



Provisioning Methods

- [Provision a Phone with BroadSoft Server](#), on page 1
- [Provisioning Examples Overview](#), on page 2
- [Basic Resync](#), on page 2
- [TFTP Resync](#), on page 3
- [Unique Profiles, Macro Expansion, and HTTP](#), on page 6
- [Resync a Device Automatically](#), on page 9
- [Set Up Your Phones for Activation Code Onboarding](#), on page 16
- [Migrate Your Phone to Enterprise Phone Directly](#), on page 18
- [Secure HTTPS Resync](#), on page 19
- [Profile Management](#), on page 26
- [Set the Phone Privacy Header](#), on page 28
- [Renew the MIC Certificate](#), on page 29

Provision a Phone with BroadSoft Server

BroadSoft Server user only.

You can register your Cisco IP multiplatform phones to a BroadWorks platform.

Procedure

- Step 1** Download the CPE Kit from BroadSoft Xchange. To get the latest CPE kits, go to this URL: <https://xchange.broadsoft.com>.
- Step 2** Upload the most recent DTAF file to the BroadWorks (system level) server.
For more information, go to this URL: (<https://xchange.broadsoft.com/node/1031047>). Access the *BroadSoft Partner Configuration Guide* and see the section “*Configure BroadWorks Device Profile Type*”.
- Step 3** Configure Broadworks Device Profile Type.
For more information on how to configure the device profile type, go to this URL: <https://xchange.broadsoft.com/node/1031047>. Access the *BroadSoft Partner Configuration Guide* and see the section “*Broadworks Device Profile Type Configuration*”.
-

Provisioning Examples Overview

This chapter provides example procedures for transferring configuration profiles between the phone and the provisioning server.

For information about creating configuration profiles, refer to [Provisioning Formats](#).

Basic Resync

This section demonstrates the basic resync functionality of the phones.

Use Syslog to Log Messages

A phone can be configured to send logging messages to a syslog server over UDP, including messages related to provisioning. To identify this server, you can access the phone Web interface (see [Access the Phone Web Interface](#)), select **Voice > System** and identify the server in the **Syslog Server** parameter of the **Optional Network Configuration** section. Configure the syslog server IP address into the device and observe the messages that are generated during the remaining procedures.

To get the information, you can access the phone Web interface, select **Info > Debug Info > Control Logs** and click **messages**.

Before you begin

Procedure

- Step 1** Install and activate a syslog server on the local PC.
- Step 2** Program the PC IP address into the Syslog_Server parameter of the profile and submit the change:

```
<Syslog_Server>192.168.1.210</Syslog_Server>
```

- Step 3** Click the **System** tab and enter the value of your local syslog server into the Syslog_Server parameter.
- Step 4** Repeat the resync operation as described in [TFTP Resync, on page 3](#).

The device generates two syslog messages during the resync. The first message indicates that a request is in progress. The second message marks success or failure of the resync.

- Step 5** Verify that your syslog server received messages similar to the following:

```
CP-78xx-3PCC 00:0e:08:ab:cd:ef -- Requesting resync tftp://192.168.1.200/basic.txt
```

Detailed messages are available by programming a Debug_Server parameter (instead of the Syslog_Server parameter) with the IP address of the syslog server, and by setting the Debug_Level to a value between 0 and 3 (3 being the most verbose):

```
<Debug_Server>192.168.1.210</Debug_Server>
```

```
<Debug_Level>3</Debug_Level>
```

The contents of these messages can be configured by using the following parameters:

- Log_Request_Msg
- Log_Success_Msg
- Log_Failure_Msg

If any of these parameters are cleared, the corresponding syslog message is not generated.

TFTP Resync

The phone supports multiple network protocols for retrieving configuration profiles. The most basic profile transfer protocol is TFTP (RFC1350). TFTP is widely used for the provisioning of network devices within private LAN networks. Although not recommended for the deployment of remote endpoints across the Internet, TFTP can be convenient for deployment within small organizations, for in-house preprovisioning, and for development and testing. See [In-House Device Preprovisioning](#) for more information on in-house preprovisioning. In the following procedure, a profile is modified after downloading a file from a TFTP server.

Procedure

- Step 1** Within a LAN environment, connect a PC and a phone to a hub, switch, or small router.
- Step 2** On the PC, install and activate a TFTP server.
- Step 3** Use a text editor to create a configuration profile that sets the value for GPP_A to 12345678 as shown in the example.
- ```
<flat-profile>
 <GPP_A> 12345678
</GPP_A>
</flat-profile>
```
- Step 4** Save the profile with the name `basic.txt` in the root directory of the TFTP server.
- You can verify that the TFTP server is properly configured: request the `basic.txt` file by using a TFTP client other than the phone. Preferably, use a TFTP client that is running on a separate host from the provisioning server.
- Step 5** Open the PC web browser to the admin/advanced configuration page. For example, if the IP address of the phone is 192.168.1.100:
- ```
http://192.168.1.100/admin/advanced
```
- Step 6** Select the **Voice > Provisioning** tab, and inspect the values of the general purpose parameters GPP_A through GPP_P. These should be empty.
- Step 7** Resync the test phone to the `basic.txt` configuration profile by opening the resync URL in a web browser window.

If the IP address of the TFTP server is 192.168.1.200, the command should be similar to the following example:

```
http://192.168.1.100/admin/resync?tftp://192.168.1.200/basic.txt
```

When the phone receives this command, the device at address 192.168.1.100 requests the file `basic.txt` from the TFTP server at IP address 192.168.1.200. The phone then parses the downloaded file and updates the `GPP_A` parameter with the value 12345678.

Step 8 Verify that the parameter was correctly updated: Refresh the configuration page on the PC web browser and select the **Voice > Provisioning** tab.

The `GPP_A` parameter should now contain the value 12345678.

Log Messages to the Syslog Server

If a syslog server is configured on the phone through the use of the parameters, the resync and upgrade operations send messages to the syslog server. A message can be generated at the start of a remote file request (configuration profile or firmware load), and at the conclusion of the operation (indicating either success or failure).

You can also configure the parameters in the phone configuration file with XML(`cfg.xml`) code. To configure each parameter, see the syntax of the string in [System Log Parameters, on page 5](#).

Before you begin

- A syslog server is installed and configured.
- Access the phone administration web page. See [Access the Phone Web Interface](#).

Procedure

- Step 1** Click **Voice > System**.
- Step 2** In the **Optional Network Configuration** section, enter the server IP in **Syslog Server** and optionally specify a **Syslog Identifier** as defined in [System Log Parameters, on page 5](#).
- Step 3** Optionally define the content of the syslog messages using **Log Request Msg**, **Log Success Msg**, and **Log Failure Msg** as defined in [System Log Parameters, on page 5](#).
- The fields defining syslog message content are located in the **Configuration Profile** section on the **Voice > Provisioning** tab. If you don't specify the message content, the default settings in the fields are used. If any of the fields are cleared, the corresponding message is not generated.
- Step 4** Click **Submit All Changes** to apply the configuration.
- Step 5** Verify the validity of the configuration.
- Perform a TFTP resync. See [TFTP Resync, on page 3](#).

The device generates two syslog messages during the resync. The first message indicates that a request is in progress. The second message marks the success or failure of the resync.
 - Verify that your syslog server received messages similar to the following:

```
CP-78xx-3PCC 00:0e:08:ab:cd:ef -- Requesting resync tftp://192.168.1.200/basic.txt
CP-88xx-3PCC 00:0e:08:ab:cd:ef -- Successful resync tftp://192.168.1.200/basic.txt
```

System Log Parameters

The following table defines the function and usage of the syslog parameters in the **Optional Network Configuration** section under the **Voice > System** tab in the phone web page. It also defines the syntax of the string that is added in the phone configuration file (cfg.xml) with XML code to configure a parameter.

