Configuration des certificats signés CA avec IOS XE PKI

Table des matières

Introduction

Ce document sert de guide général pour la configuration des certificats IOS XE signés par une autorité de certification tierce.

Ce document explique en détail comment importer une chaîne CA signée à plusieurs niveaux comme pour le périphérique servant de certificat d'identité (ID), ainsi que comment importer d'autres certificats tiers dans le but de valider le certificat.

Conditions préalables

Exigences

NTP et Heure d'horloge DOIVENT être configurés lors de l'utilisation des fonctionnalités IOS PKI.

Si un administrateur ne configure pas NTP, vous pouvez rencontrer des problèmes avec un certificat généré avec une date/heure future/passée. Ce décalage dans la date ou l'heure peut entraîner des problèmes d'importation et d'autres problèmes en cours de route.

Exemple de configuration NTP :

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

Composants utilisés

- Routeur Cisco exécutant Cisco IOS® XE17.11.1a

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Notez que certaines fonctionnalités détaillées dans ce document peuvent ne pas être disponibles dans les versions antérieures d'IOS XE. Dans la mesure du possible, on a pris soin de documenter l'introduction ou la modification d'une commande ou d'une fonction.

Reportez-vous toujours à la documentation officielle des fonctionnalités PKI d'IOS XE pour une version donnée afin de comprendre les limitations ou les modifications qui peuvent être pertinentes pour votre version spécifique :

Exemples:

- 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

Configuration IOS XE PKI

À un niveau élevé, un administrateur doit effectuer les actions suivantes lorsqu'il travaille avec des

certificats IOS XE PKI :

1. Créer une clé à utiliser avec une fonctionnalité ou un service (génération de clé de chiffrement)
2. Configurez un point de confiance avec divers paramètres et liez la clé. (crypto pki trustpoint)
3. Générer une demande de signature de certificat (CSR) (crypto pki enroll)
4. Fournir le CSR à une autorité de certification pour signature (non traité dans ce document)
5. Authentifier les certificats CA racine et/ou intermédiaires (crypto pki authenticate)
6. Importer les certificats de périphérique (importation crypto pki)
7. Facultatif : authentifier les certificats CA homologues (crypto pki authenticate)

Ces étapes sont détaillées dans les sections suivantes, regroupées par commandes requises pour l'action donnée.

génération de clé de chiffrement

De nombreux administrateurs ont entré cette commande pour activer Secure Socket Shell (SSH) sur un routeur ou dans le cadre d'un guide de configuration d'une fonctionnalité. Cependant, peu d'entre eux n'ont pas disséqué ce que la commande fait réellement.

Prenez par exemple les commandes ci-dessous :

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

Disséquer ces commandes dans les parties spécifiques détaillera l'utilisation :

- La première partie de la commande en noir (crypto key generate) indique au routeur que nous allons créer une nouvelle clé. Il existe d'autres options, telles que l'exportation de clé de chiffrement, l'importation de clé de chiffrement ou la taille de clé de chiffrement, qui seront détaillées ultérieurement.
- La partie suivante de la commande en vert (rsa general-keys, ec) indique au routeur exactement quel type de clé nous créons. Dans la plupart des cas, une paire de clés Rivest-Shamir-Adleman (RSA) composée d'une clé publique/privée sera utilisée, mais un administrateur peut également configurer une courbe elliptique (EC) pour une utilisation avec des fonctionnalités telles que celles qui nécessitent des certificats ECDSA ou pour une utilisation avec des échanges ECDHE.
- La commande en orange définit la taille de notre clé.
    - Pour RSA, le module est la terminologie et des valeurs telles que 512-4096 sont disponibles en option. La taille du module par défaut varie selon la version, mais il est conseillé de suivre les meilleures pratiques de Cisco pour la cryptographie de nouvelle génération et d'utiliser des clés supérieures à 2048.
    - Pour EC, la commande key-size est nécessaire pour spécifier le nombre de bits dans la clé. Les options sont 256, 384 ou 512.
- La commande en violet définit l'étiquette de cette clé. Ceci est important, car un

