CHAPTER 27

Configuring Cisco Unified Communications Proxy Features

This chapter describes how to configure the adaptive security appliance for Cisco Unified Communications Proxy features.

This chapter includes the following sections:

Overview of the Adaptive Security Appliance in Cisco Unified Communications

This section describes the Cisco UC Proxy features on the Cisco ASA 5500 series appliances. 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 Cisco ASA 5500 Series appliances are a strategic platform to provide proxy functions for unified communications deployments.

The Cisco UC Proxy includes the following solutions:

Phone Proxy: Secure remote access for Cisco encrypted endpoints, and VLAN traversal for Cisco softphones

The phone proxy feature enables termination of Cisco SRTP/TLS-encrypted endpoints for secure remote access. The phone proxy allows large scale deployments of secure phones without a large scale VPN remote access hardware deployment. End-user infrastructure is limited to just the IP endpoint, without VPN tunnels or hardware.

The Cisco adaptive security appliance phone proxy is the replacement product for the Cisco Unified Phone Proxy. Additionally, the phone proxy can be deployed for voice/data VLAN traversal for softphone applications. Cisco IP Communicator (CIPC) traffic (both media and signaling) can be proxied through the security appliance, thus traversing calls securely between voice and data VLANs.

For information about the differences between the TLS proxy and phone proxy, go to the following URL for Unified Communications content, including TLS Proxy vs. Phone Proxy white paper:

http://www.cisco.com/go/secureuc

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 security appliance 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 security appliance 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 security appliance 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.

TLS Proxy Applications in Cisco Unified Communications

Table 27-1 shows the Cisco Unified Communications applications that utilize the TLS proxy on the security appliance.

Table 27-1 TLS Proxy Applications and the Security Appliance

Application	TLS Client	TLS Server	Client Authentication	Security Appliance Server Role	Security Appliance Client Role
Phone Proxy and TLS Proxy	IP phone	Cisco UCM	Yes	Proxy certificate, self-signed or by internal CA	Local dynamic certificate signed by the security appliance CA (might not need certificate for phone proxy application)
Mobility Proxy	Cisco UMC	Cisco UMA	No	Using the Cisco UMA private key or certificate impersonation	Any static configured certificate
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 security appliance supports TLS proxy for various voice applications. For the phone proxy, the TLS proxy running on the security appliance has the following key features:

- The security appliance 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 security appliance 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 security appliance 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 security appliance), 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 security appliance 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 security appliance acts as a TLS proxy between the Cisco UP server and the foreign server. This allows the security appliance to proxy TLS messages on behalf of the server that initiates the TLS connection, and route the proxied TLS messages to the client. The security appliance 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 security appliance require a Unified Communications Proxy license:

- Phone proxy
- TLS proxy for encrypted voice inspection
- Mobility proxy
- Presence federation proxy

The Unified Communications proxy features are licensed by TLS session. For the phone proxy or TLS proxy, each IP phone may have a single connection to the Cisco UCM server or two connections —one connection to the primary Cisco UCM and one connection to the backup Cisco UCM. In the second scenario, the phone proxy uses two Unified Communications Proxy sessions because two TLS sessions are set up. For the mobility proxy and presence federation proxy, each endpoint utilizes one Unified Communications Proxy session.

Table 27-2 shows the Unified Communications Proxy license details by platform.

Table 27-2 License Requirements for the Security Appliance

Security Appliance Platform	Max UC Proxy Licenses	Tiers for UC Proxy Licenses
ASA 5505	24	24
ASA 5510	100	24, 50, 100
ASA 5520	1,000	24, 50, 100, 250, 500, 750, 1000
ASA 5540	2,000	24, 50, 100, 250, 500, 750, 1000, 2000
ASA 5550	3,000	24, 50, 100, 250, 500, 750, 1000, 2000, 3000

Table 27-3 shows the default and maximum TLS session details by platform.

Table 27-3 Default and Maximum TLS Sessions on the Security Appliance

Security Appliance Platform	Default TLS Sessions	Maximum TLS Sessions
ASA 5505	10	80
ASA 5510	100	200
ASA 5520	300	1200
ASA 5540	1000	4500
ASA 5550	2000	4500

A Unified Communications Proxy license is applied the same way as other licensed features (such as, SSL VPN), via the **activation-key** command. To check the license on the security appliance, use the **show version** or **show activation-key** command:

```
hostname# show activation-key
Serial Number:  P3000000179
Running Activation Key: 0xa700d24c 0x98caab35 0x88038550 0xaf383078 0x02382080

Licensed features for this platform:
Maximum Physical Interfaces  : Unlimited
Maximum VLANs                : 150
Inside Hosts                 : Unlimited
Failover                     : Active/Active
VPN-DES                      : Enabled
VPN-3DES-AES                 : Enabled
Security Contexts            : 10
GTP/GPRS                     : Enabled
VPN Peers                    : 750
WebVPN Peers                 : 750
AnyConnect for Mobile        : Disabled
AnyConnect for Linksys phone : Disabled
Advanced Endpoint Assessment : Enabled
UC Proxy Sessions            : 1000
This platform has an ASA 5520 VPN Plus license.

The flash activation key is the SAME as the running key.
hostname#
```

See the following links for additional information on licensing. If you are a registered user of Cisco.com and would like to obtain a Unified Communications Proxy license, go to the following website:

http://www.cisco.com/go/license

If you are not a registered user of Cisco.com, go to the following website:

https://tools.cisco.com/SWIFT/Licensing/RegistrationServlet

Provide your name, e-mail address, and the serial number for the security appliance as it appears in the show version command output.

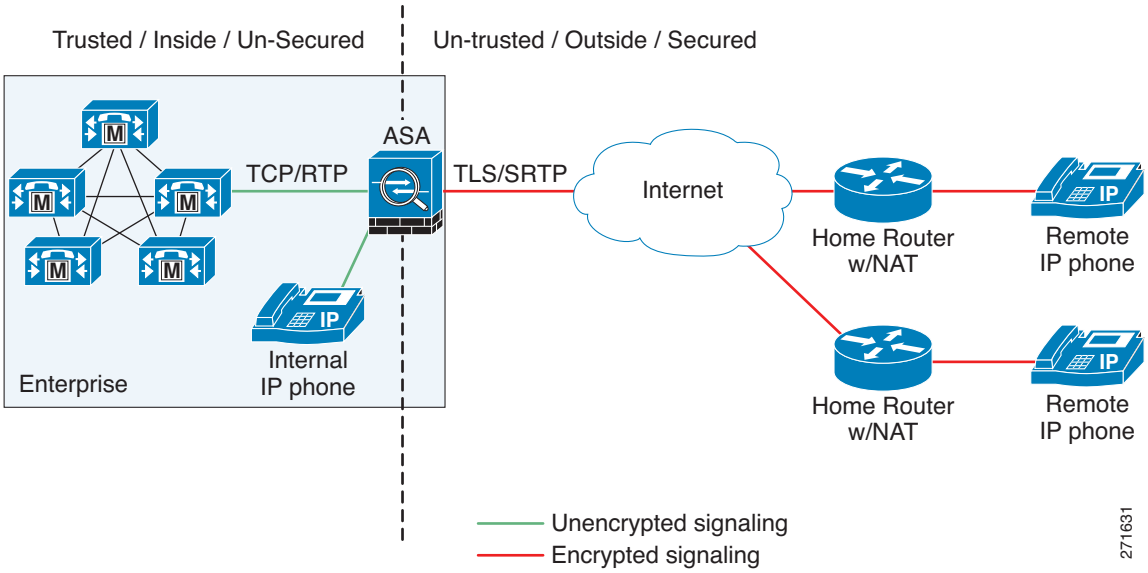
Phone Proxy

This section includes the following topics:

About the Phone Proxy

The phone proxy on the security appliance bridges IP telephony between the corporate IP telephony network and the Internet in a secure manner by forcing data from remote phones on an untrusted network to be encrypted. Telecommuters can connect their IP phones to the corporate IP telephony network over the Internet securely via the phone proxy without the need to connect over a VPN tunnel as illustrated by Figure 27-1.

Figure 27-1 Phone Proxy Secure Deployment



The phone proxy supports a Cisco UCM cluster in mixed mode or nonsecure mode. Regardless of the cluster mode, the remote phones that are capable of encryption are always forced to be in encrypted mode. TLS (signaling) and SRTP (media) are always terminated on the security appliance. The security appliance can also perform NAT, open pinholes for the media, and apply inspection policies for the SCCP and SIP protocols. In a nonsecure cluster mode or a mixed mode where the phones are configured as nonsecure, the phone proxy behaves in the following ways:

- The TLS connections from the phones are terminated on the security appliance and a TCP connection is initiated to the Cisco UCM.
- SRTP sent from external IP phones to the internal network IP phone via the security appliance is converted to RTP.

In a mixed mode cluster where the internal IP phones are configured as authenticated, the TLS connection is not converted to TCP to the Cisco UCM but the SRTP is converted to RTP.

In a mixed mode cluster where the internal IP phone is configured as encrypted, the TLS connection remains a TLS connection to the Cisco UCM and the SRTP from the remote phone remains SRTP to the internal IP phone.

Since the main purpose of the phone proxy is to make the phone behave securely while making calls to a nonsecure cluster, the phone proxy performs the following major functions:

- Creates the certificate trust list (CTL) file, which is used to perform certificate based authentication with remote phones.
- Modifies the IP phone configuration file when it is requested via TFTP, changes security fields from nonsecure to secure, and signs all files sent to the phone. These modifications secure remote phones by forcing the phones to perform encrypted signaling and media.
- Terminates TLS signaling from the phone and initiates TCP or TLS to Cisco UCM
- Inserts itself into the media path by modifying the Skinny and SIP signaling messages.
- Terminates SRTP and initiates RTP/SRTP to the called party.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance.

See "Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-72". See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Phone Proxy Limitations and Restrictions

The phone proxy has the following limitations and restrictions:

- Only one phone proxy instance can be configured on the security appliance by using the **phone-proxy** command. See the *Cisco Security Appliance Command Reference* for information about the **phone-proxy** command.
- The phone proxy only supports one Cisco UCM cluster. See Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster, page 27-14 and Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster, page 27-19 for the steps to configure the Cisco UCM cluster for the phone proxy.
- The phone proxy is not supported when the security appliance is running in transparent mode or multiple context mode.
- When a remote IP phone calls an invalid internal or external extension, the phone proxy does not support playing the annunciator message from the Cisco UCM. Instead, the remote IP phone plays a fast busy signal instead of the annunciator message "Your call cannot be completed ..." However, when an internal IP phone dials in invalid extension, the annunciator messages plays "Your call cannot be completed ..."
- The phone proxy does not support inspection of packets from phones connecting to the phone proxy over a VPN tunnel. Therefore, sending phone proxy traffic through a VPN tunnel is not supported. Configuring the phone proxy feature on the security appliance allows IP phones to connect to the corporate network without requiring that the traffic go through VPN tunnels.
- The phone proxy does not support recording calls when the recording traffic must traverse the security appliance to get to the recording device. For example, the Unified Communication Manager versions 6.x and 7.x supports using a third-party recording device with the forking feature. When the recording feature is used with the phone proxy, the feature creates a second RTP media stream that is a copy of the original RTP media stream. The existence of two RTP media streams from the outside IP phone to the recording device on behind the security device disrupts the IP phone audio.
- The security appliance supports stateful failover for the phone proxy in the following way. When the active unit goes down, any calls from IP phones going through the phone proxy fail, media stops flowing, and the IP phones should unregister from the failed unit and reregister with the active unit. Then, the calls must be re-established."
- The phone proxy does not support communication with internal IP phones that natively use the Secure Real-Time Protocol (SRTP).

- The phone proxy does not support IP phones sending Real-Time Control Protocol (RTCP) packets through the security appliance. Disable RTCP packets in the Cisco Unified CM Administration console from the Phone Configuration page. See your Cisco Unified Communications Manager (CallManager) documentation for information about setting this configuration option.
- When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Prefences > Network tab > Use this Device Name field) or Administrators resetting the devide name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.
- The phone proxy does not support IP phones sending SCCP video messages using Cisco VT Advantage because SCCP video messages do not support SRTP keys.
- For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the security appliance.
- The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.
- Multiple IP phones behind one NAT device must be configured to use the same security mode.

  When the phone proxy is configured for a mixed-mode cluster and multiple IP phones are behind one NAT device and registering through the phone proxy, all the SIP and SCCP IP phones must be configured as authenticated or encrypted, or all as non-secure on the Unified Call Manager.

  For example, if there are four IP phones behind one NAT device where two IP phones are configured using SIP and two IP phones are configured using SCCP, the following configurations on the Unified Call Manager are acceptable:

  – Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

    Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

  – Two SIP IP phones: both in non-secure mode

    Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode

  – Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode

    Two SCCP IP phones: both in non-secure mode

  This limitation results from the way the application-redirect rules (rules that convert TLS to TCP) are created for the IP phones.
- The phone proxy does not support displaying the lock icon on IP phone screens. IP phones display the lock icon on the phone screen during encrypted calls. Even though the lock icon is not displayed on the screen, the IP phone call is still encrypted because the phone proxy encrypts calls by default.

Phone Proxy Configuration

This section includes the following topics:

Configuration Prerequisites

Before configuring the phone proxy, ensure that the security appliance meets the following configuration requirements:

- The security appliance must have an IP address for media termination that meets the following criteria:
  - The IP address is a publicly routable address that is an unused IP address within an address range associated with the outside network interface on the security appliance.
  - The IP address cannot be the same address of an interface on the security appliance. This includes using the IP address of the external interface on the security appliance to which remote IP phones connect.
  - The IP address cannot overlap with existing static NAT pools or NAT rules.
  - The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
  - For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

Note If your organization security policy dictates that IP phones on internal networks must not have routes to external networks, we recommended that you use a Unified-Communications-aware NAT device on the internal network. By representing the media termination address with an address within the internal network address range, you do not need to expose the internal IP phones to external routes.

- The TFTP server must reside on the same interface as the Cisco UCM.
- If you have an fully qualified domain name (FQDN) configured for the Cisco UCM rather than an IP address, you must configure and enable DNS lookup on the security appliance. For information about the **dns domain-lookup** command and how to use it to configure DNS lookup, see *Cisco Security Appliance Command Reference*.

After configuring the DNS lookup, make sure that the security appliance can ping the Cisco UCM with the configured FQDN.

If you have a CAPF service enabled and the Cisco UCM is not running on the Publisher, and the Publisher is configured with a FQDN instead of an IP address, you must also configure DNS lookup.

- Access-list rules must be configured to allow TFTP requests.

  Table 27-4 lists the access-list rule that must be configured for TFTP on the security appliance:

Table 27-4 Access List Rule for TFTP

Address	Port	Protocol	Description
TFTP Server	69	UDP	Allow incoming TFTP

Note 3804 is the default value for the CAPF Service. This default value should be modified if it is modified on the Cisco UCM. If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the access lists.

For information about configuring access-lists on the security appliance, see Access List Overview, page 18-1.

- If the phone proxy is deployed behind an existing firewall, access-list rules to permit signaling, TFTP, and media traffic to the phone proxy must be configured. If NAT is required for Cisco UCM, it must be configured on the security appliance, not on the existing firewall.

  Table 27-5 lists the ports that are required to be configured on the existing firewall:

Table 27-5 Port Configuration Requirements

Address	Port	Protocol	Description
Media Termination	1024-65535	UDP	Allow incoming SRTP
TFTP Server	69	UDP	Allow incoming TFTP
Cisco UCM	2443	TCP	Allow incoming secure SCCP
Cisco UCM	5061	TCP	Allow incoming secure SIP
CAPF Service (on Cisco UCM)	3804	TCP	Allow CAPF service for LSC provisioning

Note All these ports are configurable on the Cisco UCM, except for TFTP. These are the default values and should be modified if they are modified on the Cisco UCM. If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the access lists.

- If NAT is configured for the TFTP server, the NAT configuration must be configured prior to configuring the **tftp-server** command under the phone proxy.
- The Cisco UCM can be on a private network on the inside but you need to have a static mapping for the Cisco UCM on the security appliance to a public routable address.
- The following PAT configuration requirements must be met for the phone proxy:

– When the Skinny inspection global port is configured to use a non-default port, then you must configure the nonsecure port as the `global_sccp_port+443`.

  Therefore, if *global_sccp_port* is 7000, then the global secure SCCP port is 7443. Reconfiguring the port might be necessary when the phone proxy deployment has more than one Cisco UCM and they must share the interface IP address or a global IP address:

```
/* use the default ports for the first CUCM */
static (inside,outside) tcp interface 2000 10.0.0.1 2000
static (inside,outside) tcp interface 2443 10.0.0.1 2443
/* use non-default ports for the 2nd CUCM */
static (inside,outside) tcp interface 7000 10.0.0.2 2000
static (inside,outside) tcp interface 7443 10.0.0.2 2443
```

Note Both PAT configurations—for the nonsecure and secure ports—must be configured.

– When the IP phones must contact the CAPF on the Cisco UCM and the Cisco UCM is configured with static PAT (LCS provisioning is required), you must configure static PAT for the default CAPF port 3804.

Requirements to Support the 7960 and 7940 IP Phones

To support the 7960 and 7940 IP phones with the phone proxy, you must meet the following requirements:

- An LSC must be installed on these IP phones because they do not come pre installed with a MIC. Install the LSC on each phone before using them with the phone proxy to avoid opening the nonsecure SCCP port for the IP phones to register in nonsecure mode with the Cisco UCM.

