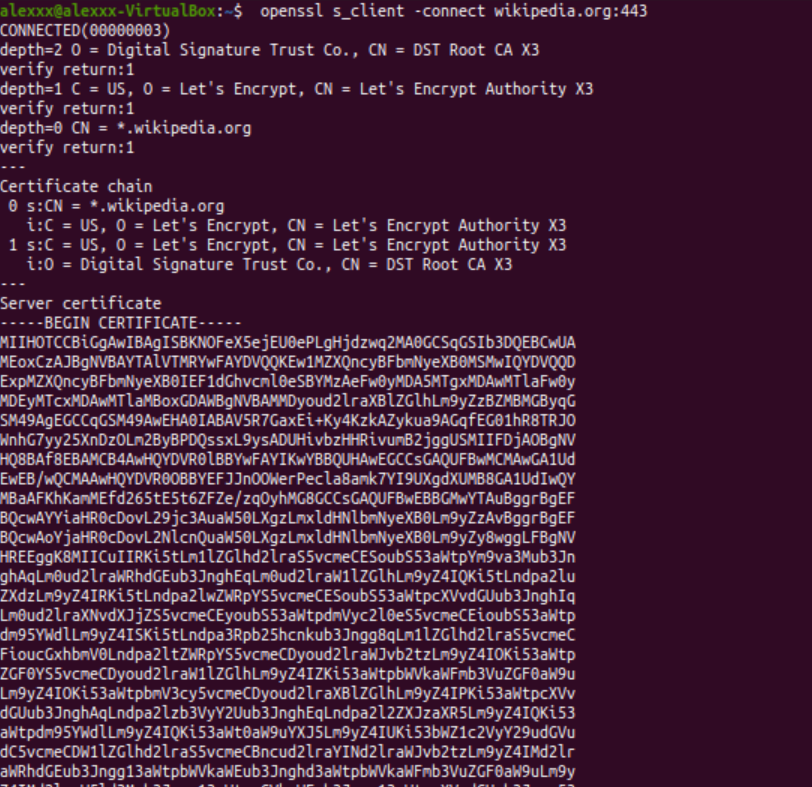
**Лабораторная работа 2.  
Тюнькин Александр Евгеньевич РИ-380014.**

