


Развёртывание и настройка oVirt 4.0. Часть 2.

Замена сертификата веб-сервера oVirt Engine

 blog.it-kb.ru/2016/09/11/install-ovirt-4-0-part-2-setup-custom-web-portal-ssl-certificate-in-apache-web-server-and-websocket-proxy-with-update-java-ca-trusted-store

Автор: Алексей Максимов

11.09.2016



После развёртывания oVirt Engine, при попытке подключения к веб-порталам oVirt мы каждый раз будем получать предупреждение системы безопасности веб-браузера о том, что веб-узел имеет сертификат, которому нет доверия. Это происходит из-за того, что на веб-узле oVirt используется сертификат выданный локальным Центром сертификации (ЦС), который был развёрнут в ходе установки oVirt Engine. Для того, чтобы избавиться от этих предупреждений, а также для того чтобы веб-браузер корректно работал со всеми функциями, доступными на веб-порталах oVirt, нам потребуется сделать так, чтобы веб-браузер доверял SSL сертификату веб-сервера oVirt. Решить этот вопрос можно двумя способами.

Первый и более простой способ – добавить корневой сертификат локального центра сертификации oVirt в хранилище доверенных корневых сертификатов на клиентском компьютере. Согласно документа Console Clients Resources для oVirt 4.0 получить этот корневой сертификат можно непосредственно через веб-браузер, обратившись по ссылке:

```
https://[your engine]/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA
```

Полученный сертификат останется только импортировать на клиентской системе через настройки веб-браузера в хранилище корневых сертификатов доверенных центров сертификации.

Если же в организации есть свой внутренний центр сертификации, к которому уже настроено доверие всех компьютеров, то, в качестве второго способа решения вышеописанной проблемы, можно использовать замену сертификата на веб-сервере oVirt Engine на сертификат, подписанный этим внутренним центром сертификации. В общих чертах процедура замены сертификата описана в параграфе **Replacing oVirt SSL Certificate** в oVirt Administration Guide.

Забегая вперёд, можно сказать, что, в частности, нам нужно будет подменить 3 файла, ссылки на которые есть в конфигурации SSL веб-сервера Apache, на базе которого и работает веб-портал oVirt:

```
# cat /etc/httpd/conf.d/ssl.conf | grep /etc/pki/ovirt-engine/
```

```
SSLCertificateFile /etc/pki/ovirt-engine/certs/apache.cer
SSLCertificateKeyFile /etc/pki/ovirt-engine/keys/apache.key.nopass
SSLCACertificateFile /etc/pki/ovirt-engine/apache-ca.pem
```

Далее все необходимые действия рассмотрим по порядку.

Для процедуры замены сертификата нам нужно будет подготовить 4 файла:

- файл корневого сертификата внутреннего ЦС в формате *.pem (**my-corp-ca.pem**);
- файл сертификата веб-сервера oVirt (Apache) в формате *.pem, выданного внутренним ЦС (**ovirt-apache.pem**);
- файл закрытого ключа от сертификата веб-сервера oVirt (**ovirt-apache.key**);
- файл сертификата веб-сервера в паре с закрытым ключом и с сертификатом корневого ЦС в формате pkcs12 (**ovirt-apache-bundle.p12**)

Кстати, ранее мы уже рассматривали процедуру генерации и установки сертификата на веб-сервер Apache. Здесь процедура будет немного сложнее.

Создаём конфигурационный файл для генерации запроса к корпоративному ЦС:

```
# cat ovirt-apache.cnf

[ req ]
prompt = no
distinguished_name = req_distinguished_name
[ req_distinguished_name ]
commonName = KOM-AD01-OVIRT1.holding.com
countryName = RU
0.organizationName = RoGa and Kopyta Ltd.
localityName = Syktyvkar
organizationalUnitName = Branch KOMI
```

Важно, чтобы атрибут **commonName** в конфигурационном файле соответствовал FQDN имени нашего сервера oVirt Engine.

Генерируем закрытый ключ и запрос к ЦС на основе этого ключа и ранее созданного конфигурационного файла:

```
# openssl genrsa -out ovirt-apache.key 2048
# openssl req -config ovirt-apache.cnf -new -key ovirt-apache.key -out ovirt-apache.req
```

Отправляем файл запроса **ovirt-apache.req** администратору корпоративного ЦС и получаем от него готовый сертификат для веб-сервера (ovirt-apache.cer) + корневой сертификат центра сертификации (my-corp-ca.cer). В моём случае локальный корпоративный ЦС работает на Windows Server CA, поэтому полученные из этого ЦС бинарные сертификаты в кодировке **DER** необходимо будет конвертировать в формат, понятный для Apache – **PEM**:

```
# openssl x509 -in ovirt-apache.cer -inform DER -out ovirt-apache.pem -outform PEM
# openssl x509 -in my-corp-ca.cer -inform d -out my-corp-ca.pem -outform PEM
```

После этого собираем сертификат/закрытый ключ сертификата и корневой сертификат в бандл формата **P12**:

```
# openssl pkcs12 -export -out ovirt-apache-bundle.p12 -inkey ovirt-apache.key -in  
ovirt-apache.pem -chain -CAfile my-corp-ca.pem
```

В процессе создания бандла задаём пароль, с помощью которого бандл будет зашифрован, - используем пароль "mypass"

В итоге у нас во временном каталоге ~/ovirt-certs/ получилось 4 файла **ovirt-apache.key**, **ovirt-apache.pem**, **my-corp-ca.pem** и **ovirt-apache-bundle.p12**. Теперь можно переходить к привязке полученных файлов к oVirt Engine.

