Configurar certificados CA assinados com IOS XE PKI

Contents

Introdução

Este documento serve como um guia geral para a configuração de certificados IOS XE assinados por uma Autoridade de Certificação (CA) de terceiros.

Este documento detalhará como importar uma cadeia de CA assinada multinível como para o dispositivo servir como um certificado de identidade (ID), bem como como como importar outros certificados de terceiros para fins de validação de certificado.

Pré-requisitos

Requisitos

O NTP e o Clock time **DEVEM** ser configurados ao utilizar os recursos IOS PKI.

Se um administrador não configurar o NTP, você pode ter problemas com um certificado que está sendo gerado com uma data/hora futura/passada. Essa diferença na data ou na hora pode causar problemas de importação e outros problemas posteriormente.

Exemplo de configuração de NTP:

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

Componentes Utilizados

- Roteador Cisco executandoCisco IOS® XE17.11.1a

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Observe que alguns recursos detalhados neste documento podem não estar disponíveis em versões mais antigas do IOS XE. Onde possível, foi tomado cuidado para documentar quando um comando ou recurso foi introduzido ou modificado.

Consulte sempre a documentação oficial dos recursos IOS XE PKI de uma determinada versão para entender quaisquer limitações ou alterações que possam ser relevantes para sua versão específica:

Examples:

- 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

Configuração PKI do IOS XE

Em um alto nível, um administrador deve executar as seguintes ações ao trabalhar com certificados PKI do IOS XE:

1. Criar uma chave para uso com um recurso ou serviço (**crypto key generate**)
2. Configure um ponto confiável com vários parâmetros e vincule a chave. (**crypto pki trustpoint**)
3. Gerar uma solicitação de assinatura de certificado (CSR) (**crypto pki enroll**)
4. Fornecer o CSR a uma CA para assinatura (*não abordado neste documento*)
5. Autenticar os certificados de CA raiz e/ou intermediários (**crypto pki authenticate**)
6. Importar os certificados do dispositivo (**importação do crypto pki**)
7. Opcional: Autenticar certificados CA de mesmo nível (**crypto pki authenticate**)

Essas etapas são detalhadas nas próximas seções agrupadas pelos comandos necessários para a ação determinada.

crypto key generate

Muitos administradores inseriram esse comando para ativar o Secure Socket Shell (SSH) em um roteador ou como parte de algum guia de configuração para um recurso. No entanto, poucos não dissecaram o que o comando realmente faz.

Tome como exemplo os comandos abaixo:

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

A dissecação desses comandos nas partes específicas detalhará o uso:

- A primeira parte do comando em preto (crypto key generate) instrui o roteador de que criaremos uma nova chave. Há outras opções, como exportação de chave de criptografia, importação de chave de criptografia ou tamanho zero de chave de criptografia, que serão detalhadas posteriormente.
- A próxima parte do comando em **verde** (rsa general-keys, ec) instrui o roteador exatamente que tipo de chave estamos criando. Para a maioria dos propósitos, um par de chaves Rivest-Shamir-Adleman (RSA) consistindo em uma chave pública/privada será usado, mas um administrador também pode configurar a curva elíptica (EC) para uso com recursos como aqueles que exigem certificados ECDSA ou para uso com handshakes ECDHE.
- O comando em **laranja** define o tamanho da nossa chave.
    - Para a RSA, o módulo é a terminologia e os valores entre 512-4096 são opções disponíveis. O tamanho padrão do módulo varia de acordo com a versão, mas é recomendável seguir as práticas recomendadas da Cisco para a criptografia de última geração e utilizar chaves maiores que 2048.
    - Para EC, o comando key-size é necessário para especificar o número de bits na chave. As opções são 256, 384 ou 512.
- O comando em **roxo** define o rótulo para esta chave. Isso é importante porque um administrador pode precisar definir várias chaves para várias finalidades no mesmo dispositivo IOS XE. O rótulo é usado para especificar a chave exata para uso com um determinado recurso. Sempre que possível, use sempre um rótulo para distinguir as teclas em uso e facilitar muito a atribuição de teclas aos recursos. Por exemplo: label SSH, label CUBE, label HTTPS criará duas chaves para uso com diferentes serviços ou recursos.
    - O rótulo padrão para uma chave é o nome do host do dispositivo.domínio. Alguns dispositivos podem gerar chaves RSA na primeira inicialização. Ao não inserir um pós-reparo de rótulo, um administrador pode estar correndo o risco de substituir/regenerar inadvertidamente a chave errada
- O comando final em **azul** é o sufixo exportável. Este comando detalha que a chave pode ser usada com o comando **crypto pki export** para exportação e uso com outros sistemas. Um exemplo pode ser importar para um dispositivo de alta disponibilidade de mesmo nível para que uma única chave seja usada por ambos os membros de um par HA ou para uso em ferramentas de solução de problemas, como o Wireshark, para descriptografar sessões TLS baseadas em RSA. Seja qual for o motivo, deve-se afirmar que as chaves RSA só podem ser criadas como exportáveis desde o início. Se um administrador criar uma chave RSA não exportável, essa chave não poderá ser definida como exportável sem gerar novamente a chave, o que pode ter efeitos de ondulação em outros recursos, como invalidar todos os certificados criados com essa chave. Dito isto, uma chave exportável pode ser desatualizada para não exportável sem gerar novamente a chave usando o comando **crypto key move rsa rsaKeyLabel non-exportable**

