In-House Preprovisioning and Provisioning Servers

In-House Preprovisioning and Provisioning Servers

The service provider preprovisions phones, other than RC units, with a profile. The preprovision profile can comprise a limited set of parameters that resynchronizes the phone. The profile can also comprise a complete set of parameters that the remote server delivers. By default, the phone resynchronizes on power-up and at intervals that are configured in the profile. When the user connects the phone at the customer premises, the device downloads the updated profile and any firmware updates.

This process of preprovisioning, deployment, and remote provisioning can be accomplished in many ways.

Server Preparation and Software Tools

The examples in this chapter require the availability of one or more servers. These servers can be installed and run on a local PC:

- TFTP (UDP port 69)
- syslog (UDP port 514)
- HTTP (TCP port 80)
- HTTPS (TCP port 443).

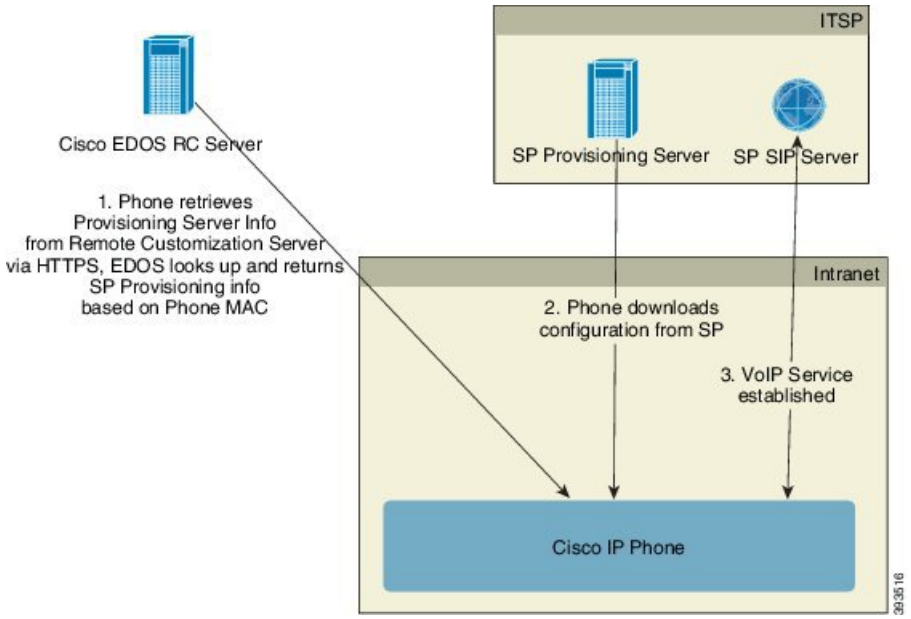
To troubleshoot server configuration, it is helpful to install clients for each type of server on a separate server machine. This practice establishes proper server operation, independent of the interaction with the phones.

We also recommend that you install these software tools:

- To generate configuration profiles, install the open source gzip compression utility.
- For profile encryption and HTTPS operations, install the open source OpenSSL software package.

- To test the dynamic profile generation and one-step remote provisioning using HTTPS, we recommend a scripting language with CGI scripting support. Open source Perl language tools is an example of such a scripting language.
- To verify secure exchanges between provisioning servers and the phones, install an Ethernet packet sniffer (such as the freely downloadable Ethereal/Wireshark). Capture an Ethernet packet trace of the interaction between the phone and the provisioning server. To do so, run the packet sniffer on a PC that is connected to a switch with port mirroring enabled. For HTTPS transactions, you can use the ssldump utility.

Remote Customization (RC) Distribution



All phones contact the Cisco EDOS RC server until they are provisioned initially.

In an RC distribution model, a customer purchases a phone that has already been associated with a specific Service Provider in the Cisco EDOS RC Server. The Internet Telephony Service Provider (ITSP) sets up and maintains a provisioning server, and registers their provisioning server information with the Cisco EDOS RC Server.

