Konfigurieren von Zertifikaten der Zertifizierungsstelle mithilfe der IOS XE PKI

Inhalt

Einleitung

Dieses Dokument dient als allgemeine Anleitung zum Konfigurieren von IOS XE-Zertifikaten, die von einer Zertifizierungsstelle eines Drittanbieters signiert wurden.

In diesem Dokument wird erläutert, wie Sie eine mehrstufige signierte Zertifizierungsstellenkette importieren, damit das Gerät als Identitäts- (ID)-Zertifikat fungieren kann, und wie Sie andere Zertifikate von Drittanbietern zum Zwecke der Zertifikatsvalidierung importieren.

Voraussetzungen

Anforderungen

NTP und Uhrzeit **MÜSSEN** bei Verwendung der IOS PKI-Funktionen konfiguriert werden.

Wenn ein Administrator kein NTP konfiguriert, kann es Probleme mit einem Zertifikat geben, das mit einem zukünftigen/vergangenen Datum/einer vergangenen Uhrzeit generiert wird. Diese Datums- oder

Zeitverschiebung kann zu Importproblemen und anderen Problemen führen.

NTP-Beispielkonfiguration:

```
ntp server 192.168.1.1
clock timezone EST -5
clock summer-time EDT recurring
```

Verwendete Komponenten

- Cisco Router mit Cisco IOS® XE17.11.1a

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Beachten Sie, dass einige der in diesem Dokument beschriebenen Funktionen in älteren IOS XE-Versionen möglicherweise nicht verfügbar sind. Wenn möglich wurde sorgfältig dokumentiert, wenn ein Befehl oder eine Funktion eingeführt oder geändert wurde.

Beachten Sie immer die offizielle Dokumentation der IOS XE PKI-Funktionen für eine bestimmte Version, um alle Einschränkungen oder Änderungen zu verstehen, die für Ihre Version relevant sein können:

Beispiele:

- IOS 15 M/T: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/15-mt/sec-pki-15-mt-book/sec-pki-overview.html
- IOS XE 16.12.x: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/xe-16-12/sec-pki-xe-16-12-book/sec-est-client-supp-pki.html
- IOS XE 17.x: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m_sec-pki-overview-0.html

IOS XE PKI-Konfiguration

Auf hoher Ebene muss ein Administrator beim Arbeiten mit IOS XE PKI-Zertifikaten folgende Aktionen ausführen:

1. Erstellen eines Schlüssels zur Verwendung mit einer Funktion oder einem Dienst (**Generierung eines Kryptografieschlüssels**)
2. Konfigurieren Sie einen Vertrauenspunkt mit verschiedenen Parametern, und verknüpfen Sie den Schlüssel. (**crypto pki trustpoint**)
3. Erstellen einer Zertifikatssignierungsanforderung (CSR) (**crypto pki enroll**)
4. Übergeben Sie den CSR an eine Zertifizierungsstelle zur Signatur (*nicht in diesem Dokument enthalten*)
5. Authentifizieren der Root- und/oder Zwischen-Zertifizierungsstellenzertifikate (**crypto pki authentication**)
6. Importieren Sie die Gerätezertifikate (**crypto pki import**)

7. Optional: Authentifizierung von Peer-CA-Zertifikaten **(crypto pki authentication**)

Diese Schritte werden in den folgenden Abschnitten beschrieben, die nach den für die jeweilige Aktion erforderlichen Befehlen gruppiert sind.

Kryptografieschlüssel erzeugen

Viele Administratoren haben diesen Befehl eingegeben, um Secure Socket Shell (SSH) auf einem Router zu aktivieren, oder als Teil eines Konfigurationsleitfadens für eine Funktion. Allerdings haben nur wenige nicht seziert, was der Befehl tatsächlich tut.

Nehmen wir zum Beispiel die folgenden Befehle:

```
crypto key generate rsa general-keys modulus 2048 label rsaKey exportable
crypto key generate ec keysize 521 exportable label ecKey
```

Durch Unterteilen dieser Befehle in die einzelnen Teile wird die Verwendung im Detail beschrieben:

- Der erste Teil des Befehls in schwarz (crypto key generate) weist den Router an, einen neuen Schlüssel zu erstellen. Es gibt noch weitere Optionen, wie z. B. den Export von Kryptoschlüsseln, den Import von Kryptoschlüsseln oder die Nullgröße von Kryptoschlüsseln, die später noch genauer beschrieben werden.
- Der nächste Teil des Befehls in **Grün (**rsa general-keys, etc) weist den Router genau an, welche Art von Schlüssel erstellt wird. Für die meisten Zwecke wird eine Rivest-Shamir-Adleman (RSA)-Tastatur verwendet, die aus einem öffentlichen/privaten Schlüssel besteht. Ein Administrator kann jedoch auch eine elliptische Kurve (EC) für die Verwendung mit Funktionen konfigurieren, die beispielsweise ECDSA-Zertifikate erfordern, oder für die Verwendung mit ECDHE-Handshakes.
- Der Befehl in **Orange** definiert die Größe unseres Schlüssels.
    - Für RSA ist der Modul die Terminologie, und es stehen Werte zwischen 512-4096 zur Verfügung. Die standardmäßige Modulgröße variiert je nach Version. Es wird jedoch empfohlen, die Cisco Best Practice für Verschlüsselung der nächsten Generation zu befolgen und Schlüssel mit einer Größe über 2048 zu verwenden.
    - Für EC ist der Befehl "key-size" erforderlich, um die Anzahl der Bits im Schlüssel anzugeben. Die Optionen sind 256, 384 oder 512.
- Der Befehl in **Violett** definiert die Bezeichnung für diesen Schlüssel. Dies ist wichtig, da ein Administrator möglicherweise mehrere Schlüssel für verschiedene Zwecke auf demselben IOS XE-Gerät definieren muss. Mit der Bezeichnung wird der genaue Schlüssel angegeben, der für eine bestimmte Funktion verwendet werden soll. Verwenden Sie nach Möglichkeit immer eine Beschriftung, um die verwendeten Schlüssel zu unterscheiden und die Zuweisung von Schlüsseln zu Funktionen zu erleichtern. Beispiel: Label SSH, Label CUBE, Label HTTPS erstellt zwei Schlüssel für die Verwendung mit verschiedenen Services oder Funktionen.
    - Das Standardlabel für einen Schlüssel ist "devices hostname.domain". Einige Geräte generieren möglicherweise RSA-Schlüssel beim ersten Start. Wenn ein Administrator kein Label-Post-Fix eingibt, besteht die Gefahr, dass er versehentlich den falschen Schlüssel überschreibt oder neu generiert.
- Der letzte Befehl in **blau** ist der exportierbare Postfix. Mit diesem Befehl wird angegeben, dass der Schlüssel mit dem Befehl **crypto pki export** zum Exportieren und zur Verwendung mit anderen Systemen verwendet werden kann. Ein Beispiel wäre der Import in ein Peer-Hochverfügbarkeitsgerät, sodass beide Mitglieder eines HA-Paares einen einzelnen Schlüssel verwenden oder zur Verwendung in Tools zur Fehlerbehebung wie Wireshark, um RSA-basierte TLS-Sitzungen zu entschlüsseln. Was auch immer der Grund ist, es muss gesagt werden, dass die RSA-Schlüssel können nur als exportfähig

von Anfang an erstellt werden. Wenn ein Administrator einen nicht exportierbaren RSA-Schlüssel erstellt, kann dieser Schlüssel nicht als exportierbar festgelegt werden, ohne den Schlüssel neu zu generieren. Dies kann Auswirkungen auf andere Funktionen haben, z. B. das Ungültigmachen aller mit diesem Schlüssel erstellten Zertifikate. Allerdings kann ein exportierbarer Schlüssel auf nicht exportierbar herabgestuft werden, ohne den Schlüssel neu zu generieren, indem der Befehl **crypto key move rsa rsaKeyLabel non-exportabel** verwendet wird

Konfigurationsbeispiele:

<#root>

```
Router(config)#

crypto key generate rsa general-keys modulus 2048 label rsaKey exportable

The name for the keys will be: rsaKey

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be exportable...
[OK] (elapsed time was 1 seconds)

Router(config)#

crypto key generate ec keysize 521 exportable label ecKey

The name for the keys will be: ecKey
```

Verifizierungsbeispiele:

<#root>

```
Router#

show crypto key mypubkey rsa rsaKey

% Key pair was generated at: 10:21:42 EDT Apr 14 2023
Key name: rsaKey
Key type: RSA KEYS      2048 bits
 Storage Device: not specified
 Usage: General Purpose Key
 Key is exportable. Redundancy enabled.
 Key Data:
  30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
  [..truncated..]
  9F020301 0001

Router#

show crypto key mypubkey ec ecKey

% Key pair was generated at: 10:03:05 EDT Apr 14 2023
Key name: ecKey
Key type: EC KEYS      p521 curve
 Storage Device: private-config
 Usage: Signature Key
 Key is exportable. Redundancy enabled.
 Key Data:
  30819B30 1006072A 8648CE3D 02010605 2B810400 23038186 000401A2 A77FCD34
  [..truncated..]
```

```
  93FAC967 96ADA79E 4A245881 B2AD2F4A 279A362D F390A20F C06D5845 06DA
```

crypto pki trustpoint

Vertrauenspunkte sind ein "ordnerähnliches" Konzept zum Speichern und Verwalten von PKI-Zertifikaten in IOS XE. (Befehlssyntax)

