Configuring the Cisco Unified SIP Proxy

Configuring Logical Networks

Each interface on the Cisco Unified SIP Proxy is associated with a logical network. Logical networks are used to organize server groups, listen points, and other properties. SIP messages are associated with the network on which they arrive.

Summary Steps

1. **cusp**
2. **configure**
3. **sip network** *network*

4. **end network**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip network** *network* **Example:** `se-10-1-0-0(cusp-config)> sip network service-provider`	Creates a network and puts you into network command mode. In this case, the network that is being created is called “service provider”.
Step 4	**end network** **Example:** `se-10-1-0-0(cusp-config-network)> end network`	Exits network command mode.

Example

The following example creates a network called “service-provider”:

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> sip network service-provider
se-10-1-0-0(cusp-config-network)> end network
```

Configuring Trigger Conditions

You create trigger conditions to allow Cisco Unified SIP Proxy to respond with the appropriate action for various call flows. In general, the more complex the call flow is, the more complex the trigger must be.

Summary Steps

1. **cusp**

2. **configure**
3. **trigger condition** *trigger-condition-name*
4. **sequence** *sequence-number*
5. (Optional) **in-network** *network-name*
6. (Optional) **mid-dialog**
7. end sequence
8. end trigger condition

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger condition** *trigger-condition-name* **Example:** `se-10-1-0-0(cusp-config)> trigger condition call-from-service-provider`	Creates a trigger condition and puts you into trigger command mode. In this case, the trigger that is being created is called “call-from-service-provider”.
Step 4	**sequence** *sequence-number* **Example:** `se-10-1-0-0(cusp-config-trigger)> sequence 1`	Creates a sequence with the specified number and puts you into trigger sequence command mode. The number indicates the order in which triggers are evaluated. In this case, the sequence that is being created is sequence number 1.
Step 5	**in-network** *network-name* **Example:** `se-10-1-0-0(cusp-config-trigger-seq)> in-network service-provider`	Optional. Specifies the incoming network name for the trigger condition. In this case, the incoming network is the “service-provider” network.
Step 6	**mid-dialog** **Example:** `se-10-1-0-0(cusp-config-trigger-seq)> mid-dialog`	Optional. A special trigger that bypasses routing policies on mid-dialog messages.

	Command or Action	Purpose
Step 7	**end sequence** **Example:** `se-10-1-0-0(cusp-config-trigger-seq)> end sequence`	Exits the trigger sequence command mode.
Step 8	**end trigger condition** **Example:** `se-10-1-0-0(cusp-config-trigger)> end trigger condition`	Exits the trigger command mode.

Example

In this example, Cisco Unified SIP Proxy only reacts based on the network the call came in on, so the triggers are simple.

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> trigger condition call-from-service-provider
se-10-1-0-0(cusp-config-trigger)> sequence 1
se-10-1-0-0(cusp-config-trigger-seq)> in-network service-provider
se-10-1-0-0(cusp-config-trigger-seq)> end sequence
se-10-1-0-0(cusp-config-trigger)> end trigger condition

se-10-1-0-0(cusp-config)> trigger condition mid-dialog
se-10-1-0-0(cusp-config-trigger)> sequence 1
se-10-1-0-0(cusp-config-trigger-seq)> mid-dialog
se-10-1-0-0(cusp-config-trigger-seq)> end sequence
se-10-1-0-0(cusp-config-trigger)> end trigger condition
```

Configuring Server Groups

About Server Groups

Server groups define the elements that Cisco Unified SIP Proxy interacts with for each network. The server group name that is used is inserted into the SIP URI of the outgoing request. Some devices, such as Cisco Unified Communications Manager, validate the URI of requests before processing, which means that the end device might need to be configured with a Fully Qualified Domain Name (FQDN) to allow for this.

Two of the fields for each individual element, q-value and weight, are important to use to specify the priorities of elements, and also for load balancing. Calls are routed to specific elements based on q-value. The element with the highest q-value receives all traffic routed to that server group. If multiple elements have the same q-value, traffic is distributed between them based on the load-balancing option used. The

default load-balancing is based on call-id, but weight can also be used. If weight is used, the percentage of traffic that an element receives is equal to its weight divided by the sum of up elements with the same q-value's weights. The sum of their weights does not need to equal 100. You can change the weights and q-values to configure a different priority or load-balancing scheme.

