プロビジョニングの例

この章では、Cisco IP Phone とプロビジョニング サーバ間で設定プロファイルを転送する手順を、例を挙げて説明します。

設定プロファイルの作成については、第 2 章「プロビジョニング スクリプト」を参照してください。

基本的な再同期

ここでは、Cisco IP Phone の基本的な再同期機能について説明します。

TFTP の再同期

Cisco IP Phone は、設定プロファイルの取得に複数のネットワーク プロトコルをサポートしています。最も基本的なプロファイルの転送プロトコルは TFTP（RFC1350）です。TFTP は、プライベート LAN ネットワーク内のネットワーク デバイスのプロビジョニングに広く使用されています。インターネット経由のリモート エンドポイントの導入に使用するのはお勧めしませんが、TFTP は、小規模な組織内での導入、社内でのプロビジョニング、および開発とテストで使用するには便利です。社内でのプロビジョニングについては、「社内デバイスのプロビジョニング」セクション（3-2 ページ）を参照してください。この演習では、TFTP サーバからファイルをダウンロードした後に、プロファイルを変更します。

演習

ステップ 1 LAN 環境で、ハブ、スイッチ、または小規模なルータに PC および Cisco IP Phone を接続します。

ステップ 2 PC で、TFTP サーバをインストールして有効化します。

ステップ 3 テキスト エディタを使用して、例に示すように、GPP_A の値に 12345678 を設定した設定プロファイルを作成します。

```
<device> <flat-profile>
  <GPP_A> 12345678
  </GPP_A>
</flat-profile> </device>
```

ステップ 4 プロファイルに basic.txt という名前を付けて、TFTP サーバのルート ディレクトリに保存します。

次の方法で、TFTP サーバが正しく設定されていることを確認できます：Cisco IP Phone 以外の TFTP クライアントを使用して basic.txt ファイルをリクエストします。できれば、プロビジョニング サーバとは別のホストで実行されている TFTP クライアントを使用します。

ステップ 5　PC の Web ブラウザで admin/advanced のページを開きます。たとえば、電話機の IP アドレスが 192.168.1.100 の場合は次の URL を使用します。

```
http://192.168.1.100/admin/advanced
```

ステップ 6　[プロビジョニング（Provisioning）] タブを選択し、汎用パラメータの GPP_A から GPP_P の値を確認します。これらは空である必要があります。

ステップ 7　Web ブラウザ ウィンドウで再同期 URL を開き、テスト用 Cisco IP Phone を設定プロファイル basic.txt に再同期します。

TFTP サーバの IP アドレスが 192.168.1.200 の場合、コマンドはこの例のようになります。

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

このコマンドを Cisco IP Phone が受信すると、アドレス 192.168.1.100 のデバイスは、IP アドレスが 192.168.1.200 の TFTP サーバにファイル basic.txt をリクエストします。電話機はダウンロードしたファイルを解析し、GPP_A パラメータを値 12345678 に更新します。

ステップ 8　パラメータが正しく更新されていることを次の手順で確認します。PC の Web ブラウザの admin/advanced のページを更新し、そのページの [プロビジョニング（Provisioning）] タブを選択します。

GPP_A パラメータが値 12345678 になっている必要があります。

syslog を使用したロギング

デバイスがプロビジョニング サーバとの再同期を開始する際、および再同期が完了または失敗した後、Cisco IP Phone は syslog メッセージを専用の syslog サーバに送信します。このサーバは、Web サーバ管理（admin/advanced、[システム（System）] タブ、Syslog_Server パラメータ）で確認できます。syslog サーバの IP アドレスをデバイスに設定し、これ以降の演習中に生成されるメッセージを監視します。

演習

ステップ 1　ローカル PC に syslog サーバをインストールして有効化します。

ステップ 2　次のように、PC の IP アドレスをプロファイルの Syslog_Server パラメータに設定して、変更を送信します。

```
<Syslog_Server ua="na">192.168.1.210</Syslog_Server>
```

ステップ 3　[システム（System）] タブをクリックし、Syslog_Server パラメータにローカル syslog サーバの値を入力します。

ステップ 4　「TFTP の再同期」の演習で説明されているようにして、再同期操作を繰り返します。

デバイスは再同期中に 2 件の syslog メッセージを生成します。最初のメッセージは、リクエストが進行中であることを示します。2 番目のメッセージは、再同期が成功または失敗したことを示します。

ステップ 5　syslog サーバが以下のようなメッセージを受信したことを確認します。

```
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
```

詳細なメッセージを利用できるようにするには、次のように、(Syslog_Server パラメータの代わりに)Debug_Server パラメータに syslog サーバの IP アドレスを設定し、Debug_Level パラメータに 0 ～ 3 の範囲(3 が最も詳細)の値を設定します。

```
<Debug_Server ua="na">192.168.1.210</Debug_Server>
<Debug_Level ua="na">3</Debug_Level>
```

