Запрос и установка серверного сертификата на CSS11500

Содержание

Введение

Если у вас нет существующих ранее ключей и сертификатов для Коммутатора контент-сервисов (CSS), можно генерировать их на CSS. CSS включает серию сертификата и программ для управления с закрытым ключом для упрощения процесса генерации секретных ключей, Запросов подписи сертификата (CSR) и самоподписанных временных сертификатов. Этот документ описывает процесс для получения нового сертификата от центра сертификации (CA) и установки его к CSS.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.

Настройка

В этом разделе содержатся сведения о настройке функций, описанных в этом документе.

Примечание: Дополнительные сведения о командах, использованных в данном документе, см. в разделе Средство поиска команд (только для зарегистрированных клиентов).

Конфигурации

В данном документе используются следующие конфигурации.

- Генерируйте Rivest, Shamir и Adelman (RSA) пара ключей
- Привяжите файл открытых и секретных ключей криптосистемы RSA
- Генерируйте CSR
- Получите сертификат из CA
- Импортируйте цепочечный файл сертификата
- Привяжите файл сертификата
- Настройте список SSL прокси
- Настройте сервис протокола SSL и правила содержимого

Генерируйте Rivest, Shamir и Adelman (RSA) пара ключей
Выполните **команду ssl genrsa** для генерации частного RSA / пара открытых ключей для асимметричного шифрования. CSS хранит генерируемые Открытые и секретные ключи криптосистемы RSA как файл на CSS. Например, для генерации Открытых и секретных ключей криптосистемы RSA myrsakey.pem введите придерживающееся: `CSS11500(config) # ssl genrsa myrsakey.pem 1024 “passwd123” Please be patient this could take a few minutes`
Соединение файла открытых и секретных ключей криптосистемы RSA
Выполните **команду ssl associate rsakey** для соединения названия Открытых и секретных ключей криптосистемы RSA к генерируемым Открытым и секретным ключам криптосистемы RSA. Например, для соединения названия myrsakey1 ключа RSA к генерируемому файлу Открытых и секретных ключей криптосистемы RSA myrsakey.pem введите придерживающееся: `CSS11500(config) # ssl associate rsakey myrsakey1 myrsakey.pem`
Генерируйте CSR
Выполните **команду ssl gencsr rsakey** для генерации файла CSR для связанного файла Открытых и

секретных ключей криптосистемы RSA. Этот CSR будет передаваться CA для подписания. Например, для генерации CSR на основе Открытых и секретных ключей криптосистемы RSA myrsakey1 введите придерживающееся:

```
CSS11503(config)# ssl gencsr myrsakey1 You are about to
be asked to enter information that will be incorporated
into your certificate request. What you are about to
enter is what is called a Distinguished Name or a DN.
For some fields there will be a default value, If you
enter '.', the field will be left blank. Country Name (2
letter code) [US] US State or Province (full name)
[SomeState] CA Locality Name (city) [SomeCity] San Jose
Organization Name (company name) [Acme Inc]Cisco
Systems, Inc. Organizational Unit Name (section) [Web
Administration] Web Admin Common Name (your domain name)
[www.acme.com] www.cisco.com Email address
[webadmin@acme.com] webadmin@cisco.com
```