Summary Steps

1. **cusp**
2. **configure**
3. **server-group sip group** *server-group-name network*
4. **element ip-address** *ipaddress port* **{udp | tcp | tls} [q-value** *q-value*] **[weight** *weight*]
5. **lb-type {global | highest-q | request-uri | call-id | to-uri | weight }**
6. **end server-group**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**server-group sip group** *server-group-name network* **Example:** `se-10-1-0-0(cusp-config)> server-group sip group sp.example.com service-provider`	Creates a SIP server group and enters server group command mode. In this case, the server group being created is called “sp.example.com” and it uses the network called “service-provider”.
Step 4	**element ip-address** *ipaddress port* **{udp | tcp | tls} [q-value** *q-value*] **[weight** *weight*] **Example:** `se-10-1-0-0(cusp-config-sg)> element ip-address 192.168.10.3 5060 tls q-value 1.0 weight 100`	Creates an IP element for a SIP server group and determines the characteristics of the SIP server group. **Note** You can enter this command multiple times.

	Command or Action	Purpose
Step 5	**lb-type {global | highest-q | request-uri | call-id | to-uri | weight }** **Example:** `se-10-1-0-0(cusp-config-sg)> lb-type weight`	Configures the load-balancing algorithm for the SIP server group. In this example, it specifies that the element will be selected proportional to its weight relative to the weights of other elements of the same q-value.
Step 6	**end server-group** **Example:** `se-10-1-0-0(cusp-config-sg)> end server-group`	Exits the server group command mode.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> server-group sip group sp.example.com service-provider
se-10-1-0-0(cusp-config-sg)> element ip-address 192.168.10.3 5060 tls q-value 1.0 weight
100
se-10-1-0-0(cusp-config-sg)> element ip-address 192.168.10.4 5060 tls q-value 1.0 weight
50
se-10-1-0-0(cusp-config-sg)> element ip-address 192.168.10.5 5060 tls q-value 1.0 weight
50
se-10-1-0-0(cusp-config-sg)> lb-type weight
se-10-1-0-0(cusp-config-sg)> end server-group
```

Configuring Route Tables

About Route Tables

You must configure route tables to direct SIP requests to their appropriate destinations. Each route table consists of a set of keys that are matched based on the lookup policy. For example, each key might represent the prefix of a phone number dialed.

Summary Steps

1. **cusp**
2. **configure**
3. **route table** *table-name*
4. **key** *key* **response** *response-code*
5. **key** *key* **target-destination** *target-destination network*

6. **end route table**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**route table** *table-name* **Example:** `se-10-1-0-0(cusp-config)> route table service-provider-table`	Creates a route table and enters route table command mode. In this case, it creates a route table called "service-provider-table".
Step 4	**key** *key* **response** *response-code* **Example:** `se-10-1-0-0(cusp-config-rt)> key * response 404`	Assigns a response code to a lookup key. In this example, it returns a response of "404" to everything.
Step 5	**key** *key* **target-destination** *target-destination network* **Example:** `se-10-1-0-0(cusp-config-rt)> key 510 target-destination cube-sp.example.com cube-sp`	Replaces the key part of the target destination with a specified value. **Note** You can enter this command multiple times.
Step 6	**end route table** **Example:** `se-10-1-0-0(cusp-config-rt)> end route table`	Exits the route table command mode.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> route table service-provider-table
se-10-1-0-0(cusp-config-rt)> key * response 404
se-10-1-0-0(cusp-config-rt)> key 510 target-destination cube-sp.example.com cube-sp
se-10-1-0-0(cusp-config-rt)> end route table
```

Configuring Normalization Policies

Normalization policies modify SIP messages to account for incompatibilities between networks. In this case, the service provider cannot handle phone numbers with the escape sequence "91," so the sequence must be removed from the request-uri and TO header.

Summary Steps

1. **cusp**
2. **configure**
3. **policy normalization** *policy_name*
4. **uri-component update request-uri {user | host | host-port | phone | uri} {all |** *match-string*}
   replace-string
5. **uri-component update header {first | last | all} {user | host | host-port | phone | uri} {all |**
   match-string} replace-string
6. **end policy**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**policy normalization** *policy-name* **Example:** `se-10-1-0-0(cusp-config)> policy normalization outgoing-norm-policy`	Creates a normalization policy and enters policy normalization command mode. In this example, the normalization policy is called "outgoing-norm-policy".
Step 4	**uri-component update request-uri {user | host | host-port | phone | uri} {all |** *match-string*} *replace-string* **Example:** `se-10-1-0-0(cusp-config-norm)> uri-component update request-uri user ^91 ""`	Configures a normalization policy step that updates a URI component field within a request URI.

