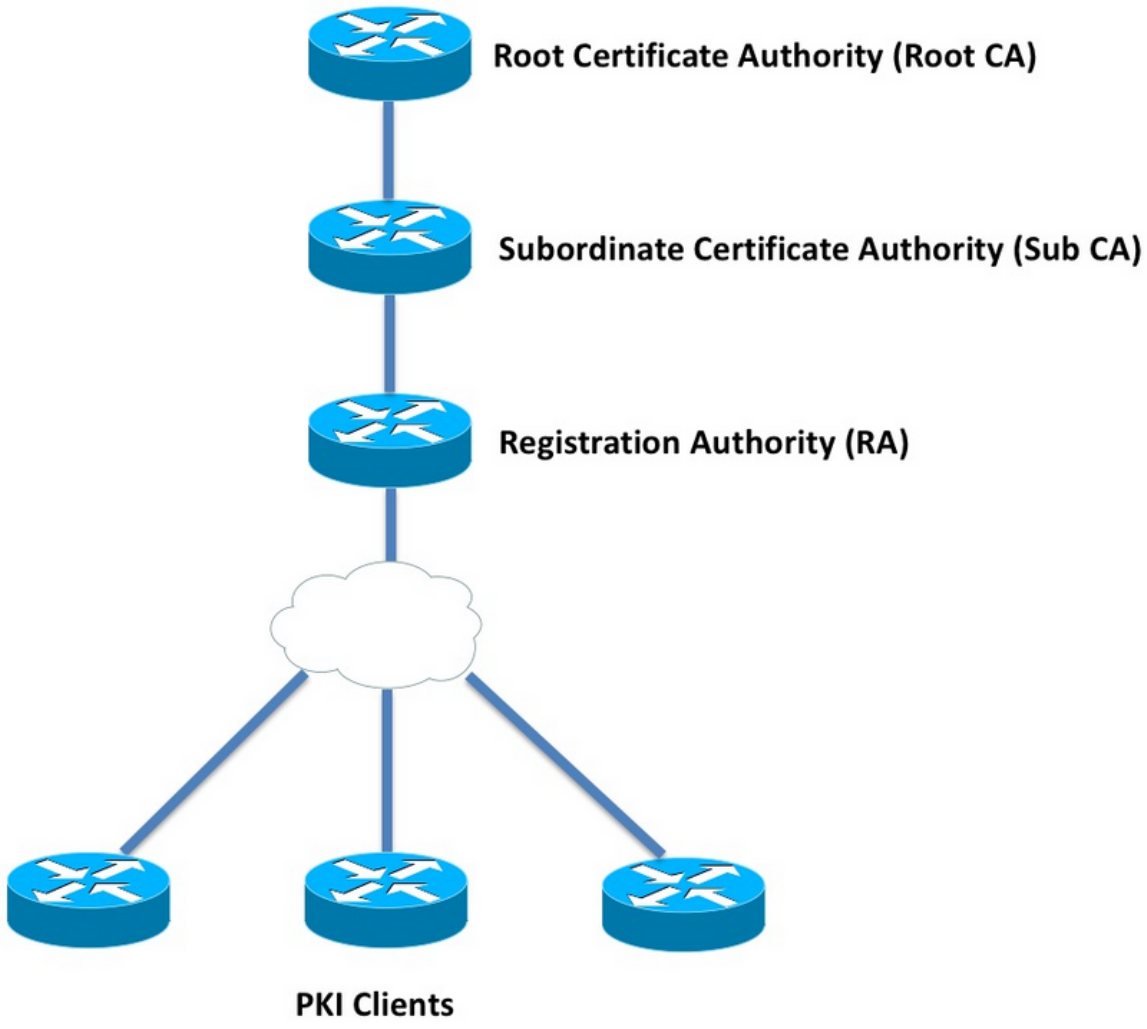
Guide de déploiement de l'ICP IOS : Conception et déploiement initiaux

Contenu

Introduction

Ce document décrit en détail les fonctionnalités du serveur et du client de l'ICP IOS. Il traite des considérations de conception et de déploiement initiaux de l'ICP IOS.

Infrastructure PKI



Autorité de certification

L'autorité de certification (AC), également appelée serveur PKI dans tout le document, est une entité de confiance qui émet des certificats. L'ICP est basée sur la confiance et la hiérarchie de confiance commence à l'autorité de certification racine (autorité de certification racine). Comme l'autorité de certification racine se trouve en haut de la hiérarchie, elle possède un certificat auto-signé.

Autorité de certification subordonnée

Dans la hiérarchie de confiance de l'ICP, toutes les autorités de certification situées en dessous de la racine sont connues sous le nom d'autorités de certification subordonnées (Sub-CA). De toute évidence, un certificat d'autorité de certification secondaire est délivré par l'autorité de certification, qui est un niveau au-dessus.

L'ICP n'impose aucune limite au nombre de sous-autorités de certification dans une hiérarchie donnée. Cependant, dans un déploiement d'entreprise comportant plus de 3 niveaux d'autorité de certification, il peut devenir difficile à gérer.

Autorité d'enregistrement

L'ICP définit une autorité de certification spéciale appelée autorité d'enregistrement (AR), qui est chargée d'autoriser les clients de l'ICP à s'inscrire à une sous-autorité de certification ou à une autorité de certification racine donnée. RA ne délivre pas de certificats aux clients de l'ICP, mais décide quel client de l'ICP peut ou ne peut pas obtenir de certificat par l'AC auxiliaire ou l'AC racine.

Le rôle principal d'une autorité de certification est de décharger la validation de base des demandes de certificat client de l'autorité de certification, et de protéger l'autorité de certification de l'exposition directe aux clients. De cette façon, RA se trouve entre les clients PKI et l'autorité de certification, protégeant ainsi l'autorité de certification de toute forme d'attaque par déni de service.

Client PKI

Tout périphérique demandant un certificat basé sur une paire de clés publique-privée résidante pour prouver son identité à d'autres périphériques est appelé client PKI.

Un client PKI doit être capable de générer ou de stocker une paire de clés public-privé telle que RSA, DSA ou ECDSA.

Un certificat est une preuve d'identité et de validité d'une clé publique donnée, à condition que la clé privée correspondante existe sur le périphérique.

