# Guia de distribuição IO PKI: Projeto inicial e desenvolvimento

## Índice

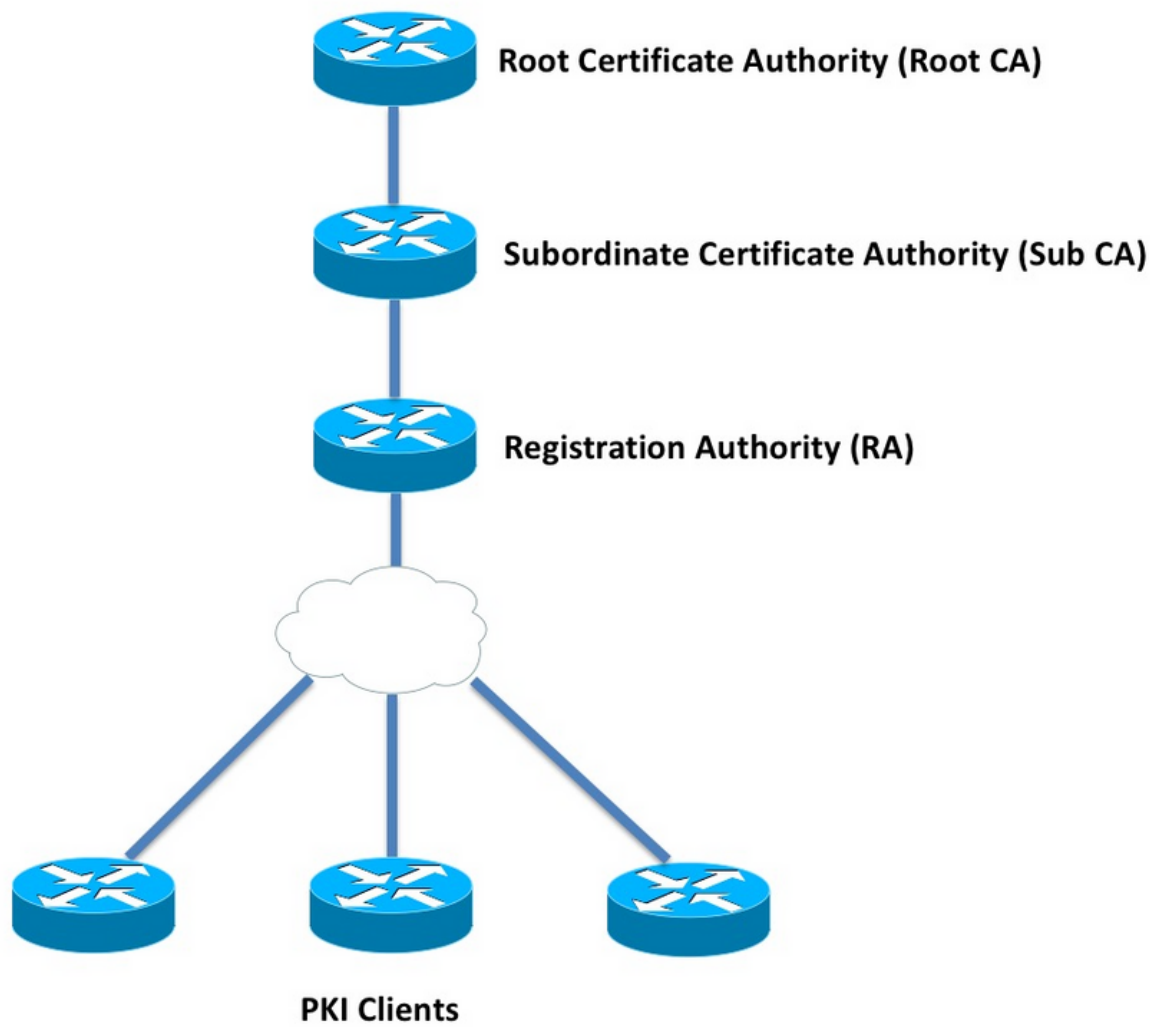

Introdução
Infraestrutura PKI
Certificate Authority
Certificate Authority subordinado
Autoridade de registro
Cliente PKI
Servidor PKI IO
Fonte competente de tempo
Hostname e Domain Name
Server do HTTP
Par de chaves RSA
consideração do temporizador do Auto-derrubamento
Considerações CRL
Publique o CRL a um Server do HTTP
Método SCEP GetCRL
Vida do CRL
Considerações do base de dados
Arquivo do base de dados
IO como Secundário-CA
IO como o RA
Cliente IO PKI
Fonte competente de tempo
Hostname e Domain Name
Par de chaves RSA
Ponto confiável
Modo do registro
Interface de origem e VRF
Certificado de registro e renovação automáticos
Revogação-verificação do certificado
Esconderijo CRL
Configuração recomendada
CA RAIZ - Configuração
SUBCA sem RA - Configuração
SUBCA com RA - Configuração
RA para SUBCA - Configuração
Certificado de registro
Inscrição manual
Cliente PKI

[Servidor PKI](#)
[Registro usando o SCEP](#)
[Concessão manual](#)
[Auto-concessão incondicional](#)
[Auto-concessão autorizada](#)
[Registro usando o SCEP através do RA](#)
[a Auto-concessão RA autorizou pedidos](#)
[certificado do derrubamento da Auto-concessão Sub-CA/RA](#)

# Introdução

Este documento descreve o servidor PKI e as funcionalidades de cliente IO em detalhe. Endereça o projeto inicial e as considerações de desenvolvimento IO PKI.

