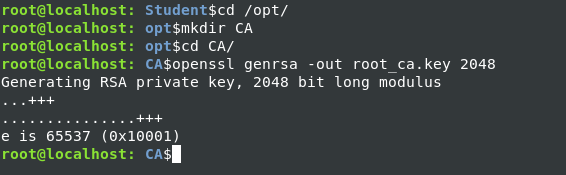
**Создание CA на базе OpenSSL для установки mTLS в браузере и в любых запросах.**

*На основе статьи https://habr.com/ru/articles/671730/*



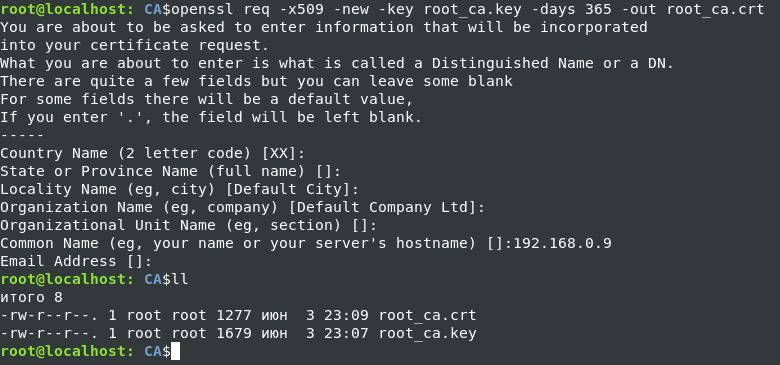
Генерируем ключ для сервера:

openssl genrsa -out root\_ca.key 2048



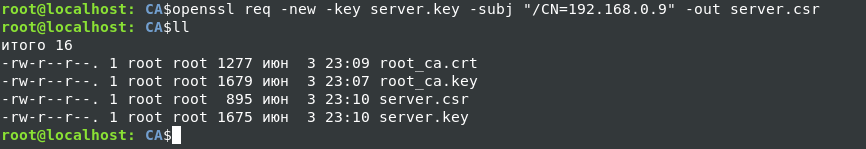
Получаем самоподписанный сертификат:

openssl req -x509 -new -key root\_ca.key -days 365 -out root\_ca.crt



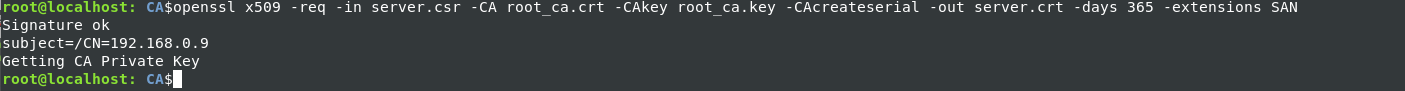
Генерируем приватный ключ для HTTPS сервера

openssl genrsa -out server.key 2048

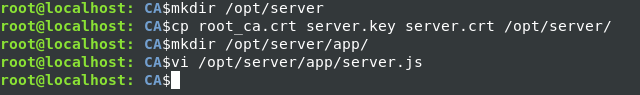


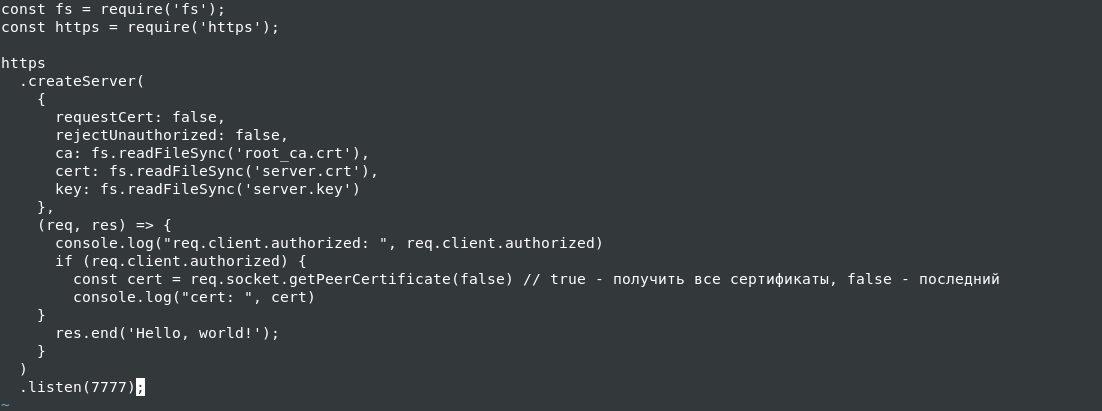
Генерируем сертификат X.509 (server.crt) на 365 дней для HTTPS сервера и подписываем его приватным ключом CA

openssl x509 -req -in server.csr -CA root\_ca.crt -CAkey root\_ca.key -CAcreateserial -out server.crt -days 365 -extensions SAN