	Command or Action	Purpose
Step 5	**uri-component update header {first | last | all} {user | host | host-port | phone | uri} {all |** *match-string} replace-string* **Example:** `se-10-1-0-0(cusp-config-norm)> uri-component update TO all user ^91 ""`	Configures a normalization policy step that updates a URI component field within a header of the source message.
Step 6	**end policy** **Example:** `se-10-1-0-0(cusp-config-norm)> end policy`	Exits policy normalization command mode.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> policy normalization outgoing-norm-policy
se-10-1-0-0(cusp-config-norm)> uri-component update request-uri user ^91 ""
se-10-1-0-0(cusp-config-norm)> uri-component update TO all user ^91 ""
se-10-1-0-0(cusp-config-norm)> end policy
```

Configuring Lookup Policies

Lookup policies decide how the keys in the route tables are used. Each key represents the beginning of the phone number dialed because each policy states to match the user component of the request-uri against the keys in its route table. The user component of the request-uri is the phone number called. The rule used to match is prefix, which means that the longest prefix match in the route table is used. So if the dialed number is 510-1XX-XXXX, the call is sent to the cme.example.com server group. If the dialed number is 510-XXX-XXXX, the call is sent to the cucm.example.com server group. The four policies in the following example are identical, except that they each refer to their specific table.

Summary Steps

1. **cusp**
2. **configure**
3. **policy lookup** *policy-name*
4. **sequence** *sequence-number table-name* **field** {**in-network | local-ip-address | local-ip-port | remote-ip-address | remote-ip-port**} **| header** {**p-asserted identity| from | to | diversion| remote-party-id**} **| request uri [uri component** {**param| user | phone | host| host-port| uri**}**]**
5. **rule {exact | prefix | subdomain | subnet | fixed** *length***} [case-insensitive]**
6. **end sequence**

7. **end policy**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**policy lookup** *policy-name* **Example:** `se-10-1-0-0(cusp-config)> policy lookup service-provider-policy`	Creates a policy with the specified name and enters policy lookup command mode. In this case, creates a policy called “service-provider-policy”.
Step 4	**sequence** *sequence-number table-name* **field {in-network | local-ip-address | local-ip-port | remote-ip-address | remote-ip-port} | header {p-asserted identity| from | to | diversion| remote-party-id} | request uri [uri component {param| user | phone | host| host-port| uri}]** **Example:** `se-10-1-0-0(cusp-config-lookup)> sequence 1`	Creates a sequence with the specified number and enters policy lookup sequence command mode. Sequences are performed according to the order of their number.
Step 5	**rule {exact | prefix | subdomain | subnet | fixed** *length***} [case-insensitive]** **Example:** `se-10-1-0-0(cusp-config-lookup-seq)> rule prefix`	Creates a rule that determines the routing algorithm for the lookup policy. In this case, it creates a rule that specifies that the lookup policy searches for the longest prefix match.
Step 6	**end sequence** **Example:** `se-10-1-0-0(cusp-config-lookup-seq)> end sequence`	Exits policy lookup sequence command mode.
Step 7	**end policy** **Example:** `se-10-1-0-0(cusp-config-lookup)> end policy`	Exits policy lookup command mode.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> policy lookup service-provider-policy
```

```
se-10-1-0-0(cusp-config-lookup)> sequence 1 service-provider-table request-uri
uri-component user
se-10-1-0-0(cusp-config-lookup-seq)> rule prefix
se-10-1-0-0(cusp-config-lookup-seq)> end sequence
se-10-1-0-0(cusp-config-lookup)> end policy
```

Configuring Routing Triggers

Routing triggers correlate trigger conditions with lookup policies. A single policy is chosen based on which corresponding condition is matched. The conditions are evaluated in ascending order based on sequence number. The mid-dialog condition is the first one so that the policy step is skipped for mid-dialog messages. Based on the following configuration, after the INVITE message is successfully routed, all subsequent messages (which are mid-dialog) bypass routing policies.

Summary Steps

1. **cusp**
2. **configure**
3. **trigger routing sequence** *sequence-number* **{by-pass | policy** *policy*} **[condition** *trigger-condition*]

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger routing sequence** *sequence-number* **{by-pass | policy** *policy*} **[condition** *trigger-condition*] **Example:** `se-10-1-0-0(cusp-config)> trigger routing sequence 2 policy service-provider-policy condition call-from-service-provider`	Associates a routing policy with a trigger condition. In this example, the second sequence follows the previously-created policy called "service-provider-policy" and the previously-created trigger called "call-from-service-provider".