# Infraestrutura PKI




## Certificate Authority

O Certificate Authority (CA), igualmente referido como o servidor PKI durante todo o documento, é

uma entidade confiável essa Certificados das edições. O PKI é baseado na confiança, e a confiança-hierarquia começa no Certificate Authority da raiz (CA raiz). Porque a CA raiz está na parte superior da hierarquia, tem um certificado auto-assinado.

### Certificate Authority subordinado

Na Confiança-hierarquia PKI toda a raiz abaixo das autoridades de certificação é sabida como autoridades de certificação subordinadas (Secundário-CA). Evidentemente, um certificado de Secundário-CA é emitido por CA, que seja um nível acima.

O PKI não impõe nenhum limite no número de Secundário-CAS em uma hierarquia dada. Contudo, em uma distribuição de empreendimento com mais de 3 níveis de autoridades de certificação pode tornar-se difícil de controlar.

### Autoridade de registro

O PKI define um Certificate Authority especial conhecido como o registration authority (RA autoridade de registro), que é responsável para autorizar os clientes PKI de se registrar a Secundário-CA dado ou à CA raiz. O RA não emite Certificados aos clientes PKI, em lugar de decide que PKI-cliente pode ou não pode ser emitido um certificado por Secundário-CA ou pela CA raiz.

O papel principal de um RA é offload a validação básica do pedido do certificado de cliente de CA, e protege CA da exposição direta aos clientes. Esta maneira, RA está entre os clientes PKI e o CA, assim protegendo CA de qualquer tipo do ataque de recusa de serviço.

## Cliente PKI

Todo o dispositivo que pede para um certificado baseado em um par de chaves público-privado residente para provar sua identidade aos outros dispositivos é sabido como um cliente PKI.

Um cliente PKI deve ser capaz da geração ou da armazenagem um par de chaves público-privado tal como o RSA ou o DSA ou o ECDSA.

Um certificado é uma prova de identidade e de validez de uma chave pública dada, desde que a chave privada correspondente existe no dispositivo.

## Servidor PKI IO

Evolução da característica do servidor PKI da tabela 1. IO

| Recurso | IO [ISR-G1, ISR-G2] | IOS-XE [ASR1K, ISR4K] |
|---|---|---|
| Server IO CA/PKI | 12.3(4)T | XE 3.14.0/15.5(1)S |
| Derrubamento do certificado de servidor PKI IO | 12.4(1)T | XE 3.14.0/15.5(1)S |
| IO PKI HA | 15.0(1)M | [Implicit Inter-RP Redundancy is available] NA |
| IO RA para a 3ª | 15.1(3)T | XE 3.14.0/15.5(1)S |

### parte CA

Antes de obter na configuração de servidor PKI, o administrador deve compreender estes conceitos do núcleo.

## Fonte competente de tempo

Uma das fundações da infraestrutura PKI é tempo. O relógio de sistema define se um certificado é válido ou não. Daqui, nos IO, o pulso de disparo deve ser feito competente ou de confiança. Sem uma fonte competente de tempo, o servidor PKI não pode funcionar como esperado, e é altamente recomendado fazer o pulso de disparo em IO competente usando estes métodos:

### NTP (protocolo Network Time Protocol)

Sincronizar o relógio de sistema com um Time Server é a única maneira verdadeira de fazer o relógio de sistema de confiança. Um IOS Router pode ser configurado como um cliente de NTP a um servidor de NTP conhecido e estável na rede:

```
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
```