Auf hoher Ebene:

1. Jeder IOS XE-Vertrauenspunkt kann ein einzelnes Stamm- oder zwischengeschaltetes CA-Zertifikat enthalten, das mithilfe des Befehls **crypto pki Authenticate** geladen wird. Stellen Sie sich authentifizierte Vertrauenspunkte als das Hinzufügen von Zertifikaten vor, die jetzt vom Gerät als vertrauenswürdig eingestuft werden.
2. Jeder IOS XE-Vertrauenspunkt kann auch ein einzelnes Identitäts-(ID-)Zertifikat importieren, das mithilfe des **crypto pki import-**Befehls geladen wird. Das ID-Zertifikat ist das Gerätezertifikat, das normalerweise mit einem Dienst oder einer Funktion verknüpft ist.
3. Ein Administrator kann den Befehl **Authenticate** and **import** auf demselben Vertrauenspunkt verwenden (der erforderlich ist, um ein ID-Zertifikat zu importieren, das später besprochen wird.). Bei Verwendung des Auth/Import-Workflows enthält der Vertrauenspunkt zwei Zertifikate (Stamm/Zwischenprodukt + Identitätszertifikat).
4. Wenn Vertrauenspunkte zum Speichern von vertrauenswürdigen Peer-Root-/Zwischen-CA-Zertifikaten verwendet werden, **crypto pki authentifizieren** ist erforderlich. In diesem Szenario enthält ein Vertrauenspunkt nur das einzelne vom Administrator authentifizierte Zertifikat.

Hinweis: Die folgenden Abschnitte für **crypto pki Authenticate** und **crypto pki import** und spätere Abschnitte mit detaillierten Auth-/Importbeispielen für mehrstufige Zertifikate bieten weiteren Kontext zu diesen vier Aufzählungszeichen.

Für Vertrauenspunkte können verschiedene Befehle konfiguriert werden. Diese Befehle können verwendet werden, um die Werte innerhalb einer CSR (Certificate Signing Request) zu beeinflussen, die vom Gerät mit dem Befehl **crypto pki enroll** auf einem Vertrauenspunkt erstellt werden.

Für einen Vertrauenspunkt stehen viele verschiedene Befehle zur Verfügung (in diesem Dokument sind zu viele im Detail beschrieben). Weitere Beispiele finden Sie im Beispiel "Vertrauenspunkt" und in der nachfolgenden Tabelle:

```
crypto pki trustpoint labTrustpoint
 enrollment terminal pem
 serial-number none
 fqdn none
 ip-address none
 subject-name cn=router.example.cisco.com
 subject-alt-name myrouter.example.cisco.com
 revocation-check none
 rsakeypair rsaKey
 hash sha256
```

Command	Beschreibung
crypto pki trustpoint labTrustpoint	Für Menschen lesbares Konfigurationslabel für diesen Vertrauenspunkt. Wird verwendet, um in späteren Befehlen eine Verknüpfung zu Funktionen

	oder Diensten herzustellen.
Anmeldeterminal Pem	Bestimmt, welche Aktion der Befehl **crypto pki enroll** ausführt. In diesem Beispiel gibt **das Registrierungsterminal pem** an, dass die Zertifikatssignierungsanforderung (Certificate Signing Request, CSR) in einem mit Base64 PEM formatierten Text an das Terminal ausgegeben wird. Andere Optionen, wie z. B. die **selbstsignierte Registrierung,** können zum Erstellen eines selbstsignierten Zertifikats verwendet werden. Die **Registrierungs-URL** kann so konfiguriert werden, dass sie eine HTTP-URL definiert und das Simple Certificate Enrollment Protocol (SCEP)-Protokoll verwendet. Beide Methoden sind nicht Gegenstand dieses Dokuments.
Seriennummer keine	Legt fest, ob die seriellen IOS XE-Geräte dem CSR hinzugefügt werden. Dadurch wird auch die Eingabeaufforderung während des Befehls crypto pki enroll deaktiviert.
FQDN keine	Bestimmt, ob der vollqualifizierte Domänenname (FQDN) zum CSR hinzugefügt wird. Dadurch wird auch die Eingabeaufforderung während des Befehls crypto pki enroll deaktiviert.
IP-Adresse keine	Legt fest, ob die IP-Adresse des IOS XE-Geräts zum CSR hinzugefügt wird. Dadurch wird auch die Eingabeaufforderung während des Befehls crypto pki enroll deaktiviert.
subject-name cn=router.example.cisco.com	Zeigt das formatierte X500 an, das dem CSR hinzugefügt wird.
subject-alt-name myrouter.example.cisco.com	Ab IOS XE 17.9.1 kann eine kommagetrennte Liste mit SAN-Werten (Subject Alternate Name) zum CSR hinzugefügt werden.
revocation-check none	Gibt an, wie das IOS XE-Gerät die Gültigkeit des Zertifikats überprüfen soll. Optionen wie die Zertifikatsperrliste (Certificate Revocation List, CRL) und das Online Certificate Status Protocol (OCSP) können verwendet werden, wenn sie von der gewünschten Zertifizierungsstelle unterstützt werden. Dies wird hauptsächlich verwendet, wenn der Vertrauenspunkt von einer anderen konfigurierten IOS XE-Funktion oder einem anderen konfigurierten IOS XE-Dienst verwendet wird. Der Sperrstatus wird auch überprüft, wenn ein Zertifikat mit einem Vertrauenspunkt authentifiziert wird.