Table 1: Syslog Parameters

Parameter Name	Description and Default Value
Syslog Server	<p>Specify the server for logging the phone system information and critical events. If both Debug Server and Syslog Server are specified, Syslog messages are also logged to the Debug Server.</p> <ul style="list-style-type: none"> • In the phone configuration file (cfg.xml) with XML, enter a string in this format: <pre><Syslog_Server ua="na">10.74.30.84</Syslog_Server></pre> • On the phone web page, specify the Syslog server.
Syslog Identifier	<p>Select the device identifier to include in syslog messages that are uploaded to the syslog server. The device identifier appears after the timestamp in each message. The options for the identifiers are:</p> <ul style="list-style-type: none"> • None: No device identifier. • \$MA: The MAC address of the phone, expressed as continuous lower case letters and digits. Example: c4b9cd811e29 • \$MAU: The MAC address of the phone, expressed as continuous upper case letters and digits. Example: C4B9CD811E29 • \$MAC: The MAC address of the phone in the standard colon-separated format. Example: c4:b9:cd:81:1e:29 • \$SN: The product serial number of the phone. • In the phone configuration file with XML(cfg.xml), enter a string in this format: <pre><Syslog_Identifier ua="na">\$MAC</Syslog_Identifier></pre> • On the phone web page, select an identifier from the list. <p>Default: None</p>

Parameter Name	Description and Default Value
Log Request Msg	<p>The message that is sent to the syslog server at the start of a resync attempt. If no value is specified, the syslog message is not generated.</p> <p>The default value is <code>\$PN \$MAC -- Requesting resync \$SCHEME://\$SERVIP:\$PORT\$PATH</code></p> <ul style="list-style-type: none"> • In the phone configuration file with XML(cfg.xml), enter a string in this format: <pre><Log_Request_Msg ua="na">\$PN \$MAC -- Requesting resync \$SCHEME://\$SERVIP:\$PORT\$PATH</Log_Request_Msg></pre>
Log Success Msg	<p>The syslog message that is issued upon successful completion of a resync attempt. If no value is specified, the syslog message is not generated.</p> <p>In the phone configuration file with XML(cfg.xml), enter a string in this format: <pre><Log_Success_Msg ua="na">\$PN \$MAC -- Successful resync \$SCHEME://\$SERVIP:\$PORT\$PATH</Log_Success_Msg></pre> </p>
Log Failure Msg	<p>The syslog message that is issued after a failed resync attempt. If no value is specified, the syslog message is not generated.</p> <p>The default value is <code>\$PN \$MAC -- Resync failed: \$ERR</code></p> <p>In the phone configuration file with XML(cfg.xml), enter a string in this format: <pre><Log_Failure_Msg ua="na">\$PN \$MAC -- Resync failed: \$ERR</Log_Failure_Msg></pre> </p>

Unique Profiles, Macro Expansion, and HTTP

In a deployment where each phone must be configured with distinct values for some parameters, such as `User_ID` or `Display_Name`, the service provider can create a unique profile for each deployed device and host those profiles on a provisioning server. Each phone, in turn, must be configured to resync to its own profile according to a predetermined profile naming convention.

The profile URL syntax can include identifying information that is specific to each phone, such as MAC address or serial number, by using the macro expansion of built-in variables. Macro expansion eliminates the need to specify these values in multiple locations within each profile.

A profile rule undergoes macro expansion before the rule is applied to the phone. The macro expansion controls a number of values, for example:

- `$MA` expands to the unit 12-digit MAC address (using lower case hex digits). For example, `000e08abcdef`.
- `$SN` expands to the unit serial number. For example, `88012BA01234`.

Other values can be macro expanded in this way, including all the general purpose parameters, `GPP_A` through `GPP_P`. An example of this process can be seen in [TFTP Resync, on page 3](#). Macro expansion is not limited to the URL file name, but can also be applied to any portion of the profile rule parameter. These parameters are referenced as `$A` through `$P`. For a complete list of variables that are available for macro expansion, see [Macro Expansion Variables](#).

In this exercise, a profile specific to a phone is provisioned on a TFTP server.

Provision a Specific IP Phone Profile on a TFTP Server

Procedure

- Step 1** Obtain the MAC address of the phone from its product label. (The MAC address is the number, using numbers and lower-case hex digits, such as 000e08aabbcc.)
- Step 2** Copy the `basic.txt` configuration file (described in [TFTP Resync, on page 3](#)) to a new file named `CP-xxxx-3PCC macaddress.cfg` (replacing `xxxx` with the model number and `macaddress` with the MAC address of the phone).
- Step 3** Move the new file in the virtual root directory of the TFTP server.
- Step 4** Access the phone administration web page. See [Access the Phone Web Interface](#).
- Step 5** Select **Voice > Provisioning**.
- Step 6** Enter `tftp://192.168.1.200/CP-78xx-3PCC$MA.cfg` in the **Profile Rule** field, where `xx` is the model number.
Example: 7841

```
<Profile_Rule>
  tftp://192.168.1.200/CP-7841-3PCC$MA.cfg
</Profile_Rule>
```

Example: 7832

```
<Profile_Rule>
  tftp://192.168.1.200/CP-7832-3PCC$MA.cfg
</Profile_Rule>
```

- Step 7** Click **Submit All Changes**. This causes an immediate reboot and resync.
When the next resync occurs, the phone retrieves the new file by expanding the `$MA` macro expression into its MAC address.
-

HTTP GET Resync

HTTP provides a more reliable resync mechanism than TFTP because HTTP establishes a TCP connection and TFTP uses the less reliable UDP. In addition, HTTP servers offer improved filtering and logging features compared to TFTP servers.

On the client side, the phone does not require any special configuration setting on the server to be able to resync by using HTTP. The `Profile_Rule` parameter syntax for using HTTP with the GET method is similar to the syntax that is used for TFTP. If a standard web browser can retrieve a profile from your HTTP server, the phone should be able to do so as well.

Resync with HTTP GET

Procedure

- Step 1** Install an HTTP server on the local PC or other accessible host.
The open source Apache server can be downloaded from the internet.
- Step 2** Copy the `basic.txt` configuration profile (described in [TFTP Resync, on page 3](#)) onto the virtual root directory of the installed server.
- Step 3** To verify proper server installation and file access to `basic.txt`, access the profile with a web browser.
- Step 4** Modify the `Profile_Rule` of the test phone to point to the HTTP server in place of the TFTP server, so as to download its profile periodically.

For example, assuming the HTTP server is at 192.168.1.300, enter the following value:
- ```
<Profile_Rule>
http://192.168.1.200/basic.txt
</Profile_Rule>
```
- Step 5** Click **Submit All Changes**. This causes an immediate reboot and resync.
- Step 6** Observe the syslog messages that the phone sends. The periodic resyncs should now be obtaining the profile from the HTTP server.
- Step 7** In the HTTP server logs, observe how information that identifies the test phone appears in the log of user agents.  
  
This information should include the manufacturer, product name, current firmware version, and serial number.
- 

## Provisioning Through Cisco XML

For each of the phones, designated as `xxxx` here, you can provision through Cisco XML functions.