administrateur peut avoir besoin de définir plusieurs clés à différentes fins sur le même périphérique IOS XE. L'étiquette est utilisée pour spécifier la clé exacte à utiliser avec une fonction donnée. Dans la mesure du possible, utilisez toujours une étiquette pour distinguer les touches utilisées et faciliter l'attribution de touches aux fonctions. Par exemple : label SSH, label CUBE, label HTTPS créera deux clés à utiliser avec différents services ou fonctionnalités.
    - L'étiquette par défaut d'une clé est « devices hostname.domain ». Certains périphériques peuvent générer des clés RSA au premier démarrage. En ne saisissant pas de post-correction d'étiquette, un administrateur risque d'écraser/de régénérer la mauvaise clé par inadvertance
- La dernière commande en bleu est le suffixe exportable. Cette commande détaille que la clé peut être utilisée avec la commande crypto pki export pour l'exportation et l'utilisation avec d'autres systèmes. Par exemple, vous pouvez importer dans un périphérique haute disponibilité homologue une clé unique utilisée par les deux membres d'une paire haute disponibilité ou l'utiliser dans des outils de dépannage tels que Wireshark pour décrypter les sessions TLS basées sur RSA. Quelle que soit la raison, il faut préciser que les clés RSA ne peuvent être créées que comme étant exportables dès le début. Si un administrateur crée une clé RSA non exportable, cette clé ne peut pas être définie comme étant exportable sans régénérer la clé, ce qui peut avoir des effets d'ondulation sur d'autres fonctionnalités telles que l'invalidation de tous les certificats créés à l'aide de cette clé. Ceci étant dit, une clé exportable peut être rétrogradée à non exportable sans régénérer la clé à l'aide de la commande crypto key move rsa rsaKeyLabel non exportable

Exemples de configuration :

<#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
```

Exemples de vérification :

<#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

Les points de confiance sont un concept de type « dossier » pour le stockage et la gestion des certificats PKI dans IOS XE. (syntaxe de commande)

À un niveau élevé :

1. Chaque point de confiance IOS XE peut contenir un certificat CA racine unique ou intermédiaire chargé au moyen de la commande crypto pki authenticate. Considérez les points de confiance authentifiés comme ajoutant des certificats qui sont maintenant approuvés par le périphérique.
2. Chaque point de confiance IOS XE peut également importer un certificat d'identité (ID) unique chargé par l'intermédiaire de la commande crypto pki import. Le certificat d'ID est ce certificat de périphérique qui est généralement lié à un service ou une fonctionnalité.
3. Un administrateur peut utiliser la commande authenticate et import sur le même point de confiance (qui est nécessaire pour importer un certificat d'ID discuté plus loin). Lors de l'utilisation du workflow d'authentification/importation, le point de confiance contiendra deux certificats (racine/intermédiaire + certificat d'identité).
4. Lorsque des points de confiance sont utilisés dans le but de stocker des certificats d'autorité de certification intermédiaires/racine d'homologue de confiance, seuls les crypto pki authenticate est nécessaire. Dans ce scénario, un point de confiance ne contiendra que le certificat unique authentifié par l'administrateur.

Remarque : les sections à venir pour crypto pki authenticate et crypto pki import et les sections ultérieures détaillant les exemples d'authentification/importation pour les certificats multiniveaux fourniront un contexte supplémentaire à ces quatre puces.

Diverses commandes peuvent être configurées pour les points de confiance. Ces commandes peuvent être utilisées pour influencer les valeurs dans une demande de signature de certificat (CSR) créée par le périphérique à l'aide de la commande crypto pki enroll sur un point de confiance.

Il existe de nombreuses commandes différentes pour un point de confiance (beaucoup trop nombreuses pour être détaillées dans ce document), mais certains exemples plus courants sont détaillés dans l'exemple de point de confiance et le tableau ci-dessous :

