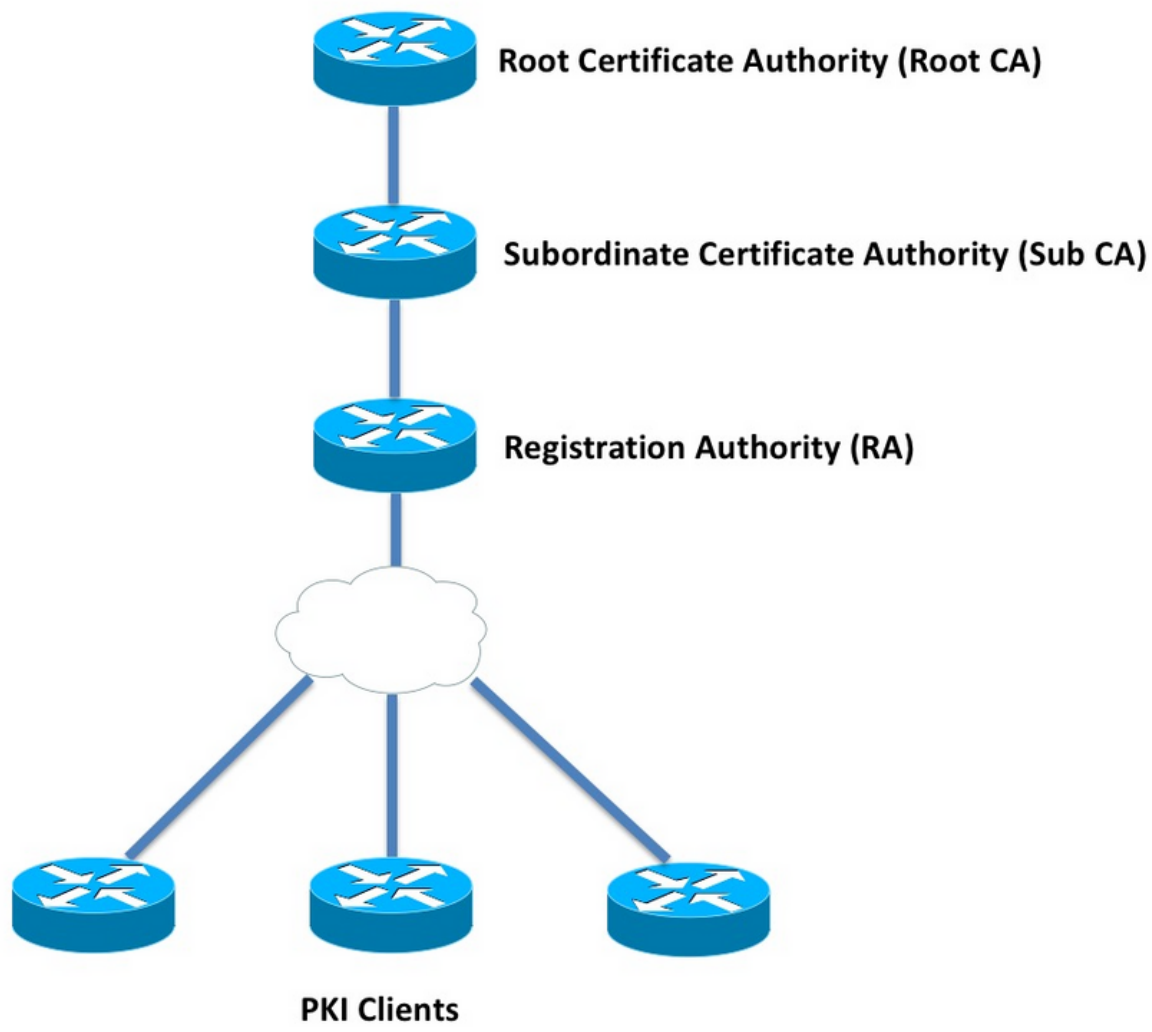
IOS PKI-implementatiegids: Aanvankelijk ontwerp en implementatie

Inhoud

Inleiding

Dit document beschrijft uitvoerig de IOS PKI Server- en clientfuncties. Het richt zich op IOS PKI eerste ontwerp en plaatsingsoverwegingen.

PKI-infrastructuur



certificaatinstantie

De certificaatinstantie (CA), die in het document ook PKI-server wordt genoemd, is een

vertrouwde entiteit die certificaten afgeeft. PKI is gebaseerd op vertrouwen, en de trust-hiërarchie begint bij Root certificaatautoriteit (Root-CA). Omdat de Root-CA bovenaan de hiërarchie staat, heeft het een zelfondertekend certificaat.

Ondergeschikte certificeringsinstantie

In PKI Trust-hiërarchie zijn alle certificeringsinstanties onder Root bekend als Subordinaat certificaatautoriteiten (Sub-CA). Het is duidelijk dat een sub-CA-certificaat wordt afgegeven door de CA, dat een hoger niveau is.

PKI stelt geen limiet op het aantal subCA's in een bepaalde hiërarchie. In een onderneming met meer dan drie niveaus van certificaathoofdenten kan het echter moeilijk worden om deze uit te voeren.

Registratieautoriteit

PKI definieert een speciale certificeringsinstantie die bekend staat als Registratieautoriteit (RA), die de PKI-klanten moet machtigen zich in te schrijven bij een bepaalde subCA of Root-CA. RA geeft geen certificaten af aan PKI-klanten, maar beslist in plaats daarvan welke PKI-client een certificaat kan of niet kan worden afgegeven door de Sub-CA of de Root-CA.

