# **Отчет о практической работе**

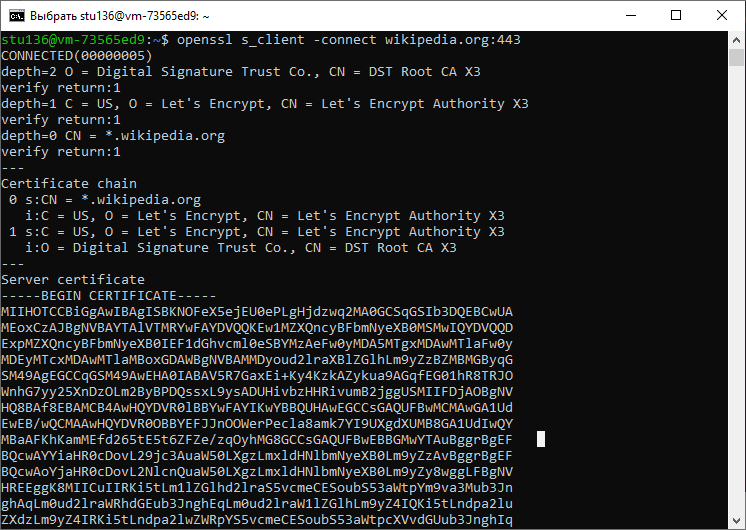
# **«Работа с протоколом HTTPS через openssl»**

# Задание 1.

Подключиться по openssl к https://wikipedia.org и отправить запросopenssl s\_client -connect wikipedia.org:443

openssl s\_client -connect wikipedia.org:443

подключимся по openssl к wikipedia.org.



Происходит обмен сертификатами с сервером для установления зашифрованного соединения, сертификат сервера можно увидеть после строки «-----BEGIN CERTIFICATE----->.

Соединение произошло успешно и теперь можно отправить к Wikipedia.org GET запрос:

GET /wiki/страница HTTP/1.1

Host: ru.wikipedia.org

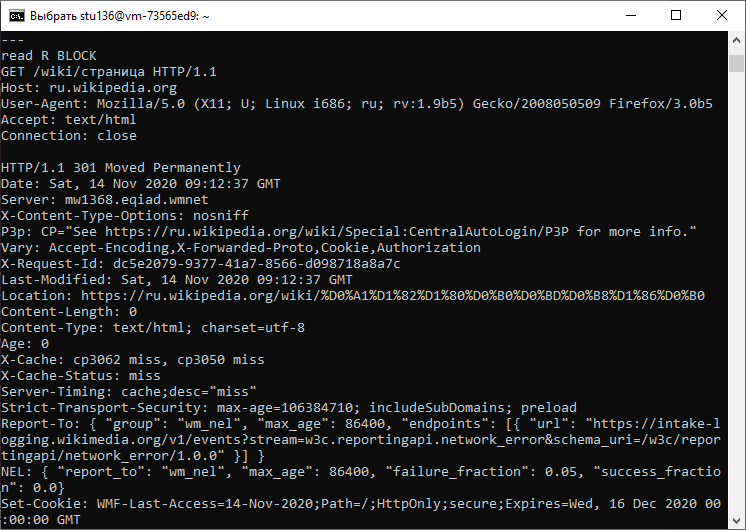
User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5

Accept: text/html

Connection: close

(пустая строка)

Перед отправкой наш запрос шифруется с помощью полученного от сервера сертификата.



Получили от сервера код 301, что говорит о том, что указанная нами страница была перемещена по адресу, указанному в заголовке “Location”.

# Задание 2.

Создадим ключ шифрования для работы по зашифрованному каналу связи

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

В результате этой операции были созданы 2 файла: публичный и закрытый ключи.



Теперь поднимем веб сервер, работающий по протоколу HTTPS, создав python файл, вставив в него следующий код и запустив из терминала:

**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()

Далее происходит запуск только что написанного нами HTTPS сервера.

