Configurare i certificati firmati dalla CA con PKI IOS XE

Sommario

[Introduzione](#)
[Prerequisiti](#)
[Requisiti](#)
[Componenti usati](#)
[Premesse](#)
[Configurazione PKI IOS XE](#)
[generazione chiave crittografica](#)
[trust point crypto pki](#)
[registrazione pki crittografica](#)
[autenticazione crypto pki](#)
[importazione crypto pki](#)
[Autenticazione dei certificati delle CA peer](#)
[Autenticazione di uno o più certificati intermedi](#)
[Verifica](#)
[Risoluzione dei problemi](#)
[Concetti avanzati di IOS PKI](#)
[Importazione di un certificato formattato PKCS12](#)
[Esportazione di certificati PKCS12 o PEM](#)
[Esportazione delle chiavi RSA](#)
[Importa chiavi RSA generate off-box](#)
[Elimina chiavi RSA](#)
[Domande frequenti](#)
[L'eliminazione di un trust point invalida il CSR o una catena di certificati concessi da un determinato CSR?](#)
[La generazione di un CSR in un trust point invaliderà il certificato esistente?](#)

Introduzione

Questo documento funge da guida generale per la configurazione dei certificati IOS XE firmati da un'Autorità di certificazione (CA) di terze parti.

In questo documento viene descritto in dettaglio come importare una catena con firma CA multilivello e come utilizzare il dispositivo come certificato di identità (ID), nonché come importare altri certificati di terze parti ai fini della convalida del certificato.

Prerequisiti

Requisiti

NTP e ora orologio **DEVONO** essere configurati quando si utilizzano le funzionalità PKI di IOS.

Se un amministratore non configura NTP, è possibile che si verifichino problemi con la generazione di un certificato con data/ora futura/passata. Questa distorsione della data o dell'ora può causare problemi durante l'importazione e altri problemi.

Esempio di configurazione NTP:

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

Componenti usati

- Router Cisco con Cisco IOS® XE17.11.1a

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Alcune funzionalità descritte in questo documento potrebbero non essere disponibili nelle versioni precedenti di IOS XE. Nei casi in cui è possibile, è stata prestata attenzione nel documentare quando un comando o una funzionalità è stata introdotta o modificata.

Fare sempre riferimento alla documentazione ufficiale per le funzionalità PKI di IOS XE per una determinata versione per comprendere eventuali limitazioni o modifiche che possono essere rilevanti per la propria versione specifica:

Esempi:

- 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

Configurazione PKI IOS XE

Quando si utilizzano i certificati PKI IOS XE, l'amministratore deve eseguire le azioni seguenti a un livello superiore:

1. Crea una chiave da utilizzare con una funzionalità o un servizio (**generazione chiave di crittografia**)
2. Configurare un trust point con vari parametri e collegare la chiave. (**trust crypto pki**)
3. Genera una richiesta di firma del certificato (CSR) (**registrazione PKI crittografica**)
4. Fornire il CSR a una CA per la firma (*non illustrato in questo documento*)
5. Autenticare i certificati CA radice e/o intermedi (**autenticazione crypto-pki**)
6. Importa i certificati del dispositivo (**importazione PKI crittografica**)
7. Facoltativo: autenticare i certificati CA peer (**autenticazione PKI crittografica**)

Questi passaggi sono descritti in dettaglio nelle sezioni successive raggruppate in base ai comandi necessari per l'azione specificata.

generazione chiave crittografica

Molti amministratori hanno immesso questo comando per abilitare Secure Socket Shell (SSH) su un router o come parte di una guida alla configurazione di una funzionalità. Tuttavia, sono pochi quelli che non hanno sezionato ciò che il comando effettivamente fa.

