Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РтФ

Школа профессионального и академического образования

Отчет по дисциплине   
«Криптографические методы защиты информации»

Лабораторная работа №3  
«*HTTPS*: как работает, для чего нужна цепочка шифрования»

Студенты: Клоченко И.Е.

Преподаватель: Соколов И.П.

Группа: РИ-300024

Екатеринбург

2023

**Оглавление**

[1. Организация HTTPS в NGINX путем создания самоподписанного SSL сертификата 3](#_Toc133856995)

[2. Организация HTTPS в IIS (Windows) 7](#_Toc133856996)

[Вывод 12](#_Toc133856997)

**Цель работы:**

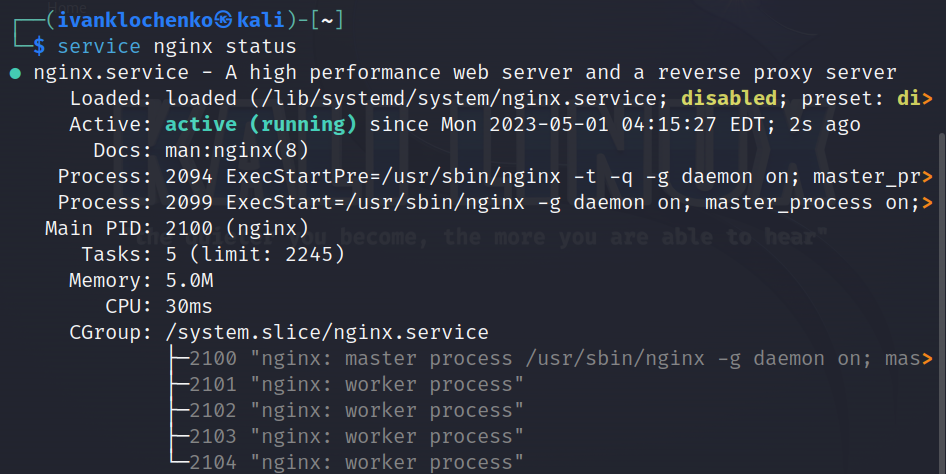
Получение теоретических и практических навыков создания и подписания сертификатов SSL в ОС Linux и Windows 7.

**Ход работы:**

## Организация HTTPS в NGINX путем создания самоподписанного SSL сертификата

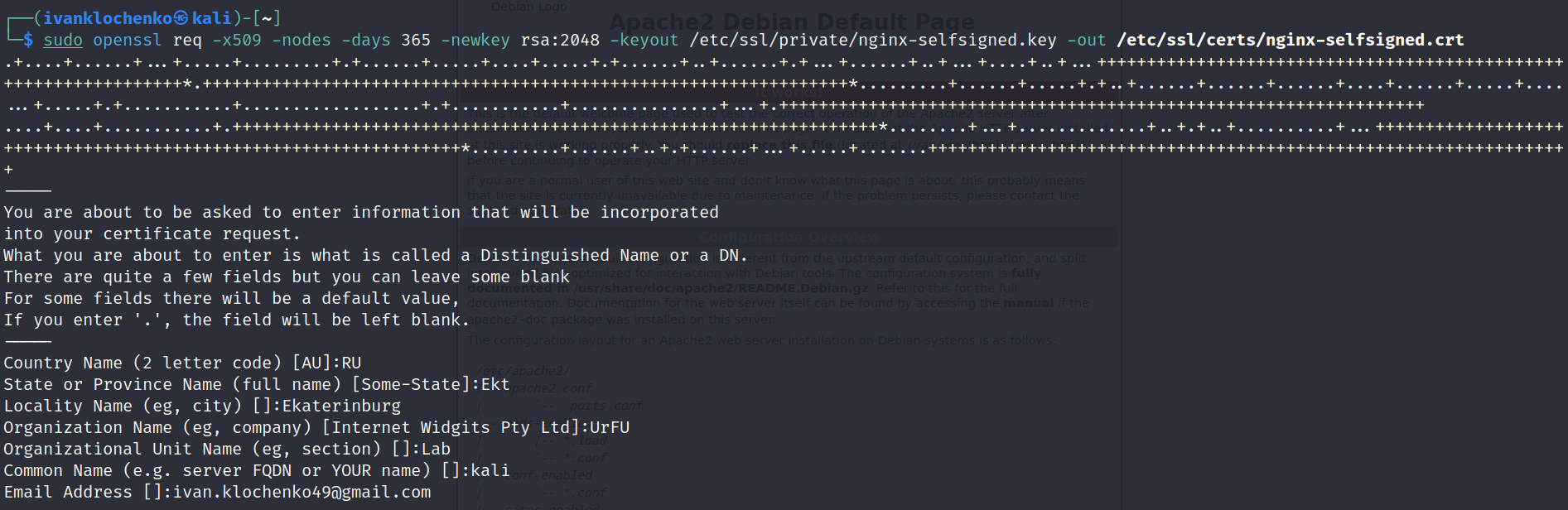
* 1. После ознакомления с теоретическим минимумом в методических указаниях к лабораторной работе перейдем к подготовке *Nginx* сервер для дальнейшего создания и тестирования. Для этого запускаем виртуальную машину (используем дистрибутив *Kali* *Linux*) и авторизуемся под пользователем *ivanklochenko*.