これらのメッセージの内容は、次のパラメータを使用して設定できます。

- Log_Request_Msg
- Log_Success_Msg
- Log_Failure_Msg

これらのパラメータのいずれかが削除されると、対応する syslog メッセージは生成されなくなります。

デバイスの自動再同期

(エンドポイントに明示的な再同期リクエストを送信するのではなく)デバイスを定期的にプロビジョニング サーバに再同期させて、サーバに対して行われたプロファイルの変更を確実にエンドポイント デバイスに伝達することができます。

Cisco IP Phone にサーバへの定期的な再同期を行わせるには、Profile_Rule パラメータを使用して設定プロファイルの URL を定義し、さらに Resync_Periodic パラメータを使用して再同期間隔を定義します。

演習

ステップ 1　Web ブラウザを使用して、admin/advanced のページの [プロビジョニング(Provisioning)] タブを開きます。

ステップ 2　Profile_Rule パラメータを定義します。次の例では、TFTP サーバの IP アドレスを 192.168.1.200 と仮定しています。

```
<Profile_Rule ua="na">tftp://192.168.1.200/basic.txt</Profile_Rule>
```

ステップ 3　Resync_Periodic パラメータに、テスト用として **30** 秒などの小さい値を入力します。

```
<Resync_Periodic ua="na">30</Resync_Periodic>
```

ステップ 4　[すべての変更を送信(Submit all Changes)] をクリックします。

新しいパラメータ設定により、Cisco IP Phone は URL で指定された設定ファイルに対して 1 分間に 2 回再同期を行います。

ステップ 5　syslog トレースの結果メッセージを(「syslog を使用したロギング」セクションで説明されているようにして)確認します。

ステップ 6　Resync_On_Reset パラメータが次のように [はい(yes)] に設定されていることを確認します。

```
<Resync_On_Reset ua="na">Yes</Resync_On_Reset>
```

ステップ 7　電源を再投入して、Cisco IP Phone を強制的にプロビジョニング サーバと再同期させます。

サーバが無応答など、何らかの理由で再同期操作が失敗すると、ユニットは(Resync_Error_Retry_Delay に設定した秒数だけ)待機した後、もう一度再同期を試みます。Resync_Error_Retry_Delay が 0 の場合、Cisco IP Phone は再同期が失敗しても、再同期を試行しません。

ステップ 8　（任意）Resync_Error_Retry_Delay の値に **30** などの小さい数値を設定します。

```
<Resync_Error_Retry_Delay ua="na">30</Resync_Error_Retry_Delay>
```

ステップ 9　TFTP サーバを無効化して、syslog 出力の結果を確認します。

固有のプロファイル、マクロ展開、および HTTP

各 Cisco IP Phone の User_ID、Display_Name といったパラメータに個別の値を設定する必要があるような導入では、サービス プロバイダーが、導入されるデバイスそれぞれに固有のプロファイルを作成して、プロビジョニング サーバでそれらのプロファイルをホストすることができます。事前に定義されたプロファイルの命名規則に従って、それぞれの Cisco IP Phone が自身のプロファイルに次々と再同期するよう設定される必要があります。

プロファイル URL の構文には、組み込み変数のマクロ展開を使用して、各 Cisco IP Phone に固有の識別情報（MAC アドレス、シリアル番号など）を含めることができます。マクロ展開を使用すれば、各プロファイルの複数箇所で前記の値を指定する必要がなくなります。

プロファイルのルールは、Cisco IP Phone に適用される前にマクロ展開の適用を受けます。マクロ展開は値の数値を制御します。たとえば、

- $MA は、ユニットの 12 桁の MAC アドレスに展開されます（小文字の 16 進数を使用）。たとえば、000e08abcdef などのようになります。
- $SN は、ユニットのシリアル番号に展開されます。たとえば、88012BA01234 などのようになります。

GPP_A から GPP_P までのすべての汎用パラメータなど、他の値も同様にマクロ展開されます。この手順の例が「TFTP の再同期」セクションで説明されています。マクロ展開は URL のファイル名だけでなく、プロファイル ルール パラメータの任意の部分に適用できます。これらのパラメータは $A ～ $P として参照されます。マクロ展開で使用できるすべての変数の一覧については、「マクロ展開変数」セクション（5-5 ページ）を参照してください。

この演習では、Cisco IP Phone に固有のプロファイルを TFTP サーバ上でプロビジョニングします。演習では例として Cisco Phone 7841 を使用しますが、この内容はすべての Cisco IP Phone 7800/8800 シリーズ モデルに共通です。

演習：TFTP サーバの特定の IP 電話プロファイルのプロビジョニング

ステップ 1　製品ラベルで電話機の MAC アドレスを確認します（MAC アドレスは、000e08aabbcc などの、数字と小文字の 16 進数を使用する番号です）。

ステップ 2　設定ファイル `basic.txt`（「TFTP の再同期」の演習で説明されています）を、**`CP-`***`x8xx`***`-3PCC_`***`macaddress`***`.cfg`** という名前の新しいファイルにコピーします（*`x8xx`* をモデル番号に、*`macaddress`* を電話機の MAC アドレスにそれぞれ置き換えます）。次に例を示します。

```
CP-7841-3PCC_000e08abcdef.cfg
```