  See the following document for the steps to install an LSC on IP phones:

  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/security/7_0_1/secugd/secucapf.html#wp1093518

Note If an IP phone already has an LSC installed on it from a different Cisco UCM cluster, delete the LSC from the different cluster and install an LSC from the current Cisco UCM cluster.

- The CAPF certificate must be imported onto the security appliance.
- The CTL file created on the security appliance must be created with a CAPF record-entry.
- The phone must be configured to use only the SCCP protocol because the SIP protocol does not support encryption on these IP phones.
- If LSC provisioning is done via the phone proxy, you must add an ACL to allow the IP phones to register with the Cisco UCM on the nonsecure port 2000.

Addressing Requirements for IP Phones on Multiple Interfaces

When IP phones reside on multiple interfaces, the phone proxy configuration must have the correct IP address set for the Cisco UCM in the CTL file.

See the following example topology for information about how to correctly set the IP address:

```
phones --- (dmz)-----|
                     |----- ASA PP --- (outside Internet) --- phones
phones --- (inside)--|
```

In this example topology, the following IP address are set:

- Cisco UCM on the inside interface is set to 10.0.0.5
- The DMZ network is 192.168.1.0/24
- The inside network is 10.0.0.0/24

The Cisco UCM is mapped with different global IP addresses from DMZ > outside and inside interfaces > outside interface.

In the CTL file, the Cisco UCM must have two entries because of the two different IP addresses. For example, if the static statements for the Cisco UCM are as follows:

```
static (inside,outside) 128.106.254.2 10.0.0.5
static (inside,dmz) 192.168.1.2 10.0.0.5
```

There must be two CTL file record entries for the Cisco UCM:

```
record-entry cucm trustpoint cucm_in_to_out address 128.106.254.2
record-entry cucm trustpoint cucm_in_to_dmz address 192.168.1.2
```

Supported Cisco UCM and IP Phones for the Phone Proxy

Cisco Unified Communications Manager

The following release of the Cisco Unified Communications Manager are supported with the phone proxy:

- Cisco Unified CallManager Version 4.x
- Cisco Unified CallManager Version 5.x
- Cisco Unified Communications Manager 6.x
- Cisco Unified Communications Manager 7.x

Cisco Unified IP Phones

The phone proxy supports these IP phone features:

- Enterprise features like conference calls on remote phones connected through the phone proxy
- XML services

Note The phone proxy supports only the features described in the list above. All other IP phone features not described by this list are unsupported by the phone proxy.

The phone proxy does not support displaying the lock icon on IP phone screens. IP phones display the lock icon on the phone screen during encrypted calls. Even though the lock icon is not displayed on the screen, the IP phone call is still encrypted because the phone proxy encrypts calls by default.

The following IP phones in the Cisco Unified IP Phones 7900 Series are supported with the phone proxy:

- 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 (SCCP protocol support only)
- 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 (SCCP protocol support only)
- Cisco Unified Wireless IP Phone 7921
- Cisco Unified Wireless IP Phone 7925

Note To support Cisco Unified Wireless IP Phone 7925, you must also configure MIC or LSC on the IP phone so that it properly works with the phone proxy.

- CIPC for softphones (CIPC versions with Authenticated mode only)

Note The Cisco IP Communicator is supported with the phone proxy VLAN Traversal in authenticated TLS mode.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

End-User Phone Provisioning

The phone proxy is a transparent proxy with respect to the TFTP and signaling transactions. If NAT is not configured for the Cisco UCM TFTP server, then the phone needs to be configured with the Cisco UCM cluster TFTP server address.

If NAT is configured for the Cisco UCM TFTP server, then the Cisco UCM TFTP server global address is configured as the TFTP server address on the phone.

- Option 1 (Recommended) – Stage the IP phones at corporate headquarters before sending them to the end users:
  - The phones register inside the network. IT ensures there are no issues with the phone configurations, image downloads, and registration.
  - If Cisco UCM cluster was in mixed mode, the CTL file should be erased before sending the phone to the end user.
  - Advantages of this option are:

    • Easier to troubleshoot and isolate problems with the network or phone proxy because you know whether the phone is registered and working with the Cisco UCM.

    • Better user experience because the phone does not have to download firmware from over a broadband connection, which can be slow and require the user to wait for a longer time.
- Option 2 – Send the new phone to the end user

– The user must be provided instructions to change the settings on phones with the appropriate Cisco UCM and TFTP server IP address.

In both options, deploying a remote IP phone behind a commercial Cable/DSL router with NAT capabilities is supported.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance.

See "Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-72". See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster

Step 1 Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phone must trust. The certificates are used in creating the CTL file.

You need to create trustpoints for each Cisco UCM (primary and secondary if a secondary Cisco UCM is used) and TFTP server in the network. The trustpoints need to be in the CTL file for the phones to trust the Cisco UCM.

a. Create a keypair that can be used for the trustpoints by entering the following command:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

b. Create the trustpoints for each Cisco UCM (primary and secondary) by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Entering these commands generates a self-signed certificate and specifies the keypair whose public key is being certified. This is the keypair created in substep a. Entering the **crypto ca enroll** command requests the certificate from the CA server and causes the security appliance to generate the certificate.

c. Create the trustpoint for the TFTP server by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Note You are only required to perform this step when the TFTP server resides on a different server from the Cisco UCM. See Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-69 for an example of this configuration.

d. When prompted to include the device serial number in the subject name, type **Y** to include the serial number or type **N** to exclude it.

e. When prompted to generate the self-signed certificate, type **Y**.

f. Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy.

- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002

See Importing Certificates from the Cisco UCM, page 27-18. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

g. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. See Importing Certificates from the Cisco UCM, page 27-18.

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 2 Create the CTL file that will be presented to the IP phones during the TFTP. The *address* here must be the translated or global address of the TFTP server or Cisco UCM if NAT is configured.

a. If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the security appliance. Add an entry for each of the outside interfaces on the security appliance into your DNS server, if such entries are not already present. Each security appliance outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup. Enable DNS lookups on your security appliance with the **dns domain-lookup** *interface_name* command (where the *interface_name* specifies the interface that has a route to your DNS server). Additionally, define your DNS server IP address on the security appliance; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).

Note You can enter the **dns domain-lookup** command multiple times to enable DNS lookup on multiple interfaces. If you enter multiple commands, the security appliance tries each interface in the order it appears in the configuration until it receives a response.

b. Create the CTL file instance by entering the following command:

```
hostname(config)# ctl-file ctl_name
```

c. Create the record entry for the TFTP server by entering the following command. Use the global or mapped IP address of the TFTP server.

```
hostname(config-ctl-file)# record-entry tftp trustpoint trustpoint_name address
TFTP_IP_address
```

d. Create the record entry for the each Cisco UCM (primary and secondary) by entering the following command. Use the global or mapped IP address of the Cisco UCM.

```
hostname(config-ctl-file)# record-entry cucm trustpoint trustpoint_name address
IP_address
```

e. (Optional) If LSC provisioning is or you have LSC enabled IP phones, create the record entry for CAPF by entering the following command:

```
hostname(config-ctl-file)# record-entry capf trustpoint trust_point address
```

f. Create the CTL file by entering the following command:

```
hostname(config-ctl-file)# no shutdown
```

When the file is created, it creates an internal trustpoint used by the phone proxy to sign the TFTP files. The trustpoint is named **_internal_PP**_*ctl-instance_filename*.

g. Save the certificate configuration to Flash memory by entering the following command:

```
hostname(config)# copy running-configuration startup-configuration
```

Step 3 Create the TLS proxy instance to handle the encrypted signaling.

a. Create the TLS proxy instance by entering the following command:

```
hostname(config)# tls-proxy proxy_name
```

b. Configure the server trustpoint and reference the internal trustpoint named **_internal_PP**_*ctl-instance_filename*:

```
hostname(config-tlsp)# server trust-point _internal_PP_ctl-instance_filename
```

Step 4 Configure the phone proxy instance.

a. Create the CTL file instance:

```
hostname(config)# phone-proxy phone_proxy_name
```

Note Only one phone proxy instance can be configured on the security appliance by using the **phone-proxy** command. See the *Cisco Security Appliance Command Reference* for information about the **phone-proxy** command.

b. Configure the media-termination address used by the phone-proxy for SRTP and RTP by entering the following command:

```
hostname(config-phone-proxy)# media-termination address ip_address
```

Note

- For the media termination address, you must select a publicly routable IP address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- Specifically, the media termination address cannot be the same as any security appliance interface IP address, cannot overlap with existing static NAT rules, and cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

c. Create the TFTP server using the actual internal address and specify the interface on which the TFTP server resides by entering the following command:

```
hostname(config-phone-proxy)# tftp-server address ip_address interface interface
```

d. Configure the TLS proxy instance created in Step 3 by entering the following command:

```
hostame(config-phone-proxy)# tls-proxy proxy_name
```

e. Configure the CTL file instance created in Step 2 by entering the following command:

```
hostname(config-phone-proxy)# ctl-file ctl_name
```

f. (Optional) If the operational environment has an external HTTP proxy to which the IP phones direct all HTTP request, enter the following command to configure a proxy server on the security appliance:

```
hostname(config-phone-proxy)# proxy-server address ip_address [listen_port] interface
ifc
```

You can configure only one proxy server while the phone proxy is in use; however, if the IP phones have already downloaded their configuration files after you have configured the proxy server, you must restart the IP phones so that they get the configuration file with the proxy server address in the file.

By default, the Phone URL Parameters configured under the Enterprise Parameters use an FQDN in the URLs. The parameters might need to be changed to use an IP address if the DNS lookup for the HTTP proxy does not resolve the FQDNs.

g. (Optional) To force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario, enter the following command:

```
hostname(config-phone-proxy)# cipc security-mode authenticated
```

See Phone Proxy Configuration for Cisco IP Communicator, page 27-24 for all requirements for using the phone proxy with CIPC.

h. (Optional) To preserve the settings configured on the Cisco UCM for each IP phone configured, enter the following command:

```
hostname(config-phone-proxy)# no disable service-settings
```

By default, the following settings are disabled on the IP phones:

- PC Port
- Gratuitous ARP
- Voice VLAN access
- Web Access
- Span to PC Port

Step 5 Enable the phone proxy with SIP and Skinny inspection.

a. Configure the secure Skinny class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 2443
```

Where *class_map_name* is the name of the Skinny class map.

b. Configure the secure SIP class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 5061
```

Where *class_map_name* is the name of the SIP class map.

c. Configure the policy map and attach the action to the class of traffic by entering the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect skinny phone-proxy pp_name
hostnae(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect sip phone-proxy pp_name
```

Where *classmap_name* is the name of the Skinny class map and the name for the SIP class map.

d. Enable the policy on the outside interface by entering the following command:

```
hostname(config)# service-policy policymap_name interface intf
```

Importing Certificates from the Cisco UCM

For the TLS proxy used by the phone proxy to complete the TLS handshake successfully, it needs to verify the certificates from the IP phone (and the Cisco UCM if doing TLS with Cisco UCM). To validate the IP phone certificate, we need the CA Manufacturer certificate which is stored on the Cisco UCM. Follow these steps to import the CA Manufacturer certificate to the security appliance.

Step 1 Go to the Cisco UCM Operating System Administration web page.

Step 2 Choose **Security > Certificate Management**.

Note Earlier versions of Cisco UCM have a different UI and way to locate the certificates. For example, in Cisco UCM version 4.x, certificates are located in the directory `C:\Program Files\Cisco\Certificates`. See your Cisco Unified Communications Manager (CallManager) documentation for information about locating certificates.

Step 3 Click Find and it will display all the certificates.

Step 4 Find the filename `Cisco_Manufacturing_CA`. This is the certificate need to verify the IP phone certificate. Click the .PEM file `Cisco_Manufacturing_CA.pem`. This will show you the certificate information and a dialog box that has the option to download the certificate.

Note If the certificate list contains more than one certificate with the filename `Cisco_Manufacturing_CA`, make you select the certificate `Cisco_Manufacturing_CA.pem`—the one with the .pem file extension.

Step 5 Click Download and save the file as a text file.

Step 6 On the security appliance, create a trustpoint for the Cisco Manufacturing CA and enroll via terminal by entering the following commands. Enroll via terminal because you will paste the certificate you downloaded in Step 4.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
```

Step 7 Authenticate the trustpoint by entering the following command:

```
hostname(config)# crypto ca authenticate trustpoint
```

Step 8 You are prompted to "Enter the base 64 encoded CA Certificate." Copy the .PEM file you downloaded in Step 4 and paste it at the command line. The file is already in base-64 encoding so no conversion is required. If the certificate is OK, you are prompted to accept it: "Do you accept this certificate? [yes/no]." Enter **yes**.

Note When you copy the certificate, make sure that you also copy also the lines with BEGIN and END.

Tip If the certificate is not ok, use the **debug crypto ca** command to show debug messages for PKI activity (used with CAs).

Step 9 Repeat the Step 1 through Step 8 for the next certificate. Table 27-6 shows the certificates that are required by the security appliance.

Table 27-6 Certificates Required by the Security Appliance for the Phone Proxy

Certificate Name	Required for...
CallManager	Authenticating the Cisco UCM during TLS handshake; only required for mixed-mode clusters.
Cisco_Manufacturing_CA	Authenticating IP phones with a Manufacturer Installed Certificate (MIC).
CAP-RTP-001	Authenticating IP phones with a MIC.
CAP-RTP-002	Authenticating IP phones with a MIC.
CAPF	Authenticating IP phones with an LSC.

Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster

When the phone proxy is being configured to run in mixed-mode clusters, you have the following options:

- If the cluster is in mixed mode, the user has the option to use the existing CTL file to install the trustpoints.
- If a CTL file exists for the cluster, copy the CTL file to Flash memory and configure the security appliance to read from that CTL file. When you copy the CTL file to Flash memory, do not name the file `CTLFile.tlv`.

Note For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the security appliance.

Step 1 Use an existing CTL file to install the trustpoints for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phones must trust. If you have an existing CTL file that contains the correct IP addresses of the entities (namely, the IP address that the IP phones use for the Cisco UCM or TFTP servers), you can be use it to create a new CTL file. Store a copy of the existing CTL file to Flash memory and rename it something other than `CTLFile.tlv` and continue to Step 2.

Or

Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phones must trust by performing the following substeps:

a. Create the trustpoints for each Cisco UCM (primary and secondary) by entering the following commands:

```
hostname(config)# crypto key generate rsa label keyname modulus 1024
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Entering these commands generates a self-signed certificate and specifies the keypair whose public key is being certified. This is the keypair created in substep a. Entering the **crypto ca enroll** command requests the certificate from the CA server and causes the security appliance to generate the certificate.

b. Create the trustpoint for the TFTP server by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Note You are only required to perform this step when the TFTP server resides on a different server from the Cisco UCM. See Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-69 for an example of this configuration.

c. When prompted to include the device serial number in the subject name, type **Y** to include the serial number or type **N** to exclude it.

d. When prompted to generate the self-signed certificate, type **Y**.

e. Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy.

- CallManager
- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002

See Importing Certificates from the Cisco UCM, page 27-18. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

f. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. See Importing Certificates from the Cisco UCM, page 27-18.

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 2 Create the CTL file that will be presented to the phones during the TFTP. The *address* here must be the translated or global address of the TFTP server or Cisco UCM if NAT is configured.

a. If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the security appliance. Add an entry for each of the outside interfaces on the security appliance into your DNS server, if such entries are not already present. Each security appliance outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup. Enable DNS lookups on your security appliance with the command **dns domain-lookup** *interface_name* (where the *interface_name* specifies the interface that has a route to your DNS server). Additionally, define your DNS server IP address on the security appliance; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).

Note You can enter the **dns domain-lookup** command multiple times to enable DNS lookup on multiple interfaces. If you enter multiple commands, the security appliance tries each interface in the order it appears in the configuration until it receives a response.

b. Create the CTL file instance by entering the following command:

```
hostname(config)# ctl-file ctl_name
```

c. If you are using an existing CTL file, use the trustpoints that are already in existing CTL file stored in Flash memory by entering the following command:

```
hostname(config-ctl-file)# cluster-ctl-file filename_path
```

Where the existing CTL file was saved to Flash memory with a filename other than `CTLFile.tlv`; for example, `old_ctlfile.tlv`.

Note Complete the remaining items in this step if you are creating a new CTL file instance or you want to add more entries to an existing CTL file.

d. Create the record entry for the TFTP server by entering the following command:

```
hostname(config-ctl-file)# record-entry tftp trustpoint trustpoint_name address
TFTP_IP_address
```

e. Create the record entry for the each Cisco UCM (primary and secondary) by entering the following command:

```
hostname(config-ctl-file)# record-entry cucm trustpoint trustpoint_name address
IP_address
```

f. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, create the record entry for CAPF by entering the following command:

```
hostname(config-ctl-file)# record-entry capf trustpoint trustpoint_name address
IP_address
```

g. Create the CTL file by entering the following command:

```
hostname(config-ctl-file)# no shutdown
```

When the file is created, it creates an internal trustpoint used by the phone proxy to sign the TFTP files. The trustpoint is named **_internal_PP**_*ctl-instance_filename*.

h. Save the certificate configuration to Flash memory by entering the following command:

```
hostname(config)# copy running-configuration startup-configuration
```

Step 3 Create the TLS proxy instance to handle the encrypted signaling.

For mixed mode clusters, there might be IP phones that are already configured as encrypted so it requires TLS to the Cisco UCM. You must configure the LDC issuer for the TLS proxy. For more information about any of the following steps, see TLS Proxy for Encrypted Voice Inspection, page 27-43.

a. Create the necessary RSA key pairs by entering the following commands:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

Where the *key-pair-label* is the LDC signer key and the key for the IP phones.

b. Create an internal local CA to sign the LDC for Cisco IP phones by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# proxy-ldc-issuer
hostname(config-ca-trustpoint)# fqdn fqdn
hostname(config-ca-trustpoint)# subject-name X.500_name
hostname(config-ca-trustpoint)# keypair keypair
hostname(config)# crypto ca enroll ldc_server
```