Prendete ad esempio i seguenti comandi:

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

La disattivazione di questi comandi nelle parti specifiche descriverà in dettaglio l'utilizzo:

- La prima parte del comando in nero (crypto key generate) indica al router che verrà creata una nuova chiave. Sono disponibili altre opzioni, ad esempio l'esportazione della chiave crittografica, l'importazione della chiave crittografica o la riduzione del valore zero della chiave crittografica, che verranno illustrate in dettaglio in seguito.
- La parte successiva del comando in **verde** (rsa general-keys, ec) indica al router esattamente il tipo di chiave che stiamo creando. Per la maggior parte degli scopi verrà utilizzata una coppia di chiavi Rivest-Shamir-Adleman (RSA) costituita da una chiave pubblica/privata, ma un amministratore può anche configurare la curva ellittica (EC) per l'utilizzo con funzioni quali quelle che richiedono certificati ECDSA o per l'utilizzo con handshake ECDHE.
- Il comando in **arancione** definisce le dimensioni della chiave.
    - Per RSA il modulo è la terminologia e valori come 512-4096 sono opzioni disponibili. Le dimensioni predefinite del modulo variano a seconda della versione, ma si consiglia di seguire le best practice di Cisco per la crittografia di nuova generazione e utilizzare chiavi superiori a 2048.
    - Per EC, il comando key-size è necessario per specificare il numero di bit nella chiave. Le opzioni sono 256, 384 o 512.
- Il comando in **viola** definisce l'etichetta per questo tasto. Questa operazione è importante perché un amministratore potrebbe dover definire più chiavi per vari scopi sullo stesso dispositivo IOS XE. L'etichetta viene utilizzata per specificare la chiave esatta da utilizzare con una determinata feature. Ove possibile, utilizzare sempre un'etichetta per distinguere i tasti in uso e semplificare notevolmente l'assegnazione dei tasti alle caratteristiche. Ad esempio: label SSH, label CUBE, label HTTPS creerà due chiavi da utilizzare con servizi o funzionalità diversi.
    - L'etichetta predefinita per una chiave è il device hostname.domain. Alcuni dispositivi possono generare le chiavi RSA al primo avvio. Se non si immette un post-correzione per l'etichetta, l'amministratore potrebbe inavvertitamente sovrascrivere/rigenerare la chiave errata
- Il comando finale in **blu** è il suffisso esportabile. Con questo comando viene specificato dettagliatamente che la chiave può essere utilizzata con il comando **crypto pki export** per l'esportazione e l'uso con altri sistemi. Un esempio potrebbe essere l'importazione in un dispositivo peer ad alta disponibilità in modo che una singola chiave venga utilizzata da entrambi i membri di una coppia HA o per l'uso in strumenti di risoluzione dei problemi come Wireshark per decrittografare sessioni TLS basate su RSA. Qualunque sia la ragione, bisogna dire che le chiavi RSA possono essere create solo come esportabili dall'inizio. Se un amministratore crea una chiave RSA non esportabile, questa chiave non può essere impostata come esportabile senza rigenerare la chiave, il che può avere effetti sulla frammentazione per altre funzionalità, ad esempio l'invalidazione di tutti i certificati creati utilizzando quella chiave. Ciò detto, una chiave esportabile può essere convertita in non esportabile senza rigenerare la chiave utilizzando il comando **crypto key move rsaKeyLabel non-exportable**

Esempi di configurazione:

<#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
```

Esempi di verifica:

<#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
```

trust point crypto pki

I trust point sono un concetto simile a una cartella per l'archiviazione e la gestione dei certificati PKI in IOS XE. (Sintassi dei comandi)

Ad alto livello:

1. Ogni trust point IOS XE può contenere un singolo certificato CA radice o intermedio caricato tramite il comando **crypto pki authentication**. I trust autenticati possono essere paragonati all'aggiunta di certificati ora considerati attendibili dal dispositivo.
2. Ogni trust point IOS XE può inoltre importare un singolo certificato di identità (ID) caricato tramite il comando **crypto pki import**. Il certificato di identificazione è il certificato di periferica generalmente associato a un servizio o a una funzionalità.
3. Un amministratore può utilizzare il comando **authentication** and **import** sullo stesso trust point, necessario per importare un certificato ID descritto più avanti. Quando si utilizza il flusso di lavoro di autenticazione/importazione, il trust point conterrà due certificati (root/intermediate + certificato di identità).
4. Quando i trust point vengono utilizzati per l'archiviazione di certificati CA radice/intermedia peer attendibili, solo il **autenticazione crypto pki** è obbligatorio. In questo scenario un trust point conterrà solo il singolo certificato autenticato dall'amministratore.

Nota: le sezioni successive relative all'**autenticazione** e all'**importazione della chiave PKI crittografica** e quelle successive che contengono esempi di autenticazione/importazione per i certificati multilivello forniranno un ulteriore contesto per questi quattro punti elenco.

È possibile configurare vari comandi per i punti di trust. Questi comandi possono essere utilizzati per influenzare i valori all'interno di una richiesta di firma di un certificato (CSR) creata dal dispositivo utilizzando il comando **crypto pki enroll** su un trust point.

Sono disponibili molti comandi diversi per un trust point (troppi per essere dettagliati in questo documento), ma alcuni esempi più comuni sono illustrati in dettaglio sia nel trustpoint di esempio che nella tabella riportata di seguito:

```
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
```

Comando	**Descrizione**
crypto pki trustpoint labTrustpoint	Etichetta di configurazione leggibile dall'utente per questo trust point. Utilizzato per il collegamento a funzionalità o servizi in comandi successivi.
pem terminale di registrazione	Determina l'azione che verrà eseguita dal comando **crypto pki enroll**. In questo esempio, la **PEM del terminale di registrazione** indica che la richiesta di firma del certificato (CSR) verrà inviata al terminale

	in un testo in formato PEM Base64. È possibile utilizzare altre opzioni, ad esempio la **registrazione autofirmata**, per creare un certificato autofirmato oppure **configurare l'URL di registrazione** in modo da definire un URL HTTP e utilizzare il protocollo SCEP (Simple Certificate Enrollment Protocol). Entrambi i metodi non rientrano nell'ambito del presente documento.
serial-number none	Determina se il numero di serie dei dispositivi IOS XE verrà aggiunto al CSR. In questo modo, viene disabilitato anche il prompt durante il comando crypto pki enroll.
fqdn nessuno	Determina se il nome di dominio completo (FQDN) verrà aggiunto al CSR. In questo modo, viene disabilitato anche il prompt durante il comando crypto pki enroll.
indirizzo-ip nessuno	Determina se l'indirizzo IP dei dispositivi IOS XE verrà aggiunto al CSR. In questo modo, viene disabilitato anche il prompt durante il comando crypto pki enroll.
subject-name cn=router.example.cisco.com	Indica il modello X500 formattato che verrà aggiunto al CSR.
subject-alt-name myrouter.example.cisco.com	A partire da IOS XE 17.9.1, è possibile aggiungere al CSR un elenco separato da virgole di valori SAN (Subject Alternate Name).
revocation-check none	Indica il modo in cui il dispositivo IOS XE deve verificare la validità del certificato. È possibile utilizzare opzioni quali CRL (Certificate Revocation List), OCSP (Online Certificate Status Protocol), se supportate dall'Autorità di certificazione scelta. Questa opzione viene utilizzata principalmente quando il trust point viene utilizzato da un'altra funzionalità o servizio IOS XE configurato. Lo stato di revoca viene controllato anche quando un certificato viene autenticato con un trust point.
rsaKey rsakeypair	Indica al comando di utilizzare la coppia di chiavi RSA con questa etichetta specifica. Per i certificati ECDSA utilizzare il comando "eckeypair ecKey" che fa riferimento all'etichetta della chiave EC
hash sha256	Questo comando influenza il tipo di algoritmo di hash da utilizzare. Le opzioni sono SHA1, SHA256, SHA384, SHA512

registrazione pki crittografica

Il comando **crypto pki enroll** viene utilizzato per attivare il comando enrollment su un determinato trust point. (Sintassi dei comandi)
Per l'esempio di trustpoint visualizzato in precedenza, il comando **crypto pki enroll labTrustpoint** visualizzerà la richiesta di firma del certificato (CSR) sul terminale in formato testo Base64 PEM, come mostrato nell'esempio seguente.

È ora possibile salvare la richiesta di firma del certificato in un file di testo o copiarla e incollarla dalla riga di comando allo scopo di consentire a qualsiasi autorità di certificazione di terze parti di convalidarla e firmarla.