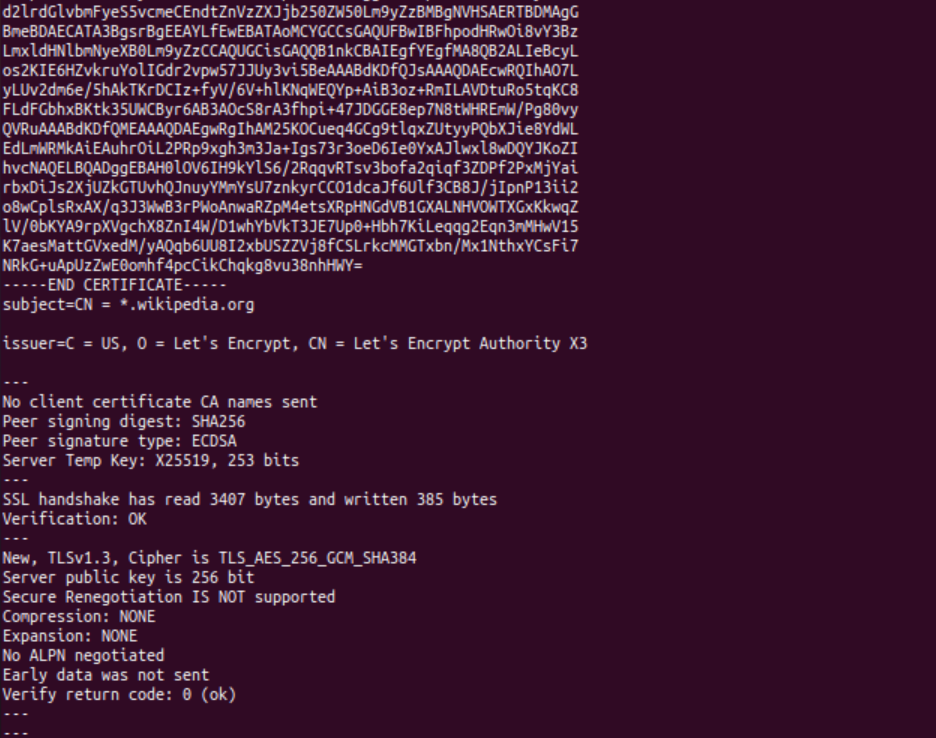
Работа с протоколом HTTPS через openssl

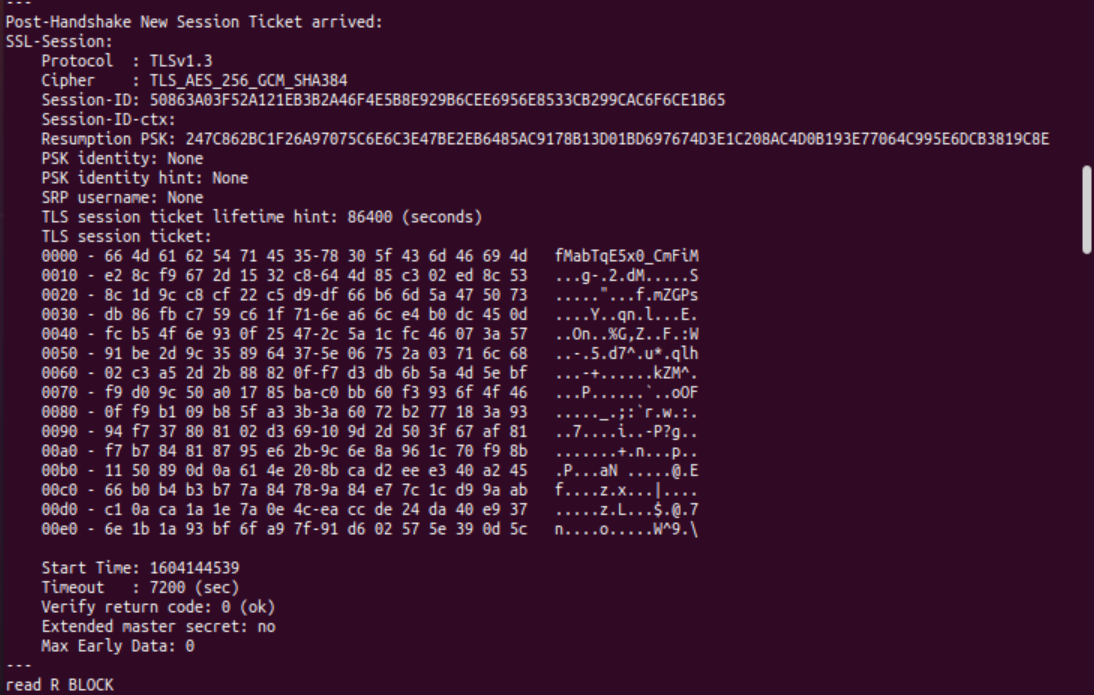
**Задание 1**

Команда при подключении к wikipedia.org через openssl по 443 порту указаны на изображении ниже. На данном этапе происходит соединение с SSL/TLS-сервером. Соединение было установлено успешно, о чем нам говорит CONNECTED в первой строке.

Далее демонстрируется результат проверки цепочки сертификатов, использованной при установке соединения, где depth – это глубина каждого сертификата в цепочке, также есть С – страна, О – организация, ОU – подразделение, СN – имя, принадлежащие субъекту сертификации. verify return – результат проверки.   
Далее мы видим текст сертификата в формате pem.

Ниже мы можем заметить, что форматом подписи является SHA256 ECDSA. Используется протокол TLSv1.3. Происходит генерация сеансового ключа.