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> trigger routing sequence 1 by-pass condition mid-dialog
se-10-1-0-0(cusp-config)> trigger routing sequence 2 policy service-provider-policy
condition call-from-service-provider
se-10-1-0-0(cusp-config)> trigger routing sequence 3 policy cube-sp-policy condition
call-from-cube-sp
se-10-1-0-0(cusp-config)> trigger routing sequence 4 policy cube-es-policy condition
call-from-cube-es
se-10-1-0-0(cusp-config)> trigger routing sequence 5 policy enterprise-policy condition
call-from-enterprise
```

Configuring Normalization Triggers

Normalization triggers correlate trigger conditions with normalization policies. There are two types of triggers: pre-normalization, which occurs before routing, and post-normalization, which occurs after routing. Similar to routing policies, a special policy bypasses normalization on mid-dialog messages.

Summary Steps

1. **cusp**
2. **configure**
3. **trigger pre-normalization sequence** *sequence-number* {**by-pass** | **policy** *policy*} [**condition** *trigger-condition*]

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.

	Command or Action	Purpose
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger pre-normalization sequence** *sequence-number* **{by-pass | policy** *policy*} [**condition** *trigger-condition*] **Example:** `se-10-1-0-0(cusp-config)> trigger pre-normalization sequence 2 policy outgoing-norm-policy condition call-from-cube-sp`	Configures a pre-normalization algorithm for incoming SIP messages to a normalization policy. In this example, the second sequence follows the previously-created policy called "outgoing-norm-policy" and the previously-created trigger called "call-from-cube-sp".

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> trigger pre-normalization sequence 1 by-pass condition
mid-dialog
se-10-1-0-0(cusp-config)> trigger pre-normalization sequence 2 policy outgoing-norm-policy
condition call-from-cube-sp
```

Configuring Listen and Record-Route Ports

You must configure listen and record-route ports for each network. For the listen and record-route ports, the actual addresses of the Cisco Unified SIP Proxy module are used. The **sip record-route** command inserts the record-route header into outgoing requests. The **sip listen** command allows for Cisco Unified SIP Proxy to accept incoming requests on that port.

Summary Steps

1. **cusp**
2. **configure**
3. **sip record-route** *network_name* **{tcp | tls | udp}** *ip_address [port]*
4. **sip listen** *network_name* **{tcp | tls | udp}** *ip_address port*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip record-route** *network_name* **{tcp | tls | udp}** *ip_address [port]* **Example:** `se-10-1-0-0(cusp-config)> sip record-route service-provider udp 10.10.10.99 5060`	Enables record-routing for a SIP network. In this example, the "service-provider" network is associated with a record-route configuration and the IP address that populates the record-route header field is "10.10.10.99" and the port that populates the record-route header is 5060.
Step 4	**sip listen** *network_name* **{tcp | tls | udp}** *ip_address port* **Example:** `se-10-1-0-0(cusp-config)> sip listen service-provider udp 10.10.10.99 5060`	Creates a listener that listens for SIP traffic on a specific SIP network, host, and port.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> sip record-route service-provider udp 10.10.10.99 5060
se-10-1-0-0(cusp-config)> sip listen service-provider udp 10.10.10.99 5060
```

Configuring a Hostname

If the upstream element is using DNS SRV for routing to the two Cisco Unified SIP Proxies in a network, you must configure the two Cisco Unified SIP Proxies to have the same FQDN by entering the **sip alias** command in Cisco Unified SIP Proxy configuration mode on both Cisco Unified SIP Proxies.

Summary Steps

1. **cusp**

2. **configure**
3. **sip alias** *hostname*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip alias** *hostname* **Example:** `se-10-1-0-0(cusp-config)> sip alias myhost`	Configures the hostname of this instance.

Example

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> sip alias myhost
```

Configuring Transport Layer Security (TLS)

Creating and Importing a Signed Certificate

Cisco Unified SIP Proxy supports TLS, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). Establishing TLS connections requires some extra steps because the connections require authentication using signed certificates.

Prerequisites

You need an SFTP server or HTTP to import certificate requests.

Summary Steps

1. **configure terminal**
2. **crypto key generate [rsa {label** *label-name* **| modulus** *modulus-size***} | default]**
3. **crypto key certreq label** *label-name* **url {sftp: | http:}**
4. **crypto key import rsa label** *label-name* **{der url {sftp: | http: } | pem { terminal | url {sftp: | http: }} [default]**
5. **crypto key import cer label mykey url sftp:**
6. **offline**
7. **reload**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `se-10-1-0-0# configure terminal`	Enters configuration mode.
Step 2	**crypto key generate [rsa {label** *label-name* **| modulus** *modulus-size***} | default]** **Example:** `se-10-1-0-0(config)> crypto key generate rsa label mykey modulus 512 default`	Creates an RSA private key.
Step 3	**crypto key certreq label** *label-name* **url {sftp: | http:}** **Example:** `se-10-1-0-0(config)> crypto key certreq label mykey url sftp:`	Creates a certificate request to be signed.
Step 4	**crypto key import rsa label** *label-name* **{der url {sftp: | http: } | pem { terminal | url {sftp: | http: }} [default]** **Example:** `se-10-1-0-0(config)> crypto key import trustcacert label rootCA url sftp:`	After the certificate request is signed, imports the trusted certificate authority (CA) certificate that you used to sign the request.

