Catalyst 9800 WLCでのCSR証明書の生成とダウンロード

内容

概要

このドキュメントでは、Catalyst 9800で証明書を生成、ダウンロード、およびインストールするための全体的なプロセスについて説明します

前提条件

要件

次の項目に関する知識があることが推奨されます。

- 9800 WLC、アクセスポイント(AP)の基本動作用の設定方法
- OpenSSL アプリケーションを使用する方法
- Public Key Infrastructure(PKI)とデジタル証明書

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- 9800-L、Cisco IOS® XEバージョン17.3.3
- OpenSSLアプリケーション

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

設定

16.10.Xでは、9800はWeb認証とWeb管理に異なる証明書をサポートしていません。Web口グインポータルでは、常にデフォルトの証明書が使用されます。

16.11.Xでは、Web認証用の専用証明書を設定し、グローバルパラメータマップ内にトラストポイントを定義できます。

9800 WLCの証明書を取得するには、2つのオプションがあります。

1. OpenSSLまたはその他のSSLアプリケーションを使用して証明書署名要求(CSR)を生成します。認証局(CA)によって署名されたPKCS12証明書を取得し、9800 WLCに直接口ードします。これは、秘密キーがその証明書にバンドルされていることを意味します。
2. 9800 WLCのCLIを使用してWLCのCLIでWLCの CSRは、CAによって署名された証明書を取得し、チェーン内の各証明書を手動で9800 WLCにロードします。

ニーズに最も適したソリューションを使用してください。

オプション1：既存のPKCS12署名付き証明書のロード

署名要求の定義

まだ証明書を持っていない場合は、署名要求を生成してCAに渡す必要があります。

（OpenSSLがインストールされているラップトップの）現在のディレクトリからopenssl.cnfファイルを編集し、これらの行をコピーして貼り付け、新しく作成されたCSRのSubject Alternate Names(SAN)フィールドに含めます。

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext
[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
stateOrProvinceName         = State or Province Name (full name)
localityName                = Locality Name (eg, city)
organizationName            = Organization Name (eg, company)
commonName                  = Common Name (e.g. server FQDN or YOUR name)
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1   = testdomain.com
DNS.2   = example.com
DNS.3   = webadmin.com
```

DNS.X名をSANに置き換えます。メインフィールドを必要な証明書の詳細に置き換えます。SANフィールド(DNS.x)内のCommon Name(CN)を繰り返していることを確認します。Google Chromeでは、証明書を信頼するために、URLに含まれる名前がSANフィールドに含まれている必要があります。

Web管理者の場合は、管理者がブラウザのアドレスバーに入力したURLに関係なく証明書が一致するように、SANフィールドにURLのバリエーション(ホスト名のみ、または完全修飾ドメイン名(FQDN)など)を入力する必要もあります。

次のコマンドを使用して、OpenSSLからCSRを生成します。

```
openssl req -out myCSR.csr -newkey rsa:4096 -nodes -keyout private.key -config openssl.cnf
```

CSRはmyCSR.csrとして生成され、そのキーはOpenSSLの実行元のディレクトリにprivate.keyとして生成されます（コマンドに完全なパスが指定されている場合を除く）。

private.keyファイルは、通信の暗号化に使用されるため、安全に保持してください。

次のコマンドを使用して、その内容を確認できます。

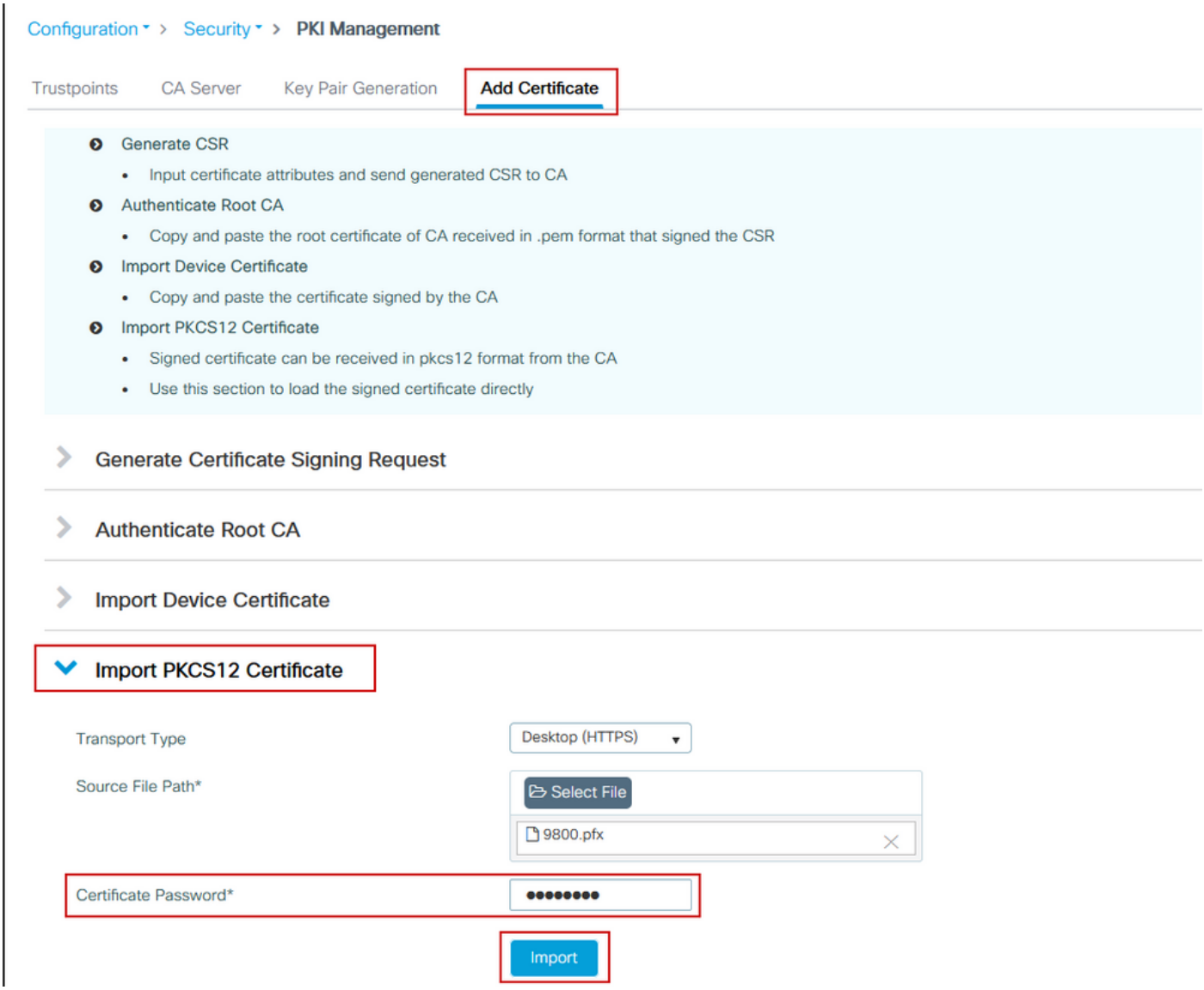
```
openssl req -noout -text -in myCSR.csr
```

その後、このCSRをCAに提供して署名を行い、証明書を受け取ることができます。チェーン全体がCAからダウンロードされ、さらに操作が必要な場合に備えて証明書がBase64形式であることを確認します。

証明書のインポート

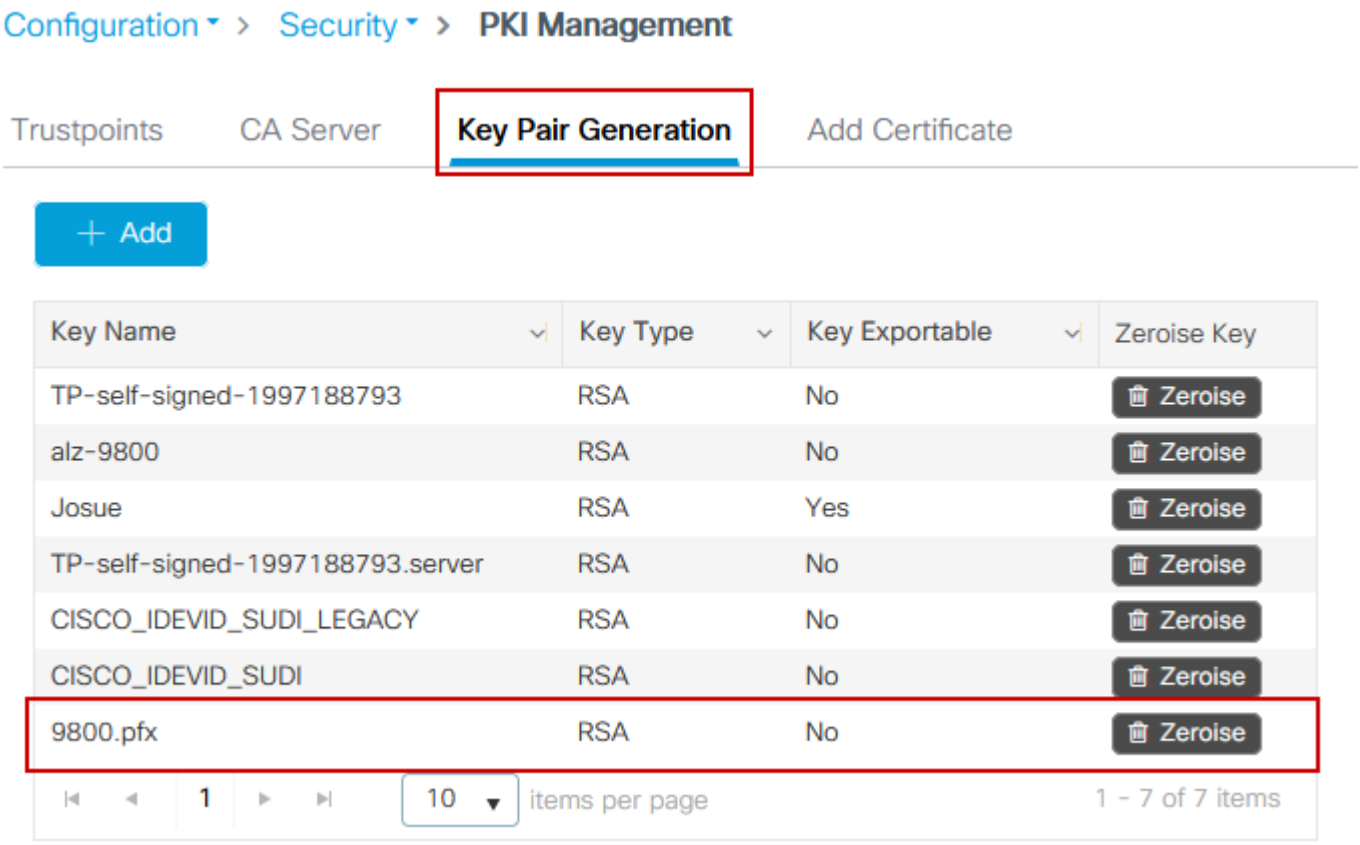
ステップ 1：9800 WLCから到達可能なTrivial File Transfer Protocol(TFTP)サーバにPKCS12証明書を保存します。PKCS12証明書には、秘密キーと、ルートCAまでの証明書チェーンが含まれている必要があります。

ステップ 2：9800 WLC GUIを開き、[Configuration] > [Security] > [PKI Management] に移動し、[Add Certificate] タブをクリックします。**[Import PKCS12 Certificate]** メニューを展開し、TFTPの詳細を入力します。または、[Transport Type] ドロップダウンリストの[Desktop (HTTPS)] オプションを使用すると、ブラウザ経由でのHTTPアップロードが可能になります。**Certificate Password**は、PKCS12証明書の生成時に使用されたパスワードです。



ステップ 3：情報が正しいことを確認し、[インポート(Import)]をクリックします。その後、この新しいトラストポイントの新しい証明書キーペアが[Key Pair Generation] タブにインストールされます。インポートが成功すると、9800 WLCはマルチレベルCA用の追加トラストポイントも作成します。

> **注**：現在、9800 WLCでは、webauthまたはwebadminに特定のトラストポイントが使用されるたびに完全な証明書チェーンが表示されるのではなく、デバイス証明書とその即時発行者が表示されます。これは、Cisco Bug ID CSCwa23606 Cisco IOS® XE 17.8で修正されています。



CLI：

