Сертификат для Linux в центре сертификации Active Directory

Если в сети есть домен с контроллером и центр сертификации Microsoft, для внутреннего использования можно создавать собственные сертификаты, для которых браузер не будет выдавать ошибку. Если с запросом и установкой сертификата на Windows возникает меньше вопросов, то получение сертификата для Linux выполнить немного сложнее.

В этой статье мы создадим правильный сертификат с указанием альтернативного DNS-имени, без которого программы могут возвращать ошибку сертификата.

1. Формируем файл запроса (на компьютере с Linux)

Для начала создаем каталог, в котором планируем хранить сертификаты и перейдем в него, например:

mkdir -p /etc/ssl/adcs

cd /etc/ssl/adcs

Создаем закрытый ключ:

openssl genrsa -out private.key 2048

* *в данном примере создан **2048**-и битный ключ с именем **private.key***

Создаем файл с описанием запроса:

vi openssl.conf

[req]
default_bits = 2048

```
prompt = no
default_md = sha256
req_extensions = req_extensions
distinguished_name = dn

[dn]
C=RU
ST=SPb
L=SPb
O=Global Security
OU=IT Department
emailAddress=hostmaster@dmosk.ru
CN = *.dmosk.local

[req_extensions]
subjectAltName = @alter_name

[alter_name]
DNS.1 = *.dmosk.local
```

** где стоит обратить внимание на следующее:*

- ***sha256 (или SHA-2)** — алгоритм шифрования;*
- ****.dmosk.local** — имя узла, к которому мы будем обращаться. Это очень важный пункт — нужно, чтобы имя совпадало с именем, по которому будут выполняться обращения. В данном примере создается wildcart (домен и все его поддомены)*
- **subjectAltName** — альтернативные имена стали иметь значение с 2017 года, когда некоторые браузеры перестали принимать сертификаты без указания альтернативных DNS-имен.

Далее создаем ключ запроса сертификата:

```
openssl req -new -key private.key -out request.csr -config openssl.conf
```

** где **private.key** — закрытый ключ, который был сформирован на предыдущем шаге;*
***request.csr** — файл, который будет сформирован. В нем будет запрос на открытый ключ;*

После выполнения команды, в каталоге появится файл request.csr. Выведем его содержимое, чтобы посмотреть запрос:

```
cat request.csr
```

Должны увидеть что-то на подобие:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC1jCCAb4CAQAwgZAxCzAJBgNVBAYTAlJVMQwwCgYDVQQIDANTUGIxDDAKBgNV
BAcMA1NQYjEYMBYGA1UECgwPR2xvYmFsIFNlY3VyaXR5MRYwFAYDVQQLDA1JVCBE
```

ZXBhcnRtZW50MR4wHAYDVQQDDBVzaW1wbGVwbGFuLnNhdHMubG9jYWwxEzARBgNV
BAMMCnNpbXBsZXBsYW4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZ
ze/WOQgnlbfzlEsQUsmfUTnkYq0rCpzgjY360lFvqek5Y8NIFX/25PRbUy4N3D8r
c/7mXt2dXmcnn7zeRQOB2g0AY8Wmeg3R6C+JH7TwxtkMj7FO8R59URxFN84lu9Sj
26Aw+Ax7474XnAoUBMSmUXbV2mAP5Xm83sjvjE1OcHXN8SPbc+EchZuLVLsIGXHz
Emz7V4D/ecahfSc2hCRG2Pc7SeFIADYdjyoLtykz5WyiIoXkpEfSNQHlt2/A1kJ5
h9/GPXMVJVL02FgsI5HlGZyGnYWA+cP7sHoEDZNpLHHuEtfwx3bLxJPFnZDa0rPO
LhV/ux1e6b9ikrge0xp9AgMBAAGgADANBgkqhkiG9w0BAQsFAAOCAQEAPVD2aVH8
ZDz+6s8ecKr08eT3mA9cRCa7dZYFj4/vO/ZYrDH9QS45gd0sYG+RN+JMGaFaY+X7
faE4m0BysPIbhEYLRY5GJYxmOGm05gM2HfPrcnnWXZbPQu/n5pR4ptvPYL7bilGb
z6hXb8ZtXUXwz1F2OgTPOPu4+w8lI23pzHRHlCXcVSoZxe/A2XwusB5MrtyMEYtj
rB2kcOqQRkt9uIv5IobwnYaVGDk/7wl/zkb9K+RKZt4izKfFaSSyPn7wvpKSIaAf
S3SQjJ0tBckLbtwKrxdFB0B8bpyIKUtHmpX/zOmityC8PLXe6/vQ/DmM3B6QC4Ba
KdnRkOSPv8BGog==
-----END CERTIFICATE REQUEST-----

Чтобы проверить содержимое запроса, вводим:

openssl req -noout -text -in request.csr

Мы увидем что-то на подобие:

Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = RU, ST = SPb, L = SPb, O = Global Security, OU = IT Department, emailAddress = hostmaster@dmosk.ru, CN = *.dmosk.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:e0:dd:17:1c:e1:24:25:80:fe:a0:b3:39:95:bf:
                    9f:3b:a6:3f:62:c6:fa:40:19:0e:11:fe:9b:22:5b:
                    0c:67:57:08:b6:f4:36:4c:9c:aa:de:0b:ee:f2:11:
                    35:fc:18:36:ed:f5:e0:f9:82:98:bf:a5:1a:6f:a8:
                    13:20:2e:96:2a:68:13:db:e6:8e:0a:ce:fc:ee:c1:
                    fb:35:55:63:3f:bd:be:21:e7:f2:f3:fc:d3:b0:fd:
                    04:84:ce:84:29:2a:66:a9:ee:53:0e:f1:06:a2:b7:
                    0d:1c:8a:b9:e3:5d:ca:55:c3:30:77:48:d3:eb:d4:
                    27:11:8f:28:33:da:d4:4d:05:b6:da:f7:ea:84:3b:
                    8f:97:7d:26:f5:6f:09:39:8a:79:fc:6e:d2:3a:db:
                    4e:e8:67:2a:a7:22:00:a0:6c:c1:99:50:ea:3c:f7:
                    27:33:d3:5f:02:ed:7e:9e:66:17:fc:f3:af:f8:ef:
                    36:9e:da:3d:6c:c7:d3:b3:8b:4b:45:12:2f:84:34:
                    02:a5:0b:41:a1:6a:a3:91:6c:b0:87:d4:5b:cc:cc:

```
                f2:6c:17:4c:8b:46:40:65:5d:ca:8c:c0:f7:af:14:
                ad:58:1e:80:f5:67:16:1e:de:da:a5:7b:05:a6:f8:
                08:75:4b:8c:fe:53:e6:c8:50:00:30:ff:47:a5:49:
                47:c3
            Exponent: 65537 (0x10001)
    Attributes:
    Requested Extensions:
        X509v3 Subject Alternative Name:
            DNS:*.dmosk.local
Signature Algorithm: sha256WithRSAEncryption
    02:03:bc:8b:d3:4e:8f:2c:b2:a2:39:dd:9b:fb:33:a9:c1:49:
    95:b7:39:11:51:b2:40:dc:57:6c:3e:d5:66:6d:16:57:3f:15:
    03:cf:9b:39:95:1f:2b:13:09:86:f8:d5:69:b0:ac:cd:dc:64:
    9f:98:b7:f4:75:79:2a:16:d6:42:15:f4:1a:28:f3:3b:e3:f5:
    ec:34:98:f5:18:3e:72:95:ed:93:e5:e4:62:b3:3e:ce:71:09:
    9d:7f:ec:84:10:59:43:42:cd:0d:bd:4e:fd:c3:3e:44:ab:73:
    e0:ae:5d:67:c7:74:f0:30:0c:29:55:dc:16:4b:5f:a5:7f:be:
    8b:ec:64:3b:63:91:bb:48:d2:64:e5:8f:c0:09:db:7b:a3:10:
    8e:69:da:f7:88:00:a2:5a:60:dd:d9:3c:ef:b9:59:f4:b5:ab:
    00:d9:f9:31:6b:c1:c6:b1:ea:77:71:be:63:2c:e7:92:57:89:
    2a:5e:83:a9:e1:1c:56:c2:60:62:73:5d:bc:27:83:fb:bc:ca:
    9a:75:52:58:d6:02:d3:e8:37:b1:28:0e:89:62:97:9f:c4:f1:
    52:60:ed:95:59:2b:cd:69:29:9d:c2:30:1e:56:12:03:f4:17:
    3f:57:a2:b6:e7:b0:47:4e:0e:5a:b8:5e:0c:05:45:76:c5:49:
    f0:42:94:fa
```

2. Выпускаем сертификат

Копируем содержимое файла запроса (request.csr) и открываем веб-страницу центра сертификации — как правило, это имя сервера или IP-адрес + certsrv, например, http://192.168.0.15/certsrv/.

В открывшемся окне переходим по ссылкам **Запроса сертификата** - **расширенный запрос сертификата**.

В поле «Сохраненный запрос» вставляем скопированный запрос, в а поле «Шаблон сертификата» выбираем **Веб-сервер**:

Вводим данные для получения сертификата

Нажимаем **Выдать** и скачиваем сертификат по ссылке **Загрузить сертификат**:

Загрузка готового сертификата

3. Установка сертификата на Linux

Переносим полученный сертификат на компьютер с Linux.

После необходимо сконвертировать DER-сертификат (от AD CS) в PEM-сертификат (для Linux). Для этого вводим следующую команду:

openssl x509 -inform der -in certnew.cer -out public.pem

** где **certnew.cer** — файл, который мы получили от центра сертификации AD CS (как правило, у него такое имя); **public.pem** — имя PEM-сертификата.*

Сертификат готов к использованию. Мы сформировали:

1. private.key — закрытый ключ.
2. public.pem — открытый ключ.

Revision #3
Created 2025-02-12 14:45:20 UTC by yuzer
Updated 2025-02-12 14:47:10 UTC by yuzer