Serveur PKI IOS

Tableau 1 . Évolution des fonctionnalités du serveur ICP IOS

Fonctionnalité	IOS [ISR-G1, ISR-G2]	IOS-XE [ASR1K, ISR4K]
Serveur CA/PKI IOS	12.3(4)T	XE 3.14.0 / 15.5(1)S
Transfert de certificat du serveur PKI IOS	12.4(1)T	XE 3.14.0 / 15.5(1)S
IOS PKI HA	15,0(1)M	NA [Redondance

		implicite entre RP disponible]
RA IOS pour CA 3e partie	15.1(3)T	XE 3.14.0 / 15.5(1)S

Avant d'accéder à la configuration du serveur PKI, l'administrateur doit comprendre ces concepts de base.

Source de temps autorisée

L'une des fondations de l'infrastructure d'ICP est Time. L'horloge système définit si un certificat est valide ou non. Par conséquent, dans IOS, l'horloge doit être faite de manière fiable ou faisant autorité. Sans source de temps faisant autorité, le serveur PKI risque de ne pas fonctionner comme prévu et il est fortement recommandé de faire en sorte que l'horloge de l'IOS fasse autorité en utilisant les méthodes suivantes :

NTP (Network Time Protocol)

La synchronisation de l'horloge système avec un serveur temporel est la seule véritable façon de rendre l'horloge système fiable. Un routeur IOS peut être configuré en tant que client NTP sur un serveur NTP connu et stable du réseau :

```
configure terminal
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

IOS peut également être configuré en tant que serveur NTP, ce qui marquera l'horloge système locale comme faisant autorité. Dans un déploiement à petite échelle de l'ICP, le serveur PKI peut être configuré en tant que serveur NTP pour ses clients PKI :

```
configure terminal
ntp master <stratum-number>

!! optionally, NTP authentication can be enforced
ntp authenticate
ntp authentication-key 1 md5 <key-1>
ntp authentication-key 2 md5 <key-2>
ntp authentication-key 2 md5 <key-2>
ntp trusted-key 1 - 3

!! optinally, an access-list can be configured to restrict NTP clients
```

```
!! first allow the local router to synchronize with the local time-server
access-list 1 permit 127.127.7.1
ntp access-group peer 1

!! define an access-list to which the local time-server will serve time-synchronization services
access-list 2 permit <NTP-Client-IP>
ntp access-group serve-only 2
```

Marquer l'horloge matérielle comme approuvée

Dans IOS, l'horloge matérielle peut être marquée comme faisant autorité à l'aide des éléments suivants :

```
config terminal
clock calendar-valid
```

Ceci peut être configuré avec le protocole NTP, et la principale raison pour cela est de maintenir l'horloge système en état d'autorité lorsqu'un routeur se recharge, par exemple en raison d'une panne de courant, et que les serveurs NTP ne sont pas accessibles. À ce stade, les temporisateurs PKI cesseront de fonctionner, ce qui entraîne des échecs de renouvellement/transfert de certificat. **clock agenda-value** agit comme une sauvegarde dans de telles situations.

Lors de la configuration, il est essentiel de comprendre que l'horloge système ne sera pas synchronisée si la batterie système meurt, et PKI commencera à faire confiance à une horloge non synchronisée. Cependant, il est relativement plus sûr de configurer ceci, plutôt que de ne pas avoir une source de temps faisant autorité du tout.

> **Note**: La commande **clock Calendar-Valide** a été ajoutée dans IOS-XE version XE 3.10.0 / 15.3(3)S ultérieure.

Nom d'hôte et nom de domaine

Il est recommandé de configurer un nom d'hôte et un nom de domaine sur Cisco IOS comme l'une des premières étapes avant de configurer des services liés à PKI. Le nom d'hôte et le nom de domaine du routeur sont utilisés dans les scénarios suivants :

- Le nom de paire de clés RSA par défaut est dérivé en combinant le nom d'hôte et le nom de domaine
- Lors de l'inscription à un certificat, le nom de sujet par défaut est constitué d'un attribut hostname et d'un nom non structuré, qui est le nom d'hôte et le nom de domaine réunis.

Quant au serveur PKI, le nom d'hôte et le nom de domaine ne sont pas utilisés :

- Le nom de la paire de clés par défaut sera identique à celui du nom du serveur PKI
- Le nom d'objet par défaut est composé de CN, qui est identique au nom du serveur PKI.

La recommandation générale est de configurer un nom d'hôte et un nom de domaine appropriés.

```
config terminal
hostname <string>
```

```
ip domain name <domain>
```

Serveur HTTP

IOS PKI Server est activé uniquement si HTTP Server est activé. Il est important de noter que, si le serveur PKI est désactivé en raison de la désactivation du serveur HTTP, il peut continuer à accorder des requêtes hors connexion [via le terminal]. La fonctionnalité HTTP Server est requise pour traiter les requêtes SCEP et envoyer des réponses SCEP.

Le serveur HTTP IOS est activé à l'aide de :

```
ip http server
```

Et le port du serveur HTTP par défaut peut être modifié de 80 à n'importe quel numéro de port valide à l'aide de :

```
ip http port 8080
```

HTTP Max-connection

L'un des goulots d'étranglement, lors du déploiement d'IOS en tant que serveur PKI à l'aide de SCEP, est le nombre maximal de connexions HTTP simultanées et la moyenne des connexions HTTP par minute.
Actuellement, le nombre maximal de connexions simultanées sur un serveur HTTP IOS est limité à 5 par défaut et peut être porté à 16, ce qui est fortement recommandé dans un déploiement à moyenne échelle :

```
ip http max-connections 16
```

Ces installations IOS permettent un maximum de connexions HTTP simultanées jusqu'à 1 000 :

- UniversalK9 IOS avec jeu de licences uck9

La CLI est automatiquement modifiée pour accepter un argument numérique compris entre 1 et 1 000

```
ip http max-connections 1000
```

Le serveur HTTP IOS autorise 80 connexions par minute [580 dans le cas de versions IOS où le nombre maximal de sessions simultanées HTTP peut être porté à 1 000] et lorsque cette limite est atteinte en une minute, l'écouteur HTTP IOS commence à limiter les connexions HTTP entrantes en arrêtant l'écouteur pendant 15 secondes. Cela entraîne la suppression des demandes de connexion client en raison de la **limite de file d'attente de connexion** TCP **atteinte**. Plus d'informations à ce sujet sont disponibles ici

Paire de clés RSA

La paire de clés RSA pour la fonctionnalité PKI Server sur IOS peut être générée automatiquement ou manuellement.
Lors de la configuration d'un serveur PKI, IOS crée automatiquement un point de confiance du même nom que le serveur PKI afin de stocker le certificat du serveur PKI.

Génération manuelle de la paire de clés RSA du serveur PKI :

Étape 1. Créez une paire de clés RSA portant le même nom que celui du serveur PKI :