You can send an XML object to the phone by a SIP Notify packet or an HTTP Post to the CGI interface of the phone: `http://IPAddressPhone/CGI/Execute`.

The `CP-xxxx-3PCC` extends the Cisco XML feature to support provisioning via an XML object:

```
<CP-xxxx-3PCCExecute>
 <ExecuteItem URL=Resync:[profile-rule]/>
</CP-xxxx-3PCCExecute>
```

After the phone receives the XML object, it downloads the provisioning file from `[profile-rule]`. This rule uses macros to simplify the development of the XML services application.

## URL Resolution with Macro Expansion

Subdirectories with multiple profiles on the server provide a convenient method for managing a large number of deployed devices. The profile URL can contain:



- A provisioning server name or an explicit IP address. If the profile identifies the provisioning server by name, the phone performs a DNS lookup to resolve the name.
- A nonstandard server port that is specified in the URL by using the standard syntax `:port` following the server name.
- The subdirectory of the server virtual root directory where the profile is stored, specified by using standard URL notation and managed by macro expansion.

For example, the following `Profile_Rule` requests the profile file (`$PN.cfg`), in the server subdirectory `/cisco/config`, from the TFTP server that is running on host `prov.telco.com` listening for a connection on port 6900:

```
<Profile_Rule>
tftp://prov.telco.com:6900/cisco/config/$PN.cfg
</Profile_Rule>
```

A profile for each phone can be identified in a general purpose parameter, with its value referred within a common profile rule by using macro expansion.

For example, assume `GPP_B` is defined as `Dj6Lmp23Q`.

The `Profile_Rule` has the value:

```
tftp://prov.telco.com/cisco/$B/$MA.cfg
```

When the device resyncs and the macros are expanded, the phone with a MAC address of `000e08012345` requests the profile with the name that contains the device MAC address at the following URL:

```
tftp://prov.telco.com/cisco/Dj6Lmp23Q/000e08012345.cfg
```

## Resync a Device Automatically

A device can resync periodically to the provisioning server to ensure that any profile changes made on the server are propagated to the endpoint device (as opposed to sending an explicit resync request to the endpoint).

To cause the phone to periodically resync to a server, a configuration profile URL is defined by using the `Profile_Rule` parameter, and a resync period is defined by using the `Resync_Periodic` parameter.

### Before you begin

Access the phone administration web page. See [Access the Phone Web Interface](#).

### Procedure

- 
- Step 1** Select **Voice > Provisioning**.
  - Step 2** Define the `Profile_Rule` parameter. This example assumes a TFTP server IP address of `192.168.1.200`.
  - Step 3** In the **Resync Periodic** field, enter a small value for testing, such as **30** seconds.
  - Step 4** Click **Submit all Changes**.

With the new parameter settings, the phone resyncs twice a minute to the configuration file that the URL specifies.

**Step 5** Observe the resulting messages in the syslog trace (as described in the [Use Syslog to Log Messages, on page 2](#) section).

**Step 6** Ensure that the **Resync On Reset** field is set to **Yes**.

```
<Resync_On_Reset>Yes</Resync_On_Reset>
```

**Step 7** Power cycle the phone to force it to resync to the provisioning server.

If the resync operation fails for any reason, such as if the server is not responding, the unit waits (for the number of seconds configured in **Resync Error Retry Delay**) before it attempts to resync again. If **Resync Error Retry Delay** is zero, the phone does not try to resync after a failed resync attempt.

**Step 8** (Optional) Set the value of **Resync Error Retry Delay** field to a small number, such as **30**.

```
<Resync_Error_Retry_Delay>30</Resync_Error_Retry_Delay>
```

**Step 9** Disable the TFTP server, and observe the results in the syslog output.

## Profile Resync Parameters


The following table defines the function and usage of the profile resync parameters in the **Configuration Profile** section under the **Voice > Provisioning** tab in the phone web page. It also defines the syntax of the string that is added in the phone configuration file (cfg.xml) with XML code to configure a parameter.

Parameter	Description
Provision Enable	<p>Allows or denies configuration profile resync actions.</p> <ul style="list-style-type: none"> <li><b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format:  <pre>&lt;Provision_Enable ua="na"&gt;Yes&lt;/Provision_Enable&gt;</pre></li> <li><b>On the phone web page</b>, set this field to <b>Yes</b> to allow resync actions, or <b>No</b> to block resync actions.</li> </ul> <p>Default: Yes</p>
Resync On Reset	<p>Specifies whether the phone resynchronizes configurations with the provisioning server after power-up and after each upgrade attempt.</p> <ul style="list-style-type: none"> <li><b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format:  <pre>&lt;Resync_On_Reset ua="na"&gt;Yes&lt;/Resync_On_Reset&gt;</pre></li> <li><b>On the phone web page</b>, set this field to <b>Yes</b> to allow resync on power-up or reset, or <b>No</b> to block resync on power-up or reset.</li> </ul> <p>Default: Yes</p>

Parameter	Description
Resync Random Delay	<p>Prevents an overload of the provisioning server when a large number of devices power-on simultaneously and attempt initial configuration. This delay is effective only on the initial configuration attempt, following a device power-on or reset.</p> <p>The parameter is the maximum time interval that the device waits before making contact with the provisioning server. The actual delay is a pseudo-random number between 0 and this value.</p> <p>This parameter is in units of 20 seconds.</p> <p>The valid value ranges between 0 and 65535.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format:  <pre>&lt;Resync_Random_Delay ua="na"&gt;2&lt;/Resync_Random_Delay&gt;</pre> </li> <li>• <b>On the phone web page</b>, specify the number of the units (20 seconds) for the phone to delay resync after power-up or reset.</li> </ul> <p>The default value is 2 (40 seconds).</p>
Resync At (HHmm)	<p>The time (HHmm) that the phone resynchronizes with the provisioning server.</p> <p>The value for this field must be a four-digit number ranging from 0000 to 2400 to indicate the time in HHmm format. For example, 0959 indicates 09:59.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format:  <pre>&lt;Resync_At_HHmm ua="na"&gt;0959&lt;/Resync_At_HHmm &gt;</pre> </li> <li>• <b>On the phone web page</b>, specify the time in HHMM format for the phone to start resync.</li> </ul> <p>The default value is empty. If the value is invalid, the parameter is ignored. If this parameter is set with a valid value, the <b>Resync Periodic</b> parameter is ignored.</p>
Resync At Random Delay	<p>Prevents an overload of the provisioning server when a large number of devices power on simultaneously.</p> <p>To avoid flooding resync requests to the server from multiple phones, the phone resynchronizes in the range between the hours and minutes, and the hours and minutes plus the random delay (hhmm, hhmm+random_delay). For example, if the random delay = (Resync At Random Delay + 30)/60minutes, the input value in seconds is converted to minutes, rounding up to the next minute to calculate the final random_delay interval.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format:  <pre>&lt;Resync_At_Random_Delay ua="na"&gt;600&lt;/Resync_At_Random_Delay&gt;</pre> </li> <li>• <b>On the phone web page</b>, specify the time period in seconds.</li> </ul> <p>The valid value ranges between 600 and 65535.</p> <p>If the value is less than 600, the random delay internal is between 0 and 600.</p> <p>The default value is 600 seconds (10minutes).</p>