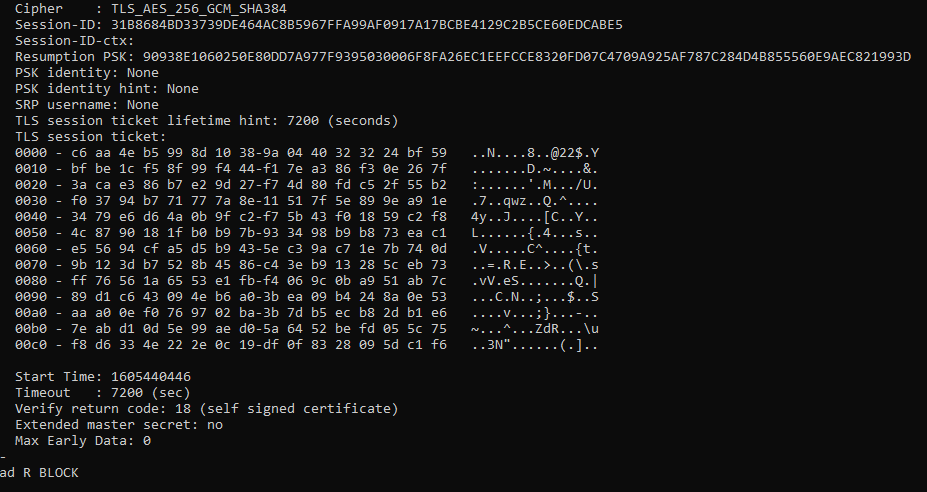


# Задание 3.

Подключимся по openssl к httpbin.org через команду

openssl s\_client -connect httpbin.org:443

Подключение прошло успешно. Произошел обмен сертификатами с сервером и установлено зашифрованное соединение.

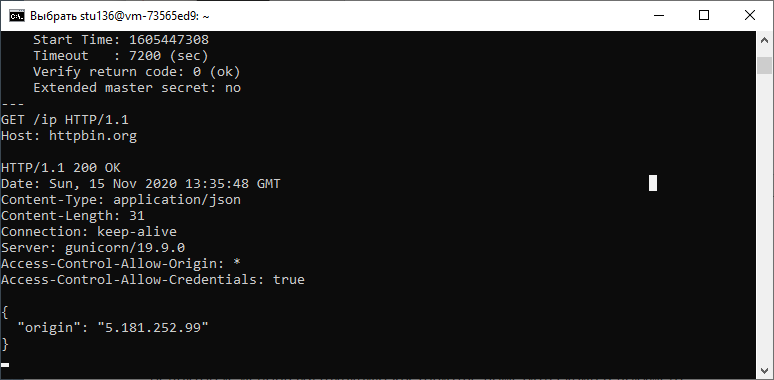


Далее запросим данные *GET* запросом с ресурса *ip.*

GET /ip HTTP/1.1

Host: httpbin.org

Accept: \*/\*



В ответ пришло сообщение с кодом «200», сообщающее об успешном выполнении запроса. Длина сообщения составила 34 символа (заголовок Content-Length), и, как можно видеть в самом сообщении, ip адрес httpbin.org – 178.176.113.142.

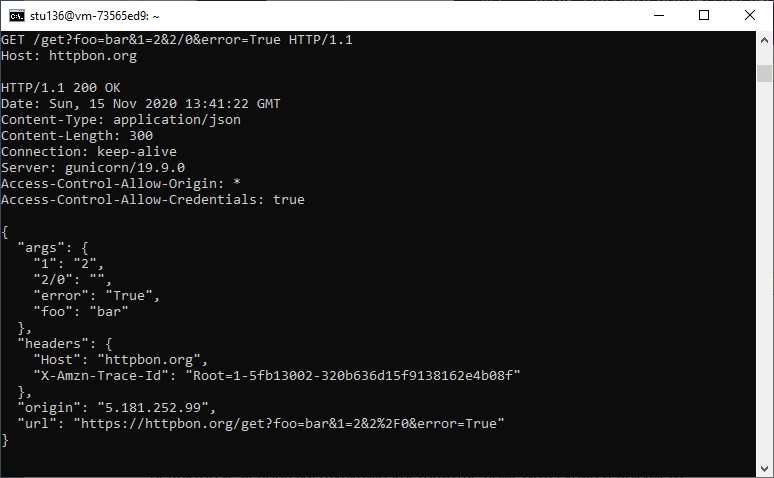
Далее выполним запрос методом *GET*:

GET /get?foo=bar&1=2&2/0&error=True HTTP/1.1

Host: httpbin.org

Accept: \*/\*

Пришел ответ с кодом 200, что говорит о том, что сервер обработал запрос успешно. В ответе мы можем видеть переданные аргументы, длина сообщения составила 324 символа.