```
crypto key generate rsa general-keys label <LABEL> modulus 2048
```

Étape 2. Avant d'activer le serveur PKI, modifiez le point de confiance du serveur PKI :

```
crypto pki trustpoint <PKI-SERVER-Name>
  rsakeypair <LABEL>
```

> **Note**: La valeur de module de paire de clés RSA mentionnée sous le point de confiance du serveur PKI n'est pas prise en compte avant IOS version 15.4(3)M4, et ceci est une mise en garde connue. Le module de clé par défaut est de 1 024 bits.

Paire de clés RSA du serveur PKI généré automatiquement :

Lors de l'activation du serveur PKI, IOS génère automatiquement une paire de clés RSA portant le même nom que celui du serveur PKI et la taille du module de clé est de 1 024 bits.

Depuis IOS version 15.4(3)M5, cette configuration crée une paire de clés RSA avec <LABEL> comme nom et la puissance de la clé sera conforme au module <MOD> défini.

```
crypto pki trustpoint <PKI-SERVER-Name>
rsakeypair <LABEL> <MOD>
```

Chauffeur

CSCuu73408 Le serveur ICP IOS doit autoriser une taille de clé autre que la taille par défaut pour le certificat de renversement.

Le serveur PKI de l'IOS CSCuu73408 doit autoriser une taille de clé autre que celle par défaut pour le certificat de renversement.

La norme actuelle du secteur consiste à utiliser au moins 2 048 bits de paire de clés RSA.

Prise en compte du temporisateur de renversement automatique

Actuellement, IOS PKI Server ne génère pas de certificat de renversement par défaut, et il doit être explicitement activé sous le serveur PKI à l'aide de la commande **auto-rollover <days-before-expiration>**. Plus d'informations sur le transfert de certificat sont expliquées dans

Cette commande spécifie le nombre de jours avant l'expiration du certificat PKI Server/CA si l'IOS crée un certificat CA inversé. Notez que le certificat d'autorité de certification de substitution est activé une fois que le certificat d'autorité de certification actif actuel expire. La valeur par défaut est actuellement de 30 jours. Cette valeur doit être définie sur une valeur raisonnable en fonction de la durée de vie du certificat de l'autorité de certification, ce qui influence à son tour la configuration du minuteur d'inscription automatique sur le client PKI.

> **Note**: Le minuteur de renversement automatique doit toujours se déclencher avant le minuteur d'inscription automatique sur le client lors du renversement de certificat CA et client [appelé]

Considérations CRL

L'infrastructure ICP IOS prend en charge deux méthodes de distribution de CRL :

Publier la liste de révocation de certificats sur un serveur HTTP

IOS PKI Server peut être configuré pour publier le fichier CRL à un emplacement spécifique sur un serveur HTTP à l'aide de cette commande sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
 database crl publish <URL>
```

Et le serveur PKI peut être configuré pour incorporer cet emplacement CRL dans tous les certificats clients PKI à l'aide de cette commande sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
cdp-url <CRL file location>
```

SCEP GetCRL, méthode

IOS PKI Server stocke automatiquement le fichier CRL à l'emplacement de base de données spécifique, qui par défaut est nvram, et il est fortement recommandé de conserver une copie sur un serveur SCP/FTP/TFTP à l'aide de cette commande sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
database url <URL>
or
database crl <URL>
```

Par défaut, IOS PKI Server n'intègre pas l'emplacement CDP dans les certificats clients PKI. Si les clients ICP IOS sont configurés pour effectuer un contrôle de révocation, mais que le certificat validé ne contient pas de CDP intégré et que le point de confiance de l'autorité de certification validant est configuré avec l'emplacement de l'autorité de certification (à l'aide de http://<CA-Server-IP ou FQDN>), IOS revient par défaut à la méthode GetCRL basée sur SCEP.
SCEP GetCRL effectue la récupération de CRL en exécutant HTTP GET sur cette URL :

```
http:://<CA-Server-IP/FQDN>/cgi-bin/pkiclient.exe?operation=GetCRL
```

Note: Dans l'interface de ligne de commande IOS, avant de saisir **?**, appuyez sur **Ctrl + V**.

IOS PKI Server peut également incorporer cette URL comme emplacement CDP. L'avantage est double :

- Il garantit que tous les clients PKI basés sur SCEP non basés sur IOS peuvent effectuer la récupération de CRL.
- Sans CDP intégré, les messages de requête GetCRL SCEP IOS sont signés (à l'aide d'un certificat auto-signé temporaire) tel que défini dans le brouillon SCEP. Cependant, il n'est pas nécessaire de signer les demandes de récupération de liste de révocation de certificats et, en incorporant l'URL CDP pour la méthode GetCRL, il est possible d'éviter de signer les demandes de liste de révocation de certificats.