Where the *trustpoint-name*, *fqdn*, *X.500_name*, *keypair*, and *trustpoint* are for the LDC.

c. Create the TLS proxy instance by entering the following commands:

```
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point _internal_PP_ctl-instance_filename
hostname(config-tlsp)# client ldc issuer ca_tp_name
hostname(config-tlsp)# client ldc keypair key_label
hostname(config-tlsp)# client cipher-suite cipher-suite
```

Where the *ca_tp_name* specifies the local CA trustpoint to issue client dynamic certificates and the *key_label* Specifies the RSA keypair to be used by client dynamic certificates.

d. Export the local CA certificate and install it as a trusted certificate on the Cisco Unified Call Manager server by performing one of the following actions:

- Use the following command to export the certificate if a trustpoint with proxy-ldc-issuer is used as the signer of the dynamic certificates:

  ```
  hostname(config)# crypto ca export trustpoint identity-certificate
  ```

- For the embedded local CA server LOCAL-CA-SERVER, use the following command to export its certificate:

  ```
  hostname(config)# show crypto ca server certificates
  ```

e. Save the output to a file and import the certificate on the Cisco Unified Call Manager. For more information, see the Cisco Unified Call Manager document:
http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040848

f. Use the Display Certificates function in the Cisco Unified Call Manager software to verify the installed certificate:

http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040354

Step 4 Configure the phone proxy instance.

a. Create the CTL file instance:

```
hostname(config)# phone-proxy phone_proxy_name
```

b. Configure the media-termination address used by the phone-proxy for SRTP and RTP by entering the following command:

```
hostname(config-phone-proxy)# media-termination address ip_address
```

Note

- For the media termination address, you must select a publicly routable IP address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- Specifically, the media termination address cannot be the same as any security appliance interface IP address, cannot overlap with existing static NAT rules, and cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

c. Create the TFTP server using the actual internal address and specify the interface on which the TFTP server resides by entering the following command:

```
hostname(config-phone-proxy)# tftp-server address ip_address interface interface
```

d. Configure the TLS proxy instance created in Step 3 by entering the following command:

```
hostame(config-phone-proxy)# tls-proxy proxy_name
```

e. Configure the CTL file instance created in Step 2 by entering the following command:

```
hostname(config-phone-proxy)# ctl-file ctl_name
```

f. Configure the mode of the cluster to be mixed mode because the default is nonsecure.

```
hostname(config-phone-proxy)# cluster-mode mixed
```

g. (Optional) If the operational environment has an external HTTP proxy to which the IP phones direct all HTTP request, enter the following command to configure a proxy server on the security appliance:

proxy-server address *ip_address* [*listen_port*] **interface** *ifc*

You can configure only one proxy server while the phone proxy is in use; however, if the IP phones have already downloaded their configuration files after you have configured the proxy server, you must restart the IP phones so that they get the configuration file with the proxy server address in the file.

By default, the Phone URL Parameters configured under the Enterprise Parameters use an FQDN in the URLs. The parameters might need to be changed to use an IP address if the DNS lookup for the HTTP proxy does not resolve the FQDNs.

h. (Optional) To force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario, enter the following command:

```
hostname(config-phone-proxy)# cipc security-mode authenticated
```

See Phone Proxy Configuration for Cisco IP Communicator, page 27-24 for all requirements for using the phone proxy with CIPC.

i. (Optional) To preserve the settings configured on the Cisco UCM for each IP phone configured, enter the following command:

```
hostname(config-phone-proxy)# no disable service-settings
```

By default, the following settings are disabled on the IP phones:

- PC Port
- Gratuitous ARP
- Voice VLAN access
- Web Access
- Span to PC Port

Step 5 Enable the phone proxy with SIP and Skinny inspection.

a. Configure the secure Skinny class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 2443
```

b. Configure the secure SIP class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 5061
```

c. Configure the policy map and attach the action to the class of traffic by entering the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect skinny phone-proxy pp_name
hostnae(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect sip phone-proxy pp_name
```

d. Enable the policy on the outside interface by entering the following command:

```
hostname(config)# service-policy policymap_name interface intf
```

Phone Proxy Configuration for Cisco IP Communicator

To configure Cisco IP Communicator (CIPC) with the phone proxy, you must meet the following requirements:

- Include the **cipc security-mode authenticated** command under the **phone-proxy** command.
- Create an ACL to allow CIPC to register with the Cisco UCM in nonsecure mode.
- Configure null-sha1 as one of the SSL encryption ciphers.

Current versions of Cisco IP Communicator (CIPC) support authenticated mode and perform TLS signaling but not voice encryption. Therefore, you must include the following command when configuring the phone proxy instance:

cipc security-mode authenticated

Because CIPC requires an LSC to perform the TLS handshake, CIPC needs to register with the Cisco UCM in nonsecure mode using cleartext signaling. To allow the CIPC to register, create an ACL that allows the CIPC to connect to the Cisco UCM on the nonsecure SIP/SCCP signalling ports (5060/2000).

CIPC uses a different cipher when doing the TLS handshake and requires the null-sha1 cipher and SSL encryption be configured. To add the null-shal cipher, use the **show run all ssl** command to see the output for the **ssl encryption** command and add **null-shal** to the end of the SSL encryption list.

Note When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Prefences > Network tab > Use this Device Name field) or Administrators resetting the devide name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the adsecurity appliance to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

Configuring Linksys Routers for UDP Port Forwarding

When IP phones are behind a NAT-capable router, the router can be configured to forward the UDP ports to the IP address of the IP phone. Specifically, configure the router for UDP port forwarding when an IP phone is failing during TFTP requests and the failure is due to the router dropping incoming TFTP data packets. Configure the router to enable UDP port forwarding on port 69 to the IP phone.

As an alternative of explicit UDP forwarding, some Cable/DSL routers require you to designate the IP phone as a DMZ host. For Cable/DSL routers, this host is a special host that receives all incoming connections from the public network.

When configuring the phone proxy, there is no functional difference between an IP phone that has UDP ports explicitly forwarded or an IP phone designated as a DMZ host. The choice is entirely dependent upon the capabilities and preference of the end user.

Configuring Your Router

Your firewall/router needs to be configured to forward a range of UDP ports to the IP phone. This will allow the IP phone to receive audio when you make/receive calls.

Note Different Cable/DSL routers have different procedures for this configuration. Furthermore most NAT-capable routers will only allow a given port range to be forwarded to a single IP address

The configuration of each brand/model of firewall/router is different, but the task is the same. For specific instructions for your brand and model of router, please contact the manufacturer's website.

Linksys Routers

Step 1 From your web browser, connect to the router administrative web page. For Linksys, this is typically something like `http://192.168.1.1`.

Step 2 Click Applications & Gaming or the Port Forwarding tab (whichever is present on your router).

Step 3 Locate the table containing the port forwarding data and add an entry containing the following values:

Table 27-7 Port Forwarding Values to Add to Router

Application	Start	End	Protocol	IP Address	Enabled
IP phone	1024	65535	UDP	*Phone IP address*	**Checked**
TFTP	69	69	UDP	*Phone IP address*	**Checked**

Step 4 Click Save Settings. Port forwarding is configured.

About Rate Limiting TFTP Requests

In a remote access scenario, we recommend that you configure rate limiting of TFTP requests because any IP phone connecting through the Internet is allowed to send TFTP requests to the TFTP server.

To configure rate limiting of TFTP requests, configure the **police** command in the Modular Policy Framework. See the *Cisco Security Appliance Command Reference* for information about using the **police** command.

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow can take over the entire resource. When traffic exceeds the maximum rate, the security appliance drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Rate Limiting Configuration Example

The following example describes how you configure rate limiting for TFTP requests by using the **police** command and the Modular Policy Framework.

Begin by determining the conformance rate that is required for the phone proxy. To determine the conformance rate, use the following formula:

```
X * Y * 8
```

Where

X = requests per second

Y = size of each packet, which includes the L2, L3, and L4 plus the payload

Therefore, if a rate of 300 TFTP requests/second is required, then the conformance rate would be calculated as follows:

```
300 requests/second * 80 bytes * 8 = 192000
```

The example configuration below shows how the calculated conformance rate is used with the **police** command:

```
access-list tftp extended permit udp any host 192.168.0.1 eq tftp

class-map tftpclass
    match access-list tftp

policy-map tftpmap
    class tftpclass
    police output 192000

service-policy tftpmap interface inside
```

About ICMP Traffic Destined for the Media Termination Address

To control which hosts can ping the media termination address, use the **icmp** command and apply the access rule to the outside interface on the security appliance.

Any rules for ICMP access applied to the outside interface apply to traffic destined for the media termination address.

For example, use the following command to deny ICMP pings from any host destined for the media termination address:

icmp deny any outside

Troubleshooting the Phone Proxy

This section includes the following topics:

Debugging Information from the Security Appliance

This section describes how to use the **debug**, **capture**, and **show** commands to obtain debugging information for the phone proxy. See the *Cisco Security Appliance Command Reference* for detailed information about the syntax for these commands.

Table 27-8 lists the **debug** commands to use with the phone proxy.

Table 27-8 Security Appliance Debug Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show error and event messages for TLS proxy inspection.	**debug inspect tls-proxy** [**events** | **errors**]	Use this command when your IP phone has successfully downloaded all TFTP files but is failing to complete the TLS handshake with the TLS proxy configured for the phone proxy.
To show error and event messages of media sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy media** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is experiencing call failures or audio problems.

Table 27-8 Security Appliance Debug Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show error and event messages of signaling sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy signaling** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is failing to register with the Cisco UCM or if you are experiencing call failure.
To show error and event messages of TFTP inspection, including creation of the CTL file and configuration file parsing.	**debug phone-proxy tftp** [**events** | **errors**]	
To show debug messages for SIP application inspection.	**debug sip**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.
To show debug messages for SCCP (Skinny) application inspection.	**debug skinny**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.

Table 27-9 lists the capture commands to use with the phone proxy. Use the **capture** command on the appropriate interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation.

Table 27-9 Security Appliance Capture Commands to Use with the Phone Proxy

To	Use the Command	Notes
To capture packets on the security appliance interfaces.	**capture** *capture_name* **interface** *interface_name*	Use this command if you are experiencing any problems that might require looking into the packets. For example, if there is a TFTP failure and the output from the **debug** command does not indicate the problem clearly, run the **capture** command on the interface on which the IP phone resides and the interface on which the TFTP server resides to see the transaction and where the problem could be.
To capture data from the TLS proxy when there is a non-secure IP phone connecting to the phone proxy on the inside interface.	**capture** *capture_name* **packet-length** *bytes* **interface inside buffer** *buf_size*	

Table 27-9 Security Appliance Capture Commands to Use with the Phone Proxy

To	Use the Command	Notes
To capture encrypted data from the TLS proxy when there are secure IP phones connecting to the phone proxy on the inside interface.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface inside**	
To capture encrypted inbound and outbound data from the TLS proxy on one or more interfaces.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface** *interface_name*	If signaling fails, you might require capturing decrypted packets to see the contents of the SIP and SCCP signaling message. Use the **type tls-proxy** option in the **capture** command.

Table 27-10 lists the **show** commands to use with the phone proxy.

Table 27-10 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the packets or connections dropped by the accelerated security path.	**show asp drop**	Use this command to troubleshoot audio quality issues with the IP phones or other traffic issues with the phone proxy. In addition to running this command, get call status from the phone to check for any dropped packets or jitter. See Debugging Information from IP Phones, page 27-31.
To show the classifier contents of the accelerated security path for the specific classifier domain.	**show asp table classify domain** *domain_name*	If the IP phones are not downloading TFTP files, use this command to check that the classification rule for the domain `inspect-phone-proxy` is set for hosts to the configured TFTP server under the phone proxy instance. If the IP phones are failing to register, use this command to make sure there is a classification rule for the domain `app-redirect` set for the IP phones that cannot register.

Table 27-10 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes

To show the connections that are to the security appliance or from the security appliance, in addition to through-traffic connections.	**show conn all**	If you are experiencing problems with audio, use this command to make sure that there are connections opened from the IP phone to the media termination address.
		
		Note Use the **show conn** command with following options to display TFTP connections that have replicated (unused) connections:
		
		`hostname# show conn | include p`
		
		The output for the TFTP connections should have a "p" flag at the end:
		
		`UDP out 64.169.58.181:9014 in 192.168.200.101:39420 idle 0:01:51 bytes 522 flags p`
		
		Using this command shows that the phone proxy has connections that are going through "inspect-phone-proxy", which inspects TFTP connections. Using this command verifies that the TFTP requests are being inspected because the p flag is there.

To show the logs in the buffer and logging settings.	**show logging**	Before entering the **show logging** command, enable the **logging buffered** command so that the **show logging command** displays the current message buffer and the current settings. Use this command to determine if the phone proxy and IP phones are successfully completing the TLS handshake. **Note** Using the **show logging** command is useful for troubleshooting many problems where packets might be denied or there are translation failures.
To show the corresponding media sessions stored by the phone proxy.	**show phone-proxy media-sessions**	Use this command to display output from successful calls. Additionally, use this command to troubleshoot problems with IP phone audio, such as one-way audio.

Table 27-10 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the IP phones capable of Secure mode stored in the database.	**show phone-proxy secure-phones**	For any problems, make sure there is an entry for the IP phone in this output and that the port for this IP phone is non-zero, which indicates that it has successfully registered with the Cisco UCM.
To show the corresponding signaling sessions stored by the phone proxy.	**show phone-proxy signaling-sessions**	Use this command to troubleshoot media or signaling failure.
To show the configured service policies.	**show service-policy**	Use this command to show statistics for the service policy.
To show active TLS proxy sessions related to the phone proxy.	**show tls-proxy sessions**	If the IP phone has failed to register, use this command to see if the IP phone has successfully completed the handshake with the TLS proxy configured for the phone proxy.

Debugging Information from IP Phones

On the IP phone, perform the following actions:

- Check the Status messages on the IP phone by selecting the **Settings** button > Status > Status Messages and selecting the status item that you want to view.
- Collect the call-statistics data from the IP phone by selecting the **Settings** button > Status > Call Statistic. Data like the following displays:

```
RxType: G.729              TxType: G.729
RxSize:   20 ms            TxSize:  20 ms
RxCnt: 0                   TxCnt: 014174
AvgJtr:   10               MaxJtr:  59
RxDisc: 0000               RxLost: 014001
```

- Check the Security settings on the IP phone by selecting the **Settings** button > Security Configuration. Settings for web access, Security mode, MIC, LSC, CTL file, trust list, and CAPF appear. Under Security mode, make sure the IP phone is set to Encrypted.
- Check the IP phone to determine which certificates are installed on the phone by selecting the **Settings** button > Security Configuration > Trust List. In the trustlist, verify the following:
  - Make sure that there is an entry for each entity that the IP phone will need to contact. If there is a primary and backup Cisco UCM, the trustlist should contain entries for each Cisco UCM.
  - If the IP phone needs an LSC, the record entry should contain a CAPF entry.
  - Make sure that the IP addresses listed for each entry are the mapped IP addresses of the entities that the IP phone can reach.
- Open a web browser and access the IP phone console logs at the URL `http://IP_phone_IP address`. The device information appears in the page. In the Device Logs section in the left pane, click Console Logs.

IP Phone Registration Failure

The following errors can make IP phones unable to register with the phone proxy:

TFTP Auth Error Displays on IP Phone Console

Problem The IP phone displays the following Status message:

```
TFTP Auth Error
```

Solution This Status message can indicate a problem with the IP phone CTL file.

To correct problems with the IP phone CTL file, perform the following:

Step 1 From the IP phone, select the **Setting** button > Security Configuration > Trust List. Verify that each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—has its own entry in the trustlist and that each entity IP address is reachable by the IP phone.

Step 2 From the security appliance, verify that the CTL file for the phone proxy contains one record entry for each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—by entering the following command:

```
hostname# show running-config all ctl-file [ctl_name]
```

Each of these record entries creates one entry on the IP phone trustlist. The phone proxy creates one entry internally with the function CUCM+TFTP.

Step 3 In the CTL file, verify that each IP address is the global or mapped IP address of the entity. If the IP phones are on multiple interfaces, additional addressing requirements apply. See Addressing Requirements for IP Phones on Multiple Interfaces, page 27-11.

Configuration File Parsing Error

Problem When the security appliance receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Enter the following URL in a web browser to obtain the IP phone configuration file from the Cisco Unified CM Administration console:

```
http://<cucm_ip>:6970/<config_file_name>
```

For example, if the Cisco UCM IP address is 128.106.254.2 and the IP phone configuration file name is SEP000100020003.cnf.xml, enter:

```
http://128.106.254.2:6970/SEP000100020003.cnf.xml
```

Step 2 Save this file, open a case with TAC and send them this file and the output from running the **debug phone-proxy tftp** command on the security appliance.

Configuration File Parsing Error: Unable to Get DNS Response

Problem When the security appliance receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Callback required for parsing config file
PP: Unable to get dns response for id 7
PP: Callback, error modifying config file
```

The error indicates that the Cisco UCM is configured as an FQDN and the phone proxy is trying to do a DNS lookup but failed to get a response.

Solution

Step 1 Verify that DNS lookup is configured on the security appliance.