```
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
```

Commande	Description
crypto pki trustpoint labTrustpoint	Étiquette de configuration lisible par l'utilisateur pour ce point de confiance. Utilisé pour établir une liaison avec des fonctions ou des services dans des commandes ultérieures.
enrollment terminal pem	Détermine l'action que la commande crypto pki enroll va effectuer. Dans cet exemple, enrollment terminal pem indique que la demande de signature de certificat (CSR) sera envoyée au terminal dans un texte au format PEM Base64. D'autres options telles que enrollment selfsigned peuvent être utilisées pour créer un certificat auto-signé ou une URL d'inscription peuvent être configurées pour définir une URL HTTP et tirer parti du protocole SCEP (Simple Certificate Enrollment Protocol). Ces deux méthodes sortent du cadre de ce document.
numéro de série none	Détermine si les périphériques série IOS

	XE seront ajoutés au CSR. Cela désactive également l'invite pendant la commande crypto pki enroll.
fqdn none	Détermine si le nom de domaine complet (FQDN) sera ajouté au CSR. Cela désactive également l'invite pendant la commande crypto pki enroll.
ip-address none	Détermine si l'adresse IP des périphériques IOS XE sera ajoutée au CSR. Cela désactive également l'invite pendant la commande crypto pki enroll.
subject-name cn=router.example.cisco.com	Indique le format X500 qui sera ajouté au CSR.
subject-alt-name myrouter.example.cisco.com	À partir de la version 17.9.1 d'IOS XE, une liste séparée par des virgules des valeurs de l'autre nom du sujet (SAN) peut être ajoutée au CSR.
revocation-check none	Indique comment le périphérique IOS XE doit vérifier la validité du certificat. Des options telles que la liste de révocation de certificats (CRL), le protocole OCSP (Online Certificate Status Protocol) peuvent être utilisées si elles sont prises en charge par l'autorité de certification de votre choix. Cette option est principalement utilisée lorsque le point de confiance est utilisé par une autre fonctionnalité ou un autre service IOS XE configuré. L'état de révocation est également vérifié lorsqu'un certificat est authentifié avec un point de confiance.
rsakeypair rsaKey	Indique à la commande d'utiliser la paire de clés RSA avec cette étiquette spécifique. Pour les certificats ECDSA, utilisez la commande "eckeypair ecKey" qui fait référence à l'étiquette de la clé EC
hachage sha256	Cette commande influence le type d'algorithme de hachage à utiliser. Les options sont SHA1, SHA256, SHA384 et SHA512

crypto pki enroll

La commande crypto pki enroll est utilisée pour déclencher la commande enrollment sur un point de confiance donné. (Syntaxe de commande)
Pour l'exemple trustpoint précédemment affiché, la commande crypto pki enroll labTrustpoint affichera la demande de signature de certificat (CSR) au terminal au format texte PEM Base64 comme indiqué dans l'exemple ci-dessous.

Cette demande de signature de certificat peut désormais être enregistrée dans un fichier texte ou copiée et collée à partir de la ligne de commande afin de fournir à toute autorité de certification tierce une validation et une signature.

<#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 authenticate

La commande crypto pki authenticate est utilisée pour ajouter un certificat CA approuvé à un point de confiance donné. Chaque point de confiance peut être authentifié une seule fois. En d'autres termes, un point de confiance ne peut contenir qu'un seul certificat CA racine ou intermédiaire. L'exécution de la commande une deuxième fois et l'ajout d'un nouveau certificat remplaceront le premier certificat.

Lorsque la commande enrollment terminal pem est configurée, la commande crypto pki authenticate invite le routeur à télécharger un certificat au format PEM Base64 via l'interface de ligne de commande (CLI). (Syntaxe de commande)

Un administrateur peut authentifier un point de confiance afin d'ajouter les certificats racine et les certificats intermédiaires facultatifs dans une chaîne de certificats dans le but d'importer ultérieurement le certificat d'ID d'un périphérique.

Un administrateur peut également authentifier un point de confiance pour ajouter d'autres autorités de certification racine de confiance au périphérique IOS XE dans le but d'activer des relations de confiance avec des périphériques homologues pendant les échanges de protocole avec ce périphérique homologue.

Pour illustrer davantage, un périphérique homologue peut comporter une chaîne de certificats signée par « Root CA 1 ». Pour que la validation du certificat lors de la connexion de protocole entre le périphérique IOS XE et le périphérique homologue réussisse, un administrateur peut utiliser la commande crypto pki authenticate pour ajouter le certificat CA à un point de confiance sur le périphérique IOS XE.

L'élément principal à retenir : l'authentification des points de confiance à l'aide de crypto pki authenticate est toujours destinée à ajouter des certificats d'origine ou intermédiaires CA à un point de confiance ; pas à ajouter des certificats d'identité. Notez que ce concept est également appliqué à l'authentification des certificats auto-signés à partir d'un autre périphérique homologue.

L'exemple ci-dessous montre comment authentifier un point de confiance d'une version antérieure en utilisant la commande 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

Cette commande est utilisée pour importer le certificat d'identité (ID) dans un point de confiance. Un seul point de confiance ne peut contenir qu'un seul certificat d'ID et l'exécution d'une deuxième commande vous invite à remplacer le certificat importé précédemment. (Syntaxe de commande)

L'exemple ci-dessous montre comment importer un certificat d'identité dans l'exemple trustpoint à partir de plus tôt en utilisant la commande 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 administrateur obtiendra une erreur s'il tente d'importer un certificat avant que le point de confiance n'ait authentifié le certificat d'autorité de certification utilisé pour signer directement ce certificat.

<#root>