Os IO podem igualmente ser configurados como um servidor de NTP, que marque o pulso de disparo de sistema local como competente. No desenvolvimento em escala reduzida PKI, o servidor PKI pode ser configurado como um servidor de NTP para seus clientes 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
```

### Relógio de hardware de marcação como confiado

Nos IO, o relógio de hardware pode ser marcado como a utilização competente:

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

Isto pode ser configurado junto com o NTP, e a razão chave para fazer isto é manter o relógio de sistema competente quando os recarregamentos de roteador, por exemplo devido a uma interrupção de energia, e aos servidores de NTP não são alcançáveis. Nesta fase, os temporizadores PKI pararão de funcionar, que conduz por sua vez para certificate falhas da renovação/derrubamento. **o clock calendar-valid** atua como uma proteção em tais situações.

Ao configurar isto, é chave compreender que o relógio de sistema sairá da sincronização se a bateria de sistema morre, e o PKI começará confiar um pulso de disparo fora de sincronia. Contudo, é relativamente mais seguro configurar isto, do que não tendo uma fonte competente de tempo de todo.

> Nota: **o comando clock calendar-valid** foi adicionado na versão XE 3.10.0 IOS-XE/15.3(3)S avante.

## Hostname e Domain Name

Recomenda-se configurar um hostname e um Domain Name no Cisco IOS como uma das primeiras etapas antes de configurar algum serviço relacionado PKI. O nome de host do roteador e o Domain Name são usados nas seguintes encenações:

- O nome do par de chaves do padrão RSA é derivado combinando o hostname e o Domain Name
- Ao registrar-se para um certificado, o assunto-nome do padrão consiste no atributo do hostname e em um não organizado-nome, que seja hostname e Domain Name unidos.

Quanto para ao servidor PKI, o hostname e o Domain Name não são usados:

- O nome do par de chaves do padrão será o mesmo que aquele do nome de servidor PKI
- O Assunto-nome do padrão consiste no CN, que é o mesmo que aquele do nome de servidor PKI.

A recomendação geral é configurar um hostname apropriado e um Domain Name.

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

# Server do HTTP

O servidor PKI IO é permitido somente se o Server do HTTP é permitido. É importante notar que, se o servidor PKI é desabilitado devido ao Server do HTTP que está sendo desabilitado, pode continuar a conceder o [via terminal] autônomo dos pedidos. A capacidade do Server do HTTP é exigida para processar pedidos SCEP, e manda respostas SCEP.

O Server do HTTP IO é utilização permitida:

```
ip http server
```

E a porta de Server do HTTP do padrão pode ser mudada de 80 a toda a utilização válida do número de porta:

```
ip http port 8080
```

### MAX-conexão HTTP

Um dos gargalos, quando distribuir IO como o servidor PKI que usa o SCEP for conexões de HTTP simultâneas máximas e conexões de HTTP médias pelo minuto.
Atualmente, as conexões simultâneas máximas em um Server do HTTP IO são limitadas a 5 à revelia e podem ser aumentadas a 16, que é altamente recomendado em um desenvolvimento da escala média:

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

Este as instalações IO permitem conexões de HTTP simultâneas máximas até 1000:

- UniversalK9 IO com licença-grupo uck9

O CLI é mudado automaticamente para aceitar um argumento numérico entre 1 a 1000

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

O Server do HTTP IO permite 80 conexões pelo minuto [580 no caso das versões do IOS onde as sessões simultâneas máximas HTTP podem ser aumentadas a 1000] e quando este limite é alcançado dentro de um minuto, ouvinte IO HTTP começa estrangular as conexões de HTTP entrantes fechando o ouvinte por 15 segundos. Isto conduz aos pedidos de conexão de cliente que são deixado cair devido ao **limite de fila da** conexão de TCP **alcançado**. Mais informação nesta pode ser encontrada aqui

## Par de chaves RSA

O par de chaves RSA para a funcionalidade do servidor PKI em IO pode ser gerado automaticamente ou gerou manualmente.
Ao configurar um servidor PKI, os IO criam automaticamente um ponto confiável pelo mesmo nome que o servidor PKI a fim armazenar o certificado de servidor PKI.

### Manualmente gerando o par de chaves do servidor PKI RSA:

Etapa 1. Crie um par de chaves RSA com o mesmo nome que aquele do servidor PKI:

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

Etapa 2. Antes de permitir o servidor PKI, altere o ponto confiável do servidor PKI:

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