Step 2 If DNS lookup is configured, determine whether you can ping the FQDN for the Cisco UCM from the security appliance.

Step 3 If security appliance cannot ping the Cisco UCM FQDN, check to see if there is a problem with the DNS server.

Step 4 Additionally, use the **name** command to associate a name with an IP address with the FQDN. See the *Cisco Security Appliance Command Reference* for information about using the **name** command.

Non-configuration File Parsing Error

Problem The security appliance receives a file other than an IP phone configuration file from the Cisco UCM and attempts to parse it. The following error appears in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49357 requesting SK72f64050-7ad5-4b47-9bfa-5e9ad9cd4aa9.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution The phone proxy should parse only the IP phone configuration file. When the phone proxy TFTP state gets out of state, the phone proxy cannot detect when it is attempting to parse a file other than the IP phone configuration file and the error above appears in the security appliance output from the **debug phone-proxy tftp** command.

Perform the following actions to troubleshoot this problem:

Step 1 Reboot the IP phone.

Step 2 On the security appliance, enter the following command to obtain the error information from the first TFTP request to the point where the first error occurred.

```
hostname# debug phone-proxy tftp
```

Step 3 Capture the packets from the IP phone to the security appliance. Make sure to capture the packets on the interface facing the IP phone and the interface facing the Cisco UCM. See Debugging Information from the Security Appliance, page 27-27.

Step 4 Save this troubleshooting data, open a case with TAC and give them this information.

Cisco UCM Does Not Respond to TFTP Request for Configuration File

Problem When the security appliance forwards the TFTP request to the Cisco UCM for the IP phone configuration file, the Cisco UCM does not respond and the following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Determine why the Cisco UCM is not responding to the TFTP request by performing the following troubleshooting actions:

- Use the Cisco UCM to ping the security appliance inside interface when PAT is configured for the outside interface so that the IP phone IP address is uses NAT for the security appliance inside interface IP address.
- Use the Cisco UCM to ping the IP phone IP address when NAT and PAT are not configured.

Step 2 Verify that the security appliance is forwarding the TFTP request. Capture the packets on the interface between the security appliance and Cisco UCM. See Debugging Information from the Security Appliance, page 27-27.

IP Phone Does Not Respond After the Security Appliance Sends TFTP Data

Problem When the security appliance receives a TFTP request from the IP phone for the CTL file and forwards the data to the IP phone, the phone might not see the data and the TFTP transaction fails.

The following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: opened 0x214b27a
PP: Data Block 1 forwarded from 168.215.146.220/20168 to 68.207.118.9/33606 ingress ifc
outside
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
```

Solution Perform the following actions to determine why the IP phone is not responding and to troubleshoot the problem:

Step 1 Verify that the security appliance is forwarding the TFTP request by entering the following command to capture the packets on the interface between the security appliance and the IP phone:

```
hostname# capture out interface outside
```

See the *Cisco Security Appliance Command Reference* for more information about using the **capture** command.

Step 2 If the IP phone is behind a router, the router might be dropping the data. Make sure UDP port forwarding is enabled on the router.

Step 3 If the router is a Linksys router, see Configuring Linksys Routers for UDP Port Forwarding, page 27-25 for information on the configuration requirements.

IP Phone Requesting Unsigned File Error

Problem The IP phone should always request a signed file. Therefore, the TFTP file being requested always has the .SGN extension.

When the IP phone does not request a signed file, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
Error: phone requesting for unsigned config file
```

Solution Most likely, this error occurs because the IP phone has not successfully installed the CTL file from the security appliance.

Determine whether the IP phone has successfully downloaded and installed the CTL file from the security appliance by checking the Status messages on the IP phone. See Debugging Information from IP Phones, page 27-31 for information.

IP Phone Unable to Download CTL File

Problem The IP phone Status message indicates it cannot download its CTL file and the IP phone cannot be converted to Secure (encrypted) mode.

Solution If the IP phone did not have an existing CTL file, check the Status messages by selecting the **Settings** button > Status > Status Messages. If the list contains a Status message indicating the IP phone encountered a CTL File Auth error, obtain the IP phone console logs, open a TAC case, and send them the logs.

Solution This error can appear in the IP phone Status messages when the IP phone already has an existing CTL file.

Step 1 Check the IP phone to see if a CTL file already exists on it. This can occur if the IP phone previously registered with a mixed mode cluster Cisco UCM. On the IP phone, select the **Settings** button > Security Configuration > CTL file.

Step 2 Erase the existing CTL file by selecting the **Settings** button > Security Configuration > CTL file > Select. Press **# on the keypad and select Erase.

Solution Problems downloading the CTL file might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

- The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- The IP address cannot be the same as any of the security appliance interface IP addresses.
- The IP address cannot overlap with existing static NAT rules.
- The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

IP Phone Registration Failure from Signaling Connections

Problem The IP phone is unable to complete the TLS handshake with the phone proxy and download its files using TFTP.

Solution

Step 1 Determine if the TLS handshake is occurring between the phone proxy and the IP phone, perform the following:

a. Enable logging with the following command:

```
hostname(config)# logging buffered debugging
```

b. To check the output from the syslogs captured by the **logging buffered** command, enter the following command:

```
hostname# show logging
```

The syslogs will contain information showing when the IP phone is attempting the TLS handshake, which happens after the IP phone downloads its configuration file.

Step 2 Determine if the TLS proxy is configured correctly for the phone proxy:

a. Display all currently running TLS proxy configurations by entering the following command:

```
hostname# show running-config tls-proxy
    tls-proxy proxy
    server trust-point _internal_PP_<ctl_file_instance_name>
    client ldc issuer ldc_signer
    client ldc key-pair phone_common
    no client cipher-suite
hostname#
```

b. Verify that the output contains the **server trust-point** command under the **tls-proxy** command (as shown in substep a.).

If you are missing the **server trust-point** command, modify the TLS proxy in the phone proxy configuration.

See Step 3 in Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster, page 27-14, or Step 3 in Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster, page 27-19.

Having this command missing from the TLS proxy configuration for the phone proxy will cause TLS handshake failure.

Step 3 Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

a. Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 27-31 for information about checking the IP phone to determine if it has MIC installed on it.

b. Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-6, Certificates Required by the Security Appliance for the Phone Proxy, for information.

c. Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-18.

Step 4 If the steps above fail to resolve the issue, perform the following actions to obtain additional troubleshooting information for Cisco Support.

a. Enter the following commands to capture additional debugging information for the phone proxy:

```
hostname# debug inspect tls-proxy error
```

```
hostname# show running-config ssl
hostname(config) show tls-proxy tls_name session host host_addr detail
```

b. Enable the **capture** command on the inside and outside interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation. See the *Cisco Security Appliance Command Reference* for information.

Problem The TLS handshake succeeds, but signaling connections are failing.

Solution Perform the following actions:

- Check to see if SIP and Skinny signaling is successful by using the following commands:
  - **debug sip**
  - **debug skinny**
- If the TLS handshake is failing and you receive the following syslog, the SSL encryption method might not be set correctly:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097
```

Set the correct ciphers by completing the following procedure:

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the *Cisco Security Appliance Command Reference* for more information about setting ciphers with the **ssl encryption** command.

SSL Handshake Failure

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the security appliance syslogs:

```
%ASA-7-725014: SSL lib error. Function: SSL3_READ_BYTES Reason: ssl handshake failure
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_CERTIFICATE Reason: no certificate
returned
%ASA-6-725006: Device failed SSL handshake with outside client:72.146.123.158/30519
%ASA-3-717009: Certificate validation failed. No suitable trustpoints found to validate
certificate serial number: 62D06172000000143FCC, subject name:
cn=CP-7962G-SEP002155554502,ou=EVVBU,o=Cisco Systems Inc.
```

```
%ASA-3-717027: Certificate chain failed validation. No suitable trustpoint was found to
validate chain.
```

Solution

Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 27-31 for information about checking the IP phone to determine if it has MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-6, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-18.

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the security appliance syslogs:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097
```

Solution the SSL encryption method might not be set correctly. Set the correct ciphers by completing the following procedure:

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the *Cisco Security Appliance Command Reference* for more information about setting ciphers with the **ssl encryption** command.

Certificate Validation Errors

Problem Errors in the security appliance log indicate that certificate validation errors occurred.

Entering the **show logging asdm** command, displayed the following errors:

```
3|Jun 19 2008 17:23:54|717009: Certificate validation failed. No suitable trustpoints
found to validate
certificate serial number: 348FD2760000000E6E27, subject name:
cn=CP-7961G-SEP001819A89CC3,ou=EVVBU,o=Cisco Systems Inc.
```

Solution

In order for the phone proxy to authenticate the MIC provided by the IP phone, it needs the Cisco Manufacturing CA (MIC) certificate imported into the security appliance.

Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. The certificate information is shown under the Security Configuration menu. See Debugging Information from IP Phones, page 27-31 for information about checking the IP phone to determine if it has the MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-6, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-18.

Media Termination Address Errors

Problem Entering the **media-termination address** command displays the following errors:

```
hostname(config-phone-proxy)# media-termination address ip_address
ERROR: Failed to apply IP address to interface Virtual254, as the network overlaps with
interface GigabitEthernet0/0. Two interfaces cannot be in the same subnet.
ERROR: Failed to set IP address for the Virtual interface
ERROR: Could not bring up Phone proxy media termination interface
ERROR: Failed to find the HWIDB for the Virtual interface
```

Solution Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
asa2(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

- The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- The IP address cannot be the same as any of the security appliance interface IP addresses.
- The IP address cannot overlap with existing static NAT rules.
- The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

Audio Problems with IP Phones

The following audio errors can occur when the IP phones connecting through the phone proxy.

Media Failure for a Voice Call

Problem The call signaling completes but there is one way audio or no audio.

Solution

- Problems with one way or no audio might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
asa2(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

  Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

  - The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
  - The IP address cannot be the same as any of the security appliance interface IP addresses.
  - The IP address cannot overlap with existing static NAT rules.
  - The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
  - For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

- If the media-termination address meets the requirements, determine whether the IP address is reachable by all IP phones.
- If the IP address is set correctly and it is reachable by all IP phones, check the call statistics on an IP phone (see Debugging Information from IP Phones, page 27-31) and determine if there are Rcvr packets and Sender packets on the IP phone, or if there are any Rcvr Lost or Discarded packets.

Saving SAST Keys

Site Administrator Security Token (SAST) keys on the security appliance can be saved in the event a recovery is required due to hardware failure and a replacement is required. The following steps shows how to recover the SAST keys and use them on the new hardware.

The SAST keys can be seen via the **show crypto key mypubkey rsa** command. The SAST keys are associated with a trustpoint that is labeled **_internal**_*ctl-file_name*_**SAST**_*X* where *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to N-1 where N is the number of SASTs configured for the CTL file (the default is 2).

Step 1 On the security appliance, export all the SAST keys in PKCS-12 format by using the **crypto ca export** command:

```
hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH
---End - This line not part of the pkcs12---

hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

mGF/hfDDNAICBAA=

---End - This line not part of the pkcs12---
hostname(config)#
```

Note Save this output somewhere secure.

Step 2 Import the SAST keys to a new security appliance.

a. To import the SAST key, enter the following command:

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase
```

Where *trustpoint* is **_internal**_*ctl-file_name*_**SAST**_*X* and *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to 4 depending on what you exported from the security appliance.

b. Using the PKCS-12 output you saved in Step 1, enter the following command and paste the output when prompted:

```
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

muMiZ6eClQICBAA=
hostname(config)# quit
```

```
INFO: Import PKCS12 operation completed successfully
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

mGF/hfDDNAICBAA=
hostname(config)# quit
INFO: Import PKCS12 operation completed successfully
hostname(config)#
```

Step 3 Create the CTL file instance on the new security appliance using the same name as the one used in the SAST trustpoints created in Step 2 by entering the following commands. Create trustpoints for each Cisco UMC (primary and secondary).

```
hostname(config)# ctl-file ctl_name
hostname(config-ctl-file)# record-entry cucm trustpoint trust_point address address
hostname(config-ctl-file)# record-entry capf trustpoint trust_point address address
hostname(config-ctl-file)# no shutdown
```

TLS Proxy for Encrypted Voice Inspection

This section describes TLS proxy for encrypted voice inspection. This section includes the following topics:

Overview

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

The security appliance acts as a TLS proxy between the Cisco IP Phone and Cisco Unified CallManager. The proxy is transparent for the voice calls between the phone and the Cisco Unified CallManager. Cisco IP Phones download a Certificate Trust List from the Cisco Unified CallManager before registration which contains identities (certificates) of the devices that the phone should trust, such as TFTP servers and Cisco Unified CallManager servers. To support server proxy, the CTL file must contain the certificate that the security appliance creates for the Cisco Unified CallManagers. To proxy calls on behalf of the Cisco IP Phone, the security appliance presents a certificate that the Cisco Unified CallManager 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 Unified CallManager. For background and detailed description of Cisco Unified CallManager security, see the Cisco Unified CallManager document.

TLS proxy applies to the encryption layer and must be configured with an application layer protocol inspection. You should be familiar with the inspection features on the ASA security appliance, especially Skinny and SIP inspection. For more information on deployment topologies and configuration, refer to the *Cisco Security Appliance Command Line Configuration Guide*.

Table 27-3 shows the default and maximum TLS session details by platform.

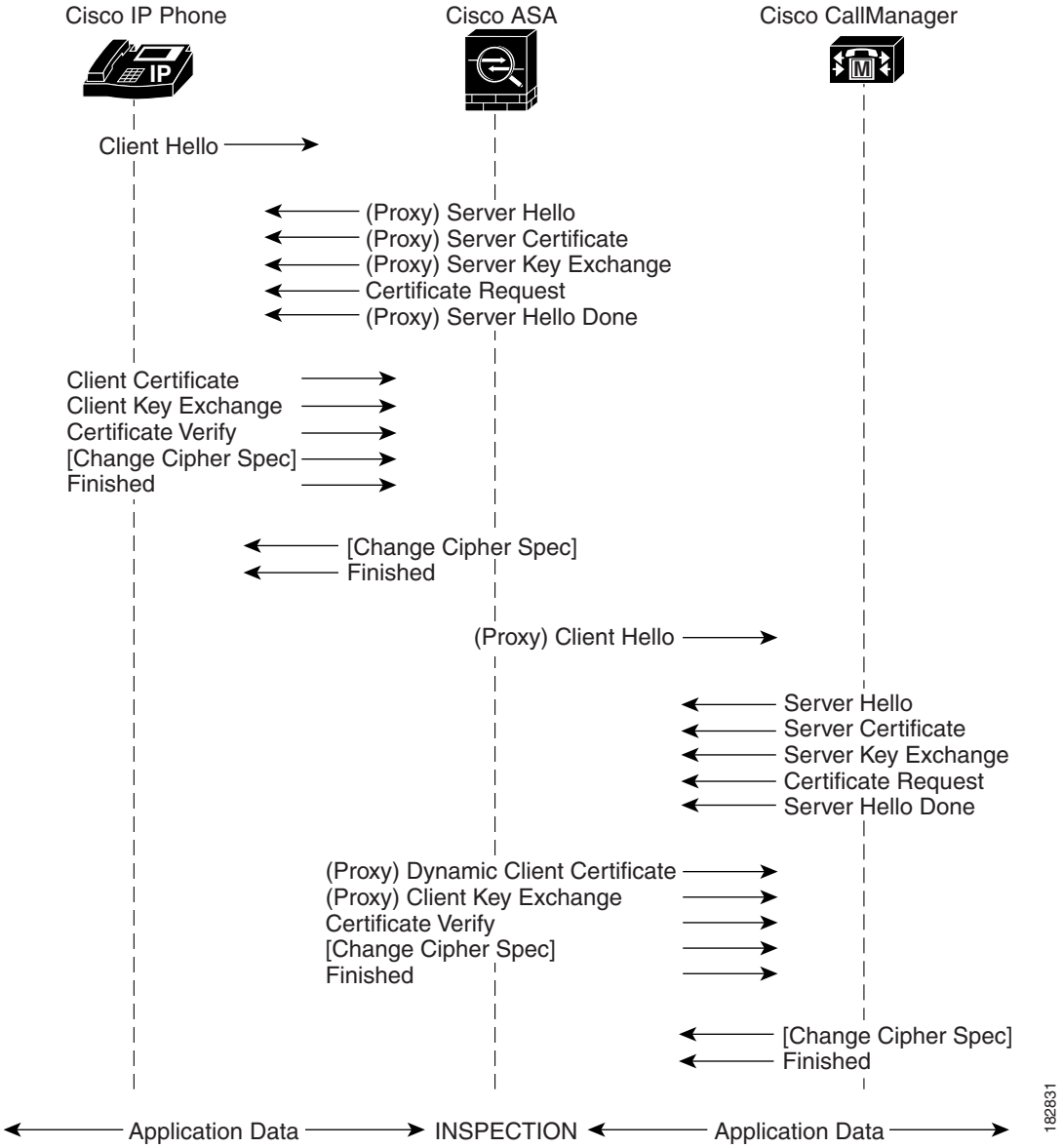
Table 27-11 Default and Maximum TLS Sessions on the Security Appliance

Security Appliance Platform	Default TLS Sessions	Maximum TLS Sessions
ASA 5505	10	80
ASA 5510	100	200
ASA 5520	300	1200
ASA 5540	1000	4500
ASA 5550	2000	4500

Configuring TLS Proxy

The security appliance in Figure 27-2 serves as a proxy for both client and server, with Cisco IP Phone and Cisco Unified CallManager interaction.

Figure 27-2 TLS Proxy Flow



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.

To configure the security appliance for TLS proxy, perform the following steps:

Step 1 (Optional) Set the maximum number of TLS proxy sessions to be supported by the security appliance using the following command, for example:

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

Note The **tls-proxy maximum-sessions** command controls the memory size reserved for cryptographic applications such as TLS proxy. Crypto memory is reserved at the time of system boot. You may need to reboot the security appliance for the configuration to take effect if the configured maximum sessions number is greater than the currently reserved.

Step 2 Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate. See Importing Certificates from the Cisco UCM, page 27-18.

- CallManager
- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002
- (Optional) CAPF—import the CAPF certificate when LSC provisioning is required or you have LSC enabled IP phones

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 3 Create necessary RSA key pairs using the following commands, for example:

```
hostname(config)# crypto key generate rsa label ccm_proxy_key modulus 1024
hostname(config)# crypto key generate rsa label ldc_signer_key modulus 1024
hostname(config)# crypto key generate rsa label phone_common modulus 1024
```

We recommend to use a different key pair for each role.

Step 4 Create the proxy certificate for the Cisco Unified CallManager cluster using the following commands, for example:

```
hostname(config)# ! for self-signed CCM proxy certificate
hostname(config)# crypto ca trustpoint ccm_proxy
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# fqdn none
hostname(config-ca-trustpoint)# subject-name cn=EJW-SV-1-Proxy
hostname(config-ca-trustpoint)# keypair ccm_proxy_key
hostname(config)# crypto ca enroll ccm_proxy
```

The Cisco Unified CallManager proxy certificate could be self-signed or issued by a third-party CA. The certificate is exported to the CTL client.

Note Cisco IP Phones require certain fields from the X.509v3 certificate to be present to validate the certificate via 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.

Each of the concatenated fields (when present) are separated by a semicolon, yielding one of the

following forms:

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

Step 5 Create an internal local CA to sign the LDC for Cisco IP Phones using the following commands, for example:

```
hostname(config)# ! for the internal local LDC issuer
hostname(config)# crypto ca trustpoint ldc-server
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# proxy-ldc-issuer
hostname(config-ca-trustpoint)# fqdn my-ldc-ca.exmaple.com
hostname(config-ca-trustpoint)# subject-name cn=FW-LDC-SIGNER-172_23_45_200
hostname(config-ca-trustpoint)# keypair ldc-signer-key
hostname(config)# crypto ca enroll ldc-server
```

This local CA is created as a regular self-signed trustpoint with **proxy-ldc-issuer** enabled. You may use the embedded local CA LOCAL-CA-SERVER on the security appliance to issue the LDC.

Step 6 Create a CTL Provider instance in preparation for a connection from the CTL Client using the following commands, for example:

```
hostname(config)# ctl-provider my_ctl
hostname(config-ctl-provider)# client interface inside address 172.23.45.1
hostname(config-ctl-provider)# client username CCMAdministrator password XXXXXX encrypted
hostname(config-ctl-provider)# export certificate ccm_proxy
hostname(config-ctl-provider)# ctl install
```

The username and password must match the username and password for Cisco Unified CallManager administration. The trustpoint name in the **export** command is the proxy certificate for the Cisco Unified CallManager server.

The default port number listened by the CTL Provider is TCP 2444, which is the default CTL port on the Cisco Unified CallManager. Use the **service port** command to change the port number if a different port is used by the Cisco Unified CallManager cluster.

Step 7 Create a TLS proxy instance using the following commands, for example:

```
hostname(config)# tls-proxy my_proxy
hostname(config-tlsp)# server trust-point ccm_proxy
hostname(config-tlsp)# client ldc issuer ldc_server
hostname(config-tlsp)# client ldc key-pair phone_common
hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1
```

The **server** commands configure the proxy parameters for the original TLS server. In other words, the parameters for the security appliance to act as the server during a TLS handshake, or facing the original TLS client. The **client** commands configure the proxy parameters for the original TLS client. In other words, the parameters for the security appliance to act as the client during a TLS handshake, or facing the original TLS server.

Step 8 Enable TLS proxy for the Cisco IP Phones and Cisco Unified CallManagers in Skinny or SIP inspection using the following commands, for example:

```
hostname(config)# class-map sec_skinny
hostname(config-cmap)# match port tcp eq 2443

hostname(config)# policy-map type inspect skinny skinny_inspect
hostname(config-pmap)# parameters
hostname(config-pmap-p)# ! Skinny inspection parameters
```

```
hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect skinny skinny_inspect
hostname(config-pmap)# class sec_skinny
hostname(config-pmap-c)# inspect skinny skinny_inspect tls-proxy my_proxy

hostname(config)# service-policy global_policy global
```

Step 9 Export the local CA certificate (ldc_server) and install it as a trusted certificate on the Cisco Unified CallManager 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:

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

b. For the embedded local CA server LOCAL-CA-SERVER, use the following command to export its certificate, for example:

```
hostname(config)# show crypto ca server certificate
```

Save the output to a file and import the certificate on the Cisco Unified CallManager. For more information, see the Cisco Unified CallManager document:
http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040848

After this step, you may use the Display Certificates function on the Cisco Unified CallManager GUI to verify the installed certificate:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040354

Step 10 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. See the Cisco Unified CallManager document for information on how to configure and use CTL Client:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_1/nci/p08/secuauth.htm

Note You will need the CTL Client that is released with Cisco Unified CallManager Release 5.1 to interoperate with the security appliance. See the "CTL Client" section on page 27-51 for more information regarding TLS proxy support.

Debugging TLS Proxy

You may 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 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
```

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

```
hostname(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
```

Use the **show tls-proxy** commands with different options to check the active TLS proxy sessions. The following are some sample outputs:

```
hostname(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

TLS-Proxy 'proxy': ref_cnt 1, seq# 1
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: ldc_signer
        Local dynamic certificate key-pair: phone_common
        Cipher-suite: <unconfigured>
    Run-time proxies:
        Proxy 0xcbadf720: Class-map: skinny_ssl, Inspect: skinny
            Active sess 1, most sess 1, byte 42916

hostname(config-tlsp)# show tls-proxy session count
2 in use, 4 most used

hostname(config-tlsp)# show tls-proxy session
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786

hostname(config-tlsp)# show tls-proxy session detail
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e498 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e478 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 29
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=TLS-Proxy-Signer
    Subject Name:
        cn=SEP0002B9EB0AAD
        o=Cisco Systems Inc
        c=US
    Validity Date:
        start date: 09:25:41 PDT Apr 16 2007
        end   date: 09:25:41 PDT Apr 15 2008
    Associated Trustpoints:

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:
```

CTL Client

The CTL Client application supplied by Cisco Unified CallManager Release 5.1 and later supports a TLS proxy server (firewall) in the CTL file. Figure 27-3 through Figure 27-6 illustrate the TLS proxy features supported in the CTL Client.

Figure 27-3 CTL Client TLS Proxy Features — Add Firewall

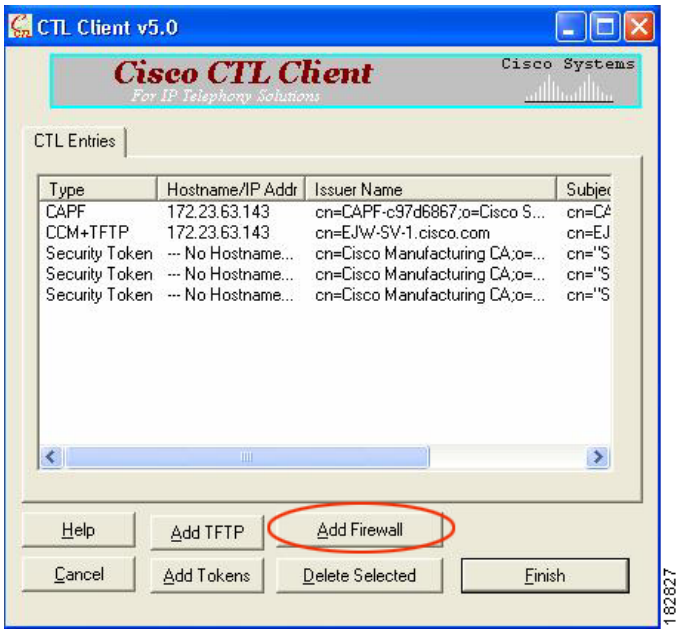


Figure 27-3 shows support for adding a CTL entry consisting of the security appliance as the TLS proxy.

Figure 27-4 CTL Client TLS Proxy Features — ASA IP Address or Domain Name

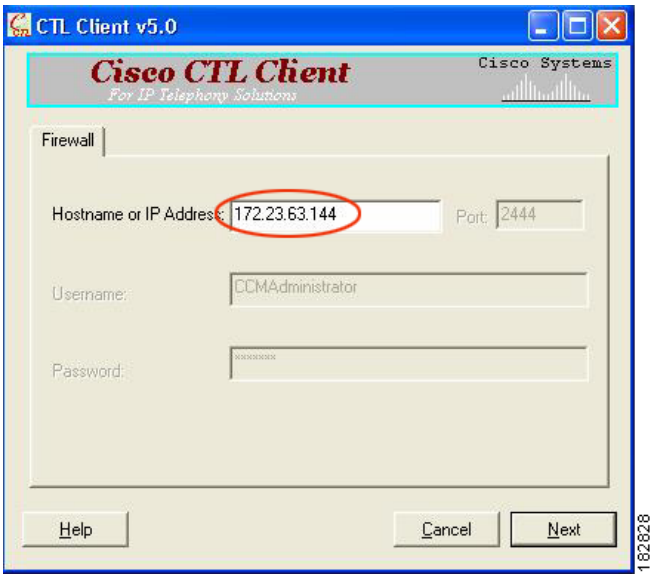


Figure 27-4 shows support for entering the security appliance IP address or domain name in the CTL Client.

Figure 27-5 CTL Client TLS Proxy Features — CTL Entry for ASA

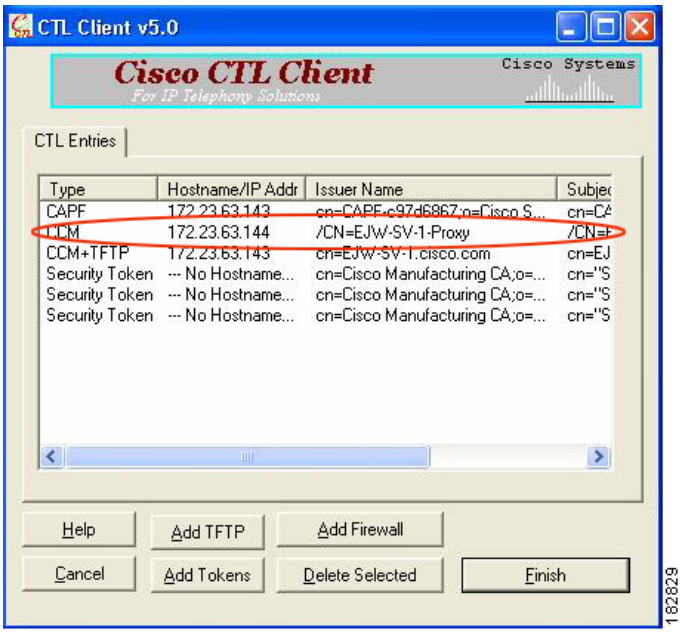
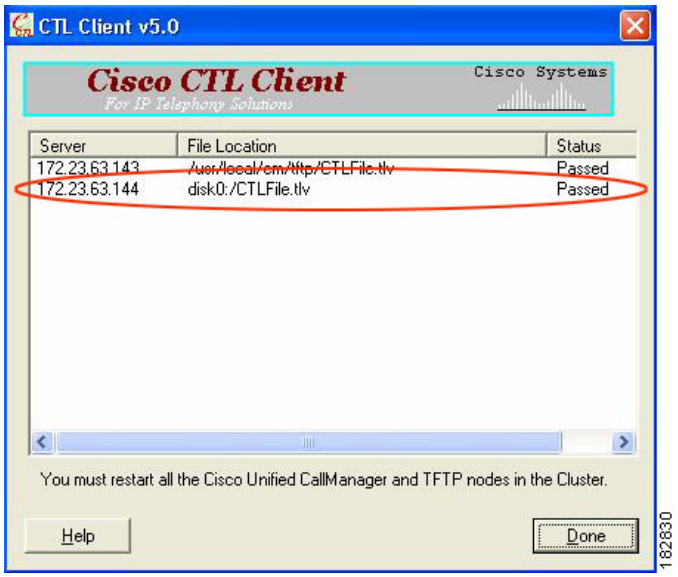


Figure 27-5 shows that the CTL entry for the security appliance as the TLS proxy has been added. The CTL entry is added after the CTL Client connects to the CTL Provider service on the security appliance and retrieves the proxy certificate.

Figure 27-6 CTL Client TLS Proxy Features — CTL File Installed on the ASA



The security appliance does not store the raw CTL file in the flash, rather, it parses the CTL file and installs appropriate trustpoints. Figure 27-6 indicates the installation was successful.

Cisco Unified Mobility and MMP Inspection Engine

This section includes the following topics:

Mobility Proxy Overview

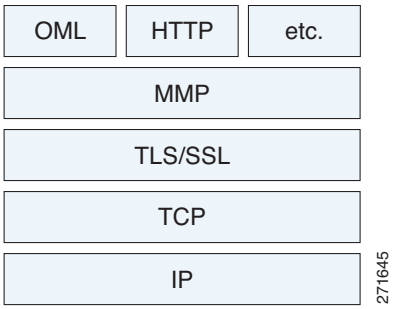
To support Cisco UMA for the Cisco Unified Mobility solution, the mobility proxy (implemented as a TLS proxy) includes the following functionality:

- 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 security appliance 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 Figure 27-7, 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 27-7 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 security appliance 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.

Note 4096 is the value currently used in MMP implementations.

Because MMP headers and entities can be split across packets, the security appliance buffers data to ensure consistent inspection. The SAPI (stream API) handles data buffering for pending inspection opportunities. MMP header text is treated as case insensitive and a space is present between header text and values. Reclaiming of MMP state is performed by monitoring the state of the TCP connection.

Mobility Proxy Deployment Scenarios

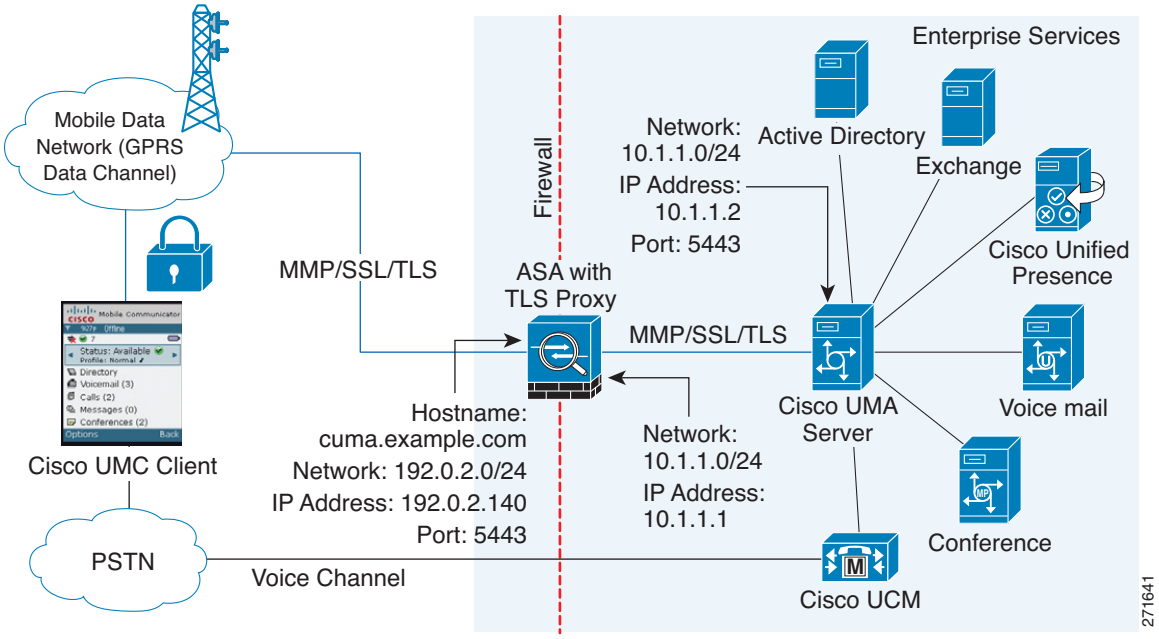
Figure 27-8 and Figure 27-9 show the two deployment scenarios for the TLS proxy used by the Cisco Unified Mobility solution. In scenario 1 (the recommended deployment architecture), the security appliance functions as both the firewall and TLS proxy. In scenario 2, the security appliance functions as 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 security appliance 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 security appliance intercepts the connections and inspects the data that the client sends to the Cisco UMA server.

