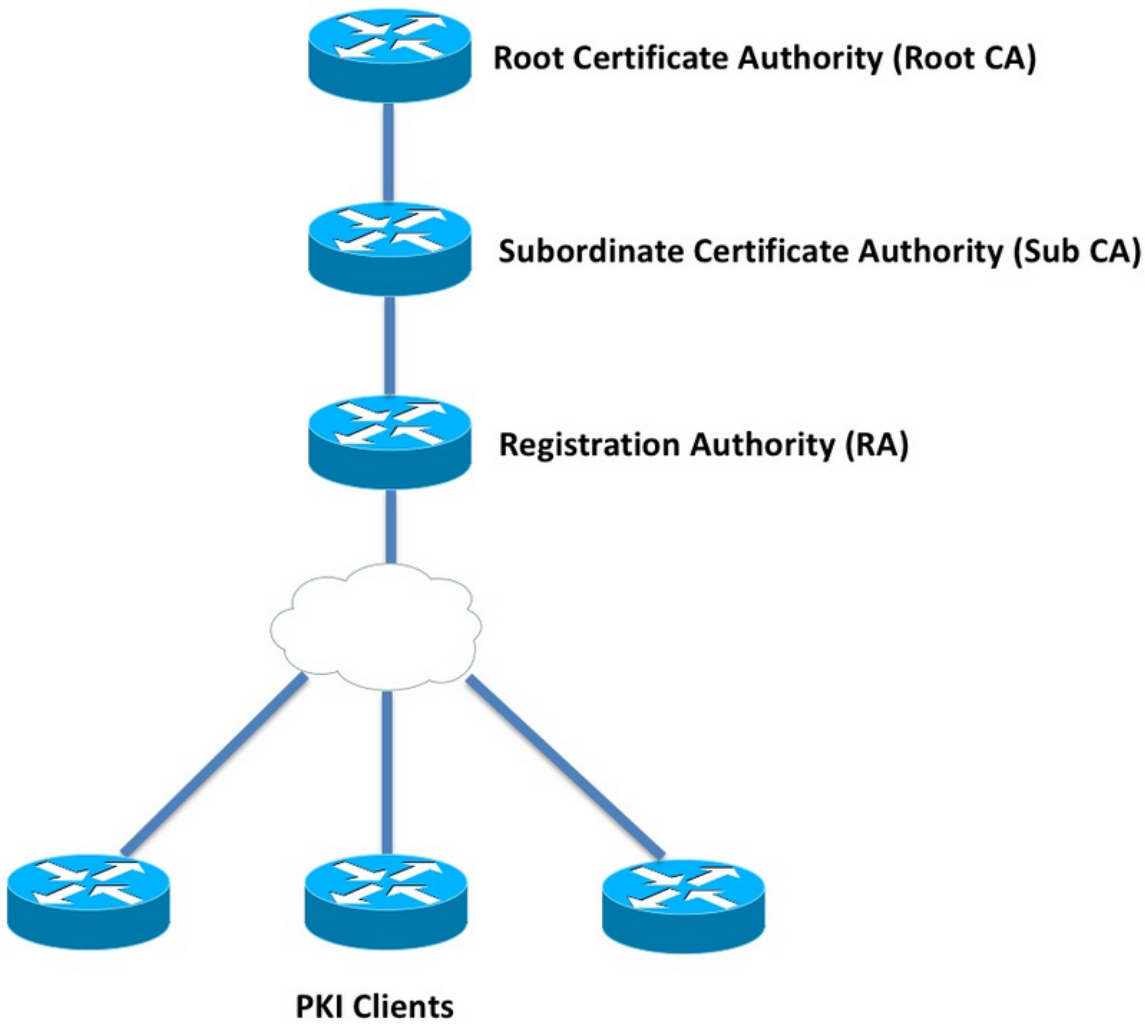
IOS PKI部署指南：初始设计和部署

目录

简介

本文档详细介绍IOS PKI服务器和客户端功能。 它解决了IOS PKI初始设计和部署注意事项。

PKI基础设施



认证中心

证书颁发机构(CA)（在整个文档中也称为PKI服务器）是颁发证书的受信任实体。PKI基于信任，信任层次结构从根证书颁发机构(Root-CA)开始。 由于根CA位于层次结构的顶部，因此它具有自签名证书。