Durée de vie de CRL

La durée de vie CRL du serveur ICP IOS peut être contrôlée à l'aide de cette commande sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
 lifetime crl <0 - 360>
```

La valeur est exprimée en heures. Par défaut, la durée de vie de la liste de révocation de certificats est définie sur 6 heures. En fonction de la fréquence de révocation des certificats, le réglage de la durée de vie des listes de révocation de certificats à une valeur optimale augmente les performances de récupération des listes de révocation de certificats sur le réseau.

Considérations de base de données

IOS PKI Server utilise nvram comme emplacement de base de données par défaut et il est fortement recommandé d'utiliser un serveur FTP ou TFTP ou SCP comme emplacement de base de données. Par défaut, IOS PKI Server crée deux fichiers :

- <Nom-serveur>.ser : contient le dernier numéro de série émis par l'autorité de certification en hexadécimal. Le fichier est au format texte brut et contient les informations suivantes :
  db_version = 1
  last_serial = 0x4
- <Nom-serveur>.crl : fichier CRL codé DER publié par l'autorité de certification

IOS PKI Server stocke les informations dans la base de données à 3 niveaux configurables :

- Minimum : niveau par défaut, et à ce niveau, aucun fichier n'est créé dans la base de données. Par conséquent, aucune information n'est disponible sur le serveur AC concernant les certificats clients octroyés dans le passé.
- Noms : à ce niveau, le serveur ICP IOS crée un fichier nommé <Numéro de série>.cnm pour chaque certificat client émis, où le nom <Numéro de série> fait référence au numéro de série du certificat client émis. Ce fichier cnm contient le nom du sujet et la date d'expiration du certificat client.
- Complet - À ce niveau, IOS PKI Server crée deux fichiers pour chaque certificat client émis :
- <Numéro de série>.cnm
- <Numéro de série>.crt

ici, le fichier crt est le fichier de certificat client, qui est codé DER.

Ces points sont importants :

- Avant d'émettre un certificat client, IOS PKI Server fait référence à <Server-Name>.ser pour déterminer et dériver le numéro de série du certificat.
- Lorsque le niveau de base de données est défini sur Noms ou Terminé, <Serial-Number>.cnm et <Serial-Number>.crt doivent être écrits dans la base de données avant d'envoyer le certificat accordé/délivré au client
- Lorsque l'URL de base de données est définie sur Noms ou Terminé, l'URL de base de données doit disposer d'un espace suffisant pour enregistrer les fichiers. Il est donc recommandé de configurer un serveur de fichiers externe [FTP ou TFTP ou SCP] comme URL de base de données.
- Avec l'URL de base de données externe configurée, il est absolument nécessaire de s'assurer que le serveur de fichiers est accessible pendant le processus d'octroi de certificat, ce qui autrement marquerait le serveur AC comme désactivé. Et une intervention manuelle est nécessaire pour remettre le serveur AC en ligne.

Archive de base de données

Lors du déploiement d'un serveur PKI, il est important de tenir compte des scénarios de défaillance et d'être préparé en cas de défaillance matérielle. Il existe deux façons d'y parvenir :

1. Redondance
   Dans ce cas, deux périphériques ou unités de traitement agissent en tant qu'Active-Standby pour fournir une redondance.
   La haute disponibilité du serveur ICP IOS peut être obtenue à l'aide de deux routeurs ISR HSRP [ISR G1 et ISR G2], comme expliqué dans la

   Les systèmes basés sur IOS XE [ISR4K et ASR1k] n'ont pas d'option de redondance de périphérique disponible. Cependant, dans ASR1k, la redondance inter-RP est disponible par défaut.
2. Archivage de la paire de clés et des fichiers du serveur AC
   IOS fournit une fonctionnalité d'archivage de la paire de clés de serveur PKI et du certificat. L'archivage peut être effectué à l'aide de deux types de fichiers :
   PEM - IOS crée des fichiers au format PEM pour stocker la clé publique RSA, la clé privée RSA cryptée, le certificat du serveur CA. La paire de clés de renversement et les certificats sont archivés automatiquementPKCS12 - IOS crée un fichier PKCS12 unique contenant le

certificat du serveur AC et la clé privée RSA correspondante chiffrée à l'aide d'un mot de passe.

L'archivage de base de données peut être activé à l'aide de cette commande sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
 database archive {pkcs12 | pem} password <password>
```

Il est également possible de stocker les fichiers archivés sur un serveur distinct, en utilisant éventuellement un protocole sécurisé (SCP) à l'aide de la commande suivante sous PKI Server :

```
crypto pki server <PKI-SERVER-Name>
 database url {p12 | pem} <URL>
```

De tous les fichiers de la base de données, à l'exception des fichiers archivés et du fichier .Ser, tous les autres fichiers sont en texte clair et ne présentent aucune menace réelle en cas de perte. Ils peuvent donc être stockés sur un serveur distinct sans subir de surcharge lors de l'écriture des fichiers, par exemple un serveur TFTP.

IOS en tant que sous-CA

Le serveur ICP IOS prend par défaut le rôle d'une autorité de certification racine. Pour configurer un serveur PKI subordonné (Sub-CA), activez d'abord cette commande dans la section Configuration du serveur PKI (avant d'activer le serveur PKI) :

```
crypto pki server <Sub-PKI-SERVER-Name>
 mode sub-cs
```

Àl'aide de cette commande, configurez l'URL de Root-CA sous le point de confiance du serveur PKI :

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```

L'activation de ce serveur PKI déclenche maintenant ces événements :

- Le point de confiance du serveur PKI est authentifié afin d'installer le certificat Root-CA.
- Une fois l'autorité de certification racine authentifiée, IOS génère un CSR pour la contrainte de base Subordonné-CA [x509 contenant l'indicateur CA : TRUE] et l'envoie à l'autorité de certification racine

Quel que soit le mode de subvention configuré sur l'autorité de certification racine, IOS place les demandes de certificat d'autorité de certification (ou d'autorité de certification) dans la file d'attente en attente. Un administrateur doit accorder manuellement les certificats CA.
Pour afficher la demande de certificat en attente et l'ID de demande :

```
show crypto pki server <Server-Name> requests
```

Pour accorder la demande :

```
crypto pki server <Server-Name> grant <request-id>
```

- Àl'aide de cette commande, l'opération SCEP POLL (GetCertInitial) suivante télécharge le

certificat Sub-CA et l'installe sur le routeur, ce qui active le serveur PKI subordonné

IOS comme RA

Le serveur d'ICP des E/S peut être configuré en tant qu'autorité d'enregistrement pour une autorité de certification subordonnée ou racine donnée. Pour configurer le serveur PKI en tant qu'autorité d'enregistrement, activez d'abord cette commande dans la section Configuration du serveur PKI (avant d'activer le serveur PKI) :

```
crypto pki server <RA-SERVER-Name>
 mode ra
```

Ensuite, configurez l'URL de l'Autorité de certification sous le point de confiance du serveur PKI. Ceci indique quelle autorité de certification est protégée par l'autorité de certification :

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

Une autorité d'enregistrement ne délivre pas de certificats, de sorte que la configuration **nom-émetteur** sous l'autorité d'enregistrement n'est pas requise et n'est pas effective même si elle est configurée. Le nom d'objet d'une RA est configuré sous le point de confiance RA à l'aide de la commande **nom-objet**. Il est important de configurer **OU = ioscs RA** comme faisant partie du nom de sujet afin que l'autorité de certification IOS identifie l'autorité de certification IOS, c'est-à-dire pour identifier les demandes de certificat autorisées par l'autorité de certification IOS.