	Command or Action	Purpose
Step 5	**crypto key import rsa label** *label-name* **{der url {sftp: | http: } | pem { terminal | url {sftp: | http: }} [default]** **Example:** `se-10-1-0-0(config)> crypto key import cer label mykey url sftp:`	After the root CA is imported, imports the signed certificate.

Step 6	**offline**
	
	Example:
	`se-10-1-0-0> offline`
	`!!!WARNING!!!: Putting the system offline will terminate all active calls.`
	`Do you wish to continue[n]?: y`
		Initiates Cisco Unified SIP Proxy offline mode.

Step 7	**reload** **Example:** `se-10-1-0-0(offline)> reload`	Restarts the Cisco Unified SIP Proxy system and enables Cisco Unified SIP Proxy to verify the imported trusted certificate.

Example of Creating a Signed Certificate

```
se-10-1-0-0# configure terminal
se-10-1-0-0(config)> crypto key generate rsa label mykey modulus 512 default
Key generation in progress. Please wait...
The label name for the key is mykey

se-10-1-0-0(config)> crypto key certreq label mykey url sftp:
Address or name of remote host? test:test123@192.168.202.216
Username (ENTER if none)? anonymous
Password (not shown)?
Destination path? netmod/mykey.csr
Uploading CSR file succeed

se-10-1-0-0(config)> crypto key import trustcacert label rootCA url sftp:
Import certificate file...
Address or name of remote host? test:test123@192.168.202.216
Source filename? netmod/rootCA/cacert.pem
1212 bytes received.

se-10-1-0-0(config)> crypto key import cer label mykey url sftp:
Import certificate file...
Address or name of remote host? test:test123@192.168.202.216
Source filename? netmod/mycert.cer
952 bytes received.
Import succeeded
```

What To Do Next

- Import the trusted CA certificates for any of the TLS peer elements.

Creating and Importing a Self-Signed Certificate

Summary Steps

1. **vim** <*filename*> (This is an example only. You can use any text editor as such.)
2. **openssl req -new -newkey rsa:2048 -nodes -keyout** <*key*> **-out** <*csr*> **-config** <*configuration file name*>
3. **openssl x509 -req -days** <*days*> **-in** <*csr*> **-signkey** <*key*> **-out** <*certificate*>
4. **configure terminal**
5. **crypto key import trustcacert label** <*label_name*> **terminal**

Note Execute the steps 4 and 5 on the Unified SIP Proxy CLI command. Use a different host to run the steps 1 through 3, such as Linux, where OpenSSL is available.

Detailed Steps

	Command or Action	Purpose

Step 1	**vim** <*filename*> (This is an example only. You can use any text editor as such.)
	
	Example:
	`Linux-server-test$ vim abc`
	`distinguished_name = req_distinguished_name`
	`[ req_distinguished_name ]`
	`countryName = Country Name (2 letter code)`
	`countryName_default = US`
	`countryName_min = 2`
	`countryName_max = 2`
	
	`stateOrProvinceName = State or Province Name (full name)`
	`stateOrProvinceName_default = California`
	
	`localityName = Locality Name (eg, city)`
	`localityName_default = San Jose`
	
	`organizationName = Organization Name (eg, company)`
	`organizationName_default = Cisco Systems, Inc.`
	
	`organizationalUnitName = Organizational Unit Name (eg, section)`
	`organizationalUnitName_default = Cisco Webex`
	
	`commonName = Common Name (eg, YOUR name)`
	`commonName_max = 64`
	`emailAddress = Email Address`
	`emailAddress_default = csg-avops@cisco.com`
	`emailAddress_max = 40`
		On a Linux server, create a configuration file.

Step 2	**openssl req -new -newkey rsa:2048 -nodes -keyout** <*key*> **-out** <*csr*> **-config** <*configuration file name*> **Example:** `openssl req -new -newkey rsa:2048 -nodes -keyout me90sjvce001.webex.com.key -out me90sjvce001.webex.com.csr -config abc`	Generate key and csr pair.
Step 3	**openssl x509 -req -days** <*days*> **-in** <*csr*> **-signkey** <*key*> **-out** <*certificate*> **Example:** `openssl x509 -req -days 720 -in me90sjvce001.webex.com.csr -signkey me90sjvce001.webex.com.key -out me90sjvce001.webex.com.cer`	Sign the CSR file with your own key.