Команда ssl gencsr генерирует CSR и выводит его на экран. Большинство главных CAs имеет Приложения на основе технологии WWW, которые требуют, чтобы вы вырезали и вставить запрос сертификата на экран.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBWDCCAQICAQAwgZwxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJNQTET
MBEGA1UE
BxMKQm94Ym9yb3VnaDEcMBoGA1UEChMTQ2lzY28gU3lzdGVtcywgSW5j
LjESMBAG
A1UECxMJV2ViIEFkbWluMRYwFAYDVQQDEw13d3cuY2lzY28uY29tMSEw
HwYJKoZI
hvcNAQkBFhJra3JvZWJlckBjaXNjby5jb20wXDANBgkqhkiG9w0BAQEF
AANLADBI
AkEAqHXjtQUVXvmo6tAWPiMpe6oYhZbJUDgTxbW4VMCygzGZn2wUJTgL
rifDB6N3
v+1tKFndE686BhKqfyOidml3wQIDAQABoAAwDQYJKoZIhvcNAQEEBQAD
QQA94yC3
4SUJJ4UQEnO2OqRGLOZpAElc4+IV9aTWK6NmiZsM9Gt0vPhIkLx5jjhV
RLlb27Ak
H6D5omXa0SPJan5x
-----END CERTIFICATE REQUEST-----
```

CA подписывает CSR и возвращает его к вам, как правило, с помощью адреса электронной почты, введенного в CSR.

Получите сертификат из CA

После отправки вашего CSR к CA это берет между одним и семью рабочими днями для получения подписанного сертификата; времена варьируются из-за CA., Как только CA подписал и отправил сертификат, это может быть добавлено к CSS.

Импортируйте цепочечный файл сертификата

Как только CSR был подписан CA, это теперь называют Сертификатом. Файл сертификата должен быть импортирован в CSS. Выполните **команду copy ssl** для упрощения импорта или экспорта сертификатов и секретных ключей от или до CSS. CSS хранит все импортированные файлы в

безопасном месте на CSS. Эта команда доступна только в Режиме суперпользователя. Например, для импорта сертификата mychainedrsacert.pem от удаленного сервера до CSS введите придерживающееся:

```
CSS11500# copy ssl sftp ssl_record import
mychainedrsacert.pem PEM "passwd123" Connecting
Completed successfully
```

Привяжите файл сертификата

Выполните **команду ssl associate cert** для соединения названия сертификата к импортированному сертификату. Например, для соединения названия mychainedrsacert1 сертификата к импортированному mychainedrsacert.pem файла сертификата введите придерживающееся:

```
CSS11500(config)# ssl associate cert mychainedrsacert1
mychainedrsacert.pem
```

Настройте список SSL прокси

Выполните **команду ssl-proxy-list** для создания Списка SSL прокси. Список SSL прокси является группой действительных связанных или SSL - серверы бэкэнда, которые привязаны к сервису SSL. Список SSL прокси содержит все сведения о конфигурации для каждого виртуального сервера SSL. Это включает Создание сервера SSL, сертификаты и соответствующую пару ключей SSL, Виртуальную IP (VIP) адрес и порт, шифры SSL, поддерживаемые, и другие параметры SSL. Например, для создания ssl-proxy-list ssl_list1 введите придерживающееся:

```
CSS11500(config)# ssl-proxy-list ssl_list1 Create ssl-
list <ssl_list1>, [y/n]: y
```

Как только вы создаете Список SSL прокси, CLI вводит вас в режим конфигурации ssl-proxy-list. Настройте свой SSL - сервер как показано ниже.

```
CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20
CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 vip
address 192.168.3.6 CSS11500(ssl-proxy-list[ssl_list1])#
ssl-server 20 rsacert mychainedrsacert1 CSS11500(ssl-
proxy-list[ssl_list1])# ssl-server 20 rsakey myrsakey1
CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20
cipher rsa-export-with-rc4-40-md5 192.168.11.2 80 5
CSS11500(ssl-proxy-list[ssl_list1])# active
```

Настройте сервис протокола SSL и правила содержимого

Как только Список SSL прокси активирован, сервис и правило содержимого должны быть настроены, чтобы позволить CSS передавать трафик SSL к модулю SSL. Эта таблица предоставляет обзор шагов, требуемых создать сервис SSL для виртуального сервера SSL, включая добавление Списка SSL прокси к сервису и созданию правила содержимого SSL.

Создайте сервис SSL

```
CSS11500(config)# service ssl_serv1Create service
<ssl_serv1>, [y/n]: y CSS11500(config-
```