De belangrijkste rol van een RI is de validering van basiscertificaten van cliënten door de CA te ontlasten en de CA te beschermen tegen rechtstreekse blootstelling aan de cliënten. Op deze manier staat RA tussen de PKI-klanten en de CA, waardoor de CA wordt beschermd tegen iedere vorm van weigering van service-aanvallen.

PKI-client

Ieder apparaat dat om een certificaat verzoekt op basis van een ingezeten openbaar/privé-sleutelpaar om zijn identiteit aan andere apparaten aan te tonen is bekend als een PKI-cliënt.

Een PKI-client moet in staat zijn om een privaat-publiek sleutelpaar zoals RSA of DSA of ECDSA op te zetten of op te slaan.

Een certificaat is een bewijs van identiteit en geldigheid van een bepaalde openbare sleutel, mits de overeenkomstige privé-sleutel op het apparaat bestaat.

IOS PKI-server

Tabel 1. IOS PKI-serverfunctieevolutie

Functie	IOS [ISR-G1, ISR-G2]	IOS-XE [ASR1K, ISR4K]
IOS CA/PKI-server	12.3(4)T	XE 3.14.0 / 15.5(1)S
IOS PKI-servercertificaatverni euwing	12.4(1)T	XE 3.14.0 / 15.5(1)S
IOS PKI HA	15,0(1)M	NA [Impliciete Inter-RP-redundantie is beschikbaar]
IOS RA voor CA van	15.1(3)T	XE 3.14.0 / 15.5(1)S

3e partijen

Alvorens in de configuratie van de PKI-server te gaan, moet de beheerder deze kernconcepten begrijpen.

Waarschuwing bron van tijd

Eén van de fundamenten van de PKI-infrastructuur is Time. De systeemklok bepaalt of een certificaat geldig is of niet. Daarom moet de klok in IOS gezag of betrouwbaar worden gemaakt. Zonder een gezaghebbende bron van tijd kan PKI-server niet functioneren zoals verwacht, en is het sterk aanbevolen om de klok op IOS autoritair te maken met behulp van deze methoden:

NTP (Network Time Protocol)

Het synchroniseren van de systeemklok met een Tijdserver is de enige ware manier om de systeemklok betrouwbaar te maken. Een IOS router kan als NTP-client worden geconfigureerd naar een bekende en stabiele NTP-server in het netwerk:

```
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 kan ook als een NTP-server worden geconfigureerd, die de lokale systeemklok als gezaghebbend zal markeren. In kleinschalige PKI-implementatie kan de PKI-server worden geconfigureerd als een NTP-server voor de PKI-clients:

```
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
```

Hardware kloktijd als vertrouwd markeren

In IOS kan de hardwareklok worden gemarkeerd als gezaghebbend met:

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

Dit kan samen met NTP worden ingesteld en de belangrijkste reden om dit te doen is de systeemkloktijd gezaghebbend te houden wanneer een router opnieuw wordt geladen, bijvoorbeeld door een stroomuitval, en de NTP-servers zijn niet bereikbaar. In dit stadium zullen PKI-timers niet meer werken, wat op zijn beurt leidt tot fouten bij de vernieuwing van het certificaat/de doorloopbeurt. **Klokkalender-geldige** handelingen als beveiliging in dergelijke situaties.

Bij het configureren van dit apparaat is het belangrijk om te begrijpen dat de systeemkloktijd niet sync is voor het geval de systeembatterij sterft, en PKI zal beginnen met het vertrouwen op een niet-sync kloktijd. Maar het is relatief veiliger om dit te configureren dan helemaal geen gezaghebbende bron van tijd hebben.

> Opmerking: De **klokkalender-geldige** opdracht werd toegevoegd in IOS-XE versie XE 3.10.0/15.3(3)S naar voren.

Naam en domeinnaam

Het wordt aanbevolen om een hostname en een domeinnaam op Cisco IOS als één van de eerste stappen te configureren voordat u een met PKI samenhangende services configureren. De routerhostname en domeinnaam worden in de volgende scenario's gebruikt:

- Standaard RSA (key-pair) naam wordt afgeleid door de hostname en de domeinnaam te combineren
- Bij het inschrijven voor een certificaat bestaat de standaard onderwerpregel uit de eigenschap hostname en een niet-gestructureerde naam, die hostname en domeinnaam bij elkaar is.

Zoals voor PKI Server worden hostname en domeinnaam niet gebruikt:

- Standaard key-paarnaam is dezelfde als die van de PKI server name
- Standaard Onderwerp-naam bestaat uit GN, dat hetzelfde is als de PKI-servernaam.

De algemene aanbeveling is om een geschikte hostname en een domeinnaam te configureren.

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

HTTP-server

IOS PKI Server is alleen ingeschakeld als HTTP Server is ingeschakeld. Het is belangrijk om op te merken dat, als de PKI server uitgeschakeld is omdat HTTP server uitgeschakeld is, deze offline verzoeken kan blijven indienen [via terminal]. HTTP Server-mogelijkheid is vereist om SCEP-verzoeken te verwerken en SCEP-antwoorden te verzenden.

IOS HTTP Server is ingeschakeld met behulp van:

```
ip http server
```

En de standaard HTTP server poort kan worden gewijzigd van 80 naar elk geldig poortnummer met behulp van:

```
ip http port 8080
```

HTTP-Max-verbinding

Eén van de knelpunten bij het implementeren van IOS als PKI server met SCEP is Maximum aantal gelijktijdige HTTP-verbindingen en gemiddelde HTTP-verbindingen per minuut.
Op dit moment zijn de maximum gelijktijdige verbindingen op een IOS HTTP Server standaard beperkt tot 5 en kan worden verhoogd tot 16, wat sterk aanbevolen wordt in een implementatie op middellange schaal:

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

Deze IOS-installaties maken maximale gelijktijdige HTTP-verbindingen tot 1000 mogelijk:

- Universal K9 IOS met uck9 licentieserver

De CLI wordt automatisch gewijzigd om een numeriek argument tussen 1 en 1000 te aanvaarden

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

IOS HTTP-server maakt 80 verbindingen per minuut mogelijk [580 in het geval van IOS-releases waar Max gelijktijdige sessies van HTTP kunnen worden verhoogd naar 1000] en wanneer deze limiet binnen een minuut wordt bereikt, start IOS HTTP-luisteraar de inkomende HTTP-verbindingen door de luisteraar 15 seconden ingedrukt te houden. Dit leidt tot verzoeken van de clientverbinding die worden ingetrokken omdat de **limiet van de** TCP**-verbindingswachtrij is bereikt**.
Meer informatie hierover is hier te vinden

RSA-sleutelpaar

RSA zeer belangrijk-paar voor de functionaliteit van de PKI Server op IOS kan auto-gegenereerd of handmatig worden gegenereerd.
Tijdens het configureren van een PKI Server maakt IOS automatisch een Trustpoint met dezelfde naam als de PKI Server om het PKI Server certificaat op te slaan.

PKI Server RSA Key-paar handmatig genereren:

Stap 1. Maak een RSA Key-paar met dezelfde naam als de PKI-server:

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

Stap 2. Voordat u de PKI-server activeert, moet u het Trustpunt van de PKI-server wijzigen:

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

> Opmerking: RSA Key-paarmodulus waarde die onder het PKI Server-betrouwbaarheidspunt wordt genoemd, wordt niet in aanmerking genomen tot IOS op 15.4(3)M4, en dit is een bekend voorbehoud. De standaard sleutelmodulus is 1024 bits.

Automatisch genereren PKI Server RSA Key-paar:

Wanneer het inschakelen van de PKI Server, genereert IOS automatisch een RSA zeer belangrijk-paar met dezelfde naam als die van de PKI Server, en de zeer belangrijke modulusgrootte is 1024 bits.

Vanaf IOS op 15.4(3)M5 leidt deze configuratie tot een RSA zeer belangrijk-paar met **<LABEL>** aangezien de naam en de belangrijkste-kracht dezelfde zullen zijn als de gedefinieerde **<MOD>**-modulus.

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

spoiler

CSCu73408 IOS PKI-server zou een niet-standaard belangrijke grootte voor het rollover-certificaat moeten toestaan.

CSCu73408 IOS PKI-server zou een niet-standaard belangrijke grootte voor het kantelhokje moeten toestaan.

De huidige industriestandaard is een minimum van 2048 bits RSA key-pair te gebruiken.

Aandacht voor automatische kanteltimer

Op dit moment genereert IOS PKI Server standaard geen rollover-certificaat en moet het expliciet worden ingeschakeld onder de PKI-server met **de** opdracht **auto-**rollover **<dagen-voor-vervaldatum>**. Meer informatie over het verlengen van het certificaat is te vinden in

Deze opdracht specificeert hoeveel dagen vóór het PKI Server/CA certificaat verstrijken wanneer IOS een CA-certificaat voor het omversen maakt. Merk op dat het overloopcertificaat CA is geactiveerd zodra het huidige actieve CA-certificaat is verlopen. De standaardwaarde is momenteel 30 dagen. Deze waarde dient op een redelijke waarde te worden ingesteld, afhankelijk

van de levensduur van het CA-certificaat, en dit heeft op zijn beurt invloed op de configuratie van de automatische inschrijving op de PKI-client.

> Opmerking: De automatische omlooptimer moet altijd voor de automatische inschrijving op de client starten tijdens het rollover van het certificaat van client [bekend als]

CRL-overwegingen

IOS PKI-infrastructuur ondersteunt twee manieren om CRL te distribueren:

CRL naar een HTTP-server publiceren

IOS PKI Server kan worden geconfigureerd om het CRL-bestand naar een specifieke locatie op een HTTP-server te publiceren met behulp van deze opdracht onder de PKI-server:

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

En de PKI server kan worden geconfigureerd om deze CRL locatie in alle PKI client-certificaten in te sluiten met deze opdracht onder de PKI-server:

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

SCEP GetCRL-methode

IOS PKI Server slaat automatisch het CRL-bestand op de specifieke database-locatie op, die standaard nvram is en sterk wordt aanbevolen om een kopie op een SCP/FTP/TFTP-server te bewaren met deze opdracht onder de PKI-server:

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

Standaard is de CDP-locatie niet in de PKI-clientcertificaten ingebouwd. Als de IOS PKI-clients zijn ingesteld om de herroepingscontrole uit te voeren, maar het certificaat dat wordt gevalideerd, bevat geen CDP ingesloten en het validerende CA-trustpunt is ingesteld bij de CA-locatie (met behulp van http://<CA-server-IP of FQDN>), valt IOS standaard terug naar de SCEP-gebaseerde GetCRL-methode.
SCEP GetCRL voert CRL-herkenning uit door HTTP GET op deze URL uit te voeren:

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

> Opmerking: In IOS CLI, alvorens in te gaan **?** drukt u op **Ctrl + V** key-sequentie.

IOS PKI Server kan deze URL ook als de CDP-locatie insluiten. Het voordeel hiervan is tweeledig:

- Dit waarborgt dat alle niet-IOS SCEP-gebaseerde PKI-clients CRL-herkenning kunnen uitvoeren.
- Zonder een ingesloten CDP worden IOS SCEP GetCRL-verzoekberichten ondertekend (met behulp van een tijdelijk zelf-ondertekend certificaat) zoals gedefinieerd in het SCEP-ontwerp. Verzoeken om CRL-ophalen hoeven echter niet te worden getekend en door de CDP-URL in te sluiten voor GetCRL-methode kunnen ondertekening van CRL-verzoeken worden vermeden.

Levensduur van CRL

IOS PKI Server kan de CRL levensduur van deze opdracht onder de PKI Server worden gecontroleerd:

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

De waarde is in uren. De levensduur van het CRL wordt standaard ingesteld op 6 uur. Afhankelijk van hoe vaak de certificaten worden ingetrokken, verhoogt het afstemmen van de CRL-levensduur op een optimale waarde de CRL-herkenningsprestaties in het netwerk.

Databaseverslagen

IOS PKI Server gebruikt nvram als de standaard database locatie en het wordt sterk aanbevolen om een FTP- of TFTP- of SCP-server als database locatie te gebruiken. Standaard maakt IOS PKI Server twee bestanden:

- <Server-naam>.ser - Dit bevat het laatste serienummer dat door de CA in hex is verleend. Het bestand is in gewone tekstindeling en bevat deze informatie:
  db_versie = 1
  last_seri= 0x4
- <Server-naam>.crl - Dit is het gecodeerde CRL-bestand dat door de CA is gepubliceerd

IOS PKI Server slaat informatie in het gegevensbestand op 3 configureerbare niveaus op:

- Minimaal - Dit is het standaardniveau, en op dit niveau wordt er geen bestand in de database aangemaakt, en dus is er geen informatie beschikbaar op de CA server over de in het verleden toegekende clientcertificaten.
- Namen - Op dit niveau maakt IOS PKI-server een bestand met de naam <Serial-Number>.cnm voor elk uitgegeven client-certificaat, waarin de naam <Serial-Number> verwijst naar het serienummer van het afgegeven client-certificaat en dit nm bestand bevat onderwerpnaam en de vervaldatum van het client-certificaat.

- Complete - Op dit niveau maakt IOS PKI Server twee bestanden voor elk uitgegeven client-certificaat:
- <serienummer> 100 nm
- <serienummer>.crt

Hier is het crt-bestand het client-certificaatbestand, dat DER is gecodeerd.

Deze punten zijn belangrijk:

- Voordat u een client-certificaat afgeeft, verwijst de IOS PKI Server naar <Server-naam>.ser om het serienummer van het certificaat te bepalen en af te leiden.
- Als het Databaseniveau op Namen of Complete is ingesteld, moeten <Serial-Number>.cnm en <Serial-number>.crt naar de database worden geschreven voordat het toegekende/afgegeven certificaat naar de client wordt verzonden
- Wanneer de database URL op Namen of Complete is ingesteld, moet de database URL voldoende ruimte hebben om de bestanden op te slaan. Vandaar dat de aanbeveling is om een externe bestands server [FTP of TFTP of SCP] te configureren als de database-URL.
- Als externe database URL is ingesteld, is het absoluut nodig om ervoor te zorgen dat de bestandsserver bereikbaar is tijdens het proces van certificaatsubsidie, wat anders de CA Server zou markeren als uitgeschakeld. En handmatige interventie is vereist om de CA server terug online te brengen.

Databaseverslag

Tijdens het implementeren van een PKI Server, is het belangrijk om de mislukkingsscenario's in overweging te nemen en voor te bereiden, mocht er een hardware storing zijn. Er zijn twee manieren om dit te bereiken:

1. Redundantie
   In dit geval fungeren twee apparaten of verwerkingseenheden als Active-Standby om redundantie te bieden.
   Hogere beschikbaarheid van IOS PKI-servers kan worden bereikt met twee HSRP-enabled ISR-routers [ISR G1 en ISR G2] zoals uitgelegd in

   IOS XE-gebaseerde systemen [ISR4K en ASR1k] hebben geen optie voor apparaatredundantie beschikbaar. In ASR1k Inter-RP redundantie is echter standaard beschikbaar.
2. Toetsen en bestanden voor CA-server archiveren
   IOS biedt een mogelijkheid om het PKI Server Key-pair en het certificaat te archiveren. De archivering kan worden uitgevoerd met twee typen bestanden:
   PEM - IOS maakt PEM-geformatteerde bestanden om RSA Public Key, Encrypted RSA Private Key, CA Server-certificaat op te slaan. Rollover Key-paar en certificaten worden automatisch gearchiveerdPKCS12 - IOS maakt één PKCS12-bestand met het CA-servercertificaat en de bijbehorende RSA Private Key versleuteld met een wachtwoord.

Het archiveren van de databank kan worden ingeschakeld met behulp van deze opdracht onder de PKI-server:

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

Het is ook mogelijk om de gearchiveerde bestanden op een afzonderlijke server op te slaan, mogelijk met behulp van een beveiligd protocol (SCP) met behulp van de volgende opdracht onder de PKI-server:

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

Van alle bestanden in de database behalve de gearchiveerde bestanden en het .SS-bestand, zijn alle andere bestanden in duidelijke tekst en vormen ze geen echte bedreiging indien verloren, en kunnen ze daarom op een afzonderlijke server worden opgeslagen zonder veel overhead te veroorzaken bij het schrijven van de bestanden, bijvoorbeeld een TFTP-server.

IOS als sub-CA

IOS PKI Server neemt standaard de rol van Root CA in beslag. Om een ondergeschikte PKI-server (sub-CA) te configureren schakelt u deze opdracht eerst in onder het configuratiescherm van de PKI-server (voordat u de PKI-server inschakelen):

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

Gebruik dit om de URL van de Root-CA te configureren onder het vertrouwde punt van de PKI Server:

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

Het in werking stellen van deze PKI Server veroorzaakt nu deze gebeurtenissen:

- PKI Server trustpoint is gecertificeerd om het Root-CA certificaat te installeren.
- Nadat de Root-CA voor authentiek is verklaard, genereert IOS een CSR voor de Subordinaat-CA [x509 basisbeperking die CA:TRUE-vlag bevat] en stuurt deze naar Root-CA

Ongeacht de subsidiemodus die op de Root-CA is ingesteld, zet IOS de CA (of RA) certificaatverzoeken in de wachtrij. Een beheerder moet de CA-certificaten handmatig verlenen. U kunt de aanvraag voor een certificaat en de aanvraag als volgt weergeven:

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

Zo willigt u het verzoek in:

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

- Wanneer u dit gebruikt, wordt in het daaropvolgende SCEP POLL (GetCertInitial) gedownloads van het sub-CA-certificaat en op de router geïnstalleerd, waardoor de subordinaire PKI-server kan worden gedownload

IOS als RA

IOs PKI Server kan als Registratie-instantie worden ingesteld voor een bepaalde subcoördinaat of Root CA. Om de PKI Server als registratieautoriteit te configureren schakelt u deze opdracht eerst in onder het configuratiescherm van de PKI-server (voordat u de PKI-server inschakelen):

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

Specificeer vervolgens de URL van de CA onder het vertrouwde punt van de PKI Server. Dit geeft aan welke CA door de RA wordt beschermd:

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

Een Registratieautoriteit geeft geen certificaten af, zodat de configuratie van de **uitgevende instelling** volgens de RI niet vereist is en niet effectief is, zelfs niet indien deze is geconfigureerd. De onderwerpregel van een RA wordt onder RA trustpoint ingesteld met behulp van **onderwerp-name** opdracht. Het is belangrijk om **OU = ioscs RA** te configureren als onderdeel van de onderwerpregel, zodat de IOS CA de IOS RA kan identificeren, d.w.z. om de certificaatverzoeken te identificeren die door de IOS RA zijn geautoriseerd.

IOS kan fungeren als Registratie-instantie voor CA's van derden zoals Microsoft CA, en om compatibel te blijven moet IOS RA worden ingeschakeld met deze opdracht onder het configuratiescherm PKI Server (voordat u de PKI Server toelaat):