从属证书颁发机构

在PKI信任层次结构中，根下的所有证书颁发机构称为从属证书颁发机构(Sub-CA)。 显然，Sub-CA证书由CA颁发，该级别高于一级。

PKI对给定层次结构中的子CA数量没有限制。但是，在具有3级以上证书颁发机构的企业部署中，可能难以管理。

注册机构

PKI定义一个称为注册机构(RA)的特殊证书颁发机构，负责授权PKI客户端注册到给定的子CA或根CA。RA不向PKI客户端颁发证书，而是决定哪些PKI客户端可以或不能由子CA或根CA颁发证书。

RA的主要作用是从CA卸载基本客户端证书请求验证，并保护CA不直接暴露到客户端。这样，RA就位于PKI客户端和CA之间，从而保护CA免受任何类型的拒绝服务攻击。

PKI客户端

任何根据驻留的公钥 — 私钥对请求证书以向其他设备证明其身份的设备都称为PKI客户端。

PKI客户端必须能够生成或存储公有私钥对，如RSA或DSA或ECDSA。

证书是给定公钥的身份和有效性的证明，前提是设备上存在相应的私钥。

IOS PKI服务器

表1. IOS PKI服务器功能演进

功能	IOS [ISR-G1、ISR-G2]	IOS-XE [ASR1K、ISR4K]
IOS CA/PKI服务器	12.3(4)T	XE 3.14.0 / 15.5(1)S
IOS PKI服务器证书滚动	12.4(1)T	XE 3.14.0 / 15.5(1)S
IOS PKI HA	15.0(1)M	不适用[隐式RP间冗余可用]
IOS RA，用于第三方CA	15.1(3)T	XE 3.14.0 / 15.5(1)S

在进入PKI服务器配置之前，管理员必须了解这些核心概念。

权威时间来源

PKI基础设施的基础之一是时间。系统时钟定义证书是否有效。因此，在IOS中，时钟必须具有权威性或可信性。如果没有权威时间源，PKI服务器可能无法按预期运行，强烈建议使用以下方法使IOS上的时钟具有权威性：

NTP（网络时间协议）

将系统时钟与时间服务器同步是使系统时钟值得信赖的唯一真正方法。IOS路由器可配置为网络中已知且稳定的NTP服务器的NTP客户端：

```
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也可配置为NTP服务器，该服务器会将本地系统时钟标记为授权。在小规模PKI部署中，PKI服务器可配置为其PKI客户端的NTP服务器：

```
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
```

将硬件时钟标记为受信任

在IOS中，硬件时钟可以使用以下命令标记为授权时钟：

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

这可以与NTP一起配置，执行此操作的关键原因是当路由器重新加载时（例如由于断电），并且NTP服务器无法访问时，保持系统时钟的权威性。在此阶段，PKI计时器将停止运行，这进而导致证书续订/全反故障。**clock calendar-valid在此**情况下充当保护。

在配置此时，了解系统时钟在系统电池失效时会不同步，而PKI将开始信任不同步的时钟，这一点至关重要。但是，配置这一点比完全没有权威时间来源要安全。

> **注意：clock calendar-valid**命令从IOS-XE版本XE 3.10.0/15.3(3)S开始添加。

主机名和域名

在配置任何PKI相关服务之前，建议先在Cisco IOS上配置主机名和域名，作为第一步之一。路由器主机名和域名用于以下场景：

- 默认RSA密钥对名称是通过将主机名和域名组合而派生的
- 注册证书时，默认的使用者名称由主机名属性和非结构化名称组成，即主机名和域名组合在一起。

对于PKI服务器，不使用主机名和域名：

- 默认密钥对名称将与PKI服务器名称的名称相同
- 默认主题名称由CN组成，与PKI服务器名称相同。

一般建议配置适当的主机名和域名。

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

HTTP 服务器

仅在启用HTTP服务器时，才启用IOS PKI服务器。请注意，如果PKI服务器因HTTP服务器被禁用而被禁用，它可以继续授予脱机请求[通过终端]。处理SCEP请求和发送SCEP响应需要HTTP服务器功能。

IOS HTTP服务器启用时使用：