Далее выполним запрос методом *POST*:

POST /post HTTP/1.1

Host: httpbin.org

Accept: \*/\*

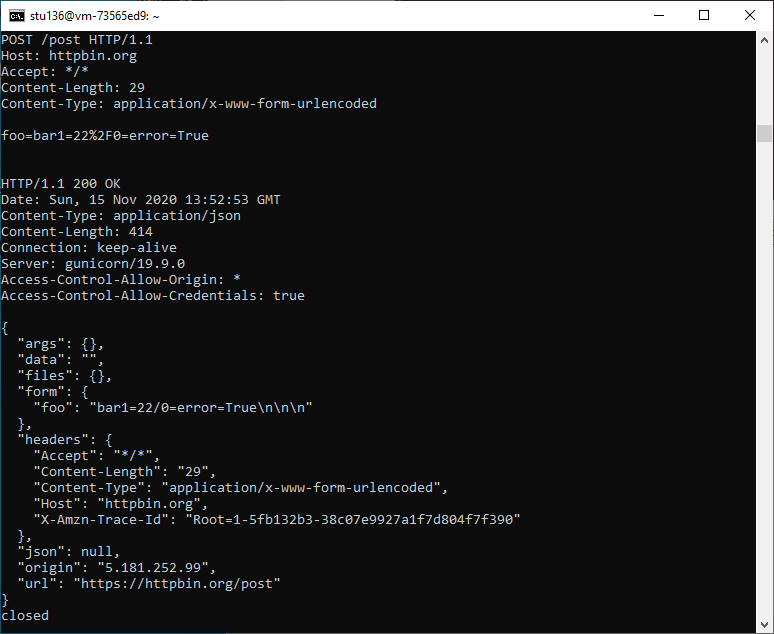
Content-Length: вычислить длину контента и вставить сюда число

Content-Type: application/x-www-form-urlencoded

foo=bar&1=2&2%2F0=&error=True

Длина передаваемого сообщения составляет 29 символов – это аргумент заголовка Content-Length.

Пришел ответ с кодом 200, что говорит о том, что сервер обработал запрос успешно. В ответе мы можем видеть переданные аргументы, длина сообщения составила 446 символов.



Переданная информация была записана в аргумент “form”.

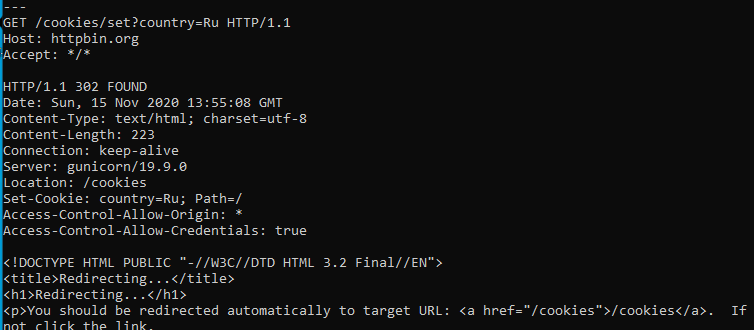
Если указать значение Content-Length меньше, чем длина строки с передаваемыми аргументами, то часть информации просто потеряется.

Далее отправим запрос на установку кук

GET /cookies/set?country=Ru HTTP/1.1

Host: httpbin.org

Accept: \*/\*



Результат операции – перенаправление на станицу /cookies. Об этом говорит код 302 в сообщении.

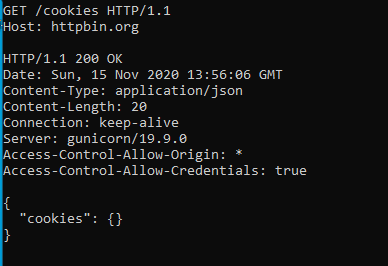
Теперь проверим заданные куки

GET /cookies HTTP/1.1

Host: httpbin.org

Accept: \*/\*

Проанализируем ответ:

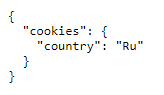


Ответ от сервера пришел с кодом 200 – это успех, но видим, что у параметра “cookies” нет значений - это связано с особенностью telnet, который не сохраняет значения кук.

Попробуем повторить то же самое через браузер.

Вводим в адресной строке httpbin.org/cookies/set?country=Ru

Теперь переходим по httpbin.org/cookies и видим, что куки установлены:

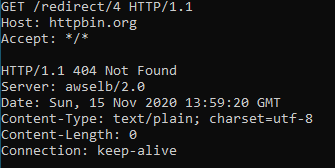


Отправим запрос на страницу с перенаправлением:

GET /redirect/4 HTTP/1.1

Host: httpbin.org

Accept: \*/\*



Пришло сообщение с кодом 404, что говорит о том, что страница не найдена.