```
service[ssl_serv1])# type ssl-accel CSS11500(config-
service[ssl_serv1])# slot 2 CSS11500(config-
service[ssl_serv1])# keepalive type none
CSS11500(config-service[ssl_serv1])# add ssl-proxy-list
ssl_list1 CSS11500(config-service[ssl_serv1])# active
```

Создайте правило содержимого SSL

```
CSS11500(config)# owner ssl_owner Create owner
<ssl_owner>, [y/n]: y CSS11500(config-owner[ssl_owner])#
content ssl_rule1 Create content <ssl_rule1>, [y/n]: y
CSS11500(config-owner-content[ssl-rule1]# vip address
192.168.3.6 CSS11500(config-owner-content[ssl-rule1]#
port 443 CSS11500(config-owner-content[ssl_rule1])# add
service ssl_serv1 CSS11500(config-owner-
content[ssl_rule1])# active
```

Создайте правило содержимого открытого текста

```
CSS11500(config-owner[ssl_owner])# content decrypted_www Create content
<decrypted_www>, [y/n]: y CSS11500(config-owner-
content[decrypted_www]# vip address 192.168.11.2
CSS11500(config-owner-content[decrypted_www]# port 80
CSS11500(config-owner-content[decrypted_www])# add
service linux_http CSS11500(config-owner-
content[decrypted_www])# add service win2k_http
CSS11500(config-owner-content[decrypted_www])# active
```

На этом этапе клиентский Трафик HTTPS может быть передан CSS в 192.168.3.6:443. CSS дешифрует Трафик HTTPS, преобразовывая его в HTTP. CSS тогда выбирает сервис и передает трафик HTTP к Web-серверу HTTP. Ниже приводится рабочая конфигурация CSS с помощью приведенных выше примеров:

```
CSS11501# show run configure
!************************** GLOBAL
************************** ssl associate rsakey
myrsakey1 myrsakey.pem ssl associate cert
mychainedrsacert1 mychainedrsacert.pem ip route 0.0.0.0
0.0.0.0 192.168.3.1 1 ftp-record conf 192.168.11.101
admin des-password 4f2bxansrcehjgka /tftpboot
!************************ INTERFACE
************************ interface 1/1 bridge vlan 10
description "Client Side" interface 1/2 bridge vlan 20
description "Server Side" !************************
CIRCUIT ************************ circuit VLAN10
description "Client Segment" ip address 192.168.3.254
255.255.255.0 circuit VLAN20 description "Server
Segment" ip address 192.168.11.1 255.255.255.0
!********************** SSL PROXY LIST
********************** ssl-proxy-list ssl_list1 ssl-
server 20 ssl-server 20 vip address 192.168.3.6 ssl-
server 20 rsakey myrsakey1 ssl-server 20 rsacert
mycertcert1 ssl-server 20 cipher rsa-with-rc4-128-md5
192.168.11.2 80 active !************************
SERVICE ************************ service linux-http ip
address 192.168.11.101 port 80 active service win2k-http
ip address 192.168.11.102 port 80 active service
ssl_serv1 type ssl-accel slot 2 keepalive type none add
ssl-proxy-list ssl_list1 active
!************************** OWNER
************************** owner ssl_owner content
ssl_rule1 vip address 192.168.3.6 protocol tcp port 443
add service ssl_serv1 active content decrypted_www vip
address 192.168.11.2 add service linux-http add service
win2k-http protocol tcp port 80 active
```

Проверка

В настоящее время для этой конфигурации нет процедуры проверки.

Устранение неполадок

Для этой конфигурации в настоящее время нет сведений об устранении проблем.

Дополнительные сведения

- Аппаратная поддержка коммутаторов контент-сервисов CSS 11500
- Аппаратная поддержка коммутаторов контент-сервисов CSS 11000
- Загрузка программного обеспечения WEBNS CSS11500 Cisco только для зарегистрированных пользователей)
- Загрузка программного обеспечения WEBNS CSS11000 Cisco (только зарегистрированные клиенты)
- Cisco Systems – техническая поддержка и документация