Заменяем файл сертификата веб-сервера oVirt используемый в конфигурации **Apache** (параметр **SSLCACertificateFile** в файле /etc/httpd/conf.d/ssl.conf), предварительно сделав копию используемого на данный момент файла. Напомню, что файл **apache-ca.pem** в нашем случае содержит только сертификат веб-сервера:

```
# mv /etc/pki/ovirt-engine/apache-ca.pem /etc/pki/ovirt-engine/apache-ca.pem-BACK  
# cp ~/ovirt-certs/ovirt-apache.pem /etc/pki/ovirt-engine/apache-ca.pem
```

До наших изменений файл /etc/pki/ovirt-engine/apache-ca.pem на самом деле не файл, а символическая ссылка на другой файл /etc/pki/ovirt-engine/ca.pem. Ссылку мы удалять не будем, а, как я уже сказал, просто переименуем, оставив её на случай отката, если что-то у нас пойдёт не так. Сам файл /etc/pki/ovirt-engine/ca.pem при этом не трогаем, так как он нужен для работы внутреннего ЦС oVirt и используется для меж-узлового обмена.

Заменяем бандл **apache.p12**, используемый в данный момент oVirt, предварительно сделав копию уже существующего в системе файла на всякий случай.

```
# mv /etc/pki/ovirt-engine/keys/apache.p12 /etc/pki/ovirt-engine/keys/apache.p12-  
BACK  
# cp ~/ovirt-certs/ovirt-apache-bundle.p12 /etc/pki/ovirt-engine/keys/apache.p12
```

Заменяем файл сертификата веб-сервера oVirt, используемый в конфигурации **Apache** (параметр **SSLCertificateFile** в файле /etc/httpd/conf.d/ssl.conf), опять же предварительно сделав копию используемого на данный момент файла. Второй командой фактически мы выгрузим из нашего обновлённого бандла **apache.p12** сертификат самого веб-сервера вместе с корневым сертификатом нашего ЦС, то есть файл **apache.cer** это тоже своего рода бандл:

```
# mv /etc/pki/ovirt-engine/certs/apache.cer /etc/pki/ovirt-  
engine/certs/apache.cer-BACK  
# openssl pkcs12 -in /etc/pki/ovirt-engine/keys/apache.p12 -nokeys >  
/etc/pki/ovirt-engine/certs/apache.cer
```

В процессе выгрузки будет запрошен пароль. Напомню, что при создании бандла P12 ранее мы использовали пароль "mypass".

Заменяем файл закрытого ключа от сертификата веб-сервера oVirt, используемый в конфигурации Apache (параметр **SSLCertificateKeyFile** в файле `/etc/httpd/conf.d/ssl.conf`), предварительно сделав копию используемого на данный момент файла. Второй командой фактически мы выгрузим из нашего обновлённого бандла **apache.p12** только закрытый ключ:

```
# mv /etc/pki/ovirt-engine/keys/apache.key.nopass /etc/pki/ovirt-engine/keys/apache.key.nopass-BACK
# openssl pkcs12 -in /etc/pki/ovirt-engine/keys/apache.p12 -nocerts -nodes > /etc/pki/ovirt-engine/keys/apache.key.nopass
```

На этом этапе, если перезапустить службу веб-сервера (**service httpd restart**), мы увидим то, что веб-портал oVirt уже использует установленный нами сертификат. Однако это не вся конфигурация, и теперь нам нужно выполнить дополнительную настройку хранилища доверенных сертификатов **java** таким образом, чтобы в нём появился корневой сертификат нашего локального ЦС. В противном случае при попытке входа на портал администрирования oVirt мы словим ошибку:

```
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
```

Решение этой проблемы ранее было описано в [Red Hat Bugzilla - Bug 1336838 - engine doesn't trust externally-issued web certificate for internal authentication in spite of issuer being in system \(and java\) trust store](#) и подтверждено в ветке обсуждения мейл группы oVirt [[ovirt-users](#)] [oVirt 4 with custom SSL-certificate and SPICE HTML5 browser client -> WebSocket error: Can't connect to websocket on URL: wss://ovirt.engine.fqdn:6100/](#). Далее привожу решение этой проблемы для версии **oVirt Engine 4.0 в CentOS Linux 7.2**.