rsakeypair rsaKey	Weist den Befehl an, das RSA-Schlüsselpaar mit diesem spezifischen Label zu verwenden. Verwenden Sie für ECDSA-Zertifikate den Befehl "eckeypair ecKey", der auf das Label des EC-Schlüssels verweist
Hash sha256	Dieser Befehl beeinflusst die Art des zu verwendenden Hash-Algorithmus. Die Optionen sind SHA1, SHA256, SHA384 und SHA512.

crypto pki registrieren

Der Befehl **crypto pki enroll** wird verwendet, um den Befehl enrollment für einen bestimmten Vertrauenspunkt auszulösen. (Befehlssyntax)
Für das Beispiel trustpoint, das zuvor angezeigt wurde, zeigt der Befehl **crypto pki enroll labTrustpoint** die Zertifikatssignierungsanforderung (Certificate Signing Request, CSR) an das Terminal im Base64 PEM-Textformat an, wie im Beispiel unten gezeigt.

Diese Zertifikatssignierungsanforderung kann jetzt in einer Textdatei gespeichert oder kopiert und über die Befehlszeile eingefügt werden, um sie für die Validierung und Signierung an eine beliebige Drittanbieter-Zertifizierungsstelle bereitzustellen.

<#root>

```
Router(config)#
```

crypto pki enroll labTrustpoint

```
% Start certificate enrollment ..

% The subject name in the certificate will include: cn=router.example.cisco.com
% The fully-qualified domain name will not be included in the certificate
Display Certificate Request to terminal? [yes/no]:
```

yes

```
Certificate Request follows:

-----BEGIN CERTIFICATE REQUEST-----
MIICrTCCAZUCAQAwIzEhMB8GA1UEAxMYcm91dGVyLmV4YW1wbGUuY2lzY28uY29t
[..truncated..]
mGvBGUpn+cDIdFcNVzn8LQk=
-----END CERTIFICATE REQUEST-----

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

crypto pki authentifizieren

Der Befehl **crypto pki authentication** wird verwendet, um einem bestimmten Vertrauenspunkt ein vertrauenswürdiges Zertifizierungsstellen-Zertifikat hinzuzufügen. Jeder Vertrauenspunkt kann einmal authentifiziert werden. Das heißt, ein Vertrauenspunkt kann nur einen einzelnen Zertifizierungsstellen-Stamm oder ein Zwischenzertifikat enthalten. Wenn Sie den Befehl ein zweites Mal ausführen und ein neues Zertifikat hinzufügen, wird das erste Zertifikat überschrieben.

Bei konfiguriertem Befehls-**Registrierungs-Terminal "pem"** fordert der Befehl **crypto pki Authenticate** den Router auf, ein mit Base64 PEM formatiertes Zertifikat hochzuladen, das über die CLI hochgeladen werden soll. (Befehlssyntax)

Ein Administrator kann einen Vertrauenspunkt authentifizieren, um die Stamm- und optionalen Zwischenzertifikate in einer Zertifikatskette hinzuzufügen, damit das ID-Zertifikat eines Geräts später importiert werden kann.

Administratoren können auch einen Vertrauenspunkt authentifizieren, um dem IOS XE-Gerät andere vertrauenswürdige Stammzertifizierungsstellen hinzuzufügen, um während Protokoll-Handshakes mit diesem Peer-Gerät Vertrauensbeziehungen zu Peer-Geräten zu ermöglichen.

Zur weiteren Veranschaulichung kann ein Peer-Gerät über eine Zertifikatskette verfügen, die von "Root CA 1" signiert ist. Damit die Zertifikatsvalidierung während des Protokoll-Handshakes zwischen dem IOS XE-Gerät und dem Peer-Gerät erfolgreich ist, kann ein Administrator das CA-Zertifikat mit dem Befehl **crypto pki Authenticate** einem Vertrauenspunkt auf dem IOS XE-Gerät hinzufügen.

Das wichtigste Element: Authentifizieren von Vertrauenspunkten mithilfe von crypto pki Authenticate dient immer dazu, einem Vertrauenspunkt CA-Root- oder Zwischenzertifikate hinzuzufügen, nicht dazu, Identitätszertifikate hinzuzufügen. Beachten Sie, dass dieses Konzept auch auf die Authentifizierung selbstsignierter Zertifikate von einem anderen Peer-Gerät angewendet wird.

Das folgende Beispiel zeigt, wie Sie einen Vertrauenspunkt von zuvor mit dem Befehl **crypto pki Authenticate** authentifizieren:

<#root>