Exemplos de configuração:

<#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
```

Exemplos de verificação:

<#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

Os pontos de confiança são um conceito "semelhante a uma pasta" para armazenar e gerenciar certificados PKI no IOS XE. (Sintaxe de Comando)

Em um alto nível:

1. Cada ponto confiável IOS XE pode conter um certificado de CA intermediário ou raiz simples

carregado por meio do comando **crypto pki authenticate**. Pense nos pontos de confiança autenticados como a adição de certificados que agora são confiáveis pelo dispositivo.
2. Cada ponto de confiança IOS XE também pode importar um único certificado de identidade (ID) carregado por meio do comando **crypto pki import**. O certificado de ID é este certificado de dispositivo que geralmente está ligado a algum serviço ou recurso.
3. Um administrador pode usar o comando **authenticate** e **import** no mesmo ponto confiável (que é necessário para importar um certificado de ID discutido posteriormente). Ao usar o fluxo de trabalho de autenticação/importação, o ponto de confiança conterá dois certificados (raiz/intermediário + certificado de identidade).
4. Quando os pontos de confiança são usados para armazenar certificados CA intermediários/raiz de pares confiáveis, somente o **crypto pki authentication** é necessário. Neste cenário, um ponto confiável conterá apenas o único certificado autenticado pelo administrador.

Observação: as próximas seções para **crypto pki authenticate** e **crypto pki import** e as seções posteriores detalhando exemplos de autenticação/importação para certificados multinível fornecerão contexto adicional para esses quatro marcadores.

Os pontos de confiança podem ter vários comandos configurados. Esses comandos podem ser usados para influenciar os valores em uma CSR (Certificate Signing Request, Solicitação de assinatura de certificado) criada pelo dispositivo usando o comando **crypto pki enroll** em um ponto confiável.

Há muitos comandos diferentes disponíveis para um ponto confiável (muitos demais para detalhar neste documento), mas alguns exemplos mais comuns são detalhados no ponto confiável de exemplo e na tabela abaixo:

```
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	**Descrição**
crypto pki trustpoint labTrustpoint	Rótulo de configuração legível por humanos para este ponto de confiança. Usado para vincular a recursos ou serviços em comandos posteriores.
enrollment terminal pem	Determina a ação que o comando **crypto pki enroll** executará. Neste exemplo, **enrollment terminal pem** indica que a solicitação de assinatura de certificado (CSR) será enviada ao terminal em um texto formatado em PEM Base64.

	Outras opções, como **enrollment selfsigned**, podem ser usadas para criar um certificado autoassinado ou **enrollment url** podem ser configuradas para definir um URL HTTP e aproveitar o protocolo SCEP (Simple Certificate Enrollment Protocol). Ambos os métodos estão fora do escopo deste documento.
serial-number none	Determina se a série dos dispositivos IOS XE será adicionada ao CSR. Isso também desabilita o prompt durante o comando crypto pki enroll.
fqdn none	Determina se o FQDN (Fully Qualified Domain Name, Nome de domínio totalmente qualificado) será adicionado ao CSR. Isso também desabilita o prompt durante o comando crypto pki enroll.
ip-address none	Determina se o endereço IP dos dispositivos IOS XE será adicionado ao CSR. Isso também desabilita o prompt durante o comando crypto pki enroll.
subject-name cn=router.example.cisco.com	Indica o X500 formatado que será adicionado ao CSR.
subject-alt-name myrouter.example.cisco.com	A partir do IOS XE 17.9.1, uma lista separada por vírgulas de valores de Nome alternativo do assunto (SAN) pode ser adicionada ao CSR.
revocation-check none	Indica como o dispositivo IOS XE deve verificar a validade do certificado. Opções como Lista de Certificados Revogados (CRL), Protocolo de Status de Certificados Online (OCSP) podem ser usadas se forem suportadas pela Autoridade de Certificação de sua escolha. Isso é usado principalmente quando o ponto confiável é utilizado por algum outro recurso ou serviço configurado do IOS XE. O status de revogação também é verificado quando um certificado é autenticado com um ponto confiável.
rsakeypair rsaKey	Instrui o comando a utilizar o par de chaves RSA com este rótulo específico. Para os certificados ECDSA, utilizar o comando "eckeypair ecKey" que faz referência ao rótulo da chave EC
hash sha256	Esse comando influencia o tipo de algoritmo de hash a ser usado. As opções são SHA1, SHA256, SHA384, SHA512

