

??????? ?????????????? (??? - ?????????????) KUMA

Процесс перевыпуска сертификата ядра в версии KUMA 3.2 был изменен! Актуальная информация приведена в онлайн-справке: <https://support.kaspersky.ru/help/KUMA/3.2/ru-RU/275543.htm> и <https://support.kaspersky.ru/kuma/3.2/217747>

????? ??????????????

После установки Ядра KUMA установщик создает следующие сертификаты в папке /opt/kaspersky/kuma/core/certificates:

- Самоподписанный корневой сертификат ca.cert с ключом ca.key. Подписывает все другие сертификаты, которые используются для внутренней связи между компонентами KUMA.
- Сертификат internal.cert, подписанный корневым сертификатом, и ключ internal.key сервера Ядра. Используется для внутренней связи между компонентами KUMA.
- Сертификат веб-консоли KUMA external.cert и ключ external.key. Используется в веб-консоли KUMA и для запросов REST API.

Между взаимодействием компонентов KUMA не должно быть SSL-инспекции

??????? ?????????????? ?? ??????? ????????????????? CA

Создаем файл `openssl.cnf`:

```
====это пример, вписываем свои значения====  
[ req ]  
default_bits = 4096 # RSA key size  
encrypt_key = no # Protect private key  
default_md = sha256 # MD to use  
utf8 = yes # Input is UTF-8
```

```
string_mask = utf8only # Emit UTF-8 strings
prompt = no # Prompt for DN
distinguished_name = server_dn # DN template
req_extensions = server_reqext # Desired extensions
[ server_dn ]
countryName = RU # ISO 3166
localityName = Moscow
organizationName = MYCOMPANY
organizationalUnitName = SOC
commonName = kuma.mycompany.ru # Should match a SAN under alt_names

[ server_reqext ]
basicConstraints = CA:FALSE
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth,clientAuth
subjectKeyIdentifier = hash
subjectAltName = @alt_names
[alt_names]
DNS.1 = kuma.mycompany.ru
IP.1 = 1.2.3.4
```

Делаем запрос на сертификат:

```
openssl req -new -nodes -sha256 -out external.csr -config /root/cert/openssl.cnf -keyout
external.key
```

В текущем каталоге будут созданы файлы external.key (файл с закрытым ключом) и external.csr (файл запроса на сертификат)

Отдаем запрос на сертификат в центр сертификации. От центра сертификации должны получить сертификат сервера и корневой сертификат центра сертификации и все сертификаты промежуточных центров сертификации если они есть.

- Все сертификаты должны быть в формате PEM (BASE64).
- Содержимое сертификатов должно выглядеть - (-----BEGIN CERTIFICATE----- и -----END CERTIFICATE-----)

Полученные сертификаты необходимо объединить в одну полную цепочку сертификатов в правильном порядке.

```
openssl x509 -inform PEM -in server.pem > external.cert
openssl x509 -inform PEM -in subca.pem >> external.cert
```

```
openssl x509 -inform PEM -in root.pem >> external.cert
```

В файле с сертификатами важен порядок и должен быть примерно такой: rootca-subca1-subca2-subca3-...

Полученный файл external.cert необходимо проверить на корректность (должен быть вывод списка цепочки сертификатов):

```
openssl crl2pkcs7 -nocrl -certfile external.cert | openssl pkcs7 -print_certs -noout  
  
openssl verify external.cert  
#external.cert: OK
```

Проверяем корректность полученного сертификата и ключа (результат должен совпадать):

```
openssl x509 -noout -modulus -in external.cert | openssl md5  
#(stdin)= 239994c3bd2a90507a548bf373127e18  
openssl rsa -noout -modulus -in external.key | openssl md5  
#(stdin)= 239994c3bd2a90507a548bf373127e18
```

?????? ? ????????? ????????????? (?? ?????????
?????)

Перед заменой пары external.cert external.key создайте их резервную копию.

Взять сертификат компании (вероятно это будет pfx), но может и другие форматы - например DER.

Сконвертировать его в ключ и сертификат в формате PEM (BASE64) и положить в папку например, для конвертации из PFX это будут команды ниже.

Получение ключа из kumaWebIssuedByCorporateCA.pfx:

```
openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nocerts -out external.key
```

Получение сертификата из kumaWebIssuedByCorporateCA.pfx:

```
openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nokeys -out external.cert
```

(Опционально) если ключ версии PKCS#1 то его нужно сконвертировать в PKCS#8, это можно сделать соединяющей командой:

Revision #8

Created 2023-09-06 13:21:44 UTC by Boris RZR

Updated 2025-02-25 10:40:00 UTC by Boris RZR