```
rsakeypair <LABEL>
```

Nota: O valor do módulo do par de chaves RSA mencionado sob o ponto confiável do servidor PKI não é tomado na consideração até o ver 15.4(3)M4 IO, e esta é umas advertências conhecidas. O módulo do chave padrão é 1024 bit.

### Auto-gerando o par de chaves do servidor PKI RSA:

Ao permitir o servidor PKI, os IO gerenciem automaticamente um par de chaves RSA com o mesmo nome que aquele do servidor PKI, e o tamanho chave do módulo são 1024 bit.

Começando o ver 15.4(3)M5 IO, esta configuração cria um par de chaves RSA com o <LABEL> porque o nome e a força chave serão conforme o módulo definido do <MOD>.

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

Desmancha prazeres

O servidor PKI CSCuu73408 IO deve permitir o tamanho chave não-padrão para o CERT do derrubamento.

O servidor PKI CSCuu73408 IO deve permitir o tamanho chave não-padrão para o CERT do derrubamento.

O padrão de mercado atual é usar um mínimo de par de chaves de 2048 bit RSA.

## consideração do temporizador do Auto-derrubamento

Atualmente, o servidor PKI IO não gerencie um certificado do derrubamento à revelia, e tem que explicitamente ser permitido sob o servidor PKI usando o comando do **<days-before-expiry> do auto-derrubamento**. Mais no derrubamento do certificado é explicado dentro

Este comando especifica quantos dias antes que a expiração do certificado PKI Server/CA se os IO criarem um certificado de CA do derrubamento. Note que o certificado de CA do derrubamento está ativado uma vez o certificado de CA ativo atual expira. O valor padrão é atualmente 30 dias. Este valor deve ser ajustado a um valor razoável segundo a vida do certificado de CA, e este influencia por sua vez a configuração do temporizador auto-registrar-se no cliente PKI.

Nota: o temporizador do Auto-derrubamento deve sempre provocar antes de auto-registra o temporizador no cliente durante CA e o [known as] do derrubamento do certificado de cliente

## Considerações CRL

A infraestrutura IO PKI apoia duas maneiras de distribuir o CRL:

### Publique o CRL a um Server do HTTP

O servidor PKI IO pode ser configurado para publicar o arquivo CRL a um lugar específico em um Server do HTTP usando este comando sob o servidor PKI:

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

E o servidor PKI pode ser configurado para encaixar este lugar CRL em todos os certificados de cliente PKI usando este comando sob o servidor PKI:

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

### Método SCEP GetCRL

O servidor PKI IO armazena automaticamente o arquivo CRL na localização do base de dados específica, que é à revelia nvram, e é altamente recomendado manter uma cópia em um server SCP/FTP/TFTP usando este comando sob o servidor PKI:

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

Àrevelia, o servidor PKI IO não encaixa o lugar CDP nos certificados de cliente PKI. Se os clientes IO PKI estão configurados para executar a verificação da revogação, mas o certificado que está sendo validado não tem um CDP encaixado nele, e o ponto confiável de validação de CA está configurado com o lugar de CA (usando o <CA-Server-IP ou o FQDN> de http://), quedas IO de volta ao método baseado SCEP de GetCRL à revelia.
O SCEP GetCRL executa a recuperação CRL executando HTTP GET nesta URL:

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

> Nota: No IOS CLI, antes de entrar**?** , sequência chave da imprensa **CTRL +V**.

O servidor PKI IO pode igualmente encaixar esta URL como o lugar CDP. A vantagem de fazer isto é dupla:

- Assegura-se de que todos os clientes baseados SCEP não-IOS PKI possam executar a recuperação CRL.
- Sem um CDP encaixado, os mensagens request IO SCEP GetCRL são assinados (usando um certificado auto-assinado provisório) como definido no esboço SCEP. Contudo, os pedidos de recuperação CRL não precisam de ser assinados, e encaixando o CDP URL para o método de GetCRL, assinar os pedidos CRL pode ser evitada.

## Vida do CRL

O tempo vida de CRL do servidor PKI IO pode ser controlado usando este comando sob o servidor PKI:

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

O valor realiza-se nas horas. A vida do CRL é ajustada à revelia às horas 6. Segundo como os Certificados são revogados frequentemente, o tempo vida de CRL de ajustamento a um valor ótimo aumenta o desempenho da recuperação CRL na rede.

## Considerações do base de dados

O servidor PKI IO usa o nvram como a localização do base de dados do padrão, e é altamente recomendado usar um server FTP ou TFTP ou SCP como a localização do base de dados. À revelia, o servidor PKI IO cria dois arquivos:

- <Server-Name>.ser – Isto contém o último número de série emitido por CA encanta dentro. O arquivo está no formato em texto simples, e contém esta informação:
  db_version = 1
  last_serial = 0x4

- <Server-Name>.crl – Este é o arquivo codificado DER CRL publicado por CA

O servidor PKI IO armazena a informação no base de dados a 3 níveis configuráveis:

- Mínimo – Este é o nível padrão, e nenhum arquivo é criado neste nível no base de dados, e daqui nenhuma informação está disponível no server de CA em relação aos certificados de cliente concedida no passado.
- Nomes – Neste nível o servidor PKI IO cria um arquivo nomeado <Serial-Number>.cnm para cada certificado de cliente emitido, onde o <Serial-Number> do nome refere o número de série do certificado de cliente emitido e este arquivo do cnm contém o assunto-nome e a data de expiração do certificado de cliente.
- Termine – Neste nível, o servidor PKI IO cria dois arquivos para cada certificado de cliente emitido:
- <Serial-Number>.cnm
- <Serial-Number>.crt

aqui, o arquivo CRT é o arquivo de certificado de cliente, que é DER codificado.

Estes pontos são importantes:

- Antes de emitir um certificado de cliente, o servidor PKI IO refere <Server-Name>.ser para determinar e derivar o número de série do certificado.
- Com base de dados o nível ajustou-se aos nomes ou necessidade terminou-se, <Serial-Number>.cnm e <Serial-Number>.crt de ser escrito ao base de dados antes de enviar certificado concedido/emitido ao cliente
- Com base de dados a URL ajustou-se aos nomes ou terminou-se, o base de dados URL deve ter bastante espaço para salvar os arquivos. Daqui a recomendação é configurar um [FTP or TFTP or SCP] do servidor de arquivo externo como o base de dados URL.
- Com o base de dados externo URL configurado, é absolutamente necessário certificar-se de que o servidor de arquivo é alcançável durante o processo da concessão do certificado, que de outra maneira marcaria o server de CA como enfermos. E a intervenção manual é exigida

para trazer em linha a parte traseira do server de CA.

## Arquivo do base de dados

Ao distribuir um servidor PKI, é importante considerar os cenários de falha e para ser preparado, deve haver uma falha do hadrware. Há duas maneiras de conseguir isso:

1. Redundância
   Neste caso, duas dispositivos ou unidades de processamento atuam como Ativo-à espera para fornecer a Redundância.
   O servidor PKI IO highavailability pode ser conseguido usando dois Roteadores permitidos HSRP ISR [ISR G1 e ISR G2] como explicado em

   OS IO que O XE baseou os sistemas [ISR4K e ASR1k] não têm a opção da dispositivo-Redundância disponível. Contudo, em ASR1k a Redundância inter-RP está disponível à revelia.
2. Arquivando o par de chaves e os arquivos do server de CA
   Os IO fornecem uma facilidade para arquivar o par de chaves do servidor PKI e o certificado. Arquivar pode ser feita usando dois tipos de arquivos:
   PEM - Os IO criam arquivos formatados PEM para armazenar a chave pública RSA, chave privada cifrada RSA, certificado de servidor de CA. O par de chaves e os Certificados do derrubamento são arquivados automaticamentePKCS12 - Os IO criam um único arquivo do PKCS12 que contém o certificado de servidor de CA e a chave privada correspondente RSA cifrados usando uma senha.

A arquivística do base de dados pode ser permitida usando este comando sob o servidor PKI:

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

Éigualmente possível armazenar os arquivos arquivados a um servidor separado, usando possivelmente um protocolo seguro (SCP) que usa o comando seguinte sob o servidor PKI:

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

De todos os arquivos no base de dados à exceção dos arquivos arquivados e. O arquivo de Ser, todos arquivos restantes está no texto claro e não levanta nenhuma ameaça real se perdido, e daqui pode ser armazenado em um servidor separado sem incorrer muitas despesas gerais ao redigir os arquivos, por exemplo um servidor TFTP.

## IO como Secundário-CA

O servidor PKI IO pega à revelia o papel de uma CA raiz. Para configurar um servidor PKI subordinado (Secundário-CA), permita primeiramente este comando sob a seção de configuração do servidor PKI (antes de permitir o servidor PKI):

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

Usando isto configurar a URL de Raiz-CA sob o ponto confiável do servidor PKI:

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

Permitir este servidor PKI provoca agora estes eventos:

- O ponto confiável do servidor PKI é autenticado a fim instalar o certificado CA raiz.
- Depois que a CA raiz é autenticada, os IO gerenciem um CSR para a limitação básica de Subordinado-CA [x509 que contém CA: A bandeira VERDADEIRA] e envia-o à CA raiz

Independentemente do modo da concessão configurado na CA raiz, os IO põem os pedidos do certificado de CA (ou RA) na fila pendente. Um administrador tem que manualmente conceder os certificados de CA.
Para ver o pedido do certificado pendente e a pedido-identificação:

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

Para conceder o pedido:

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

- Usando isto, a operação subsequente da VOTAÇÃO SCEP (GetCertInitial) transfere o certificado de Secundário-CA e instala-o no roteador, que permite o servidor PKI subordinado

### IO como o RA

O servidor PKI IO pode ser configurado como uma autoridade de registro a um subordinado dado ou à CA raiz. Para configurar o servidor PKI como uma autoridade de registro, permita primeiramente este comando sob a seção de configuração do servidor PKI (antes de permitir o servidor PKI):

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

Depois disto, configurar a URL de CA sob o ponto confiável do servidor PKI. Isto indica que CA é protegido pelo RA:

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

Uma autoridade de registro não emite Certificados, daqui a configuração do **nome de emissor** sob o RA não é exigida, e não é eficaz mesmo se é configurada. O assunto-nome de um RA é configurado sob o ponto confiável RA usando o **comando subject-name**. É importante configurar **OU = ioscs RA** como parte do assunto-nome para que os IO CA para identificar os IO RA isto é para identificar os pedidos do certificado autorizados pelos IO RA.

Os IO podem atuar como uma autoridade de registro à 3ª parte CA tais como Microsoft CA, e a fim ficar compatível os IO RA tem que ser permitido usando este comando sob a seção de configuração do servidor PKI (antes de permitir o servidor PKI):

```
mode ra transparent
```

No modo do padrão RA, os IO assinam os pedidos do cliente [PKCS#10] usando o certificado RA. Esta operação indica o servidor PKI IO que o pedido do certificado esteve autorizado por um RA.

Com modo transparente RA, os IO para a frente os pedidos do cliente em seu formato original sem introduzir o certificado RA, e este são compatíveis com Microsoft CA como um exemplo conhecido.

## Cliente IO PKI

Uma da entidade de configuração a mais importante no cliente IO PKI é um ponto confiável. Os

parâmetros de configuração do ponto confiável são explicados em detalhe nesta seção.

## Fonte competente de tempo

Como a fonte mais adiantada, competente indicada de tempo é uma exigência no cliente PKI também. O cliente IO PKI pode ser configurado como uma utilização do cliente de NTP esta configuração:

```
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
```

## Hostname e Domain Name

Uma recomendação geral é configurar um hostname e um Domain Name no roteador:

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

## Par de chaves RSA

No cliente IO PKI, o par de chaves RSA para um registro dado do ponto confiável pode automaticamente ser gerado ou manualmente gerado.

O processo de geração chave automático RSA envolve o seguinte:

- Os IO criam à revelia o par de chaves de 512 bit RSA
- O nome automaticamente gerado do par de chaves é hostname.domain-name, que é o hostname do dispositivo combinado com o Domain Name do dispositivo
- O par de chaves gerado automaticamente não é marcado como exportable.

O processo de geração chave automático RSA envolve o seguinte:

- Opcionalmente, um par de chaves do uso geral RSA de uma força apropriada pode ser utilização manualmente gerada:
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]` Aqui, ETIQUETA - o nome do par de chaves RSA
  Modificação - O módulo ou a força da chave RSA nos bit entre 360 lavram 4096, que são tradicionalmente 512, 1024, 2048 ou 4096.
  A vantagem manualmente de gerar o par de chaves RSA é a capacidade para marcar o par de chaves como exportable, que permite por sua vez o certificado de identidade ser exportado completamente, que pode então ser restaurado em um outro dispositivo. Contudo, se deve compreender as implicações de segurança desta ação.
- Um par de chaves RSA é ligado a um ponto confiável antes do registro usando este comando

`crypto pki trustpoint MGMT`

`rsakeypair <LABEL> [<MOD> <MOD>]` Aqui, se um par de chaves RSA nomeado <LABEL> já existe, a seguir é pegarado durante o registro do ponto confiável.

Se um par de chaves RSA nomeado <LABEL> não existe, a seguir uma da seguinte ação está executado durante o registro:

- Se nenhum argumento do <MOD> é passado, a seguir 512 bit <LABEL> nomeado par de chaves estão gerados.

- se um argumento do <MOD> é passado, a seguir um par de chaves de uso geral dos bit do <MOD> nomeado <LABEL> está gerado

- se dois argumentos do <MOD> são passados, a seguir um par de chaves da assinatura dos bit do <MOD> e um par de chaves da criptografia dos bit do <MOD>, ambos os <LABEL> Nomeados estão gerados

## Ponto confiável

Um ponto confiável é um recipiente abstrato para guardar um certificado nos IO. Um único ponto confiável é capaz de armazenar dois Certificados ativos a um momento determinado:

- Um certificado de CA - Carregando um certificado de CA em um ponto confiável dado é sabido como o processo de autenticação do ponto confiável.
- Um certificado ID emitido por CA - a carga ou a importação de um certificado ID em um ponto confiável dado são sabidas como o processo do registro do ponto confiável.

Uma configuração do ponto confiável é sabida como uma política da confiança, e esta define aquela:

- Que certificado de CA é carregado no ponto confiável?
- Que CA o ponto confiável se registra?
- Como os IO registram o ponto confiável?
- Como um certificado emitido pelo [loaded in the trustpoint] dado de CA é validado?

Os componentes principais de um ponto confiável são explicados aqui.

### Modo do registro

Um modo do registro do ponto confiável, que igualmente defina o modo de autenticação do ponto confiável, pode ser executado através de 3 métodos principais:

1. Registro terminal - método manual de executar a autenticação e o certificado de registro do ponto confiável usando a cópia-pasta no terminal CLI.
2. Registro SCEP - Autenticação e registro do ponto confiável usando o SCEP sobre o HTTP.
3. Perfil do registro - Aqui, os métodos da autenticação e do registro são definidos separadamente. Junto com o terminal e os métodos do registro SCEP, os perfis do registro fornecem uma opção para especificar comandos HTTP/TFTP executar a recuperação de arquivo do server, que é definido usando uma autenticação ou um registro URL sob o perfil.

### Interface de origem e VRF

A autenticação e a matrícula do ponto confiável sobre HTTP (SCEP) ou TFTP (perfil do registro) usam o sistema de arquivo IOS para executar operações de entrada/saída do arquivo. Estes intercâmbios de pacotes podem ser originado de uma interface de origem específica e de um VRF.

Em caso da configuração clássica do ponto confiável, esta funcionalidade é permitida usando secundário-comandos da **interface de origem** e do **vrf** sob o ponto confiável.

Em caso dos perfis, da **interface de origem** e do **registro do** registro **| a autenticação URL <http/tftp://Server-location > os** comandos do **<vrf-name> do vrf** fornecem a mesma funcionalidade.

Exemplo de configuração:

```
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
```

### Certificado de registro e renovação automáticos

O cliente IO PKI pode ser configurado para executar o registro automático e a renovação usando este comando sob a seção do ponto confiável PKI:

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

Aqui, auto-**registre** estados do comando do **[regenerate] do <percentage>** que os IO devem executar a renovação do certificado em exatamente 80% da vida do certificado atual.

**O regenerado da** palavra-chave indica que os IO devem regenerar o par de chaves RSA conhecido como o par de chaves da sombra durante cada operação da renovação do certificado.

Este é o comportamento do registro automático:

- O momento auto-**registra-se** é configurado, se o ponto confiável é autenticado, IO executará um registro automático ao server situado na URL mencionada como parte do **comando enrollment url** sob a seção do ponto confiável PKI ou sob o perfil do registro.
- O momento onde um ponto confiável é registrado com um servidor PKI ou CA, uma RENOVAÇÃO ou um temporizador da SOMBRA são inicializados no cliente PKI basearam na porcentagem auto-**registrar-se do** certificado de identidade atual instalaram sob o ponto confiável. Este temporizador é visível sob o comando **cripto do temporizador do pki da mostra**. Mais nas funções do temporizador *referem*
- O apoio da capacidade da renovação vem do servidor PKI. Mais nisto dentro

O cliente IO PKI executa dois tipos de renovação:
Renovação implícita: Se o servidor PKI não envia a “renovação” como uma capacidade apoiada, os IO executam um registro inicial no definido auto-registram a porcentagem. isto é os IO usam um certificado auto-assinado para assinar a requisição de renovação.Renovação explícita: Quando o servidor PKI apoia a característica da renovação do certificado de cliente PKI, anuncia a “renovação” como uma capacidade apoiada. Os IO tomam esta capacidade na consideração durante a renovação isto é IO do certificado usam o certificado de identidade ativo atual para assinar o pedido do certificado da renovação.

Deve ser tomado ao configurar auto-registre a porcentagem. Em todo o cliente dado PKI no desenvolvimento, se uma circunstância elevara onde o certificado de identidade expira ao mesmo tempo que o certificado de CA de emissão, a seguir o valor auto-registrar-se deve sempre provocar a operação da renovação do [shadow] depois que CA criou o certificado do derrubamento. *Refira a* seção das **dependências do temporizador PKI** dentro

## Certificate a Revogação-verificação

Um ponto confiável autenticado PKI isto é um ponto confiável PKI que contém um certificado de CA é capaz de executar a validação certificada durante um IKE ou uma negociação de SSL, onde o certificado de peer seja sujeitado a uma validação certificada completa. Um dos métodos da validação é verificar o status de revogação do certificado de peer usando um dos seguintes dois métodos:

- Certificate Revocation List (CRL) - Este é um arquivo que contém os números de série dos Certificados revogados por CA dado. Este arquivo é assinado usando o certificado de CA de emissão. O método CRL envolve transferir o arquivo CRL usando o HTTP ou o LDAP.

- Protocolo status em linha do certificado (OCSP) - Os IO estabelecem o canal de comunicação com uma entidade chamada como o que responde OCSP, que é um server designado por CA de emissão. Um cliente tal como IO envia um pedido que contém o número de série do certificado está sendo validado. O que responde OCSP responde com o status de revogação do número de série dado. O canal de comunicação poderia ser estabelecido usando todo o aplicativo suportado/protocolo de transporte, que for geralmente HTTP.

A verificação da revogação pode ser definida usando estes comanda sob a seção do ponto confiável PKI:

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

Àrevelia, um ponto confiável é configurado para executar a verificação da revogação usando o crl.

Os métodos podem ser requisitados novamente, e a verificação do status de revogação é executada na ordem definida. O método “nenhuns” contorneia a revogação-verificação.

## Esconderijo CRL

Com revogação-verificação baseada CRL, cada validação certificada pode provocar uma transferência fresca do arquivo CRL. E como o arquivo CRL obtém mais grande ou se o CRL Distribution Point (CDP) está mais distante ausente, transferir o arquivo durante cada processo de validação impede do desempenho do dependente de protocolo na validação certificada. Daqui, pôr em esconderijo CRL é executado para melhorar o desempenho, e pôr em esconderijo o CRL

toma a validez CRL na consideração.

A validez CRL é definida usando dois parâmetros do tempo: **LastUpdate**, que é a última vez o CRL foi publicado por CA de emissão, e por **NextUpdate**, que é o tempo é o futuro em que uma nova versão do arquivo CRL é publicada por CA de emissão.

Os IO põem em esconderijo cada CRL transferido para enquanto o CRL é válido. Contudo, em certas circunstâncias como o CDP que não é alcançável temporariamente, pode ser necessário reter por um período de tempo prolongado o CRL no esconderijo. Nos IO um CRL posto em esconderijo pode retido para enquanto 24 horas depois que a validez CRL expira, e este pode ser configurado usando este comando sob a seção do ponto confiável PKI:

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

Em certas circunstâncias como CA de emissão que revoga Certificados dentro do período de validade CRL, os IO enlatam confiugured para suprimir mais frequentemente do esconderijo. Suprimindo o CRL prematuramente, os IO são forçados para transferir mais frequentemente o CRL para manter o esconderijo CRL atualizado. Esta opção de configuração está disponível sob a seção do ponto confiável PKI:

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

E finalmente, os IO podem ser configurados para não pôr em esconderijo o arquivo CRL usando este comando sob a seção do ponto confiável PKI:

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

# Configuração recomendada

Um desenvolvimento típico de CA com CA raiz e uma configuração de Secundário-CA é como abaixo. O exemplo igualmente inclui uma configuração de Secundário-CA protegida por um RA.

Com par de chaves de 2048 bit RSA em toda a linha, este exemplo recomenda uma instalação onde:
A CA raiz tem uma vida de 8 anos
Secundário-CA tem uma vida de 3 anos
Os certificados de cliente são emitidos por um ano, que são configurados para pedir para uma renovação do certificado, automaticamente.

## CA RAIZ - Configuração

```
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 sem RA - Configuração

```
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 com RA - Configuração

```
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 para SUBCA - Configuração

```
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
```

# Certificado de registro

## Inscrição manual

A Inscrição manual envolve a geração autônoma CSR no cliente PKI, que é copiado manualmente sobre ao administrador CA assina manualmente o pedido, que é importado então no cliente.