```
mode ra transparent
```

In de standaard RA modus, tekent IOS de clientverzoeken [PKCS#10] met behulp van het RA certificaat. Deze handeling geeft de IOS PKI Server aan dat het certificaatverzoek is goedgekeurd door een RA.

Met transparante RA-modus stuurt IOS de clientverzoeken in hun oorspronkelijke indeling door zonder het RA-certificaat te introduceren en dit is compatibel met Microsoft CA als een bekend voorbeeld.

IOS PKI-client

Eén van de belangrijkste configuratie-entiteit in IOS PKI-client is een schaalpunt. De parameters voor de configuratie van het trustpunt worden in dit gedeelte uitvoerig uitgelegd.

Waarschuwing bron van tijd

Zoals eerder is opgemerkt, is ook de PKI-cliënt een gezaghebbende bron van tijd. IOS PKI-client kan als een NTP-client worden geconfigureerd met behulp van deze configuratie:

```
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
```

Naam en domeinnaam

Een algemene aanbeveling is om een hostname en een domeinnaam op de router te configureren:

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

RSA-toetsuur

In IOS PKI-client kan RSA Key-pair voor een bepaalde vertrouwenspuntinschrijving automatisch gegenereerd worden of handmatig gegenereerd worden.

Automatische RSA-sleutelgeneratieproces omvat het volgende:

- IOS-paar met standaard 152-bits RSA-toetsuitdrukking
- De automatisch gegenereerde key-paarnaam is hostname.domein-name, wat de device hostname is in combinatie met de device domeinnaam
- Het automatisch gegenereerde sleutelpaar wordt niet gemarkeerd als exporteerbaar.

Automatische RSA-sleutelgeneratieproces omvat het volgende:

- Optioneel kan een RSA Key-paar van een geschikte sterkte handmatig worden gegenereerd met behulp van:
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  Hier, LABEL - de naam van het RSA-sleutelpaar
  MOD - RSA-sleutelmodulus of sterkte in bits tussen 360 en 4096, die van oudsher 512, 1024, 2048 of 4096 is.
  Het voordeel van het handmatig genereren van het RSA key-paar is de mogelijkheid om het key-pair-als exportbaar te markeren, wat op zijn beurt betekent dat het identiteitsbewijs volledig geëxporteerd wordt, wat dan op een ander apparaat hersteld kan worden. We moeten echter wel de veiligheidsimplicaties van deze actie begrijpen.
- Een RSA-key-pair is gekoppeld aan een betrouwbaar punt voordat u deze opdracht gebruikt

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

Hier bestaat een RSA Key-pair met de naam <LABEL> al, en wordt deze optie opgenomen tijdens de inschrijving als betrouwbaar.

Als een RSA Key-pair met de naam <LABEL> niet bestaat, dan wordt één van de volgende actie uitgevoerd tijdens de inschrijving:

- Als geen <MOD> argument wordt doorgegeven, wordt er een toetsenbord met 512 bits met de naam <LABEL> gegenereerd.

- Als één <MOD>-argument wordt doorgegeven, wordt er een <MOD>-bits algemeen doel-toetsenbord met de naam <LABEL> gegenereerd

- als twee <MOD>-argumenten worden doorgegeven, wordt één <MOD>-bits key-pair voor handtekening en één <MOD>-bits coderingskey-paar met de naam <LABEL>

gegenereerd

Trustpunt

Een betrouwbaar punt is een abstracte container om een certificaat in IOS te houden. Een enkel trustpoint is in staat twee actieve certificaten op elk moment op te slaan:

- Een CA-certificaat - Het laden van een CA-certificaat in een bepaald trustpunt is bekend als een betrouwbaar verificatieproces.
- Een identiteitsbewijs dat is afgegeven door de CA - Lading of Importeer van een ID - certificaat in een bepaald trustpunt staat bekend als trustpuntinschrijvingsproces.

Een vertrouwenspuntconfiguratie staat bekend als een trustbeleid, en dit definieert het volgende:

- Welk CA-certificaat is geladen in het trustpunt?
- Welke CA doet de trustpoint?
- Hoe registreert de IOS het trustpunt?
- Hoe wordt een door de genoemde CA afgegeven certificaat [geladen in het trustpunt] gevalideerd?

De belangrijkste onderdelen van een vertrouwenspunt worden hier toegelicht.

Invoermodus

Een trustpoint inschrijvingsmodus, die ook de trustpoint authenticatiemodus definieert, kan worden uitgevoerd via drie hoofdmethoden:

1. Terminalinschrijving - handmatige methode voor het uitvoeren van betrouwbaarheidskeuringen en certificatie met behulp van kopieerpasta in de CLI-terminal.
2. SCEP-inschrijving - Trustpuntverificatie en -inschrijving met behulp van SCEP via HTTP.
3. Inschrijvingsprofiel - Hier worden de authenticatie- en inschrijvingsmethoden afzonderlijk gedefinieerd. Samen met terminaal- en SCEP-inschrijvingsmethoden bieden inschrijvingsprofielen een optie om HTTP/TFTP-opdrachten te specificeren om het ophalen van bestanden van de server uit te voeren, die is gedefinieerd met behulp van een verificatie- of inschrijvingsregel onder het profiel.

Bron-interface en VRF

Verificatie en inschrijving via HTTP (SCEP) of TFTP (Encapsulation Profile) gebruiken het IOS-bestandssysteem om bestanden in/uit-bewerkingen uit te voeren. Deze pakketuitwisselingen kunnen uit een specifieke broninterface en een VRF komen.

In het geval van klassieke trustpuntconfiguratie, wordt deze functionaliteit ingeschakeld met behulp van **broninterface** en vrf-sub-opdrachten onder het trustpunt.

In het geval van inschrijvingsprofielen, **broninterface** en **inschrijving |** Opdrachten **van <tftp://Server-location> vrf-naam>** bieden dezelfde functionaliteit.

Voorbeelden:

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

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

of