ステップ 3　TFTP サーバの仮想ルート ディレクトリに新しいファイルを移動します。

ステップ 4　admin/advanced のページの [プロビジョニング（Provisioning）] タブを開きます。

ステップ 5　Profile_Rule パラメータに `tftp://192.168.1.200/CP-7841-3PCC$MA.cfg` と入力します。

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

ステップ 6　[すべての変更を送信（Submit All Changes）] をクリックします。これにより、リブートと再同期がただちに行われます。

次の再同期時に、$MA マクロの式が MAC アドレスに展開されて、Cisco IP Phone は新しいファイルを取得します。

HTTP GET 再同期

HTTP は TCP 接続を確立し、TFTP は信頼性に劣る UDP を使用するため、HTTP は TFTP より信頼性の高い再同期方式を提供します。また、HTTP サーバは、TFTP サーバと比べてより強化されたフィルタリング機能とロギング機能を備えています。

クライアント側の Cisco IP Phone が HTTP を使用した再同期を使用できるようにするために、サーバで特別な構成設定を行う必要はありません。GET メソッドで HTTP を使用するための Profile_Rule パラメータの構文は、TFTP の場合の構文と同様です。標準的な Web ブラウザで HTTP サーバからプロファイルを取得できるならば、Cisco IP Phone も同様にできます。

演習

ステップ 1　ローカル PC またはその他のアクセス可能なホストに HTTP サーバをインストールします（オープン ソースの Apache サーバがインターネットからダウンロードできます）。

ステップ 2　設定プロファイル `basic.txt`（「TFTP の再同期」の演習で説明されています）をそのインストールしたサーバの仮想ルート ディレクトリにコピーします。

ステップ 3　サーバが適切にインストールされ、basic.txt にアクセスできることを確認するために、Web ブラウザを使用してプロファイルにアクセスします。

ステップ 4　プロファイルが定期的にダウンロードできるようにするために、テスト用 Cisco IP Phone の Profile_Rule を変更し、TFTP サーバの代わりに HTTP サーバを指すようにします。

たとえば、HTTP サーバが 192.168.1.300 と仮定した場合、次の値を入力します。

```
<Profile_Rule ua="na">
http://192.168.1.200/basic.txt
</Profile_Rule>
```

ステップ 5　[すべての変更を送信（Submit All Changes）] をクリックします。これにより、リブートと再同期がただちに行われます。

ステップ 6　Cisco IP Phone が送信した syslog メッセージを確認します。定期的な再同期で、HTTP サーバからプロファイルが取得されている必要があります。

ステップ 7　HTTP サーバのログで、テスト用 Cisco IP Phone を特定する情報がユーザ エージェントのログにどのように表示されるかを確認します。

この情報には、製造者、製品名、現在のファームウェア バージョン、およびシリアル番号が含まれている必要があります。

Cisco XML を介したプロビジョニング

ここでは x8xx として表される Cisco IP Phone 7800/8800 シリーズはそれぞれ、Cisco XLM の機能を介して以下のようにしてプロビジョニングされます。

CP-x8xx-3PCC では Cisco XML の機能が拡張され、XML オブジェクトを介したプロビジョニングがサポートされています。

```
<CP-x8xx-3PCCExecute>
        <ExecuteItem URL=Resync:[profile-rule]/>
</CP-x8xx-3PCCExecute>
```

XML オブジェクトを受信した後、CP-x8xx-3PCC はプロビジョニング ファイルを [profile-rule] からダウンロードします。このルールでは、XML サービス アプリケーションの開発を容易にするマクロが使用されています。

マクロ展開を使用した URL の解決

複数のプロファイルがあるサーバ上のサブディレクトリでは、多数の導入済みデバイスを管理するための便利な方法が提供されます。プロファイル URL には以下を含めることができます。

- プロビジョニング サーバ名または明示的な IP アドレス。プロファイルで、プロビジョニング サーバが名前で指定されている場合、Cisco IP Phone は DNS ルックアップを実行して名前を解決します。
- サーバ名の後に続く標準の構文 *:port* を使用して、URL で指定される非標準サーバ ポート。
- 標準 URL 表記を使用して指定され、マクロ展開により管理されるプロファイルが保存されている、サーバ仮想ルート ディレクトリのサブディレクトリ。

たとえば、次の Profile_Rule は、ポート 6900 の接続をリスニングするホスト prov.telco.com で実行されている TFTP サーバから、サーバの /cisco/config サブディレクトリにあるプロファイル CP-7841-3PCC.cfg をリクエストします。

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

各 Cisco IP Phone のプロファイルは汎用パラメータにより特定できます。これらはマクロ展開を使用して共通プロファイル ルール内で値が参照されます。

たとえば、GPP_B に `Dj6Lmp23Q` が定義されていると仮定します。

Profile_Rule は次の値になります。

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

デバイスの再同期およびマクロの展開時に、MAC アドレスが 000e08012345 の Cisco IP Phone は、デバイスの MAC アドレスを含む名前が記載されたプロファイルを、次の URL でリクエストします。

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

安全な HTTPS 再同期

安全な通信プロセスを使用して再同期を行うために、Cisco IP Phone では以下の方式が使用できます。

- 基本的な HTTPS 再同期
- クライアント証明書認証を使用した HTTPS
- HTTPS クライアントのフィルタリングとダイナミック コンテンツ

関連項目

- 基本的な HTTPS 再同期（4-7 ページ）
- クライアント証明書認証を使用した HTTPS（4-9 ページ）
- HTTPS クライアントのフィルタリングとダイナミック コンテンツ（4-9 ページ）

基本的な HTTPS 再同期

HTTPS では、リモート プロビジョニングの HTTP に SSL が追加されるため、以下が可能になります。

- Cisco IP Phone はプロビジョニング サーバを認証することができます。
- プロビジョニング サーバは Cisco IP Phone を認証することができます。
- Cisco IP Phone とプロビジョニング サーバ間で交換される情報の機密性が保証されます。

SSL は、Cisco IP Phone とプロビジョニング サーバに事前にインストールされている公開キー/秘密キーのペアを使用して、Cisco IP Phone とサーバ間の各接続に対して、秘密の（対称）キーを生成して交換します。

クライアント側の Cisco IP Phone が HTTPS を使用した再同期を使用できるようにするために、サーバで特別な構成設定を行う必要はありません。GET メソッドで HTTPS を使用するための Profile_Rule パラメータの構文は、HTTP または TFTP の場合の構文と同様です。標準的な Web ブラウザで HTTPS サーバからプロファイルを取得できるならば、Cisco IP Phone も同様にできます。

HTTPS サーバのインストールに加えて、Cisco が署名した SSL サーバ証明書がプロビジョニング サーバにインストールされている必要があります。Cisco が署名したサーバ証明書をサーバが提供しない場合、デバイスは HTTPS を使用するサーバに再同期できません。音声製品向けの署名付き SSL 証明書を作成する手順については、https://supportforums.cisco.com/docs/DOC-9852 を参照してください。

演習：基本的な HTTPS 再同期

ステップ 1　通常のホスト名変換により、ネットワーク DNS サーバによって IP アドレスが特定されるホストに、HTTPS サーバをインストールします。

オープン ソース Apache サーバが、オープン ソース mod_ssl パッケージと共にインストールされている場合には、HTTPS サーバとして動作するように設定できます。

ステップ 2　そのサーバのサーバ証明書署名要求を生成します。この手順では、オープン ソースの OpenSSL パッケージまたは同等のソフトウェアのインストールが必要な場合があります。OpenSSL を使用する場合、基本 CSR ファイルを生成するコマンドは次のとおりです。

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

このコマンドにより公開キー/秘密キーのペアが生成され、privkey.pem ファイルに保存されます。

ステップ 3　署名のために Cisco に CSR ファイル（provserver.csr）を提出します（詳細については、https://supportforums.cisco.com/docs/DOC-9852 を参照してください）。署名付きサーバ証明書が、Sipura CA クライアント ルート証明書 spacroot.cert と共に返送されます。

ステップ 4　署名付きサーバ証明書、秘密キーのペアのファイル、およびクライアント ルート証明書をサーバの適切な場所に保存します。

Linux に Apache をインストールしている場合、これらは通常以下の場所にあります。

```
# 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
```

ステップ 5　サーバを再起動します。

ステップ 6　設定ファイル `basic.txt`（「TFTP の再同期」の演習で説明されています）を HTTPS サーバの仮想ルート ディレクトリにコピーします。

ステップ 7　ローカル PC で標準的なブラウザを使用して、HTTPS サーバから basic.txt をダウンロードし、サーバの動作が適切であることを確認します。

ステップ 8　サーバが提供したサーバ証明書を確認します。

Cisco をルート CA として受け入れるようにブラウザが事前に設定されていない場合、ブラウザは証明書が有効であることほとんど認識しません。ただし、Cisco IP Phone では証明書がこの方法で署名されることを期待しています。

テスト用デバイスの Profile_Rule を変更して、HTTPS サーバへの参照が含まれるようにします。たとえば次のように設定します。

```
<Profile_Rule ua="na">
https://my.server.com/basic.txt
</Profile_Rule>
```

この例では、HTTPS サーバの名前が `my.server.com` であると仮定しています。

ステップ 9　[すべての変更を送信（Submit All Changes）] をクリックします。

ステップ 10　Cisco IP Phone が送信した syslog トレースを確認します。

再同期で HTTPS サーバからプロファイルが取得されたことが、syslog メッセージに示されている必要があります。

ステップ 11　（任意）Cisco IP Phone サブネットでイーサネット プロトコル アナライザを使用して、パケットが暗号化されていることを確認します。

この演習では、クライアント証明書の検証は有効になっていません。Cisco IP Phone とサーバ間の接続は暗号化されます。ただし、ファイル名とディレクトリの場所が分かれば、あらゆるクライアントがサーバに接続し、ファイルをリクエストできるため、転送は安全ではありません。安全に再同期するためには、サーバは、「クライアント証明書認証を使用した HTTPS」セクションで説明されている演習の手順に従ってクライアントを認証する必要もあります。