### Cliente PKI

Configuração de cliente 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
```

Etapa 1. Autentique primeiramente o ponto confiável (isto pode igualmente ser executado após etapa 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
```

## Etapa 2. Gerencia a solicitação de assinatura de certificado e tome o CSR a CA e obtenha o certificat concedido:

```
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+
UqkqUZZar9TdmB8I7AHku5m79142o8cuhwOccehxE6jmzh9P+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
```

## Etapa 3. Importe agora o certificado concedido através do 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
```

## Servidor PKI

Etapa 1. Exporte primeiramente o certificado de CA de emissão de CA, que é neste caso certificado SUBCA. Isto é importado durante etapa 1 acima no cliente PKI, isto é autenticação do ponto confiável.

```
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-----
```

Etapa 2. Após Step-2 no PKI-cliente, tome o CSR do cliente e forneça-o assinando no SUBCA usando este comando:

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

Este comando sugere que o SUBCA aceite uma solicitação de assinatura de certificado do terminal, e concedido uma vez, os dados do certificado são imprimidos no formato 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
```

Se CA reage do modo da auto-concessão, o certificado concedido está indicado no formato PEM acima. Quando CA reage do modo de concessão manual, o pedido do certificado está marcado como **pendente**, atribuído um valor identificação e enfileirado na fila de pedido do registro.

```
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
```

Etapa 3. Conceda manualmente este pedido usando este comando:

```
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-----
```

Nota: A Inscrição manual de Secundário-CA a uma CA raiz não é possível.

Nota: CA em um estado desabilitado devido desabilitou o Server do HTTP pode manualmente conceder os pedidos do certificado.

## Registro usando o SCEP

A configuração de cliente PKI é:

```
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
```

A configuração de servidor PKI é:

```
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/
```

O modo padrão de concessão do pedido do certificado é manual:

```
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
```

### Concessão manual

Etapa 1. Cliente PKI: Em primeiro, que é imperativo, autentique o ponto confiável no cliente 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.
```

Etapa 2. PKI-cliente: Depois da autenticação do ponto confiável, o cliente PKI pode ser registrado para um certificado.

Nota: Se auto-registre é configurado, o cliente executará automaticamente o registro.

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

Atrás das cenas, estes eventos ocorrem:

- Os IO procuram um par de chaves RSA nomeado PKI-Chave. Se existe, está pegarado pedindo um certificado de identidade. Se não, os IO criam 2048 um bit PKI-Chave nomeada par de chaves, e usam-no então pedindo um certificado de identidade.
- Os IO criam uma solicitação de assinatura de certificado no formato PKCS10.
- Os IO cifram então este CSR usando uma chave simétrica aleatória. A chave simétrica aleatória é cifrada usando a chave pública do receptor, que é o SUBCA (a chave pública de SUBCA é disponível devido à autenticação do ponto confiável). O CSR cifrado, a chave simétrica aleatória cifrada e a informação de receptor são unidos em dados envolvidos PKCS#7.
- Estes dados envolvidos PKCS#7 são assinados usando um certificado auto-assinado provisório durante o registro inicial. O PKCS#7 envolveu dados, o certificado de assinatura usado pelo cliente e a assinatura do cliente é unida em um pacote de dados assinado PKCS#7. Esta é base64 codificado, e então URL codificada. A gota resultante dos dados é enviada como o argumento da “mensagem” em HTTP URI enviado a CA:

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

Etapa 3. Servidor PKI:

Quando o servidor PKI IO recebe o pedido, verifica estes:

1. Verifica se o base de dados do pedido do registro contém um pedido do certificado com o mesmo ID de transação associado com o pedido novo.

Nota: Um ID de transação é uma mistura MD5 da chave pública, para que um certificado de identidade está sendo pedido pelo cliente.

2. Verifica se o base de dados do pedido do registro contém um pedido do certificado com a mesma senha do desafio que essa enviada pelo cliente.

Nota: Se (1) retornam verdadeiro ou (1) e (2) junto verdadeiro do retorno, a seguir um server de CA é capaz de rejeitar o pedido com base no pedido duplicado da identidade. Contudo, em tal servidor PKI do caso IO substitui o pedido mais velho com o pedido mais novo.

Etapa 4. Servidor PKI:

Conceda manualmente os pedidos no servidor PKI:

Para ver o pedido:

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

Para conceder um pedido específico ou todos os pedidos:

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

Etapa 5. PKI-cliente:

Entrementes, um cliente PKI começa um temporizador da VOTAÇÃO. Aqui, os IO executam GetCertInitial em intervalos regulares até que o SCEP CertRep = CONCEDIDO junto com o certificado concedido for recebido pelo cliente.

Uma vez que o certificado concedido é recebido, os IO instalam-no automaticamente.