<#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---
```

autenticazione crypto pki

Il comando **crypto pki authentication** viene utilizzato per aggiungere un certificato CA attendibile a un determinato trust point. Ogni trust point può essere autenticato una sola volta. In altre parole, un trust point può contenere solo un singolo certificato radice o intermedio della CA. Se si esegue nuovamente il comando e si aggiunge un nuovo certificato, il primo certificato verrà sovrascritto.

Con il comando **enrollment terminal pem** configurato, il comando **crypto pki authentication** chiede al router di caricare un certificato in formato PEM Base64 dalla CLI. (Sintassi dei comandi)

Un amministratore può autenticare un trust point per aggiungere i certificati radice e i certificati intermedi facoltativi in una catena di certificati allo scopo di importare il certificato di identificazione di un dispositivo in un secondo momento.

Gli amministratori possono inoltre autenticare un trust point per aggiungere altre CA radice attendibili al dispositivo IOS XE allo scopo di abilitare le relazioni di trust con i dispositivi peer durante gli handshake del protocollo con tale dispositivo peer.

Per ulteriori informazioni, un dispositivo peer può includere una catena di certificati firmata da "Root CA

1". Affinché la convalida del certificato durante l'handshake del protocollo tra il dispositivo IOS XE e il dispositivo peer abbia esito positivo, un amministratore può utilizzare il comando **crypto pki authentication** per aggiungere il certificato CA a un trust point sul dispositivo IOS XE.

Elemento principale da ricordare: l'autenticazione dei trust point tramite l'autenticazione PKI crittografica viene sempre utilizzata per aggiungere certificati radice o intermedi CA a un trust point e non per aggiungere certificati di identità. Si noti che questo concetto viene applicato anche all'autenticazione di certificati autofirmati da un altro dispositivo peer.

Nell'esempio seguente viene illustrato come autenticare un trust point precedente utilizzando il comando **crypto pki authenticate**:

```
<#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
```

importazione crypto pki

Questo comando viene utilizzato per importare il certificato di identità (ID) in un trust point. Un singolo trust point può contenere un solo certificato ID e l'esecuzione del comando una seconda volta richiederà di sovrascrivere il certificato importato in precedenza. (Sintassi dei comandi)

Nell'esempio seguente viene illustrato come importare un certificato di identità nel trust point di esempio da una versione precedente utilizzando il comando **crypto pki import**.

```
<#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
```

Un amministratore riceverà un errore se tenta di importare un certificato prima che il trust point abbia autenticato il certificato CA utilizzato per firmare direttamente il certificato.

<#root>

```
Router(config)#
```

```
crypto pki import labTrustpoint certificate
```

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

Autenticazione dei certificati delle CA peer

I certificati CA peer vengono aggiunti a IOS XE utilizzando lo stesso metodo di aggiunta di qualsiasi certificato CA. In altri termini, vengono autenticati in un trust point utilizzando il comando **crypto pki authentication**.

Il comando seguente mostra come creare un trust point e autenticare un certificato CA di terze parti peer.

1. Creare innanzitutto un trust point con un nome descrittivo che conterrà il certificato CA peer
2. configurare **enrollment terminal pem** in modo che il comando crypto pki authentication richieda il certificato tramite la riga di comando.
3. Configurare **revocation-check none** per ignorare il controllo CRL/OCSP durante il processo di importazione
4. Autentica il trust point e fornisce il certificato
5. Ripetere i passaggi da 1 a 4 per come richiesto per i certificati CA peer (ricordare un solo certificato CA per punto di attendibilità!)

<#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
```

Autenticazione di uno o più certificati intermedi

Negli esempi precedenti viene descritto in dettaglio come generare un CSR utilizzando la **registrazione della chiave PKI crittografica**, autenticare il certificato CA radice utilizzando l'**autenticazione della chiave PKI crittografica** e quindi importare il certificato di identità utilizzando l'**importazione della chiave PKI crittografica**.
Tuttavia, quando si introducono i certificati intermedi, il processo differisce leggermente. Non abbiate paura, gli stessi concetti e comandi sono ancora validi! La differenza sta nella modalità di layout dei trust point che contengono i certificati.