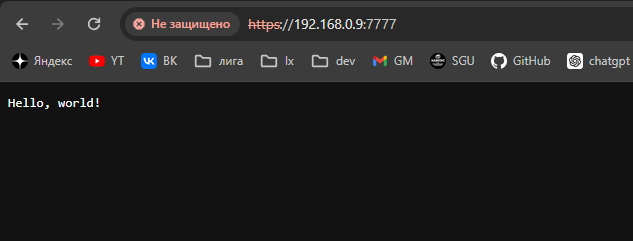


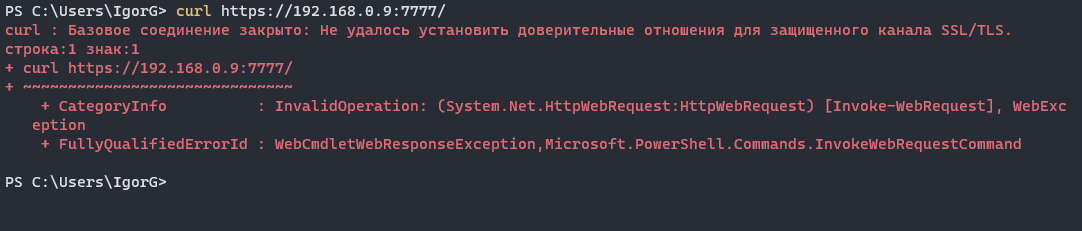
Копируем root\_ca.crt, server.key, server.crt на сервер с приложением (это эта же ВМ), и пишем небольшой сервер, работающий на https (и не забываем открыть соответствующий порт):



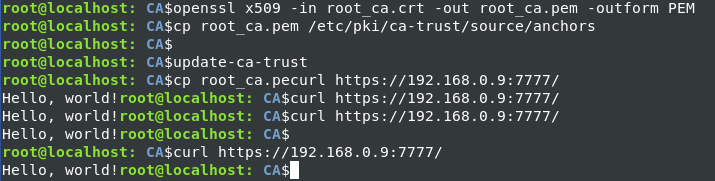


Можем получить доступ к серверу, но видим предупреждения о проблемах с сертификатом:





Это значит, что самоподписанный корневой сертификат нашего CA (root\_ca.crt) не находится в хранилище доверительных сертификатов. Процесс загрузки сертификатов в это хранилище специфичен для операционной системы. Пример для CentOS7

