Уральский федеральный университет

имени первого Президента России Б. Н. Ельцина

Кафедра «Интеллектуальные информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА 2**

ОТЧЁТ

Дисциплина «Web-программирование»

Преподаватель:

Д.В. Свинцов

Студент:

гр. РИ-380022

Борисова В. Б.

Екатеринбург, 2020

**Лабораторная работа 2**

**Работа с протоколом HTTPS через openssl**

1. Задание:
   1. Создать ключ шифрования для работы по зашифрованному каналу связи:

openssl req – new -x509 -keyout key pem -out server.pem – days 365 – nodes

* 1. Поднять веб сервер работающий по протоколу HTTPS:

import ssl  
from http.server import HTTPServer, SimpleHTTPRequestHandler  
  
httpd = HTTPServer(("0.0.0.0", 4443), SimpleHTTPRequestHandler)  
httpd.socket = ssl.wrap\_socket(  
httpd.socket,  
certfile="server.pem",  
keyfile="key.pem",  
server\_side=True,  
ssl\_version=ssl.PROTOCOL\_TLS,  
)  
httpd.serve\_forever()

* 1. Отправить запрос на локальный сервер:

openssl s\_client -connect 127.0.0.1:4443

1. Ход работы:
   1. Для того, чтобы создать ключ шифрования для работы по зашифрованному каналу связи и поднять веб-сервер работающий по протоколу HTTPS, нужно отправить HTTPS-запрос.

Выполним следующую команду:

$ openssl req -new -x509 -keyout key.pem -out server.pem -days 365 –nodes

В результате выполнения команды будут созданы два ключа. Один приватного использования, второй публичного, со сроком использования 365 дней (рисунок 1).



Рисунок 1 — Созданные ключи

* 1. Поднимем веб-сервер, работающий по протоколу HTTPS.

Для этого через командную строку создадим файл с расширением .py, запишем в него следующее (рисунок 2):

import ssl  
from http.server import HTTPServer, SimpleHTTPRequestHandler  
  
httpd = HTTPServer(("0.0.0.0", 4443), SimpleHTTPRequestHandler)  
httpd.socket = ssl.wrap\_socket(  
httpd.socket,  
certfile="server.pem",  
keyfile="key.pem",  
server\_side=True,  
ssl\_version=ssl.PROTOCOL\_TLS,  
)  
httpd.serve\_forever()

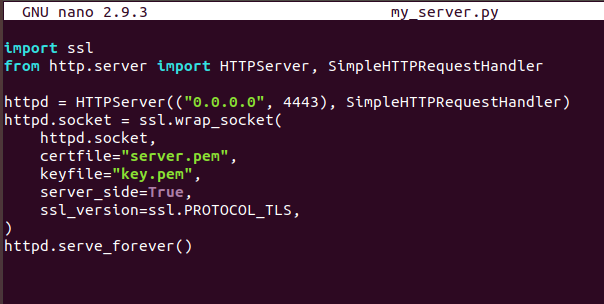


Рисунок 2 — Код сервера

Дальше с помощью команды: python3 my\_server.py запустим сохраненный файл. В результате запуска сервер не выводит никакую информацию о работе, поэтому отправим ему HTTPS-запрос, с помощью следующей команды:

sudo openssl s\_client -connect 127.0.0.1:4443

Результат выполнения команды (рисунок 3):

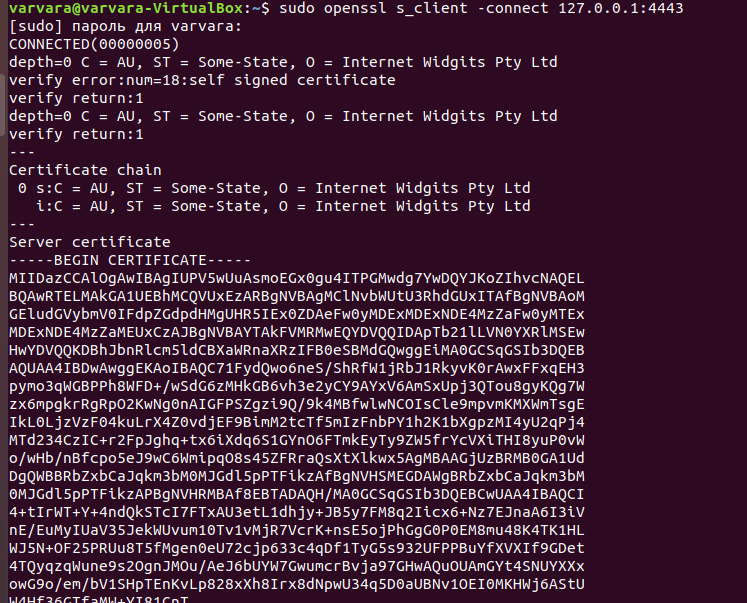


Рисунок 3 — Результат подключения к локальному серверу