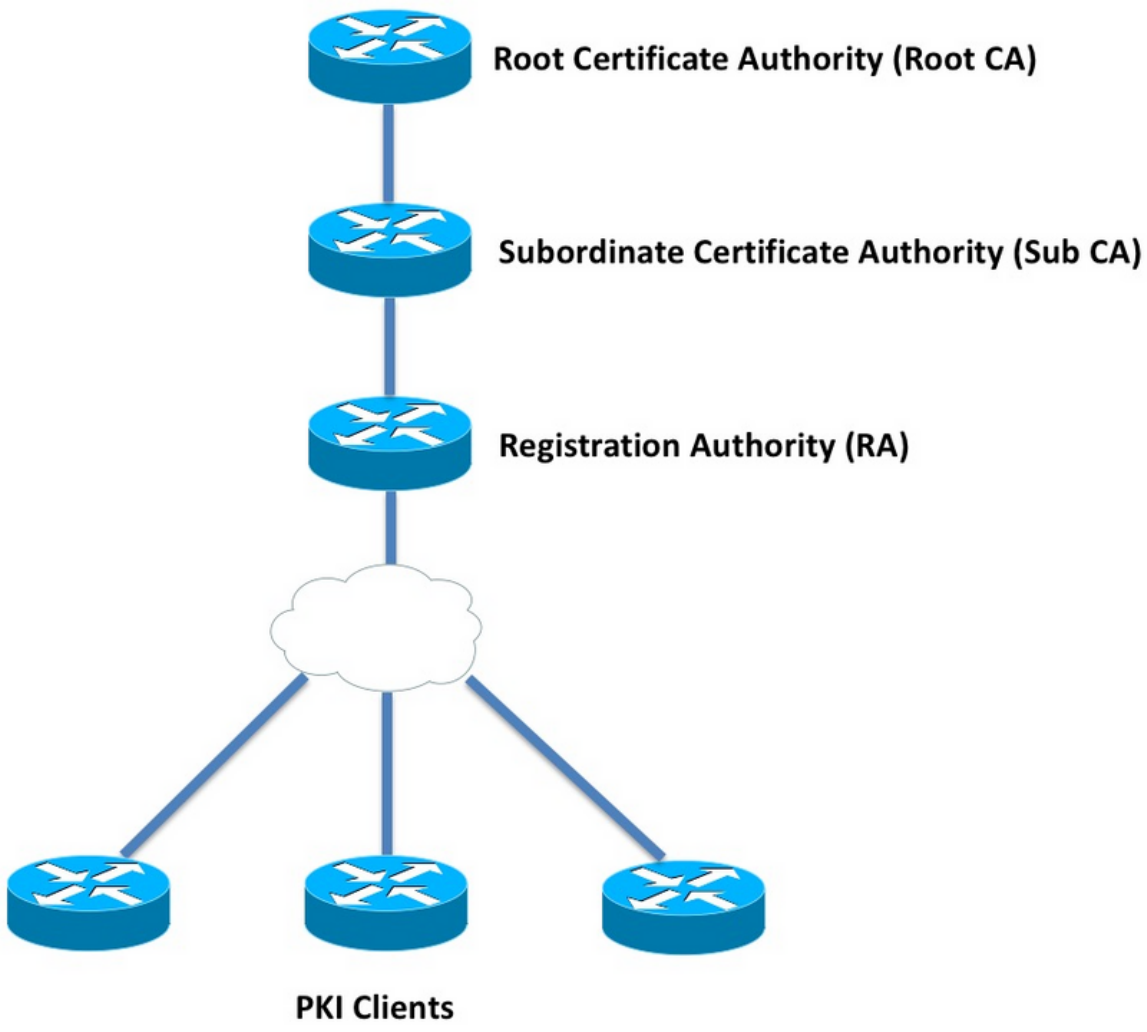
Руководство по развертыванию PKI IOS: исходное проектное решение и развертывания

Содержание

Введение

Этот документ описывает Сервер pki IOS и функциональность клиента подробно. Это обращается к исходному проектному решению PKI IOS и вопросам развертывания.

Инфраструктура PKI



Центр сертификации

Центр сертификации (CA), также называемый Сервером pki всюду по документу, является надежным объектом, который выполняет сертификаты. PKI основывается на доверии, и трастовая иерархия запускается в Корневом центре сертификации (Узел CA). Поскольку Узел CA наверху иерархии, он имеет подписанный сертификат.

Зависимый центр сертификации

В Трастовой иерархии PKI все центры сертификации ниже Root известны как Зависимые Центры сертификации (подCA). Очевидно, ПОДСЕРТИФИКАТ CA выполнен CA, который является одним уровнем выше.

PKI не накладывает ограничения на количество ПОДАВАРИИ в данной иерархии. Однако в развертывании в масштабе предприятия больше чем с 3 уровнями центров сертификации может стать трудным управлять.