```
Router(config)#
```

```
crypto pki authenticate labTrustpoint
```

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

-----BEGIN CERTIFICATE-----
[..truncated..]
-----END CERTIFICATE-----

Certificate has the following attributes:
       Fingerprint MD5: C955FC74 7AABC184 D8A75DE7 3C9E7218
      Fingerprint SHA1: 3A99FF61 1E9E6C7B D0E567A9 96D882F5 2279C534

% Do you accept this certificate? [yes/no]:
```

```
yes
```

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

crypto pki import

Dieser Befehl wird verwendet, um das Identitätszertifikat (ID-Zertifikat) in einen Vertrauenspunkt zu importieren. Ein einzelner Vertrauenspunkt kann nur ein einzelnes ID-Zertifikat enthalten. Wenn Sie den Befehl ein zweites Mal ausführen, werden Sie aufgefordert, das zuvor importierte Zertifikat zu überschreiben. (Befehlssyntax)

Im folgenden Beispiel wird veranschaulicht, wie ein Identitätszertifikat mithilfe des Befehls **crypto pki import** in den Beispielvertrauenspunkt von zuvor importiert wird.

<#root>

```
Router(config)#
```

```
crypto pki import labTrustpoint certificate
```

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

-----BEGIN CERTIFICATE-----
[..truncated..]
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

Ein Administrator erhält einen Fehler, wenn er versucht, ein Zertifikat zu importieren, bevor der Vertrauenspunkt das Zertifizierungsstellenzertifikat authentifiziert hat, mit dem dieses Zertifikat direkt signiert wird.

<#root>

```
Router(config)#
```

```
crypto pki import labTrustpoint certificate
```

```
% You must authenticate the Certificate Authority before
 you can import the router's certificate.
```

Authentifizieren von Peer-Zertifizierungsstellenzertifikaten

Zertifikate der Peer-Zertifizierungsstelle werden IOS XE auf dieselbe Weise hinzugefügt wie Zertifizierungsstellenzertifikate. Das heißt, sie werden mithilfe des Befehls **crypto pki Authenticate** anhand eines Vertrauenspunkts authentifiziert.

Der folgende Befehl zeigt, wie Sie einen Vertrauenspunkt erstellen und ein CA-Zertifikat eines Peer-Drittanbieters authentifizieren.

1. Erstellen Sie zunächst einen Vertrauenspunkt mit einem beschreibenden Namen, der das Zertifikat der Peer-Zertifizierungsstelle enthält.
2. konfigurieren Sie **das Anmeldeterminal pem** so, dass der Befehl crypto pki authentication das Zertifikat über die Befehlszeile anfordert.
3. Konfigurieren Sie **die Sperrüberprüfung 'Keine'**, um die CRL-/OCSP-Prüfung während des Importvorgangs zu überspringen.
4. Authentifizierung des Vertrauenspunkts und Bereitstellung des Zertifikats
5. Wiederholen Sie die Schritte 1-4 für wie für Peer-Zertifizierungsstellenzertifikate erforderlich (denken Sie nur an ein Zertifizierungsstellenzertifikat pro Vertrauenspunkt!)