Note The TLS proxy for the Cisco Unified Mobility 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 27-8 Security Appliance as Firewall with Mobility Proxy and MMP Inspection



In Figure 27-8, the security appliance performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

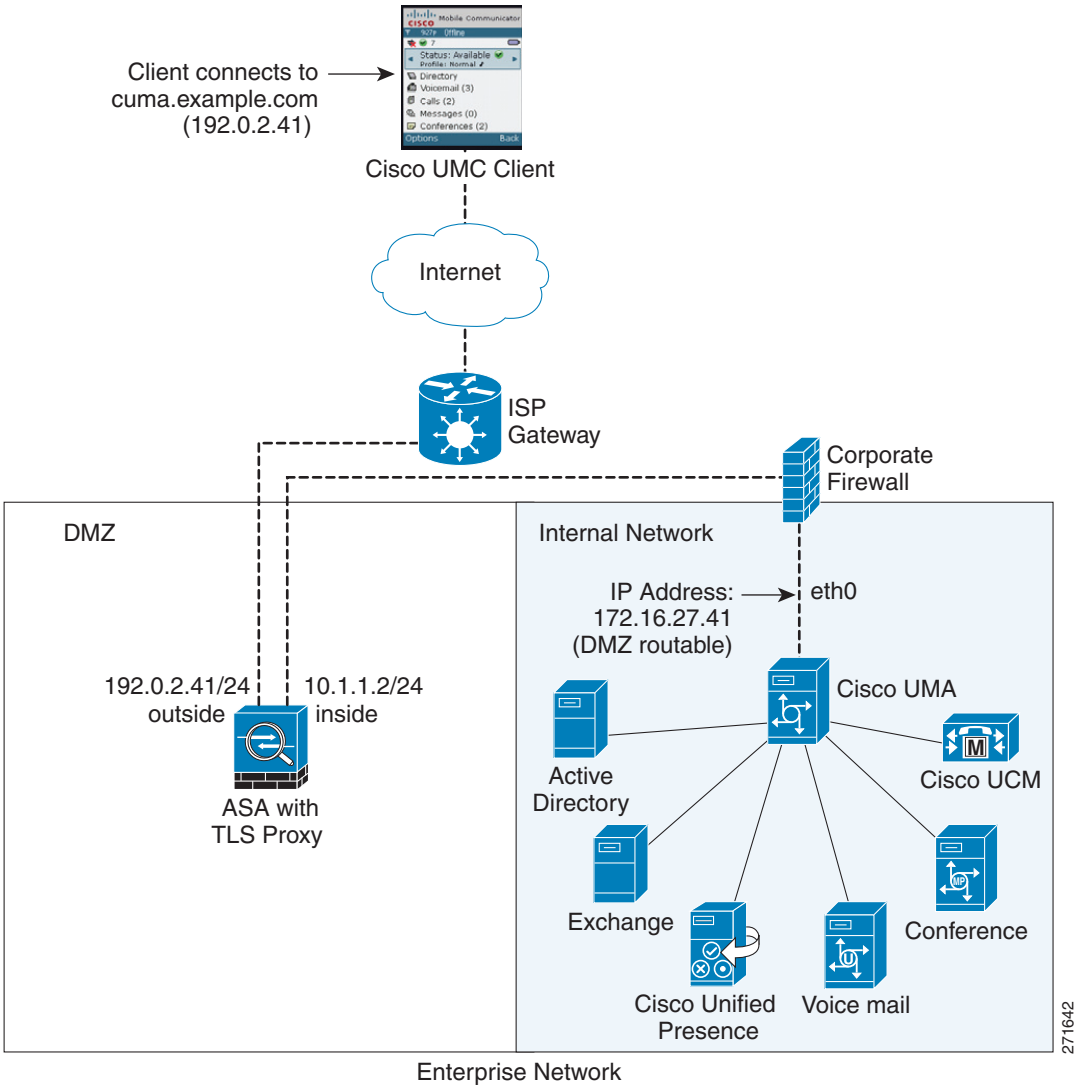
Figure 27-9 shows deployment scenario 2, where the security appliance functions as the TLS proxy only and does not function as the corporate firewall. In this scenario, the security appliance and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client 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)# nat (outside) 1 0.0.0.0 0.0.0.0 outside
hostname(config)# global (inside) 1 192.0.2.183 netmask 255.255.255.255
```

Note This interface PAT rule converges the Cisco UMA client IP addresses on the outside interface of the security appliance 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 Unified Mobility is enabled on the same interface of the security appliance 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 27-9 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as Mobility Proxy Only



Mobility Proxy Using NAT/PAT

In both scenarios (Figure 27-8 and Figure 27-9), NAT can be used to hide the private address of the Cisco UMA servers.

In scenario 2 (Figure 27-9), 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
```

Establishing Trust Relationships for Cisco UMA Deployments

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

Figure 27-10 shows how you can import the Cisco UMA server certificate onto the security appliance. When the Cisco UMA server has already enrolled with a third-party CA, you can import the certificate with the private key onto the security appliance. Then, the security appliance has the full credentials of the Cisco UMA server. When a Cisco UMA client connects to the Cisco UMA server, the security appliance intercepts the handshake and uses the Cisco UMA server certificate to perform the handshake with the client. The security appliance also performs a handshake with the server.

Figure 27-10 How the Security Appliance Represents Cisco UMA – Private Key Sharing

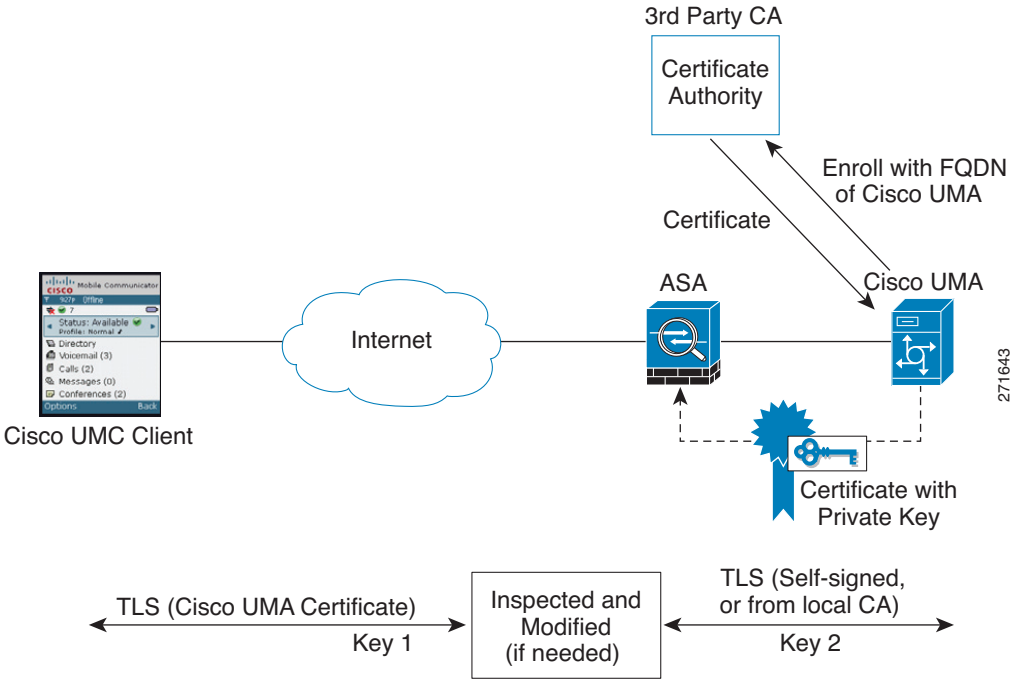
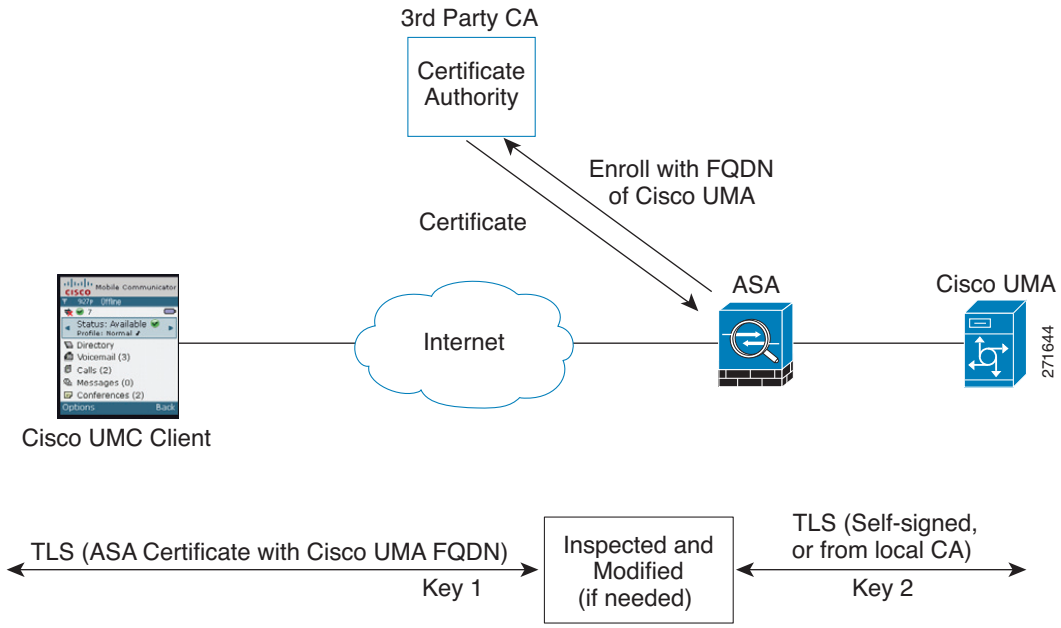


Figure 27-11 shows another way to establish the trust relationship. Figure 27-11 shows a green field deployment, because each component of the deployment has been newly installed. The security appliance enrolls with the third-party CA by using the Cisco UMA server FQDN as if the security appliance is the Cisco UMA server. When the Cisco UMA client connects to the security appliance, the security appliance 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 27-11 How the Security Appliance Represents Cisco UMA – Certificate Impersonation



A trusted relationship between the security appliance and the Cisco UMA server can be established with self-signed certificates. The security appliance'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 security appliance truststore by creating a trustpoint and using the **crypto ca authenticate** command.

Configuring the Security Appliance for Cisco Unified Mobility

To configure for the security appliance to perform TLS proxy and MMP inspection as shown in Figure 27-8 and Figure 27-9, perform the following steps. It is assumed that self-signed certificates are used between the security appliance and the Cisco UMA server.

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

```
hostname(config)# static (real_ifc,mapped_ifc) mapped_ip real_ip netmask mask
```

Step 2 Export the Cisco UMA server certificate and keypair in PKCS-12 format. Import it onto the security appliance. The certificate will be used during the handshake with the Cisco UMA clients.

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

Step 3 Install the Cisco UMA server self-signed certificate in the security appliance truststore. This step is necessary for the security appliance to authenticate the Cisco UMA server during the handshake between the security appliance proxy and Cisco UMA server.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
```

```
  [ certificate data omitted ]
hostname(config)# quit
```

Step 4 Create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server:

```
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_name
hostname(config-tlsp)# no server authenticate-client
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Step 5 Enable the TLS proxy for MMP inspection:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq port
hostname(config-cmap)# exit
hostname(config)# policy-map name
hostname(config-pmap)# class name
hostname(config-pmap)# inspect mmp tls-proxy proxy_name
hostname(config-pmap)# exit
hostname(config)# service-policy policy_map_name global
```

Debugging for Cisco Unified Mobility

Mobility 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 TLS Proxy for Encrypted Voice Inspection, page 27-43.

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
```

Cisco Unified Presence

This section includes the following topics:

Architecture for Cisco Unified Presence

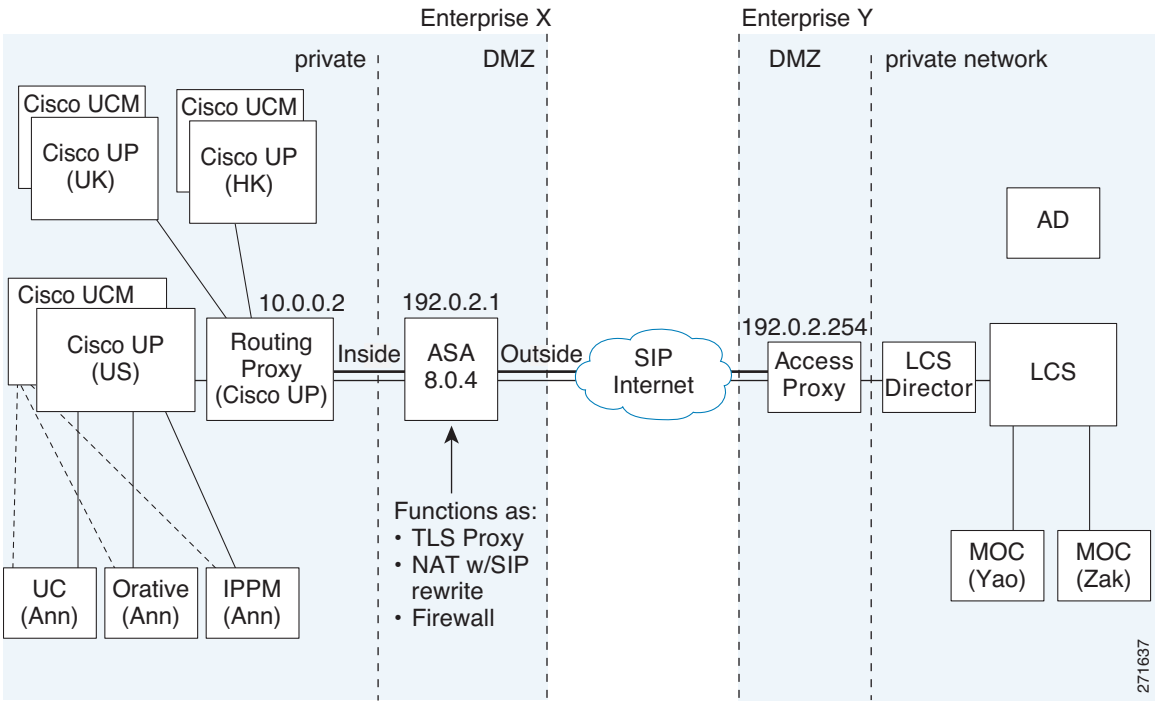
Figure 27-12 depicts a Cisco Unified Presence/LCS Federation scenario with the security appliance as the presence federation proxy (implemented as a TLS proxy). The two entities with a TLS connection are the “Routing Proxy” (a dedicated Cisco UP) in Enterprise X and the Microsoft Access Proxy in Enterprise 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 security appliance; 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 security appliance 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 27-12 Typical Cisco Unified Presence/LCS Federation Scenario



In the above architecture, the security appliance functions as a firewall, NAT, and TLS proxy, which is the recommended architecture. However, the security appliance 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 security appliance as the TLS proxy.

In Figure 27-12, 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:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 5061 10.0.0.2 5061 netmask
255.255.255.255
```

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:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 5062 10.0.0.2 5062 netmask
255.255.255.255
hostname(config)# static (inside,outside) udp 192.0.2.1 5070 10.0.0.2 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 5060 10.0.0.2 5060 netmask
255.255.255.255
```

For another Cisco UP with the address 10.0.0.3, you must use a different set of PAT ports, such as 45062 or 45070:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 45061 10.0.0.3 5061 netmask
255.255.255.255
```

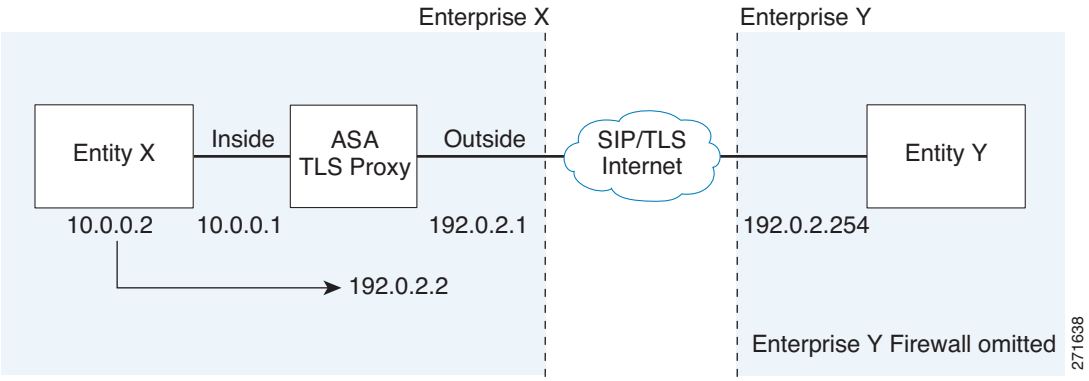
```
hostname(config)# static (inside,outside) tcp 192.0.2.1 45062 10.0.0.3 5062 netmask
255.255.255.255
hostname(config)# static (inside,outside) udp 192.0.2.1 45070 10.0.0.3 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 5070 10.0.0.2 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 45060 10.0.0.3 5060 netmask
255.255.255.255
```

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