```
9800# configure terminal
9800(config)#crypto pki import
```

注：マルチレベルCA用に追加のトラストポイントを作成するには、9800 WLCで証明書ファイル名とトラストポイント名の両方が正確に一致していることが重要です。

マルチレベルCAシナリオでのPKCS12形式の変換と証明書チェーン。

PEMまたはCRT形式の秘密キーファイルと証明書があり、それらをPKCS12(.pfx)形式で組み合わせて9800 WLCにアップロードする場合があります。これを行うには、次のコマンドを入力します。

```
openssl pkcs12 -export -in
```

証明書のチェーン（1つまたは複数の中間CAとルートCA）がすべてPEM形式である場合は、すべてを1つの.pfxファイルに結合する必要があります。

最初に、CA証明書を1つのファイルに手動で結合します。内容をコピーして貼り付けます（.pem形式でファイルを保存します）。

```
----- BEGIN Certificate --------
<intermediate CA cert>
------END Certificate --------
-----BEGIN Certificate -----
<root CA cert>
-----END Certificate--------
```

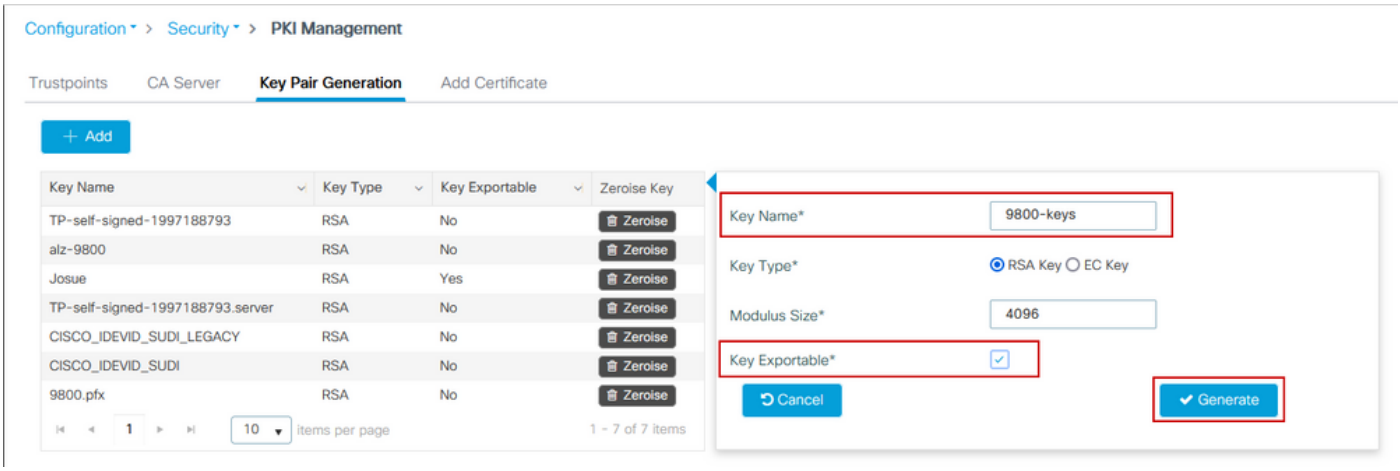
その後、1つのPKCS12証明書ファイル内のすべての証明書を次の証明書と結合できます。