	Command or Action	Purpose
Step 4	**configure terminal** **Example:** `se-10-0-0-0> configure terminal`	Log in to Cisco Unified SIP Proxy and enter the configuration mode.
Step 5	**crypto key import trustcacert label** *<label_name>* **terminal** **Example:** `se-10-0-0-0(config)#` **crypto key import trustcacert label sample_cert terminal** **% Self-signed CA certificate:** **-----BEGIN CERTIFICATE-----** **MIIDLjCCAhagAwIBAgIBATANBgkqhkiG9w0BAQUFADAwMS4wLAYDVQQ** **DEyVJT1Mt** **U2VsZi1TaWduZWQtQ2VydGlmaWNhdGUtMzc4ODk3MTYzMB4XDTE3MDE** **xODA2MzYy** **N1oXDTIwMDEwMTAwMDAwMFowMDEuMCwGA1UEAxMlSU9TLVNlbGYtU2l** **nbmVkLUNl** **cnRpZmljYXRlLTM3ODg5NzE2MzCCASIwDQYJKoZIhvcNAQEBBQADggE** **PADCCAQoC** **ggEBAI/k+Jl/RdXkUu3aBp8qIMVA7ifpRehG9AXJKlqOafc9Ly92hwN** **xeLGV/U8k** **Xlo/fuoyaNyLIu9GwS1BfvM3yHOthhX+T5RHgcj3s1Yctl6HUW93M/E** **JYluo5RDE** **NAXJ2UXa/Utl9ZGjCvat8h3N4QduP2ulIsK1IqyYLDRwD1fiSNFrdZB** **2zzIE1M7g** **eeitn4n1INHiVtH0jOmO4En/FjUa3YPCFEyB1/U17YGWN/GOHguCsZl** **uL8WywAT5** **Pq1uaipVxWoCzXCb74BSxTJiHs/tmPGkIHi57RvLKxgqr5vHXCOsWsQ** **6/C9z6My3** **tvE6dtLHuP2RgR6r+3xOhKdqcHECAwEAAaNTMFEwDwYDVR0TAQH/BAU** **wAwEB/zAf** **BgNVHSMEGDAWgBSIzQOOrJrnxzR8LEQ2VIIfVFpO2DAdBgNVHQ4EFgQ** **UiM0Djqya** **58c0fCxENlSCH1RaTtgwDQYJKoZIhvcNAQEFBQADggEBABhYrhWv9DZ** **0sZZt7Smc** **o5pgIIFFOtGQYc+ei7H6QNzW5iNSZbSPBAIpmVMQWHVS6cOvJ/N63ay** **Q+1TN3rZm** **wmOU9tFExBzjge0nX+Go+0KdWNNQG4XO8SU7BKwM8iWTsM1jT1j6cb9** **Bv1kMgXW0** **5K5AzVYTbaTP/OMoMCsuOJts+GI/Q82H7tlIbdJFbbu3iVEN+gf3coU** **rHa4X2jLr** **K3EVLniCLedkcXdy5TppTvQM9j1FzkGMIrWAlFlp/Vh2CTigJy8GZ4p** **Wt5QzjO6m** **KuP6FZxGPNe8F5BsFCWNM5aHPa8MUqlFKZMuUb50w43SZRT3xfI2WLv** **1yd49f65T** **mBA=** **-----END CERTIFICATE-----**	Import the Self-Signed Certificate to Cisco Unified SIP Proxy. Copy the content of the certificate created and paste it when prompted.

Example

On a Linux server, execute the following commands:

```
Linux-server-test$ vim abc
distinguished_name     = req_distinguished_name
[ req_distinguished_name ]
countryName          = Country Name (2 letter code)
countryName_default = US
countryName_min     = 2
countryName_max     = 2

stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = California

localityName          = Locality Name (eg, city)
localityName_default  = San Jose

organizationName         = Organization Name (eg, company)
organizationName_default = Cisco Systems, Inc.

organizationalUnitName         = Organizational Unit Name (eg, section)
organizationalUnitName_default = Cisco Webex

commonName       = Common Name (eg, YOUR name)
commonName_max   = 64
emailAddress     = Email Address
emailAddress_default = csg-avops@cisco.com
emailAddress_max = 40


openssl req   -new -newkey rsa:2048 -nodes  -keyout me90sjvce001.webex.com.key -out
me90sjvce001.webex.com.csr -config abc

openssl x509 -req -days 720 -in me90sjvce001.webex.com.csr -signkey
me90sjvce001.webex.com.key -out me90sjvce001.webex.com.cer
```