クライアント証明書認証を使用した HTTPS

工場出荷時のデフォルト設定では、サーバはクライアントから SSL クライアント証明書をリクエストしません。あらゆるクライアントがサーバに接続し、プロファイルをリクエストできるため、プロファイルの転送は安全ではありません。設定を編集して、クライアント認証を有効にすることができます。サーバは、接続リクエストを受け入れる前に Cisco IP Phone を認証するために、クライアント証明書が必要です。

この要件があるため、適切な認証情報がないブラウザを使用して、再同期操作を個別にテストすることはできません。テスト用 Cisco IP Phone とサーバ間の HTTPS 接続での SSL キー交換は、ssldump ユーティリティで確認できます。このユーティリティのトレースにより、クライアントとサーバ間の相互通信が表示されます。

クライアント証明書認証を使用した HTTPS の演習

ステップ 1　HTTPS サーバでクライアント証明書認証を有効化します。

ステップ 2　Apache（v.2）では、サーバ設定ファイルを次のように設定します。

```
SSLVerifyClient  require
```

また、spacroot.cert が、「基本的な HTTPS 再同期」の演習で説明されている手順で保存されていることを確認します。

ステップ 3　HTTPS サーバを再起動した後、Cisco IP Phone からの syslog トレースを確認します。

これで、サーバに再同期するたびに対称認証が実行されるようになりました。これにより、プロファイルが転送される前にサーバ証明書とクライアント証明書の両方が検証されます。

ステップ 4　ssldump を使用して、Cisco IP Phone と HTTPS サーバ間の再同期接続を採取します。

クライアント証明書の検証がサーバで適切に有効化されている場合には、ssldump トレースには、プロファイルを含む暗号化されたパケットの前に、証明書が相互に交換されたことが示されます（最初にサーバからクライアントへ、次にクライアントからサーバへ）。

クライアントの認証が有効化されていると、有効なクライアント証明書と一致する MAC アドレスの Cisco IP Phone のみが、プロビジョニング サーバのプロファイルをリクエストできます。サーバは、通常のブラウザやその他の不正なデバイスからのリクエストを拒否します。

HTTPS クライアントのフィルタリングとダイナミック コンテンツ

クライアント証明書が必要となるように HTTPS サーバが設定されている場合、再同期している Cisco IP Phone が証明書の情報により識別され、正しい設定情報が渡されます。

HTTPS サーバにより、証明書の情報が、再同期リクエストの一環として起動される CGI スクリプト（またはコンパイルされた CGI プログラム）で使用可能になります。説明の都合から、この演習ではオープン ソースの Perl スクリプト言語を使用し、HTTPS サーバとして Apache（v.2）が使用されていると仮定します。

演習

ステップ 1　HTTPS サーバを実行しているホストに Perl をインストールします。

ステップ 2　以下の Perl リフレクタ スクリプトを生成します。

```
#!/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>";
```

ステップ 3　このファイルに reflect.pl という名前を付けて、HTTPS サーバの CGI スクリプトのディレクトリに、実行権限（Linux では chmod 755）で保存します。

ステップ 4　サーバの CGI スクリプトにアクセスできるかどうかを確認します（/cgi-bin/...で）。

ステップ 5　テスト用デバイスで Profile_Rule を変更し、次の例のようにしてリフレクタ スクリプトに再同期します。

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

ステップ 6　[すべての変更を送信（Submit All Changes）] をクリックします。

ステップ 7　syslog トレースを参照し、再同期が成功したことを確認します。

ステップ 8　admin/advanced のページの [プロビジョニング（Provisioning）] タブを開きます。

ステップ 9　GPP_D パラメータにスクリプトが採取した情報が含まれていることを確認します。

テスト用デバイスが製造者からの固有の証明書を保持している場合、この情報には、製品名、MAC アドレス、およびシリアル番号が含まれています。ユニットがファームウェア リリース 2.0 より前に製造されている場合、この情報には一般的な文字列が含まれています。

同様なスクリプトにより、再同期デバイスに関する情報を識別し、適切な設定パラメータ値をデバイスに提供できます。

HTTPS 証明書

Cisco IP Phone は、デバイスからプロビジョニング サーバへの HTTPS リクエストに基づく信頼性の高い安全なプロビジョニング手段を提供します。サーバ証明書とクライアント証明書の両方が、Cisco IP Phone からサーバ、およびサーバから Cisco IP Phone の認証で使用されます。

電話機で HTTPS を使用するには、証明書署名要求（CSR）を生成し、Cisco に提出する必要があります。Cisco IP Phone は、プロビジョニング サーバへのインストール用の証明書を生成します。Cisco IP Phone は、プロビジョニング サーバとの HTTPS 接続を確立しようとする際に証明書を受け入れます。

HTTPS 方式

HTTPS によりクライアントとサーバ間の通信が暗号化されるため、他のネットワーク デバイスからメッセージの内容が保護されます。クライアントとサーバ間の通信本文の暗号化方式は、対称キー暗号化に基づいています。対称キー暗号化では、公開キー/秘密キーの暗号化によって保護された安全なチャネル上で、1 つの秘密キーをクライアントとサーバで共有します。