Tenere presente che ogni trust point può contenere un solo certificato CA radice o intermedio. Pertanto, in un esempio in cui è presente una catena di CA come quella illustrata di seguito, è impossibile utilizzare il comando crypto pki authentication per aggiungere più di un certificato CA:

<#root>

- Root CA

- Intermediate CA 1

- Identity Certificate

Soluzione:

1. Creare un trust point che conterrà la CA radice autenticata.
2. Autenticare quindi il certificato intermedio con il trust point utilizzato per creare il CSR
3. Importare infine il certificato di identità nel trust point finale.

Nella tabella seguente è illustrata la mappatura del certificato al comando trust point con i colori che corrispondono alla catena precedente per agevolare la visualizzazione.

Nome certificato	Trustpoint da utilizzare	Comando da utilizzare
CA radice	crypto pki trustpoint **ROOT-CA**	crypto pki autentica **CA RADICE**

CA intermedia 1	crypto pki trustpoint **labTrustpoint**	crypto pki autentica **labTrustpoint**
Certificato di identità	crypto pki trustpoint **labTrustpoint**	certificato **labTrustpoint di** importazione crittografia PKI

È possibile applicare la stessa logica a una catena di certificati con due certificati CA intermedi. Anche in questo caso vengono forniti colori per facilitare la visualizzazione della posizione in cui la nuova CA intermedia viene applicata alla configurazione di IOS XE.

<#root>

```
- Root CA
```

```
 - Intermediate CA 1
```

```
 - Intermediate CA 2
```

```
 - Identity Certificate
```

Nome certificato	**Trustpoint da utilizzare**	**Comando da utilizzare**
CA radice	crypto pki trustpoint **ROOT-CA**	crypto pki autentica **CA RADICE**
CA intermedia 1	crypto pki trustpoint **INTER-CA**	crypto pki autentica **inter-CA**
CA intermedia 2	crypto pki trustpoint **labTrustpoint**	crypto pki autentica **labTrustpoint**
Certificato di identità	crypto pki trustpoint **labTrustpoint**	certificato **labTrustpoint di** importazione crittografia PKI

Guardando da vicino si notano due modelli:

1. Tutti i certificati radice o intermedi vengono caricati nei trust point utilizzando l'**autenticazione PKI crittografica** (indipendentemente dal numero di certificati).
2. È inoltre possibile notare che il certificato finale prima del certificato di identità del dispositivo (leggere quello che ha firmato direttamente il certificato di identità) viene sempre autenticato nello stesso trust in cui deve essere importato il certificato di identità.
    - Analogamente all'errore mostrato in precedenza, IOS XE non consente a un amministratore di importare un certificato senza prima autenticare il certificato CA utilizzato per firmare direttamente il certificato.

Questi due modelli possono essere utilizzati per qualsiasi numero di certificati intermedi oltre i due, anche se nella maggior parte delle distribuzioni un amministratore può vedere più di due CA intermedie in una catena di certificati.

Per completezza, viene fornita anche la seguente tabella di certificati di identità radice:

<#root>

- Root CA

- Identity Certificate

Nome certificato	Trustpoint da utilizzare	Comando da utilizzare
CA radice	crypto pki trustpoint **labTrustpoint**	crypto pki autentica **labTrustpoint**
Certificato di identità	crypto pki trustpoint **labTrustpoint**	certificato **labTrustpoint di** importazione crittografia PKI

Verifica

- Durante il processo di autenticazione o importazione, IOS XE esegue diversi controlli di integrità per verificare che il certificato sia valido e ben formato. Questi errori verranno stampati sullo schermo o nei log (mostra log) verranno cercate le righe che iniziano con "CRYPTO_PKI"

Di seguito sono riportati alcuni esempi comuni:

I controlli validi prima/dopo l'esecuzione vengono eseguiti in base all'ora configurata rispetto a quella rilevata nel certificato

<#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

se il controllo delle revoche non è disabilitato, IOS XE eseguirà un controllo delle revoche tramite il metodo configurato prima di importare il certificato

<#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

Per visualizzare i dettagli relativi alla configurazione del trust point, all'autenticazione o all'importazione, utilizzare i comandi seguenti:

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

Risoluzione dei problemi

Durante il debug di problemi di importazione o di altri problemi PKI, utilizzare i seguenti debug.

```
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
```

Concetti avanzati di IOS PKI

Importazione di un certificato formattato PKCS12

Alcuni provider CA possono fornire file in formato PKCS#12 (.pfx, .p12).