Log in to the Cisco Unified SIP Proxy and execute the following commands:

```
se-10-0-0-0> configure terminal
se-10-0-0-0(config)# crypto key import trustcacert label sample_cert terminal

% Self-signed CA certificate:
-----BEGIN CERTIFICATE-----
MIIDLjCCAhagAwIBAgIBATANBgkqhkiG9w0BAQUFADAwMS4wLAYDVQQDEyVJT1Mt
U2VsZi1TaWduZWQtQ2VydGlmaWNhdGUtMzc4ODk3MTYzMB4XDTE3MDExODA2MzYy
N1oXDTIwMDEwMTAwMDAwMFowMDEuMCwGA1UEAxMlSU9TLVNlbGYtU2lnbmVkLUNl
cnRpZmljYXRlLTM3ODg5NzE2MzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAI/k+Jl/RdXkUu3aBp8qIMVA7ifpRehG9AXJKlqOafc9Ly92hwNxeLGV/U8k
Xlo/fuoyaNyLIu9GwS1BfvM3yHOthhX+T5RHgcj3s1Yct16HUW93M/EJYluo5RDE
NAXJ2UXa/Utl9ZGjCvat8h3N4QduP2ulIsK1IqyYLDRwD1fiSNFrdZB2zzIE1M7g
eeitn4n1INHiVtH0jOmO4En/FjUa3YPCFEyB1/U17YGWN/GOHguCsZluL8WywAT5
Pq1uaipVxWoCzXCb74BSxTJiHs/tmPGkIHi57RvLKxgqr5vHXCOsWsQ6/C9z6My3
tvE6dtLHuP2RgR6r+3xOhKdqcHECAwEAAaNTMFEwDwYDVR0TAQH/BAUwAwEB/zAf
BgNVHSMEGDAWgBSIzQOOrJrnxzR8LEQ2VIIfVFpO2DAdBgNVHQ4EFgQUiM0Djqya
58c0fCxEN1SCH1RaTtgwDQYJKoZIhvcNAQEFBQADggEBABhYrhWv9DZ0sZZt7Smc
o5pgIIFFOtGQYc+ei7H6QNzW5iNSZbSPBAIpmVMQWHVS6cOvJ/N63ayQ+1TN3rZm
wmOU9tFExBzjge0nX+Go+0KdWNNQG4XO8SU7BKwM8iWTsM1jT1j6cb9Bv1kMgXW0
5K5AzVYTbaTP/OMoMCsuOJts+GI/Q82H7tlIbdJFbbu3iVEN+gf3coUrHa4X2jLr
K3EVLniCLedkcXdy5TppTvQM9j1FzkGMIrWAlFlp/Vh2CTigJy8GZ4pWt5QzjO6m
KuP6FZxGPNe8F5BsFCWNM5aHPa8MUqlFKZMuUb50w43SZRT3xfI2WLv1yd49f65T
mBA=
-----END CERTIFICATE-----
```

Updating Web Session with an Imported Signed Certificate

From Cisco Unified SIP Proxy Release 10.1 onwards, HTTPS is enabled by default. You need not manually generate a crypto key and pass it to the web session security to enable HTTPS. However, you should be able to import a signed certificate that you generated externally, and update the web session with this new key label.

Summary Steps

1. **configure**
2. **crypto key import rsa label** *label-name* **{der url {sftp: | http: } | pem { terminal | url {sftp: | http: }} [default]**
3. **web session security keylabel** *labelname*
4. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `se-10-1-0-0# configure terminal`	Enters configuration mode.
Step 2	**crypto key import rsa label** *label-name* **{der url {sftp: | http: } | pem { terminal | url {sftp: | http: }} [default]** **Example:** `se-10-1-0-0(config)> crypto key import cer label mykey url sftp:`	Imports the signed certificate.
Step 3	**web session security keylabel** *labelname* **Example:** `se-10-1-0-0(cusp-config)> web session security keylabel mykey`	Associates a security key for HTTPS.
Step 4	**end** **Example:** `se-10-1-0-0(cusp-config)> end`	Exits to privileged EXEC mode.