IOS peut agir en tant qu'autorité d'enregistrement auprès d'autorités de certification tierces telles que Microsoft CA, et pour rester compatible, l'autorité d'accès IOS doit être activée à l'aide de cette commande dans la section Configuration du serveur PKI (avant d'activer le serveur PKI) :

```
mode ra transparent
```

En mode RA par défaut, IOS signe les requêtes client [PKCS#10] à l'aide du certificat RA. Cette opération indique au serveur ICP IOS que la demande de certificat a été autorisée par une RA.

Avec le mode RA transparent, IOS transfère les demandes client dans leur format d'origine sans introduire le certificat RA, et ceci est compatible avec Microsoft CA comme exemple connu.

Client PKI IOS

L'une des entités de configuration les plus importantes du client ICP IOS est un point de confiance. Les paramètres de configuration du point de confiance sont expliqués en détail dans cette section.

Source de temps autorisée

Comme nous l'avons déjà souligné, le client de l'ICP doit également disposer d'une source de temps faisant autorité. Le client ICP IOS peut être configuré en tant que client NTP à l'aide de la configuration suivante :

```
configure terminal
```

```
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! Optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

Nom d'hôte et nom de domaine

Il est généralement recommandé de configurer un nom d'hôte et un nom de domaine sur le routeur :

```
configure terminal
hostname <string>
ip domain name <domain>
```

Paire de clés RSA

Dans IOS PKI Client, la paire de clés RSA pour une inscription de point de confiance donnée peut être générée automatiquement ou manuellement.

Le processus de génération automatique de clé RSA implique les éléments suivants :

- IOS crée par défaut une paire de clés RSA de 512 bits
- Le nom de paire de clés généré automatiquement est hostname.domain-name, qui est le nom d'hôte du périphérique combiné au nom de domaine du périphérique
- La paire de clés générée automatiquement n'est pas marquée comme exportable.

Le processus de génération automatique de clé RSA implique les éléments suivants :

- Vous pouvez également générer manuellement une paire de clés RSA à usage général d'une puissance appropriée à l'aide des éléments suivants :
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  Ici, LABEL : nom de la paire de clés RSA
  MOD - Module ou puissance de clé RSA en bits entre 360 et 4096, qui est traditionnellement 512, 1024, 2048 ou 4096.
  L'avantage de la génération manuelle de la paire de clés RSA est la possibilité de marquer la paire de clés comme exportable, ce qui permet à son tour l'exportation complète du certificat d'identité, qui peut ensuite être restauré sur un autre périphérique. Cependant, il faut comprendre les conséquences de cette action sur la sécurité.
- Une paire de clés RSA est liée à un point de confiance avant l'inscription à l'aide de cette commande

```
crypto pki trustpoint MGMT
rsakeypair <LABEL> [<MOD> <MOD>]
```

Ici, si une paire de clés RSA nommée **<LABEL>** existe déjà, elle est récupérée lors de

l'inscription au point de confiance.

Si une paire de clés RSA nommée <LABEL> n'existe pas, l'une des actions suivantes est exécutée lors de l'inscription :

- Si aucun argument <MOD> n'est passé, une paire de clés de 512 bits nommée <LABEL> est générée.
- si un argument <MOD> est passé, une paire de clés à usage général <MOD> de bits nommée <LABEL> est générée
- si deux arguments <MOD> sont passés, une paire de clés de signature <MOD> bits et une paire de clés de chiffrement <MOD> bits, nommées <LABEL> sont générées

Point de confiance

Un point de confiance est un conteneur abstrait pour détenir un certificat dans IOS. Un seul point de confiance est capable de stocker deux certificats actifs à un moment donné :

- Certificat CA : le chargement d'un certificat CA dans un point de confiance donné est appelé processus d'authentification de point de confiance.
- Un certificat d'ID émis par l'autorité de certification - Chargement ou importation d'un certificat d'ID dans un point de confiance donné est appelé processus d'inscription de point de confiance.

Une configuration de point de confiance est connue sous le nom de stratégie d'approbation, et ceci définit que :

- Quel certificat CA est chargé dans le point de confiance ?
- Àquelle autorité de certification le point de confiance s'inscrit-il ?
- Comment l'IOS inscrit-il le point de confiance ?
- Comment un certificat émis par l'autorité de certification donnée [chargé dans le point de confiance] est-il validé ?

Les principaux composants d'un point de confiance sont expliqués ici.

Mode d'inscription

Un mode d'inscription de point de confiance, qui définit également le mode d'authentification de point de confiance, peut être exécuté via 3 méthodes principales :

1. Terminal Enrollment : méthode manuelle permettant d'effectuer l'authentification des points de confiance et l'inscription des certificats à l'aide du copier-coller dans le terminal CLI.
2. SCEP Enrollment - Authentification et inscription des points de confiance à l'aide de SCEP sur HTTP.
3. Profil d'inscription : ici, les méthodes d'authentification et d'inscription sont définies séparément. Outre les méthodes d'inscription au terminal et au SCEP, les profils d'inscription fournissent une option permettant de spécifier des commandes HTTP/TFTP pour effectuer la récupération de fichiers à partir du serveur, qui est défini à l'aide d'une url d'authentification ou d'inscription sous le profil.

Interface source et VRF

L'authentification et l'inscription de Trustpoint sur HTTP (SCEP) ou TFTP (Enrollment Profile)

utilisent le système de fichiers IOS pour effectuer des opérations d'E/S de fichier. Ces échanges de paquets peuvent provenir d'une interface source spécifique et d'un VRF.

En cas de configuration de point de confiance classique, cette fonctionnalité est activée à l'aide des sous-commandes **interface source** et **vrf** sous le point de confiance.

En cas de profils d'inscription, **interface source** et **inscription** Les commandes **| authentication url <http/tftp://Server-location> vrf <vrf-name>** offrent la même fonctionnalité.

Exemple de configuration :

```
vrf definition MGMT
 rd 1:1
 address-family ipv4
 exit-address-family

crypto pki trustpoint MGMT
 source interface Ethernet0/0
 vrf MGMT
```

ou

```
crypto pki profile enrollment MGMT-Prof
 enrollment url http://10.1.1.1:80 vrf MGMT
 source-interface Ethernet0/0
crypto pki trustpoint MGMT
 enrollment profile MGMT-Prof
```

Inscription et renouvellement automatiques des certificats

Le client ICP IOS peut être configuré pour effectuer l'inscription et le renouvellement automatiques à l'aide de cette commande sous la section PKI trustpoint :