Parameter	Description
Resync Periodic	<p>The time interval between periodic resynchronization with the provisioning server. The associated resync timer is active only after the first successful sync with the server.</p> <p>The valid formats are as follows:</p> <ul style="list-style-type: none"> <li>• An integer Example: An input of <b>3000</b> indicates that the next resync occurs in 3000 seconds.</li> <li>• Multiple integers Example: An input of <b>600 , 1200 , 300</b> indicates that the first resync occurs in 600 seconds, the second resync occurs in 1200 seconds after the first one, and the third resync occurs in 300 seconds after the second one.</li> <li>• A time range Example, an input of <b>2400+30</b> indicates that the next resync occurs in between 2400 and 2430 seconds after a successful resync.</li> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <code>&lt;Resync_Periodic ua="na"&gt;3600&lt;/Resync_Periodic&gt;</code></li> <li>• <b>On the phone web page</b>, specify the time period in seconds.</li> </ul> <p>Set this parameter to zero to disable periodic resynchronization.</p> <p>The default value is 3600 seconds.</p>

Parameter	Description
Resync Error Retry Delay	<p>If a resync operation fails because the phone was unable to retrieve a profile from the server, or the downloaded file is corrupt, or an internal error occurs, the phone tries to resync again after a time specified in seconds.</p> <p>The valid formats are as follows:</p> <ul style="list-style-type: none"> <li>• An integer Example: An input of <b>300</b> indicates that the next retry for resync occurs in 300 seconds.</li> <li>• Multiple integers Example: An input of <b>600 , 1200 , 300</b> indicates that the first retry occurs in 600 seconds after the failure, the second retry occurs in 1200 seconds after the failure of the first retry, and the third retry occurs in 300 seconds after the failure of the second retry.</li> <li>• A time range Example, an input of <b>2400+30</b> indicates that the next retry occurs in between 2400 and 2430 seconds after a resync failure.</li> </ul> <p>If the delay is set to 0, the device does not try to resync again following a failed resync attempt.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Resync_Error_Retry_Delay ua="na"&gt;60,120,240,480,960,1920,3840,7680,15360,30720,61440,86400&lt;/Resync_Error_Retry_Delay&gt;</pre></li> <li>• <b>On the phone web page</b>, specify the time period in seconds.</li> </ul> <p>Default: 60,120,240,480,960,1920,3840,7680,15360,30720,61440,86400</p>
Forced Resync Delay	<p>Maximum delay (in seconds) the phone waits before performing a resynchronization.</p> <p>The device does not resync while one of its phone lines is active. Because a resync can take several seconds, it is desirable to wait until the device has been idle for an extended period before resynchronizing. This allows a user to make calls in succession without interruption.</p> <p>The device has a timer that begins counting down when all of its lines become idle. This parameter is the initial value of the counter. Resync events are delayed until this counter decrements to zero.</p> <p>The valid value ranges between 0 and 65535.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Forced_Resync_Delay ua="na"&gt;14400&lt;/Forced_Resync_Delay&gt;</pre></li> <li>• <b>On the phone web page</b>, specify the time period in seconds.</li> </ul> <p>The default value is 14,400seconds.</p>

Parameter	Description
Resync From SIP	<p>Controls requests for resync operations via a SIP NOTIFY event sent from the service provider proxy server to the phone. If enabled, the proxy can request a resync by sending a SIP NOTIFY message containing the Event: resync header to the device.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Resync_From_SIP ua="na"&gt;Yes&lt;/Resync_From_SIP&gt;</pre> </li> <li>• <b>On the phone web page</b>, select <b>Yes</b> to enable this feature, or <b>No</b> to disable it.</li> </ul> <p>Default: Yes</p>
Resync After Upgrade Attempt	<p>Enables or disables the resync operation after any upgrade occurs. If <b>Yes</b> is selected, sync is triggered after a firmware upgrade.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Resync_After_Upgrade_Attempt ua="na"&gt;Yes&lt;/Resync_After_Upgrade_Attempt&gt;</pre> </li> <li>• <b>On the phone web page</b>, select <b>Yes</b> to trigger resync after a firmware upgrade, or <b>No</b> to not resync.</li> </ul> <p>Default: Yes</p>
Resync Trigger 1 Resync Trigger 2	<p>If the logical equation in these parameters evaluates to FALSE, resync is not triggered even when <b>Resync On Reset</b> is set to <b>TRUE</b>. Only the resync via direct action URL and SIP notify ignores these resync triggers.</p> <p>The parameters can be programmed with a conditional expression that undergoes macro expansion. For the valid macro expansions, see <a href="#">Macro Expansion Variables</a>.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Resync_Trigger_1 ua="na"&gt;\$SUPGTMR gt 300 and \$PRVTMR ge 600&lt;/Resync_Trigger_1&gt; &lt;Resync_Trigger_2 ua="na"/&gt;</pre> </li> <li>• <b>On the phone web page</b>, specify the triggers.</li> </ul> <p>Default: Blank</p>
User Configurable Resync	<p>Allows a user to resync the phone from the phone screen menu. When set to <b>Yes</b>, a user can resync the phone configuration by entering the profile rule from the phone. When set to <b>No</b>, the <b>Profile rule</b> parameter isn't displayed on the phone screen menu. The <b>Profile rule</b> parameter is located under <b>Applications</b>  <b>&gt; Device administration</b>.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;User_Configurable_Resync ua="na"&gt;Yes&lt;/User_Configurable_Resync&gt;</pre> </li> <li>• <b>On the phone web page</b>, select <b>Yes</b> to show the <b>Profile rule</b> parameter on the phone menu, or select <b>No</b> to hide this parameter.</li> </ul> <p>Default: Yes</p>

Parameter	Description
Resync Fails On FNF	<p>A resync is typically considered unsuccessful if a requested profile is not received from the server. This parameter override this behavior. When set to <b>No</b>, the device accepts a <code>file-not-found</code> response from the server as a successful resync.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Resync_Fails_On_FNF ua="na"&gt;Yes&lt;/Resync_Fails_On_FNF&gt;</pre> </li> <li>• <b>On the phone web page</b>, select <b>Yes</b> to take a <code>file-not-found</code> response as an unsuccessful resync, or select <b>No</b> to take a <code>file-not-found</code> response as a successful resync.</li> </ul> <p>Default: Yes</p>
Profile Authentication Type	<p>Specifies the credentials to use for profile account authentication. The available options are:</p> <ul style="list-style-type: none"> <li>• <b>Disabled</b>: Disables the profile account feature. When this feature is disabled, the <b>Profile account setup</b> menu doesn't display on the phone screen.</li> <li>• <b>Basic HTTP Authentication</b>: The HTTP login credentials are used to authenticate the profile account.</li> <li>• <b>XSI Authentication</b>: XSI login credentials or XSI SIP credentials are used to authenticate the profile account. The authentication credentials depend on the <b>XSI Authentication Type</b> for the phone: <ul style="list-style-type: none"> <li>• When the <b>XSI Authentication Type</b> for the phone is set to <b>Login Credentials</b>, the XSI login credentials are used.</li> <li>• When the <b>XSI Authentication Type</b> for the phone is set to <b>SIP Credentials</b>, the XSI SIP credentials are used.</li> </ul> </li> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Profile_Authentication_Type ua="na"&gt;Basic Http Authentication&lt;/Profile_Authentication_Type&gt;</pre> </li> <li>• <b>On the phone web page</b>, select an option from the list for the phone to authenticate profile resync.</li> </ul> <p>Default: Basic HTTP Authentication</p>

