

Оглавление

1	Создание центра сертификации	•
2	Добавление сертификата	8

1. Создание центра сертификации

Для создания локального центра сертификации в $\operatorname{PEД}$ OC 7.3 необходимо выполнить следующие действия:

1. Создать закрытый ключ Root:

```
# openssl genrsa -des3 -out root.key 2048
```

```
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)
Enter pass phrase for root.key: *******
Verifying - Enter pass phrase for root.key: *******
```

Важно! Рекомендуется указать парольную фразу и защитить закрытый ключ.

2. Сгенерировать корневой сертификат. В процессе выполнения команды будет предложено ввести указанную на предыдущем шаге парольную фразу. После этого потребуется ввести некоторые данные для запроса сертификата - страну, область, город или другой населенный пункт, наименование организации, наименование подразделения организации и имя сертификата.

```
# openssl req -x509 -new -nodes -key root.key -sha256 -days 7200 -out
root.pem
```

где:

- -х509 экземпляр сертификата;
- -new новый запрос сертификата;
- -nodes отключить шифрование выходного ключа;
- -key root.key файл ключа;
- -sha256 алгоритм подписи;
- -days 7200 период действия сертификата (в днях);
- -out root.pem имя сгенерированного сертификата.

```
Enter pass phrase for root.key: ******
  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.
  There are quite a few fields but you can leave some blank
  For some fields there will be a default value,
  If you enter '.', the field will be left blank.
  Country Name (2 letter code) [XX]:ru
  State or Province Name (full name) []:moscow
  Locality Name (eg, city) [Default City]:moscow
  Organization Name (eg, company) [Default Company Ltd]:redsoft
  Organizational Unit Name (eg, section) []:drsp
  Common Name (eg, your name or your server's hostname) []:Private RED
Virt Authority
  Email Address []:
```

3. Проверить содержание сгенерированного сертификата, выполнив команду:

```
# openssl x509 -text -noout -in root.pem | head -15
```

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
60:05:f3:81:4d:b9:33:bd:a6:d7:34:9b:bb:31:40:d3:c3:8a:1d:86
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = ru, ST = moscow, L = moscow, O = redsoft,
OU = drsp, CN = Private RED Virt Authority
Validity
Not Before: Sep 6 13:29:10 2023 GMT
Not After: May 24 13:29:10 2043 GMT
Subject: C = ru, ST = moscow, L = moscow, O = redsoft,
OU = drsp, CN = Private RED Virt Authority
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
```

```
RSA Public-Key: (2048 bit)
Modulus:
```

4. Убедиться в том, что был создан именно центр сертификации, выполнив команду:

```
# openssl x509 -text -noout -in root.pem | grep CA:
```

```
CA:TRUE
```

5. Для использования созданного сертификата в качестве локального центра сертификации необходимо импортировать его на все доступные устройства. Корневой сертификат можно добавить в хранилище ключей/сертификатов ОС либо загрузить напрямую в браузер.

Существуют различные типы сертификатов (OV, EV, Wildcard и т. д.) и иерархии полномочий (Root, Intermediate, Sub CA и т. д.). В рамках приведенной инструкции будет рассмотрен вариант иерархии Root Authority с выпуском сертификата типа Wildcard, что позволит иметь один SSL-сертификат для всех внутренних доменов redvirt.home. Сертификат типа Wildcard может быть применен к домену и всем его поддоменам. Для генерации самоподписанного группового сертификата, необходимо создать файл с расширением csr и закрытый ключ.

Для создания закрытого ключа необходимо выполнить:

```
# openssl genrsa -out wildcard.redvirt.home.key 2048
```

```
Generating RSA private key, 2048 bit long modulus (2 primes)
......+++++
e is 65537 (0x010001)
```

6. Для создания запроса самоподписанного сертификата следует использовать файл конфигурации, содержимое которого приведено ниже.

B разделе [alt_names] необходимо определить расширение Subject Alternative Name (SAN).