crypto pki enroll

O comando **crypto pki enroll** é usado para disparar o comando enrollment em um determinado ponto confiável. (Sintaxe do comando)
Para o exemplo trustpoint exibido anteriormente, o comando **crypto pki enroll labTrustpoint** exibirá a solicitação de assinatura de certificado (CSR) para o terminal no formato de texto PEM Base64 como mostrado no exemplo abaixo.

Esta solicitação de assinatura de certificado agora pode ser salva em um arquivo de texto ou cópia e colada da linha de comando para fornecer a qualquer CA de terceiros para validação e assinatura.

<#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 authentication

O comando **crypto pki authenticate** é usado para adicionar um certificado CA confiável a um determinado ponto confiável. Cada ponto confiável pode ser autenticado uma única vez. Ou seja, um ponto confiável só pode conter um único certificado intermediário ou raiz de CA. Executar o comando uma segunda vez e adicionar um novo certificado substituirá o primeiro certificado.

Com o comando **enrollment terminal pem** configurado, o comando **crypto pki authenticate** solicitará ao roteador um certificado formatado em PEM Base64 a ser carregado via CLI. (Sintaxe do comando)

Um administrador pode autenticar um ponto confiável para adicionar a raiz e os certificados intermediários opcionais em uma cadeia de certificados com a finalidade de importar um certificado de ID do dispositivo posteriormente.

Os administradores também podem autenticar um ponto confiável para adicionar outras CAs raiz confiáveis ao dispositivo IOS XE com a finalidade de ativar relações de confiança com dispositivos pares durante handshakes de protocolo com esse dispositivo par.

Para ilustrar ainda mais, um dispositivo de mesmo nível pode apresentar uma cadeia de certificados assinada por "CA 1 raiz". Para que a validação do certificado durante o handshake de protocolo entre o dispositivo

IOS XE e o dispositivo par seja bem-sucedida; um administrador pode usar o comando **crypto pki authenticate** para adicionar o certificado CA a um ponto de confiança no dispositivo IOS XE.

O item principal a ser lembrado: a autenticação de pontos confiáveis usando o crypto pki authenticate é sempre para adicionar certificados raiz de CA ou intermediários a um ponto confiável; não para adicionar certificados de identidade. Observe que esse conceito também é aplicado à autenticação de certificados autoassinados de outro dispositivo de mesmo nível.

O exemplo abaixo mostra como autenticar um ponto confiável de um ponto anterior usando o 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
```

crypto pki import

Este comando é usado para importar o certificado de identidade (ID) para um ponto confiável. Um único ponto de confiança só pode conter um único certificado de ID e a emissão do comando pela segunda vez solicitará a substituição do certificado importado anteriormente. (Sintaxe do comando)

O exemplo abaixo mostra como importar um certificado Identity para o ponto confiável do exemplo anterior usando o 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
```

Um administrador receberá um erro se tentar importar um certificado antes que o ponto confiável tenha autenticado o certificado CA usado para assinar diretamente esse certificado.

<#root>

```
Router(config)#
```

```
crypto pki import labTrustpoint certificate
```

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

Autenticando certificados CA de mesmo nível

Os certificados de CA de mesmo nível são adicionados ao IOS XE usando o mesmo método de adicionar qualquer certificado de CA. Ou seja, eles são autenticados em um ponto confiável usando o comando **crypto pki authenticate**.

O comando abaixo mostra como criar um ponto confiável e autenticar um certificado CA de terceiros de mesmo nível.