```
crypto pki trustpoint MGMT
auto-enroll <percentage> <regenerate>
```

Ici, la commande **auto-enroll <pourcentage> [régénération]** indique que IOS doit effectuer le renouvellement du certificat à 80 % exactement de la durée de vie du certificat actuel.

Le mot clé **régénération** indique qu'IOS doit régénérer la paire de clés RSA connue sous le nom de paire de clés cachées lors de chaque opération de renouvellement de certificat.

Voici le comportement d'inscription automatique :

- Le moment où **l'inscription automatique** est configurée, si le point de confiance est authentifié, IOS effectue une inscription automatique au serveur situé à l'URL mentionnée dans la commande **enrollment url** sous la section PKI trustpoint ou sous le profil d'inscription.
- Le moment où un point de confiance est inscrit auprès d'un serveur PKI ou d'une autorité de certification, un compteur RENEW ou SHADOW est initialisé sur le client PKI en fonction du pourcentage **d'inscription automatique** du certificat d'identité actuel installé sous le point de confiance. Ce compteur est visible sous la commande **show crypto pki timer**. Pour plus d'informations sur les fonctions du minuteur, *reportez-vous à*
- La prise en charge des fonctionnalités de renouvellement provient du serveur PKI. En savoir

plus sur ce sujet dans
Le client ICP IOS effectue deux types de renouvellement :
Renouvellement implicite : Si le serveur PKI n'envoie pas « Renouvellement » comme fonctionnalité prise en charge, IOS effectue une inscription initiale au pourcentage d'inscription automatique défini. IOS utilise un certificat auto-signé pour signer la demande de renouvellement.Renouvellement explicite : Lorsque le serveur PKI prend en charge la fonctionnalité de renouvellement de certificat client PKI, il annonce le renouvellement comme une fonctionnalité prise en charge. IOS prend cette fonctionnalité en compte lors du renouvellement du certificat, c'est-à-dire qu'IOS utilise le certificat d'identité actif actuel pour signer la demande de certificat de renouvellement.

Soyez prudent lors de la configuration du pourcentage d'inscription automatique. Sur un client PKI donné dans le déploiement, si une condition se produit lorsque le certificat d'identité expire en même temps que le certificat d'autorité de certification émetteur, la valeur d'inscription automatique doit toujours déclencher l'opération de renouvellement [shadow] après que l'autorité de certification a créé le certificat de substitution. *Reportez-vous à la* section **Dépendances du temporisateur PKI** dans

Vérification de révocation de certificat

Un point de confiance PKI authentifié, c'est-à-dire un point de confiance PKI contenant un certificat d'autorité de certification, est capable d'effectuer la validation du certificat lors d'une négociation IKE ou SSL, lorsque le certificat homologue est soumis à une validation approfondie du certificat. L'une des méthodes de validation consiste à vérifier l'état de révocation du certificat homologue à l'aide de l'une des deux méthodes suivantes :

- Liste de révocation de certificats (CRL) : fichier contenant les numéros de série des certificats révoqués par une autorité de certification donnée. Ce fichier est signé à l'aide du certificat d'autorité de certification émetteur. La méthode CRL consiste à télécharger le fichier CRL à l'aide de HTTP ou LDAP.

- Protocole OCSP (Online Certificate Status Protocol) : IOS établit un canal de communication avec une entité appelée répondeur OCSP, qui est un serveur désigné par l'autorité de certification émettrice. Un client tel que IOS envoie une demande contenant le numéro de série du certificat est en cours de validation. Le répondeur OCSP répond avec l'état de révocation du numéro de série donné. Le canal de communication peut être établi à l'aide de n'importe quel protocole d'application/transport pris en charge, généralement HTTP.

La vérification de révocation peut être définie à l'aide de cette commande sous la section PKI trustpoint :

```
crypto pki trustpoint MGMT
 revocation-check crl ocsp none
```

Par défaut, un point de confiance est configuré pour effectuer une vérification de révocation à l'aide de crl.

Les méthodes peuvent être réordonnées et le contrôle d'état de révocation est effectué dans l'ordre défini. La méthode « none » contourne le contrôle de révocation.

Cache CRL

Avec le contrôle de révocation basé sur CRL, chaque validation de certificat peut déclencher un nouveau téléchargement de fichier CRL. Et comme le fichier CRL s'agrandit ou si le point de distribution CRL (CDP) est plus éloigné, le téléchargement du fichier pendant chaque processus de validation entrave les performances du protocole en fonction de la validation du certificat. Par conséquent, la mise en cache des LCR est effectuée pour améliorer les performances, et la mise en cache des LCR prend en compte la validité des LCR.

La validité CRL est définie à l'aide de deux paramètres temporels : **LastUpdate**, qui est la dernière fois que la liste de révocation de certificats a été publiée par l'autorité de certification émettrice, et **NextUpdate**, qui est le moment où une nouvelle version du fichier de liste de révocation de certificats est publiée par l'autorité de certification émettrice.

IOS met en cache chaque liste de révocation de certificats téléchargée tant que la liste de révocation de certificats est valide. Toutefois, dans certaines circonstances, telles que le CDP n'étant pas accessible temporairement, il peut être nécessaire de conserver la liste de révocation de certificats dans le cache pendant une longue période. Dans IOS, une liste de révocation de certificats mise en cache peut être conservée pendant 24 heures après l'expiration de la validité de la liste de révocation de certificats. Vous pouvez la configurer à l'aide de cette commande dans la section PKI trustpoint :

```
crypto pki trustpoint MGMT
 crl cache extend <0 - 1440>
!! here the value is in minutes
```

Dans certaines circonstances, telles qu'une autorité de certification émettrice révoquant des certificats au cours de la période de validité de la liste de révocation de certificats, IOS peut configurer pour supprimer le cache plus fréquemment. En supprimant prématurément la liste de révocation de certificats, IOS est forcé de télécharger la liste de révocation de certificats plus fréquemment pour maintenir le cache de la liste de révocation de certificats à jour. Cette option de configuration est disponible dans la section PKI trustpoint :

```
crypto pki trustpoint MGMT
 crl cache delete-after <1-43200>
!! here the value is in minutes
```

Enfin, IOS peut être configuré pour ne pas mettre en cache le fichier CRL à l'aide de cette commande sous la section PKI trustpoint :