```
openssl pkcs12 -export -out chaincert.pfx -inkey
```

最終的な証明書がどのように表示されるかを確認するには、記事の最後にある「確認」セクションを参照してください。

オプション2:9800 WLCでのキーと署名要求(CSR)の定義

ステップ 1：汎用RSAキーペアを生成します。[Configuration] > [Security] > [PKI Management] に移動し、[Key Pair Generation] タブを選択して、[Add] をクリックします。詳細を入力し、[Key Exportable] チェックボックスがオンになっていることを確認して、[Generate] をクリックします。



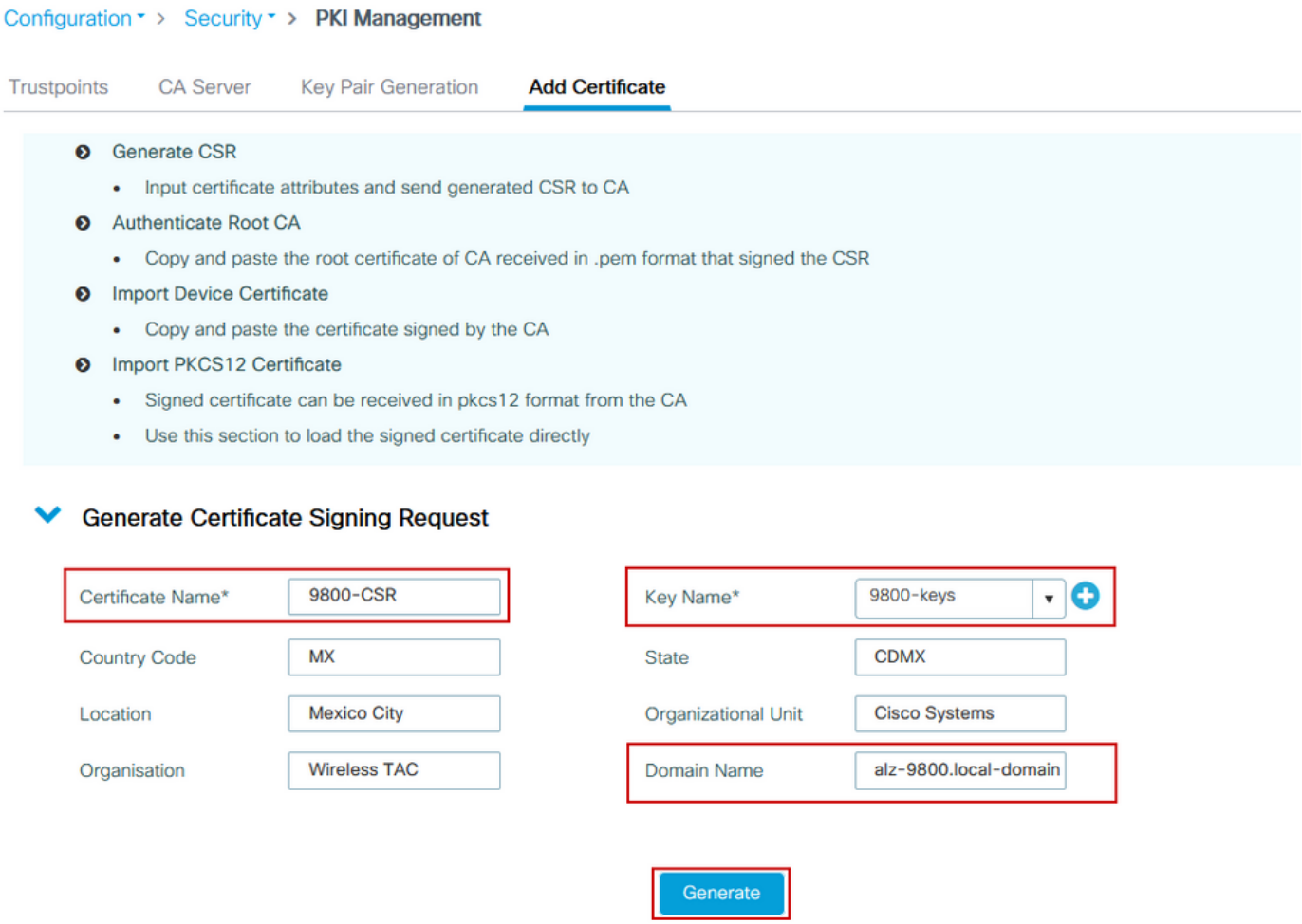
CLI による設定：

```
9800(config)#crypto key generate rsa general-keys label 9800-keys exportable
The name for the keys will be: 9800-keys
Choose the size of the key modulus in the range of 512 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [1024]: 4096
% Generating 4096 bit RSA keys, keys will be exportable...
[OK] (elapsed time was 9 seconds)
```

ステップ 2：9800 WLCのCSRを生成します。[Add Certificate] タブに移動し、[Generate Certificate Signing Request] を展開して詳細を入力し、ドロップダウンリストから以前に作成し

たキーペアを選択します。[Domain Name] が9800 WLC上のクライアントアクセス用に定義されたURL（Web管理ページ、Web認証ページなど）と一致することが重要です。[Certificate Name] はトラストポイント名なので、用途に基づいて名前を付けることができます。

> **注**:9800 WLCでは、共通名にワイルドカードパラメータを含む証明書がサポートされています。



情報が正しいことを確認し、**Generate**をクリックします。元のフォームの横にあるテキストボックスにCSRが表示されます。

Copyは、コピーをクリップボードに保存します。これにより、コピーをテキストエディタに貼り付けてCSRを保存できます。[Save to device] が選択されている場合、9800 WLCはCSRのコピーを作成し、**bootflash:**/csrに保存します。例として、次のコマンドを実行します。

```
9800#dir bootflash:/csr
Directory of bootflash:/csr/
```

```
1046531 -rw- 1844 Sep 28 2021 18:33:49 +00:00 9800-CSR1632856570.csr

26458804224 bytes total (21492699136 bytes free)
9800#more bootflash:/csr/9800-CSR1632856570.csr
-----BEGIN CERTIFICATE REQUEST-----
<Certificate Request>
-----END CERTIFICATE REQUEST-----
```

CLI による設定：

```
9800(config)#crypto pki trustpoint 9800-CSR
9800(ca-trustpoint)#enrollment terminal pem
9800(ca-trustpoint)#revocation-check none
9800(ca-trustpoint)#subject-name C=MX, ST=CDMX, L=Mexico City, O=Cisco Systems, OU=Wireless TaC,
CN=alz-9800.local-domain.com
9800(ca-trustpoint)#rsakeypair 9800-keys
9800(ca-trustpoint)#subject-alt-name domain1.mydomain.com,domain2.mydomain.com
9800(ca-trustpoint)#exit

(config)#crypto pki enroll 9800-CSR
% Start certificate enrollment ..