<#root>

```
Router(config)#
crypto pki trustpoint PEER-ROOT
Router(ca-trustpoint)#
enrollment terminal pem
Router(ca-trustpoint)#
revocation-check none


Router(ca-trustpoint)#
crypto pki authenticate PEER-ROOT


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

-----BEGIN CERTIFICATE-----
[..truncated..]
-----END CERTIFICATE-----

Certificate has the following attributes:
       Fingerprint MD5: 62D1381E 3E03D06A 912BAC4D 247EEF17
      Fingerprint SHA1: 3C97CBB4 491FC8D6 3D12B489 0C285481 64198EDB

% Do you accept this certificate? [yes/no]:
yes


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

Authentifizieren eines oder mehrerer Zwischenzertifikate

In den vorherigen Beispielen wird erläutert, wie Sie einen CSR mit **crypto pki enroll** generieren, das Stammzertifikat der Zertifizierungsstelle mit **crypto pki Authenticate** authentifizieren und dann das Identitätszertifikat mit **crypto pki import** importieren.
Bei der Einführung von Zwischenzertifikaten unterscheidet sich das Verfahren jedoch geringfügig. Keine Angst, es gelten immer noch die gleichen Konzepte und Befehle! Der Unterschied liegt in der Art und Weise, wie die Trustpoints, die die Zertifikate besitzen, ausgelegt sind.

Denken Sie daran, dass jeder Vertrauenspunkt nur ein einzelnes Root- oder Intermediate-Zertifizierungsstellenzertifikat enthalten kann. In einem Beispiel mit einer CA-Kette, wie unten gezeigt, ist es nicht möglich, mit dem Befehl crypto pki Authenticate mehr als ein CA-Zertifikat hinzuzufügen:

<#root>

```
- Root CA


 - Intermediate CA 1
```

- Identity Certificate

Lösung:

1. Erstellen Sie einen Vertrauenspunkt, der die authentifizierte Stammzertifizierungsstelle enthält.
2. Dann authentifizieren Sie das Zwischenzertifikat mit dem Vertrauenspunkt, der zum Erstellen des CSR verwendet wird.
3. Importieren Sie anschließend das Identitätszertifikat in den endgültigen Vertrauenspunkt.

Anhand der Tabelle unten kann man das Zertifikat veranschaulichen, das man zur Vertrauensstellung der Zuordnung mit Farben verwenden soll, die der vorherigen Kette entsprechen, um die Visualisierung zu unterstützen.

Zertifikatsname	Vertrauenswürdiger Punkt	Zu verwendender Befehl
Stamm-CA	crypto pki trustpoint **ROOT-CA**	crypto pki authentifizieren **ROOT-CA**
Zwischengeschaltete Zertifizierungsstelle 1	crypto pki trustpoint **labTrustpoint**	crypto pki authentifizieren **labTrustpoint**
Identitätszertifikat	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint**-Zertifikat

Dieselbe Logik kann auf eine Zertifikatkette mit zwei zwischengeschalteten Zertifizierungsstellenzertifikaten angewendet werden. Auch hier werden Farben bereitgestellt, die die Anwendung der neuen Zwischen-CA auf die IOS XE-Konfiguration veranschaulichen.

<#root>

- Root CA

- Intermediate CA 1

- Intermediate CA 2

- Identity Certificate

Zertifikatsname	Vertrauenswürdiger Punkt	Zu verwendender Befehl
Stamm-CA	crypto pki trustpoint **ROOT-CA**	crypto pki authentifizieren **ROOT-CA**
Zwischengeschaltete Zertifizierungsstelle 1	crypto pki trustpoint **INTER-CA**	crypto pki authentifizieren **INTERCA**
Zwischengeschaltete CA 2	crypto pki trustpoint **labTrustpoint**	crypto pki authentifizieren **labTrustpoint**
Identitätszertifikat	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint**-Zertifikat

Bei näherem Hinsehen sind zwei Muster zu erkennen:

1. Alle Root- oder Intermediate-Zertifikate werden mithilfe von **crypto pki Authenticate (**unabhängig von der Anzahl der Zertifikate) in Trustpoints geladen.
2. Man kann auch bemerken, dass das endgültige Zertifikat vor dem Identitätszertifikat des Geräts (lesen Sie das, das das Identitätszertifikat direkt signiert hat) immer an demselben Vertrauenspunkt authentifiziert wird, an dem das Identitätszertifikat importiert werden soll.
    - Ähnlich wie bei dem zuvor angezeigten Fehler kann ein Administrator mit IOS XE kein Zertifikat importieren, ohne zuerst das CA-Zertifikat zu authentifizieren, das zum direkten Signieren dieses Zertifikats verwendet wird.

Diese beiden Muster können für eine beliebige Anzahl von Zwischenzertifikaten verwendet werden, die zwei übersteigen. In den meisten Bereitstellungen sieht der Administrator jedoch wahrscheinlich mehr als zwei Zwischenzertifikate in einer Zertifikatskette.

Der Vollständigkeit halber wird auch die folgende Stamm-/Identitätszertifikatstabelle bereitgestellt:

<#root>