Parameter	Description
Profile Rule Profile Rule B Profile Rule C Profile Rule D	<p>Each profile rule informs the phone of a source from which to obtain a profile (configuration file). During every resync operation, the phone applies all the profiles in sequence.</p> <p>If you are applying AES-256-CBC encryption to the configuration files, specify the encryption key with the <b>--key</b> keyword as follows:</p> <p><b>[--key &lt;encryption key&gt;]</b></p> <p>You can enclose the encryption key in double-quotes (") optionally.</p> <ul style="list-style-type: none"> <li>• <b>In the phone configuration file (cfg.xml) with XML</b>, enter a string in this format: <pre>&lt;Profile_Rule ua="na"/&gt;/\$PSN.xml&lt;/Profile_Rule&gt; &lt;Profile_Rule_B ua="na"/&gt; &lt;Profile_Rule_C ua="na"/&gt; &lt;Profile_Rule_D ua="na"/&gt;</pre> </li> <li>• <b>On the phone web page</b>, specify the profile rule.</li> </ul> <p>Default: <b>/\$PSN.xml</b></p>
DHCP Option To Use	<p>DHCP options, delimited by commas, used to retrieve firmware and profiles.</p> <p>Default: 66,160,159,150,60,43,125</p>
DHCPv6 Option To Use	<p>DHCP options, delimited by commas, used to retrieve firmware and profiles.</p> <p>Default: 17,160,159</p>

## Set Up Your Phones for Activation Code Onboarding

If your network is configured for Activation Code Onboarding, you can set up new phones to register automatically in a secure way. You generate and provide each user with a unique 16-digit activation code. The user enters the activation code, and the phone automatically registers. This feature keeps your network secure because the phone can't register until the user enters a valid activation code.

Activation codes can be used only once, and have an expiry date. If a user enters an expired code, the phone displays `Invalid activation code` on the screen. If this happens, provide the user with a new code.

This feature is available in firmware release 11-2-3MSR1, BroadWorks Application Server Release 22.0 (patch AP.as.22.0.1123.ap368163 and its dependencies). However, you can change phones with older firmware to use this feature. To do this, use the following procedure.

### Before you begin

Ensure that you allow the `activation.webex.com` service through your firewall to support onboarding via activation code.

If you want to set up a proxy server for the onboarding, ensure that the proxy server is configured correctly. See [Set Up a Proxy Server](#).



Access the phone web page. [Access the Phone Web Interface](#)

### Procedure

- 
- Step 1** Reset the phone to the factory settings.
- Step 2** Select **Voice > Provisioning > Configuration Profile**.
- Step 3** Enter the profile rule in the **Profile Rule** field as described in the [Activation Code Provisioning Parameters, on page 17](#) table.
- Step 4** (Optional) In the **Firmware Upgrade** section, enter the upgrade rule in the **Upgrade Rule** field as described in the [Activation Code Provisioning Parameters, on page 17](#) table.
- Step 5** Submit All Changes.
- 

## Activation Code Provisioning Parameters

The following table defines the function and usage of the activation code parameters in the **Configuration Profile** section under the **Voice > Provisioning** tab in the phone web page. It also defines the syntax of the string that is added in the phone configuration file (cfg.xml) with XML code to configure a parameter.

Parameter	Description
Profile Rule	Remote configuration profile rules evaluated in sequence. Each resync operation can retrieve multiple files, potentially managed by different servers.  Perform one of the following: <ul style="list-style-type: none"> <li>In the phone configuration file (cfg.xml) with XML, enter a string in this format:               <pre>&lt;Profile_Rule ua="na"&gt;gds://&lt;/Profile_Rule&gt;</pre> </li> <li>In the phone web interface, enter a string in this format:               <pre>gds://</pre> </li> </ul> Default: /\$PSN.xml
Profile Rule B	
Profile Rule C	
Profile Rule D	

Parameter	Description
Upgrade Rule	<p>Specifies the firmware upgrade script that defines upgrade conditions and associated firmware URLs. It uses the same syntax as Profile Rule.</p> <p>Perform one of the following:</p> <ul style="list-style-type: none"> <li>In the phone configuration file (cfg.xml) with XML, enter a string in this format: <pre>&lt;Upgrade_Rule ua="na"&gt;http://&lt;server ip address&gt;/ sip88xx.11-2-3MSR1-1.loads&lt;/Upgrade_Rule&gt;</pre> </li> <li>In the phone web interface, enter the upgrade rule: <pre>protocol://server[:port]/profile_pathname</pre> <p>For example:</p> <pre>tftp://192.168.1.5/image/sip88xx.11-2-3MSR1-1.loads</pre> </li> </ul> <p>If no protocol is specified, TFTP is assumed. If no server-name is specified, the host that requests the URL is used as the server name. If no port is specified, the default port is used (69 for TFTP, 80 for HTTP, or 443 for HTTPS).</p> <p>Default: Blank</p>

## Migrate Your Phone to Enterprise Phone Directly

You can now migrate your phone to enterprise phone easily in one step without using transition firmware load.

### Before you begin

Access the phone administration web page. See [Access the Phone Web Interface](#).

### Procedure

- 
- Step 1** Select **Voice > Provisioning**.
- Step 2** In the **Upgrade Rule** field, set the Upgrade Rule parameter by entering a firmware upgrade script. For the syntax details, see that defines upgrade conditions and associated firmware URLs. It uses the same syntax as Profile Rule. Enter a script and use the following format to enter the upgrade rule:
- ```
<tftp|http|https>://<ipaddress>/image/<load name>
```
- For example:
- ```
tftp://192.168.1.5/image/sip78xx.14-1-1MN-366.loads
```
- Step 3** Configure the **Transition Authorization Rule** parameter by entering a value to obtain and authorize the licence from the server.
- You can also configure this parameter in the configuration file (cfg.xml) by entering a string in this format:
- ```
<Trans_Auth_Rule ua="na">http://10.74.51.81/prov/migration/E2312.lic</Trans_Auth_Rule>
```
- Step 4** In the **Transition Authorization Type** parameter, set the license type as **Classic**.

You can also configure this parameter in the configuration file (cfg.xml) by entering a string in this format:

```
<Trans_Auth_Type ua="na">Classic</Trans_Auth_Type>
```

Step 5 Click **Submit All Changes**.

Secure HTTPS Resync

These mechanisms are available on the phone for resyncing by using a secure communication process:

- Basic HTTPS Resync
- HTTPS with Client Certificate Authentication
- HTTPS Client Filtering and Dynamic Content

Basic HTTPS Resync

HTTPS adds SSL to HTTP for remote provisioning so that the:

- The phone can authenticate the provisioning server.
- Provisioning server can authenticate the phone.
- Confidentiality of information exchanged between the phone and the provisioning server is ensured.

SSL generates and exchanges secret (symmetric) keys for each connection between the phone and the server, using public/private key pairs that are pre-installed in the phone and the provisioning server.

On the client side, the phone does not require any special configuration setting on the server to be able to resync using HTTPS. The Profile_Rule parameter syntax for using HTTPS with the GET method is similar to the syntax that is used for HTTP or TFTP. If a standard web browser can retrieve a profile from a your HTTPS server, the phone should be able to do so as well.

In addition to installing a HTTPS server, a SSL server certificate that Cisco signs must be installed on the provisioning server. The devices cannot resync to a server that is using HTTPS unless the server supplies a Cisco-signed server certificate. Instructions for creating signed SSL Certificates for Voice products can be found at <https://supportforums.cisco.com/docs/DOC-9852>.

Authenticate with Basic HTTPS Resync

Procedure

Step 1 Install an HTTPS server on a host whose IP address is known to the network DNS server through normal hostname translation.

The open source Apache server can be configured to operate as an HTTPS server when installed with the open source mod_ssl package.