```
Router(config)#

crypto pki import labTrustpoint certificate

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

Authentification des certificats CA homologues

Les certificats d'autorité de certification homologues sont ajoutés à IOS XE en utilisant la même méthode d'ajout de tout certificat d'autorité de certification. En d'autres termes, ils sont authentifiés par rapport à un point de confiance à l'aide de la commande crypto pki authenticate.

La commande ci-dessous montre comment créer un point de confiance et authentifier un certificat d'autorité de certification tierce homologue.

1. Commencez par créer un point de confiance avec un nom descriptif qui contiendra le certificat de l'autorité de certification homologue
2. configurez enrollment terminal pem de sorte que la commande crypto pki authenticate demande le certificat via la ligne de commande.
3. Configurez revocation-check none pour ignorer la vérification CRL/OCSP pendant le processus d'importation
4. Authentifier le point de confiance et fournir le certificat
5. Répétez les étapes 1 à 4 pour, comme requis pour les certificats d'autorité de certification homologues (rappelez-vous un seul certificat d'autorité de certification par point de confiance !)

<#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
```

Authentification d'un ou plusieurs certificats intermédiaires

Les exemples précédents expliquent en détail comment générer un CSR à l'aide de crypto pki enroll, authentifier le certificat d'autorité de certification racine à l'aide de crypto pki authenticate, puis importer le certificat d'identité à l'aide de crypto pki import.
Toutefois, lors de l'introduction de certificats intermédiaires, le processus diffère légèrement. Ne craignez rien, les mêmes concepts et commandes s'appliquent toujours ! La différence réside dans la manière dont les points de confiance qui détiennent les certificats sont présentés.

N’oubliez pas que chaque point de confiance ne peut contenir qu’un seul certificat CA racine ou intermédiaire. Ainsi, dans un exemple où nous avons une chaîne CA comme ci-dessous, il est impossible d'utiliser la commande crypto pki authenticate pour ajouter plus d'un certificat CA :

<#root>

- Root CA

 - Intermediate CA 1

- Identity Certificate

Solution :

1. Créez un point de confiance qui contiendra l'autorité de certification racine authentifiée.
2. Ensuite, authentifiez le certificat intermédiaire avec le point de confiance utilisé pour créer le CSR
3. Enfin, importez le certificat d'identité dans le point de confiance final.

En utilisant le tableau ci-dessous, on peut illustrer le certificat à la commande de mappage de point de confiance avec des couleurs qui correspondent à la chaîne précédente pour aider à la visualisation.

Nom du certificat	Trustpoint à utiliser	Commande à utiliser
Autorité de certification racine	crypto pki trustpoint ROOT-CA	crypto pki authenticate ROOT-CA
CA intermédiaire 1	crypto pki trustpoint labTrustpoint	crypto pki authenticate labTrustpoint
Certificat D'Identité	crypto pki trustpoint labTrustpoint	crypto pki import labTrustpoint, certificat