Центр регистрации

PKI определяет специальный центр сертификации, известный как Центр регистрации (RA), который ответственен за авторизацию клиентов PKI от регистрации до данного подCA или Узла CA. RA не выполняет сертификаты клиентам PKI, вместо этого это решает, какой Клиент PKI может или не может быть выполнен сертификат подCA или Узлом CA.

Основная роль RA должна разгрузить основную проверку запроса сертификата клиента от CA и защитить CA от прямого воздействия до клиентов. Таким образом, RA стоит между клиентами PKI и CA, таким образом защищая CA от любого вида атак "отказ в обслуживании".

Клиент PKI

Любое устройство, запрашивающее на сертификат на основе резидентской обще-частной пары согласованных ключей удостоверять ее личность к другим устройствам, известно как клиент PKI.

Клиент PKI должен быть способен к генерации или хранению обще-частная пара согласованных ключей, такая как RSA или DSA или ECDSA.

Сертификат является доказательством идентичности и законностью данного общего ключа, если соответствующий закрытый ключ существует на устройстве.

Сервер pki IOS

Таблица 1. Развитие Функции Сервера pki IOS

Функция	IOS [ISR-G1, ISR-G2]	XE IOS [ASR1K, ISR4K]
CA/СЕРВЕР PKI IOS	12.3 (4) T	XE 3.14.0 / 15.5 (1) S
Одновременное нажатие клавиш сертификата сервера pki IOS	12.4 (1) T	XE 3.14.0 / 15.5 (1) S

PKI IOS HA	15.0 (1) M	NA [Неявное Резервирование Inter-RP доступен]
IOS RA для третьей стороны CA	15.1 (3) T	XE 3.14.0 / 15.5 (1) S

Перед вхождением в конфигурацию Сервера pki администратор должен понять эти базовые понятия.

Авторитетный источник времени

Одна из основ инфраструктуры PKI время. Системные часы определяют, допустим ли сертификат или нет. Следовательно, в IOS, часы должны быть сделаны авторитетными или защищенными. Без авторитетного источника времени Сервер pki может не функционировать как ожидалось, и это настоятельно рекомендовано для создания часов на IOS авторитетным использованием этих методов:

NTP (протокол сетевого времени)

Синхронизация системных часов с Временным сервером является единственным истинным способом сделать системные часы защищенными. Маршрутизатор IOS может быть настроен как клиент NTP к известному и стабильному серверу NTP в сети:

```
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 может также быть настроен как сервер NTP, который отметит часы локальной системы как авторитетные. В небольших развертываниях PKI Сервер pki может быть настроен как сервер NTP для его клиентов 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
```

Маркирование Аппаратных часов, как доверяется

В IOS аппаратные часы могут быть отмечены как авторитетное использование:

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

Это может быть настроено наряду с NTP и основной причиной для того, чтобы сделать, это должно поддержать системные часы авторитетными, когда перезагрузки маршрутизатора, например из-за перебоя в питании и серверов NTP не достижимы. На данном этапе таймеры PKI прекратят функционировать, который в свою очередь приводит к сбоям Обновления/Одновременного нажатия клавиш Сертификата. **clock calendar-valid** действует как гарантия в таких ситуациях.

При настройке этого это является ключевым, чтобы понять, что системные часы выйдут из синхронизования, если системная батарейка умрет, и PKI начнет доверять часам из синхронизования. Однако относительно более безопасно настроить это, чем не наличие авторитетного источника времени вообще.

> **Примечание: команда clock calendar-valid** была добавлена в XE версии XE IOS 3.10.0 / 15.3 (3) S вперед.

Имя хоста и доменное имя

Рекомендуется настроить имя хоста и domain-name на Cisco IOS как один из первых шагов прежде, чем настроить любые связанные сервисы PKI. Имя хоста маршрутизатора и domain-name используются в следующих сценариях:

- Название Открытых и секретных ключей криптосистемы RSA по умолчанию получено путем объединения имени хоста и domain-name
- При регистрации на сертификат subject-name по умолчанию состоит из атрибута имени хоста и неструктурировать-названия, которое является именем хоста и соединенным domain-name.

Что касается Сервера pki, не используются имя хоста и domain-name:

- Название пары согласованных ключей по умолчанию совпадет с названием названия Сервера pki
- Subject-name по умолчанию состоит из CN, который совпадает с cn названия Сервера pki.

Общие рекомендации должны настроить соответствующее имя хоста и domain-name.

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

Сервер HTTP

Сервер pki IOS включен, только если включен Сервер HTTP. Следует отметить, что, если Сервер pki отключен из-за отключаемого сервера HTTP, это может продолжить предоставлять офлайновые запросы [через терминал]. Возможность Сервера HTTP требуется, чтобы обрабатывать запросы SCEP и отсылать ответы SCEP.