Обновляем базу пакетов приложений и устанавливаем веб-сервер *ngin*.



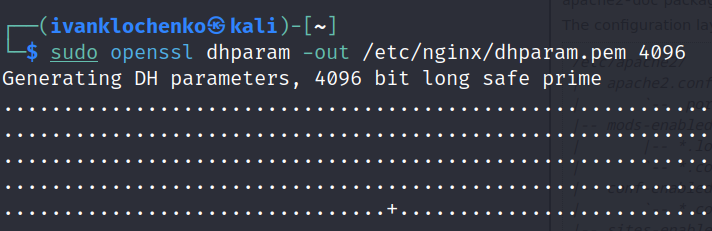
**Рисунок 1**. Проверка статуса сервиса Nginx

* 1. Переходим непосредственно к настройке *HTTPS*. Создадим самоподписанный ключ и пару сертификатов с *OpenSSL*.



**Рисунок 2**. Создание самоподписанного ключа и пары сертификатов с OpenSSL

* 1. Далее создаем сильную группу Диффи-Хелмана (протокол Диффи-Хелмана), используемая для согласования *Perfect* *Forward* *Secrecy* (совершенная прямая секретность) с клиентами.

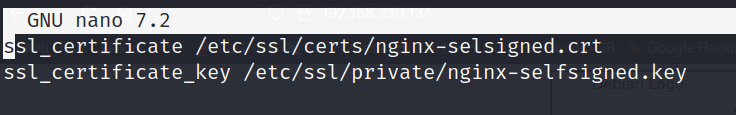


**Рисунок 3**. Создание сильной группы Диффи-Хелмана

* 1. Теперь нужно изменить конфигурацию *Nginx* для использования *SSL* сертификата.

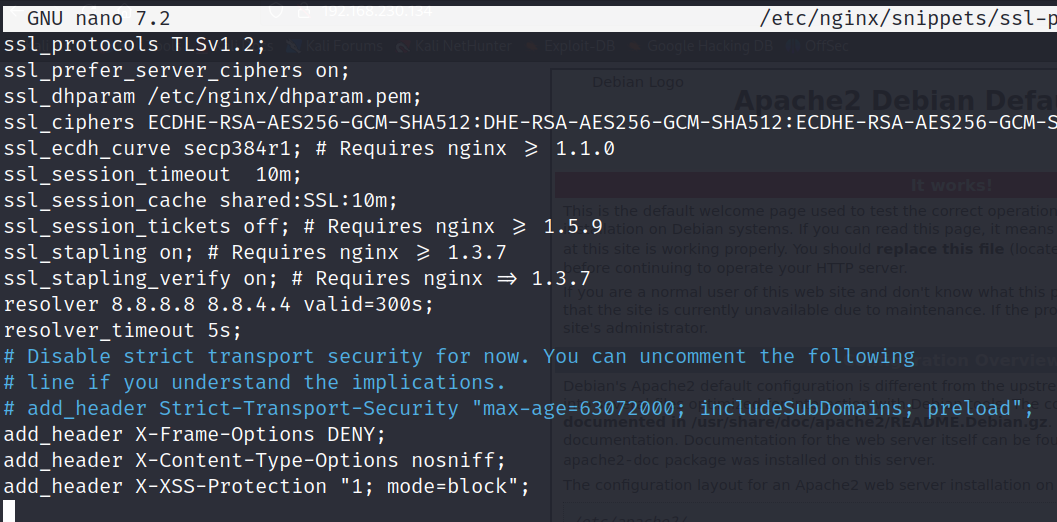
Создаем новый фрагмент конфигурации *Nginx* в каталоге /*etc*/*nginx*/*snippets*.

Заполняем следующими строками этот файл:



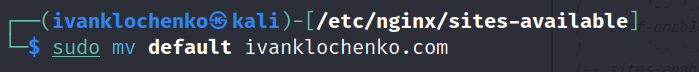
**Рисунок 4**. Установка директивы для файла сертификата и ключа

* 1. Создаем еще один фрагмент, который определит некоторые настройки *SSL*.



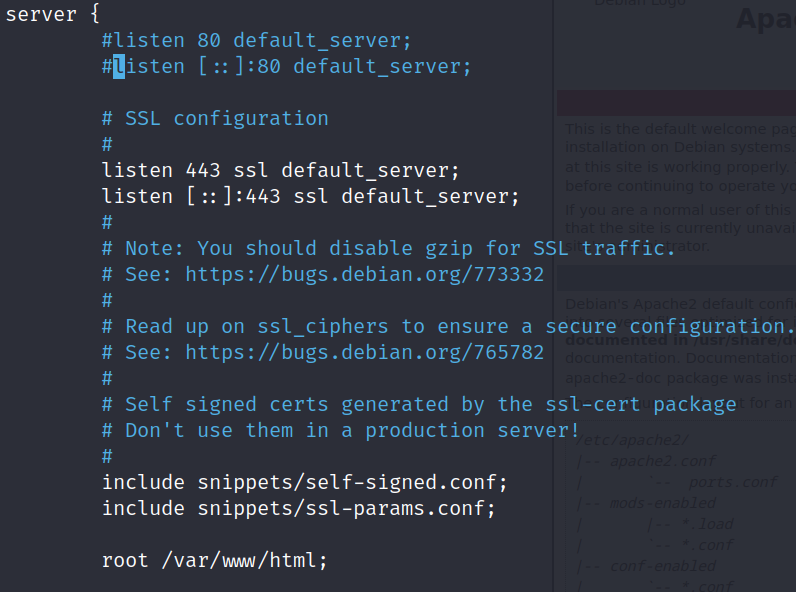
**Рисунок 5**. Создание параметра настроек SSL

* 1. В каталоге /*etc*/*nginx*/*sites*-*available* находим файл *default*. Переименуем его на *ivanklochenko*.*com*



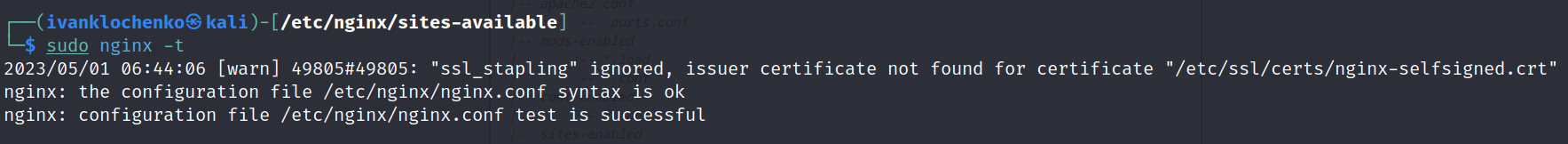
**Рисунок 6**. Переименовывание файла конфигурации

* 1. Открываем этот файл. Меняем конфигурацию следующим образом:



**Рисунок 7**. Изменение конфигурации (файл /etc/nginx/sites-available/ivanklochenko.com)

* 1. Теперь убедимся, что в файлах конфигурации нет ошибок.

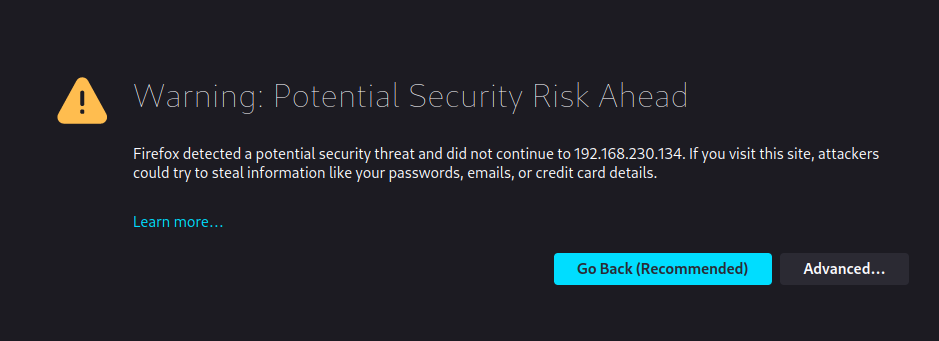


**Рисунок 8**. Проверка на ошибки в конфигурации

Единственная ошибка (предупреждение), которое выдает *Nginx*, было предусмотрено при формировании файла /*etc*/*nginx*/*snippets*/*ssl*-*params*.*conf*.

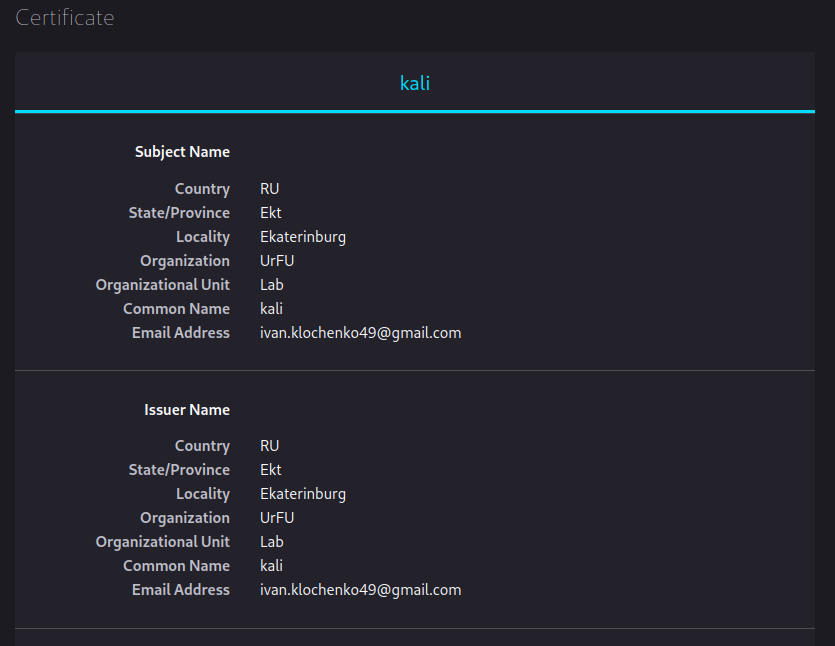
* 1. Перезагружаем сервер *Nginx*.
  2. Тестируем *SSL* сертификат. Вводим в адресной строке браузера *IP* адрес нашей виртуальной машины.

Получаем следующее предупреждение, которое ожидаемо, так как мы используем самоподписанный сертификат.



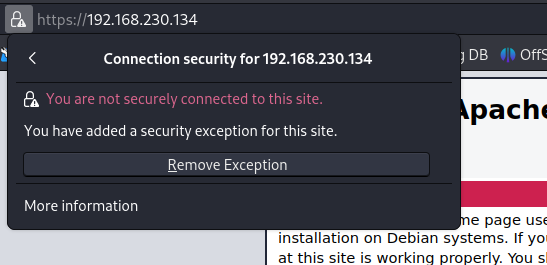
**Рисунок 9**. Получение предупреждения браузера о не защищенном соединении

Можно просмотреть подробности сертификата.