When the phone is powered on with an internet connection, the customization state for the unprovisioned phone is **Open**. The phone first queries the local DHCP server for provisioning server information and sets the customization state of the phone. If DHCP query is successful, Customization State is set to **Aborted** and RC is not attempted due to DHCP providing the needed provisioning server information.

When a phone connects to a network for the first time or after a factory reset, if there are no DHCP options setup, it contacts a device activation server for zero touch provisioning. New phones will use “activate.cisco.com” instead of “webapps.cisco.com” for provisioning. Phones with firmware release prior to 11.2(1), will continue to use webapps.cisco.com. Cisco recommends that you allow both the domain names through your firewall.

If DHCP server does not provide provisioning server information, the phone queries the Cisco EDOS RC Server and provides its MAC address and model and the Customization State is set to **Pending**. The Cisco EDOS server responds with the associated service provider's provisioning server information including

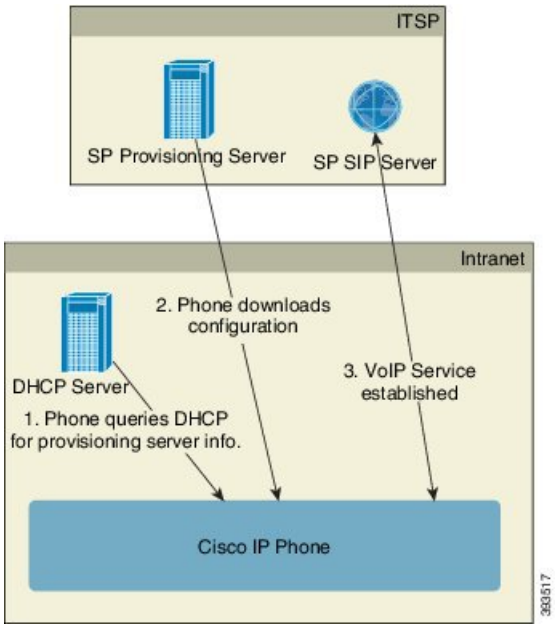
provisioning server URL and the phone's Customization State is set to **Custom Pending**. The phone then performs a resync URL command to retrieve the Service Provider's configuration and, if successful, the Customization State is set to **Acquired**.

If the Cisco EDOS RC Server does not have a service provider associated with the phone, the customization state of the phone is set to **Unavailable**. The phone can be manually configured or an association added for the service provider of the phone to the Cisco EDOS Server.

If a phone is provisioned via either the LCD or Web Configuration Utility, prior to the Customization State becoming **Acquired**, the Customization State is set to **Aborted** and the Cisco EDOS Server will not be queried unless the phone is factory reset.

Once the phone has been provisioned, the Cisco EDOS RC Server is not utilized unless the phone is factory reset.

In-House Device Preprovisioning



With the Cisco factory default configuration, the phone automatically tries to resync to a profile on a TFTP server. A managed DHCP server on a LAN delivers the information about the profile and TFTP server that is configured for preprovisioning to the device. The service provider connects each new phone to the LAN. The phone automatically resyncs to the local TFTP server and initializes its internal state in preparation for deployment. This preprovisioning profile typically includes the URL of a remote provisioning server. The provisioning server keeps the device updated after the device is deployed and connected to the customer network.

The preprovisioned device bar code can be scanned to record its MAC address or serial number before the phone is shipped to the customer. This information can be used to create the profile to which the phone resynchronizes.

Upon receiving the phone, the customer connects it to the broadband link. On power-up, the phone contacts the provisioning server through the URL that is configured through preprovisioning. The phone can thus resync and update the profile and firmware, as necessary.

Related Topics

Retail Distribution

TFTP Provisioning, on page 4

Provisioning Server Setup

This section describes setup requirements for provisioning a phone by using various servers and different scenarios. For the purposes of this document and for testing, provisioning servers are installed and run on a local PC. Also, generally available software tools are useful for provisioning the phones.