С помощью Сервер HTTP IOS включают:

```
ip http server
```

И порт сервера HTTP по умолчанию может быть изменен от 80 до любого использования номера действительного порта:

```
ip http port 8080
```

Max-connection HTTP

Одно из узких мест, при развертывании IOS как Сервера pki с помощью SCEP является Максимальными параллельными соединениями HTTP и средними соединениями HTTP в минуту.
В настоящее время, максимальные параллельные соединения на Сервере HTTP IOS ограничен 5 по умолчанию и может быть увеличен до 16, который настоятельно рекомендован в средних развертываниях масштаба:

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

Этот IOS установки позволяет максимальные параллельные соединения HTTP до 1000:

- UniversalK9 IOS с установленным в лицензию uck9

CLI автоматически изменен для принятия числового аргумента между 1 - 1000

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

Сервер HTTP IOS позволяет 80 соединений в минуту [580 в случае IOS Release, где параллельные сеансы HTTP Max могут быть увеличены до 1000] и когда этот предел достигнут в течение минуты, HTTP-слушатель IOS начинает регулировать входящие соединения HTTP путем завершения слушателя в течение 15 секунд. Это приводит к запросам клиентского соединения, отбрасываемым из-за **достигнутого предельного размера очереди TCP - подключения**. Дополнительные сведения об этом могут быть найдены здесь

Открытые и секретные ключи криптосистемы RSA

Открытые и секретные ключи криптосистемы RSA для функциональности Сервера pki на IOS могут автоматически генерироваться или вручную генерироваться.
При настройке Сервера pki IOS автоматически создает Точку доверия тем же названием как Сервер pki для хранения сертификата Сервера pki.

Вручную генерирующие Открытые и секретные ключи криптосистемы RSA Сервера pki:

Шаг 1. Создайте Открытые и секретные ключи криптосистемы RSA с тем же названием как тот из Сервера pki:

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

Шаг 2. Прежде, чем включить Сервер pki, модифицируйте Точку доверия Сервера pki:

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

> **Примечание:** Значение модуля Открытых и секретных ключей криптосистемы RSA, упомянутое под точкой доверия Сервера pki, не учтено до версии IOS 15.4 (3) M4, и это - известное предупреждение. Модуль ключа по умолчанию составляет 1024 бита.

Автоматическая генерация открытых и секретных ключей криптосистемы RSA сервера pki:

При включении Сервера pki IOS автоматически генерирует Открытые и секретные ключи криптосистемы RSA с тем же названием как тот из Сервера pki, и ключевой размер модуля составляет 1024 бита.

Стартовая версия IOS 15.4 (3) M5, эта конфигурация создает Открытые и секретные ключи криптосистемы RSA с <МЕТКОЙ> как название, и эффективность ключа будет согласно определенному модулю <mod>.

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

Спойлер

Сервер pki CSCuu73408 IOS должен обеспечить размер ключа ня по умолчанию для свидетельства одновременного нажатия клавиш.

Сервер pki CSCuu73408 IOS должен обеспечить размер ключа ня по умолчанию для свидетельства одновременного нажатия клавиш.

Современный промышленный стандарт должен использовать минимум Открытых и секретных ключей криптосистемы RSA на 2048 битов.

Рассмотрение таймера auto-rollover

В настоящее время Сервер pki IOS не генерирует сертификат одновременного нажатия клавиш по умолчанию, и он должен быть явно включен под Сервером pki с помощью команды **<days-before-expiry> auto-rollover**. В больше на одновременном нажатии клавиш Сертификата объясняют

Эта команда задает, сколько за дни до того, как Сервер pki / истечение СЕРТИФИКАТА CA должен IOS создавать сертификат CA одновременного нажатия клавиш. Обратите внимание на то, что сертификат CA одновременного нажатия клавиш активирован, как

только истекает текущий активный сертификат CA. Значение по умолчанию в настоящее время составляет 30 дней. Это значение должно быть установлено в рыночную стоимость в зависимости от срока действия сертификата CA, и это в свою очередь влияет на конфигурацию таймера автоматической регистрации на клиенте PKI.

> **Примечание:** Таймер Auto-rollover должен всегда инициировать до таймера автоматической регистрации на клиенте во время CA и одновременного нажатия клавиш Сертификата клиента [известный как]

Факторы CRL

Поддержка инфраструктуры PKI IOS два способа распределить CRL:

Публикуйте CRL в сервере HTTP

Сервер pki IOS может быть настроен для публикации файла CRL в определенном местоположении на Сервере HTTP с помощью этой команды под Сервером pki:

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

И Сервер pki может быть настроен для встраивания этого местоположения CRL во все сертификаты клиента PKI с помощью этой команды под Сервером pki:

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

Метод SCEP GetCRL

Сервер pki IOS автоматически хранит файл CRL в определенном расположении базы данных, которое по умолчанию является nvram и настоятельно рекомендовано для хранения копии на SCP/FTP/СЕРВЕРЕ TFTP с помощью этой команды под Сервером pki:

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

По умолчанию Сервер pki IOS не встраивает местоположение CDP в сертификаты клиента PKI. Если клиенты PKI IOS настроены для выполнения проверки аннулирования, но проверяемому сертификату не встроили CDP в нее, и проверка CA точка доверия настроена с местоположением CA (использующий http://<IP - сервер CA или FQDN>), IOS переключается на основанный метод GetCRL SCEP по умолчанию.
SCEP GetCRL выполняет, извлечение CRL путем выполнения HTTP Входят в этот URL:

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

> **Примечание:** В интерфейсе командной строки IOS, до ввода**?**, нажмите **Ctrl + V** основных последовательностей.