La même logique peut être appliquée à une chaîne de certificats avec deux certificats CA intermédiaires. De nouveau, des couleurs sont fournies pour faciliter la visualisation de l'emplacement d'application de la nouvelle autorité de certification intermédiaire à la configuration IOS XE.

<#root>

- Root CA

- Intermediate CA 1

- Intermediate CA 2

- Identity Certificate

Nom du certificat	Trustpoint à utiliser	Commande à utiliser
Autorité de certification racine	crypto pki trustpoint ROOT-CA	crypto pki authenticate ROOT-CA

CA intermédiaire 1	crypto pki trustpoint INTER-CA	crypto pki authenticate INTER-CA
CA 2 intermédiaire	crypto pki trustpoint labTrustpoint	crypto pki authenticate labTrustpoint
Certificat D'Identité	crypto pki trustpoint labTrustpoint	crypto pki import labTrustpoint, certificat

En regardant de plus près, on peut remarquer deux modèles :

1. Tous les certificats racine ou intermédiaires sont chargés dans des points de confiance à l'aide de crypto pki authenticate (quel que soit le nombre).
2. On peut également remarquer que le certificat final avant le certificat d'identité du périphérique (lire celui qui a directement signé le certificat d'identité) est toujours authentifié sur le même point de confiance où le certificat d'identité doit être importé.
    - Comme pour l'erreur affichée précédemment, IOS XE ne permet pas à un administrateur d'importer un certificat sans authentifier au préalable le certificat CA utilisé pour signer directement ce certificat.

Ces deux modèles ci-dessus peuvent être utilisés pour un nombre quelconque de certificats intermédiaires au-delà de deux, bien que dans la plupart des déploiements, un administrateur est susceptible de voir plus de deux autorités de certification intermédiaires dans une chaîne de certificats.

Par souci d'exhaustivité, le tableau de certificats racine/d'identité suivant est également fourni :

<#root>

```
- Root CA
```

```
 - Identity Certificate
```

Nom du certificat	Trustpoint à utiliser	Commande à utiliser
Autorité de certification racine	crypto pki trustpoint labTrustpoint	crypto pki authenticate labTrustpoint
Certificat D'Identité	crypto pki trustpoint labTrustpoint	crypto pki import labTrustpoint, certificat

Vérification

- Au cours du processus d'authentification ou d'importation, IOS XE effectue divers contrôles de validité pour s'assurer que le certificat est valide et bien formé. Ces erreurs s'affichent à l'écran ou les journaux (show logging) recherchent les lignes commençant par « CRYPTO_PKI »

Quelques exemples courants sont détaillés ci-dessous :

Les vérifications avant/après valides sont effectuées en fonction de l'heure configurée par rapport à celle trouvée dans le certificat

<#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

Si la vérification de révocation n’est pas désactivée, IOS XE effectue une vérification de révocation via la méthode configurée avant d’importer le certificat

<#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

Pour afficher des détails sur la configuration, l'authentification ou l'importation des points de confiance, utilisez les commandes ci-dessous :

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

Dépannage

Lorsque vous déboguez des problèmes d'importation ou d'autres problèmes PKI, utilisez les débogages suivants.