```
# nano opensslsan.cnf
```

```
[req]
distinguished_name=req_distinguished_name
req_extensions=v3_req
prompt=no

[req_distinguished_name]
C=ru
ST=moscow
L=moscow
```

```
O=redsoft
OU=drsp
CN=*.redvirt.home

[v3_req]
keyUsage=keyEncipherment, dataEncipherment, digitalSignature
extendedKeyUsage=serverAuth
subjectAltName=@alt_names

[alt_names]
DNS.1 = *.redvirt.home
```

7. Сгенерировать корневой сертификат wildcard.redvirt.home.csr с помощью созданного файла конфигурации:

```
# openssl req -new -out wildcard.redvirt.home.csr -key wildcard.
redvirt.home.key -config opensslsan.cnf
```

8. Далее необходимо подписать файл с расширением **csr** собственным закрытым ключом.

```
# openssl x509 -req -in wildcard.redvirt.home.csr -CA root.pem -CAkey
root.key -CAcreateserial -out wildcard.redvirt.home.crt -days 7200
-sha256 -extensions v3_req -extfile opensslsan.cnf
```

```
Signature ok
subject=C = ru, ST = moscow, L = moscow, O = redsoft, OU = drsp, CN =
*.redvirt.home
Getting CA Private Key
Enter pass phrase for root.key:
```

- 9. Ввести пароль корневого закрытого ключа.
- 10. Проверить, что сертификат действителен и цепочка доверена.

```
# openssl verify -CAfile root.pem wildcard.redvirt.home.crt
```

```
wildcard.redvirt.home.crt: OK
```

11. Проверить содержимое сертификата Wildcard, выполнив команду:

```
# openssl x509 -text -noout -in wildcard.redvirt.home.crt | head -15
```

```
Certificate:
    Data:
    Version: 3 (0x2)
    Serial Number:
    64:27:99:f3:81:3e:b3:ae:df:4c:35:78:b1:e6:0f:87:6e:01:eb:a6
```

```
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = ru, ST = moscow, L = moscow, O = redsoft, OU = drsp,
CN = Private RED Virt Authority
Validity
Not Before: Sep 7 08:14:35 2023 GMT
Not After: May 25 08:14:35 2043 GMT
Subject: C = ru, ST = moscow, L = moscow, O = redsoft, OU = drsp, CN = *.redvirt.home
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public-Key: (2048 bit)
Modulus:
```

```
# openssl x509 -text -noout -in wildcard.redvirt.home.crt | grep DNS
```

```
DNS:*.redvirt.home
```

Сертификат выдан и будет действителен в течение следующих 20 лет. Все файлы, включая wildcard.redvirt.home.crt (сертификат), wildcard.redvirt.home.key (закрытый ключ) и root.pem (сертификат ЦС), будут использоваться для настройки SSL на любом из веб-серверов.

```
# 11
-rw-r--r-. 1 root root 325 ceh 6 16:39 opensslsan.cnf
-rw-----. 1 root root 1743 ceh 6 16:27 root.key
-rw-r--r-. 1 root root 1375 ceh 6 16:29 root.pem
-rw-r--r-. 1 root root 41 ceh 7 11:14 root.srl
-rw-r--r-. 1 root root 1334 ceh 7 11:14 wildcard.redvirt.home.crt
-rw-r--r-. 1 root root 1110 ceh 7 11:14 wildcard.redvirt.home.csr
-rw-----. 1 root root 1679 ceh 6 16:39 wildcard.redvirt.home.key
```

2. Добавление сертификата

Все последующие действия должны производиться на развернутом Engine РЕД Виртуализации.

Для добавления сертификата в систему РЕД Виртуализации необходимо:

- 1. Скопировать с созданного panee центра сертификации файлы wildcard.redvirt.home.crt, wildcard.redvirt.home.key, root.pem на хост РЕД Виртуализации в папку /tmp.
 - 2. Создать файл с расширением .p12 (в примере apache.p12):