TFTP Provisioning

The phones support TFTP for both provisioning resync and firmware upgrade operations. When devices are deployed remotely, HTTPS is recommended, but HTTP and TFTP can also be used. This then requires provisioning file encryption to add security, as it offers greater reliability, given NAT and router protection mechanisms. TFTP is useful for the in-house preprovisioning of a large number of unprovisioned devices.

The phone is able to obtain a TFTP server IP address directly from the DHCP server through DHCP option 66. If a Profile_Rule is configured with the filepath of that TFTP server, the device downloads its profile from the TFTP server. The download occurs when the device is connected to a LAN and powered up.

The Profile_Rule provided with the factory default configuration is *&PN*.cfg, where *&PN* represents the phone model name.

For example, for a CP-6841-3PCC, the filename is CP-6841-3PCC.cfg.

For a device with the factory default profile, upon powering up, the device resyncs to this file on the local TFTP server that DHCP option 66 specifies. The filepath is relative to the TFTP server virtual root directory.

Related Topics

In-House Device Preprovisioning, on page 3

Remote Endpoint Control and NAT

The phone is compatible with network address translation (NAT) to access the Internet through a router. For enhanced security, the router might attempt to block unauthorized incoming packets by implementing symmetric NAT, a packet-filtering strategy that severely restricts the packets that are allowed to enter the protected network from the Internet. For this reason, remote provisioning by using TFTP is not recommended.

VoIP can coexist with NAT only when some form of NAT traversal is provided. Configure Simple Traversal of UDP through NAT (STUN). This option requires that the user have:

- A dynamic external (public) IP address from your service
- A computer that is running STUN server software
- An edge device with an asymmetric NAT mechanism

HTTP Provisioning

The phone behaves like a browser that requests web pages from a remote Internet site. This provides a reliable means of reaching the provisioning server, even when a customer router implements symmetric NAT or other

protection mechanisms. HTTP and HTTPS work more reliably than TFTP in remote deployments, especially when the deployed units are connected behind residential firewalls or NAT-enabled routers. HTTP and HTTPs are used interchangeably in the following request type descriptions.

Basic HTTP-based provisioning relies on the HTTP GET method to retrieve configuration profiles. Typically, a configuration file is created for each deployed phone, and these files are stored within an HTTP server directory. When the server receives the GET request, it simply returns the file that is specified in the GET request header.

Rather than a static profile, the configuration profile can be generated dynamically by querying a customer database and producing the profile on-the-fly.

When the phone requests a resynch, it can use the HTTP POST method to request the resync configuration data. The device can be configured to convey certain status and identification information to the server within the body of the HTTP POST request. The server uses this information to generate a desired response configuration profile, or to store the status information for later analysis and tracking.

As part of both GET and POST requests, the phone automatically includes basic identifying information in the User-Agent field of the request header. This information conveys the manufacturer, product name, current firmware version, and product serial number of the device.

The following example is the User-Agent request field from a CP-6841-3PCC:

```
User-Agent: Cisco-CP-6841-3PCC/11.0 (00562b043615)
```

When the phone is configured to resync to a configuration profile by using HTTP, it is recommended that HTTPS be used or the profile be encrypted to protect confidential information. Encrypted profiles that the phone downloads by using HTTP avoid the danger of exposing confidential information that is contained in the configuration profile. This resync mode produces a lower computational load on the provisioning server when compared to using HTTPS.

The phone can decrypt profiles encrypted with one of these encryption methods:

- AES-256-CBC encryption
- RFC-8188 based encryption with AES-128-GCM ciphering

Note The phones support HTTP Version 1.0, HTTP Version 1.1, and Chunk Encoding when HTTP Version 1.1 is the negotiated transport protocol.