Step 2 Generate a server Certificate Signing Request for the server. For this step, you might need to install the open source OpenSSL package or equivalent software. If using OpenSSL, the command to generate the basic CSR file is as follows:

```
openssl req -new -out provserver.csr
```

This command generates a public/private key pair, which is saved in the `privkey.pem` file.

Step 3 Submit the CSR file (`provserver.csr`) to Cisco for signing.

A signed server certificate is returned (`provserver.cert`) along with a Sipura CA Client Root Certificate, `spacroot.cert`.

See <https://supportforums.cisco.com/docs/DOC-9852> for more information

Step 4 Store the signed server certificate, the private key pair file, and the client root certificate in the appropriate locations on the server.

In the case of an Apache installation on Linux, these locations are typically as follows:

```
# Server Certificate:
SSLCertificateFile /etc/httpd/conf/provserver.cert
# Server Private Key:
SSLCertificateKeyFile /etc/httpd/conf/pivkey.pem
# Certificate Authority:
SSLCACertificateFile /etc/httpd/conf/spacroot.cert
```

Step 5 Restart the server.

Step 6 Copy the `basic.txt` configuration file (described in [TFTP Resync, on page 3](#)) onto the virtual root directory of the HTTPS server.

Step 7 Verify proper server operation by downloading `basic.txt` from the HTTPS server by using a standard browser from the local PC.

Step 8 Inspect the server certificate that the server supplies.

The browser probably does not recognize the certificate as valid unless the browser has been pre-configured to accept Cisco as a root CA. However, the phones expect the certificate to be signed this way.

Modify the `Profile_Rule` of the test device to contain a reference to the HTTPS server, for example:

```
<Profile_Rule>
https://my.server.com/basic.txt
</Profile_Rule>
```

This example assumes the name of the HTTPS server is `my.server.com`.

Step 9 Click **Submit All Changes**.

Step 10 Observe the syslog trace that the phone sends.

The syslog message should indicate that the resync obtained the profile from the HTTPS server.

Step 11 (Optional) Use an Ethernet protocol analyzer on the phone subnet to verify that the packets are encrypted.

In this exercise, client certificate verification was not enabled. The connection between the phone and server is encrypted. However, the transfer is not secure because any client can connect to the server and request the file, given knowledge of the file name and directory location. For secure resync, the server must also authenticate

the client, as demonstrated in the exercise described in [HTTPS with Client Certificate Authentication, on page 21](#).

HTTPS with Client Certificate Authentication

In the factory default configuration, the server does not request an SSL client certificate from a client. Transfer of the profile is not secure because any client can connect to the server and request the profile. You can edit the configuration to enable client authentication; the server requires a client certificate to authenticate the phone before it accepts a connection request.

Because of this requirement, the resync operation cannot be independently tested by using a browser that lacks the proper credentials. The SSL key exchange within the HTTPS connection between the test phone and the server can be observed with the `ssldump` utility. The utility trace shows the interaction between client and server.

Authenticate HTTPS with Client Certificate

Procedure

- Step 1** Enable client certificate authentication on the HTTPS server.
- Step 2** In Apache (v.2), set the following in the server configuration file:

```
SSLVerifyClient require
```

Also, ensure that the `spacroot.cert` has been stored as shown in the [Basic HTTPS Resync, on page 19](#) exercise.

- Step 3** Restart the HTTPS server and observe the syslog trace from the phone.
- Each resync to the server now performs symmetric authentication, so that both the server certificate and the client certificate are verified before the profile is transferred.
- Step 4** Use `ssldump` to capture a resync connection between the phone and the HTTPS server.
- If client certificate verification is properly enabled on the server, the `ssldump` trace shows the symmetric exchange of certificates (first server-to-client, then client-to-server) before the encrypted packets that contain the profile.
- With client authentication enabled, only a phone with a MAC address that matches a valid client certificate can request the profile from the provisioning server. The server rejects a request from an ordinary browser or other unauthorized device.
-

Configure a HTTPS Server for Client Filtering and Dynamic Content

If the HTTPS server is configured to require a client certificate, the information in the certificate identifies the resyncing phone and supplies it with the correct configuration information.

The HTTPS server makes the certificate information available to CGI scripts (or compiled CGI programs) that are invoked as part of the resync request. For the purpose of illustration, this exercise uses the open source Perl scripting language, and assumes that Apache (v.2) is used as the HTTPS server.

Procedure

Step 1 Install Perl on the host that is running the HTTPS server.

Step 2 Generate the following Perl reflector script:

```
#!/usr/bin/perl -wT
use strict;
print "Content-Type: text/plain\n\n";
print "<flat-profile><GPP_D>";

print "OU=$ENV{'SSL_CLIENT_I_DN_OU'},\n";
print "L=$ENV{'SSL_CLIENT_I_DN_L'},\n";
print "S=$ENV{'SSL_CLIENT_I_DN_S'}\n";
print "</GPP_D></flat-profile>";
```

Step 3 Save this file with the file name `reflect.pl`, with executable permission (`chmod 755` on Linux), in the CGI scripts directory of the HTTPS server.

Step 4 Verify accessibility of CGI scripts on the server (that is, `/cgi-bin/...`).

Step 5 Modify the Profile_Rule on the test device to resync to the reflector script, as in the following example:

```
https://prov.server.com/cgi-bin/reflect.pl?
```

Step 6 Click **Submit All Changes**.

Step 7 Observe the syslog trace to ensure a successful resync.

Step 8 Access the phone administration web page. See [Access the Phone Web Interface](#).

Step 9 Select **Voice > Provisioning**.

Step 10 Verify that the GPP_D parameter contains the information that the script captured.

This information contains the product name, MAC address, and serial number if the test device carries a unique certificate from the manufacturer. The information contains generic strings if the unit was manufactured before firmware release 2.0.

A similar script can determine information about the resyncing device and then provide the device with appropriate configuration parameter values.

HTTPS Certificates

The phone provides a reliable and secure provisioning strategy that is based on HTTPS requests from the device to the provisioning server. Both a server certificate and a client certificate are used to authenticate the phone to the server and the server to the phone.

In addition to Cisco issued certifications, the phone also accepts server certificates from a set of commonly used SSL certificate providers.

To use HTTPS with the phone, you must generate a Certificate Signing Request (CSR) and submit it to Cisco. The phone generates a certificate for installation on the provisioning server. The phone accepts the certificate when it seeks to establish an HTTPS connection with the provisioning server.

HTTPS Methodology

HTTPS encrypts the communication between a client and a server, thus protecting the message contents from other network devices. The encryption method for the body of the communication between a client and a server is based on symmetric key cryptography. With symmetric key cryptography, a client and a server share a single secret key over a secure channel that is protected by Public/Private key encryption.

Messages encrypted by the secret key can only be decrypted by using the same key. HTTPS supports a wide range of symmetric encryption algorithms. The phone implements up to 256-bit symmetric encryption, using the American Encryption Standard (AES), in addition to 128-bit RC4.

HTTPS also provides for the authentication of a server and a client engaged in a secure transaction. This feature ensures that a provisioning server and an individual client cannot be spoofed by other devices on the network. This capability is essential in the context of remote endpoint provisioning.

Server and client authentication is performed by using public/private key encryption with a certificate that contains the public key. Text that is encrypted with a public key can be decrypted only by its corresponding private key (and vice versa). The phone supports the Rivest-Shamir-Adleman (RSA) algorithm for public/private key cryptography.