```
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
```

Concepts PKI IOS avancés

Importation d'un certificat formaté PKCS12

Certains fournisseurs d'AC peuvent fournir des fichiers au format PKCS#12 (.pfx, .p12).

PKCS#12 est un type spécial de format de certificat dans lequel toute la chaîne de certificats, du certificat racine au certificat d'identité, est regroupée avec la paire de clés rsa.

Ce format est très pratique pour l'importation avec IOS XE et peut être facilement importé à l'aide de la commande ci-dessous :

<#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.
```

Exportation de certificats PKCS12 ou PEM

Un administrateur peut exporter des certificats vers le terminal au format texte clair PEM Base64,

texte clair chiffré Base64 ou PKCS12 pour les importer dans d'autres périphériques homologues.

Cela est pratique lorsque vous abordez de nouveaux périphériques homologues et qu'un administrateur doit partager un certificat d'autorité de certification racine qui a signé le certificat d'identité des périphériques.

Voici un exemple de syntaxe :

<#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
```

Exporter les clés RSA

Il peut être nécessaire d'exporter des clés RSA pour les importer dans un autre périphérique ou pour les utiliser dans le cadre de dépannages. En supposant que la paire de clés a été créée comme exportable, les clés peuvent être exportées à l'aide de la commande crypto key export avec une méthode de cryptage (DES, 3DES, AES) et un mot de passe.

Exemple d'utilisation :

<#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-----
```

Si la clé n'est pas exportable, une erreur s'affiche.

```
<#root>

Router(config)#

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


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

Importer les clés RSA générées hors boîte

Certains administrateurs peuvent exécuter RSA et la création de certificat hors boîte, il est possible d'importer les clés RSA en utilisant la commande crypto key import comme indiqué ci-dessous en utilisant le mot de passe.

```
<#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.
```

Supprimer les clés RSA

Utilisez la commande crypto key zeroize rsa rsaKey pour supprimer une paire de clés RSA nommée rsaKey.

Importer l'offre groupée Cisco Trusted CA via Trustpool

Les pools de confiance varient légèrement d'un point de confiance, mais leur utilisation principale est la même. Lorsque les points de confiance contiennent généralement un seul certificat d'autorité de certification, un pool de confiance contient un certain nombre d'autorités de certification approuvées.

Cisco publie des bundles CA à l'adresse https://www.cisco.com/security/pki/

Il est courant de télécharger le fichier ios_core.p7b à l'aide de la commande ci-dessous :

<#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)#
```

Forum aux questions

La suppression d'un point de confiance invalide-t-elle le CSR ou une chaîne de certificats accordée à partir d'un CSR donné ?

Non, une fois le CSR généré et enregistré, le point de confiance peut être supprimé et rajouté sans invalider le CSR.

Cette méthode est souvent utilisée par le support technique de Cisco pour redémarrer lorsque l'authentification/l'importation de certificats a mal tourné.

Tant que l'administrateur ou l'ingénieur de support ne régénère pas les clés RSA, le CSR ou la chaîne de certificats signés peuvent être importés et authentifiés.

Important ! La suppression du point de confiance ENTRAÎNERA la suppression de tout certificat authentifié/importé qui pourrait être plus problématique si ces certificats sont actuellement utilisés par un service ou une fonctionnalité.

La génération d’un CSR sur un point de confiance invalidera-t-elle le certificat existant ?

Non, c'est fréquent lorsque les certificats arrivent à expiration. Un administrateur peut exécuter une commande crypto pki enroll pour créer un nouveau CSR et démarrer le processus de signature de certificat avec une CA pendant que les certificats existants qui ont été authentifiés/importés restent en cours d'utilisation. Le moment où un administrateur remplace les certificats par crypto pki authenticate/crypto pki import est le moment où les anciens certificats sont remplacés.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.