% The subject name in the certificate will include: C=MX, ST=CDMX, L=Mexico City, O=Cisco
Systems, OU=Wireless TaC, CN=alz-9800.local-domain.com
% The subject name in the certificate will include: alz-9800
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Display Certificate Request to terminal? [yes/no]: yes
Certificate Request follows:

-----BEGIN CERTIFICATE REQUEST-----
<Certificate Request>
-----END CERTIFICATE REQUEST-----

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: no
```

サブジェクト名の設定に使用できるパラメータ：

C:国。2文字の大文字のみにする必要があります。

ST: Some Stateは、州または州名を指します。

L:Location Nameは市区町村を示します。

O：組織名は会社を指します。

OU: Organizational Unit Name（組織単位名）。セクションを参照できます。

CN:(Common Name)証明書の発行先のサブジェクトを参照します。アクセスする特定のIPアドレス（ワイヤレス管理IP、仮想IPなど）またはFQDNを使用して構成されたホスト名を指定する必要があります。

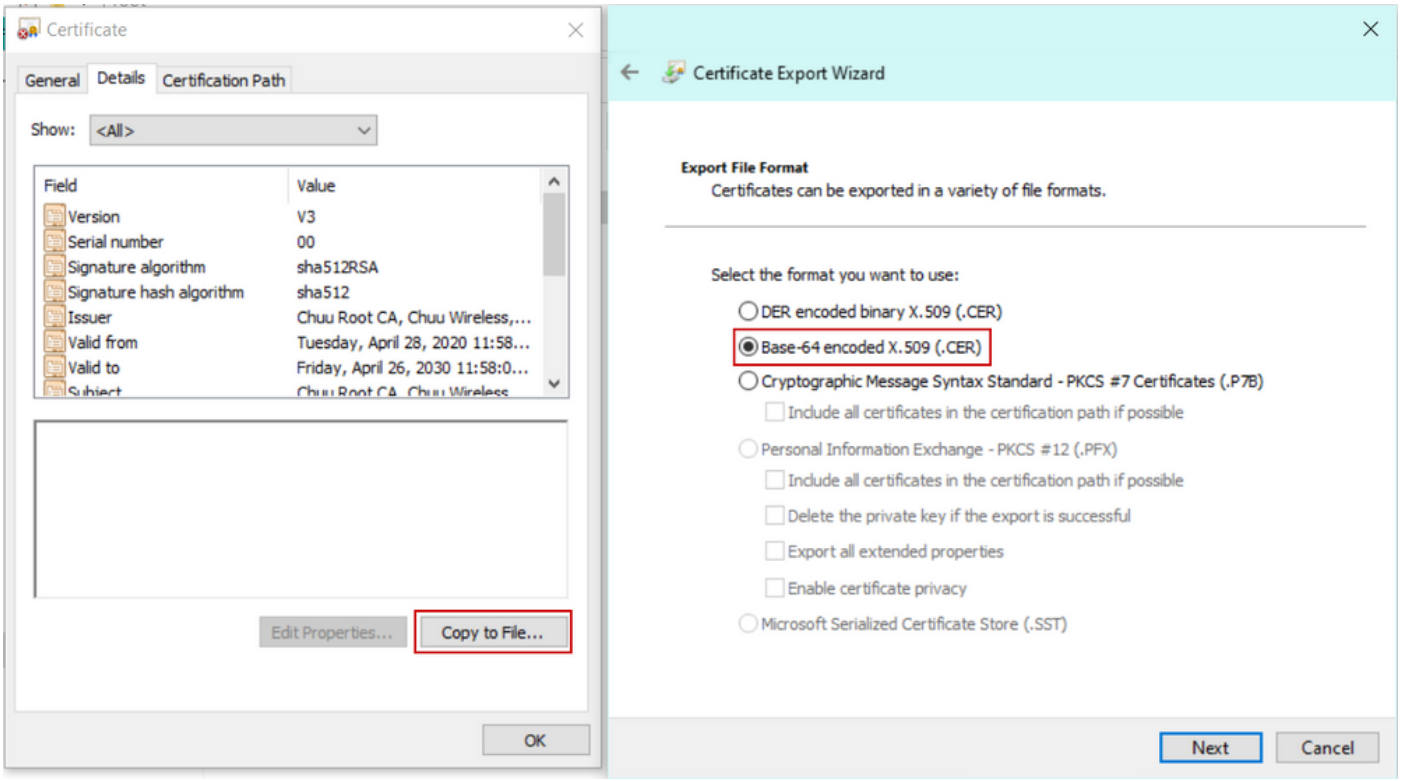
> **注**：サブジェクト代替名を追加する場合、Cisco Bug ID CSCvt15177により、17.8.1より前のCisco IOS XEバージョンでは追加できません .このシナリオでは、SANが存在しないために一部のブラウザアラートが発生する可能性があります。これを回避するには、オプション

1に示すように、キーとCSRをオフボックスで作成します。

ステップ 3：認証局(CA)からCSRに署名してもらいます。完全な文字列は、署名を取得するためにCAに送信する必要があります。

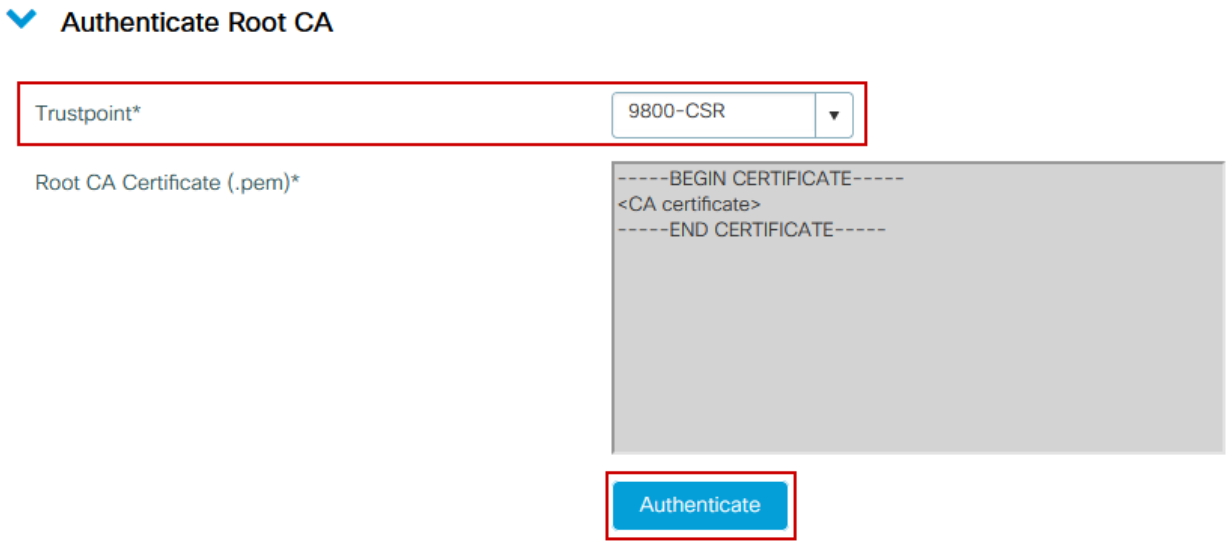
```
-----BEGIN CERTIFICATE REQUEST-----
<Certificate Request>
-----END CERTIFICATE REQUEST-----
```

Windows Server CAを使用して証明書に署名する場合は、署名付き証明書をBase64形式でダウンロードします。それ以外の場合は、Windows Cert Managerなどのユーティリティを使用してエクスポートする必要があります。



注：トラストポイント認証プロセスは、CSRに署名したCAの数によって異なります。シングルレベルCAがある場合は、**ステップ4a**を確認します。マルチレベルCAがある場合は、**ステップ4b**に進みます。トラストポイントは一度に2つの証明書（サブジェクト証明書と発行者証明書）しか保存できないため、これが必要です。

ステップ4a:9800に発行者CAを信頼させます。.pem形式(Base64)で発行者CA証明書をダウンロードします。同じメニュー内の[Authentication Root CA] セクションを展開し、[Trustpoint] ドロップダウンリストから以前に定義したトラストポイントを選択し、発行者CA証明書を貼り付けます。詳細が正しく設定されていることを確認し、**Authenticate**をクリックします。



CLI による設定：