Сервер pki IOS может также встроить этот URL как местоположение CDP. Преимущество выполнения этого является двукратным:

- Это гарантирует, что весь SCEP не-IOS базировался, клиенты PKI могут выполнить извлечение CRL.
- Без встроенного CDP сообщения запроса IOS SCEP GetCRL подписаны (использование временного подписанного сертификата), как определено в проекте SCEP. Однако поисковые запросы CRL не должны быть подписаны, и путем встраивания URL CDP для метода GetCRL, подписания запросов CRL можно избежать.

Срок действия CRL

Срок действия CRL Сервера pki IOS может управляться с помощью этой команды под Сервером pki:

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

Значение находится в часах. По умолчанию срок действия CRL установлен в 6 часов. В зависимости от того, как часто отозваны сертификаты, настраивание срока действия CRL к оптимальному значению увеличивает производительность извлечения CRL в сети.

Факторы базы данных

Сервер pki IOS использует nvram в качестве расположения базы данных по умолчанию, и это настоятельно рекомендовано для использования FTP или TFTP или сервера SCP как расположение базы данных. По умолчанию Сервер pki IOS создает два файла:

- <Имя сервера> .ser – Это содержит последний серийный номер, выполненный CA в hex. Файл находится в формате простого текста, и это содержит эту информацию:
  db_version = 1
  last_serial = 0x4
- <Имя сервера> .crl – Это - закодированный файл CRL DER, публикованный CA

Сервер pki IOS хранит информацию в базе данных на 3 конфигурируемых уровнях:

- Минимум – Это - уровень по умолчанию, и на этом уровне никакой файл не создан в базе данных, и следовательно никакая информация не доступна на сервере CA относительно сертификатов клиента, предоставленных в прошлом.
- Названия – В этом Сервере pki IOS уровня создают файл под названием <Серийный номер> .cnm для каждого выполненного сертификата клиента, где название <Серийный номер> обращается к серийному номеру выполненного сертификата клиента, И этот cnm файл содержит subject-name и дату окончания срока действия сертификата клиента.

- Завершенный – На этом уровне, Сервер pki IOS создает два файла для каждого выполненного сертификата клиента:
- <Серийный номер> .cnm
- <Серийный номер> .crt

здесь, crt файл является файлом сертификата клиента, который является закодированным DER.

Эти точки важны:

- Прежде, чем выполнить сертификат клиента, Сервер pki IOS обращается к <Имени сервера> .ser определить и получить Серийный номер сертификата.
- С набором Database level к Названиям или Завершенный, <Серийный номер> .cnm и <Серийный номер> .crt должен быть записан в базу данных прежде, чем передать, предоставил/выполнил сертификат клиенту
- С набором database url к Названиям или Завершенный, database URL должен иметь достаточно пространства, чтобы сохранить файлы. Следовательно рекомендация состоит в том, чтобы настроить внешний файловый сервер [FTP или TFTP или SCP] как database URL.
- С настроенным URL Внешней базы данных абсолютно необходимо удостовериться, что файловый сервер достижим во время процесса предоставления сертификата, который иначе отметил бы Сервер CA, как отключено. И ручное вмешательство требуется, чтобы возвращать сервер CA онлайн.

Database archive

При развертывании Сервера pki важно рассмотреть сценарии отказов и быть подготовленным, должен там быть hadrware сбой. Это можно достичь двумя способами:

1. Резервирование
   В этом случае два устройства или процессоры действуют как Активный Резерв для обеспечивания избыточности.
   Сервер pki IOS highavailability может быть достигнут с помощью включенных комплектов маршрутизаторов ISR двух HSRP [ISR G1 и ISR G2], как объяснено в

   XE IOS базировал системы [ISR4K и ASR1k] не имеют опцию избыточности устройства в наличии. Однако в резервировании ASR1k Inter-RP доступно по умолчанию.
2. Архивация пары согласованных ключей Сервера CA и файлов
   IOS предоставляет услугу для архивации Пары согласованных ключей Сервера pki и сертификата. Архивация может быть сделана с помощью двух типов файлов:
   PEM - IOS создает отформатированные файлы PEM для хранения открытого ключа RSA, Зашифрованного закрытого ключа RSA, CA Серверный сертификат. Пара согласованных ключей одновременного нажатия клавиш и сертификаты автоматически заархивированыPKCS12 - IOS создает одиночный файл PKCS12, содержащий Серверный сертификат CA, и соответствующий закрытый ключ RSA зашифровал использование пароля.