```
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
```

Automatische inschrijving en vernieuwing van certificaten

IOS PKI-client kan worden geconfigureerd voor automatische inschrijving en vernieuwing onder deze opdracht in de PKI-sectie:

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

Hier staat, de **auto-inschrijving <percentage> [regenereert]** opdracht dat IOS certificatievernieuwing moet uitvoeren op precies 80% van de levensduur van het huidige certificaat.

Het sleutelwoord **regenereert** staten dat IOS het belangrijkste-paar van RSA zou moeten regenereren dat als schaduw zeer belangrijk-paar bekend is tijdens elke certificaat vernieuwingsoperatie.

Dit is het automatische inschrijvingsgedrag:

- Op het moment dat **de automatische inschrijving** wordt geconfigureerd, als het vertrouwde point is geauthentiseerd, zal IOS een automatische inschrijving op de server uitvoeren die zich op de URL bevindt die als deel van de **inschrijving url**-opdracht onder de PKI-trustsectie of onder het inschrijvingsprofiel wordt vermeld.
- Zodra een trustpoint wordt ingevoerd met een PKI-server of een CA, wordt een RENEW of een SHADOW-timer op de PKI-client geïnitialiseerd op basis van het **automatische inschrijvingspercentage** van het huidige identiteitsbewijs dat onder het trustpunt is geïnstalleerd. Deze timer is zichtbaar onder opdracht **voor crypto-**pki. Meer over de timer functies *verwijzen naar*
- Ondersteuning van herkiesbaarheid komt van de PKI-server. Meer hierover in
  IOS PKI-client voert twee soorten vernieuwing uit:
  Impliciete vernieuwing: Als de PKI server "Renewal" niet als ondersteunde mogelijkheid verstuurt, voert IOS een eerste inschrijving uit bij het gedefinieerde percentage auto-inschrijving. d.w.z. IOS gebruikt een zichzelf ondertekend certificaat om het vernieuwingsverzoek te ondertekenen.Expliciete vernieuwing: Wanneer de PKI Server de optie voor vernieuwing van het PKI-clientcertificaat ondersteunt, wordt "Renewal" als een ondersteunde mogelijkheid geadverteerd. IOS houdt rekening met deze mogelijkheid tijdens de vernieuwing van het certificaat, d.w.z. IOS gebruikt het huidige actieve identiteitsbewijs om

de aanvraag voor het vernieuwingscertificaat te ondertekenen.

Let goed op bij het configureren van het percentage automatische inschrijving. Op elke PKI-cliënt in de installatie, indien een voorwaarde ontstaat wanneer het identiteitsbewijs op hetzelfde tijdstip als het certificaat van uitgifte van CA afloopt, moet de waarde van de auto-inschrijving altijd de [schaduw] vernieuwingsoperatie starten nadat de CA het certificaat van wederomloop heeft aangelegd. *Raadpleeg de* sectie **PKI-timerafhankelijkheden** in

Revocatie van het certificaat

Een gewaarmerkt PKI-trustpunt, d.w.z. een PKI-trustpoint met een CA-certificaat, kan certificatie uitvoeren tijdens een IKE- of SSL-onderhandeling, waarbij het peer-certificaat aan een grondige certificatie is onderworpen. Een van de validatiemethoden is de controle van de status van peer-certificaatherroeping met behulp van een van de volgende twee methoden:

- certificaatherroeping (CRL) - Dit is een bestand dat de serienummer van de certificaten bevat die door een bepaalde CA zijn ingetrokken. Dit bestand is ondertekend met behulp van het CA-certificaat dat wordt afgegeven. CRL-methode houdt in dat het CRL-bestand met HTTP of LDAP wordt gedownload.

- Online Certificate Status Protocol (OCSP) - IOS stelt communicatiekanaal in met een entiteit die als OCSP Responder wordt genoemd, en die een aangewezen Server door de CA van uitgifte is. Een client als IOS stuurt een aanvraag met het serienummer van het certificaat dat wordt gevalideerd. De OCSP-responder reageert met de herroepingsstatus van het opgegeven serienummer. Het communicatiekanaal kan worden opgezet met behulp van elk ondersteund toepassing/transportprotocol, dat doorgaans HTTP is.

De herroepingscontrole kan worden gedefinieerd met behulp van deze opdracht onder de PKI-trustpuntsectie:

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

Standaard wordt een trustpunt ingesteld om de herroepingscontrole met behulp van crl uit te voeren.

De methoden kunnen opnieuw worden geordend en de controle van de herroepingsstatus wordt in de gedefinieerde volgorde uitgevoerd. De methode "geen" passeert de herroepingscontrole.

CRL-cache

Met een op CRL gebaseerde herroeping-check kan elke certificatie een nieuw CRL-bestand downloaden. En omdat het CRL-bestand groter wordt of als het CRL-distributiepunt (CDP) verder weg is, belemmert het downloaden van het bestand tijdens elk valideringsproces de prestaties van het protocol afhankelijk van certificatie. Daarom wordt de CRL-caching uitgevoerd om de prestaties te verbeteren, en het caching van het CRL houdt rekening met de CRL-geldigheid.

CRL-validiteit wordt gedefinieerd aan de hand van twee tijdparameters: **LaatsteUpdate**, de laatste keer dat het CRL door het uitgevende CA is gepubliceerd, en **NextUpdate**, is het tijdstip waarop een nieuwe versie van het CRL-bestand door het uitgevende CA wordt gepubliceerd.

IOS slaat elke gedownload CRL op zolang het CRL geldig is. Onder bepaalde omstandigheden, zoals het feit dat het CDP niet tijdelijk bereikbaar is, kan het echter noodzakelijk zijn het CRL in het cache gedurende een langere periode te bewaren. In IOS kan een gecached CRL tot 24 uur na het verlopen van de CRL-geldigheid behouden blijven en dit kan worden geconfigureerd met deze opdracht in het PKI-betrouwbaarheidsgedeelte:

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

Onder bepaalde omstandigheden zoals het uitgeven van CA intrekken van certificaten binnen de CRL geldigheidsperiode kan IOS bevestigen om het cache vaker te verwijderen. Door CRL voortijdig te verwijderen, wordt IOS gedwongen om CRL vaker te downloaden om het CRL cachegeheugen up-to-date te houden. Deze configuratieoptie is beschikbaar in het PKI-betrouwbaarheidsgedeelte:

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

En tenslotte kan IOS worden gevormd om het CRL dossier niet in het voorgeheugen op te stellen met deze opdracht onder de sectie van het PKI trustpoint:

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

Aanbevolen configuratie

Een typische CA-plaatsing met Root-CA en een Sub-CA configuratie is zoals hieronder. Het voorbeeld omvat ook een sub-CA-configuratie die door een RA wordt beschermd.

Met 2048 bits RSA Key-pair over de hele groep raadt dit voorbeeld een instelling aan waarin:
Root-CA heeft een levensduur van 8 jaar
SubCA heeft een levensduur van 3 jaar
Clientcertificaten worden voor een jaar afgegeven, die automatisch worden gevormd om een verlenging van het certificaat aan te vragen.

ROOT CA - configuratie