```
ip http server
```

默认HTTP服务器端口可从80更改为任何有效端口号，使用：

```
ip http port 8080
```

HTTP最大连接数

使用SCEP将IOS部署为PKI服务器时的一个瓶颈是每分钟最大并发HTTP连接数和平均HTTP连接数。
目前，IOS HTTP服务器上的最大并发连接数默认限制为5，可增加到16，这在中型部署中强烈建议：

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

此IOS安装允许最多1000个并发HTTP连接：

- 带uck9许可证集的UniversalK9 IOS

CLI会自动更改为接受介于1和1000之间的数值参数

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

IOS HTTP服务器允许每分钟80个连接[在IOS版本中，最大HTTP并发会话数可增加到1000]，并且当在一分钟内达到此限制时，IOS HTTP侦听程序通过关闭侦听程序15秒来开始限制传入HTTP连接。这会导致由于达到TCP连接队列限制而丢弃客**户端连接请求**。有关此项的详细信息，请在此处

RSA密钥对

IOS上用于PKI服务器功能的RSA密钥对可以自动生成或手动生成。
在配置PKI服务器时，IOS会以与PKI服务器相同的名称自动创建信任点以存储PKI服务器证书。

手动生成PKI服务器RSA密钥对：

步骤1.创建与PKI服务器同名的RSA密钥对：

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

步骤2.在启用PKI服务器之前，请修改PKI服务器信任点：

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

> **注意**：在IOS版本15.4(3)M4之前，PKI服务器信任点下提及的RSA密钥对模数值不会被考虑在内，这是已知的警告。默认密钥模数为1024位。

自动生成PKI服务器RSA密钥对：

启用PKI服务器时，IOS会自动生成与PKI服务器同名的RSA密钥对，密钥模数大小为1024位。

从IOS 15.4(3)M5版开始，此配置将创建一个RSA密钥对，其名称为<LABEL>，密钥强度将根据定义的<MOD>模数。

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

Spoiler
CSCuu73408 IOS PKI服务器应允许滚动证书的非默认密钥大小。
CSCuu73408 IOS PKI服务器应允许全反证书的非默认密钥大小。

当前的行业标准是至少使用2048位RSA密钥对。

自动回滚计时器注意事项

目前，IOS PKI服务器默认不生成全反证书，并且必须在PKI服务器下使用**auto-rollover <days-before-expiry>命令显式启用该证**书。有关证书滚动更新的详细信息，请参阅

此命令指定IOS应创建全反CA证书，在PKI服务器/CA证书到期前的天数。请注意，当前活动CA证书过期后，将激活全反CA证书。当前默认值为30天。此值应根据CA证书有效期设置为合理的值，这进而影响PKI客户端上的自动注册计时器配置。

> **注意：**在CA和客户端证书滚动[称为]期间，自动滚动计时器应始终在客户端上自动注册计时器之前触发

CRL注意事项

IOS PKI基础设施支持两种分配CRL的方式：

将CRL发布到HTTP服务器

IOS PKI服务器可以配置为在PKI服务器下使用以下命令将CRL文件发布到HTTP服务器上的特定位置：

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

而且，PKI服务器可以配置为使用PKI服务器下的以下命令将此CRL位置嵌入到所有PKI客户端证书：

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

SCEP GetCRL方法

IOS PKI服务器自动将CRL文件存储在特定数据库位置（默认情况下为nvram），强烈建议在PKI服务器下使用以下命令在SCP/FTP/TFTP服务器上保留副本：

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

默认情况下，IOS PKI服务器不将CDP位置嵌入PKI客户端证书。如果IOS PKI客户端配置为执行撤销检查，但被验证的证书中未嵌入CDP，且验证CA信任点配置有CA位置(使用http://<CA-Server-IP或FQDN>)，则默认情况下，IOS将回退到基于SCEP的GetCRL方法。
SCEP GetCRL通过对此URL执行HTTP GET来执行CRL检索：

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

> **注意：**在IOS CLI中，在输入**?**之前，**按Ctrl +** V键序列。

IOS PKI服务器也可以将此URL嵌入为CDP位置。这样做的好处是两倍：

- 它确保所有非IOS SCEP的PKI客户端都可以执行CRL检索。
- 如果没有嵌入式CDP，IOS SCEP GetCRL请求消息将按照SCEP草案中的定义签名（使用临时自签名证书）。但是，CRL检索请求无需签名，通过嵌入GetCRL方法的CDP URL，可以避免对CRL请求进行签名。

CRL的生存期

IOS PKI服务器的CRL生存期可在PKI服务器下使用以下命令进行控制：

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

值以小时为单位。默认情况下，CRL的生存期设置为6小时。根据证书撤销的频率，将CRL寿命调整为最佳值会提高网络中的CRL检索性能。

数据库注意事项

IOS PKI服务器使用nvram作为默认数据库位置，强烈建议使用FTP或TFTP或SCP服务器作为数据库位置。默认情况下，IOS PKI服务器创建两个文件：

- <Server-Name>.ser — 包含CA以十六进制发出的最后一个序列号。该文件采用纯文本格式，包含以下信息：
  db_version = 1
  last_serial = 0x4
- <Server-Name>.crl — 这是CA发布的DER编码的CRL文件

IOS PKI服务器将信息存储在数据库中的3个可配置级别：

- 最低 — 这是默认级别，在此级别数据库中未创建任何文件，因此CA服务器上没有有关过去授予的客户端证书的信息。
- 名称 — 在此级别，IOS PKI服务器为颁发的每个客户端证书创建一个名为<Serial-Number>.cnm的文件，其中名称<Serial-Number>是指颁发的客户端证书的序列号，此cnm文件包含使用者名称和客户端证书的到期日期。
- 完成 — 在此级别，IOS PKI服务器为每个颁发的客户端证书创建两个文件：
- <序列号>.cnm
- <序列号>.crt

此处，crt文件是客户端证书文件，该文件是DER编码。

以下要点非常重要：

- 在颁发客户端证书之前，IOS PKI服务器引用<Server-Name>.ser来确定并派生证书的序列号。
- 将“数据库”级别设置为“名称”或“完整”时，在将授予/颁发的证书发送到客户端之前，需要先将<Serial-Number>.cnm和<Serial-Number>.crt写入数据库
- 将数据库URL设置为“名称”或“完成”时，数据库URL必须有足够的空间来保存文件。因此，建议将外部文件服务器[FTP或TFTP或SCP]配置为数据库URL。
- 配置了外部数据库URL后，绝对有必要确保在证书授予过程中文件服务器可访问，否则会将CA服务器标记为禁用。而且，需要手动干预才能使CA服务器重新联机。

数据库存档

在部署PKI服务器时，必须考虑故障场景并做好准备，以防硬件故障。实现这一点有两种方法：

1. 冗余
   在这种情况下，两个设备或处理单元充当主备模式以提供冗余。
   IOS PKI服务器高可用性可通过使用两个启用HSRP的ISR路由器[ISR G1和ISR G2]实现，如中所述

   基于IOS XE的系统[ISR4K和ASR1k]没有可用的设备冗余选项。但是，在ASR1k中，RP间冗余默认可用。
2. 存档CA服务器密钥对和文件
   IOS提供了存档PKI服务器密钥对和证书的工具。可使用两种类型的文件完成归档：
   PEM - IOS创建PEM格式化文件以存储RSA公钥、加密RSA私钥、CA服务器证书。将鼠标指针置于密钥对和证书之上会自动存档PKCS12 - IOS创建一个PKCS12文件，其中包含使用密码加密的CA服务器证书和相应的RSA私钥。

可在PKI服务器下使用以下命令启用数据库存档：

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

也可以将存档的文件存储到单独的服务器，可能使用PKI服务器下的以下命令使用安全协议(SCP):

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

在数据库中除存档文件和.Ser文件外的所有文件中，所有其他文件都采用明文形式，如果丢失，不会造成任何实际威胁，因此可以存储在单独的服务器上，而不会在编写文件时产生太多开销，例如TFTP服务器。

IOS作为子CA

默认情况下，IOS PKI服务器将接替根CA的角色。要配置从属PKI服务器(Sub-CA)，请首先在PKI服务器配置部分（启用PKI服务器之前）下启用此命令：

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

使用此配置PKI服务器信任点下的根CA URL:

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

启用此PKI服务器将立即触发以下事件：

- PKI服务器信任点经过身份验证以安装根CA证书。
- 在对Root-CA进行身份验证后，IOS会为Surdiant-CA [x509 basic constraint int CA:TRUE flag]生成CSR，并将其发送到Root-CA

无论根CA上配置的授权模式如何，IOS都会将CA（或RA）证书请求置于挂起队列中。管理员必须手动授予CA证书。
要查看待处理的证书请求和请求ID，请执行以下操作：

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

要授予请求，请执行以下操作：

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

- 使用此操作，后续SCEP POLL(GetCertInitial)操作将下载子CA证书并将其安装到路由器上，从而启用从属PKI服务器

IOS作为RA

IOS PKI服务器可配置为指定从属或根CA的注册机构。要将PKI服务器配置为注册机构，请首先在PKI服务器配置部分（启用PKI服务器之前）下启用此命令：

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

然后，在PKI服务器的信任点下配置CA的URL。这表示哪个CA受RA保护：

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

注册机构不颁发证书，因此RA下的**issuer-name**配置不是必需的，并且即使已配置也无效。RA的subject-name在RA信任点下使用subject-name**命令配**置。为了使IOS CA识别IOS RA（即识别IOS RA授权的证书请求），将**OU = ioscs RA**配置为主题名称的一部分非常重要。

IOS可以充当第三方CA（如Microsoft CA）的注册机构，为保持兼容，必须在PKI服务器配置部分（启用PKI服务器之前）下使用以下命令启用IOS RA:

```
mode ra transparent
```

在默认RA模式下，IOS使用RA证书对客户端请求[PKCS#10]进行签名。此操作指示IOS PKI服务器证书请求已由RA授权。

在透明RA模式下，IOS以原始格式转发客户端请求而不引入RA证书，这与Microsoft CA兼容，这是一个众所周知的示例。

IOS PKI客户端

IOS PKI客户端中最重要的配置实体之一是信任点。本节将详细介绍信任点配置参数。

权威时间来源

如前所述，权威时间源也是PKI客户端的要求。IOS PKI客户端可以使用以下配置配置为NTP客户端：

```
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
```

主机名和域名

一般建议在路由器上配置主机名和域名：

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

RSA密钥对

在IOS PKI客户端中，可以自动生成或手动生成给定信任点注册的RSA密钥对。

自动RSA密钥生成过程涉及以下内容：

- 默认情况下，IOS会创建512位RSA密钥对
- 自动生成的密钥对名称是hostname.domain-name，它是与设备域名组合的设备主机名
- 自动生成的密钥对未标记为可导出。

自动RSA密钥生成过程涉及以下内容：

- 或者，可以使用以下方法手动生成具有适当强度的通用RSA密钥对：
-
  ```
  crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]
  ```
  此处为LABEL — RSA密钥对名称
  MOD - RSA密钥模数或强度（位数），介于360到4096之间，通常为512、1024、2048或4096。
  手动生成RSA密钥对的优点是能够将密钥对标记为可导出，这反过来又允许完全导出身份证书，然后可以在另一设备上恢复。但是，您应该了解此操作的安全影响。
- 在使用此命令进行注册之前，RSA密钥对会链接到信任点

  ```
  crypto pki trustpoint MGMT
  rsakeypair <LABEL> [<MOD> <MOD>]
  ```
  此处，如果名为<LABEL>的RSA密钥对已存在，则在信任点注册期间会拾取该密钥对。

如果名为<LABEL>的RSA密钥对不存在，则在注册期间执行以下操作之一：
— 如果未传递<MOD>参数，则生成名为<LABEL>的512位密钥对。
— 如果传递了一个<MOD>参数，则会生成名为<LABEL>的<MOD>位通用密钥对
— 如果传递了两个<MOD>参数，则生成一个<MOD>位签名密钥对和一个<MOD>位加密密钥对，两个都名为<LABEL>

信任点

信任点是在IOS中保存证书的抽象容器。单个信任点能够在任何给定时间存储两个活动证书：

- CA证书 — 将CA证书加载到给定信任点称为信任点身份验证过程。
- 由CA — 加载ID证书或将ID证书导入给定信任点颁发的ID证书称为信任点注册过程。

信任点配置称为信任策略，这定义：

- 信任点中加载了哪个CA证书？
- 信任点注册到哪个CA?
- IOS如何注册信任点？
- 如何验证由给定CA [加载到信任点]颁发的证书？

此处解释了信任点的主要组件。

注册模式

信任点注册模式（也定义了信任点身份验证模式）可以通过3种主要方法执行：

1. 终端注册 — 在CLI终端中使用复制粘贴执行信任点身份验证和证书注册的手动方法。
2. SCEP注册 — 使用SCEP over HTTP的信任点身份验证和注册。
3. 注册配置文件 — 此处，身份验证和注册方法分别定义。与终端和SCEP注册方法一起，注册配置文件提供了一个选项，用于指定HTTP/TFTP命令以从服务器执行文件检索，该服务器使用配置文件下的身份验证或注册URL进行定义。

源接口和VRF

HTTP(SCEP)或TFTP（注册配置文件）上的信任点身份验证和注册使用IOS文件系统执行文件i/o操作。这些数据包交换可从特定源接口和VRF发出。

在传统信任点配置中，此功能是使用信任点下**的源接**口和**vrf** 子命令启用的。

在注册配置文件、源接**口和注**册情**况 | authentication url <http/tftp://Server-location> vrf <vrf-name>命令提**供相同的功能。

配置示例:

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

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

```
 vrf MGMT
```

或

```
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
```

自动证书注册和续约

IOS PKI客户端可以配置为使用PKI信任点部分下的以下命令执行自动注册和续订：

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

此处， **auto-enroll <percentage> [regenerate]**命令指出，IOS应以当前证书生存期的80%执行证书续约。

关键字**regenerate**表示IOS应在每次证书续订操作期间重新生成RSA密钥对（称为影子密钥对）。

以下是自动注册行为：

- 配置**自动注册**后，如果信任点经过身份验证，IOS将自动注册到PKI信任点部分或注册配置文件下作为注册url命令的一部分提及的URL上的服务器。
- 当信任点向PKI服务器或CA注册时，RENEW或SHADOW计时器根据在信任点下安装的当前身份证书的**自动注册**百分比在PKI客户端上初始化。此计时器在show crypto pki timer**命令下可**见。有关计时器功能的详细*信息，请参阅*
- 续订功能支持来自PKI服务器。有关此内容的详细信息，请参阅
  IOS PKI客户端执行两种类型的续约：
  隐式续约：如果PKI服务器不发送“续约”作为支持的功能，则IOS会按定义的自动注册百分比执行初始注册。即IOS使用自签名证书来签署续约请求。明确续约：当PKI服务器支持PKI客户端证书续订功能时，它会将“续订”作为支持的功能通告。IOS在证书续约期间考虑此功能，即IOS使用当前活动身份证书签署续约证书请求。

配置自动注册百分比时应小心。在部署中的任何给定PKI客户端上，如果出现身份证书与颁发CA证书同时过期的情况，则自动注册值应始终在CA创建滚动证书后触发[影子]续订操作。*请参阅***中的PKI计时器**依赖部分

证书撤销检查

经过身份验证的PKI信任点，即包含CA证书的PKI信任点能够在IKE或SSL协商期间执行证书验证，其中对等体证书经过彻底的证书验证。验证方法之一是使用以下两种方法之一检查对等证书撤销状态：

- 证书撤销列表(CRL) — 此文件包含由给定CA撤销的证书的序列号。此文件使用颁发的CA证书签名。CRL方法包括使用HTTP或LDAP下载CRL文件。

- 在线证书状态协议(OCSP)- IOS与名为OCSP Responder的实体建立通信通道，该实体是发证CA指定的服务器。客户端（如IOS）发送包含正在验证的证书序列号的请求。OCSP响应器以给定序列号的撤销状态做出响应。可以使用任何受支持的应用/传输协议（通常为HTTP）建立

通信信道。
可在PKI信任点部分下使用以下命令定义撤销检查：

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

默认情况下，信任点配置为使用crl执行撤销检查。

可以重新排序这些方法，并且按照定义的顺序执行撤销状态检查。方法“none”绕过撤销检查。

CRL缓存

使用基于CRL的撤销检查时，每个证书验证都可能触发新的CRL文件下载。而且，随着CRL文件变大或CRL分发点(CDP)越远，在每个验证过程中下载文件都会影响协议的性能，这取决于证书验证。因此，执行CRL缓存以提高性能，并且缓存CRL时会考虑CRL的有效性。

CRL有效性使用两个时间参数定义：**LastUpdate**，是上次由发证CA发布CRL的时间，而**NextUpdate**，是将来由发证CA发布新版本CRL文件的时间。

只要CRL有效，IOS就会缓存每个下载的CRL。但是，在某些情况下（例如CDP暂时无法访问），可能需要将CRL保留在缓存中一段较长的时间。在IOS中，缓存的CRL在CRL有效期到期后可保留24小时，并且可在PKI信任点部分下使用以下命令进行配置：

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

在某些情况下（例如，在CRL有效期内颁发CA撤销证书），IOS可以配置为更频繁地删除缓存。过早删除CRL，IOS将被迫更频繁地下载CRL，以使CRL缓存保持最新。此配置选项在PKI信任点部分下可用：

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

最后，IOS可以配置为不使用PKI信任点部分下的以下命令缓存CRL文件：

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

推荐 的 配置

具有根CA和子CA配置的典型CA部署如下所示。本示例还包括由RA保护的子CA配置。

此示例建议在全局使用2048位RSA密钥对时进行以下设置：
根CA有8年的寿命
Sub-CA有3年的寿命
客户端证书签发一年，配置为自动请求证书续签。

根CA — 配置

```
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
```

无RA的SUBCA — 配置

```
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
```

带RA的SUBCA — 配置

```
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 for SUBCA — 配置

```
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
```

证书注册

手动注册

手动注册涉及在PKI客户端上离线生成CSR，PKI客户端会手动复制到CA。管理员手动签署请求，然后将请求导入客户端。

PKI客户端

PKI客户端配置：

```
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
```

步骤1.首先对信任点进行身份验证（这也可在步骤2后执行）。

```
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
```

步骤2.生成证书签名请求，将CSR带到CA并获得授予的证书：

```
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
```

步骤3.现在通过终端导入授权的证书：

```
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服务器

步骤1.首先从CA导出Issing CA证书，在本例中为SUBCA证书。在上述步骤1中，在PKI客户端（即信任点身份验证）上导入此项。

```
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-----
```

步骤2.在PKI客户端上执行步骤2后，从客户端获取CSR，并使用以下命令在SUBCA上进行签名：

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

此命令建议SUBCA接受来自终端的证书签名请求，一旦获得授权，证书数据将以PEM格式打印。

```
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
```

如果CA处于自动授予模式，则以上PEM格式显示授予的证书。当CA处于手动授予模式时，证书请求被标记为**挂起**，被分配ID值并排入注册请求队列。

```
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
```

步骤3.使用以下命令手动授予此请求：

```
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-----
```

注意：无法手动将子CA注册到根CA。

注意：由于HTTP服务器已禁用而处于禁用状态的CA可以手动授予证书请求。

使用SCEP注册

PKI客户端配置为：

```
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服务器配置为：

```
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/
```

证书请求授予的默认模式是手动：

```
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
```

手动授予

步骤1. PKI客户端：作为第一步（必需），在PKI客户端上验证信任点：

```
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.
```

步骤2. PKI客户端：在信任点身份验证后，可以为证书注册PKI客户端。

注意：如果配置了自动注册，客户端将自动执行注册。

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

幕后发生了以下事件：

- IOS查找名为PKI-Key的RSA密钥对。如果存在，则会为请求身份证书而选择该证书。否则，IOS将创建名为PKI-Key的2048位密钥对，然后使用它请求身份证书。
- IOS创建PKCS10格式的证书签名请求。
- 然后，IOS使用随机对称密钥加密此CSR。随机对称密钥使用接收方的公钥进行加密，该公钥是SUBCA（SUBCA的公钥由于信任点身份验证而可用）。 加密的CSR、加密的随机对称密钥和收件人信息被组合在PKCS#7封装数据中。

- 此PKCS#7封装数据在初始注册期间使用临时自签名证书进行签名。PKCS#7封装了数据、客户端使用的签名证书和客户端的签名将放在PKCS#7签名的数据包中。这是base64编码，然后是URL编码。结果的数据blob在发送到CA的HTTP URI中作为“message”参数发送：

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

步骤3. PKI-Server:

当IOS PKI服务器收到请求时，它会检查以下内容：

1.检查注册请求数据库是否包含与新请求关联的具有相同事务ID的证书请求。

注意：事务ID是公钥的MD5哈希值，客户端正为其请求身份证书。

2.检查注册请求数据库是否包含与客户端发送的证书请求具有相同质询密码的证书请求。

注意：如果(1)返回true或(1)和(2)同时返回true，则CA服务器能够以身份请求重复为由拒绝请求。但是，在这种情况下，IOS PKI服务器会用较新的请求替换较旧的请求。

步骤4. PKI-Server:

在PKI服务器上手动授予请求：

要查看请求，请执行以下操作：

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

要授予特定请求或所有请求，请执行以下操作：

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

步骤5. PKI-Client:

同时，PKI客户端启动POLL计时器。在此，IOS定期执行GetCertInitial，直到客户端收到SCEP CertRep = GRANTED以及已授予的证书。

收到授权证书后，IOS会自动安装该证书。