HTTP Status Code Handling on Resync and Upgrade

The phone supports HTTP response for remote provisioning (Resync). Current phone behavior is categorized in three ways:

- A—Success, where the “Resync Periodic” and “Resync Random Delay” values determine subsequent requests.
- B—Failure when File Not Found or corrupt profile. The “Resync Error Retry Delay” value determines subsequent requests.
- C—Other failure when a bad URL or IP address causes a connection error. The “Resync Error Retry Delay” value determines subsequent requests.

Table 1: Phone Behavior for HTTP Responses

HTTP Status Code	Description	Phone Behavior
301 Moved Permanently	This and future requests should be directed to a new location.	Retry request immediately with new location.
302 Found	Known as Temporarily Moved.	Retry request immediately with new location.
3xx	Other 3xx responses not processed.	C
400 Bad Request	The request cannot be fulfilled due to bad syntax.	C
401 Unauthorized	Basic or digest access authentication challenge.	Immediately retry request with authentication credentials. Maximum 2 retries. Upon failure, the phone behavior is C.
403 Forbidden	Server refuses to respond.	C
404 Not Found	Requested resource not found. Subsequent requests by client are permissible.	B
407 Proxy Authentication Required	Basic or digest access authentication challenge.	Immediately retry request with authentication credentials. Maximum two retries. Upon failure, the phone behavior is C.
4xx	Other client error status codes are not processed.	C
500 Internal Server Error	Generic error message.	Phone behavior is C.
501 Not Implemented	The server does not recognize the request method, or it lacks the ability to fulfill the request.	Phone behavior is C.
502 Bad Gateway	The server is acting as a gateway or proxy and receives an invalid response from the upstream server.	Phone behavior is C.
503 Service Unavailable	The server is currently unavailable (overloaded or down for maintenance). This is a temporary state.	Phone behavior is C.
504 Gateway Timeout	The server behaves as a gateway or proxy and does not receive timely response from the upstream server.	C
5xx	Other server error	C

HTTPS Provisioning

The phone supports HTTPS for provisioning for increased security in managing remotely deployed units. Each phone carries a unique SLL Client Certificate (and associated private key), in addition to a Sipura CA server root certificate. The latter allows the phone to recognize authorized provisioning servers, and reject non-authorized servers. On the other hand, the client certificate allows the provisioning server to identify the individual device that issues the request.

For a service provider to manage deployment by using HTTPS, a server certificate must be generated for each provisioning server to which a phone resyncs by using HTTPS. The server certificate must be signed by the Cisco Server CA Root Key, whose certificate is carried by all deployed units. To obtain a signed server certificate, the service provider must forward a certificate signing request to Cisco, which signs and returns the server certificate for installation on the provisioning server.

The provisioning server certificate must contain the Common Name (CN) field, and the FQDN of the host running the server in the subject. It might optionally contain information following the host FQDN, separated by a slash (/) character. The following examples are of CN entries that are accepted as valid by the phone:

```
CN=sprov.callme.com
CN=pv.telco.net/mailto:admin@telco.net
CN=prof.voice.com/info@voice.com
```

In addition to verifying the server certificate, the phone tests the server IP address against a DNS lookup of the server name that is specified in the server certificate.

Get a Signed Server Certificate

The OpenSSL utility can generate a certificate signing request. The following example shows the **openssl** command that produces a 1024-bit RSA public/private key pair and a certificate signing request:

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

This command generates the server private key in **privkey.pem** and a corresponding certificate signing request in **provserver.csr**. The service provider keeps the **privkey.pem** secret and submits **provserver.csr** to Cisco for signing. Upon receiving the **provserver.csr** file, Cisco generates **provserver.crt**, the signed server certificate.

Procedure

Step 1 Navigate to https://software.cisco.com/software/cda/home and log in with your CCO credentials.

Note When a phone connects to a network for the first time or after a factory reset, and there are no DHCP options set up, it contacts a device activation server for zero touch provisioning. New phones use “activate.cisco.com” instead of “webapps.cisco.com” for provisioning. Phones with firmware release earlier than 11.2(1) continues to use “webapps.cisco.com”. We recommend that you allow both the domain names through your firewall.