```
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 zonder RA - configuratie

```
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 met RA - configuratie

```
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 voor SUBCA - configuratie

```
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
```

certificaatinschrijving

Handmatige inschrijving

Handmatige inschrijving omvat offline CSR-generatie op de PKI-client, die handmatig naar de CA wordt gekopieerd. De beheerder tekent het verzoek handmatig, dat dan in de client wordt geïmporteerd.

PKI-client

PKI-clientconfiguratie:

```
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
```

Stap 1. bevestig eerst het Trustpoint (dit kan ook na stap 2 worden uitgevoerd).

```
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
```

Stap 2. Generate certificaataanvraag en breng de CSR naar de CA en verkrijg het toegekende certificaat:

```
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

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

Redisplay enrollment request? [yes/no]: no
```

Stap 3. Voer nu het toegekende certificaat via de terminal in:

```
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
```

PKI-server

Stap 1. Exporteer eerst het certificaat van uitgifte van CA van de CA, dat in dit geval het SUBCA-certificaat is. Dit wordt ingevoerd in stap 1 hierboven op de PKI-client, d.w.z. Trustpoint-verificatie.

```
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-----
```

Stap 2. Na Stap 2 op de PKI-client neemt u de CSR van de client en geeft u deze op voor het ondertekenen van de SUBCA met deze opdracht:

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

Deze opdracht suggereert dat de SUBCA een certificaatondertekeningsverzoek van de terminal aanvaardt en dat de certificeringsgegevens, zodra ze worden toegekend, in PEM-indeling worden afgedrukt.

```
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
```

Als de CA in de automatische subsidiemodus staat, wordt het toegekende certificaat in bovenstaande PEM-indeling weergegeven. Wanneer CA in handmatige toekenningsmodus is, wordt het certificaatverzoek gemarkeerd als **hangend**, krijgt u een id waarde toegewezen en wordt u in de wachtrij voor inschrijvingsaanvraag geplaatst.

```
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
```

Stap 3. Geef dit verzoek handmatig toe met deze opdracht:

```
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-----
```

Opmerking: Handmatige inschrijving van een sub-CA aan een Root-CA is niet mogelijk.

Opmerking: Een CA in een gehandicapte staat wegens gehandicapte HTTP server kan de certificaatverzoeken handmatig verlenen.

Inschrijving met SCEP

PKI-clientconfiguratie is:

```
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
```

PKI-serverconfiguratie is:

```
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/
```

De standaardmodus voor het toekennen van een certificaataanvraag is handmatig:

```
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
```

Handmatige beurs

Stap 1. PKI-client: Als eerste stap, die verplicht is, moet je het vertrouwenspunt op de PKI-client echt maken:

```
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.
```

Stap 2. PKI-client: Na de verificatie van het vertrouwen kan de PKI-cliënt voor een certificaat worden ingeschreven.

> Opmerking: Als de automatische inschrijving is geconfigureerd zal de client automatisch inschrijving uitvoeren.

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

Achter de schermen vinden deze gebeurtenissen plaats:

- IOS zoekt naar een RSA-sleutelpaar met de naam PKI-Key. Indien deze bestaat, wordt zij ingehaald om een identiteitsbewijs te kunnen aanvragen. Als niet, maakt IOS een 2048 bit key-pair met de naam PKI-Key, en gebruikt deze voor het aanvragen van een identiteitsbewijs.
- IOS maakt een certificaataanvraag in de PKCS10-indeling.
- IOS versleutelt vervolgens deze CSR met behulp van een willekeurige symmetrische toets.

De willekeurige symmetrische sleutel is versleuteld met behulp van de openbare sleutel van de ontvanger, die de SUBCA is (de openbare sleutel van SUBCA is beschikbaar vanwege de authenticatie van het vertrouwenspunt). De gecodeerde CSR, de gecodeerde willekeurige symmetrische sleutel en de ontvangende informatie worden samengevoegd in PKCS#7 ingekapselde gegevens.

- Deze PKCS#7 enveloped data wordt getekend met een tijdelijk zelfgetekend certificaat tijdens de eerste inschrijving. De gegevens van PKCS#7 zijn ingekapseld, het ondertekeningscertificaat dat door de cliënt wordt gebruikt en de handtekening van de cliënt worden samengevoegd in een door PKCS#7 ondertekend gegevenspakket. Dit is base64 gecodeerd en dan URL gecodeerd. Het resulterende blok gegevens wordt verzonden als "bericht" argument in HTTP URI dat naar CA wordt verzonden:

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

Stap 3. PKI-server:

Wanneer de IOS PKI Server het verzoek ontvangt, controleert het deze:

1. Controleert of de database van de inschrijvingsaanvraag een certificaataanvraag bevat met dezelfde transactieid die aan het nieuwe verzoek is gekoppeld.

> Opmerking: Een transactie-id is een MD5-hash van de openbare sleutel, waarvoor de cliënt om een identiteitsbewijs verzoekt.

2. Controleert of de database van de inschrijvingsaanvraag een certificaataanvraag bevat met hetzelfde uitdaging-wachtwoord als die welke door de client wordt verstuurd.

> Opmerking: Als (1) zowel waar als beide (1) en (2) samen teruggeven waar, dan kan een CA server het verzoek afwijzen op grond van een dubbel identiteitsverzoek. In zo'n geval vervangt IOS PKI Server het oudere verzoek echter door het nieuwere verzoek.

Stap 4. PKI-server:

Geef de aanvragen handmatig op de PKI-server:

U kunt het verzoek als volgt weergeven:

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

Om een specifiek verzoek of alle verzoeken in te dienen:

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

Stap 5. PKI-client:

Intussen start een PKI-client een POLL-timer. Hier voert IOS GetCertInitiatie uit met regelmatige tussenpozen tot SCEP CertRep = GRANTED samen met het toegekende certificaat door de client wordt ontvangen.

Zodra het toegekende certificaat wordt ontvangen, installeert IOS automatisch het.