```
crypto pki trustpoint MGMT
 crl cache none
```

Configuration recommandée

Un déploiement CA type avec Root-CA et une configuration Sub-CA est indiqué ci-dessous. L'exemple inclut également une configuration Sub-CA protégée par une RA.

Avec la paire de clés RSA de 2 048 bits, cet exemple recommande une configuration où :
Root-CA a une durée de vie de 8 ans
Sous-CA a une durée de vie de 3 ans
Les certificats clients sont émis pour un an, configurés pour demander un renouvellement de certificat, automatiquement.

CA RACINE - Configuration

```
crypto pki server ROOTCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=RootCA,OU=TAC,O=Cisco
lifetime crl 120
lifetime certificate 1095
lifetime ca-certificate 2920
grant auto rollover ca-cert
auto-rollover 85
database url ftp:://10.1.1.1/CA/ROOT/
database url crl ftp:://10.1.1.1/CA/ROOT/
database url crl publish ftp:://10.1.1.1/WWW/CRL/ROOT/
cdp-url http://10.1.1.1/WWW/CRL/ROOT/ROOTCA.crl
```

SUBCA sans RA - Configuration

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant auto SUBCA
auto-rollover 85
database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs


crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

SUBCA avec RA - Configuration

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant ra-auto
grant auto rollover ra-cert
auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs

crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

RA pour SUBCA - Configuration

```
crypto pki server RA-FOR-SUBCA
database level complete
database archive pkcs12 password p12password
mode ra
grant auto RA-FOR-SUBCA
auto-rollover 85
database url ftp:://10.1.1.1/CA/RA4SUB/


crypto pki trustpoint RA-FOR-SUBCA
enrollment url http://172.16.1.2:80
password ChallengePW123
subject-name CN=RA,OU=ioscs RA,OU=TAC,O=Cisco
revocation-check crl
rsakeypair RA 2048
```

Inscription de certificat

Inscription manuelle

L'inscription manuelle implique la génération de CSR hors ligne sur le client PKI, qui est copiée manuellement vers l'autorité de certification. L'administrateur signe manuellement la demande, qui est ensuite importée dans le client.

Client PKI

Configuration du client PKI :

```
crypto pki trustpoint MGMT
enrollment terminal
serial-number
ip-address none
password ChallengePW123
subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
revocation-check crl
rsakeypair PKI-Key
```

Étape 1. Commencez par authentifier le point de confiance (cette opération peut également être effectuée après l'étape 2).

```
crypto pki authenticate MGMT
!! paste the CA, in this case the SUBCA, certificate in pem format and enter “quit” at the end
in a line by itself]

PKI-Client-1(config)# crypto pki authenticate MGMT

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

-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
```

```
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
quit
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

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

Étape 2. Générez une demande de signature de certificat et transmettez le CSR à l'AC et obtenez le certificat accordé :

```
PKI-Client-1(config)# crypto pki enroll MGMT
% Start certificate enrollment ..

% The subject name in the certificate will include: CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
% The subject name in the certificate will include: PKI-Client-1.cisco.com
% The serial number in the certificate will be: 104Certificate Request follows:

MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO
```

Étape 3. Maintenant, importez le certificat accordé via le terminal :

```
PKI-Client-1(config)# crypto pki import MGMT certificate

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

-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
-----END CERTIFICATE-----
quit
% Router Certificate successfully imported
```

Serveur PKI

Étape 1. Tout d'abord, exportez le certificat de l'autorité de certification émettrice de l'autorité de certification, qui dans ce cas est le certificat SUBCA. Ceci est importé au cours de l'étape 1 ci-dessus sur le client PKI, c'est-à-dire l'authentification Trustpoint.

```
SUBCA(config)# crypto pki export SUBCA pem terminal
% CA certificate: !! Root-CA certificate
-----BEGIN CERTIFICATE-----
MIIDPDCCAiSgAwIBAgIBATANBgkqhkiG9w0BAQQFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjAwOTIx
WhcNMjMxMDE2MjAwOTIxWjAvMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ8wDQYDVQQDEwZSb290Q0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQCajfMy8gU3ZXQfKgP/wYKLB0cuywzYcDaSoNVlEvUZOWgUltCGP4CiCXyw0U0U
Zmy0rusibMV7mtkTX5muaPC0XfT98rswPiZV0qvEYpHF2YodPOUoqR3FeKj/tDbI
IikcLrfj87aeMJjCrWD888wfTN9Hw9x2QVDoSxLbzTLticXdXxwS5wxlM16GspmT
WL4fg1JRWgjRqMmOcpf716Or88XJ2N2HeWxxVFIwYQf3thHR6DgTdCgJ1uqjVE6q
1LQ1g8k81mvuCZX0uLZiTMJ69xo+Ot/RpeeE2RShxK5rh56ObQq4MT4lbIPKqIxU
lbKzWdh10NiYwjgTNwTs9GGvAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaAFPqDQXSI/Zo6YnkNme7+/SYSpy+vMB0G
A1UdDgQWBBT6g0F0iP2aOmJ5DZnu/v0mEqcvrzANBgkqhkiG9w0BAQQFAAOCAQEA
VKwqI9vpmoRh9QoOJGtOA3qEgV4eCfXdMuYxmmo0sdaBYBfQm2RhZeQ1X90vVBso
G4Wx6cJVSXCtkqZTm1IoMtya+gdhLbKqZmxc+I5/js88SrbrBIm4zj+sOoySV9kW
THEEmZjdTCWXo2wNCr23gGdnb4RqZ0FTOfoZO/2Xnpcbvhz2/K7wlDRJ5k1wrsRW
RRwsQEh4LYMFIg0aBs4gmRLZ8ytwrvvrhQTVrAA/MeomUEPhcIYESg1AlWxoCYZU
0iqKfDa9+4wEJ+PMGDhM2UV0fuP0rWitKWxecSVbo54z3VHYwwCbz2jCs8XGE61S
+XlxCZKFVdlVaMmuaZTdFg==
-----END CERTIFICATE-----
```

```
% General Purpose Certificate: !! SUBCA certificate
-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
```

Étape 2. Après l'étape 2 sur le client PKI, prenez le CSR du client et fournissez-le pour la signature sur SUBCA à l'aide de cette commande :

```
crypto pki server SUBCA request pkcs10 terminal pem
```

Cette commande suggère que le SUBCA accepte une demande de signature de certificat du terminal, et une fois accordée, les données de certificat sont imprimées au format PEM.

```
SUBCA# crypto pki server SUBCA request pkcs10 terminal pem
PKCS10 request in base64 or pem

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO
quit
% Enrollment request pending, reqId=1
```