Архивация базы данных может быть включена с помощью этой команды под Сервером pki:

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

Также возможно хранить заархивированные файлы к отдельному серверу, возможно с помощью защищенного протокола (SCP) с помощью следующей команды под Сервером pki:

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

Из всех файлов в базе данных за исключением заархивированных файлов и.Ser файла, все другие файлы находятся в открытом тексте и не представляют реальной угрозы, если потеряно, и следовательно могут храниться на отдельном сервере, не подвергаясь большому количеству издержек при записи файлов, например Сервер TFTP.

IOS как подCA

Сервер pki IOS по умолчанию приводит роль в рабочее состояние Узла CA. Для настройки зависимого Сервера pki (подCA) сначала выполните эту команду под разделом конфигурации Сервера pki (перед включением Сервера pki):

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

Использование этого настраивает URL Узла CA под точкой доверия Сервера pki:

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

Включение этого Сервера pki теперь инициирует эти события:

- Точка доверия Сервера pki аутентифицируется для установки Корневого сертификата CA.
- После того, как Узел CA аутентифицируется, IOS генерирует CSR для Подчиненного CA [x509 основное ограничение, содержащее флаг CA:TRUE], и передает его к Узлу CA

Независимо от режима предоставления, настроенного на Узле CA, IOS помещает CA (или RA) запросы сертификата в очередь в состоянии ожидания. Администратор должен вручную предоставить сертификаты CA.
Просмотреть запрос сертификата в состоянии ожидания и идентификатор запроса:

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

Предоставить запрос:

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

- Использование этого, последующий ОПРОС SCEP (GetCertInitial) операция загружает ПОДСЕРТИФИКАТ CA и устанавливает его на маршрутизаторе, который включает Зависимый Сервер pki

IOS как RA

Сервер pki IOs может быть настроен как Центр регистрации данному Подчиненному или Узлу CA. Для настройки Сервера pki как центра регистрации сначала выполните эту команду под разделом конфигурации Сервера pki (перед включением Сервера pki):

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

После этого настройте URL CA под точкой доверия Сервера pki. Это указывает, который CA защищен RA:

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

Центр регистрации не выполняет сертификаты, следовательно конфигурация **issuer-name** под RA не требуется и не эффективная, даже если это настроено. Subject-name RA настроен под точкой доверия RA с помощью **команды subject-name**. Важно настроить **OU = ioscs RA** как часть subject-name для IOS CA, чтобы определить IOS RA т.е. определить запросы сертификата, авторизовавшие IOS RA.

IOS может действовать как Центр регистрации к третьей стороне CAs, такой как Microsoft CA, и для пребывания совместимым IOS, который RA должен быть включен с помощью этой команды под разделом конфигурации Сервера pki (перед включением Сервера pki):

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

В режиме RA по умолчанию IOS подписывает запросы клиента [PKCS#10] с помощью сертификата RA. Эта операция указывает на Сервер pki IOS, что запрос сертификата авторизовался RA.

С прозрачным режимом RA, IOS вперед запросы клиента в их исходном формате, не представляя сертификат RA, и это совместимо с Microsoft CA как известный пример.

Клиент PKI IOS

Одним из самого важного экземпляра конфигурации в Клиенте PKI IOS является Точка доверия. Параметры конфигурации точки доверия объяснены подробно в этом разделе.

Авторитетный источник времени

Как указал ранее, авторитетный источник времени является требованием на клиенте PKI также. Клиент PKI IOS может быть настроен как клиент NTP, использующий их конфигурация:

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

Имя хоста и доменное имя

Общие рекомендации должны настроить имя хоста и domain-name на маршрутизаторе:

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

Открытые и секретные ключи криптосистемы RSA

В Клиенте PKI IOS Открытые и секретные ключи криптосистемы RSA для данной регистрации точки доверия могут или автоматически генерироваться или вручную генерироваться.

Автоматический процесс генерации ключа RSA включает придерживающееся:

- IOS по умолчанию создает Открытые и секретные ключи криптосистемы RSA на 512 битов
- Автоматически генерируемое название пары согласованных ключей является hostname.domain-названием, которое является именем хоста устройства, объединенным с domain-name устройства
- Автоматически созданная пара согласованных ключей не отмечена как экспортная.