```
- Root CA
```

```
 - Identity Certificate
```

Zertifikatsname	Vertrauenswürdiger Punkt	Zu verwendender Befehl
Stamm-CA	crypto pki trustpoint **labTrustpoint**	crypto pki authentifizieren **labTrustpoint**
Identitätszertifikat	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint**-Zertifikat

Verifizierung

- Während des Authentifizierungs- oder Importvorgangs führt IOS XE verschiedene Integritätsprüfungen durch, um sicherzustellen, dass das Zertifikat gültig und fehlerfrei ist. Diese Fehler werden auf dem Bildschirm ausgegeben oder Protokolle (show logging) suchen nach Zeilen, die mit "CRYPTO_PKI" beginnen

Nachfolgend finden Sie einige gängige Beispiele:

Gültige Prüfungen vor und nach dem Start werden basierend auf der konfigurierten Zeit im Vergleich zur im Zertifikat angegebenen Zeit durchgeführt

<#root>

```
004458:
```

```
Aug  9
```

```
 21:05:34.403: CRYPTO_PKI: trustpoint labTrustpoint authentication status = 0
```

```
%CRYPTO_PKI: Cert not yet valid or is expired -
```

```
    start date: 05:54:04 EDT
```

Aug 29

```
 2019
    end   date: 05:54:04 EDT Aug 28 2022
```

Wenn die Sperrprüfung nicht deaktiviert ist, führt IOS XE eine Sperrprüfung mithilfe der konfigurierten Methode durch, bevor das Zertifikat importiert wird.

<#root>

```
003375: Aug  9 20:24:14:
```

%PKI-3-CRL_FETCH_FAIL: CRL fetch for trustpoint ROOT failed

```
003376: Aug  9 20:24:14.121:
```

CRYPTO_PKI: enrollment url not configured

Verwenden Sie die folgenden Befehle, um Details zur Konfiguration, Authentifizierung oder Import von Vertrauenspunkten anzuzeigen:

```
show crypto pki trustpoints trustpoint_name
show crypto pki certificates trustpoint_name
show crypto pki certificates verbose trustpoint_name
```

Fehlerbehebung

Beim Debuggen von Importproblemen oder anderen PKI-Problemen verwenden Sie die folgenden Debugs.

```
debug crypto pki messages
debug crypto pki transactions
debug crypto pki validation
debug crypto pki api
debug crypto pki callback
!
debug ssl openssl error
debug ssl openssl msg
debug ssl openssl states
debug ssl openssl ext
```

Erweiterte IOS PKI-Konzepte

Importieren eines Zertifikats im PKCS12-Format

Einige Zertifizierungsstellenanbieter stellen möglicherweise Dateien im PKCS#12-Format (.pfx, .p12)

bereit.

PKCS#12 ist ein spezielles Zertifikatformat, bei dem die gesamte Zertifikatskette vom Stammzertifikat bis zum Identitätszertifikat mit dem RSA-Schlüsselpaar gebündelt wird.

Dieses Format ist sehr praktisch für den Import mit IOS XE und kann einfach mit dem folgenden Befehl importiert werden:

```
<#root>

Router(config)#

crypto pki import PKCS12-TP pkcs12 terminal password Cisco123


or
Router(config)#

crypto pki import PKCS12-TP pkcs12 ftp://cisco:cisco@192.168.1.1/certificate.pfx password Cisco123

% Importing pkcs12...
Address or name of remote host [192.168.1.1]?
Source filename [certificate.pfx]?
Reading file from ftp://cisco@192.168.1.1/certificate.pfx!
[OK - 2389/4096 bytes]
% You already have RSA keys named PKCS12.
% If you replace them, all router certs issued using these keys
% will be removed.
% Do you really want to replace them? [yes/no]:

yes

CRYPTO_PKI: Imported PKCS12 file successfully.
```

PKCS12- oder PEM-Zertifikate exportieren

Ein Administrator kann Zertifikate als unverschlüsseltes Base64-PEM, verschlüsseltes Base64-PEM oder PKCS12-Format in das Terminal exportieren, um sie in andere Peer-Geräte zu importieren.

Dies ist nützlich, wenn neue Peer-Geräte aktiviert werden und ein Administrator ein Root-Zertifizierungsstellenzertifikat freigeben muss, das das Geräte-Identitätszertifikat signiert hat.

Nachfolgend finden Sie eine Beispielsyntax:

```
<#root>

Router(config)#

crypto pki export labTrustpoint pem terminal


Router(config)#

crypto pki export labTrustpoint pem terminal 3des password Cisco!123

Router(config)#

crypto pki export labTrustpoint pkcs12 terminal password cisco!123
```

RSA-Schlüssel exportieren

Möglicherweise müssen RSA-Schlüssel exportiert werden, um sie auf ein anderes Gerät zu importieren oder um sie für die Fehlerbehebung zu verwenden. Wenn das Schlüsselpaar als exportierbar erstellt wurde, können die Schlüssel mit dem Befehl crypto key export zusammen mit einer Verschlüsselungsmethode (DES, 3DES, AES) und einem Kennwort exportiert werden.

Anwendungsbeispiel:

<#root>