Si l'autorité de certification est en mode d'octroi automatique, le certificat accordé est affiché au format PEM ci-dessus. Lorsque l'autorité de certification est en mode d'octroi manuel, la demande de certificat est marquée comme **en attente**, une valeur d'ID est affectée et mise en file d'attente dans la file d'attente des demandes d'inscription.

```
SUBCA#show crypto pki server SUBCA requests
Enrollment Request Database:

Router certificates requests:
ReqID  State      Fingerprint                      SubjectName
------------------------------------------------------------
1      pending    7710276982EA176324393D863C9E350E serialNumber=104+hostname=PKI-Client-
1.cisco.com,cn=PKI-Client,ou=MGMT,ou=TAC,o=Cisco
```

Étape 3. Accorder manuellement cette demande à l'aide de cette commande :

```
SUBCA# crypto pki server SUBCA grant 1
% Granted certificate:
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
-----END CERTIFICATE-----
```

> **Note**: L'inscription manuelle d'une sous-autorité de certification à une autorité de certification racine n'est pas possible.

> **Note**: Une autorité de certification désactivée en raison d'un serveur HTTP désactivé peut accorder manuellement les demandes de certificat.

Inscription à l'aide de SCEP

La configuration du client PKI est la suivante :

```
crypto pki trustpoint MGMT
enrollment url http://172.16.1.2:80
 serial-number
 ip-address none
 password 7 110A1016141D5A5E57
 subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
 revocation-check crl
 rsakeypair PKI-Key 2048
```

La configuration du serveur PKI est la suivante :

```
SUBCA# show run all | section pki server
crypto pki server SUBCA
 database level complete
 database archive pkcs12 password 7 01100F175804575D72
 issuer-name CN=SubCA,OU=TAC,O=Cisco
 lifetime crl 12
 lifetime certificate 365
 lifetime ca-certificate 1095
 lifetime enrollment-request 168
 mode sub-cs
 auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
```

Le mode par défaut d'octroi de la demande de certificat est manuel :

```
SUBCA# show crypto pki server
Certificate Server SUBCA:
   Status: enabled
   State: enabled
   Server's configuration is locked  (enter "shut" to unlock it)
   Issuer name: CN=SubCA,OU=TAC,O=Cisco
   CA cert fingerprint: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
   Server configured in subordinate server mode
   Upper CA cert fingerprint: CD0DE4C7 955EFD60 296B7204 41FB6EF6
    Granting mode is: manual
   Last certificate issued serial number (hex): 4
   CA certificate expiration timer: 21:42:27 CET Oct 17 2018
   CRL NextUpdate timer: 09:42:37 CET Oct 20 2015
   Current primary storage dir: unix:/SUB/
   Current storage dir for .crl files: unix:/SUB/
   Database Level: Complete - all issued certs written as <serialnum>.cer
   Auto-Rollover configured, overlap period 85 days
   Autorollover timer: 21:42:27 CET Jul 24 2018
```

Subvention manuelle

Étape 1. Client PKI : Dans un premier temps, qui est obligatoire, authentifiez le point de confiance sur le client PKI :

```
PKI-Client-1(config)# crypto pki authenticate MGMT
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

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

Étape 2. PKI-Client : Après l'authentification du point de confiance, le client PKI peut être inscrit à un certificat.

> **Note**: Si l'inscription automatique est configurée, le client effectue automatiquement l'inscription.

```
config terminal
crypto pki enroll MGMT
```

En coulisses, ces événements ont lieu :

- IOS recherche une paire de clés RSA nommée PKI-Key. S'il existe, il est récupéré pour demander un certificat d'identité. Si ce n'est pas le cas, IOS crée une paire de clés de 2 048 bits nommée PKI-Key, puis l'utilise pour demander un certificat d'identité.
- IOS crée une demande de signature de certificat au format PKCS10.
- IOS chiffre ensuite cette CSR à l'aide d'une clé symétrique aléatoire. La clé symétrique aléatoire est cryptée à l'aide de la clé publique du destinataire, qui est SUBCA (la clé publique de SUBCA est disponible en raison de l'authentification des points de confiance). Le CSR chiffré, la clé symétrique aléatoire chiffrée et les informations sur le destinataire sont regroupés dans les données enveloppées PKCS#7.
- Ces données enveloppées PKCS#7 sont signées à l'aide d'un certificat auto-signé temporaire lors de l'inscription initiale. Les données enveloppées PKCS#7, le certificat de signature utilisé par le client et la signature du client sont regroupés dans un paquet de données signé PKCS#7. Il s'agit du code base64, puis de l'URL. Le blob de données résultant est envoyé en tant qu'argument " message " dans l'URI HTTP envoyé à l'autorité de certification :

```
GET /cgi-bin/pkiclient.exe?operation=PKIOperation&message=MII…. HTTP/1.0
```

Étape 3. PKI-Server :

Lorsque le serveur ICP IOS reçoit la demande, il vérifie les éléments suivants :

1. Vérifie si la base de données de demande d'inscription contient une demande de certificat avec le même ID de transaction associé à la nouvelle demande.

> **Note**: Un ID de transaction est un hachage MD5 de la clé publique, pour lequel un certificat d'identité est demandé par le client.

2. Vérifie si la base de données des demandes d'inscription contient une demande de certificat avec le même mot de passe de confirmation que celui envoyé par le client.

> **Note**: Si (1) retourne true ou si (1) et (2) ensemble retournent true, alors un serveur AC est capable de rejeter la demande en raison d'une demande d'identité en double. Cependant, dans ce cas, le serveur ICP IOS remplace l'ancienne requête par la nouvelle requête.

Étape 4. PKI-Server :

Accorder manuellement les demandes sur le serveur PKI :

Pour afficher la demande :

```
show crypto pki server SUBCA requests
```

Pour accorder une demande spécifique ou toutes les demandes :

```
crypto pki server SUBCA grant <id|all>
```

Étape 5. PKI-Client :

Pendant ce temps, un client PKI démarre un compteur de POLL. Ici, IOS exécute GetCertInitial à intervalles réguliers jusqu'à ce que SCEP CertRep = GRANTED avec le certificat accordé soit reçu par le client.

Une fois le certificat accordé reçu, IOS l'installe automatiquement.