```
hostname(config)# global (outside) 102 192.0.2.1 netmask 255.255.255.255
hostname(config)# nat (inside) 102 0.0.0.0 0.0.0.0
```

Figure 27-13 illustrates an abstracted scenario with Entity X connected to Entity Y through the presence federation proxy on the security appliance. The proxy is in the same administrative domain as Entity X. Entity Y could have another security appliance as the proxy but this is omitted for simplicity.

Figure 27-13 Abstracted Presence Federation Proxy Scenario between Two Server Entities



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

Establishing a 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.

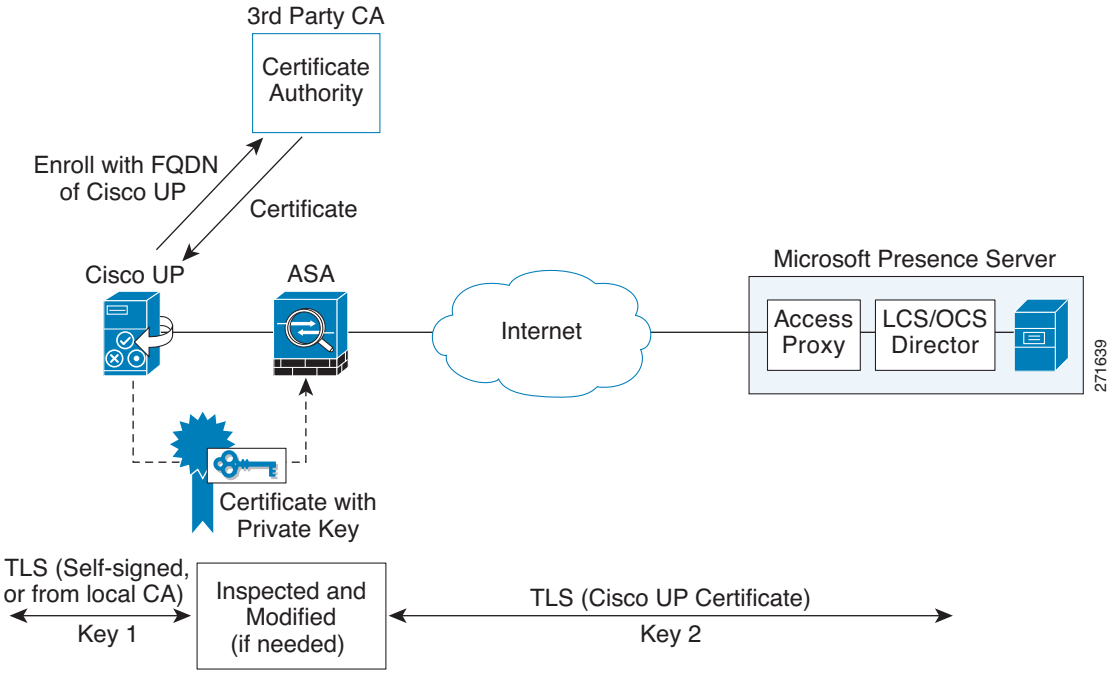
Establishing a trust relationship cross enterprises or across administrative domains is key for federation. Cross enterprises you must use a trusted third-party CA (such as, VeriSign). The security appliance obtains a certificate with the FQDN of the Cisco UP (certificate impersonation).

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 security appliance as the TLS proxy must be trusted by both entities. The security appliance is always associated with one of the enterprises. Within that enterprise (Enterprise X in Figure 27-12), the entity and the security appliance could authenticate each other via a local CA, or by using self-signed certificates.

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

Figure 27-14 shows the way to establish the trust relationship. The security appliance enrolls with the third party CA by using the Cisco UP FQDN as if the security appliance is the Cisco UP.

Figure 27-14 How the Security Appliance Represents Cisco Unified Presence – Certificate Impersonate



About the 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 security appliance, and configure a trustpoint to identify the self-signed certificate sent by the security appliance to Cisco UP (such as `cup_proxy`) in the TLS handshake.

For the security appliance 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.

Configuring the Presence Federation Proxy for Cisco Unified Presence

To configure a Cisco Unified Presence/LCS Federation scenario with the security appliance 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 security appliance (like the scenario shown in Figure 27-12), perform the following steps.

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)# static (real_ifc,mapped_ifc) tcp mapped_ip mapped_port netmask mask
```

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 security appliance SIP inspection engine takes care of the necessary translation (fixup).

```
hostname(config)# global (mapped_ifc) nat_id mapped_ip netmask mask
hostname(config)# nat (real_ifc) nat_id real_ip mask
```

Step 2 Create the necessary RSA keypairs by entering the following command:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

The keypair is used by the self-signed certificate presented to the local domain containing the Cisco UP (proxy for the remote entity).

Step 3 Create a proxy certificate, which is a self-signed certificate, for the remote entity by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# fqdn none
hostname(config-ca-trustpoint)# subject-name X.500_name
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

You will install the certificate on the local entity truststore. You could also enroll the certificate with a local CA trusted by the local entity.

Step 4 Export the self-signed certificate for the security appliance created in Step 3 and install it as a trusted certificate on the local entity. This step is necessary for local entity to authenticate the security appliance.

Export the security appliance self-signed (identity) certificate by entering the following command:

```
hostname(config)# crypto ca export trustpoint identity-certificate
```

Step 5 Export the local entity certificate and install it on the security appliance by entering the following commands. This step is needed for the security appliance to authenticate the local entity during the handshake. 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. The following configuration shows the commands for using a self-signed certificate.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
  [ certificate data omitted ]
hostname(config)# quit
```

Step 6 To create a proxy certificate on the security appliance 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 Certificate Configuration, page 41-5.

Step 7 Install the CA certificate that signs the remote entity certificate on the security appliance by entering the following commands. This step is necessary for the security appliance to authenticate the remote entity.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
```

```
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
  [ certificate data omitted ]
hostname(config)# quit
```

Step 8 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.

```
! Local entity to remote entity
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_trustpoint
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Where the *proxy_name* for the **server trust-point** command is the remote entity proxy name and the *proxy_trustpoint* for the **client trust-point** command is the local entity proxy.

```
! Remote entity to local entity
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_trustpoint
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Where the *proxy_name* for the **server trust-point** command is the local entity proxy name and the *proxy_trustpoint* for the **client trust-point** command is the remote entity proxy.

Step 9 Enable the TLS proxy for SIP inspection and define policies for both entities that could initiate the connection by entering the following commands:

```
hostname(config)# access-list id extended permit tcp host src_ip host dest_ip eq port
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list access_list_name
hostname(config-cmap)# exit
hostname(config)# policy-map type inspect sip policy_map_name
hostname(config-pmap)# parameters
    ! SIP inspection parameters
hostname(config-pmap)# exit
hostname(config)# policy-map name
hostname(config-pmap)# class name
hostname(config-pmap)# inspect sip sip_map tls-proxy proxy_name
hostname(config-pmap)# exit
hostname(config)# service-policy policy_map_name global
```

Where *name* for the **policy-map** command is the name of the global policy map.

Debugging the Security Appliance for 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 TLS Proxy for Encrypted Voice Inspection, page 27-43.

Enable the **debug sip** command for SIP inspection engine debugging. See the *Cisco Security Appliance 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
```

Sample Configurations for Cisco Unified Communications Proxy Features

This section includes the following topics:

- Phone Proxy Sample Configurations, page 27-66
- Cisco Unified Mobility Sample Configurations, page 27-76
- Cisco Unified Presence Sample Configuration, page 27-79

Phone Proxy Sample Configurations

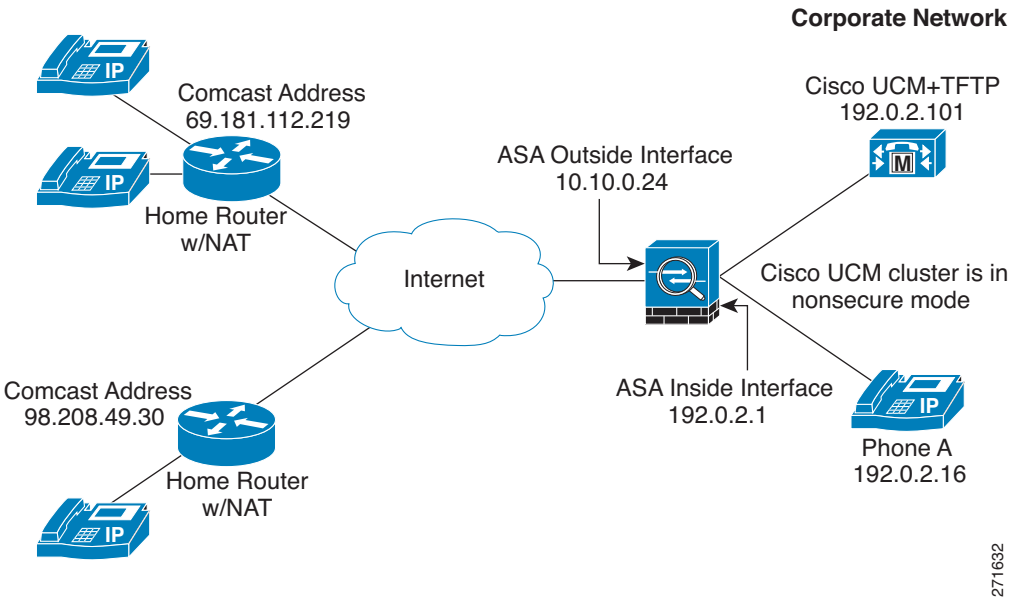
This section includes the following topics:

- Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 27-66
- Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 27-67
- Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-69
- Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers, page 27-70
- Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-72
- Example 6: VLAN Transversal, page 27-74

Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 27-15 shows an example of the configuration for a non-secure Cisco UCM cluster using the following topology.

Figure 27-15 Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

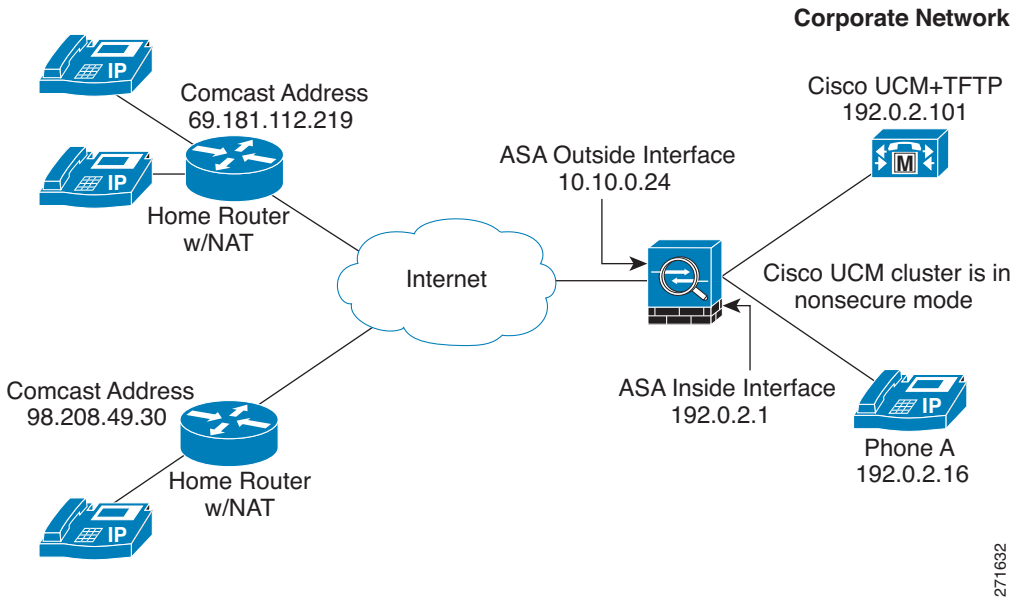



```
static (inside,outside) 10.10.0.26 192.0.2.101
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
phone-proxy mypp
    media-termination address 192.0.2.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 27-16 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology.

Figure 27-16 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher




```
static (inside,outside) 10.10.0.26 192.0.2.101
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
```

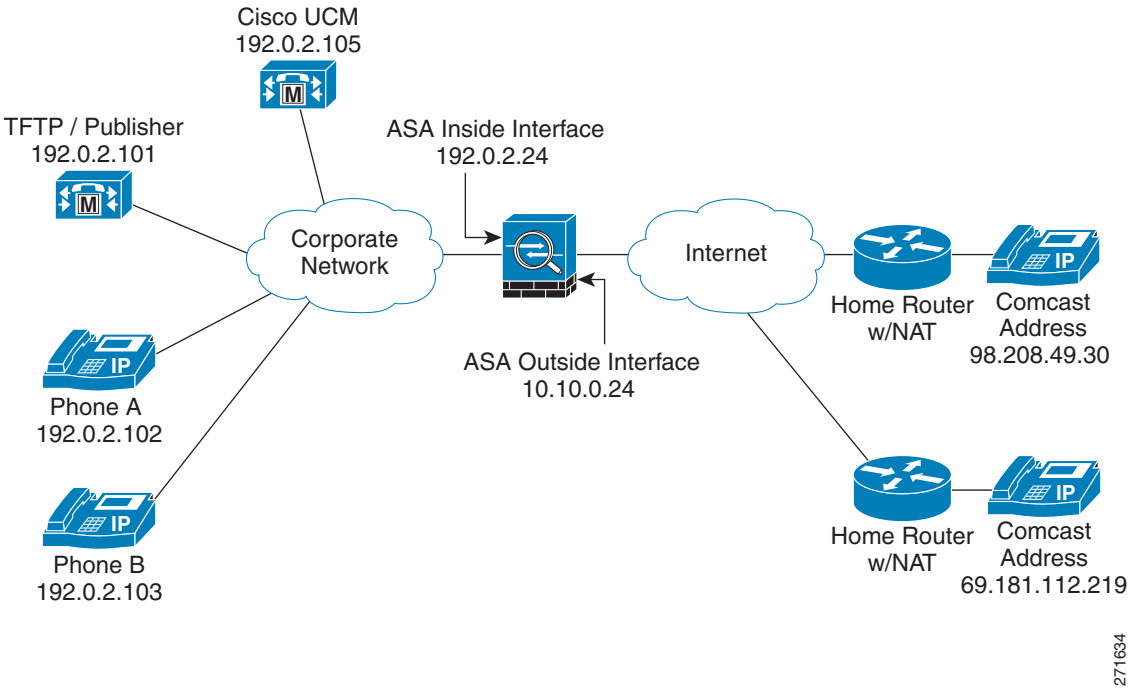
```
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers

Figure 27-17 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the Cisco UCM.

In this sample, the static interface PAT for the TFTP server is configured to appear like the security appliance's outside interface IP address.

Figure 27-17 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers




```
static (inside,outside) 10.10.0.26 192.0.2.105
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cucm_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cucm_kp
crypto ca enroll cucm
crypto key generate rsa label tftp_kp modulus 1024
crypto ca trustpoint tftp_server
    enrollment self
    keypair tftp_kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
```

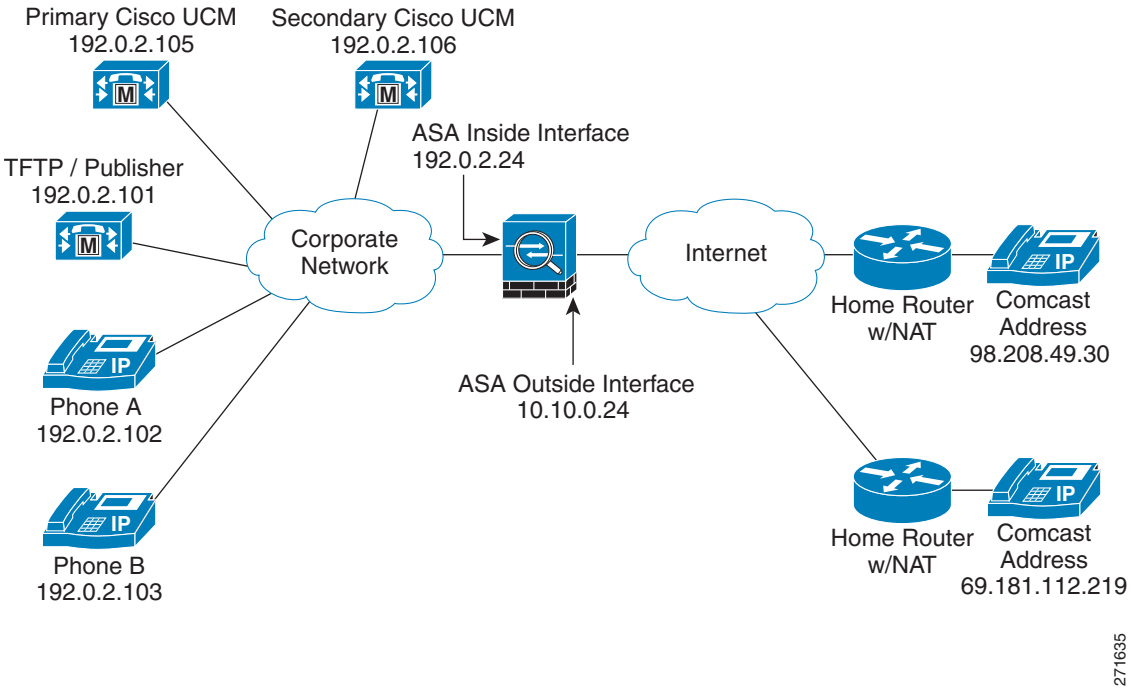
```
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers

Figure 27-18 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the primary and secondary Cisco UCMs.

In this sample, the static interface PAT for the TFTP server is configured to appear like the security appliance's outside interface IP address.

Figure 27-18 Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary Cisco UCM, and TFTP Server on Different Servers




```
static (inside,outside) 10.10.0.27 192.0.2.105
static (inside,outside) 10.10.0.26 192.0.2.106
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint pri_cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll pri_cucm
crypto ca trustpoint sec_cucm
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll sec_cucm
crypto ca trustpoint tftp_server
    enrollment self
    fqdn my_tftp.example.com
    keypair cluster_kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry tftp trustpoint tftp_server address 10.10.0.24
    record-entry cucm trustpoint pri_cucm_server address 10.10.0.27
    record-entry cucm trustpoint sec_cucm_server address 10.10.0.2
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
```

```
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher

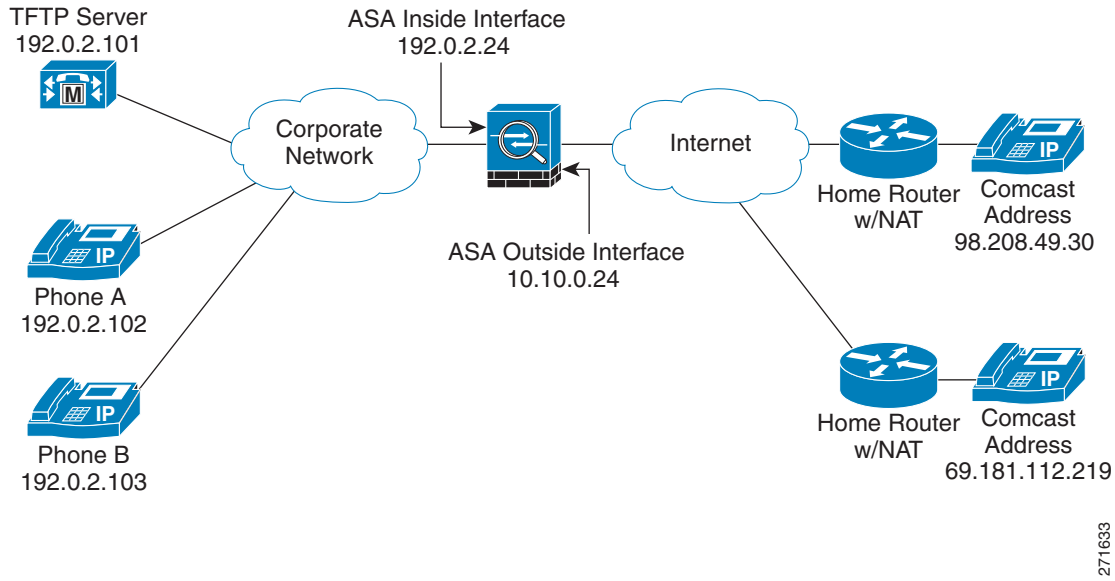
Figure 27-19 shows an example of the configuration for a mixed-mode Cisco UCM cluster where LSC provisioning is required using the following topology.

Note Doing LSC provisioning for remote IP phones is not recommended because it requires that the IP phones first register and they have to register in nonsecure mode. Having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance. If possible, LSC provisioning should be done inside the corporate network before giving the IP phones to the end-users.

In this sample, you create an access list to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Figure 27-19 LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher




```
static (inside,outside) 10.10.0.26 192.0.2.105
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-list pp extended permit tcp any host 10.10.0.26 eq 2000
access-list pp extended permit tcp any host 10.10.0.26 eq 5060
access-list pp extended permit tcp any host 10.10.0.26 eq 3804
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll cucm
crypto ca trustpoint tftp_server
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll tftp_server
crypto ca trustpoint capf
    enroll terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    record-entry capf trustpoint capf address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
```

```
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 6: VLAN Transversal

Figure 27-20 shows an example of the configuration to force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario. VLAN transversal is required between CIPC softphones on the data VLAN and hard phones on the voice VLAN.

In this sample, the Cisco UCM cluster mode is nonsecure.

In this sample, you create an access list to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

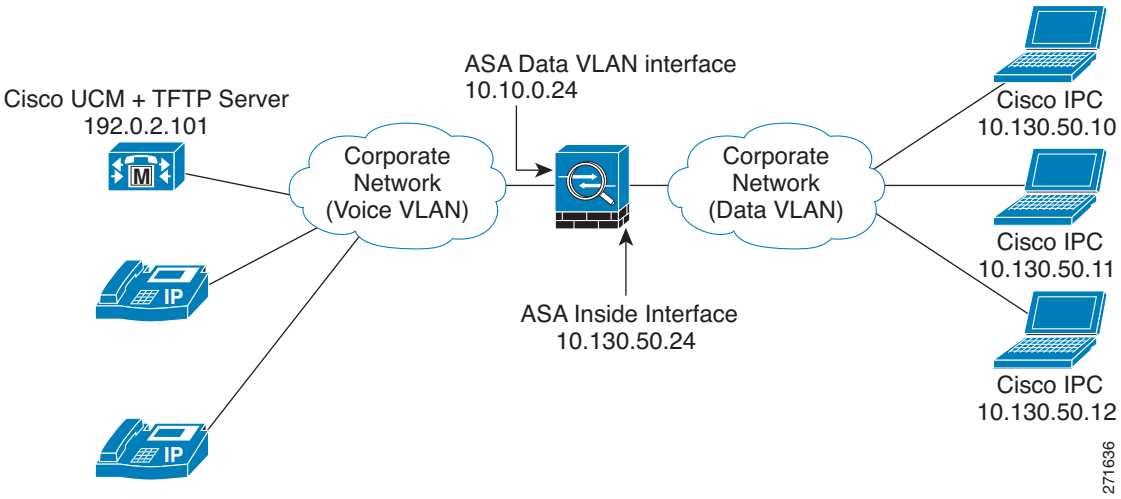
In this sample, you configure NAT for the CIPC by using PAT so that each CIPC is mapped to an IP address space in the Voice VLAN.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Note Cisco IP Communicator supports authenticated mode only and does not support encrypted mode; therefore, there is no encrypted voice traffic (SRTP) flowing from the CIPC softphones.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

Figure 27-20 VLAN Transversal Between CIPC Softphones on the Data VLAN and Hard Phones on the Voice VLAN




```
static (voice,data) 10.130.50.5 192.0.2.101
nat (data) 101 10.130.50.0 255.255.255.0 outside
global (voice) 101 192.0.2.10
access-list pp extended permit udp any host 10.130.50.5 eq 69
access-list pp extended permit tcp any host 10.130.50.5 eq 2000
access-list pp extended permit tcp any host 10.130.50.5 eq 5060
access-list pp extended permit tcp any host 10.130.50.5 eq 3804
access-group pp in interface data
crypto ca generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
crypto ca trustpoint capf
    enrollment terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.130.50.5
    record-entry capf trustpoint capf address 10.130.50.5
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
phone-proxy mypp
    media-termination address 10.130.50.2
    tftp-server address 10.10.0.20 interface inside
    tls-proxy mytls
    ctl-file myctl
    cipc security-mode authenticated
class-map sec_sccp
    match port tcp eq 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface data
```

Cisco Unified Mobility Sample Configurations

This section includes the following topics:

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

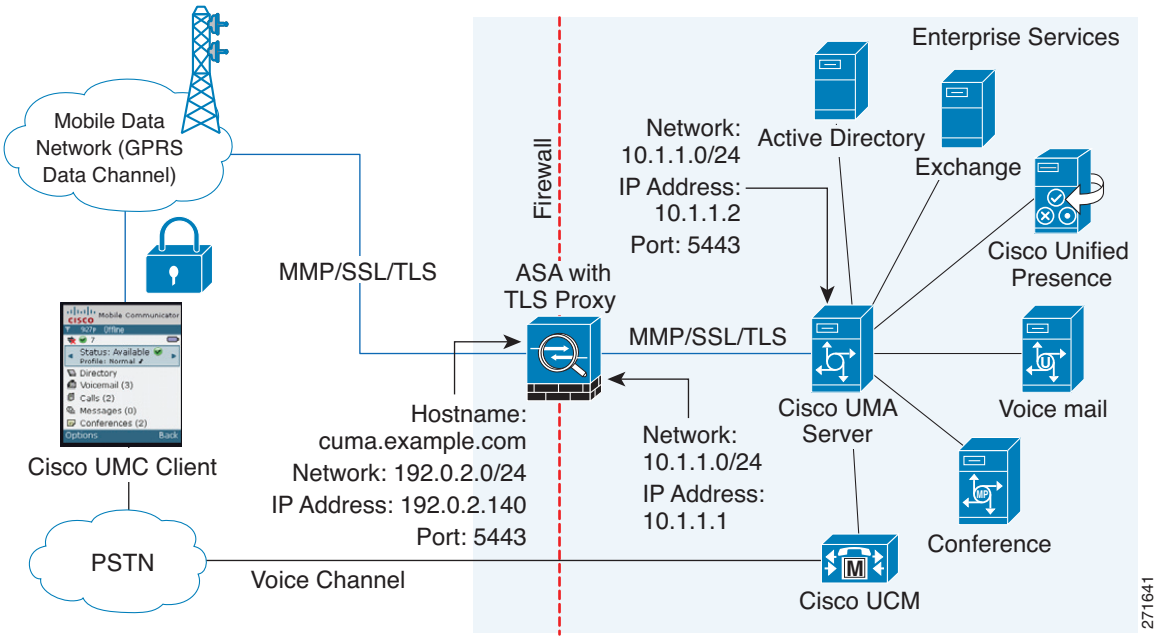
In the samples, you export the Cisco UMA server certificate and key-pair in PKCS-12 format and import it to the security appliance. The certificate will be used during handshake with the Cisco UMA clients.

Installing the Cisco UMA server self-signed certificate in the security appliance truststore is necessary for the security appliance to authenticate the Cisco UMA server during handshake between the security appliance 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 Figure 27-21 (scenario 1—the recommended architecture), the security appliance functions as both the firewall and TLS proxy. In the scenario 1 deployment, the security appliance is between a Cisco UMA client and a Cisco UMA server. In this scenario, the security appliance performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

Figure 27-21 Cisco UMC/Cisco UMA Architecture – Scenario 1: Security Appliance as Firewall with TLS Proxy and MMP Inspection



```
static (inside,outside) 192.0.2.140 10.1.1.2 netmask 255.255.255.255
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
```

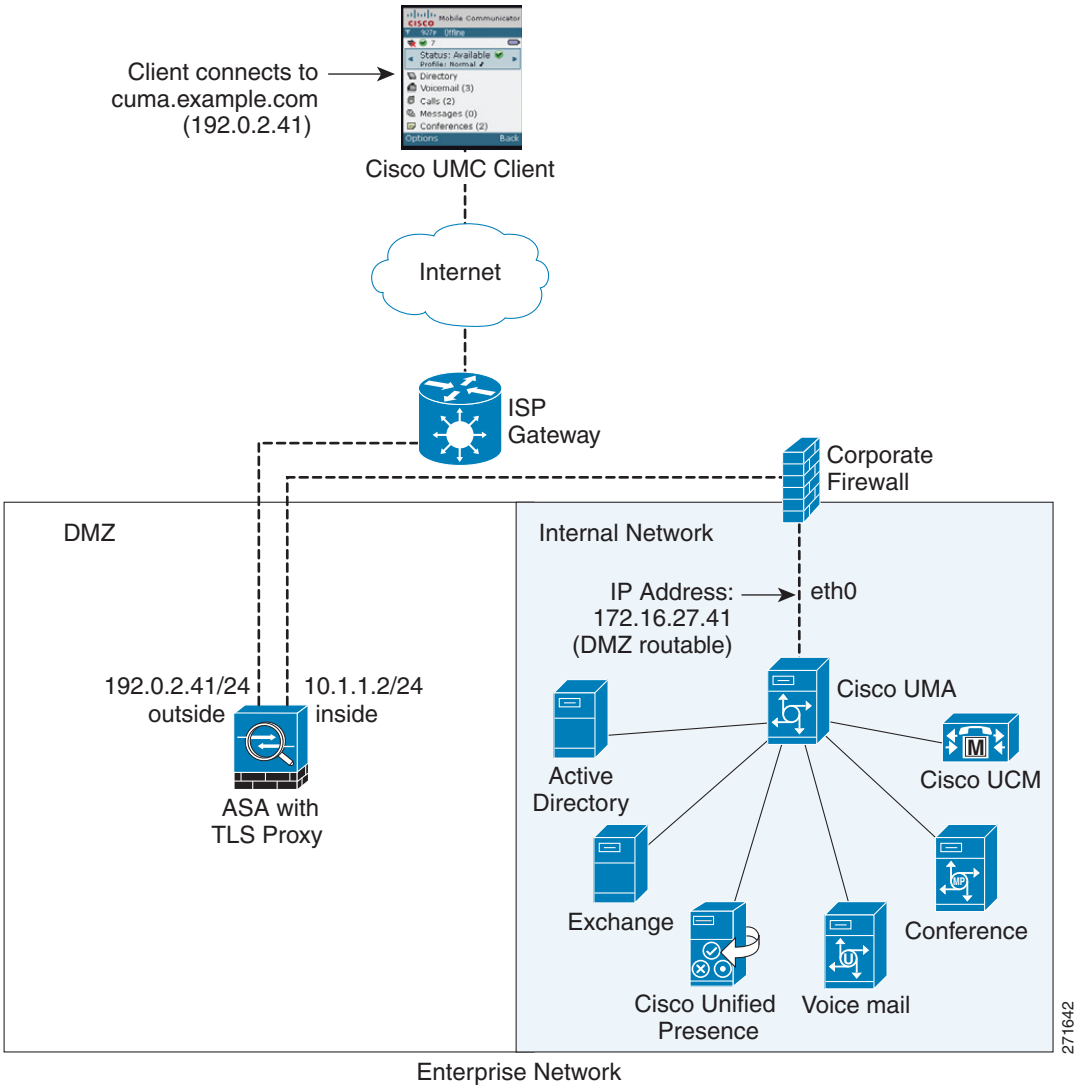
Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only

As shown in Figure 27-22 (scenario 2), the security appliance functions as the TLS proxy only and works with an existing firewall. The security appliance 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 67.11.12.183.

```
hostname(config)# nat (outside) 1 0.0.0.0 0.0.0.0 outside
hostname(config)# global (inside) 1 10.1.1.2 netmask 255.255.255.255
```

Figure 27-22 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as TLS Proxy Only




```
static (inside,outside) 192.0.2.41 172.16.27.41 netmask 255.255.255.255
nat (outside) 1 0.0.0.0 0.0.0.0 outside
global (inside) 1 10.1.1.2 netmask 255.255.255.255
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
```

Cisco Unified Presence Sample Configuration

The following sample illustrates the necessary configuration for the security appliance to perform TLS proxy for Cisco Unified Presence as shown in Figure 27-23. 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 45062 or 45070). Dynamic NAT or PAT can be used for outbound connections or TLS handshake. The security appliance 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 security appliance self-signed certificate (ent_y_proxy) and installing it as a trusted certificate on Entity X is necessary for Entity X to authenticate the security appliance. Exporting the Entity X certificate and installing it on the security appliance is needed for the security appliance 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.

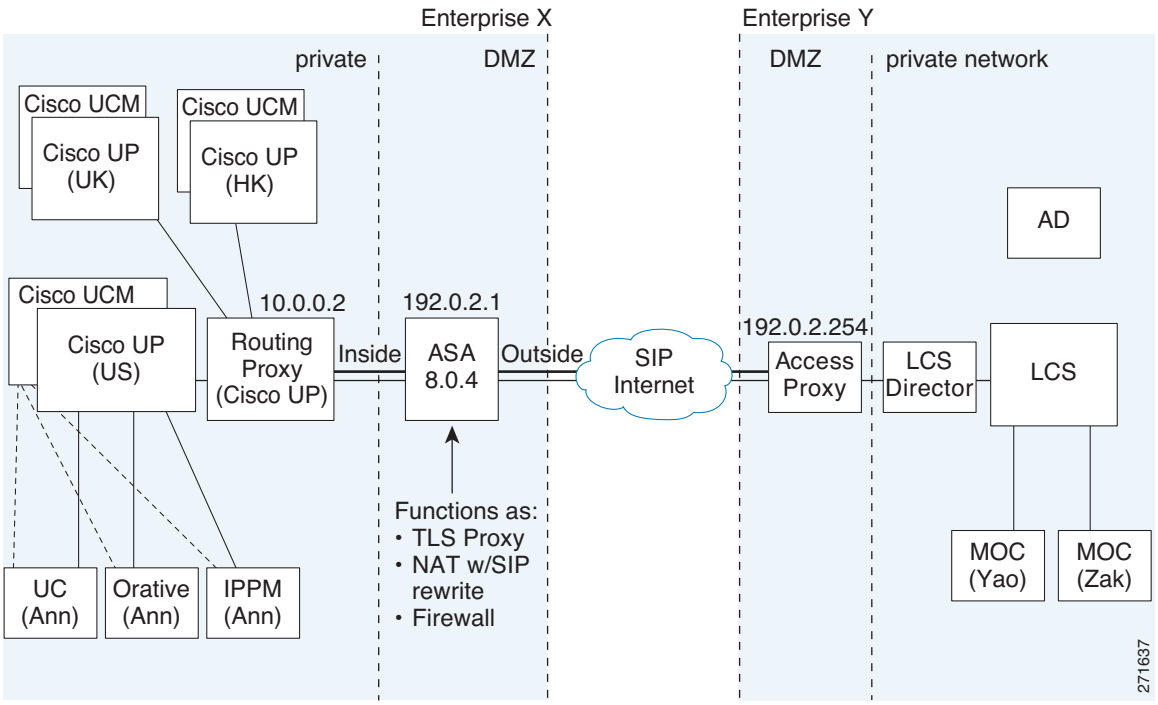
For about obtaining a certificate from a trusted CA, see Certificate Configuration, page 41-5.

Installing the CA certificate that signs the Entity Y certificate on the security appliance is necessary for the security appliance 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 27-23 Typical Cisco Unified Presence/LCS Federation Scenario




```
static (inside,outside) tcp 192.0.2.1 5061 10.0.0.2 5061 netmask 255.255.255.255
static (inside,outside) tcp 192.0.2.1 5062 10.0.0.2 5062 netmask 255.255.255.255
static (inside,outside) udp 192.0.2.1 5070 10.0.0.2 5070 netmask 255.255.255.255
static (inside,outside) tcp 192.0.2.1 45062 10.0.0.3 5062 netmask 255.255.255.255
static (inside,outside) udp 192.0.2.1 45070 10.0.0.3 5070 netmask 255.255.255.255
global (outside) 102 192.0.2.1 netmask 255.255.255.255
nat (inside) 102 0.0.0.0 0.0.0.0
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_proxy
    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_proxy
    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
```