```
9800(config)# crypto pki authenticate 9800-CSR

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
```

```
Certificate has the following attributes: Fingerprint MD5: DD05391A 05B62573 A38C18DD CDA2337C
Fingerprint SHA1: 596DD2DC 4BF26768 CFB14546 BC992C3F F1408809 % Do you accept this certificate?
[yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

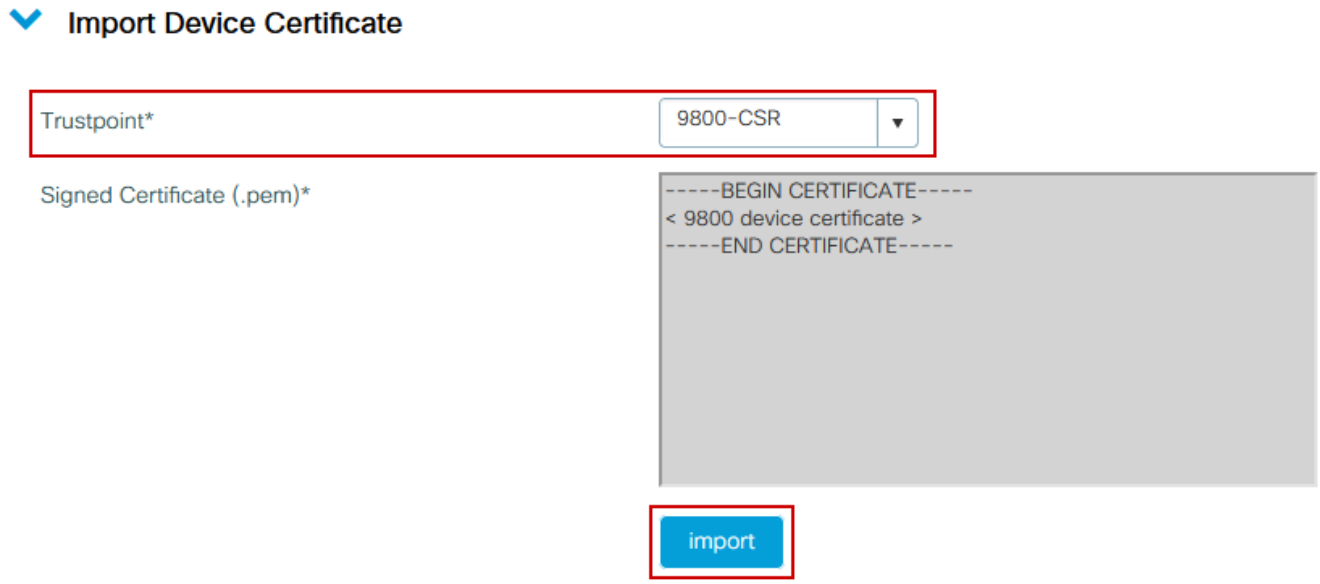
ステップ4b:複数の認可レベルが存在するシナリオでは、CAレベルごとに新しいトラストポイントが必要です。これらのトラストポイントには認証証明書のみが含まれ、次のレベルの認証をポイントします。このプロセスはCLIでのみ実行され、この例では1つの中間CAと1つのルートCAがあります。

```
9800(config)#crypto pki trustpoint root
9800(ca-trustpoint)#enrollment terminal
9800(ca-trustpoint)#chain-validation stop
9800(ca-trustpoint)#revocation-check none
9800(ca-trustpoint)#exit
9800(config)#crypto pki authenticate root

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
```

```
-----END CERTIFICATE-----

Certificate has the following attributes:
Fingerprint MD5: 6CAC00D5 C5932D01 B514E413 D41B37A8
Fingerprint SHA1: 5ABD5667 26B7BD0D 83BDFC34 543297B7 3D3B3F24

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported

9800(config)#crypto pki trustpoint 9800-CSR
9800(ca-trustpoint)#chain-validation continue root
9800(config)#crypto pki authenticate 9800-CSR

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----




-----END CERTIFICATE-----

Certificate has the following attributes:
Fingerprint MD5: DD05391A 05B62573 A38C18DD CDA2337C
Fingerprint SHA1: 596DD2DC 4BF26768 CFB14546 BC992C3F F1408809
Certificate validated - Signed by existing trustpoint CA certificate.

Trustpoint CA certificate accepted.
% Certificate successfully imported
```

> **注**：認証チェーンに複数の中間CAがある場合は、追加の認証レベルごとに新しいトラストポイントを生成する必要があります。このトラストポイントは、コマンド**chain-validation continue <trustpoint-name>**を使用して、次のレベルの証明書を含むトラストポイントを参照する必要があります。

ステップ 5：9800 WLCに署名付き証明書をロードします。同じメニューで[Import Device Certificate] セクションを展開します。以前に定義した**トラストポイント**を選択し、CAによって提供された署名付きデバイス証明書を貼り付けます。次に、証明書情報を確認したら、[import] をクリックします。