PKCS#12 è un tipo speciale di formato di certificato in cui l'intera catena di certificati, dal certificato radice fino al certificato di identità, è inclusa insieme alla coppia di chiavi rsa.

Questo formato è molto utile per l'importazione con IOS XE e può essere facilmente importato utilizzando il comando seguente:

<#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.
```

Esportazione di certificati PKCS12 o PEM

Un amministratore può esportare i certificati nel terminale in formato PEM in testo normale Base64, testo normale crittografato Base64 o PKCS12 per l'importazione in altri dispositivi peer.

Questa funzionalità risulta utile quando si attivano nuovi dispositivi peer e un amministratore deve condividere un certificato CA radice che ha firmato il certificato di identità dei dispositivi.

Di seguito sono riportati alcuni esempi di sintassi:

<#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

Esportazione delle chiavi RSA

Potrebbe essere necessario esportare le chiavi RSA per importarle in altri dispositivi o utilizzarle per la risoluzione dei problemi. Supponendo che la coppia di chiavi sia stata creata come esportabile, le chiavi possono essere esportate utilizzando il comando crypto key export insieme a un metodo di crittografia (DES, 3DES, AES) e a una password.

Utilizzo di esempio:

<#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-----
```

Se la chiave non è esportabile, verrà visualizzato un errore.

<#root>

```
Router(config)#
```

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

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

Importa chiavi RSA generate off-box

Alcuni amministratori possono eseguire RSA e la creazione dei certificati off-box; è possibile importare le chiavi RSA usando il comando **crypto key import**, come mostrato di seguito, usando la password.

<#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.
```

Elimina chiavi RSA

Usare il comando **crypto key zeroize rsa rsaKey** per eliminare una coppia di chiavi RSA chiamata rsaKey.

Importa pacchetto CA attendibile Cisco tramite trust pool

I pool di trust variano leggermente da un trust point, ma l'utilizzo principale è lo stesso. Se i trust point in

genere contengono un singolo certificato CA, un trust pool conterrà un numero di CA attendibili.

Cisco pubblica i bundle CA sul sito https://www.cisco.com/security/pki/

Un utilizzo comune consiste nel scaricare il file ios_core.p7b utilizzando il comando seguente:

<#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)#
```

Domande frequenti

L'eliminazione di un trust point invalida il CSR o una catena di certificati concessi da un determinato CSR?

No, una volta generato e salvato il CSR, il trust point può essere eliminato e riaggiunto senza invalidare il CSR.

Questa funzione viene spesso utilizzata dal supporto tecnico Cisco per ricominciare da capo quando l'autenticazione/importazione dei certificati non è più corretta.

Se l'amministratore o il tecnico di assistenza non rigenera le chiavi RSA, è possibile importare la catena di certificati firmata o CSR autenticandola o importandola.

Importante! La rimozione del trust point **comporterà l'**eliminazione di tutti i certificati autenticati/importati che potrebbero presentare problemi maggiori se tali certificati sono attualmente utilizzati da un servizio o una funzionalità.

La generazione di un CSR in un trust point invaliderà il certificato esistente?

Risposta errata. Questa condizione è comune quando i certificati sono prossimi alla scadenza. Un amministratore può eseguire un comando **crypto pki enroll** per creare un nuovo CSR e avviare il processo di firma dei certificati con una CA mentre i certificati esistenti autenticati/importati rimangono in uso. Il momento in cui un amministratore sostituisce i certificati con l'**importazione crypto pki authentication**/**crypto pki** è il momento in cui i vecchi certificati vengono sostituiti.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).