秘密キーで暗号化されたメッセージは、同じキーを使用しなければ復号化できません。HTTPS は、対称暗号化アルゴリズムを広くサポートしています。Cisco IP Phone では、128 ビット RC4 に加えて、米国の暗号化標準（AES）を使用した 256 ビットまでの対称暗号化を実装しています。

HTTPS はまた、安全なトランザクションで実行されるサーバとクライアントの認証も提供します。これにより、プロビジョニング サーバと個々のクライアントは、ネットワーク上の他のデバイスによってスプーフィングできなくなります。この機能は、リモート エンドポイント プロビジョニングでは必須です。

サーバとクライアント間の認証は、公開キーが含まれている証明書を使用し、公開キー/秘密キー暗号化によって実行されます。公開キーで暗号化されたテキストは、対応する秘密キーでのみ復号化できます（逆も同様です）。Cisco IP Phone は、Rivest-Shamir-Adleman（RSA）アルゴリズムを公開キー/秘密キーの暗号化でサポートしています。

SSL サーバ証明書

安全なプロビジョニング サーバには個別に、Cisco が直接署名したセキュア ソケット レイヤ（SSL）サーバ証明書が発行されています。Cisco IP Phone で動作するファームウェアは、Cisco の証明書のみを有効として認識します。クライアントは、HTTPS を使用してサーバに接続する際、Cisco によって署名されていないサーバ証明書を拒否します。

この方式により、Cisco IP Phone への不正アクセスや、プロビジョニング サーバをスプーフィングする試みからサービス プロバイダーを保護します。このような保護がない場合、攻撃者は、Cisco IP Phone を再プロビジョニングして、設定情報を取得したり、別の VoIP サービスを使用したりする可能性があります。有効なサーバ証明書に対応する秘密キーを使用しないと、攻撃者は Cisco IP Phone との通信を確立できません。

サーバ証明書の取得

ステップ 1　証明書のプロセスについては、ユーザを担当する Cisco のサポート担当者に確認してください。特定のサポート担当者がいない場合は、電子メールで ciscosb-certadmin@cisco.com にリクエストを送信してください。

ステップ 2　CSR（証明書署名要求）で使用される秘密キーを生成します。このキーは秘密であり、Cisco のサポートに提供する必要はありません。オープン ソースの「openssl」を使用してキーを生成します。次に例を示します。

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

ステップ 3　ユーザの組織と場所を識別するフィールドを含む CSR を生成します。次に例を示します。

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

以下の情報が必要です。

- 件名フィールド：共通名（CN）を入力します。これは FQDN（完全修飾ドメイン名）構文である必要があります。SSL 認証のハンドシェイク中に、Cisco IP Phone は、受信した証明書がそれを提出した装置からのものであるかどうかを確認します。
- サーバ ホスト名：たとえば、provserv.domain.com など。

- 電子メールアドレス：必要な場合にカスタマー サポートがユーザに連絡を取れるようにするために、電子メールアドレスを入力します。この電子メールアドレスは、CSR に表示されます。

ステップ 4　Cisco のサポート担当者または次のアドレスに、CSR（zip ファイル形式）を電子メールで送信します ciscosb-certadmin@cisco.com. 証明書が Cisco によって署名されます。Cisco は、システムにインストールする証明書をユーザに送信します。