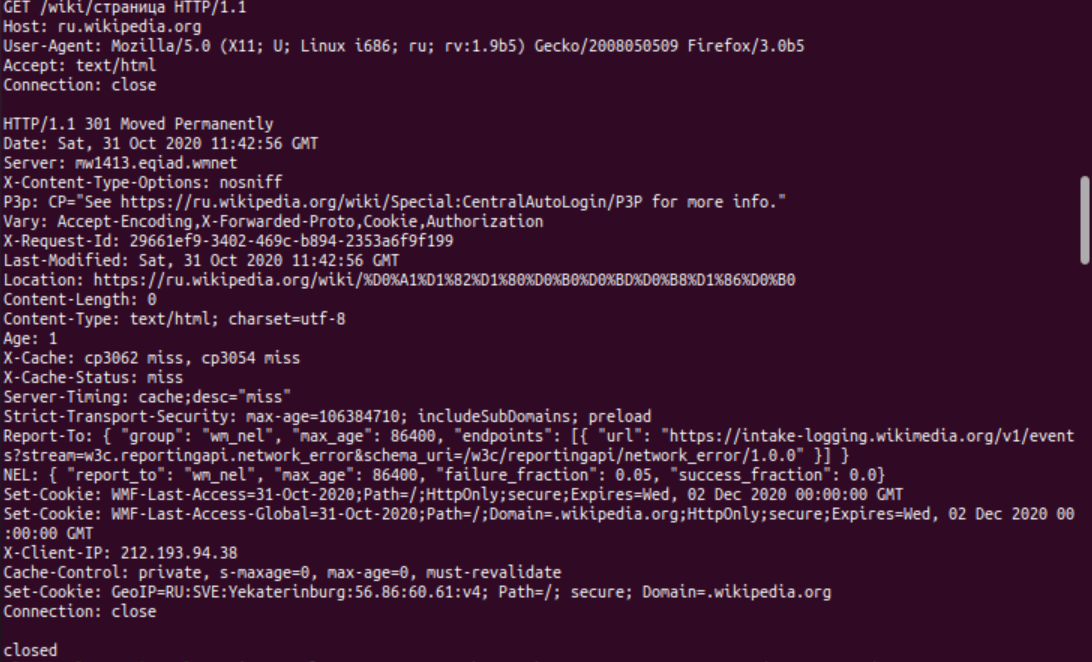




Как мы можем заметить, в результате подключения к серверу с использованием openssl, мы видим, что при подключении к серверу используется ssl-шифрование в виде сертификата сервера. Данные зашифрованы.

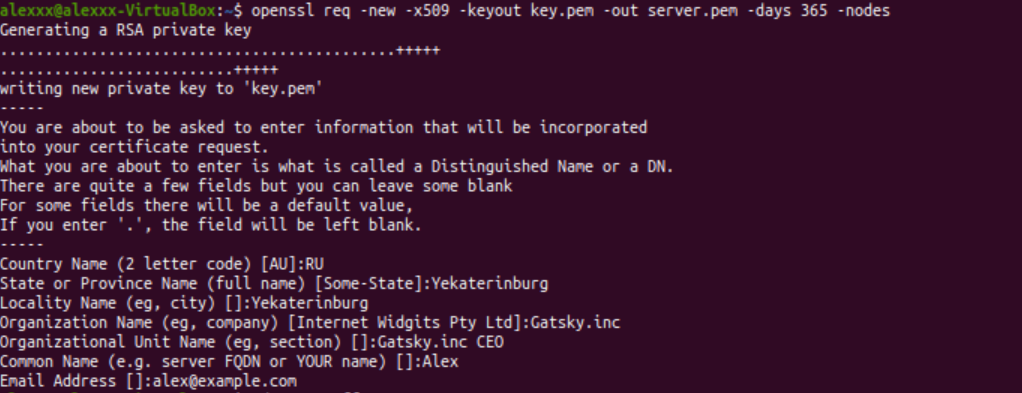
Далее мы можем увидеть, что введенный нами запрос завершился с кодом 301 – ресурс, к которому мы пытаемся обратиться, перемещен. Также мы можем видеть клиентский IP адрес, установку куки. Тип предоставляемого ответа виден в поле Content-Type, и в нашем случае сервер должен был вернуть данные в формате html/css. В конце соединение разрывается.

Данные сессии используются для обмена с ключом симметричного шифрования.