SSL Server Certificate

Each secure provisioning server is issued a secure sockets layer (SSL) server certificate that Cisco signs directly. The firmware that runs on the phone recognizes only a Cisco certificate as valid. When a client connects to a server by using HTTPS, it rejects any server certificate that is not signed by Cisco.

This mechanism protects the service provider from unauthorized access to the phone, or any attempt to spoof the provisioning server. Without such protection, an attacker might be able to reprovision the phone, to gain configuration information, or to use a different VoIP service. Without the private key that corresponds to a valid server certificate, the attacker is unable to establish communication with a phone.

Obtain a Server Certificate

Procedure

Step 1 Contact a Cisco support person who will work with you on the certificate process. If you are not working with a specific support person, email your request to ciscosb-certadmin@cisco.com.

Step 2 Generate a private key that will be used in a CSR (Certificate Signing Request). This key is private and you do not need to provide this key to Cisco support. Use open source “openssl” to generate the key. For example:

```
openssl genrsa -out <file.key> 1024
```

Step 3 Generate a CSR that contains fields that identify your organization and location. For example:

```
openssl req -new -key <file.key> -out <file.csr>
```

You must have the following information:

- Subject field—Enter the Common Name (CN) that must be an FQDN (Fully Qualified Domain Name) syntax. During SSL authentication handshake, the phone verifies that the certificate it receives is from the machine that presented it.
- Server hostname—For example, provserv.domain.com.
- Email address—Enter an email address so that customer support can contact you if needed. This email address is visible in the CSR.

Step 4 Email the CSR (in zip file format) to the Cisco support person or to ciscosb-certadmin@cisco.com. The certificate is signed by Cisco. Cisco sends the certificate to you to install on your system.

Client Certificate

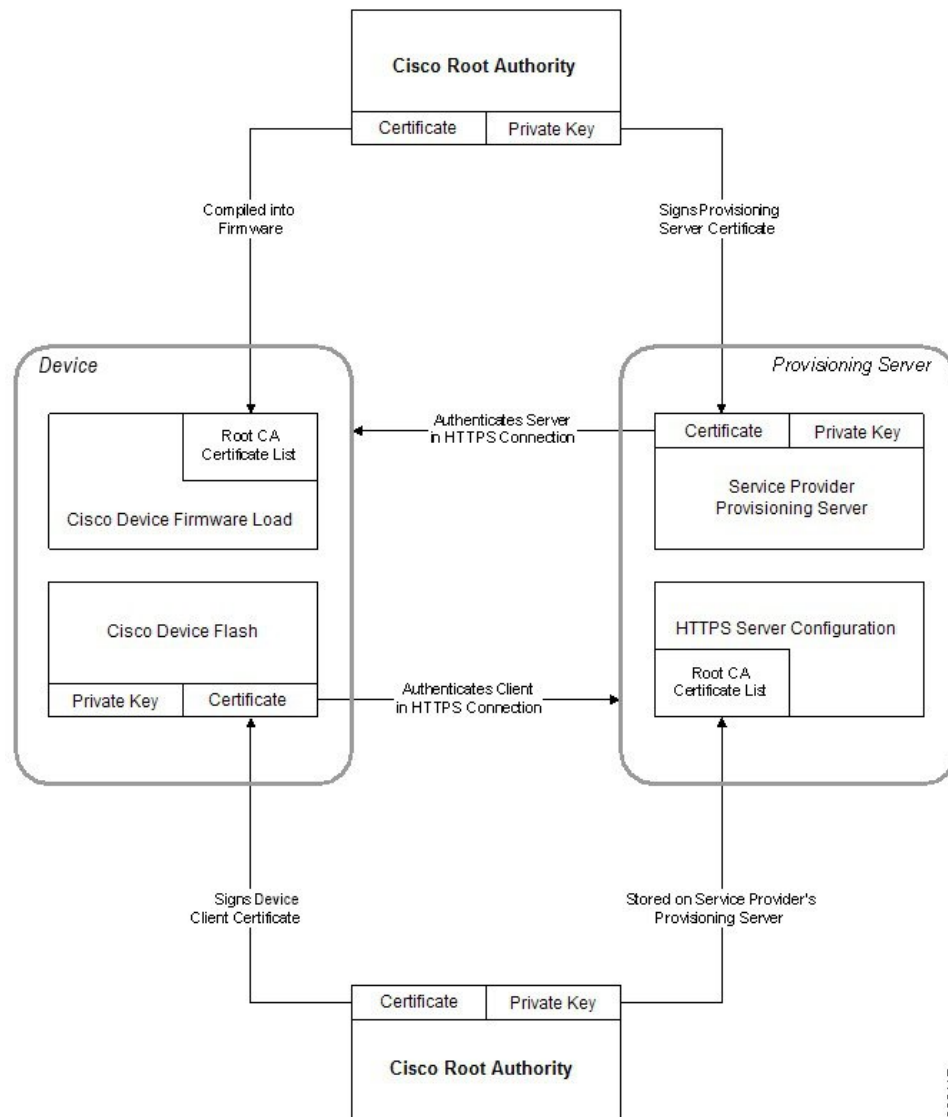
In addition to a direct attack on a phone, an attacker might attempt to contact a provisioning server through a standard web browser or another HTTPS client to obtain the configuration profile from the provisioning server. To prevent this kind of attack, each phone also carries a unique client certificate, signed by Cisco, that includes identifying information about each individual endpoint. A certificate authority root certificate that is capable of authenticating the device client certificate is given to each service provider. This authentication path allows the provisioning server to reject unauthorized requests for configuration profiles.

Certificate Structure

The combination of a server certificate and a client certificate ensures secure communication between a remote phone and its provisioning server. The figure below illustrates the relationship and placement of certificates, public/private key pairs, and signing root authorities, among the Cisco client, the provisioning server, and the certification authority.

The upper half of the diagram shows the Provisioning Server Root Authority that is used to sign the individual provisioning server certificate. The corresponding root certificate is compiled into the firmware, which allows the phone to authenticate authorized provisioning servers.

Figure 1: Certificate Authority Flow



Configure a Custom Certificate Authority

Digital certificates can be used to authenticate network devices and users on the network. They can be used to negotiate IPSec sessions between network nodes.

A third party uses a Certificate Authority certificate to validate and authenticate two or more nodes that are attempting to communicate. Each node has a public and private key. The public key encrypts data. The private key decrypts data. Because the nodes have obtained their certificates from the same source, they are assured of their respective identities.

The device can use digital certificates provided by a third-party Certificate Authority (CA) to authenticate IPSec connections.

The phones support a set of preloaded Root Certificate Authority embedded in the firmware:

- Cisco Small Business CA Certificate

- CyberTrust CA Certificate
- Verisign CA certificate
- Sipura Root CA Certificate
- Linksys Root CA Certificate

Before you begin

Access the phone administration web page. See [Access the Phone Web Interface](#).

Procedure

Step 1 Select **Info > Status**.

Step 2 Scroll to **Custom CA Status** and see the following fields:

- Custom CA Provisioning Status—Indicates the provisioning status.
 - Last provisioning succeeded on mm/dd/yyyy HH:MM:SS; or
 - Last provisioning failed on mm/dd/yyyy HH:MM:SS
 - Custom CA Info—Displays information about the custom CA.
 - Installed—Displays the “CN Value,” where “CN Value” is the value of the CN parameter for the Subject field in the first certificate.
 - Not Installed—Displays if no custom CA certificate is installed.
-

Profile Management

This section demonstrates the formation of configuration profiles in preparation for downloading. To explain the functionality, TFTP from a local PC is used as the resync method, although HTTP or HTTPS can be used as well.