```
\hbox{\tt\# openssl pkcs12-export-out apache.p12-inkey wildcard.redvirt.home.}\\ key -in wildcard.redvirt.home.crt
```

Пароль указывать не нужно.

3. Экспортировать ключ из созданного на предыдущем шаге файла с расширением .p12 (в примере apache.p12):

```
# openssl pkcs12 -in apache.p12 -nocerts -nodes > apache.key
```

Пароль указывать не нужно.

4. Экспортировать файл с расширением сет из файла apache.p12:

```
# openssl pkcs12 -in apache.p12 -nokeys > apache.cer
```

Теперь с точки зрения самоподписанного SSL-сертификата в системе РЕД Виртуализации существуют все необходимые файлы:

- /tmp/root.pem;
- /tmp/apache.p12;
- /tmp/apache.key;
- /tmp/apache.cer.

- 5. Затем создать резервную копию текущего файла apache.p12:
- # cp -p /etc/pki/ovirt-engine/keys/apache.p12 /tmp/apache.p12.bck
- 6. Заменить текущий файл арасhe.p12 на созданный в п. 2 файл:
- # cp /tmp/apache.p12 /etc/pki/ovirt-engine/keys/apache.p12
- 7. Заменить имеющийся ЦС на созданный в п. 2 раздела 1 «Создание центра сертификации» корневой сертификат и обновить хранилище доверенных сертификатов:
 - # cp /tmp/root.pem /etc/pki/ca-trust/source/anchors
 # update-ca-trust
- 8. Удалить символическую ссылку и сохранить сертификат как apache-ca.pem в соответствующий каталог:
 - # rm /etc/pki/ovirt-engine/apache-ca.pem
 # cp /tmp/root.pem /etc/pki/ovirt-engine/apache-ca.pem
 - 9. Создать резервную копию существующего закрытого ключа и сертификата:
- $\label{lem:condition} \mbox{$\#$ cp /etc/pki/ovirt-engine/keys/apache.key.nopass.bck} \\ \mbox{$\#$ cp /etc/pki/ovirt-engine/keys/apache.key.nopass.bck}$
- # cp /etc/pki/ovirt-engine/certs/apache.cer /etc/pki/ovirtengine/certs/apache.cer.bck
- 10. Скопировать выданный закрытый ключ и сертификат в соответствующие каталоги:
 - # cp /tmp/apache.key /etc/pki/ovirt-engine/keys/apache.key.nopass
 # cp /tmp/apache.cer /etc/pki/ovirt-engine/certs/apache.cer
 - 11. Перезапустить сервер арасће:
 - # systemctl restart httpd.service
- 12. Создать новый файл конфигурации доверенного хранилища со следующим содержимым:
 - # nano /etc/ovirt-engine/engine.conf.d/99-custom-truststore.conf

```
ENGINE_HTTPS_PKI_TRUST_STORE="/etc/pki/java/cacerts"
ENGINE_HTTPS_PKI_TRUST_STORE_PASSWORD=""
```

- 13. Сохранить файл.
- 14. Отредактировать файл /etc/ovirt-engine/ovirt-websocket-proxy.conf.d/ 10-setup.conf:

nano /etc/ovirt-engine/ovirt-websocket-proxy.conf.d/10-setup.conf

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

15. Переопределить конфигурацию ovirt-engine /etc/ovirt-imageio/conf.d/50 -engine.conf:

```
 \begin{tabular}{ll} \# cp /etc/ovirt-imageio/conf.d/50-engine.conf /etc/ovirt-imageio/conf.d/99-local.conf \end{tabular}
```

16. Перезапустить необходимые службы:

```
$ systemctl restart ovirt-provider-ovn.service
$ systemctl restart ovirt-imageio
$ systemctl restart ovirt-websocket-proxy
$ systemctl restart ovirt-engine.service
```

17. Скопировать и установить сертификат root.pem в качестве корневого на машине, где будет производиться администрирование системы РЕД Виртуализация.

Если все настроено верно, подключение к порталу администратора и порталу виртуальных машин будет производиться без вывода предупреждений о подлинности сертификата, используемого для шифрования HTTPS-трафика.