Автоматический процесс генерации ключа RSA включает придерживающееся:

- Дополнительно, Открытые и секретные ключи криптосистемы RSA общего назначения подходящей силы могут вручную генерироваться с помощью:
- `crypto pki trustpoint <RA-SERVER-Name>`
  `enrollment url <CA URL>`
  `subject-name CN=<Common Name>, **OU=ioscs RA**, OU=TAC, O=Cisco`Здесь, МЕТКА - название Открытых и секретных ключей криптосистемы RSA

  MOD - модуль ключа RSA или сила в битах между 360 до 4096, который является традиционно 512, 1024, 2048 или 4096.
  Преимущество ручной генерации Открытых и секретных ключей криптосистемы RSA является способностью отметить пару согласованных ключей как экспортную, который в свою очередь обеспечивает сертификат идентификации, который будет полностью экспортироваться, который может тогда быть восстановлен на другом устройстве. Однако нужно понять последствия для системы безопасности этого действия.
- Открытые и секретные ключи криптосистемы RSA связаны с точкой доверия перед регистрацией с помощью этой команды

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

  Здесь, если Открытые и секретные ключи криптосистемы RSA под названием <МЕТКА> уже существуют, то это взято во время регистрации точки доверия.

  Если Открытые и секретные ключи криптосистемы RSA под названием <МЕТКА> не существуют, то одно из следующего действия выполняется во время регистрации:

  - Если никакой аргумент <mod> не передают, то пара согласованных ключей на 512 битов под названием <МЕТКА> генерируется.

  - если один аргумент <mod> передают, то пара согласованных ключей общей цели битов <mod> под названием <МЕТКА> генерируется

  - если два аргумента <mod> передают, то одна пара согласованных ключей подписи битов <mod> и одна пара согласованных ключей шифрования битов <mod>, оба названных <МАРКИРУЮТ>, генерируются

Точка доверия

Точка доверия является абстрактным контейнером для удержания сертификата в IOS. Одиночная точка доверия способна к хранению двух активных сертификатов в любое заданное время:

- Сертификат CA - Загрузка сертификата CA в данную точку доверия известна как процесс проверки подлинности точки доверия.
- Сертификат ID, выполненный CA - Загрузка или Импорт сертификата ID в данную точку доверия, известен как процесс регистрации точки доверия.

Конфигурация точки доверия известна как трастовая политика, и это определяет это:

- Какой сертификат CA загружен в точке доверия?
- К которому CA регистрируется точка доверия?
- Как IOS регистрирует точку доверия?
- Как проверен сертификат, выполненный данным CA [загруженный в точке доверия]?

Основные компоненты точки доверия объяснены здесь.

Режим регистрации

Режим регистрации точки доверия, который также определяет режим аутентификации точки доверия, может быть выполнен через 3 основных способа:

1. Предельная Регистрация - ручной способ выполняющей аутентификации точки доверия и приема сертификата с помощью вставки копии в терминале CLI.
2. Регистрация SCEP - аутентификация Точки доверия и регистрация с помощью SCEP по HTTP.
3. Профиль регистрации - Здесь, аутентификация и методы регистрации определен отдельно. Наряду с терминалом и методами регистрации SCEP, профили регистрации предоставляют возможность задавать команды HTTP/TFTP для выполнения извлечения файла от Сервера, который определен с помощью аутентификации или enrollment url под профилем.

Исходный интерфейс и VRF

Аутентификация точки доверия и прием по HTTP (SCEP) или TFTP (Профиль Регистрации) используют Файловую систему IOS для выполнения операций ввода-вывода файла. Эти обмены пакетами могут быть получены от определенного исходного интерфейса и VRF.

В случае классической конфигурации точки доверия эта функциональность добавлена с помощью **подкоманд исходного интерфейса и vrf** под точкой доверия.

В случае профилей регистрации **исходного интерфейса** и **регистрации | URL аутентификации <http/tftp://Расположение сервера>** команды **<vrf-name> VRF** предоставляют ту же функциональность.

Пример конфигурации:

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

или

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

Автоматическое хранилище сертификатов и обновление

Клиент PKI IOS может быть настроен для выполнения автоматической регистрации и обновления с помощью этой команды под разделом точки доверия PKI:

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

Здесь, **автоматические регистрации <процент> [восстанавливают]** состояния команды, что IOS должен выполнить обновление сертификата точно в 80% срока действия текущего сертификата.

Ключевое слово **восстанавливает** состояния, что IOS должен восстановить Открытые и секретные ключи криптосистемы RSA, известные как теневая пара согласованных ключей во время каждой операции обновления сертификата.

