Пример настройки доступа к коммутатору по SSL

Протокол HTTP не предусматривает шифрование и не обеспечивает безопасность передаваемых данных, поэтому для доступа к Web-интерфейсу управления коммутатора рекомендуется использовать безопасный протокол HTTPS. Для работы протокола HTTPS требуется установить на коммутаторе цифровой сертификат X.509.

Сертификат X.509 — это файл, в котором содержится открытый ключ, информация об организации, IP-адрес или доменное имя. Сертификат проверяет и подписывает закрытым ключом **удостоверяющий центр** (Certificate Authorities, CA). Для публичных сайтов важно, чтобы сертификат X.509 подписал один из доверенных удостоверяющих центров — Comodo, GeoTrust, Rapid SSL, Symantec. Помимо защиты соединения, цифровой сертификат, подписанный удостоверяющим центром, даёт гарантию, что сертификат не был подделан и подтверждает, что открытый ключ, содержащийся в сертификате, принадлежит владельцу сертификата.

Для доступа к коммутатору по SSL можно использовать самоподписанный сертификат X.509. Создать его можно с помощью инструмента OpenSSL в Linux.

> **Примечание**
>
> В дистрибутиве Linux Ubuntu 18.04 утилита **OpenSSL** установлена по умолчанию. В примере даны команды для версии `OpenSSL 1.0.2n 7 Dec 2017`. Для проверки версии введите команду:
>
> ```
> $ ssh -V
> ```

Задача

Нужно обеспечить безопасный доступ к веб-интерфейсу коммутатора. Задача решается настройкой доступа к веб-интерфейсу коммутатора по протоколу HTTPS.

Примечание к настройке

Рассматриваемый пример настройки подходит для следующих серий коммутаторов: DGS-1250, DGS-1510, DGS-1520, DGS-3130, DGS-3630, DXS-3610.

Рис. 1. Схема подключения

Создание корневого сертификата удостоверяющего центра (root CA)

1. На рабочей станции запустите терминал и создайте 2048-битный закрытый ключ для удостоверяющего центра CA:

```
$ openssl genrsa -out rootCA.key 2048
```

2. Сгенерируйте самоподписанный корневой сертификат CA:

```
$ openssl req -x509 -new -key rootCA.key -days 365 -out rootCA.crt
```

Заполните поля, как показано на рисунке 2.

Описание опций:
-x509 — создает самоподписанный сертификат;
-new — запрашивает у пользователя информацию об организации и доменном имени;
-key — указывает файл, из которого следует считать закрытый ключ;
-days — задает срок действия сертификата в днях. По умолчанию 30 дней;
-out — указывает имя файла для записи результатов выполнения команды

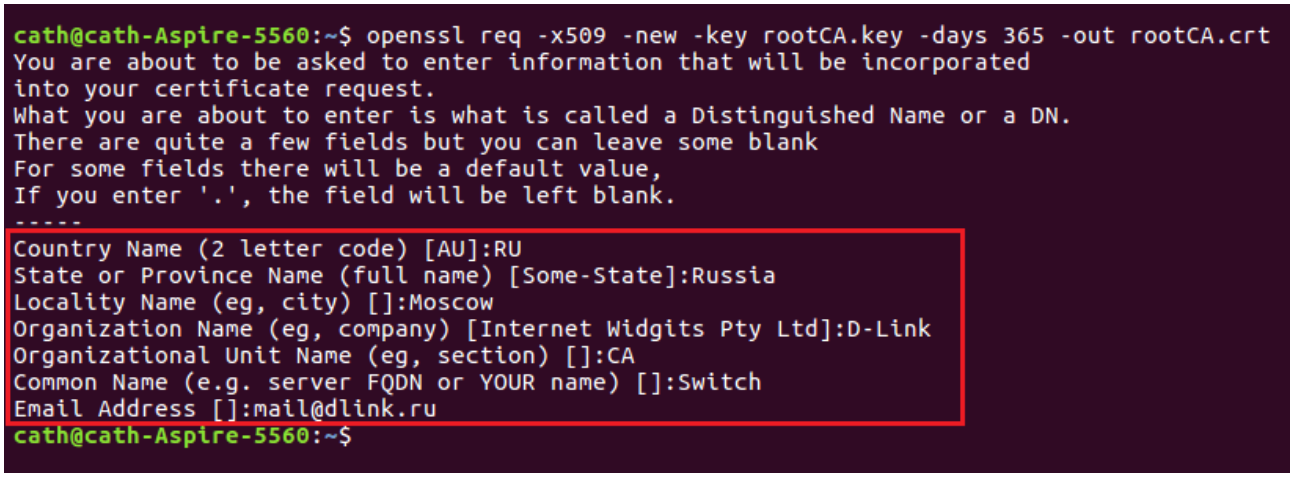

Рис. 2. Создание корневого сертификата CA

Примечание

В корневом сертификате `rootCA.crt` содержится открытый ключ и информация об удостоверяющем центре.

Закрытым ключом `rootCA.key` в дальнейшем будет подписан пользовательский сертификат, который нужно загрузить на коммутатор.

Посмотреть закрытый ключ `rootCA.key` можно командой в терминале:

```
$ openssl rsa -check -in rootCA.key
```

Описание опций:
-in — указывает файл с закрытым ключом;
-check — проверяет целостность закрытого ключа RSA.

Закрытый ключ `rootCA.key` корневого сертификата CA следует держать в секрете. В случае компрометации ключа необходимо создать новый закрытый ключ и самоподписанный корневой сертификат CA, а также заново подписать ранее выданные пользовательские сертификаты.

Посмотреть содержимое корневого сертификата `rootCA.crt` можно командой в терминале:

```
$ openssl x509 -text -in rootCA.crt
```

Описание опций:
-text — выводит сертификат для просмотра в текстовом виде;

-in — указывает файл сертификата

> **Примечание**
>
> Если доступ к Web-интерфейсу коммутатора выполняется с рабочей станции Windows, установите корневой сертификат `rootCA.crt` в браузер:
>
> **Firefox:** Настройки → Сертификаты → Просмотр сертификатов → Центры сертификации → Импортировать
>
> **Chrome:** Настройки → Настроить сертификаты → Доверенные корневые центры сертификации → Импорт
>
> **Internet Explorer:** Свойство браузера → Содержание → Сертификаты → Импорт

Создание пользовательского сертификата

1. Создайте 2048-битный закрытый ключ для пользовательского сертификата:

```
$ openssl genrsa -out switch.key 2048
```

2. Сгенерируйте запрос на подпись сертификата:

```
$ openssl req -new -key switch.key -out switch.csr
```

Заполните поля, как показано на рисунке 3.

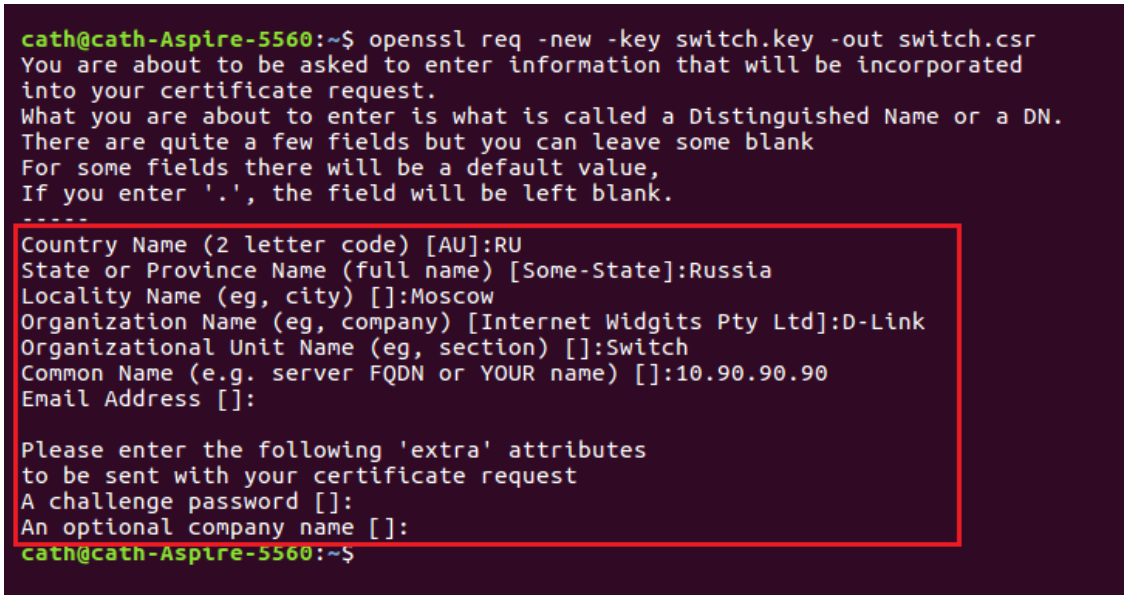

Рис. 3. Создание запроса на подпись сертификата

3. Подпишите запрос закрытым ключом `rootCA.key` корневого сертификата удостоверяющего центра (команда вводится в одну строку):

```
$ openssl x509 -req -in switch.csr -CA rootCA.crt -CAkey rootCA.key
-CAcreateserial -out switch.crt -days 365
```

Описание опций:
-req — ожидает заявку на сертификат;
-in — указывает файл с запросом на подпись сертификата;
-CA — указывает корневой сертификат CA;
-CAkey — указывает закрытый ключ корневого сертификата CA;
-CAcreateserial — создает файл с серийным номером сертификата;
-out — указывает имя файла для записи результатов выполнения команды;
-days — задает срок действия сертификата в днях. По умолчанию 30 дней.

4. Переименуйте файлы:

```
$ sudo mv rootCA.crt cacert.ca
$ sudo mv switch.crt cacert.crt
$ sudo mv switch.key cacert.prv
```

5. Переместите закрытый ключ и корневой сертификат CA в `/etc/ssl/`

```
$ sudo mv rootCA.key /etc/ssl/private/
$ sudo mv cacert.ca /etc/ssl/certs/
```

Настройка TFTP-сервера в Linux

1. Переместите подписанный сертификат и закрытый ключ в рабочую директорию TFTP-сервера:

```
$ sudo mv cacert.crt /var/lib/tftpboot/
$ sudo mv cacert.prv /var/lib/tftpboot/
$ sudo mv cacert.ca /var/lib/tftpboot/
```

> **Примечание**
>
> Подписанный сертификат и закрытый ключ загружается на коммутатор с помощью TFTP-сервера. При этом сервер TFTP должен находиться в той же IP-подсети, что и коммутатор. В рабочую папку установленного на рабочей станции сервера TFTP необходимо поместить подписанный сертификат `cacert.crt` и закрытый ключ `cacert.prv`. По умолчанию TFTP-сервер в Linux не установлен.
>
> Проверить статус TFTP-сервера можно командой:
> `$ sudo service tftpd-hpa status`

2. Настройте на рабочей станции статический IP-адрес:

```
$ sudo ifconfig enp0s3 10.90.90.99/8
```

Настройка коммутатора

1. Запустите утилиту **Minicom**:

```
$ sudo minicom
```

2. Создайте центр доверия с именем **test**:

```
Switch(config)# crypto pki trustpoint test
```

3. Импортируйте на коммутатор подписанные сертификаты `cacert.ca`, `cacert.crt` и закрытый ключ `cacert.prv`:

```
Switch(config)# crypto pki import test pem tftp: //10.90.90.99/cacert both
```

> **Примечание**
>
> Посмотреть настройки точек доверия на коммутаторе можно командой:
> `show crypto pki trustpoints`

4. Настройте политику службы SSL с именем **po**:

```
Switch(config)# ssl-service-policy po
```

Примечание

Посмотреть настройки политики службы SSL на коммутаторе можно командой:

```
show ssl-service-policy
```

5. Включите протокол SSL:

```
Switch(config)# ip http secure-server ssl-service-policy po
```

Примечание

При включении протокола SSL автоматически отключается доступ к Web-интерфейсу по протоколу HTTP.