Step 2 Select **Certificate Management**.

On the **Sign CSR** tab, the CSR of the previous step is uploaded for signing.

Step 3 From the **Select Product** drop-down list box, select **SPA1xx firmware 1.3.3 and newer/SPA232D firmware 1.3.3 and newer/SPA5xx firmware 7.5.6 and newer/CP-78xx-3PCC/CP-88xx-3PCC**.

Note This product includes the Cisco IP Phone 6800 Series Multiplatform Phones.

Step 4 In the **CSR File** field, click **Browse** and select the CSR for signing.

Step 5 Select the encryption method:

- MD5
- SHA1
- SHA256

Cisco recommends that you select SHA256 encryption.

Step 6 From the **Sign in Duration** drop-down list box, select the applicable duration (for example, 1 year).

Step 7 Click **Sign Certificate Request**.

Step 8 Select one of the following options to receive the signed certificate:

- **Enter Recipient's Email Address**—If you wish to receive the certificate via email, enter your email address in this field.
- **Download**—If you wish to download the signed certificate, select this option.

Step 9 Click **Submit**.

The signed server certificate is either emailed to the email address previously provided or downloaded.

Multiplatform Phone CA Client Root Certificate

Cisco also provides a Multiplatform Phone Client Root Certificate to the service provider. This root certificate certifies the authenticity of the client certificate that each phone carries. The Multiplatform Phones also support third-party signed certificates such as those provided by Verisign, Cybertrust, and so on.

The unique client certificate that each device offers during an HTTPS session carries identifying information that is embedded in its subject field. This information can be made available by the HTTPS server to a CGI script invoked to handle secure requests. In particular, the certificate subject indicates the unit product name (OU element), MAC address (S element), and serial number (L element).

The following example from the Cisco IP Phone 6841 Multiplatform Phones client certificate subject field shows these elements:

```
OU=CP-6841-3PCC, L=88012BA01234, S=000e08abcdef
```

To determine if a phone carries an individualized certificate, use the $CCERT provisioning macro variable. The variable value expands to either Installed or Not Installed, according to the presence or absence of a unique client certificate. In the case of a generic certificate, it is possible to obtain the serial number of the unit from the HTTP request header in the User-Agent field.

HTTPS servers can be configured to request SSL certificates from connecting clients. If enabled, the server can use the Multiplatform Phone Client Root Certificate that Cisco supplies to verify the client certificate. The server can then provide the certificate information to a CGI for further processing.

The location for certificate storage may vary. For example, in an Apache installation, the file paths for storage of the provisioning server-signed certificate, its associated private key, and the Multiplatform Phone CA client root certificate are as follows:

```
# Server Certificate:
SSLCertificateFile /etc/httpd/conf/provserver.crt

# Server Private Key:
SSLCertificateKeyFile /etc/httpd/conf/provserver.key

# Certificate Authority (CA):
SSLCACertificateFile /etc/httpd/conf/spacroot.crt
```

For specific information, refer to the documentation for an HTTPS server.

The Cisco Client Certificate Root Authority signs each unique certificate. The corresponding root certificate is made available to service providers for client authentication purposes.

Redundant Provisioning Servers

The provisioning server can be specified as an IP address or as a Fully Qualified Domain Name (FQDN). The use of an FQDN facilitates the deployment of redundant provisioning servers. When the provisioning server is identified through an FQDN, the phone attempts to resolve the FQDN to an IP address through DNS. Only DNS A-records are supported for provisioning; DNS SRV address resolution is not available for provisioning. The phone continues to process A-records until a server responds. If no server that is associated with the A-records responds, the phone logs an error to the syslog server.

Syslog Server

If a syslog server is configured on the phone through use of the <Syslog Server> 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).

The logged messages are configured in the following parameters and macro expanded into the actual syslog messages:

- Log_Request_Msg
- Log_Success_Msg
- Log_Failure_Msg