Example of Updating Web Session with an Imported Signed Certificate

```
se-10-1-0-0# configure terminal
se-10-1-0-0(config)> crypto key import cer label mykey url sftp:
Import certificate file...
Address or name of remote host? 192.0.2.2
```

```
Source filename? netmod/mycert.cer
952 bytes received.
Import succeeded
se-10-1-0-0(cusp-config)> web session security keylabel mykey
se-10-1-0-0(cusp-config)> end
```

Configuring TLS on Cisco Unified SIP Proxy

After you import the certificates, you must enable TLS connections. If you want more security, you can create a list of trusted peers. If you create such a list, only connections from those peers are accepted. The peer's hostname entry must be the peer's subjectAltName in its certificate. If subjectAltName is not used in the certificate, the peer's hostname entry must be CN.

Summary Steps

1. **cusp**
2. **configure**
3. **sip tls**
4. **sip tls trusted-peer** {*peer's-hostname*}
5. **sip tls connection-setup-timeout** *{value in seconds}*
6. **sip tls [v1.0 | v1.1 | v1.2]**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip tls** **Example:** `se-10-1-0-0(cusp-config)> sip tls`	Enables the use of SIP TLS connections with other SIP entities, providing secure communication over the Internet.

	Command or Action	Purpose
Step 4	**sip tls trusted-peer** {*peer's-hostname*} **Example:** `se-10-1-0-0(cusp-config)> sip tls trusted-peer example.com`	Creates a list of trusted peers.
Step 5	**sip tls connection-setup-timeout** {*value in seconds*} **Example:** `se-10-1-0-0(cusp-config)> sip tls connection-setup-timeout <1-60>`	It is the time specified in Cisco Unified SIP Proxy by the user to establish connection with the trusted peer. The default value is 1 second. The range of values is 1 to 60 seconds.
Step 6	**sip tls [v1.0 | v1.1 | v1.2]** **Example:** `se-10-1-0-0(cusp-config)> sip tls v1.0`	Enables SIP TLS versions. The default value is all TLS versions with fall-back.The connection between the user and the trusted peer fails to establish when the user tries to connect using the TLS version that the trusted peer does not support. In the case where the trusted peer does not support a specific TLS version, the user retries the connection with the trusted peer using the downgraded version of TLS. For example, if the trusted peer does not support TLS v1.2, then the user retries the connection using TLS v1.1.

Example of Configuring TLS

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> sip tls
se-10-1-0-0(cusp-config)> sip tls trusted-peer example.com
se-10-1-0-0(cusp-config)> sip tls connection-setup-timeout <1-60>
se-10-1-0-0(cusp-config)> sip tls v1.2
```

Note From Cisco Unified Proxy Release 10.1 onwards, HTTPS is enabled by default. You need not manually generate a crypto key and pass it to the web session security to enable HTTPS. Cisco Unified Proxy Release 10.1 supports only TLS v1.2 for HTTPS. If you delete the certificate from the web session security and try to login through HTTP, you will be redirected to HTTPS. Only the latest connection is retained and the remaining connections are logged out.

Configuring Lite Mode

One of the ways you can configure the performance of the Cisco Unified SIP Proxy is to switch the module to Lite Mode. In Lite Mode, which requires you to disable record-route, the module's performance is boosted. In standard mode, the module processes calls up to the licensed limit.

By default, the module is in standard mode.

For information on the performance difference when using Lite Mode versus standard mode, see the Release Notes for Cisco Unified SIP Proxy Release 10.1.

Summary Steps

1. **cusp**
2. **configure**
3. **lite-mode**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**lite-mode** **Example:** `se-10-1-0-0(cusp-config)> lite-mode`	Puts the Cisco Unified SIP Proxy module into Lite Mode.

Example

The following example puts the module into Lite Mode:

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> lite-mode
```

Configuring Performance Control

About Performance Control

One of the ways you can configure the performance of the Cisco Unified SIP Proxy is to restrict the number of calls that the Cisco Unified SIP Proxy can handle.

Summary Steps

1. **cusp**
2. **configure**
3. **call-rate-limit** *limit*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-1-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-1-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**call-rate-limit** *limit* **Example:** `se-10-1-0-0(cusp-config)> call-rate-limit 50`	Sets the maximum call rate that the Cisco Unified SIP Proxy can handle.

Example

The following example limits the number of calls that the system can process to 50:

```
se-10-1-0-0> cusp
se-10-1-0-0(cusp)> configure
se-10-1-0-0(cusp-config)> call-rate-limit 50
```

Committing the Configuration

Now you must commit the configuration. Committing the configuration serves two purposes: the configuration becomes active, and is persisted.

- To see the current active configuration, enter the **show configuration active** command.
- To see what the active configuration will be after you commit your changes, enter the **show configuration candidate** command.
- To commit the configuration for this example, enter the following command:

```
se-10-1-0-0(cusp-config)> commit
```