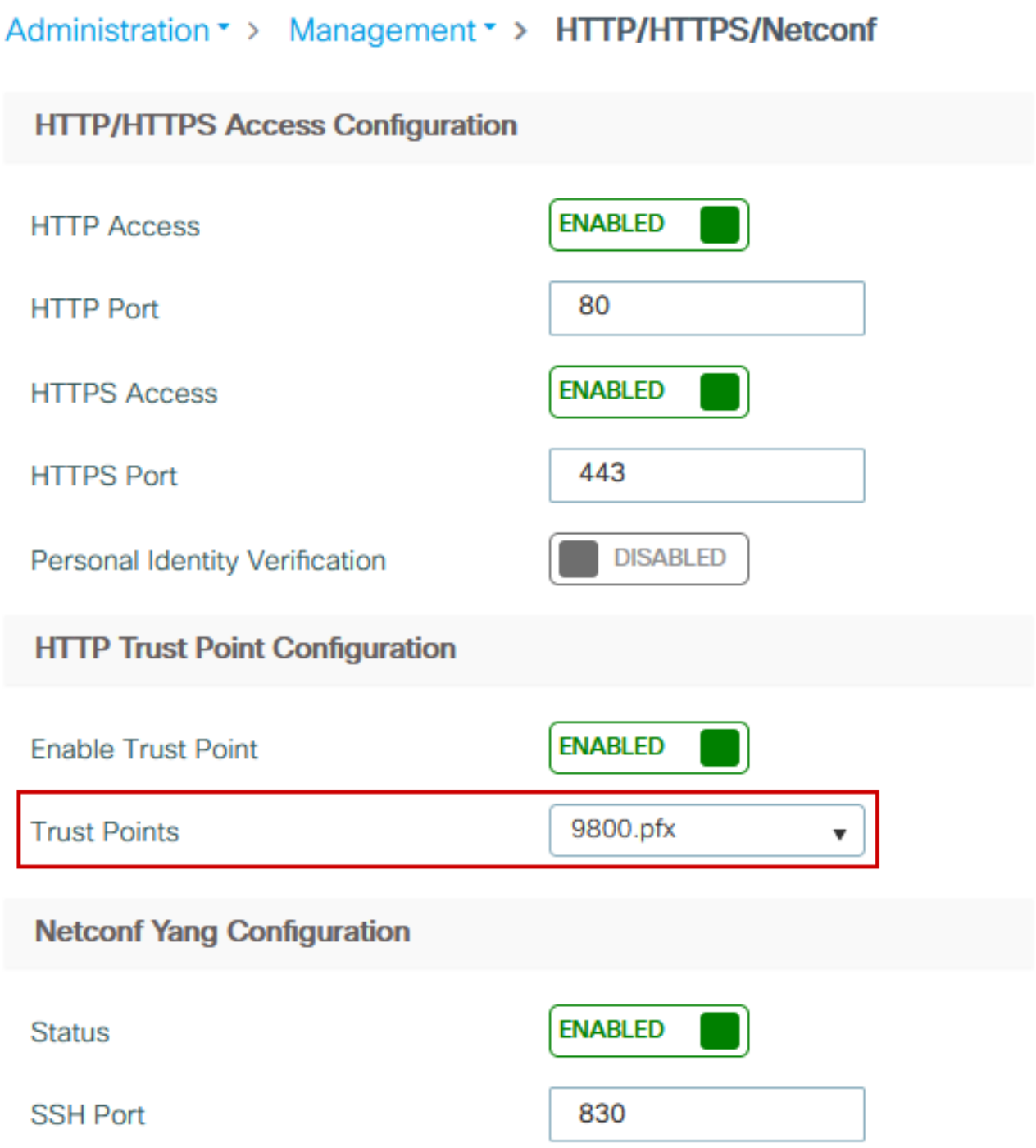
CLI による設定：

```
9800(config)#crypto pki import 9800-CSR certificate

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
<9800 device certificate >
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

新しい証明書の使用

Web管理

[Administration] > [Management] > [HTTP/HTTPS/Netconf] に移動し、[Trust Points] ドロップダウンリストからインポートした証明書を選択します。

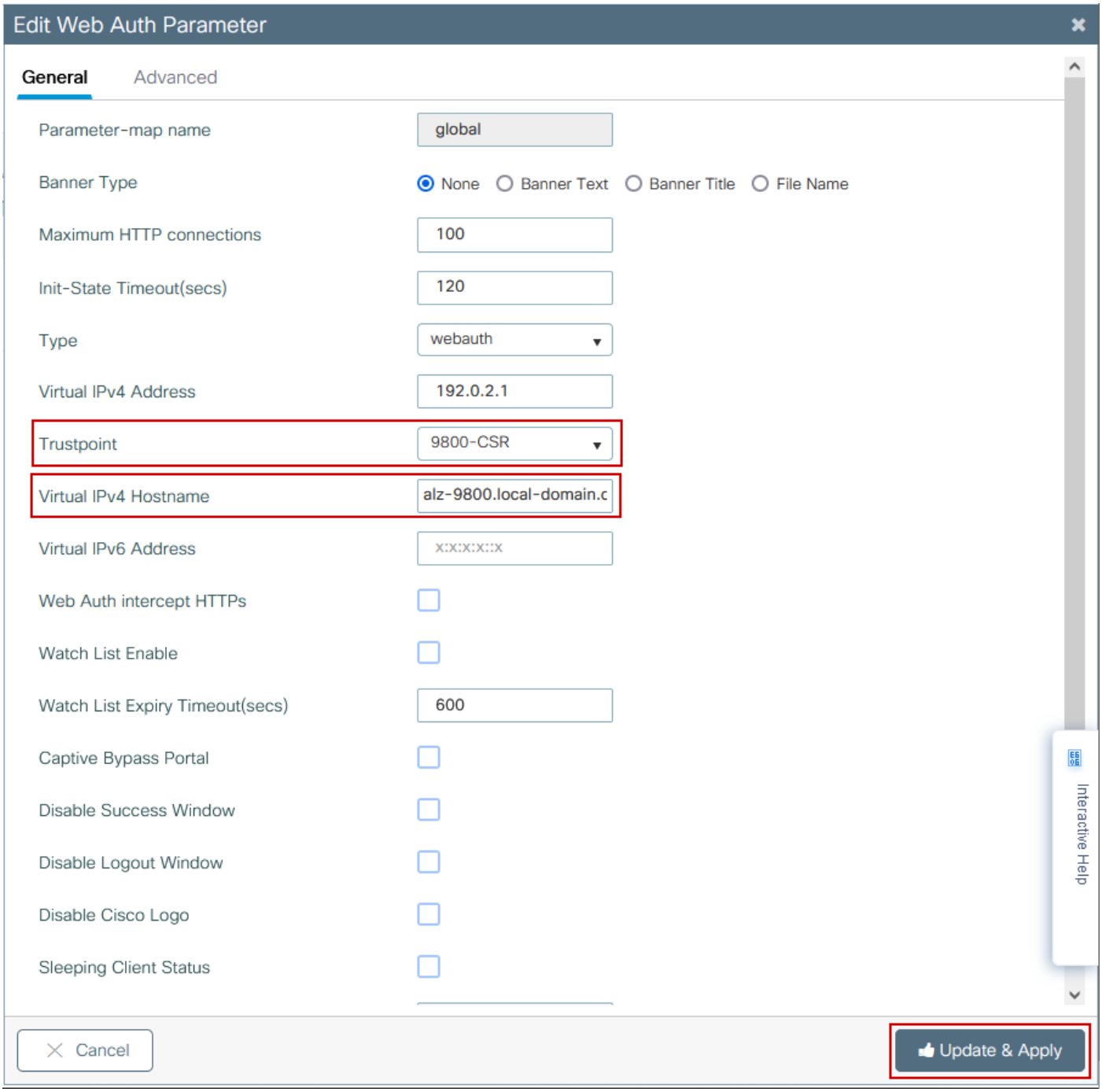


CLI による設定：

```
9800(config)#ip http secure-trustpoint 9800.pfx
9800(config)#no ip http secure-server
9800(config)#ip http secure-server
```

ローカル Web 認証

[Configuration] > [Security] > [Web Auth] に移動し、[global] パラメータマップを選択し、[Trustpoint] ドロップダウンリストからインポートしたトラストポイントを選択します。[Update & Apply] をクリックして変更を保存します。[Virtual IPv4 Hostname] が証明書の[Common Name]と一致することを確認します。



CLI による設定：

```
9800(config)#parameter-map type webauth global
9800(config-params-parameter-map)#type webauth
9800(config-params-parameter-map)#virtual-ip ipv4 192.0.2.1 virtual-host alz-9800.local-domain.com
9800(config-params-parameter-map)#trustpoint 9800-CSR
```

証明書の使用状況を更新するには、HTTPサービスを再起動します。

```
9800(config)#no ip http server
9800(config)#ip http server
```

.

ハイアベイラビリティの考慮事項

Stateful Switchover High Availability(HA SSO)用に設定された9800ペアでは、最初のバルク同期ですべての証明書がプライマリからセカンダリに複製されます。これには、RSAキーがエクスポート不可に設定されている場合でも、コントローラ自体で秘密キーが生成された証明書が含まれます。HAペアが確立されると、インストールされた新しい証明書が両方のコントローラにインストールされ、すべての証明書がリアルタイムで複製されます。

障害が発生した後、元のセカンダリ現在アクティブなコントローラは、プライマリから透過的に継承された証明書を使用します。

証明書がWebブラウザによって信頼されていることを確認する方法

証明書がWebブラウザによって信頼されていることを確認する上で、いくつかの重要な考慮事項があります。

- [Common Name]（または[SAN]フィールド）は、ブラウザがアクセスするURLと一致する必要があります。
- 有効期間の範囲内である必要があります。
- これは、ブラウザによって信頼されるルートを持つCAまたはCAのチェーンによって発行される必要があります。このため、Webサーバによって提供される証明書には、クライアントブラウザ（通常はルートCA）によって信頼される証明書（必ずしも含まれるとは限らない）まで、チェーンのすべての証明書が含まれている必要があります。
- 失効リストが含まれている場合は、ブラウザで失効リストをダウンロードできる必要があり、証明書CNはリストされません。

確認

次のコマンドを使用して、証明書の設定を確認できます。