Создаём дополнительный конфигурационный файл, который расширит механизм проверки хранилища доверенных сертификатов в oVirt:

```
# install -o ovirt -g ovirt -m 600 /dev/null /etc/ovirt-engine/engine.conf.d/99-custom-truststore.conf
```

Наполняем файл `99-custom-truststore.conf` содержимым (пароль 'changeit' используется по умолчанию для java truststore):

```
# cat > /etc/ovirt-engine/engine.conf.d/99-custom-truststore.conf << EOF
ENGINE_HTTPS_PKI_TRUST_STORE="/etc/pki/java/cacerts"
ENGINE_HTTPS_PKI_TRUST_STORE_PASSWORD="changeit"
EOF
```

Заданные нами переменные ENGINE_HTTPS_PKI_TRUST_STORE и ENGINE_HTTPS_PKI_TRUST_STORE_PASSWORD будут добавлены к конфигурации PKI для oVirt Engine. В конфигурации по умолчанию эти переменные ссылаются только на внутренний ЦС oVirt.

После скопируем корневой сертификат нашего локального корпоративного ЦС в специальный каталог и выполним обновление хранилища доверенных корневых сертификатов:

```
# cp ~/ovirt-certs/my-corp-ca.pem /etc/pki/ca-trust/source/anchors/  
# update-ca-trust extract
```

Убедимся в том, что java truststore теперь содержит сведения о нашем корневом сертификате локального корпоративного ЦС. Для этого нам может потребоваться отпечаток **SHA1** этого сертификата. Получить его можно командой:

```
# openssl x509 -in /etc/pki/ca-trust/source/anchors/my-corp-ca.pem -fingerprint -  
sha1 -noout
```

```
SHA1 Fingerprint=DE:43:0C:73:A5:8F:85:04:77:A7:64:FB:55:48:7C:D1:59:0F:7B:0A
```

Ну и соответственно запрос к хранилищу может выглядеть так (здесь в параметре -storepass мы используем пароль, ранее заданный в 99-custom-truststore.conf (по умолчанию 'changeit')):

```
# keytool -list -keystore /etc/pki/java/cacerts -storepass changeit | grep "$(  
openssl x509 -in /etc/pki/ca-trust/source/anchors/my-corp-ca.pem -fingerprint -  
sha1 -noout | sed -e '/SHA1/s/.*=//;' )"
```

```
Certificate fingerprint (SHA1):  
DE:43:0C:73:A5:8F:85:04:77:A7:64:FB:55:48:7C:D1:59:0F:7B:0A
```

Как видим, наш корневой сертификат локального ЦС присутствует в хранилище java truststore.

Дополнительно можно проверить то, что сертификат также присутствует в подкаталогах /etc/pki/ca-trust/extracted/:

```
# grep -IR "$(sed -n '2p' /etc/pki/ca-trust/source/anchors/my-corp-ca.pem)"  
/etc/pki/ca-trust/extracted/
```

```
/etc/pki/ca-trust/extracted/pem/email-ca-  
bundle.pem:MIIERzCCAy+gIBAgIQb63kBz...BgkiG9w0BAQsFADBh  
/etc/pki/ca-trust/extracted/pem/objsign-ca-  
bundle.pem:MIIERzCCAy+gIBAgIQb63kBz...BgkiG9w0BAQsFADBh  
/etc/pki/ca-trust/extracted/pem/tls-ca-  
bundle.pem:MIIERzCCAy+gIBAgIQb63kBz...BgkiG9w0BAQsFADBh  
/etc/pki/ca-trust/extracted/openssl/ca-  
bundle.trust.crt:MIIERzCCAy+gIBAgIQb63kBz...BgkiG9w0BAQsFADBh
```

Перезапускаем службы oVirt Engine и проверяем результат.

```
# service ovirt-engine restart
```

Теперь веб-консоль порталов oVirt 4.0 должна открываться без предупреждений и ошибок.

Помимо вышеуказанных действий потребуется дополнительная конфигурация такой компоненты oVirt Engine, как **WebSocket Proxy**, без которой, в частности, не будет работать альтернативный метод подключения к консоли виртуальных машин с помощью **HTML5 SPICE Web browser client**. Приведём файл 10-setup.conf к следующему виду:

```
# cat /etc/ovirt-engine/ovirt-websocket-proxy.conf.d/10-setup.conf
```

```
PROXY_PORT=6100
```

```
SSL_CERTIFICATE=/etc/pki/ovirt-engine/apache-ca.pem
```

```
SSL_KEY=/etc/pki/ovirt-engine/keys/apache.key.nopass
```

```
CERT_FOR_DATA_VERIFICATION=/etc/pki/ovirt-engine/certs/engine.cer
```

```
SSL_ONLY=True
```

После чего перезапустим службу WebSocket Proxy:

```
# service ovirt-websocket-proxy restart
```

На этом пока всё. В следующей записи мы рассмотрим базовые настройки oVirt.