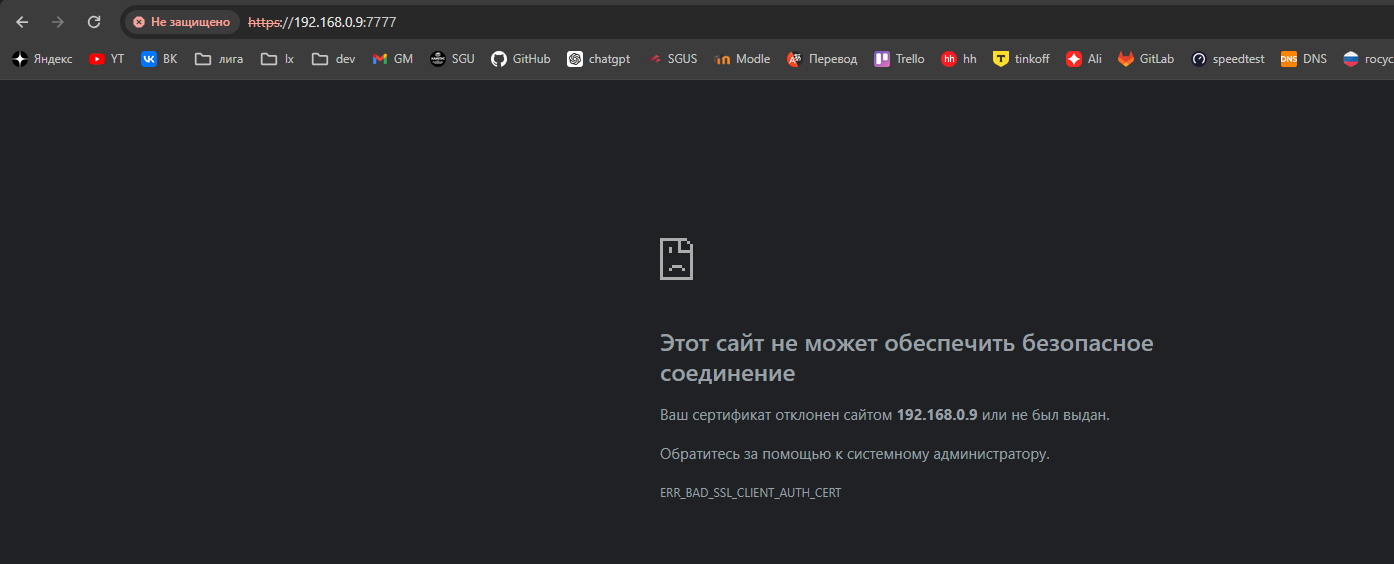


Реализация mTLS

В коде сервера включаем mTLS, а именно выставляем в true параметр requestCert, чтобы HTTPS сервер запрашивал клиентский сертификат и параметр rejectUnauthorized, чтобы сервер отклонял запросы клиентов, которые не прошли авторизацию:

{  
 requestCert: true,  
 rejectUnauthorized: true,  
}

Теперь клиент без ключа не может попасть на сайт:



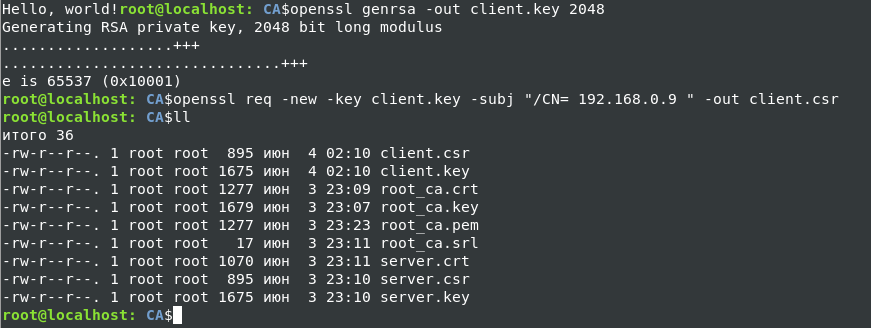
Алгоритм генерации клиентских сертификатов аналогичен серверным.

Генерируем приватный ключ для клиента:

openssl genrsa -out client.key 2048

Генерируем запрос на сертификат CSR со списком CN:

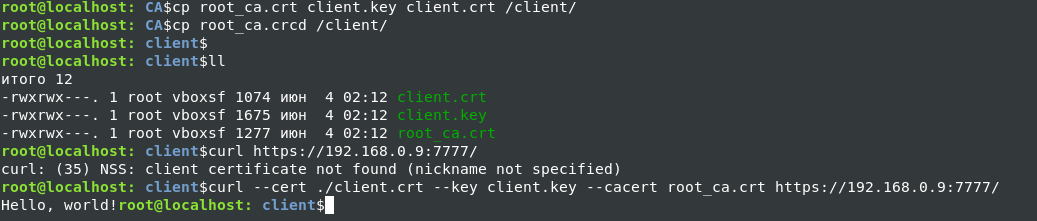
openssl req -new -key client.key -subj "/CN= 192.168.0.9 " -out client.csr



Генерируем сертификат для клиента и подписываем его приватным ключом CA:

openssl x509 -req -in client.csr -CA root\_ca.crt -CAkey root\_ca.key -CAcreateserial -out client.crt -days 365 -extensions SAN -extfile openssl.cnf

Копируем следующие файлы на компьютер, на котором будет работать клиент: root\_ca.crt, client.key, client.crt