```
9800#show crypto pki certificate 9800.pfx

Certificate
Status: Available
Certificate Serial Number (hex): 1236
Certificate Usage: General Purpose
Issuer:
cn=Chuu Intermediate CA
ou=Chuu Wireless
o=Chuu Inc
st=CDMX
c=MX
Subject:
Name: alz-9800
e=user@example.com
cn=alz-9800
ou=Cisco Systems
o=Wireless TAC
l=CDMX
st=CDMX
c=MX
Validity Date:
start date: 17:54:45 Pacific Sep 28 2021
end date: 17:54:45 Pacific Sep 26 2031
Associated Trustpoints: 9800.pfx
```

```
CA Certificate
Status: Available
Certificate Serial Number (hex): 1000
Certificate Usage: Signature
Issuer:
cn=Chuu Root CA
ou=Chuu Wireless
o=Chuu Inc
l=Iztapalapa
st=CDMX
c=MX
Subject:
cn=Chuu Intermediate CA
ou=Chuu Wireless
o=Chuu Inc
st=CDMX
c=MX
Validity Date:
start date: 05:10:34 Pacific Apr 29 2020
end date: 05:10:34 Pacific Apr 27 2030
Associated Trustpoints: 9800.pfx
```

```
9800#show ip http server secure status

HTTP secure server status: Enabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha aes-128-cbc-sha
aes-256-cbc-sha dhe-aes-128-cbc-sha ecdhe-rsa-3des-ede-cbc-sha
rsa-aes-cbc-sha2 rsa-aes-gcm-sha2 dhe-aes-cbc-sha2 dhe-aes-gcm-sha2
ecdhe-rsa-aes-cbc-sha2 ecdhe-rsa-aes-gcm-sha2
HTTP secure server TLS version: TLSv1.2 TLSv1.1 TLSv1.0
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: 9800.pfx
HTTP secure server active session modules: ALL
```

9800で証明書チェーンを確認できます。中間CAによって発行されたデバイス証明書の場合、ルートCAによって自身が発行されたデバイス証明書の場合、2つの証明書のグループによって1つのトラストポイントが作成され、各レベルに独自のトラストポイントが作成されます。この場合、9800 WLCにはデバイス証明書（WLC証明書）と発行側CA（中間CA）を持つ9800.pfxがあります。次に、その中間CAを発行したルートCAを持つ別のトラストポイントが作成されます。

```
9800#show crypto pki certificate 9800.pfx
Certificate
Status: Available
Certificate Serial Number (hex): 1236
Certificate Usage: General Purpose
Issuer:
cn=Chuu Intermediate CA
ou=Chuu Wireless
o=Chuu Inc
st=CDMX
c=MX
Subject:
Name: alz-9800
e=user@example.com
cn=alz-9800
ou=Cisco Systems
o=Wireless TAC
l=CDMX
st=CDMX
c=MX
Validity Date:
```

```
start date: 17:54:45 Pacific Sep 28 2021
end date: 17:54:45 Pacific Sep 26 2031
Associated Trustpoints: 9800.pfx

CA Certificate
Status: Available
Certificate Serial Number (hex): 1000
Certificate Usage: Signature
Issuer:
cn=Chuu Root CA
ou=Chuu Wireless
o=Chuu Inc
l=Iztapalapa
st=CDMX
c=MX
Subject:
cn=Chuu Intermediate CA
ou=Chuu Wireless
o=Chuu Inc
st=CDMX
c=MX
Validity Date:
start date: 05:10:34 Pacific Apr 29 2020
end date: 05:10:34 Pacific Apr 27 2030
Associated Trustpoints: 9800.pfx

9800#show crypto pki certificate 9800.pfx-rrr1
CA Certificate
Status: Available
Certificate Serial Number (hex): 00
Certificate Usage: Signature
Issuer:
cn=Chuu Root CA
ou=Chuu Wireless
o=Chuu Inc
l=Iztapalapa
st=CDMX
c=MX
Subject:
cn=Chuu Root CA
ou=Chuu Wireless
o=Chuu Inc
l=Iztapalapa
st=CDMX
c=MX
Validity Date:
start date: 04:58:05 Pacific Apr 29 2020
end date: 04:58:05 Pacific Apr 27 2030
Associated Trustpoints: 9800-CSR 9800.pfx-rrr1
```

OpenSSLを使用した証明書の検証

OpenSSLは、証明書自体を確認したり、変換操作を行ったりするのに役立ちます。

OpenSSLで証明書を表示するには、次の手順を実行します。

```
openssl x509 -in
```

CSRの内容を表示するには、次の手順を実行します。

```
openssl req -noout -text -in
```

9800 WLC上の終了証明書を確認したいが、ブラウザ以外のものを使用したい場合は、OpenSSLでこれを実行して多くの詳細を提供できます。

```
openssl s_client -showcerts -verify 5 -connect
```

<wlcURL>は、9800のwebadminのURL（仮想IP）またはゲストポータルのURL（仮想IP）に置き換えることができます。IPアドレスを設定することもできます。これにより、どの証明書チェーンが受信されたかを確認できますが、ホスト名の代わりにIPアドレスを使用する場合、証明書の検証が100%正しいとは限りません。

内容を表示し、PKCS12(.pfx)証明書または証明書チェーンを確認するには、次の手順を実行します。

```
openssl pkcs12 -info -in
```

次に、証明書のチェーンに対するこのコマンドの例を示します。このコマンドでは、「intermediate.com」という中間CAによってデバイス証明書がTechnical Assistance Center(TAC)に発行されます。この中間CA自体は「root.com」というルートCAによって発行されます。

```
openssl pkcs12 -info -in chainscript2.pfx
Enter Import Password:
MAC Iteration 2048
MAC verified OK
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 2048
Certificate bag
Bag Attributes
localKeyID: 1D 36 8F C2 4B 18 0B 0D B2 57 A2 55 18 96 7A 8B 57 F9 CD FD
subject=/C=BE/ST=Diegem/L=Diegem/O=Cisco/CN=TAC
issuer=/C=BE/ST=Diegem/O=Cisco/OU=TAC/CN=intermediate.com/emailAddress=int@int.com
-----BEGIN CERTIFICATE-----
<Device certificate >
-----END CERTIFICATE-----
Certificate bag
Bag Attributes: <No Attributes>
subject=/C=BE/ST=Diegem/O=Cisco/OU=TAC/CN=intermediate.com/emailAddress=int@int.com
issuer=/C=BE/ST=Diegem/L=Diegem/O=Cisco/OU=TAC/CN=RootCA.root.com/emailAddress=root@root.com
-----BEGIN CERTIFICATE-----
<Intermediate certificate >
-----END CERTIFICATE-----
```

```
Certificate bag
Bag Attributes: <No Attributes>
subject=/C=BE/ST=Diegem/L=Diegem/O=Cisco/OU=TAC/CN=RootCA.root.com/emailAddress=root@root.com
issuer=/C=BE/ST=Diegem/L=Diegem/O=Cisco/OU=TAC/CN=RootCA.root.com/emailAddress=root@root.com
-----BEGIN CERTIFICATE-----
<Root certificate >
-----END CERTIFICATE-----
PKCS7 Data
Shrouded Keybag: pbeWithSHA1And3-KeyTripleDES-CBC, Iteration 2048
Bag Attributes
localKeyID: 1D 36 8F C2 4B 18 0B 0D B2 57 A2 55 18 96 7A 8B 57 F9 CD FD
Key Attributes: <No Attributes>
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----BEGIN ENCRYPTED PRIVATE KEY-----
<Private key >
-----END ENCRYPTED PRIVATE KEY-----
```

トラブルシュート

このコマンドを使用してトラブルシューティングを行います。リモートセッション（SSHまたはTelnet）で実行する場合、出力を表示するにはterminal monitorが必要です。

```
9800#debug crypto pki transactions
```

成功したシナリオデバッグ出力

次の出力は、9800で証明書のインポートが成功した場合に予想される出力を示しています。これを参考にして、障害の状態を特定してください。

```
Sep 28 17:35:23.242: CRYPTO_PKI: Copying pkcs12 from bootflash:9800.pfx
Sep 28 17:35:23.322: CRYPTO_PKI: Creating trustpoint 9800.pfx
Sep 28 17:35:23.322: %PKI-6-TRUSTPOINT_CREATE: Trustpoint: 9800.pfx created succesfully
Sep 28 17:35:23.324: CRYPTO_PKI: examining cert:
Sep 28 17:35:23.324: CRYPTO_PKI: issuerName=cn=Chuu Intermediate CA,ou=Chuu Wireless,o=Chuu
Inc,st=CDMX,c=MX
Sep 28 17:35:23.324: CRYPTO_PKI: subjectname=e=user@example.com,cn=alz-9800,ou=Cisco
Systems,o=Wireless TAC,l=CDMX,st=CDMX,c=MX
Sep 28 17:35:23.324: CRYPTO_PKI: adding RSA Keypair
Sep 28 17:35:23.324: CRYPTO_PKI: bitValue of ET_KEY_USAGE = 140
Sep 28 17:35:23.324: CRYPTO_PKI: Certificate Key Usage = GENERAL_PURPOSE
Sep 28 17:35:23.324: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named 9800.pfx has been generated or
imported by pki-pkcs12
Sep 28 17:35:23.331: CRYPTO_PKI: adding as a router certificate.Public key in cert and stored
public key 9800.pfx match

Sep 28 17:35:23.333: CRYPTO_PKI: examining cert:
Sep 28 17:35:23.333: CRYPTO_PKI: issuerName=cn=Chuu Root CA,ou=Chuu Wireless,o=Chuu
Inc,l=Iztapalapa,st=CDMX,c=MX
Sep 28 17:35:23.333: CRYPTO_PKI: subjectname=cn=Chuu Intermediate CA,ou=Chuu Wireless,o=Chuu
Inc,st=CDMX,c=MX
Sep 28 17:35:23.333: CRYPTO_PKI: no matching private key presents.

[...]

Sep 28 17:35:23.335: CRYPTO_PKI: Setting the key_type as RSA
Sep 28 17:35:23.335: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Sep 28 17:35:23.335: CRYPTO_PKI:Peer's public inserted successfully with key id 21
```

```
Sep 28 17:35:23.336: Calling pkiSendCertInstallTrap to send alert
Sep 28 17:35:23.337: CRYPTO_PKI: Deleting cached key having key id 31
Sep 28 17:35:23.337: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Sep 28 17:35:23.337: CRYPTO_PKI:Peer's public inserted successfully with key id 32
Sep 28 17:35:23.338: CRYPTO_PKI: (A0323) Session started - identity selected (9800.pfx)
Sep 28 17:35:23.338: CRYPTO_PKI: Rcvd request to end PKI session A0323.
Sep 28 17:35:23.338: CRYPTO_PKI
alz-9800#: PKI session A0323 has ended. Freeing all resources.
Sep 28 17:35:23.338: CRYPTO_PKI: unlocked trustpoint 9800.pfx, refcount is 0
Sep 28 17:35:23.338: CRYPTO_PKI: Expiring peer's cached key with key id 32Public key in cert and
stored public key 9800.pfx match

Sep 28 17:35:23.341: Calling pkiSendCertInstallTrap to send alert
Sep 28 17:35:23.341: CRYPTO_PKI: cert verified and inserted.
Sep 28 17:35:23.402: CRYPTO_PKI: Creating trustpoint 9800.pfx-rrr1
Sep 28 17:35:23.402: %PKI-6-TRUSTPOINT_CREATE: Trustpoint: 9800.pfx-rrr1 created succesfully
Sep 28 17:35:23.403: CRYPTO_PKI: Setting the key_type as RSA
Sep 28 17:35:23.404: CRYPTO_PKI: Attempting to insert the peer's public key into cache
Sep 28 17:35:23.404: CRYPTO_PKI:Peer's public inserted successfully with key id 22
Sep 28 17:35:23.405: Calling pkiSendCertInstallTrap to send alert
Sep 28 17:35:23.406: CRYPTO_PKI: no CRLs present (expected)
Sep 28 17:35:23.406: %PKI-6-PKCS12_IMPORT_SUCCESS: PKCS #12 import in to trustpoint 9800.pfx
successfully imported.
```

CAを持たないPKCS12証明書のインポートの試行

証明書をインポートして「CA cert is not found.」というエラーが表示された場合は、.pfxファイルにチェーン全体が含まれていないか、1つのCAが存在しないことを意味します。

```
9800(config)#crypto pki import pkcs12.pfx pkcs12 bootflash:pks12.pfx password
```

```
% Importing pkcs12...
Source filename [pks12.pfx]?
Reading file from bootflash:pks12.pfx
% Warning: CA cert is not found. The imported certs might not be usable.
```

openssl pkcs12 -info -in <path to cert>コマンドを実行し、秘密キーを1つ持つ証明書が1つだけ表示される場合は、そのCAが存在しないことを意味します。原則として、このコマンドは証明書のチェーン全体をリストするのが理想的です。すでにクライアントブラウザで認識されている場合は、最上位ルートCAを含める必要はありません。

これを修正する1つの方法は、PEMにPKCS12をデコンストラクトし、チェーンを正しく再構築することです。次の例では、デバイス(WLC)証明書とそのキーのみを含む.pfxファイルがありました。これは、PKCS12ファイルに存在しない中間CAによって発行され、次に既知のルートCAによって署名されました。

ステップ 1：秘密キーをエクスポートします。

```
openssl pkcs12 -in
```

ステップ 2：証明書をPEMとしてエクスポートします。

```
openssl pkcs12 -in
```

ステップ 3：中間CA証明書をPEMとしてダウンロードします。

CAのソースはその性質に依存します。パブリックCAの場合、オンライン検索でリポジトリを見つけるのに十分です。それ以外の場合、CA管理者はBase64形式(.pem)で証明書を提供する必要があります。CAのレベルが複数ある場合は、**オプション1**のインポートプロセスの最後に示したような単一のファイルにグループ化します。

ステップ 4：キー、デバイス証明書、およびCA証明書からPKCS 12を再構築します。

```
openssl pkcs12 -export -out fixedcertchain.pfx -inkey cert.key -in certificate.pem -certfile
CA.pem
```

これで「fixedcertchain.pfx」が作成されました。このファイルをCatalyst 9800にインポートできます。

注意と制限

- Cisco IOS® XEは、2099年以降の有効なCA証明書をサポートしていません。Cisco Bug ID CSCvp64208
- Cisco IOS® XEはSHA256 Message Digest PKCS 12バンドルをサポートしていません（SHA256証明書はサポートされていますが、PKCS12バンドル自体がSHA256で署名されている場合はサポートされません）。Cisco Bug ID CSCvz41428
- WLCがユーザ証明書を伝送する必要があり、NAC/ISEアプライアンスがインターネット経由で到達可能な場合（SD-WAN展開など）、フラグメンテーションを確認できます。証明書は、ほぼ常に1500バイトより大きくなります（つまり、証明書メッセージを伝送するためにいくつかのRADIUSパケットが送信されます）。また、ネットワークパス上に複数の異なるMTUがある場合は、RADIUSパケット自体のフラグメンテーションが発生する可能性があります。このような場合、インターネットの天候によって引き起こされる可能性がある遅延/ジッタなどの問題を回避するために、WLCトラフィックのすべてのUDPデータグラムを同じパス経由で送信することを推奨します

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。