1. Primeiro, crie um ponto de confiança com algum nome descritivo que armazenará o certificado CA do par
2. configure **enrollment terminal pem** para que o comando crypto pki authenticate solicite o certificado por meio da linha de comando.
3. Configure **revocation-check none** para ignorar a verificação CRL/OCSP durante o processo de importação
4. Autenticar o ponto confiável e fornecer o certificado
5. Repita as etapas de 1 a 4 para conforme necessário para certificados de CA de mesmo nível (lembre-se apenas de um certificado de CA por ponto de confiança!)

<#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
```

Autenticando um ou mais certificados intermediários

Os exemplos anteriores detalham como gerar um CSR usando **crypto pki enroll**, autenticar o certificado CA raiz usando **crypto pki authenticate** e importar o certificado de identidade usando **crypto pki import**. No entanto, ao introduzir certificados intermédios, o processo difere ligeiramente. Não tema, os mesmos conceitos e comandos ainda se aplicam! A diferença reside na forma como são estabelecidos os pontos de confiança que detêm os certificados.

Lembre-se de que cada ponto confiável pode conter apenas um único certificado de CA raiz ou intermediário. Assim, em um exemplo em que temos uma cadeia de CA como abaixo da mostrada abaixo, é impossível usar o comando crypto pki authenticate para adicionar mais de um certificado CA:

<#root>

```
- Root CA


 - Intermediate CA 1


   - Identity Certificate
```

Solução:

1. Crie um ponto confiável que armazenará a CA raiz autenticada.
2. Em seguida, autentique o certificado intermediário com o ponto confiável usado para criar o CSR
3. Finalmente, importe o certificado de identidade para o ponto de confiança final.

Usando a tabela abaixo, pode-se ilustrar o comando certificate to para o mapeamento de ponto confiável com cores que correspondem à cadeia anterior para auxiliar na visualização.

Nome do certificado	Ponto confiável a ser usado	Comando a ser usado
CA raiz	crypto pki trustpoint **ROOT-CA**	crypto pki authenticate **ROOT-CA**
CA 1 intermediário	crypto pki trustpoint	crypto pki authenticate **labTrustpoint**

	labTrustpoint	
Certificado de identidade	crypto pki trustpoint **labTrustpoint**	certificado crypto pki import **labTrustpoint**

A mesma lógica pode ser aplicada a uma cadeia de certificados com dois certificados CA intermediários. Novamente, as cores são fornecidas para ajudar na visualização de onde a nova CA intermediária é aplicada à configuração do IOS XE.

<#root>

- Root CA

- Intermediate CA 1

- Intermediate CA 2

- Identity Certificate

Nome do certificado	**Ponto confiável a ser usado**	**Comando a ser usado**
CA raiz	crypto pki trustpoint **ROOT-CA**	crypto pki authenticate **ROOT-CA**
CA 1 intermediário	crypto pki trustpoint **INTER-CA**	crypto pki authenticate **INTER-CA**
CA 2 intermediário	crypto pki trustpoint **labTrustpoint**	crypto pki authenticate **labTrustpoint**
Certificado de identidade	crypto pki trustpoint **labTrustpoint**	certificado crypto pki import **labTrustpoint**

Olhando de perto, pode-se observar dois padrões:

1. Todos os certificados Raiz ou Intermediários são carregados em pontos confiáveis usando **crypto pki authenticate** (independentemente de quantos existam).
2. Também é possível observar que o certificado final antes do certificado de identidade do dispositivo (leia aquele que assinou diretamente o certificado de identidade) é sempre autenticado no mesmo ponto confiável onde o certificado de identidade deve ser importado.
    - Semelhante ao erro mostrado anteriormente, o IOS XE não permitirá que um administrador importe um certificado sem primeiro autenticar o certificado de CA usado para assinar diretamente este certificado.

Esses dois padrões acima podem ser usados para qualquer número de certificados intermediários além de dois, embora na maioria das implantações um administrador provavelmente veja mais de duas CAs intermediárias em uma cadeia de certificados.

Para fins de integridade, a tabela de certificado de raiz/identidade a seguir também é fornecida:

<#root>

```
- Root CA
```

```
 - Identity Certificate
```

Nome do certificado	Ponto confiável a ser usado	Comando a ser usado
CA raiz	crypto pki trustpoint **labTrustpoint**	crypto pki authenticate **labTrustpoint**
Certificado de identidade	crypto pki trustpoint **labTrustpoint**	certificado crypto pki import **labTrustpoint**

Verificação

- Durante o processo de autenticação ou importação, várias verificações de sanidade são realizadas pelo IOS XE para garantir que o certificado seja válido e bem formado. Esses erros serão impressos na tela ou nos logs (show logging) para linhas que começam com "CRYPTO_PKI"

Alguns exemplos comuns são detalhados abaixo:

As verificações Válidas Antes/Depois são realizadas com base no tempo configurado versus o encontrado no certificado

<#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 a verificação de revogação não estiver desativada, o IOS XE executará uma verificação de revogação pelo método configurado antes de importar o certificado

<#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

Para exibir detalhes sobre a configuração do ponto confiável, autenticado ou importado, use os comandos abaixo:

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

Troubleshooting

Ao depurar problemas de importação ou outros problemas de PKI, utilize as seguintes depurações.

```
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
```

Conceitos avançados de PKI do IOS

Importando um certificado formatado por PKCS12

Alguns provedores de CA podem fornecer arquivos no formato PKCS#12 (.pfx, .p12).

PKCS#12 é um tipo especial de formato de certificado em que toda a cadeia de certificados, desde o certificado raiz até o certificado de identidade, é agrupada junto com o par de chaves rsa.

Esse formato é muito útil para importação com o IOS XE e pode ser facilmente importado usando o comando abaixo:

<#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.
```

Exportação de certificados PKCS12 ou PEM

Um administrador pode exportar certificados para o terminal como PEM de texto simples Base64, texto simples criptografado Base64 ou formato PKCS12 para importar para outros dispositivos pares.

Isso é útil ao ativar novos dispositivos pares e um administrador precisa compartilhar um certificado CA raiz que assinou o certificado de identidade dos dispositivos.

Alguns exemplos de sintaxe estão abaixo:

<#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

Exportar chaves RSA

Pode ser necessário exportar chaves RSA para importação em algum outro dispositivo ou para uso nos esforços de solução de problemas. Supondo que o par de chaves tenha sido criado como exportável, as chaves podem ser exportadas usando o comando crypto key export junto com um método de criptografia (DES, 3DES, AES) e senha.

Uso de exemplo:

<#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 a chave não for exportável, um erro será exibido.

<#root>

```
Router(config)#
```

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

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

Importar chaves RSA geradas fora da caixa

Alguns administradores podem executar RSA e criação de certificado off-box, é possível importar as chaves RSA usando o comando **crypto key import** como mostrado abaixo usando a senha.

<#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.
```

Excluir chaves RSA

Utilize o comando **crypto key zeroize rsa rsaKey** para excluir um par de chaves RSA chamado rsaKey.

Importar o pacote Cisco Trusted CA via Trustpool

Os pools confiáveis variam um pouco de um ponto confiável, mas o uso do núcleo é o mesmo. Onde os

pontos de confiança normalmente contêm um único certificado de CA, um pool de confiança conterá várias CAs confiáveis.

A Cisco publica pacotes de CA em https://www.cisco.com/security/pki/

Um uso comum é fazer o download do arquivo ios_core.p7b usando o comando abaixo:

<#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)#
```

Perguntas mais freqüentes

A exclusão de um ponto confiável invalida o CSR ou uma cadeia de certificados concedida a partir de um determinado CSR?

Não, depois que o CSR é gerado e salvo, o ponto de confiança pode ser excluído e adicionado novamente sem invalidar o CSR.

Isso é frequentemente usado pelo suporte técnico da Cisco para começar do zero quando a autenticação/importação de certificados tiver dado errado.

Desde que o administrador ou o engenheiro de suporte não gere novamente as chaves RSA; o CSR ou a cadeia de certificados assinados pode ser importada e autenticada/importada.

Importante! A remoção do ponto confiável **WILL** excluirá todos os certificados autenticados/importados que possam ser mais problemáticos, supondo que esses certificados estejam atualmente em uso por algum serviço ou recurso.

A geração de um CSR em um ponto de confiança invalidará o certificado existente?

Não, isso é comum quando os certificados estão prestes a expirar. Um administrador pode executar um comando **crypto pki enroll** para criar um novo CSR e iniciar o processo de assinatura de certificado com uma CA enquanto os certificados existentes que foram autenticados/importados permanecem em uso. No momento em que um administrador substitui os certificados por **crypto pki authenticate**/**crypto pki import** é o momento em que os certificados antigos são substituídos.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.