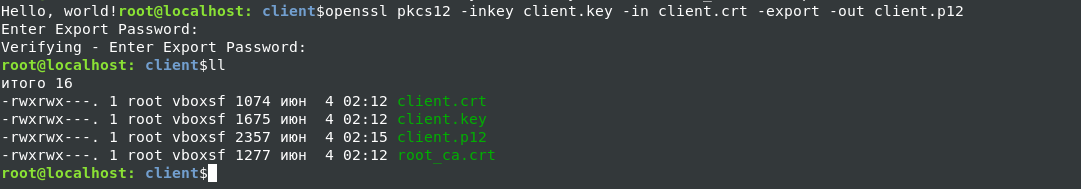


Это значит, что мы установили mTLS соединение с проверкой сертификата как сервера, так и клиента.

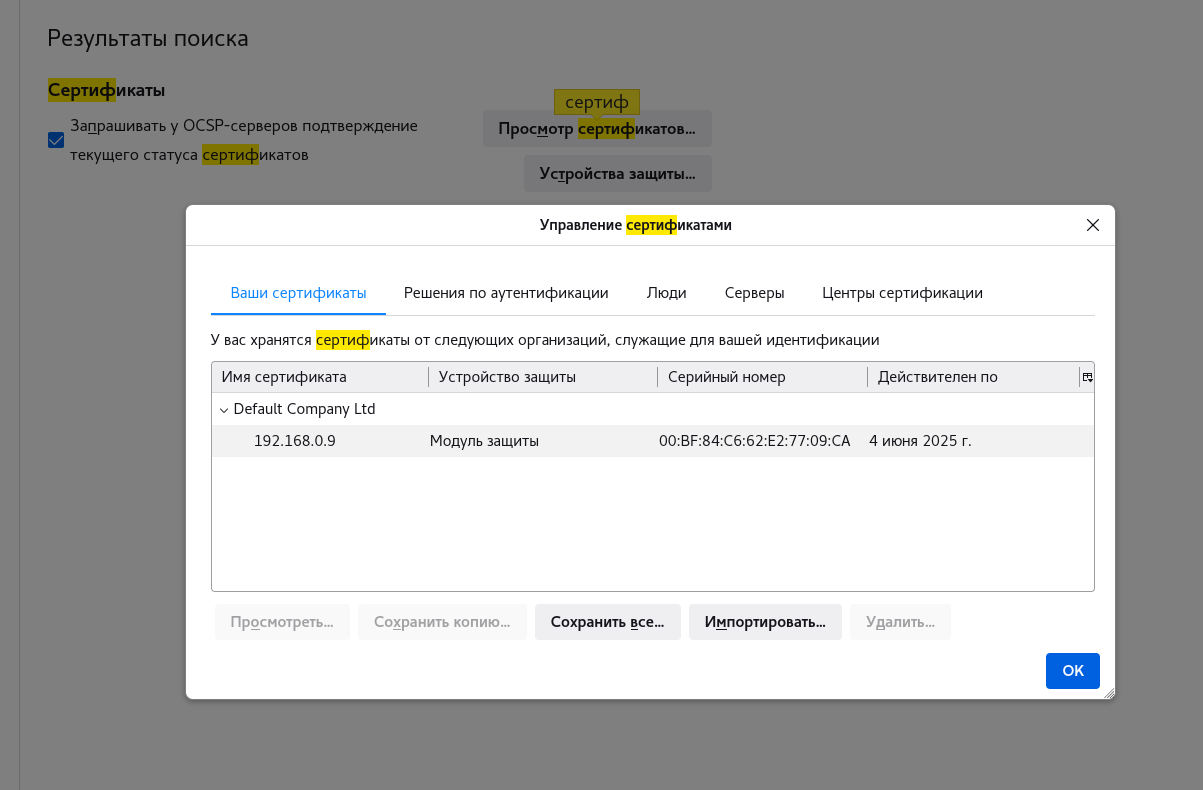
Реализация mTLS в браузере. Чтобы открыть сайт в браузере, нужно:

Клиентский сертификат и приватный ключ надо сконвертировать в формат p12.

openssl pkcs12 -inkey client.key -in client.crt -export -out client.p12



Импортируем полученный сертификат в браузер:



Сайт доступен по httpsбез предупреждений, с действительным сертификатом:

