forms:

- CN=xxx;OU=yyy;O=zzz
- CN=xxx;OU=yyy
- CN=xxx;O=zzz
- CN=xxx

For example, **cn=EJW-SV-1-Proxy**.

Step 6 Choose **Generate Self-Signed Certificate**.

Note Do not select the Act as local certificate authority and issue dynamic certificates for TLS proxy option for this trustpoint.

Step 7 Click the **Advanced** button, and clear the **FQDN** field on the Certificate Parameters tab. Click **OK**.

Do not to include a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.

Step 8 Click **Add Certificate**.

Create the Internal Local CA to Sign Local Dynamic Certificates for Phones (ASDM)

Create an internal local CA to sign the LDC for Cisco IP Phones. This local CA is a regular self-signed trustpoint with the ability to issue local dynamic certificates. The ASA presents these certificates to the Call Manager server on behalf of the phones, securing the connection between the ASA and the Call Manager.

You can alternatively use the embedded local CA (LOCAL-CA-SERVER) on the ASA to issue the LDC, but this is not recommended.

Step 1 Choose **Device Management** > **Certificate Management** > **Identity Certificates** then click the **Add** button.

Step 2 Enter a Trustpoint Name. For example, **ldc_server**.

Step 3 Choose **Add a New Identity Certificate**.

Step 4 For **Key Pair**, create a new key pair:

a. Click **New**.

b. Select **RSA** as the key type.

c. Select **Enter New Key Pair Name**, then enter a name. For example, **ldc_signer_key**.

d. For **Size**, select **1024**.

Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits.

e. For **Usage**, select **General Purpose**.

f. Click **Generate Now**.

The key is generated and you are returned to the Add Identity Certificate dialog box with the new key automatically selected.

Step 5 For **Certificate Subject DN**, specify a subject DN.

The CN field is mandatory; the others are optional. If you use multiple attributes, separate them with a semicolon. Use one of the following forms:

- CN=xxx;OU=yyy;O=zzz
- CN=xxx;OU=yyy
- CN=xxx;O=zzz
- CN=xxx

For example, **cn=FW_LDC_SIGNER_172_23_45_200**.

Step 6 Choose **Generate Self-Signed Certificate**.

Step 7 Choose the **Act as local certificate authority and issue dynamic certificates for TLS proxy** option.

The ASA generates a local dynamic certificate for each phone that registers with the Call Manager. When the phone unregisters, the dynamic certificate is automatically deleted. These certificates do not appear in the running configuration; they are created and destroyed as needed.

Step 8 Click the **Advanced** button, and enter an fully-qualified domain name in the **FQDN** field on the Certificate Parameters tab. Click **OK**.

For example, **my-ldc-ca.example.com**.

Step 9 Click **Add Certificate**.

Create a CTL Provider (ASDM)

Create a CTL Provider in preparation for a connection from the CTL Client.

Step 1 Select **Configuration** > **Firewall** > **Unified Communications** > **CTL Provider**

Step 2 Click **Add** to create a new provider, or select a provider and click **Edit**.

Step 3 Enter a **CTL Provider Name**. For example, **my_ctl**.

Step 4 In **Certificate to be Exported**, select the proxy trustpoint you created for the Cisco UCM server to the CTL client.

The certificate will be added to the Certificate Trust List file composed by the CTL client and subsequently distributed to all the phones that download the CTL file. Specify the name of the trustpoint you created in Create the Proxy Trustpoint for the Unified Call Manager Cluster (ASDM). This ensures that each phone has the server certificate the ASA uses to authenticate the connection.

For example, select **ccm_proxy**.

Step 5 Specify the addresses of the CTL clients that should be able to connect with the CTL provider on the ASA.

For each client, select the interface through which the client can be reached, and then enter the IPv4 address of the workstation on which the CTL client is installed. Click **Add>>** to add it to the list of clients. For example, **inside 172.23.45.1**.

Repeat the process as many times as needed to identify all CTL clients that you want to allow.

Step 6 Click **More Options**, and specify the username and password for client authentication.

The username and password must be the username and password for Cisco UCM administration. For example, CCMAdministrator, password XXXXXX.

Step 7 If necessary, change the port on which the CTL provider listens in the **Port Number** field.

The default port number listened to by the CTL Provider is TCP 2444, which is the default CTL port on the Cisco UCM. If you changed this port in the Cisco UCM, you must specify the port on this command so that the CTL Provider can communicate with the CTL client. The port must be in the range 2000-9999.

Step 8 Ensure that the **Parse the CTL File Provided by the CTL Client and Install Trustpoints** option is selected.

The system installs trustpoints for entries from the CTL file. Trustpoints installed from the file have names prefixed with "_internal_CTL_<ctl_name>." If you disable this option, you must manually import and install each Call Manager server and CAPF certificate.

Step 9 Click **OK**.

Create the TLS Proxy (ASDM)

Create the TLS proxy to handle the encrypted signaling. The following procedure explains how to configure the proxy for encrypted SIP inspection.

Step 1 Select **Configuration** > **Firewall** > **Unified Communications** > **TLS Proxy**.

Step 2 (Optional) Set the maximum number of TLS proxy sessions to be supported by the security appliance.

The default and maximum differ by device model. The **Maximum TLS Sessions** option controls the memory size reserved for cryptographic applications such as TLS proxy. Crypto memory is reserved at the time of system boot. If you increase the number, you must reboot the system to reserve the additional memory.

To change the default, select **Specify the Maximum Number of TLS Proxy Sessions that the ASA Needs to Support**, and enter the maximum number of sessions. These fields are below the table of TLS proxies.

Step 3 Click **Add** to create a new TLS proxy, or select a proxy and click **Edit**.

When you create a new proxy, you are taken through a wizard to configure the required properties. When editing an existing proxy, the wizard steps are presented as separate tabs. The following steps assume you are adding a new proxy.

Step 4 Enter a name for the proxy in **TLS Proxy Name**. For example, **my_proxy**. Click **Next**.

Step 5 Configure the options to use when the ASA acts as the TLS server for the phone clients.

For encrypted SIP inspection as explained in this procedure, the ASA acts as a proxy for Cisco Unified Call Manager. However, you could also configure the proxy for a Cisco Unified Presence Server (CUPS), or when configuring MMP inspection for a Cisco Unified Mobility Advantage (CUMA) server.

a. In **Server Proxy Certificate**, select the trustpoint you created in Create the Proxy Trustpoint for the Unified Call Manager Cluster (ASDM). For example, ccm_proxy.

The server proxy configures the proxy parameters for the original TLS server. In other words, the parameters for the ASA to act as the server during a TLS handshake the TLS clients.

If you have not already created the identity certificate, you can click **Manage** to add it.

b. (Optional.) You can define the security algorithms (ciphers) that the server can use by moving them from the available algorithms to the active algorithms list. If you do not specify ciphers, the default system ciphers are used. You can move the algorithms up or down to change their relative priority.

c. (Optional.) Click **Install TLS Server's Certificate** to install the Call Manager certificate. Because this certificate will be installed from the CTL file, you do not need to install it. However, if you are not installing trustpoints from the CTL file, obtain the certificate from the Call Manager server and install it now.

d. Select **Enable client authentication during TLS Proxy handshake**. You would deselect this option only for testing purposes, or for MMP inspection.

For MMP inspection, used for Unified Mobility Advantage (CUMA), the client is not able to present a certificate, so authentication is not possible.

e. Click **Next**.

Step 6 Configure the options to use when the ASA acts as the TLS client for the original TLS server.

For encrypted SIP inspection as explained in this procedure, the ASA acts as a proxy for IP phones. However, you could also configure the proxy for a Microsoft LCS/OCS client for Cisco Unified Presence Federation, or when configuring MMP inspection for a Cisco Unified Mobility Advantage (CUMA) client.

a. Choose the **Specify the Internal Certificate Authority to Sign the Local Dynamic Certificates for Phones**.

The other options are used for other types of inspection:

- **Configure the proxy client to use clear text to communicate with the remote TCP server**. This option configures TLS offload for Diameter inspection, for use when the ASA is in the same data center as the Diameter server.
- **Specify the proxy certificate for TLS Client**. This option is for use when the client is also a server (in the case of Presence Federation), or when you want to use a single certificate to represent all clients (in Diameter inspection).

b. In **Certificate**, select the LDC issuer you created in Create the Internal Local CA to Sign Local Dynamic Certificates for Phones (ASDM). For example, **ldc_server**. This trustpoint must include the **proxy-ldc-issuer** command. If you have not created this trustpoint yet, click **Manage** and create it now.

Alternatively, if you want to use the default local CA server (LOCAL-CA-SERVER), select **Certificate Authority Server**.

c. For **Key-Pair Name**, click **New** and create a new RSA general purpose key pair, size 1024. (Cisco UCM releases 10.5.2su3 and earlier allow a maximum key of 1024 bits.) Give the key a new name, for example, **phone_common**. When you click **Generate Now**, the key is generated and you are returned to the wizard with the new key pair selected.

d. (Optional.) You can define the security algorithms (ciphers) to use when the proxy acts as a client to the Cisco UCM. Move the allowed ciphers from the available algorithms to the active algorithms list. These ciphers replace the original ones from the hello message. If you do not specify ciphers, the default system ciphers are used. You can move the algorithms up or down to change their relative priority. You can select the null cipher if you are confident communication with the Call Manager server is secure (for example, the server is in the same data center as the ASA).

e. Click **Next**.

Step 7 The final step of the wizard explains the additional steps you must perform. Click **Finish**.

Enable TLS Proxy for SIP Inspection (ASDM)

Configure the required service policies to enable encrypted voice inspection for SIP.

Because secure protocols use different ports than the regular unencrypted versions, you need to configure unique classes for TLS proxy inspection.

- Secure SIP (SIPS) uses TCP/UDP 5061 (rather than SIP's 5060).

The inspection_default class filters on the unencrypted ports only.

The following procedure explains how to create these classes and add the TLS proxy inspections to the existing global_policy policy map. Alternatively, you can create service policies for specific interfaces.

For more detailed information on how to configure service policies, see the firewall configuration guide.

Before you begin

You can configure inspection policy maps to customize the inspection. If you do not want to use the default settings for the inspections, configure the inspection policy maps before configuring the service policy. For

details on customizing SIP, see the firewall configuration guide. The following procedure assumes you are using the default settings.

Step 1 Choose **Configuration** > **Firewall** > **Service Policy**.

Step 2 Create the service policy for secure SIP.

a. Click **Add** > **Add Service Policy Rule**.

b. Select **Global** and click **Next**.

c. Choose **Create a New Class** and enter a class name, for example, **sec_sip**.

d. Choose **Source and Destination IP Address (Uses ACL)**.

Because SIP endpoints can use TCP or UDP, you cannot create a simple port match for the class. Instead, create a service group for TCP/5061 and UDP/5061, then use that object in an ACL.

e. Click **Next** to define the ACL.

f. Select the following ACL options:

- **Action = Match**.
- **Source** = any
- **Destination** = any

g. For **Destination Service**, click the ... button to open the Browse Service dialog box.

h. Click **Add** > **Service Group** , and configure the following:

- **Group Name** = Something meaningful, such as **sec_sip_ports**.
- Select **Create New Member**, select **tcp-udp** for **Service Type**, enter 5061 for **Destination Port/Range**, then click **Add>>**.
- Click **OK** to save the object. This returns you to the Browse Service dialog box.

i. Double-click the object you just created to select it and click **OK**.

The Add Service Policy Rule - Traffic Match dialog box should now show your object selected in the Destination Service field for the ACL.

j. Click **Next**.

k. On the **Rule Actions** > **Protocol Inspection** tab, select **SIP** and click the associated **Configure** button.

l. Configure the following SIP inspection options:

- If you configured an inspection map to customize SIP inspection, choose **Select a SIP Inspection Map**, then select your map. Otherwise, leave **Use the Default SIP Inspection Map** selected.
- Select **Enable Encrypted Traffic Inspection**, then select the TLS proxy you configured for SIP inspection.

m. Click **OK** to save the SIP inspection options.

n. Click **Finish** to save the SIP inspection service policy.

Step 3 Click **Apply** to write the new policies to the device.

Verifying TLS Proxy Setup for a Phone

You can verify that encrypted voice inspection is working by logging into the CLI and setting up tls-proxy debugging.