Это - автоматическое поведение регистрации:

- **Автоматическая регистрация** момента настроена, если точка доверия будет аутентифицироваться, то IOS выполнит автоматическую регистрацию к серверу, расположенному в URL, упомянутом как часть **команды enrollment url** под разделом точки доверия PKI или под профилем регистрации.
- Момент точка доверия зарегистрирована с Сервером pki или CA, RENEW или таймером SHADOW, инициализирован на клиенте PKI на основе процента **автоматической регистрации** от текущего сертификата идентификации, установленного под точкой доверия. Этот таймер видим под **показом крипто-команда timer pki**. Больше на таймере fuctions *обращаются к*
- Поддержка возможности обновления приходит от Сервера pki. Больше на этом в Клиент PKI IOS выполняет два типа обновления:
  Неявное Обновление: Если Сервер pki не передает "Обновление" как поддерживаемую возможность, IOS выполняет начальную регистрацию в определенном проценте автоматической регистрации. т.е. IOS использует подписанный сертификат для подписания запроса на обновление.Явное Обновление: Когда Сервер pki поддерживает функцию обновления сертификата клиента PKI, он объявляет "Обновление" как поддерживаемую возможность. IOS принимает эту возможность во внимание во время обновления сертификата, т.е. IOS использует текущий активный сертификат идентификации для подписания запроса сертификата обновления.

Меры должны быть приняты при настройке процента автоматической регистрации. На любом данном клиенте PKI в развертываниях, если условие возникает, где сертификат идентификации истекает в то же время, что и сертификат CA запуска, тогда значение автоматической регистрации должно всегда инициировать [теневую] операцию обновления после того, как CA создал сертификат одновременного нажатия клавиш. *См.* **PKI зависимости от Таймера** разделяют в

Revocation-check сертификата

Аутентифицируемая точка доверия PKI т.е. точка доверия PKI, содержащая сертификат CA, способны к выполняющей проверке достоверности сертификата во время IKE или согласования SSL, где Сертификат однорангового узла подвергнут полной проверке достоверности сертификата. Один из методов проверки должен проверить статус аннулирования сертификата однорангового узла с помощью одного из следующих двух методов:

- Список отозванных сертификатов (CRL) - Это - файл, содержащий Серийные номера

сертификатов, отозванных данным CA., Этот файл подписан с помощью сертификата CA запуска. Метод CRL включает загрузку файла CRL с помощью HTTP или LDAP.

- Онлайновый протокол статуса сертификата (OCSP) - IOS устанавливает канал связи с объектом, вызванным как Респондент OCSP, который является определяемым Сервером CA. Запуска, Клиент, такой как IOS отправляет запрос, содержащий серийный номер сертификата, проверяется. Респондент OCSP отвечает статусом аннулирования данного серийного номера. Канал связи мог быть установлен с помощью любого Поддерживаемого приложения / транспортный протокол, который обычно является HTTP.

Проверка аннулирования может быть определена с помощью них, дают команду под разделом точки доверия PKI:

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

По умолчанию точка доверия настроена для выполнения проверки аннулирования с помощью crl.

Методы могут быть переупорядочены, и проверка статуса аннулирования выполнена в определенном заказе. Метод "ни один" не обходит revocation-check.

Кэш CRL

С основанным revocation-check CRL каждая проверка достоверности сертификата может инициировать новую загрузку файла CRL. И поскольку файл CRL становится больше или если CRL Distribution Point (CDP) более далек, загружение файла во время каждого процесса проверки данных препятствует производительности зависимости от протокола на проверке достоверности сертификата. Следовательно, кэширование CRL выполнено для улучшения производительности, и кэширование CRL принимает законность CRL во внимание.

Законность CRL определена с помощью двух раз параметры: LastUpdate, который является прошлый раз CRL, был публикован запуском CA, и NextUpdate, который является временем, является будущим, когда новая версия файла CRL публикована CA. запуска

IOS кэширует каждый загруженный CRL столько, сколько CRL допустим. Однако при определенных обстоятельствах, таких как CDP, не являющийся достижимым временно, может быть необходимо сохранить CRL в кэше для длительного периода времени. В IOS кэшируемый CRL может сохраненный столько, сколько спустя 24 часа после того, как законность CRL истекает, и это может быть настроено с помощью этой команды под разделом точки доверия PKI:

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

При определенных обстоятельствах, таких как запуск, CA отзывающий сертификаты в периоде достоверности CRL, IOS может confiugured для удаления кэша более часто. Путем удаления CRL преждевременно, IOS вынужден загружать CRL более часто для хранения кэша CRL актуальным. Эти параметры конфигурации доступны под разделом точки доверия PKI:

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

И наконец, IOS может быть настроен для не кэширования файла CRL с помощью этой команды под разделом точки доверия PKI:

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

Рекомендуемая конфигурация

Типичные развертывания CA с Узлом CA и конфигурацией подCA как ниже. Пример также включает конфигурацию подCA, защищенную RA.

С Открытыми и секретными ключами криптосистемы RSA на 2048 битов через плату данный пример рекомендует настройку где:
Узел CA имеет срок действия 8 лет
ПодCA имеет срок действия 3 лет
Сертификаты клиента выполнены в течение года, которые являются configred для запроса на обновление сертификата, автоматически.

УЗЕЛ CA - конфигурация

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

SUBCA без RA - конфигурация

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

SUBCA с RA - конфигурация

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

RA для SUBCA - конфигурация

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

Хранилище сертификатов

Manual enrollment

Manual enrollment включает офлайновую генерацию CSR на клиенте PKI, который вручную скопирован Администратору CA., вручную подписывает запрос, который тогда импортирован в клиента.

Клиент PKI

Конфигурация клиента PKI:

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

Шаг 1. Сначала аутентифицируйте Точку доверия (это может также быть выполнено после шага 2).

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

Шаг 2. Генерируйте Запрос подписи сертификата и Возьмите CSR к CA и получите предоставленный сертификат:

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

Шаг 3. Теперь импортируйте предоставленный сертификат через терминал:

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

Сервер pki

Шаг 1. Сначала экспортируйте сертификат CA Запуска от CA, который в этом случае является сертификатом SUBCA. Это импортировано во время шага 1 выше в клиента PKI, т.е. аутентификацию Точки доверия.

```
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-----
```

Шаг 2. После неродной 2 на Клиенте PKI, возьмите CSR от клиента и предоставьте его для подписания на SUBCA, использующем эту команду:

```
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-----
```

Эта команда предполагает, что SUBCA принимает запрос подписи сертификата от терминала, и когда-то предоставленный, данные сертификата распечатаны в формате PEM.

```
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-----
```

Если CA находится в режиме автопредоставления, предоставленный сертификат отображен в формате PEM выше. Когда CA находится в ручном режиме предоставления, запрос сертификата отмечен как **ожидание**, назначен значение идентификатора и помещен в очередь в очередь запроса регистрации.

```
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-----
```

Шаг 3. Вручную предоставьте этот запрос с помощью этой команды:

```
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-----
```

Примечание: Manual enrollment подCA к Узлу CA не возможен.

Примечание: CA в отключенном состоянии из-за отключенного сервера HTTP может вручную предоставить запросы сертификата.

Регистрация с помощью SCEP

Конфигурация клиента PKI:

```
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
```

Конфигурация Сервера pki:

```
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
```

Режим по умолчанию запроса сертификата, предоставляющего, является ручным:

```
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
```

Ручное предоставление

Шаг 1. Клиент PKI: Как первый шаг, который является обязательным, аутентифицируют точку доверия на клиенте PKI:

```
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
```

Шаг 2. Клиент PKI: После аутентификации точки доверия клиент PKI может быть зарегистрирован на сертификат.

> **Примечание:** Если автоматическая регистрация будет настроена, то клиент автоматически выполнит регистрацию.

```
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
```

Негласно, эти события имеют место:

- IOS ищет Открытые и секретные ключи криптосистемы RSA под названием Ключ PKI. Если это существует, это взято для запроса сертификата идентификации. В противном случае IOS создает пару согласованных ключей на 2048 битов под названием Ключ PKI, и затем используйте его для запроса сертификата идентификации.

- IOS создает запрос подписи сертификата в формате PKCS10.

- IOS тогда шифрует этот CSR с помощью случайного симметричного ключа. Случайный симметричный ключ зашифрован с помощью открытого ключа получателя, который является SUBCA (открытый ключ SUBCA доступен из-за аутентификации точки доверия). Зашифрованный CSR, зашифрованный случайный симметричный ключ и сведения о получателе соединены в окутанных данных PKCS#7.

- Окутанные данные этого PKCS#7 подписаны с помощью временного подписанного сертификата во время начальной регистрации. PKCS#7 окутал данные, сертификат подписания, используемый клиентом и подписью клиента, соединен в подписанном пакете данных PKCS#7. Это - base64, закодированный, и затем закодированный URL. Получающийся блоб данных передается как аргумент “сообщения” в URI HTTP, передаваемом CA:

```
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
```

Шаг 3. PKI-Server:

Когда Сервер pki IOS получает запрос, он проверяет их:

1. Проверки, если база данных запроса регистрации содержит запрос сертификата с тем же идентификатором транзакции, привязанным к новому запросу.

Примечание: Идентификатор транзакции является хэшем MD5 общего ключа, на который сертификат идентификации запрашивает клиент.

2. Проверки, если база данных запроса регистрации содержит запрос сертификата с тем же паролем вызова как тот, передаваемый клиентом.

Примечание: Если (1) возвращает true или и (1), и (2) вместе возвращают true, то сервер CA способен к отклонению запроса по причине двойного идентификационного запроса. Однако в таком IOS случая Сервер pki заменяет более старый запрос более новым запросом.

Шаг 4. . PKI-Server:

Вручную предоставьте запросы на Сервере pki:

Просмотреть запрос:

```
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
```

Предоставить конкретный запрос или все запросы:

```
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
```

Шаг 5. . Клиент PKI:

Между тем клиент PKI запускает таймер ОПРОСА. Здесь, IOS выполняет GetCertInitial через определенные промежутки времени, пока SCEP CertRep = ПРЕДОСТАВЛЕННЫЙ наряду с предоставленным сертификатом не получен клиентом.

Как только предоставленный сертификат получен, IOS автоматически устанавливает его.