```
Router(config)#
```

```
crypto key export rsa rsaKey pem terminal aes Cisco!123
```

```
% Key name: IOS-VG
   Usage: General Purpose Key
   Key data:
-----BEGIN PUBLIC KEY-----
[..truncated..]
-----END PUBLIC KEY-----

base64 len 1664-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,40E087AFF0886DA7C468D2084A0DECFB

[..truncated..]
-----END RSA PRIVATE KEY-----
```

Wenn der Schlüssel nicht exportierbar ist, wird ein Fehler angezeigt.

<#root>

```
Router(config)#
```

```
crypto key export rsa kydavis.cisco.com pem terminal 3des mySecretPassword
```

```
% RSA keypair kydavis.cisco.com' is not exportable.
```

Generierte RSA-Schlüssel extern importieren

Einige Administratoren können RSA- und Zertifikaterstellung offline durchführen. Es ist möglich, die RSA-Schlüssel mit dem Befehl **crypto key import (Schlüsselimport) zu** importieren, wie unten gezeigt, und das Kennwort zu verwenden.

<#root>

```
Router(config)#
```

```
crypto key import rsa rsaKey general-purpose exportable terminal mySecretPassword
```

```
% Enter PEM-formatted public General Purpose key or certificate.
% End with a blank line or "quit" on a line by itself.
-----BEGIN PUBLIC KEY-----
[..truncated..]
-----END PUBLIC KEY-----

% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,9E31AAD9B7463502
[..truncated..]
-----END RSA PRIVATE KEY-----
quit
% Key pair import succeeded.
```

RSA-Schlüssel löschen

Verwenden Sie den Befehl **crypto key zeroize rsa rsaKey,** um eine RSA-Tastatur mit dem Namen rsaKey zu löschen.

Cisco Trusted CA-Paket über Trustpool importieren

TrustPools unterscheiden sich geringfügig von einem TrustPoint, aber die Kernverwendung ist dieselbe. Wenn Trustpoints in der Regel ein einzelnes CA-Zertifikat enthalten, enthält ein Trustpool eine Reihe von vertrauenswürdigen CAs.

Cisco veröffentlicht CA-Pakete unter https://www.cisco.com/security/pki/

Eine gängige Verwendung ist das Herunterladen der Datei ios_core.p7b mithilfe des folgenden Befehls:

<#root>

```
Router(config)#

crypto pki trustpool import clean url http://www.cisco.com/security/pki/trs/ios_core.p7b

Reading file from http://www.cisco.com/security/pki/trs/ios_core.p7b
Loading http://www.cisco.com/security/pki/trs/ios_core.p7b
% PEM files import succeeded.
Router(config)#
```

Häufig gestellte Fragen

Wird durch das Löschen eines Vertrauenspunkts die CSR-Anfrage oder eine Zertifikatskette ungültig, die von einer bestimmten CSR-Anfrage erteilt wurde?

Nein. Sobald der CSR generiert und gespeichert wurde, kann der Vertrauenspunkt gelöscht und erneut hinzugefügt werden, ohne den CSR zu ungültig zu machen.

Dies wird häufig vom technischen Support von Cisco verwendet, um neu zu starten, wenn die Authentifizierung/der Import von Zertifikaten fehlschlug.

Solange der Administrator oder der Support-Techniker keine RSA-Schlüssel neu generieren, können der CSR oder die signierte Zertifikatskette importiert und authentifiziert/importiert werden.

Wichtig! Durch Entfernen des Vertrauenspunkts **werden** alle authentifizierten/importierten Zertifikate gelöscht, die problematischer sein könnten, wenn diese Zertifikate derzeit von einem Dienst oder einer Funktion verwendet werden.

Wird durch das Generieren einer CSR-Anfrage an einem Vertrauenspunkt das vorhandene Zertifikat ungültig?

Nein, dies ist häufig der Fall, wenn Zertifikate bald ablaufen. Ein Administrator kann einen **crypto pki enroll**-Befehl ausführen, um einen neuen CSR zu erstellen und den Zertifikatsignierungsprozess mit einer Zertifizierungsstelle zu starten, während die vorhandenen Zertifikate, die authentifiziert/importiert wurden, weiter verwendet werden. Der Zeitpunkt, an dem ein Administrator die Zertifikate durch **crypto pki Authenticate/**crypto **pki import** ersetzt, ist der Zeitpunkt, an dem die alten Zertifikate ersetzt werden.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.