**Рисунок 10**. Подробности сертификата

Значок «замка» рядом со строкой поиска:

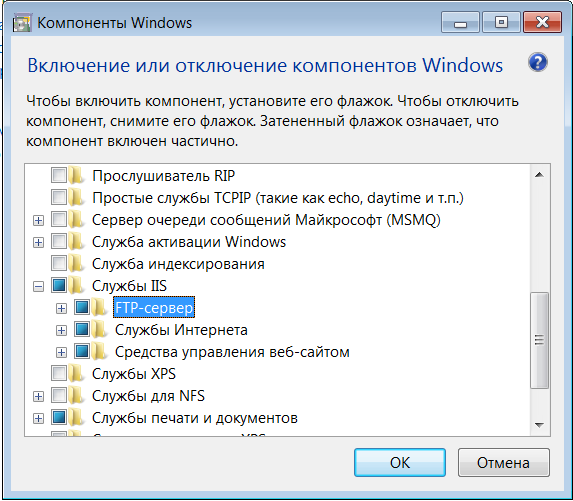


**Рисунок 11**. Значок «замка»

Главная цель достигнута. Сервер работает и *SSL* сертификат тоже.

## Организация HTTPS в IIS (Windows)

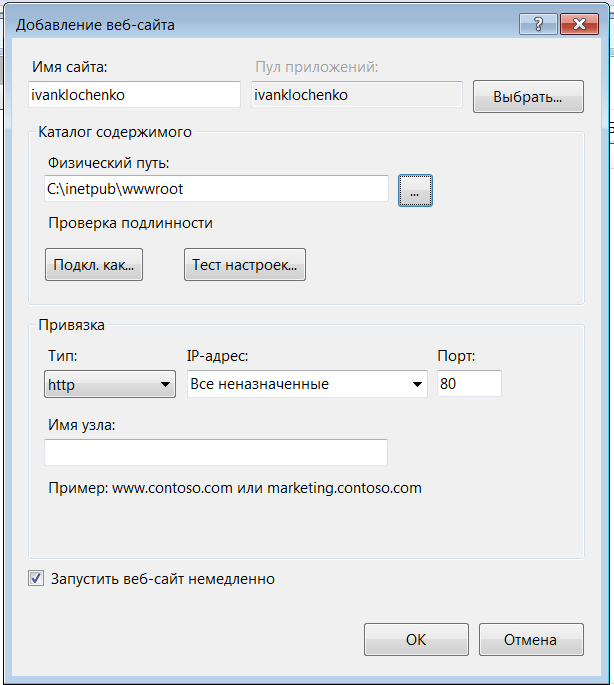
* 1. Запускаем виртуальную машину *Windows* 7.
  2. Включаем компоненты служб *IIS*



**Рисунок 12**. Включение компонентов служб IIS

Перезагружаем виртуальную машину.

* 1. Создадим тестовый сайт для тестирования в нем сертификатов *SSL*.

****

**Рисунок 13**. Создание тестового сайта

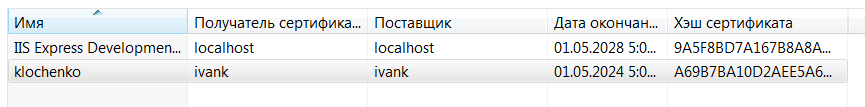
После подтверждения, нас предупредят что на порту 80 уже есть работающий сайт (*Default* *Web* *Site*), поэтому соглашаемся на вторичную привязку и завершаем настройку сервера. В окне с сайтами останавливаем сайт *Default* *Web* *Site* и запускам *ivanklochenko*.

Убедимся, что сервер работает.