クライアント証明書

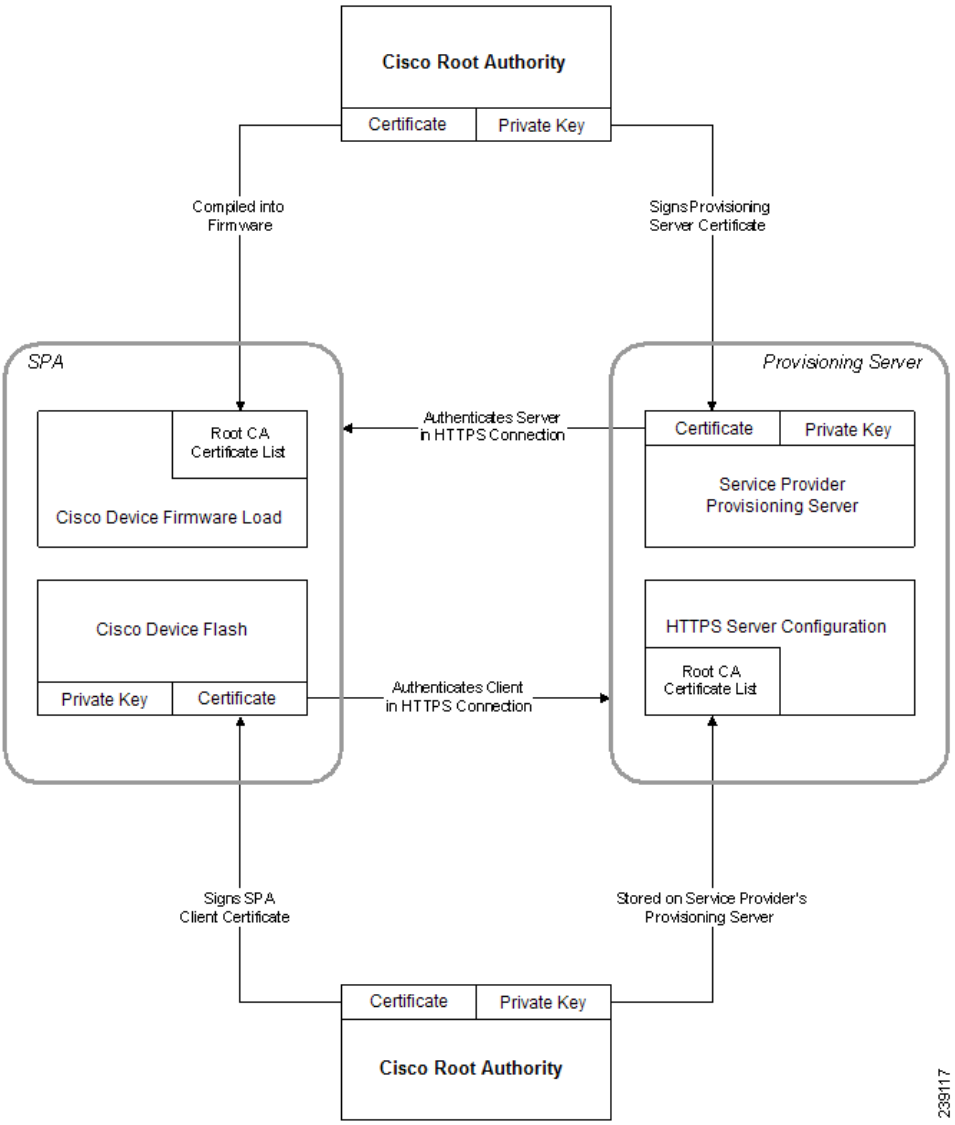
Cisco IP Phone への直接攻撃以外にも、攻撃者は、標準的な Web ブラウザや他の HTTPS クライアントを介してプロビジョニング サーバに接続し、プロビジョニング サーバから設定プロファイルを取得しようとする可能性があります。この種の攻撃を防ぐため、各 Cisco IP Phone は、Cisco によって署名され、個々のエンドポイントに関する識別情報を含む固有のクライアント証明書も保持しています。デバイスのクライアント証明書を認証できる認証局ルート証明書が、各サービス プロバイダーに提供されます。この認証パスにより、プロビジョニング サーバは不正な設定プロファイルのリクエストを拒否することができます。

証明書の構造

サーバ証明書とクライアント証明書の組み合わせにより、リモート Cisco IP Phone とそのプロビジョニング サーバ間のセキュア通信が保証されます。図 4-1 には、Cisco クライアント、プロビジョニング サーバ、および認証局間での、証明書、公開キー/秘密キーのペア、および署名するルート認証局の関係が図示されています。

図の上半分には、個別のプロビジョニング サーバ証明書への署名に使用されるプロビジョニング サーバ ルート認証局が示されています。対応するルート証明書がファームウェアに組み込まれ、これを使用して、Cisco IP Phone は正規のプロビジョニング サーバを認証することができます。

図 4-1　　認証局のフロー



カスタム認証局の設定

ネットワーク上のネットワーク デバイスおよびユーザの認証にデジタル証明書が使用できます。これは、ネットワーク ノード間の IPSec セッションのネゴシエートに使用できます。

サード パーティは認証局証明書を使用して、通信を試みている複数のノードを検証し、認証します。各ノードには公開キーと秘密キーがあります。公開キーでデータを暗号化します。秘密キーでデータを復号化します。ノードは同じソースから証明書を取得しているため、それぞれの同一性が保証されます。

デバイスは、サードパーティの認証局（CA）により提供されるデジタル証明書を使用して、IPSec 接続を認証することができます。

電話機は、ファームウェアに組み込まれて事前にロードされる、以下の一連のルート認証局をサポートしています。

- Cisco Small Business CA 証明書
- CyberTrust CA 証明書
- Verisign CA 証明書
- Sipura ルート CA 証明書
- Linksys ルート CA 証明書

ステップ 1　[管理者ログイン（Admin Login）] > [詳細（advanced）] > [情報（Info）] > [ステータスをダウンロード（Download Status）] の順にクリックします。

ステップ 2　[カスタム CA ステータス（Custom CA Status）] までスクロールし、以下のフィールドを確認します。

- [カスタム CA プロビジョニング ステータス（Custom CA Provisioning Status）]：プロビジョニングのステータスを示します。
  - 最後のプロビジョニングが mm/dd/yyyy HH:MM:SS に成功した
  - 最後のプロビジョニングが mm/dd/yyyy HH:MM:SS に失敗した
- [カスタム CA 情報（Custom CA Info）]：カスタム CA に関する情報を表示します。
  - [インストール済み（Installed）]：「CN 値」が表示されます。ここで、「CN 値」は最初の証明書の件名フィールドの CN パラメータの値です。
  - [未インストール（Not Installed）]：カスタム CA 証明書がインストールされていない場合に表示されます。

プロファイル管理

ここでは、ダウンロードの準備として設定プロファイルの構成を説明します。機能の説明のために、ローカル PC からの TFTP を再同期手段として使用しますが、HTTP または HTTPS も同様に使用できます。

プロファイルの gzip 圧縮を開く

プロファイルですべてのパラメータを個々に指定すると、XML 形式の設定プロファイルはかなり大きくなる可能性があります。プロビジョニング サーバの負荷を軽減するため、Cisco IP Phone では、gzip ユーティリティ（RFC 1951）がサポートするデフレート圧縮形式を使用した XML ファイルの圧縮がサポートされています。

（注）　圧縮され暗号化された XML プロファイルを Cisco IP Phone が認識できるようにするために、暗号化より先に圧縮が行われる必要があります。

カスタマイズされたバックエンド プロビジョニング サーバ ソリューションに統合する場合は、スタンドアロン gzip ユーティリティの代わりにオープン ソースの zlib 圧縮ライブラリを使用して、プロファイルの圧縮が実行できます。ただし、Cisco IP Phone はファイルに有効な gzip ヘッダーが含まれていることを期待しています。

演習

ステップ 1　ローカル PC に gzip をインストールします。

ステップ 2　コマンド ラインから gzip を起動して、設定プロファイル `basic.txt`（「TFTP の再同期」の演習で説明されています）を圧縮します。

```
gzip basic.txt
```

これにより、縮小ファイル `basic.txt.gz` が生成されます。

ステップ 3　TFTP サーバの仮想ルート ディレクトリにファイル `basic.txt.gz` を保存します。

ステップ 4　次の例に示すようにして、テスト用デバイスで Profile_Rule を変更し、元の XML ファイルの代わりに縮小ファイルに再同期するようにします。

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

ステップ 5　[すべての変更を送信（Submit All Changes）] をクリックします。

ステップ 6　Cisco IP Phone からの syslog トレースを確認します。

再同期時、Cisco IP Phone は新しいファイルをダウンロードし、これを使用して自身のパラメータを更新します。

関連項目

- オープン プロファイルの圧縮（2-6 ページ）

OpenSSL を使用したプロファイルの暗号化

圧縮または未圧縮のプロファイルが暗号化できます（ただし、ファイルは暗号化される前に圧縮されている必要があります）。暗号化は、Cisco IP Phone とプロビジョニング サーバ間の通信に TFTP または HTTP が使用される場合など、プロファイル情報の機密性が特に問題となる場面で有用です。

Cisco IP Phone は、256 ビット AES アルゴリズムを使用する対称キー暗号化をサポートしています。この暗号化は、オープン ソースの OpenSSL パッケージを使用して実行できます。

演習

ステップ 1　ローカル PC に OpenSSL をインストールします。この際、AES を有効にするために OpenSSL アプリケーションの再コンパイルが必要な場合があります。

ステップ 2　設定ファイル `basic.txt`（「TFTP の再同期」の演習で説明されています）を使用し、次のコマンドを実行して暗号化されたファイルを生成します。

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

XML プロファイルは圧縮と暗号化の両方が行えるため、プロファイルの gzip 圧縮を開くで作成された圧縮ファイル basic.txt.gz も使用できます。

ステップ 3　TFTP サーバの仮想ルート ディレクトリに、暗号化された basic.cfg ファイルを保存します。

ステップ 4　テスト用デバイスで Profile_Rule を変更し、元の XML ファイルの代わりに暗号化されたファイルに再同期するようにします。暗号キーは、次の URL オプションで Cisco IP Phone に通知されます。

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

ステップ 5　[すべての変更を送信（Submit All Changes）] をクリックします。

ステップ 6　Cisco IP Phone からの syslog トレースを確認します。

再同期時、Cisco IP Phone は新しいファイルをダウンロードし、これを使用して自身のパラメータを更新します。

関連項目

- AES の使用によるオープン プロファイルの暗号化（2-6 ページ）

分けられたプロファイル

Cisco IP Phone は再同期のたびに複数の個別のプロファイルをダウンロードします。この作業により、別々のサーバのさまざまなプロファイル情報を管理し、アカウント固有の値とは別の共通の設定パラメータ値をメンテナンスすることが可能になります。

演習

ステップ 1　これ以前の演習とは異なるパラメータに値を指定する、新しい XML プロファイル basic2.txt を作成します。たとえば、プロファイル basic.txt に次を追加します。

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

ステップ 2　TFTP サーバの仮想ルート ディレクトリにプロファイル basic2.txt を保存します。

ステップ 3　フォルダにある、以前の演習で使用した 1 番目のプロファイル ルールはそのままにして、2 番目のプロファイル ルール（Profile_Rule_B）を設定し、新しいファイルを指すようにします。

```
<Profile_Rule_B ua="na">tftp://192.168.1.200/basic2.txt
</Profile_Rule_B>
```

ステップ 4　[すべての変更を送信（Submit All Changes）] をクリックします。

これで、Cisco IP Phone は、再同期操作の時刻になるたびに、1 番目と 2 番目の両方のプロファイルに再同期するようになりました。

ステップ 5　syslog トレースを参照し、期待した動作が行われていることを確認します。