Compress an Open Profile with Gzip

A configuration profile in XML format can become quite large if the profile specifies all parameters individually. To reduce the load on the provisioning server, the phone supports compression of the XML file, by using the deflate compression format that the gzip utility (RFC 1951) supports.



Note Compression must precede encryption for the phone to recognize a compressed and encrypted XML profile.

For integration into customized back-end provisioning server solutions, the open source zlib compression library can be used in place of the standalone gzip utility to perform the profile compression. However, the phone expects the file to contain a valid gzip header.

Procedure

Step 1 Install gzip on the local PC.

Step 2 Compress the `basic.txt` configuration profile (described in [TFTP Resync, on page 3](#)) by invoking gzip from the command line:

```
gzip basic.txt
```

This generates the deflated file `basic.txt.gz`.

Step 3 Save the `basic.txt.gz` file in the TFTP server virtual root directory.

Step 4 Modify the Profile_Rule on the test device to resync to the deflated file in place of the original XML file, as shown in the following example:

```
tftp://192.168.1.200/basic.txt.gz
```

Step 5 Click **Submit All Changes**.

Step 6 Observe the syslog trace from the phone.

Upon resync, the phone downloads the new file and uses it to update its parameters.

Encrypt a Profile with OpenSSL

A compressed or uncompressed profile can be encrypted (however, a file must be compressed before it is encrypted). Encryption is useful when the confidentiality of the profile information is of particular concern, such as when TFTP or HTTP is used for communication between the phone and the provisioning server.

The phone supports symmetric key encryption by using the 256-bit AES algorithm. This encryption can be performed by using the open source OpenSSL package.

Procedure

Step 1 Install OpenSSL on a local PC. This might require that the OpenSSL application be recompiled to enable AES.

Step 2 Using the `basic.txt` configuration file (described in [TFTP Resync, on page 3](#)), generate an encrypted file with the following command:

```
>openssl enc -aes-256-cbc -k MyOwnSecret -in basic.txt -out basic.cfg
```

The compressed `basic.txt.gz` file that was created in [Compress an Open Profile with Gzip, on page 26](#) also can be used, because the XML profile can be both compressed and encrypted.

- Step 3** Store the encrypted `basic.cfg` file in the TFTP server virtual root directory.
- Step 4** Modify the `Profile_Rule` on the test device to resync to the encrypted file in place of the original XML file. The encryption key is made known to the phone with the following URL option:

```
[--key MyOwnSecret ] tftp://192.168.1.200/basic.cfg
```

- Step 5** Click **Submit All Changes**.
- Step 6** Observe the syslog trace from the phone.
- Upon resync, the phone downloads the new file and uses it to update its parameters.

Create Partitioned Profiles

A phone downloads multiple separate profiles during each resync. This practice allows management of different kinds of profile information on separate servers and maintenance of common configuration parameter values that are separate from account specific values.

Procedure

- Step 1** Create a new XML profile, `basic2.txt`, that specifies a value for a parameter that makes it distinct from the earlier exercises. For instance, to the `basic.txt` profile, add the following:

```
<GPP_B>ABCD</GPP_B>
```

- Step 2** Store the `basic2.txt` profile in the virtual root directory of the TFTP server.
- Step 3** Leave the first profile rule from the earlier exercises in the folder, but configure the second profile rule (`Profile_Rule_B`) to point to the new file:

```
<Profile_Rule_B>tftp://192.168.1.200/basic2.txt
</Profile_Rule_B>
```

- Step 4** Click **Submit All Changes**.
- The phone now resyncs to both the first and second profiles, in that order, whenever a resync operation is due.
- Step 5** Observe the syslog trace to confirm the expected behavior.

Set the Phone Privacy Header

A user privacy header in the SIP message sets user privacy needs from the trusted network.

You can set the user privacy header value for each line extension using an XML tag in the `config.xml` file.

The privacy header options are:

- Disabled (default)
- none—The user requests that a privacy service applies no privacy functions to this SIP message.
- header—The user needs a privacy service to obscure headers which cannot be purged of identifying information.
- session—The user requests that a privacy service provide anonymity for the sessions.
- user—The user requests a privacy level only by intermediaries.
- id—The user requests that the system substitute an id that doesn't reveal the IP address or host name.

Procedure

- Step 1** Edit the phone `config.xml` file in a text or XML editor.
- Step 2** Insert the `<Privacy_Header_N_ua="na">Value</Privacy_Header_N_>` tag, where N is the line extension number (1–10), and use one of the following values.
- Default value: **Disabled**
 - **none**
 - **header**
 - **session**
 - **user**
 - **id**
- Step 3** (Optional) Provision any addition line extensions using the same tag with the required line extension number.
- Step 4** Save the changes to the `config.xml` file.
-

Renew the MIC Certificate

You can renew the Manufacture Installed Certificate (MIC) by a specified or default Secure Unique Device Identifier (SUDI) service. If the MIC certificate expires, the features that use SSL/TLS don't work.

Before you begin

- Ensure that you allow the `sudirenewal.cisco.com` service (port 80) through your firewall to support the MIC certificate renewal.
- Access the phone administration web page. See [Access the Phone Web Interface](#).

Procedure

- Step 1** Select **Voice > Provisioning**.

- Step 2** Under the **MIC Cert Settings** section, set the parameters as defined in [Parameters for MIC Certificate Renewal by SUDI Service, on page 30](#).
- Step 3** Click **Submit All Changes**.
After the certificate renewal is completed successfully, the phone reboots.
- Step 4** (Optional) Check the latest status of the MIC certificate renewal under the **MIC Cert Refresh Status** section from **Info > Download Status**.
- Note** If you restore the phone to factory settings, the phone still uses the renewed certificate.

Parameters for MIC Certificate Renewal by SUDI Service

The following table defines the function and usage of each parameter in the **MIC Cert Settings** section of the **Voice > Provisioning** tab.

Table 2: Parameters for MIC Certificate Renewal by SUDI Service

| Parameter Name | Description and Default Value |
|-------------------------|---|
| MIC Cert Refresh Enable | <p>Controls whether to enable the Manufacture Installed Certificate (MIC) renewal by the default or specified Secure Unique Device Identifier (SUDI) service.</p> <p>Perform one of the following:</p> <ul style="list-style-type: none"> In the phone configuration file (cfg.xml) with XML, enter a string in this format:
 <pre><MIC_Cert_Refresh_Enable ua="na">Yes</MIC_Cert_Refresh_Enable></pre> In the phone web interface, select Yes or No to enable or disable the MIC certificate renewal. <p>Valid values: Yes and No
Default: No</p> |

| Parameter Name | Description and Default Value |
|-----------------------|---|
| MIC Cert Refresh Rule | <p>Enter the HTTP URL of the SUDI service that provides the renewed MIC certificate, for example,</p> <pre>http://sudirenewal.cisco.com/</pre> <p>Note Don't change the URL. Only the default URL is supported for the MIC certificate renewal.</p> <p>Perform one of the following:</p> <ul style="list-style-type: none">• In the phone configuration file (cfg.xml) with XML, enter a string in this format:
<pre><MIC_Cert_Refresh_Rule
ua="na">http://sudirenewal.cisco.com/</MIC_Cert_Refresh_Rule></pre>• In the phone web interface, enter the HTTP URL to use. <p>Allowed values: A valid URL not exceeding 1024 characters</p> <p>Default: <code>http://sudirenewal.cisco.com/</code></p> |