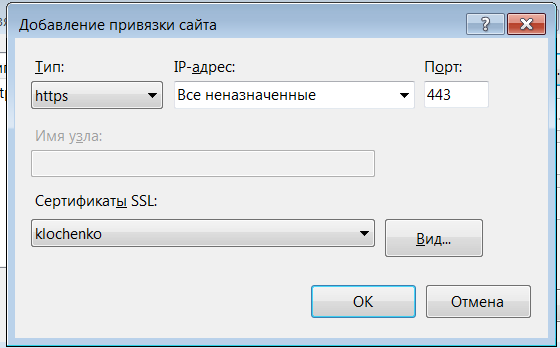
**Рисунок 14**. Сервер работает

* 1. Повторно открываем *IIS* менеджер и переходим в корень настроек. Создаем самозаверенный сертификат.



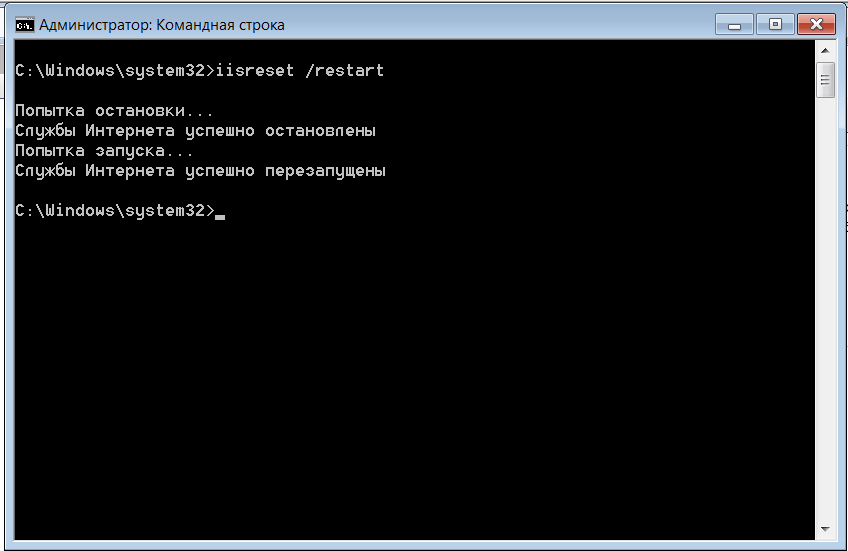
**Рисунок 15**. Создание самозаверенного сертификата

* 1. Теперь изменим привязки для созданного ранее сайта.



**Рисунок 16**. Изменение привязок сайта на HTTPS (порт 443)

* 1. Теперь открываем командную строку и перезапускаем *IIS* сервер.



**Рисунок 17**. Перезапуск IIS сервера

Важно отметить, большинство действий в этой части лабораторной работы необходимо выполнять с правами администратора. Этот шаг, не исключение.

Перезагружаем виртуальную машину.

* 1. Проверяем верность настроек. Открываем браузер и переходим по ссылке нашего сайта:



**Рисунок 18**. Сообщение о проблемах с сертификатом

Видим это сообщение – это знак того, что все работает как надо. Мы создали сертификат, который заверен только нами, а это значит, что браузер будет нас предупреждать об этом при каждом открытии сайта.

Продолжим открытие сайта.



**Рисунок 19**. Открытие сайта с самозаверенным сертификатом

Подобно ситуации с *Nginx*, нас встречает «шапка» адресной строки, которая всеми силами пытается привлечь внимание, чтобы напомнить об ошибке сертификата. Мы успешно установили *SSL* сертификат, это главное.

## Вывод

В ходе ознакомлением теоретической части лабораторной работы были усвоены такие понятия, как протоколы SSL и TLS, цепочка шифрования, протоколы HTTP и HTTPS. Была изучена структура работы SSL сертификатов.

В ходе выполнения лабораторной работы, были изучены и практически применены навыки создания *SSL* сертификатов в ОС *Linux* (*Kali* *Linux*) и *Windows* 7.

Для создания веб-страницы мы использовали *Nginx* (*Linux*) и *IIS* (*Windows* 7).

Суть и результат, к которым мы стремились и в первой, и во второй части лабораторной работы одна – создание самозаверенного (самоподписанного) *SSL* сертификата для *Web*-ресурса. У нас это получилось в обоих случаях.