You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems. For example, use the following commands to enable TLS proxy-related debug and syslog output only:

```
ciscoasa(config)# debug inspect tls-proxy events
ciscoasa(config)# debug inspect tls-proxy errors
ciscoasa(config)# logging enable
ciscoasa(config)# logging timestamp
ciscoasa(config)# logging list loglist message 711001
ciscoasa(config)# logging list loglist message 725001-725014
ciscoasa(config)# logging list loglist message 717001-717038
ciscoasa(config)# logging buffer-size 1000000
ciscoasa(config)# logging buffered loglist
ciscoasa(config)# logging debug-trace
```

The following is sample output reflecting a successful TLS proxy session setup for a SIP phone:

```
ciscoasa(config)# show log

Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with client
outside:133.9.0.218/49159 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Set up proxy for Client
outside:133.9.0.218/49159 <-> Server inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Using trust point 'local_ccm' with the
Client, RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Waiting for SSL handshake from Client
outside:133.9.0.218/49159.
Apr 17 2007 23:13:47: %ASA-7-725010: Device supports the following 4 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : RC4-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[3] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[4] : DES-CBC3-SHA
Apr 17 2007 23:13:47: %ASA-7-725008: SSL client outside:133.9.0.218/49159 proposes the
following 2 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725012: Device chooses cipher : AES128-SHA for the SSL
session with client outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-7-725014: SSL lib error. Function: SSL23_READ Reason: ssl
handshake failure
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 01, subject name: cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-7-717030: Found a suitable trustpoint
_internal_ejw-sv-2_cn=CAPF-08a91c01 to validate certificate.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. serial
number: 01, subject name: cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
warning, revocation status was not checked.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with client
outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with server
inside:195.168.2.201/5061 for TLSv1 session.
```

```
Apr 17 2007 23:13:47: %ASA-7-725009: Device proposes the following 2 cipher(s) to server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Generating LDC for client
'cn=SEP0017593F50A8', key-pair 'phone_common', issuer 'LOCAL-CA-SERVER', RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Started SSL handshake with Server
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Client
Apr 17 2007 23:13:47: %ASA-7-725013: SSL Server inside:195.168.2.201/5061 choose cipher :
AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 76022D3D9314743A, subject name: cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. Certificate
is resident and trusted, serial number: 76022D3D9314743A, subject name:
cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
revocation status check.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Server
```

Monitoring the TLS Proxy

Use the **show tls-proxy** commands with different options to check the active TLS proxy sessions. In ASDM, you can issue the commands through the **Tools** > **Command Line Interface** dialog box. The following are some sample outputs:

```
ciscoasa(config-tlsp)# show tls-proxy
Maximum number of sessions: 1200
TLS-Proxy 'sip_proxy': ref_cnt 1, seq# 3
Server proxy:
Trust-point: local_ccm
Client proxy:
Local dynamic certificate issuer: LOCAL-CA-SERVER
Local dynamic certificate key-pair: phone_common
Cipher suite: aes128-sha1 aes256-sha1
Run-time proxies:
Proxy 0xcbae1538: Class-map: sip_ssl, Inspect: sip
Active sess 1, most sess 3, byte 3456043
ciscoasa(config-tlsp)# show tls-proxy session count
2 in use, 4 most used
ciscoasa(config-tlsp)# show tls-proxy session
2 in use, 4 most used
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786
ciscoasa(config-tlsp)# show tls-proxy session detail
2 in use, 4 most used
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786
Client: State SSLOK Cipher AES128-SHA Ch 0xca55e398 TxQSize 0 LastTxLeft 0 Flags 0x1
Server: State SSLOK Cipher AES128-SHA Ch 0xca55e378 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
Status: Available
Certificate Serial Number: 2b
Certificate Usage: General Purpose
Public Key Type: RSA (1024 bits)
Issuer Name:
cn=F1-ASA.default.domain.invalid
Subject Name:
```

```
cn=SEP0017593F50A8
Validity Date:
start date: 23:13:47 PDT Apr 16 2007
end date: 23:13:47 PDT Apr 15 2008
Associated Trustpoints:
```

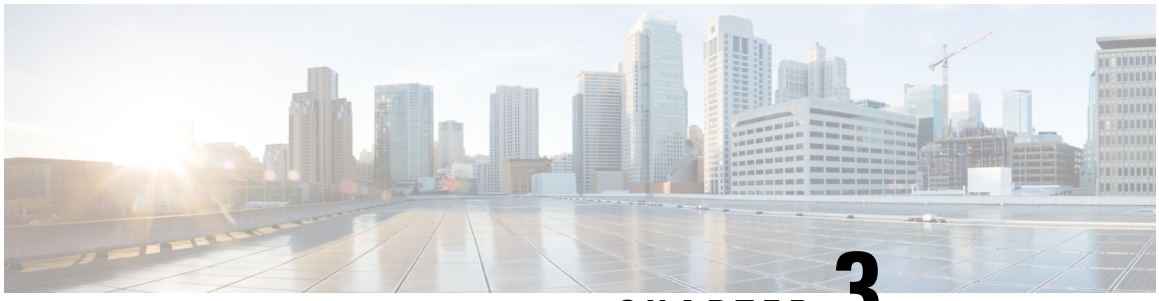
Feature History for the TLS Proxy for Encrypted Voice Inspection

The following table lists the release history for this feature.

Table 1: Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
TLS Proxy	8.0(2)	The TLS proxy feature was introduced.
SIP, SCCP, and TLS Proxy support for IPv6	9.3(1)	You can now inspect IPv6 traffic when using SIP, SCCP, and TLS Proxy (using SIP or SCCP). We did not modify any commands. We did not modify any ASDM screens.
Support for Cisco Unified Communications Manager 8.6	9.3(1)	The ASA now interoperates with Cisco Unified Communications Manager Version 8.6 (including SCCPv21 support). We did not modify any commands. We did not modify any ASDM screens.
SIP support for Trust Verification Services, NAT66, CUCM 10.5, and model 8831 phones.	9.3(2)	You can now configure Trust Verification Services servers in SIP inspection. You can also use NAT66. SIP inspection has been tested with CUCM 10.5. We added the **trust-verification-server** parameter command. (ASDM) We added Trust Verification Services Server support to the SIP inspection policy map.

Feature Name	Releases	Feature Information
Support for TLSv1.2 in TLS proxy and Cisco Unified Communications Manager 10.5.2.	9.7(1)	You can now use TLSv1.2 with TLS proxy for encrypted SIP or SCCP inspection with the Cisco Unified Communications Manager 10.5.2. The TLS proxy supports the additional TLSv1.2 cipher suites added as part of the **client cipher-suite** command. We modified the following commands: **client cipher-suite**. (ASDM) We did not modify any screens.
Support for setting the TLS proxy server SSL cipher suite.	9.8(1)	You can now set the SSL cipher suite when the ASA acts as a TLS proxy server. Formerly, you could only set global settings for the ASA using the ssl-cipher command. In ASDM, this is on the **Configuration** > **Device Management** > **Advanced** > **SSL Settings** > **Encryption** page We added the following command: **server cipher-suite**. (ASDM) We modified the following pages: **Configuration** > **Firewall** > **Unified Communications** > **TLS Proxy Add/Edit** dialog boxes, **Server Configuration** page.
TLS proxy deprecated for SCCP (Skinny) inspection.	9.13(1)	The **tls-proxy** keyword, and support for SCCP/Skinny encrypted inspection, was deprecated. The keyword will be removed from the **inspect skinny** command in a future release.



CHAPTER 3

ASA and Cisco Unified Presence

This chapter describes how to configure the ASA for Cisco Unified Presence.

Information About Cisco Unified Presence

This section includes the following topics:

Architecture for Cisco Unified Presence for SIP Federation Deployments

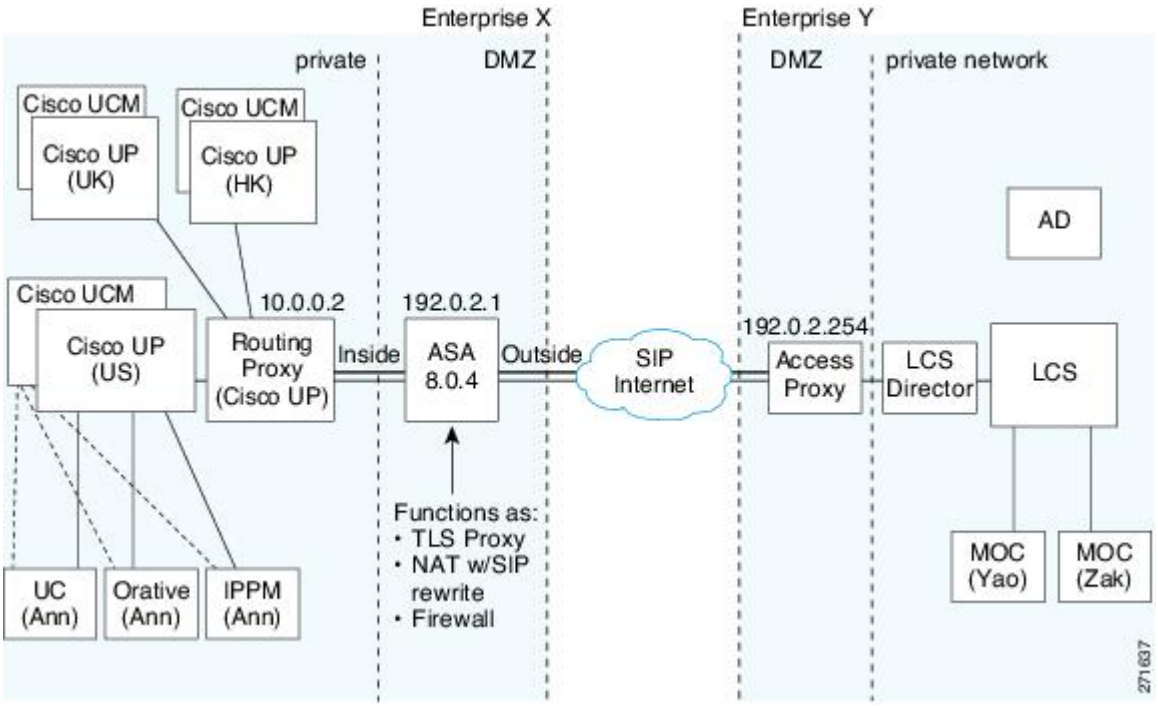
The following figure depicts a Cisco Unified Presence/LCS Federation scenario with the ASA as the presence federation proxy (implemented as a TLS proxy). The two entities with a TLS connection are the “Routing Proxy” (a dedicated Cisco UP) in Enterprise X and the Microsoft Access Proxy in Enterprise Y. However, the deployment is not limited to this scenario. Any Cisco UP or Cisco UP cluster could be deployed on the left side of the ASA; the remote entity could be any server (an LCS, an OCS, or another Cisco UP).

The following architecture is generic for two servers using SIP (or other ASA inspected protocols) with a TLS connection.

Entity X: Cisco UP/Routing Proxy in Enterprise X

Entity Y: Microsoft Access Proxy/Edge server for LCS/OCS in Enterprise Y

Figure 2: Typical Cisco Unified Presence/LCS Federation Scenario



In the above architecture, the ASA functions as a firewall, NAT, and TLS proxy, which is the recommended architecture. However, the ASA can also function as NAT and the TLS proxy alone, working with an existing firewall.

Either server can initiate the TLS handshake (unlike IP Telephony or Cisco Unified Mobility, where only the clients initiate the TLS handshake). There are by-directional TLS proxy rules and configuration. Each enterprise can have an ASA as the TLS proxy.

Note The Cisco UP server listens to port 5062 by default, whereas AOL and OCS listen to port 5061. If you use the defaults, you must use static NAT to translate 5061 on the outside to 5062 on the inside. However, you can configure peer auth on Cisco UP to listen to 5061, in which case you do not need to translate 5062. Changing the Cisco UP port is the best solution, and examples assume you reconfigure the port to 5061.

In the above figure, NAT or PAT can be used to hide the private address of Entity X. In this situation, static NAT or PAT must be configured for foreign server (Entity Y) initiated connections or the TLS handshake (inbound). Typically, the public port should be 5061. The following static PAT command is required for the Cisco UP that accepts inbound connections:

```
ciscoasa(config)# object network obj-10.0.0.2-01
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
5061
```

The following static PAT must be configured for each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server.

For Cisco UP with the address 10.0.0.2, enter the following command:

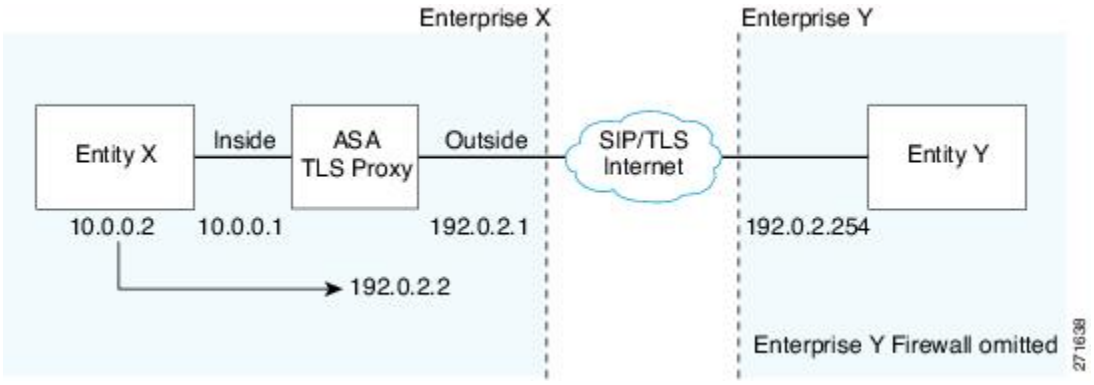
```
ciscoasa(config)# object network obj-10.0.0.2-03
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
ciscoasa(config)# object network obj-10.0.0.2-04
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
5060
For another Cisco UP with the address 10.0.0.3, you must use a different set of PAT ports,
 such as 45061
or 45070:
ciscoasa(config)# object network obj-10.0.0.3-01
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
45061
ciscoasa(config)# object network obj-10.0.0.3-03
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
ciscoasa(config)# object network obj-10.0.0.2-03
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5070
45070
ciscoasa(config)# object network obj-10.0.0.3-04
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
45060
```

Dynamic NAT or PAT can be used for the rest of the outbound connections or the TLS handshake. The ASA SIP inspection engine takes care of the necessary translation.

```
ciscoasa(config)# object network obj-0.0.0.0-01
ciscoasa(config-network-object)# subnet 0.0.0.0 0.0.0.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic 192.0.2.1
```

The following figure illustrates an abstracted scenario with Entity X connected to Entity Y through the presence federation proxy on the ASA. The proxy is in the same administrative domain as Entity X. Entity Y could have another ASA as the proxy but this is omitted for simplicity.

Figure 3: Abstracted Presence Federation Proxy Scenario between Two Server Entities



For the Entity X domain name to be resolved correctly when the ASA holds its credential, the ASA could be configured to perform NAT for Entity X, and the domain name is resolved as the Entity X public address for which the ASA provides proxy service.

For further information about configuring Cisco Unified Presence Federation for SIP Federation, see the Integration Guide for Configuring Cisco Unified Presence for Interdomain Federation.: http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

Trust Relationship in the Presence Federation

Within an enterprise, setting up a trust relationship is achievable by using self-signed certificates or you can set it up on an internal CA.

Establishing a trust relationship cross enterprises or across administrative domains is key for federation. Cross enterprises you must use a trusted third-party CA (such as, VeriSign). The ASA obtains a certificate with the FQDN of the Cisco UP (certificate impersonation).

For the TLS handshake, the two entities could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. Both entities enroll with the CAs. The ASA as the TLS proxy must be trusted by both entities. The ASA is always associated with one of the enterprises. Within that enterprise (Enterprise X), the entity and the ASA could authenticate each other via a local CA, or by using self-signed certificates.

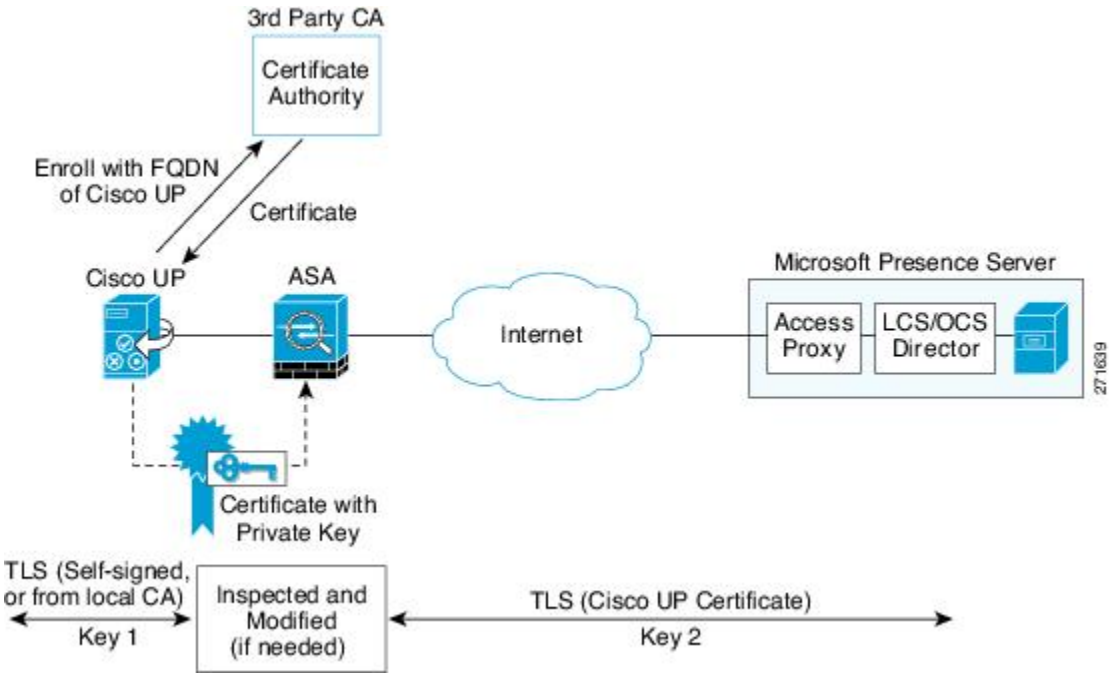
Note Ensure that the required DNS SRV records are created for verifying the FQDN in the certificate. You can verify the presence of an SRV record using the nslookup command, setting the type to srv. The SRV hostname should be correct.

To establish a trusted relationship between the ASA and the remote entity (Entity Y), the ASA can enroll with the CA on behalf of Entity X (Cisco UP). In the enrollment request, the Entity X identity (domain name) is used.

The following figure shows the way to establish the trust relationship. The ASA enrolls with the third party CA by using the Cisco UP FQDN as if the ASA is the Cisco UP.

Note The ASA generates the CSR needed for enrolling in the third party CA, not the Cisco UP server. You also need to import the ASA self-signed certificate into the Cisco UP server. In addition, you need to import the Entity Y certificate into the ASA.

Figure 4: How the Security Appliance Represents Cisco Unified Presence – Certificate Impersonate



Security Certificate Exchange Between Cisco UP and the Security Appliance

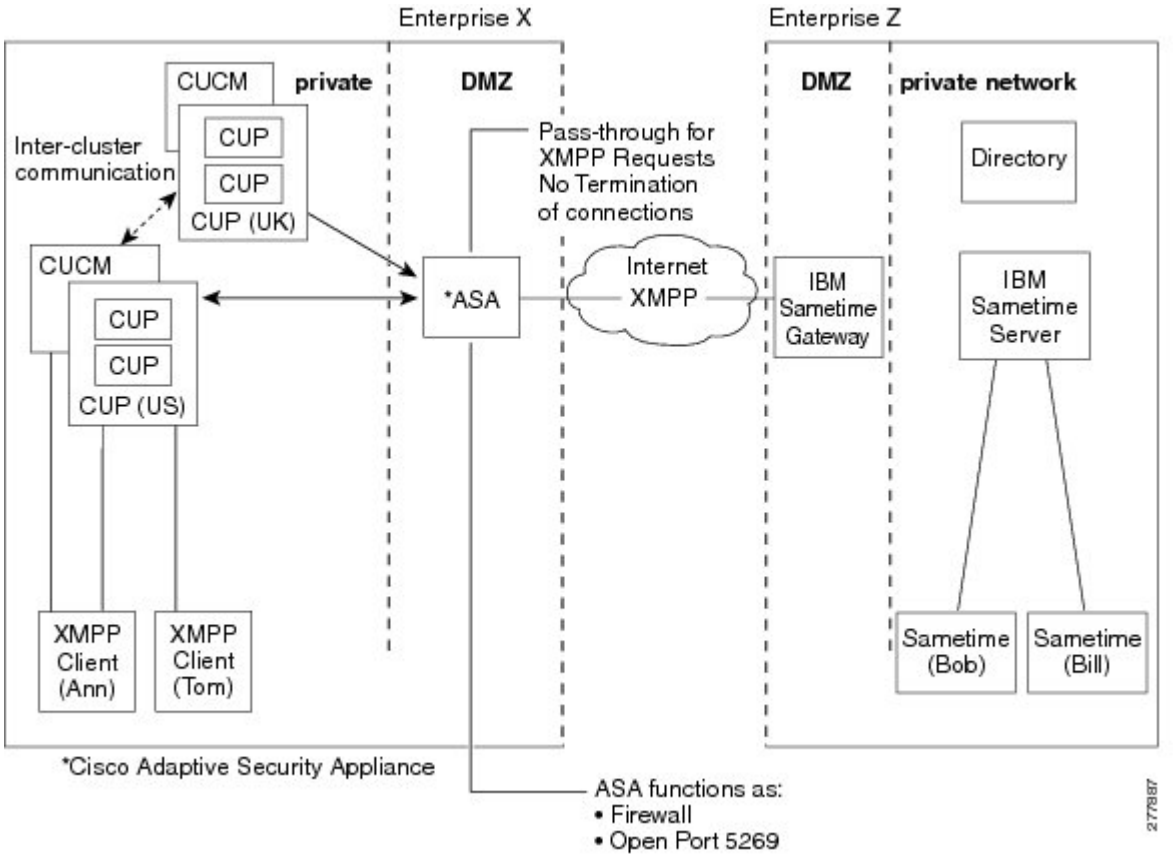
You need to generate the keypair for the certificate (such as `cup_proxy_key`) used by the ASA, and configure a trustpoint to identify the self-signed certificate sent by the ASA to Cisco UP (such as cup_proxy) in the TLS handshake.

For the ASA to trust the Cisco UP certificate, you need to create a trustpoint to identify the certificate from the Cisco UP (such as `cert_from_cup`), and specify the enrollment type as terminal to indicate that you will paste the certificate received from the Cisco UP into the terminal.

XMPP Federation Deployments

The following figure provides an example of an XMPP federated network between Cisco Unified Presence enterprise deployment and an IBM Sametime enterprise deployment. TLS is optional for XMPP federation. ASA acts only as a firewall for XMPP federation; it does not provide TLS proxy functionality or PAT for XMPP federation.

Figure 5: Basic XMPP Federated Network between Cisco Unified Presence and IBM Sametime



There are two DNS servers within the internal Cisco Unified Presence enterprise deployment. One DNS server hosts the Cisco Unified Presence private address. The other DNS server hosts the Cisco Unified Presence public address and a DNS SRV records for SIP federation (_sipfederationtle), and XMPP federation (_xmpp-server) with Cisco Unified Presence. The DNS server that hosts the Cisco Unified Presence public address is located in the local DMZ.

For further information about configuring Cisco Unified Presence Federation for XMPP Federation, see: *the Integration Guide for Configuring Cisco Unified Presence Release 8.0 for Interdomain Federation*: http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

Configuration Requirements for XMPP Federation

For XMPP Federation, ASA acts as a firewall only. You must open port 5269 for both incoming and outgoing XMPP federated traffic on ASA.

These are sample ACLs to open port 5269 on ASA.

Allow traffic from any address to any address on port 5269:

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

Allow traffic from any address to any single node on port 5269:

```
access-list ALLOW-ALL extended permit tcp any host <private cup IP address> eq 5269
```

If you do not configure the ACL above, and you publish additional XMPP federation nodes in DNS, you must configure access to each of these nodes, for example:

```
object network obj_host_<private cup ip address>
#host <private cup ip address>
object network obj_host_<private cup2 ip address>
#host <private cup2 ip address>
object network obj_host_<public cup ip address>
#host <public cup ip address>
....
```

Configure the following NAT commands:

```
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

If you publish a single public IP address in DNS, and use arbitrary ports, configure the following:

(This example is for two additional XMPP federation nodes)

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

If you publish multiple public IP addresses in DNS all using port 5269, configure the following:

(This example is for two additional XMPP federation nodes)

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup3 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Configuring Cisco Unified Presence Proxy for SIP Federation (CLI)

Task Flow for Configuring Cisco Unified Presence Federation Proxy for SIP Federation

To configure a Cisco Unified Presence/LCS Federation scenario with the ASA as the TLS proxy where there is a single Cisco UP that is in the local domain and self-signed certificates are used between the Cisco UP and the ASAm, perform the following tasks.

Step 1 Create the following static NAT for the local domain containing the Cisco UP.

For the inbound connection to the local domain containing the Cisco UP, create static PAT by entering the following command:

```
hostname(config)# object network name
hostname(config-network-object)# host real_ip
hostname(config-network-object)# nat (real_ifc,mapped_ifc) static mapped_ip service {tcp |
udp} real_port mapped_port
```

Note For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT by using a different set of PAT ports.

For outbound connections or the TLS handshake, use dynamic NAT or PAT. The ASA SIP inspection engine takes care of the necessary translation (fixup).

```
hostname(config)# object network name
hostname(config-network-object)# subnet real_ip netmask
hostname(config-network-object)# nat (real_ifc,mapped_ifc) dynamic mapped_ip
```

Step 2 Create the necessary RSA keypairs and proxy certificate, which is a self-signed certificate, for the remote entity. See Creating Trustpoints and Generating Certificates.

Step 3 Install the certificates. See Installing Certificates.

Step 4 Create the TLS proxy instance for the Cisco UP clients connecting to the Cisco UP server. See Creating the TLS Proxy Instance.

Step 5 Enable the TLS proxy for SIP inspection. See Enabling the TLS Proxy for SIP Inspection.

Creating Trustpoints and Generating Certificates

You need to generate the keypair for the certificate (such as cup_proxy_key) used by the ASA, and configure a trustpoint to identify the self-signed certificate sent by the ASA to Cisco UP (such as cup_proxy) in the TLS handshake.

Procedure

	Command or Action	Purpose

Step 1	`hostname(config)# crypto key generate rsa label key-pair-label modulus size`
	Example:
	`crypto key generate rsa label ent_y_proxy_key modulus 1024`
	`INFO: The name for the keys will be: ent_y_proxy_key`
	`Keypair generation process begin. Please wait...`
	`hostname(config)#`
		Creates the RSA keypair that can be used for the trustpoints.
		The keypair is used by the self-signed certificate presented to the local domain containing the Cisco UP (proxy for the remote entity).

Step 2	`hostname(config)# crypto ca trustpoint trustpoint_name`
	Example:
	`hostname(config)# crypto ca trustpoint ent_y_proxy`
		Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the remote entity.
		A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.

Step 3	`hostname(config-ca-trustpoint)# enrollment self`	Generates a self-signed certificate.
Step 4	`hostname(config-ca-trustpoint)# fqdn none`	Specifies not to include a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.

Step 5	`hostname(config-ca-trustpoint)# subject-name X.500_name`
	Example:
	`hostname(config-ca-trustpoint)# subject-name cn=Ent-Y-Proxy`
		Includes the indicated subject DN in the certificate during enrollment

Step 6	`hostname(config-ca-trustpoint)# keypair keyname`
	Example:
	`hostname(config-ca-trustpoint)# keypair ent_y_proxy_key`
		Specifies the key pair whose public key is to be certified.

Step 7	`hostname(config-ca-trustpoint)# exit`	Exits from the CA Trustpoint configuration mode.

Step 8	`hostname(config)# crypto ca enroll trustpoint`
	Example:
	`hostname(config)# crypto ca enroll ent_y_proxy`
		Starts the enrollment process with the CA and specifies the name of the trustpoint to enroll with.

What to do next

Install the certificate on the local entity truststore. You could also enroll the certificate with a local CA trusted by the local entity. See Installing Certificates.

Installing Certificates

Export the self-signed certificate for the ASA created in the Creating Trustpoints and Generating Certificates and install it as a trusted certificate on the local entity. This task is necessary for local entity to authenticate the ASA.

Before you begin

To create a proxy certificate on the ASA that is trusted by the remote entity, obtain a certificate from a trusted CA. For information about obtaining a certificate from a trusted CA, see the general operations configuration guide.

Procedure

	Command or Action	Purpose

Step 1	`hostname(config)# crypto ca export trustpoint identity-certificate`
	Example:
	`hostname(config)# crypto ca export ent_y_proxy identity-certificate`
		Export the ASA self-signed (identity) certificate.

Step 2	`hostname(config)# crypto ca trustpoint trustpoint_name`
	Example:
	`hostname(config)# crypto ca trustpoint ent_x_cert`
	`! for Entity X's self-signed certificate`
		Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the local entity.
		A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.

Step 3	`hostname(config-ca-trustpoint)# enrollment terminal`	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment). If the local entity uses a self-signed certificate, the self-signed certificate must be installed; if the local entity uses a CA-issued certificate, the CA certificate needs to be installed. This configuration shows the commands for using a self-signedcertificate.
Step 4	`hostname(config-ca-trustpoint)# exit`	Exits from the CA Trustpoint configuration mode.

Step 5	`hostname(config)# crypto ca authenticate trustpoint`
	Example:
	`hostname(config)# crypto ca authenticate ent_x_cert`
	`Enter the base 64 encoded CA certificate.`
	`End with a blank line or the word "quit" on a line by itself`
	`[ certificate data omitted ]`
	`Certificate has the following attributes:`
	`Fingerprint: 21B598D5 4A81F3E5 0B24D12E 3F89C2E4`
	`% Do you accept this certificate? [yes/no]: yes`
	`Trustpoint CA certificate accepted.`
	`% Certificate successfully imported`
		Installs and authenticates the CA certificates associated with a trustpoint created for the local entity.
		Where *trustpoint* specifies the trustpoint from which to obtain the CA certificate. Maximum name length is 128 characters.
		The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.

	Command or Action	Purpose

Step 6	`hostname(config)# crypto ca trustpoint trustpoint_name`
	Example:
	`hostname(config)# crypto ca trustpoint ent_y_ca`
	`! for Entity Y's CA certificate`
		Install the CA certificate that signs the remote entity certificate on the ASA by entering the following commands. This step is necessary for the ASA to authenticate the remote entity.

Step 7	`hostname(config-ca-trustpoint)# enrollment terminal`	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment).
Step 8	`hostname(config-ca-trustpoint)# exit`	Exits from the CA Trustpoint configuration mode.

Step 9	`hostname(config)# crypto ca authenticate trustpoint trustpoint`
	Example:
	`hostname(config)# crypto ca authenticate ent_y_ca`
	`Enter the base 64 encoded CA certificate.`
	`End with a blank line or the word "quit" on a line by itself`
	`MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG 9w0BAQUFADCB`
	`[ certificate data omitted ]`
	`/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==`
		Installs and authenticates the CA certificates associated with a trustpoint created for the local entity.
		The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.

What to do next

Once you have created the trustpoints and installed the certificates for the local and remote entities on the ASA, create the TLS proxy instance. See Creating the TLS Proxy Instance.

Creating the TLS Proxy Instance

Because either server can initiate the TLS handshake (unlike IP Telephony or Cisco Unified Mobility, where only the clients initiate the TLS handshake), you must configure by-directional TLS proxy rules. Each enterprise can have an ASA as the TLS proxy.

Create TLS proxy instances for the local and remote entity initiated connections respectively. The entity that initiates the TLS connection is in the role of "TLS client". Because the TLS proxy has a strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

Procedure

	Command or Action	Purpose

Step 1	`! Local entity to remote entity hostname(config)# tls-proxy proxy_name`
	Example:
	`hostname(config)# tls-proxy ent_x_to_y`
		Creates the TLS proxy instance.

Step 2	`hostname(config-tlsp)# server trust-point proxy_name`	Specifies the proxy trustpoint certificate presented during TLS handshake.

	Command or Action	Purpose
	Example: `hostname(config-tlsp)# server trust-point ent_y_proxy`	The certificate must be owned by the ASA (identity certificate). Where the *proxy_name* for the **server trust-point** command is the remote entity proxy name.

Step 3	`hostname(config-tlsp)# client trust-point proxy_trustpoint`
	Example:
	`hostname(config-tlsp)# client trust-point ent_x_cert`
		Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client.
		The certificate must be owned by the ASA (identity certificate).
		Where the *proxy_trustpoint* for the **client trust-point** command is the local entity proxy.

Step 4	`hostname(config-tlsp)# client cipher-suite cipher_suite`
	Example:
	`hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1`
		Specifies cipher suite configuration.
		For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite.

Step 5	`! Remote entity to local entity hostname(config)# tls-proxy proxy_name`
	Example:
	`tls-proxy ent_y_to_x`
		Creates the TLS proxy instance.

Step 6	`hostname(config-tlsp)# server trust-point proxy_name`
	Example:
	`hostname(config-tlsp)# server trust-point ent_x_cert`
		Specifies the proxy trustpoint certificate presented during TLS handshake.
		Where the *proxy_name* for the **server trust-point** command is the local entity proxy name

Step 7	`hostname(config-tlsp)# client trust-point proxy_trustpoint`
	Example:
	`hostname(config-tlsp)# client trust-point ent_y_proxy`
		Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client.
		Where the *proxy_trustpoint* for the **client trust-point** command is the remote entity proxy.

Step 8	`hostname(config-tlsp)# client cipher-suite cipher_suite`
	Example:
	`hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1`
		Specifies cipher suite configuration.

What to do next

Once you have created the TLS proxy instance, enable it for SIP inspection. See Enabling the TLS Proxy for SIP Inspection.

Enabling the TLS Proxy for SIP Inspection

Enable the TLS proxy for SIP inspection and define policies for both entities that could initiate the connection.

Procedure

	Command or Action	Purpose

Step 1	`hostname(config)# access-list id extended permit tcp host src_ip host dest_ip eq port`
	Example:
	`access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061`
	`access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061`
		Adds an Access Control Entry. The ACL is used to specify the class of traffic to inspect.

Step 2	`hostname(config)# class-map class_map_name`
	Example:
	`hostname(config)# class-map ent_x_to_y`
		Configures the secure SIP class of traffic to inspect.
		Where *class_map_name* is the name of the SIP class map.

Step 3	`hostname(config-cmap)# match access-list access_list_name`
	Example:
	`hostname(config-cmap)# match access-list ent_x_to_y`
		Identifies the traffic to inspect.

Step 4	`hostname(config-cmap)# exit`	Exits from Class Map configuration mode.

Step 5	`hostname(config)# policy-map type inspect sip policy_map_name`
	Example:
	`hostname(config)# policy-map type inspect sip sip_inspect`
		Defines special actions for SIP inspection application traffic.

Step 6	`hostname(config-pmap)# parameters ! SIP inspection parameters`	Specifies the parameters for SIP inspection. Parameters affect the behavior of the inspection engine. The commands available in parameters configuration mode depend on the application.
Step 7	`hostname(config-pmap)# exit`	Exits from Policy Map configuration mode.

Step 8	`hostname(config)# policy-map name`
	Example:
	`hostname(config)# policy-map global_policy`
		Configure the policy map and attach the action to the class of traffic.

Step 9	`hostname(config-pmap)# class classmap_name`
	Example:
	`hostname(config-pmap)# class ent_x_to_ylicy`
		Assigns a class map to the policy map so that you can assign actions to the class map traffic.
		Where *classmap_name* is the name of the SIP class map.

	Command or Action	Purpose
Step 10	`hostname(config-pmap)# inspect sip sip_map tls-proxy proxy_name hostname(config-pmap)# inspect sip sip_inspect tls-proxy ent_x_to_y`	Enables TLS proxy for the specified SIP inspection session.
Step 11	`hostname(config-pmap)# exit`	Exits from Policy Map configuration mode.

Step 12	`hostname(config)# service-policy policy_map_name global`
	Example:
	`hostname(config)# service-policy global_policy global`
		Enables the service policy for SIP inspection for all interfaces.
		Where name for the policy-map command is the name of the global policy map.

Configuring Cisco Unified Presence Proxy for SIP Federation (ASDM)

To configure the Cisco Unified Presence proxy by using ASDM, choose **Wizards** > **Unified Communications Wizard** from the menu. From the first page, select the Cisco Unified Presence Proxy option under the Business-to-Business section.

When using the wizard to create the Cisco Presence Federation proxy, ASDM automatically creates the necessary TLS proxies, enables SIP inspection for the Presence Federation traffic, generates address translation (static PAT) statements for the local Cisco Unified Presence server, and creates ACLs to allow traffic between the local Cisco Unified Presence server and remote servers.

The wizard guides you through four steps to create the Presence Federation Proxy:

Step 1 Specify settings to define the private and public network topology, such the private and public IP address of the Presence Federation server. See Configuring the Topology for the Cisco Presence Federation Proxy.

Step 2 Configure the local-side certificate management, namely the certificates that are exchanged between the local Unified Presence Federation server and the ASA. See Configuring the Local-Side Certificates for the Cisco Presence Federation Proxy.

Step 3 Configure the remote-side certificate management, namely the certificates that are exchanged between the remote server and the ASA. See Configuring the Remote-Side Certificates for the Cisco Presence Federation Proxy.

The wizard completes by displaying a summary of the configuration created for the Presence Federation proxy.

Configuring the Topology for the Cisco Presence Federation Proxy

When configuring the Presence Federation Proxy, you specify settings to define the private and public network topology, such the private and public network interfaces, and the private and public IP addresses of the Cisco Unified Presence server.

The values that you specify in this page generate the following configuration settings for the Presence Federation Proxy:

- Static PAT for the local Cisco Unified Presence server
- ACLs for traffic between the local Cisco Unified Presence server and remote servers

Step 1 In the Private Network area, choose the interface from the drop-down list.

Step 2 In the Unified Presence Server area, enter the private and public IP address for the Unified Presence server. Entering ports for these IP addresses is optional. By default port number 5061 is entered, which is the default TCP port for SIP inspection.

Step 3 In the FQDN field, enter the domain name for the Unified Presence server. This domain name is included in the certificate signing request that you generate later in this wizard.

Step 4 In the Public Network area, choose the interface of the public network from the drop-down list. The proxy uses this interface for configuring static PAT for the local Cisco Unified Presence server and for configuring ACLs to allow remote servers to access the Cisco Unified Presence server.

Step 5 Click **Next**.

Configuring the Local-Side Certificates for the Cisco Presence Federation Proxy

Within an enterprise, setting up a trust relationship is achievable by using self-signed certificates. The supports using self-signed certificates only at this step.

Step 1 In the ASA's Identity Certificate area, click **Generate and Export ASA's Identity Certificate**.

An information dialog box appears indicating that enrollment succeeded. In the Enrollment Status dialog box, click **OK**. The Export certificate dialog box appears.

Note
- If an identity certificate for the ASA has already been created, the button in this area appears as **Export ASA's Identity Certificate** and the Export certificate dialog box immediately appears.
- When using the wizard to configure the Cisco Presence Federation proxy, the wizard only supports installing self-signed certificates.

Step 2 Export the identity certificate generated by the wizard for the ASA.

You must install this certificate into the Cisco Presence Federation server.

Step 3 Local Unified Presence Server's Certificate area, click **Install Server's Certificate**. The Install Certificate dialog appears.

Step 4 Locate the file containing the Cisco Unified Presence server certificate or paste the certificate details in the dialog box.

See the Cisco Unified Presence server documentation for information on how to export the certificate for this server.

Step 5 Click **Next**.

Configuring the Remote-Side Certificates for the Cisco Presence Federation Proxy

Establishing a trust relationship across enterprises or across administrative domains is key for federation. Across enterprises you must use a trusted third-party CA (such as, VeriSign). The security appliance obtains a certificate with the FQDN of the Cisco Unified Presence server (certificate impersonation).

For the TLS handshake, the two entities, namely the local entity and a remote entity, could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. The local entity and the remote entity enroll with the CAs. The ASA as the TLS proxy must be trusted by both the local and remote entities. The security appliance is always associated with one of the enterprises. Within that enterprise, the entity and the security appliance authenticate each other by using a self-signed certificate.

To establish a trusted relationship between the security appliance and the remote entity, the security appliance can enroll with the CA on behalf of the Cisco Unified Presence server for the local entity. In the enrollment request, the local entity identity (domain name) is used.

To establish the trust relationship, the security appliance enrolls with the third party CA by using the Cisco Unified Presence server FQDN as if the security appliance is the Cisco Unified Presence server.

Step 1 In the ASA's Identity Certificate area, click **Generate CSR**. The CSR parameters dialog box appears. If the ASA already has a signed identity certificate, you can skip this step.

This certificate is presented to remote Presence Federation servers. When configuring the certificate:

- Choose a key size that provides sufficient security. Your CA might have a minimum key size requirement.
- The wizard provides the common name (CN), which is the FQDN of the Cisco Unified Presence server.
- Add additional DNs as appropriate.

Information dialog boxes appear indicating that the wizard is delivering the settings to the ASA and retrieving the certificate key pair information. The Identity Certificate Request dialog box appears. Save the certificate to a file and submit it to the CA for signing.

Step 2 Click Install **ASA's Identity Certificate**. See Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers.

Step 3 Click **Remote Server's CA's Certificate**. The Install Certificate dialog box appears. Select the certificate file and install it.

Note You must install a root CA certificate for each remote entity that communicates with the ASA because different organizations might be using different CAs.

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for the Presence Federation proxy.

Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers

When configuring certificates for the Cisco Presence Federation Proxy and Cisco Intercompany Media Engine Proxy, you must install the ASA identity certificate and the root certificate on the Cisco Presence Federation server and Cisco Intercompany Media Engine server, respectively.

Typically, a certificate authority returns two certificates: your signed identity certificate and the certificate authority's certificate (referred to as the root certificate). The root certificate from the certificate authority is used to sign other certificates. The root certificate is used by the ASA to authenticate your signed identity certificate received from the certificate authority.

Step 1 In the Root CA's Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 2 In the ASA's Identity Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format radio** button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 3 Click **Install Certificate**.

Monitoring Cisco Unified Presence

Debugging is similar to debugging TLS proxy for IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, use the following commands to enable TLS proxy-related debug and syslog output only:

```
hostname(config)# debug inspect tls-proxy events
hostname(config)# debug inspect tls-proxy errors
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see Monitoring the TLS Proxy.

Enable the **debug sip** command for SIP inspection engine debugging. See the command reference.

Additionally, you can capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Configuration Example for Cisco Unified Presence

This section contains the following topics:

Example Configuration for SIP Federation Deployments

The following sample illustrates the necessary configuration for the ASA to perform TLS proxy for Cisco Unified Presence as shown in the following figure. It is assumed that a single Cisco UP (Entity X) is in the local domain and self-signed certificates are used between Entity X and the ASA.

For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT and if you have another Cisco UP with the address (10.0.0.3 in this sample), it must use a different set of PAT ports (such as 45061 or 45070). Dynamic NAT or PAT can be used for outbound connections or TLS handshake. The ASA SIP inspection engine takes care of the necessary translation (fixup).

When you create the necessary RSA key pairs, a key pair is used by the self-signed certificate presented to Entity X (proxy for Entity Y). When you create a proxy certificate for Entity Y, the certificate is installed on the Entity X truststore. It could also be enrolled with a local CA trusted by Entity X.

Exporting the ASA self-signed certificate (ent_y_proxy) and installing it as a trusted certificate on Entity X is necessary for Entity X to authenticate the ASA. Exporting the Entity X certificate and installing it on the ASA is needed for the ASA to authenticate Entity X during handshake with X. If Entity X uses a self-signed certificate, the self-signed certificate must be installed; if Entity X uses a CA issued the certificate, the CA's certificated needs to be installed.

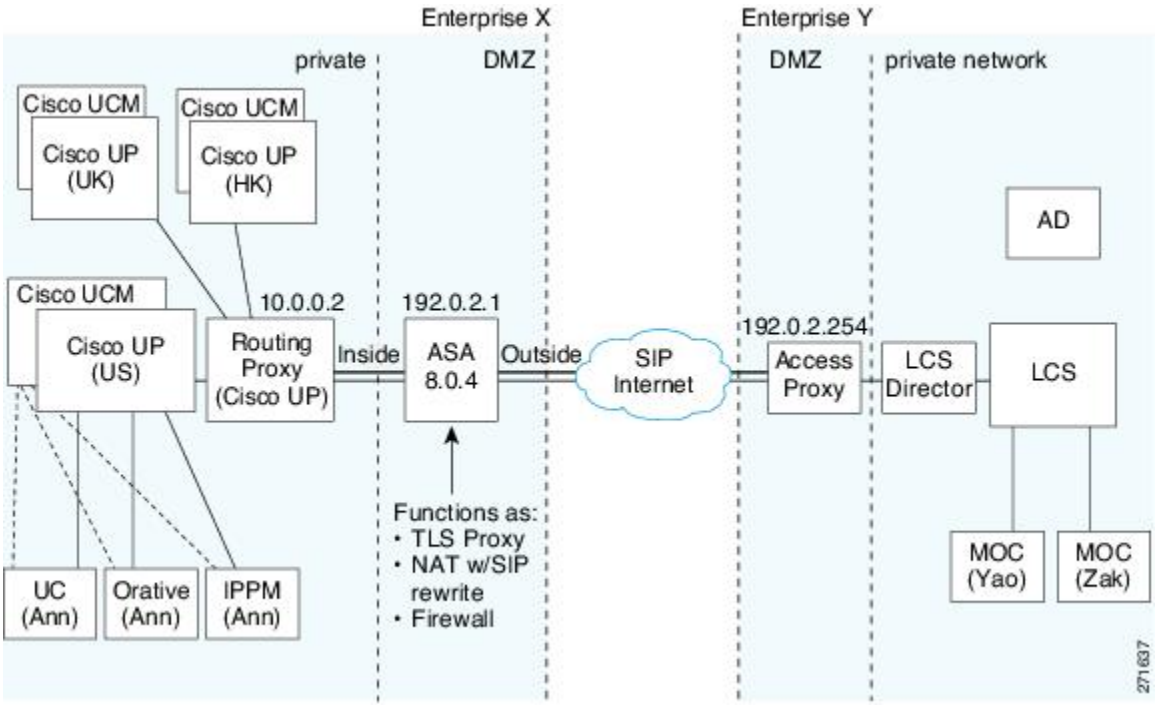
For about obtaining a certificate from a trusted CA, see the general operations configuration guide.

Installing the CA certificate that signs the Entity Y certificate on the ASA is necessary for the ASA to authenticate Entity Y.

When creating TLS proxy instances for Entity X and Entity Y, the entity that initiates the TLS connection is in the role of "TLS client". Because the TLS proxy has strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

When enabling the TLS proxy for SIP inspection, policies must be defined for both entities that could initiate the connection.

Figure 6: Typical Cisco Unified Presence/LCS Federation Scenario




```
object network obj-10.0.0.2-01
host 10.0.0.2
nat (inside,outside) static 192.0.2.1 service tcp 5061 5061
object network obj-10.0.0.2-02
host 10.0.0.2
nat (inside,outside) static 192.0.2.1 service tcp 5062 5062
object network obj-10.0.0.2-03
host 10.0.0.2
nat (inside,outside) static 192.0.2.1 service udp 5070 5070
object network obj-10.0.0.3-01
host 10.0.0.3
nat (inside,outside) static 192.0.2.1 service tcp 5062 45062
object network obj-10.0.0.3-02
host 10.0.0.3
nat (inside,outside) static 192.0.2.1 service udp 5070 45070
object network obj-0.0.0.0-01
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) dynamic 192.0.2.1
crypto key generate rsa label ent_y_proxy_key modulus 1024
! for self-signed Entity Y proxy certificate
crypto ca trustpoint ent_y_proxy
enrollment self
fqdn none
subject-name cn=Ent-Y-Proxy
keypair ent_y_proxy_key
crypto ca enroll ent_y_proxy
crypto ca export ent_y_proxy identity-certificate
! for Entity X's self-signed certificate
crypto ca trustpoint ent_x_cert
enrollment terminal
crypto ca authenticate ent_x_cert
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
[ certificate data omitted ]
```

```
quit
! for Entity Y's CA certificate
crypto ca trustpoint ent_y_ca
enrollment terminal
crypto ca authenticate ent_y_ca
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
! Entity X to Entity Y
tls-proxy ent_x_to_y
server trust-point ent_y_proxy
client trust-point ent_x_cert
client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
! Entity Y to Entity X
tls-proxy ent_y_to_x
server trust-point ent_x_cert
client trust-point ent_y_proxy
client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061
access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061
class-map ent_x_to_y
match access-list ent_x_to_y
class-map ent_y_to_x
match access-list ent_y_to_x
policy-map type inspect sip sip_inspect
parameters
! SIP inspection parameters
policy-map global_policy
class ent_x_to_y
inspect sip sip_inspect tls-proxy ent_x_to_y
class ent_y_to_x
inspect sip sip_inspect tls-proxy ent_y_to_x
service-policy global_policy global
```

Example ACL Configuration for XMPP Federation

Example 1: This example ACL configuration allows from any address to any address on port 5269:

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

Example 2: This example ACL configuration allows from any address to any single XMPP federation node on port 5269. The following values are used in this example:

- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1

  XMPP federation listening port = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
```

Example 3: This example ACL configuration allows from any address to specific XMPP federation nodes published in DNS.

Note The public addresses are published in DNS, but the private addresses are configured in the access-list command.

The following values are used in this sample configuration:

- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1

- Private second Cisco Unified Presence Release 8.0 IP address= 2.2.2.2
- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3
- XMPP federation listening port = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp any host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp any host 3.3.3.3 eq 5269
```

Example 4: This example ACL configuration allows only from a specific federated domain interface to specific XMPP federation nodes published in DNS.

Note The public addresses are published in DNS, but the private addresses are configured in the access-list command.

The following values are used in this sample configuration:

- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2
- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3
- XMPP federation listening port = 5269
- External interface of the foreign XMPP enterprise = 100.100.100.100

```
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 3.3.3.3 eq 5269
```

Example NAT Configuration for XMPP Federation

Example 1: Single node with XMPP federation enabled

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP address = 10.10.10.10
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- XMPP federation listening port = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Example 2: Multiple nodes with XMPP federation, each with a public IP address in DNS

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP addresses = 10.10.10.10, 20.20.20.20, 30.30.30.30
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2

- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3
- XMPP federation listening port = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Example 3: Multiple nodes with XMPP federation, but a single public IP address in DNS with arbitrary ports published in DNS (PAT).

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP Address = 10.10.10.10
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1, port 5269
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2, arbitrary port 25269
- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3, arbitrary port 35269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

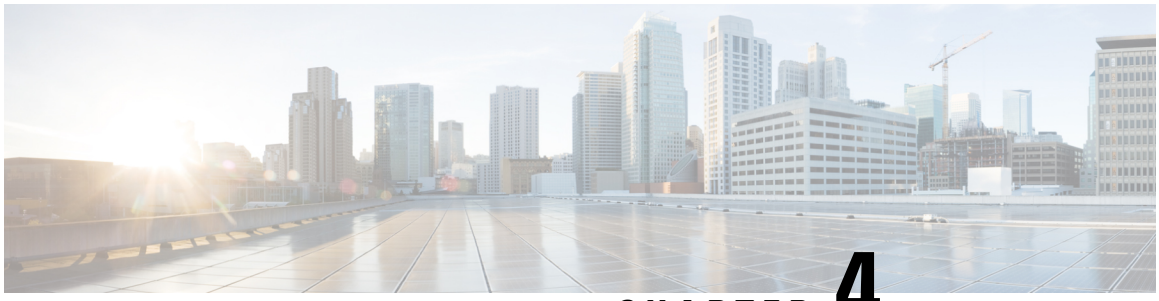
Feature History for Cisco Unified Presence

The following table lists the release history for this feature.

Table 2: Feature History for Cisco Unified Presence

Feature Name	Releases	Feature Information
Cisco Presence Federation Proxy	8.0(4)	The Cisco Unified Presence proxy feature was introduced.

Feature Name	Releases	Feature Information
Cisco Presence Federation Proxy	8.3(1)	The Unified Communications Wizard was added to ASDM. By using the wizard, you can configure the Cisco Presence Federation Proxy. Support for XMPP Federation was introduced.
SIP, SCCP, and TLS Proxy support for IPv6	9.3(1)	You can now inspect IPv6 traffic when using SIP, SCCP, and TLS Proxy (using SIP or SCCP). We did not modify any commands. We did not modify any ASDM screens.
Support for Cisco Unified Communications Manager 8.6	9.3(1)	The ASA now interoperates with Cisco Unified Communications Manager Version 8.6 (including SCCPv21 support). We did not modify any commands. We did not modify any ASDM screens.



CHAPTER 4

ASA and Cisco Mobility Advantage

This chapter describes how to configure the ASA for Cisco Unified Communications Mobility Advantage Proxy features.

Information about the Cisco Mobility Advantage Proxy Feature

This section contains the following topics:

Cisco Mobility Advantage Proxy Functionality

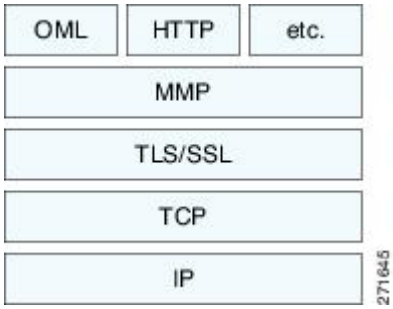
To support Cisco UMA for the Cisco Mobility Advantage solution, the mobility advantage proxy (implemented as a TLS proxy) includes the following functionality:

- The ability to allow no client authentication during the handshake with clients.
- Allowing an imported PKCS-12 certificate to server as a proxy certificate.

The ASA includes an inspection engine to validate the Cisco UMA Mobile Multiplexing Protocol (MMP).

MMP is a data transport protocol for transmitting data entities between Cisco UMA clients and servers. As shown in the following figure, MMP must be run on top of a connection-oriented protocol (the underlying transport) and is intended to be run on top of a secure transport protocol such as TLS. The Orative Markup Language (OML) protocol is intended to be run on top of MMP for the purposes of data synchronization, as well as the HTTP protocol for uploading and downloading large files.

Figure 7: MMP Stack



The TCP/TLS default port is 5443. There are no embedded NAT or secondary connections.

Cisco UMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler. The ASA takes the following actions on the MMP headers and data:

- Verifies that client MMP headers are well-formed. Upon detection of a malformed header, the TCP session is terminated.
- Verifies that client to server MMP header lengths are not exceeded. If an MMP header length is exceeded (4096), then the TCP session is terminated.
- Verifies that client to server MMP content lengths are not exceeded. If an entity content length is exceeded (4096), the TCP session is terminated.

Mobility Advantage Proxy Deployment Scenarios

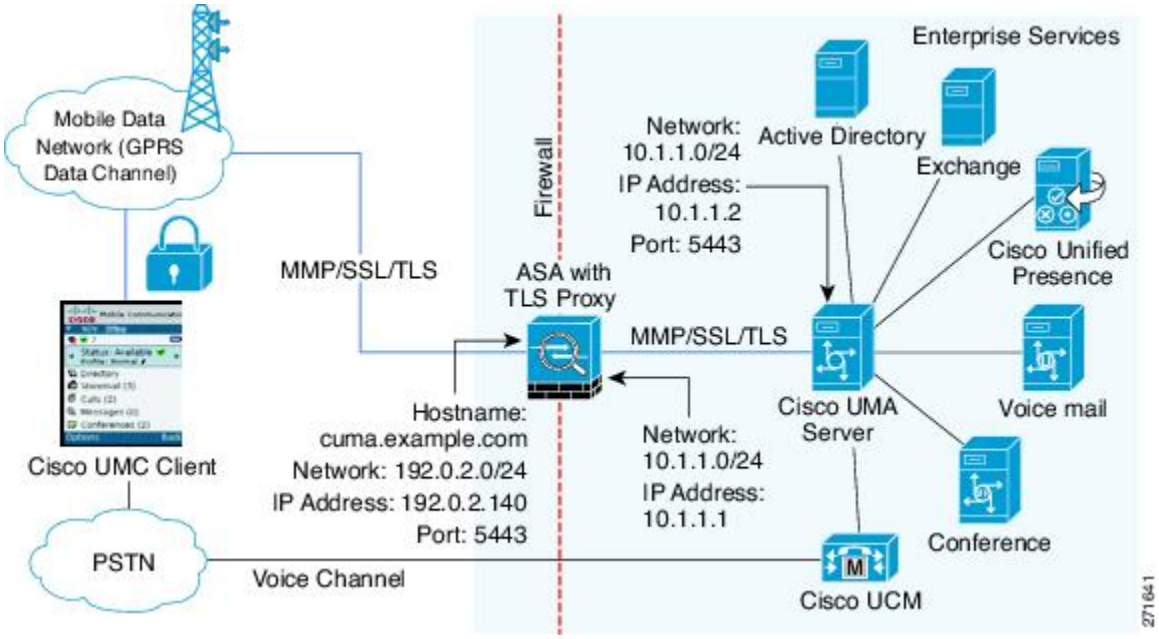
The following figures show the two deployment scenarios for the TLS proxy used by the Cisco Mobility Advantage solution. In scenario 1 (the recommended deployment architecture), the ASA functions as both the firewall and TLS proxy. In scenario 2, the ASA functions as the TLS proxy only and works with an existing firewall. In both scenarios, the clients connect from the Internet.

In the scenario 1 deployment, the ASA is between a Cisco UMA client and a Cisco UMA server. The Cisco UMA client is an executable that is downloaded to each smartphone. The Cisco UMA client applications establishes a data connection, which is a TLS connection, to the corporate Cisco UMA server. The ASA intercepts the connections and inspects the data that the client sends to the Cisco UMA server.

Note The TLS proxy for the Cisco Mobility Advantage solution does not support client authentication because the Cisco UMA client cannot present a certificate. The following commands can be used to disable authentication during the TLS handshake.

```
hostname(config)# tls-proxy my_proxy
hostname(config-tlsp)# no server authenticate-client
```

Figure 8: Security Appliance as Firewall with Mobility Advantage Proxy and MMP Inspection



In the above figure, the ASA performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

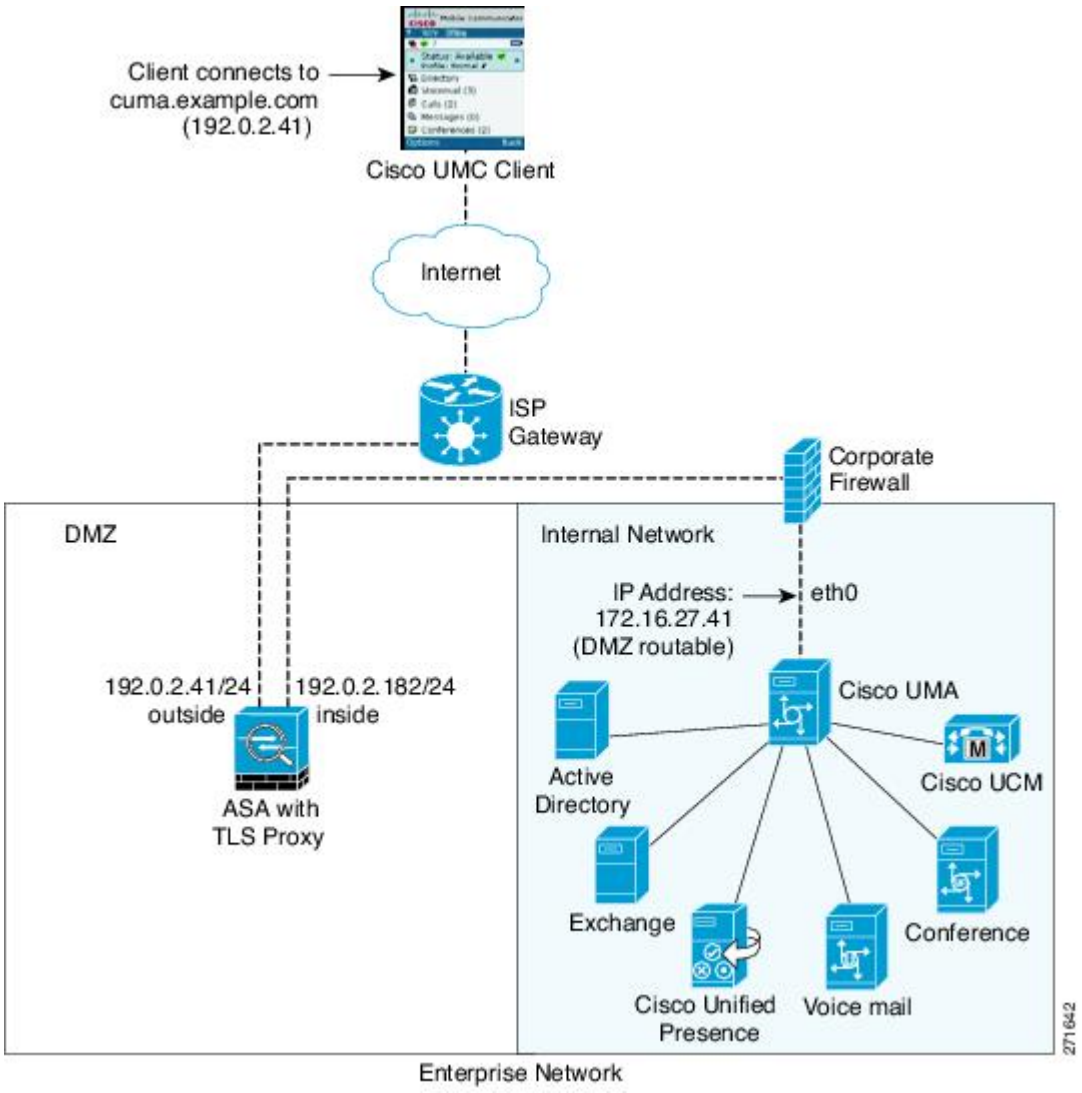
The following figure shows deployment scenario 2, where the ASA functions as the TLS proxy only and does not function as the corporate firewall. In this scenario, the ASA and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 192.0.12.183.

```
hostname(config)# object network obj-0.0.0.0-01
hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0
hostname(config-network-object)# nat (outside,inside) dynamic 192.0.2.183
```

Note This interface PAT rule converges the Cisco UMA client IP addresses on the outside interface of the ASA into a single IP address on the inside interface by using different source ports. Performing this action is often referred as “outside PAT”. “Outside PAT” is not recommended when TLS proxy for Cisco Mobility Advantage is enabled on the same interface of the ASA with phone proxy, Cisco Unified Presence, or any other features involving application inspection. “Outside PAT” is not supported completely by application inspection when embedded address translation is needed.

Figure 9: Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as Mobility Advantage Proxy Only



Mobility Advantage Proxy Using NAT/PAT

In both scenarios, NAT can be used to hide the private address of the Cisco UMA servers.

In scenario 2, PAT can be used to converge all client traffic into one source IP, so that the firewall does not have to open up a wildcard pinhole for inbound traffic.

```
hostname(config)# access-list cumc extended permit tcp any host 172.16.27.41 eq 5443
```

versus

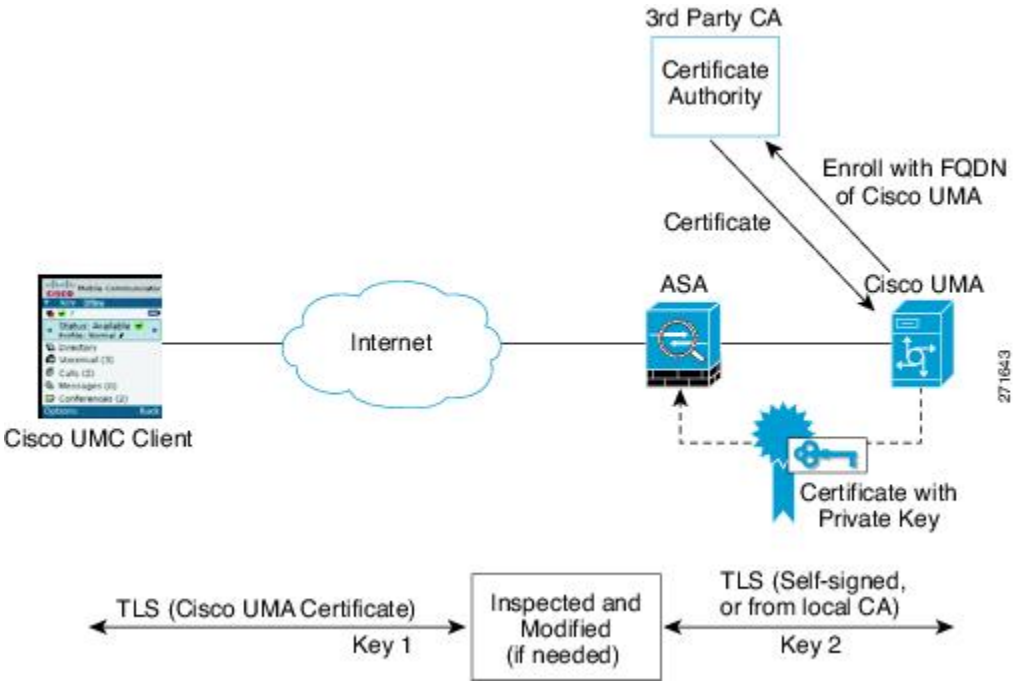
```
hostname(config)# access-list cumc extended permit tcp host 192.0.2.183 host 172.16.27.41
eq 5443
```

Trust Relationships for Cisco UMA Deployments

To establish a trust relationship between the Cisco UMC client and the ASA, the ASA uses the Cisco UMA server certificate and keypair or the ASA obtains a certificate with the Cisco UMA server FQDN (certificate impersonation). Between the ASA and the Cisco UMA server, the ASA and Cisco UMA server use self-signed certificates or certificates issued by a local certificate authority.

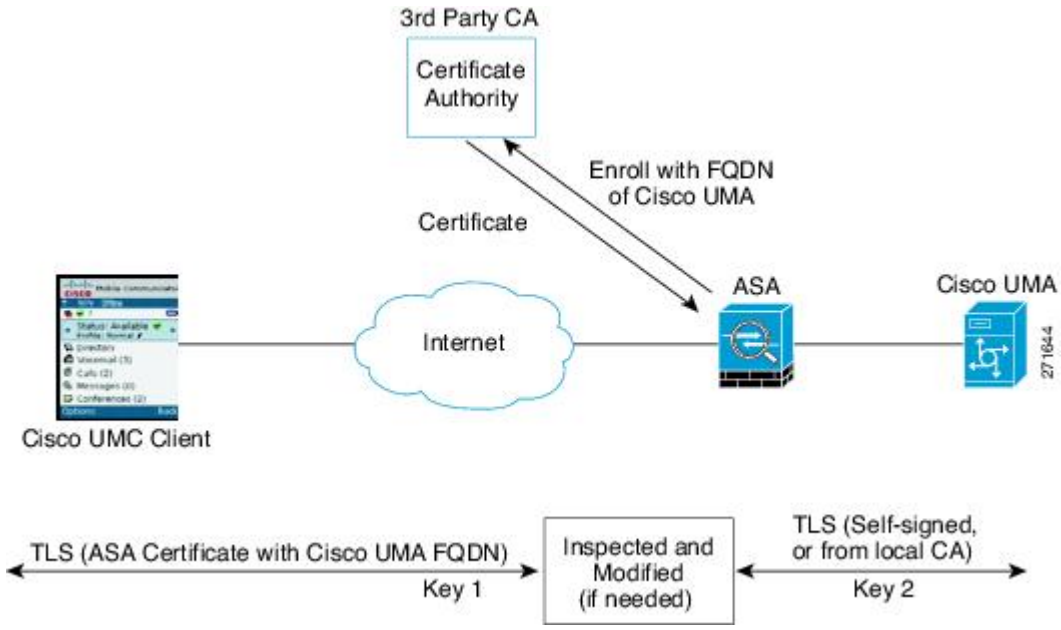
The following figure shows how you can import the Cisco UMA server certificate onto the ASA. When the Cisco UMA server has already enrolled with a third-party CA, you can import the certificate with the private key onto the ASA. Then, the ASA has the full credentials of the Cisco UMA server. When a Cisco UMA client connects to the Cisco UMA server, the ASA intercepts the handshake and uses the Cisco UMA server certificate to perform the handshake with the client. The ASA also performs a handshake with the server.

Figure 10: How the Security Appliance Represents Cisco UMA – Private Key Sharing



The following figure shows another way to establish the trust relationship. The following figure shows a green field deployment, because each component of the deployment has been newly installed. The ASA enrolls with the third-party CA by using the Cisco UMA server FQDN as if the ASA is the Cisco UMA server. When the Cisco UMA client connects to the ASA, the ASA presents the certificate that has the Cisco UMA server FQDN. The Cisco UMA client believes it is communicating to with the Cisco UMA server.

Figure 11: How the Security Appliance Represents Cisco UMA – Certificate Impersonation



A trusted relationship between the ASA and the Cisco UMA server can be established with self-signed certificates. The ASA's identity certificate is exported, and then uploaded on the Cisco UMA server truststore. The Cisco UMA server certificate is downloaded, and then uploaded on the ASA truststore by creating a trustpoint and using the **crypto ca authenticate** command.

Configuring Cisco Mobility Advantage (CLI)

This section includes the following topics:

Task Flow for Configuring Cisco Mobility Advantage

To configure for the ASA to perform TLS proxy and MMP inspection, perform the following tasks.

It is assumed that self-signed certificates are used between the ASA and the Cisco UMA server.

Before you begin

Export the Cisco UMA server certificate and keypair in PKCS-12 format so that you can import it onto the ASA. The certificate will be used during the handshake with the Cisco UMA clients.

Step 1 Create the static NAT for the Cisco UMA server by entering the following commands:

```
hostname(config)# object network name
hostname(config-network-object)# host real_ip
hostname(config-network-object)# nat (real_ifc,mapped_ifc) static mapped_ip
```

Step 2 Import the Cisco UMA server certificate onto the ASA by entering the following commands:

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase [paste base 64 encoded pkcs12]
hostname(config)# quit
hostname(config)# crypto ca import trustpoint pkcs12 passphrase [paste base 64 encoded pkcs12]
hostname(config)# quit
```

Step 3 Install the Cisco UMA server certificate on the ASA. See Installing the Cisco UMA Server Certificate.

Step 4 Create the TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server. See Creating the TLS Proxy Instance.

Step 5 Enable the TLS proxy for MMP inspection. See Enabling the TLS Proxy for MMP Inspection.

Installing the Cisco UMA Server Certificate

Install the Cisco UMA server self-signed certificate in the ASA truststore. This task is necessary for the ASA to authenticate the Cisco UMA server during the handshake between the ASA proxy and Cisco UMA server.

Before you begin

Export the Cisco UMA server certificate and keypair in PKCS-12 format so that you can import it onto the ASA.

Procedure

	Command or Action	Purpose
Step 1	`hostname(config)# crypto ca trustpoint` *trustpoint_name* **Example:** `hostname(config)# crypto ca trustpoint cuma_server`	Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the Cisco UMA server. A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.
Step 2	`hostname(config-ca-trustpoint)# enrollment terminal`	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment).
Step 3	`hostname(config-ca-trustpoint)# exit`	Exits from the CA Trustpoint configuration mode.

Step 4	`hostname(config)# crypto ca authenticate` *trustpoint*
	Example:
	`hostname(config)# crypto ca authenticate cuma_server`
	`Enter the base 64 encoded CA certificate.`
	`End with a blank line or the word "quit" on a line by itself`
	`[ certificate data omitted ]`
	`Certificate has the following attributes:`
	`Fingerprint: 21B598D5 4A81F3E5 0B24D12E 3F89C2E4`
	`% Do you accept this certificate? [yes/no]: yes`
	`Trustpoint CA certificate accepted.`
	`% Certificate successfully imported`
	`hostname(config)#`
		Installs and authenticates the CA certificates associated with a trustpoint created for the Cisco UMA server.
		Where *trustpoint* specifies the trustpoint from which to obtain the CA certificate. Maximum name length is 128 characters.
		The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.

What to do next

Once you have created the trustpoints and installed the Cisco UMA certificate on the ASA, create the TLS proxy instance. SeeCreating the TLS Proxy Instance.

Creating the TLS Proxy Instance

Create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server.

Before you begin

Before you can create the TLS proxy instance, you must have installed the Cisco UMA server self-signed certificate in the ASA truststore.

Procedure

	Command or Action	Purpose

Step 1	`hostname(config)# tls-proxy proxy_name`
	Example:
	`tls-proxy cuma_tlsproxy`
		Creates the TLS proxy instance.

Step 2	`hostname(config-tlsp)# server trust-point proxy_name`
	Example:
	`hostname(config-tlsp)# server trust-point cuma_proxy`
		Specifies the proxy trustpoint certificate presented during TLS handshake.
		The certificate must be owned by the ASA (identity certificate).

Step 3	`hostname(config-tlsp)# client trust-point proxy_name`
	Example:
	`hostname(config-tlsp)# client trust-point cuma_proxy`
		Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client.
		The certificate must be owned by the ASA (identity certificate).

Step 4	`hostname(config-tlsp)# no server authenticate-client`	Disables client authentication. Disabling TLS client authentication is required when the ASA must interoperate with a Cisco UMA client or clients such as a Web browser that are incapable of sending a client certificate.

Step 5	`hostname(config-tlsp)# client cipher-suite cipher_suite`
	Example:
	`hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1`
		Specifies cipher suite configuration.
		For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite.

What to do next

Once you have created the TLS proxy instance, enable it for MMP inspection. See Enabling the TLS Proxy for MMP Inspection.

Enabling the TLS Proxy for MMP Inspection

Cisco UMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler.

Procedure

	Command or Action	Purpose

Step 1	`hostname(config)# class-map class_map_name`
	Example:
	`hostname(config)# class-map cuma_tlsproxy`
		Configures the class of traffic to inspect. Traffic between the Cisco UMA server and client uses MMP and is handled by MMP inspection.
		Where *class_map_name* is the name of the MMP class map.

Step 2	`hostname(config-cmap)# match port tcp eq port`
	Example:
	`hostname(config-cmap)# match port tcp eq 5443`
		Matches the TCP port to which you want to apply actions for MMP inspection.
		The TCP/TLS default port for MMP inspection is 5443.

Step 3	`hostname(config-cmap)# exit`	Exits from the Class Map configuration mode.

Step 4	`hostname(config)# policy-map name`
	Example:
	`hostname(config)# policy-map global_policy`
		Configures the policy map and attaches the action to the class of traffic.

Step 5	`hostname(config-pmap)# class classmap-name`
	Example:
	`hostname(config-pmap)# class cuma_proxy`
		Assigns a class map to the policy map so that you can assign actions to the class map traffic.
		Where *classmap_name* is the name of the MMP class map.

Step 6	`hostname(config-pmap)# inspect mmp tls-proxy proxy_name`
	Example:
	`hostname(config-pmap)# inspect mmp tls-proxy cuma_proxy`
		Enables MMP application inspection using the TLS proxy.

Step 7	`hostname(config-pmap)# exit`	Exits from the Policy Map configuration mode.

Step 8	`hostname(config)# service-policy policy_map_name global`
	Example:
	`service-policy global_policy global`
		Enables the service policy on all interfaces.

Configuring Cisco Mobility Advantage (ASDM)

To configure the Cisco Mobility Advantage Proxy by using ASDM, choose **Wizards** > **Unified Communications Wizard** from the menu. From the first page, select the Cisco Mobility Advantage Proxy option under the Remote Access section.

The wizard automatically creates the necessary TLS proxy, then guides you through creating the Unified Presence Proxy instance, importing and installing the required certificates, and finally enables the MMP inspection for the Mobility Advantage traffic automatically.

When using the wizard to create the Mobility Advantage proxy, ASDM automatically generates address translation (NAT) statements, and creates the access rules that are necessary to allow traffic between the Cisco Mobility Advantage server and the mobility clients.

The following steps provide the high-level overview for configuring the Mobility Advantage proxy:

Step 1 Specify settings to define the private and public network topology, such the public and private network interfaces, and the IP addresses of the Cisco Mobility Advantage server. See Configuring the Topology for the Cisco Mobility Advantage Proxy.

Step 2 Configure the certificates that are exchanged between the Cisco Mobility Advantage server and the ASA. SeeConfiguring the Server-Side Certificates for the Cisco Mobility Advantage Proxy.

Step 3 Configure the client-side certificate management, namely the certificates that are exchanged between the Unified Mobile Communicator clients and the ASA. See Configuring the Client-Side Certificates for the Cisco Mobility Advantage Proxy.

The wizard completes by displaying a summary of the configuration created for Mobility Advantage Proxy.

Configuring the Topology for the Cisco Mobility Advantage Proxy

When configuring the Mobility Advantage Proxy, you specify settings to define the private and public network topology, such the private and public network interfaces, and the private and public IP addresses of the Cisco Mobility Advantage server.

The values that you specify in this page generate the following configuration settings for the Mobility Advantage Proxy:

- Static PAT for the Cisco Mobility Advantage server
- Static NAT for Cisco Unified Mobile Communicator clients if the Enable address translation for Mobility clients check box is checked.
- ACLs to allow Cisco Unified Mobile Communicator clients to access the Cisco Mobility Advantage server

Step 1 In the Private Network area, choose the interface from the drop-down list.

Step 2 In the Unified MA Server area, enter the private and public IP address for the Cisco Mobility Advantage server. Entering ports for these IP addresses is optional. By default port number 5443 is entered, which is the default TCP port for MMP inspection.

Step 3 In the FQDN field, enter the domain name for the Cisco Mobility Advantage server. This domain name is included in the certificate signing request that you generate later in this wizard.

Step 4 In the Public Network area, choose an interface from the drop-down list. The proxy uses this interface for configuring static PAT for the Cisco Mobility Advantage server and the ACLs to allow Cisco Unified Mobile Communicator clients to access the Cisco Mobility Advantage server.

Step 5 **To configure whether address translation (NAT) is used by** Cisco Unified Mobile Communicator clients, check the **Enable address translation for Mobility clients** check box and choose whether to use the IP address of the public interface or whether to enter an IP address.

Step 6 Click **Next**.

Configuring the Server-Side Certificates for the Cisco Mobility Advantage Proxy

A trusted relationship between the ASA and the Cisco UMA server can be established with self-signed certificates. The ASA's identity certificate is exported, and then uploaded on the Cisco UMA server truststore. The Cisco UMA server certificate is downloaded, and then uploaded on the ASA truststore.

The supports using self-signed certificates only at this step.

Step 1 In the ASA's Identity Certificate area, click **Generate and Export ASA's Identity Certificate**.

An information dialog boxes appear indicating that the enrollment seceded. In the Enrollment Status dialog box, click **OK**. The Export certificate dialog box appears.

Note
- If an identity certificate for the ASA has already been created, the button in this area appears as **Export ASA's Identity Certificate** and the Export certificate dialog box immediately appears.
- When using the wizard to configure the Cisco Mobility Advantage proxy, the wizard only supports installing self-signed certificates.

Step 2 Export the identity certificate generated by the wizard for the ASA.

You must install this certificate into the Cisco Mobility Advantage server.

Step 3 In the Unified MA Server's Certificate area, click **Install Unified MA Server's Certificate**. The Install Certificate dialog appears. Either select the certificate file, or paste the certificate contents into the dialog box.

See the Cisco Mobility Advantage server documentation for information on how to export the certificate for this server.

Step 4 Click **Next**.

Configuring the Client-Side Certificates for the Cisco Mobility Advantage Proxy

To establish a trust relationship between the Cisco Unified Mobile Communicator (UMC) clients and the ASA, the ASA uses a CA-signed certificate that is configured with the Cisco Mobility Advantage server's FQDN (also referred to as certificate impersonation).

In the Client-Side Certificate Management page, you enter both the intermediate CA certificate (if applicable, as in the cases of Verisign) and the signed ASA identity certificate.

Step 1 In the ASA's Identity Certificate area, click **Generate CSR**. The CSR parameters dialog box appears. If the ASA already has a signed identity certificate, you can skip this step.

This certificate is presented to Unified Mobile Communicator clients. When configuring the certificate:

- Choose a key size that provides sufficient security. Your CA might have a minimum key size requirement.
- The wizard provides the common name (CN), which is the FQDN of the Cisco Mobility Advantage server.
- Add additional DNs as appropriate.

Information dialog boxes appear indicating that the wizard is delivering the settings to the ASA and retrieving the certificate key pair information. The Identity Certificate Request dialog box appears. Save the certificate to a file and submit it to the CA for signing.

Step 2 Click **Install ASA's Identity Certificate**. Install the certificate. See Installing the ASA Identity Certificate on the Mobility Advantage Server.

Step 3 Click **Install Root CA's Certificate**. The Install Certificate dialog box appears. Select the certificate file and install it.

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for Mobility Advantage Proxy.

Installing the ASA Identity Certificate on the Mobility Advantage Server

When configuring certificates for the Cisco Mobility Advantage Proxy, you must install the ASA identity certificate on the Cisco Mobility Advantage server.

Typically, a certificate authority returns two certificates: your signed identity certificate and the certificate authority's certificate (referred to as the root certificate). However, some certificate authorities (for example, VeriSign) might also send you an intermediate certificate.

The root certificate from the certificate authority is used to sign other certificates. The root certificate is used by the ASA to authenticate your signed identity certificate received from the certificate authority.

If the certificate authority provided an intermediate certificate, you must enter the certificate text in the Intermediate Certificate (If Applicable) area of the Install ASA's Identity Certificate dialog box.

For the Cisco Mobility Advantage Proxy, you install the root certificate in another dialog box.

Step 1 In the Intermediate Certificate (If Applicable) area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 2 In the ASA's Identity Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click Install Certificate.

  To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 3 Click **Install Certificate**.

Monitoring for Cisco Mobility Advantage

Mobility advantage proxy can be debugged the same way as IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, using the following commands to enable TLS proxy-related debugging and syslog output only:

```
hostname# debug inspect tls-proxy events
hostname# debug inspect tls-proxy errors
hostname# config terminal
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see the Monitoring the TLS Proxy.

Enable the **debug mmp** command for MMP inspection engine debugging:

```
MMP:: received 60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: version OLWP-2.0
MMP:: forward 60/60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: received 100 bytes from inside:2.2.2.2/5443 to outside:1.1.1.1/2000
MMP:: session-id: ABCD_1234
MMP:: status: 201
MMP:: forward 100/100 bytes from inside:2.2.2.2/5443 to outside 1.1.1.1/2000
MMP:: received 80 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: content-type: http/1.1
MMP:: content-length: 40
```

You can also capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Configuration Examples for Cisco Mobility Advantage

- Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection

  Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only

This section describes sample configurations that apply to two deployment scenarios for the TLS proxy used by the Cisco Mobility Advantage solution—scenario 1 where the ASA functions as both the firewall and TLS proxy and scenario 2 where the ASA functions as the TLS proxy only. In both scenarios, the clients connect from the Internet.

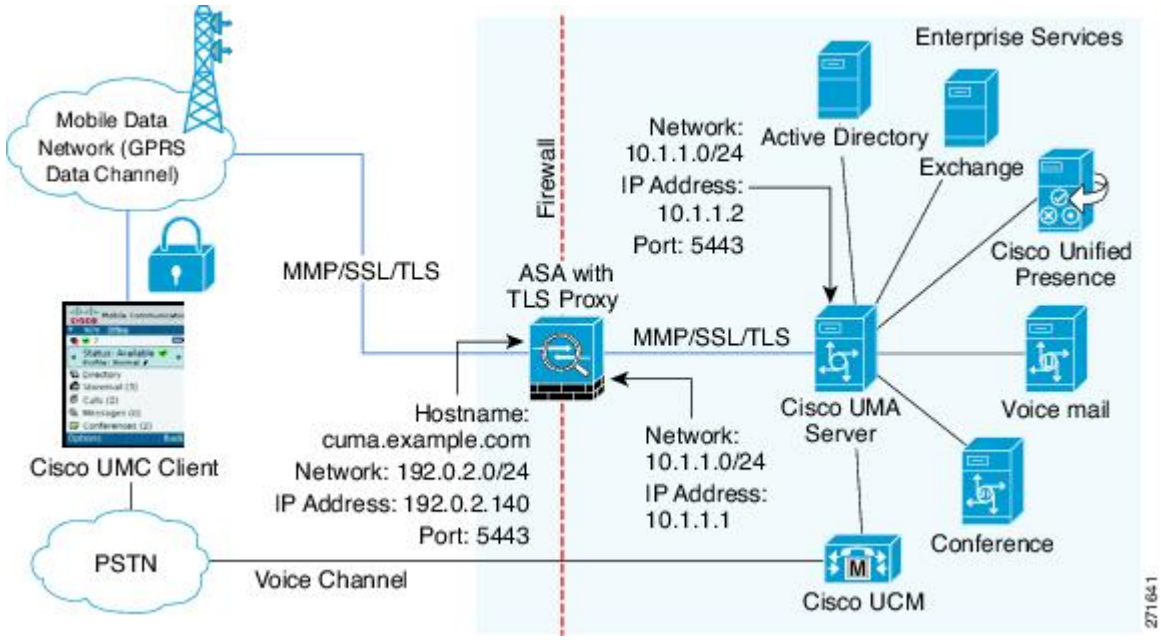
In the samples, you export the Cisco UMA server certificate and key-pair in PKCS-12 format and import it to the ASA. The certificate will be used during handshake with the Cisco UMA clients.

Installing the Cisco UMA server self-signed certificate in the ASA truststore is necessary for the ASA to authenticate the Cisco UMA server during handshake between the ASA proxy and Cisco UMA server. You create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server. Lastly, you must enable TLS proxy for MMP inspection.

Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection

As shown in the following figure, the ASA functions as both the firewall and TLS proxy. In the scenario 1 deployment, the ASA is between a Cisco UMA client and a Cisco UMA server. In this scenario, the ASA performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

Figure 12: Cisco UMC/Cisco UMA Architecture – Scenario 1: Security Appliance as Firewall with TLS Proxy and MMP Inspection




```
object network obj-10.1.1.2-01
host 10.1.1.2
nat (inside,outside) static 192.0.2.140
crypto ca import cuma_proxy pkcs12 sample_passphrase
<cut-paste base 64 encoded pkcs12 here>
quit
! for CUMA server's self-signed certificate
crypto ca trustpoint cuma_server
enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
tls-proxy cuma_proxy
server trust-point cuma_proxy
no server authenticate-client
client cipher-suite aes128-sha1 aes256-sha1
```

```
class-map cuma_proxy
match port tcp eq 5443
policy-map global_policy
class cuma_proxy
inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

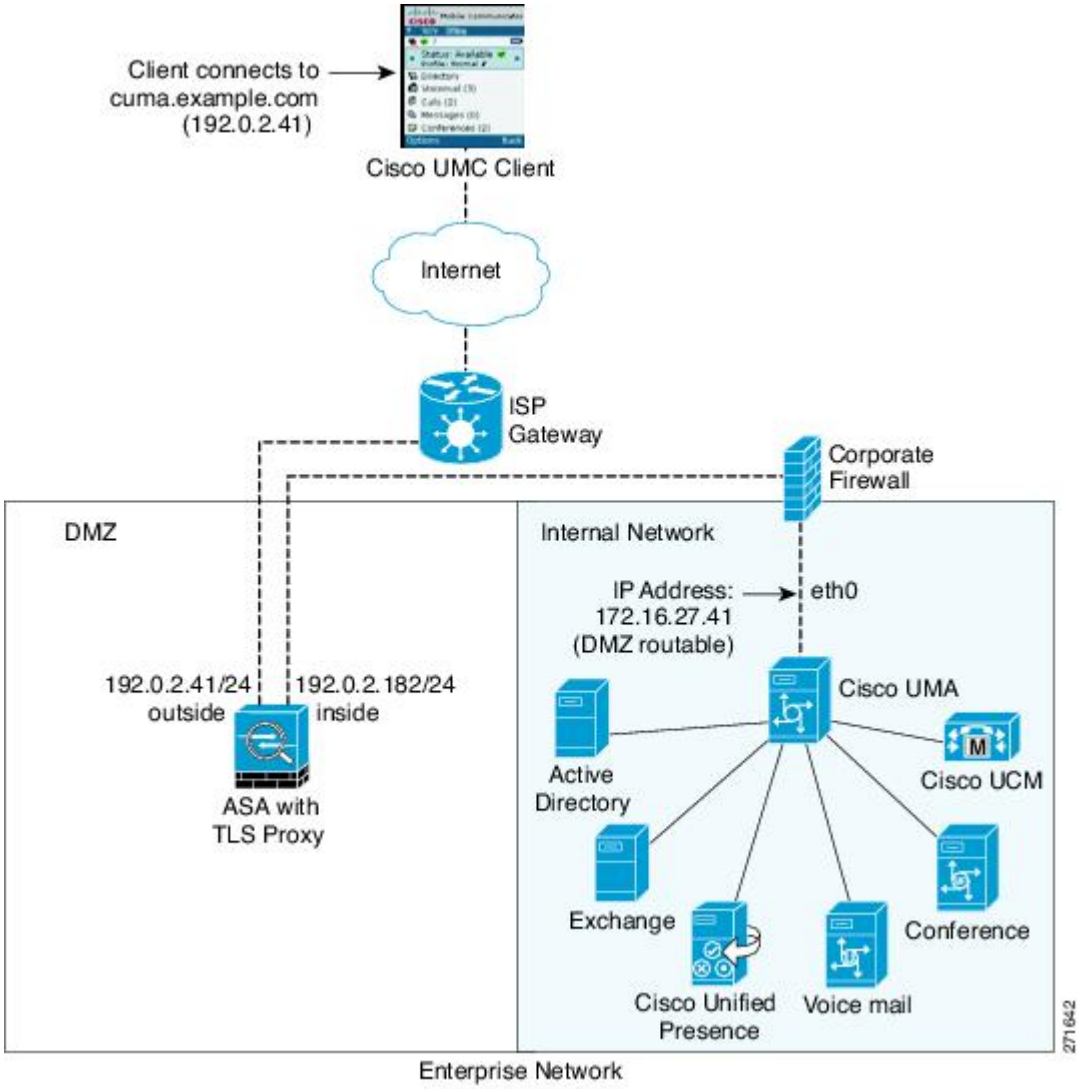
Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only

As shown in the following figure (scenario 2), the ASA functions as the TLS proxy only and works with an existing firewall. The ASA and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 192.0.2.183.

```
hostname(config)# object network obj-0.0.0.0-01
hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0
hostname(config-network-object)# nat (outside,inside) dynamic 192.0.2.183
```

Figure 13: Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as TLS Proxy Only




```
object network obj-172.16.27.41-01
host 172.16.27.41
nat (inside,outside) static 192.0.2.140
object network obj-0.0.0.0-01
subnet 0.0.0.0 0.0.0.0
nat (outside,inside) dynamic 192.0.2.183
crypto ca import cuma_proxy pkcs12 sample_passphrase
<cut-paste base 64 encoded pkcs12 here>
quit
! for CUMA server's self-signed certificate
crypto ca trustpoint cuma_server
enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
```

```
tls-proxy cuma_proxy
server trust-point cuma_proxy
no server authenticate-client
client cipher-suite aes128-sha1 aes256-sha1
class-map cuma_proxy
match port tcp eq 5443
policy-map global_policy
class cuma_proxy
inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

Feature History for Cisco Mobility Advantage

The following table lists the release history for this feature.

Table 3: Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
Cisco Mobility Advantage Proxy	8.0(4)	The Cisco Mobility Advantage Proxy feature was introduced.
Cisco Mobility Advantage Proxy	8.3(1)	The Unified Communications Wizard was added to ASDM. By using the wizard, you can configure the Cisco Mobility Advantage Proxy.
SIP, SCCP, and TLS Proxy support for IPv6	9.3(1)	You can now inspect IPv6 traffic when using SIP, SCCP, and TLS Proxy (using SIP or SCCP). We did not modify any commands. We did not modify any ASDM screens.
Support for Cisco Unified Communications Manager 8.6	9.3(1)	The ASA now interoperates with Cisco Unified Communications Manager Version 8.6 (including SCCPv21 support). We did not modify any commands. We did not modify any ASDM screens.