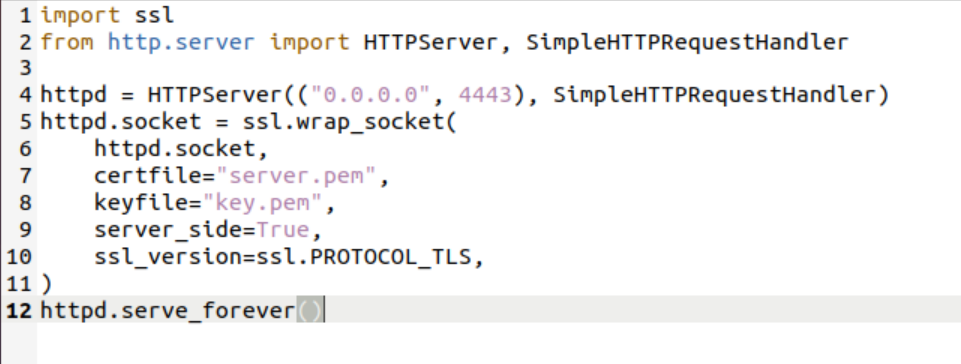


**Задание 2**

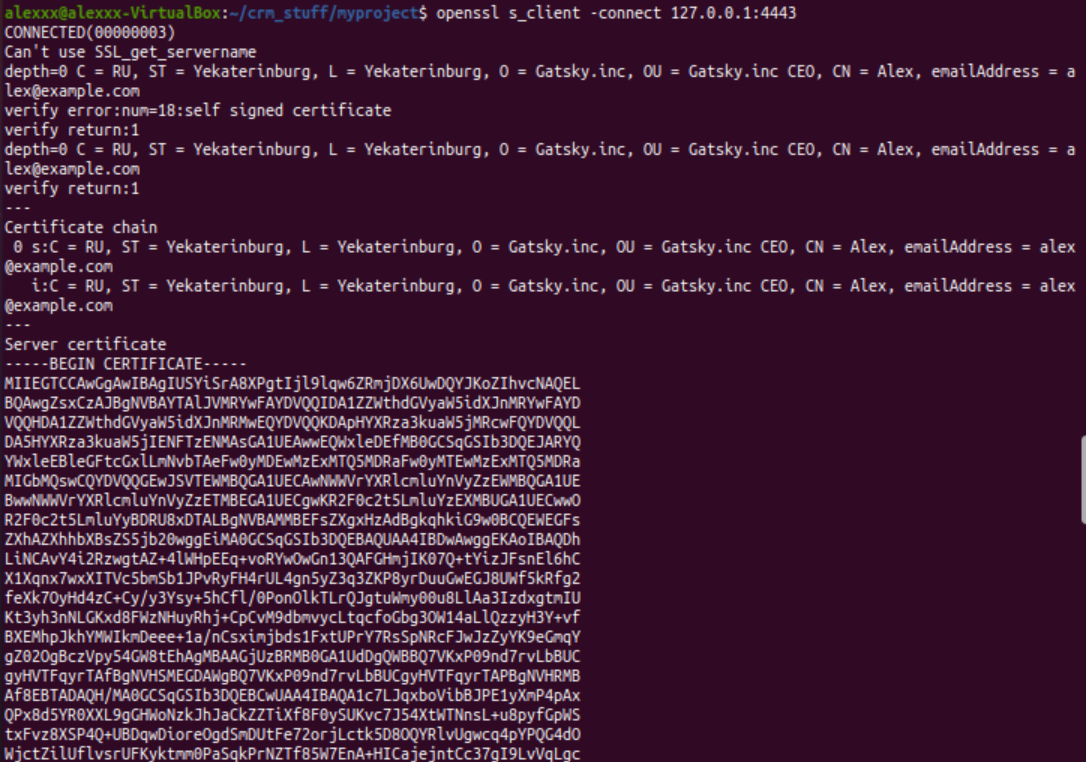
Происходит создание сертификата сервера для работы по защищенному каналу связи. Он доступен в течение 365 дней. Для этого нам нужно указать название страны, региона и города, а также название организации и ее подразделения, на которые будет зарегистрирован ключ. Также необходимо указать имя и электронный адрес.



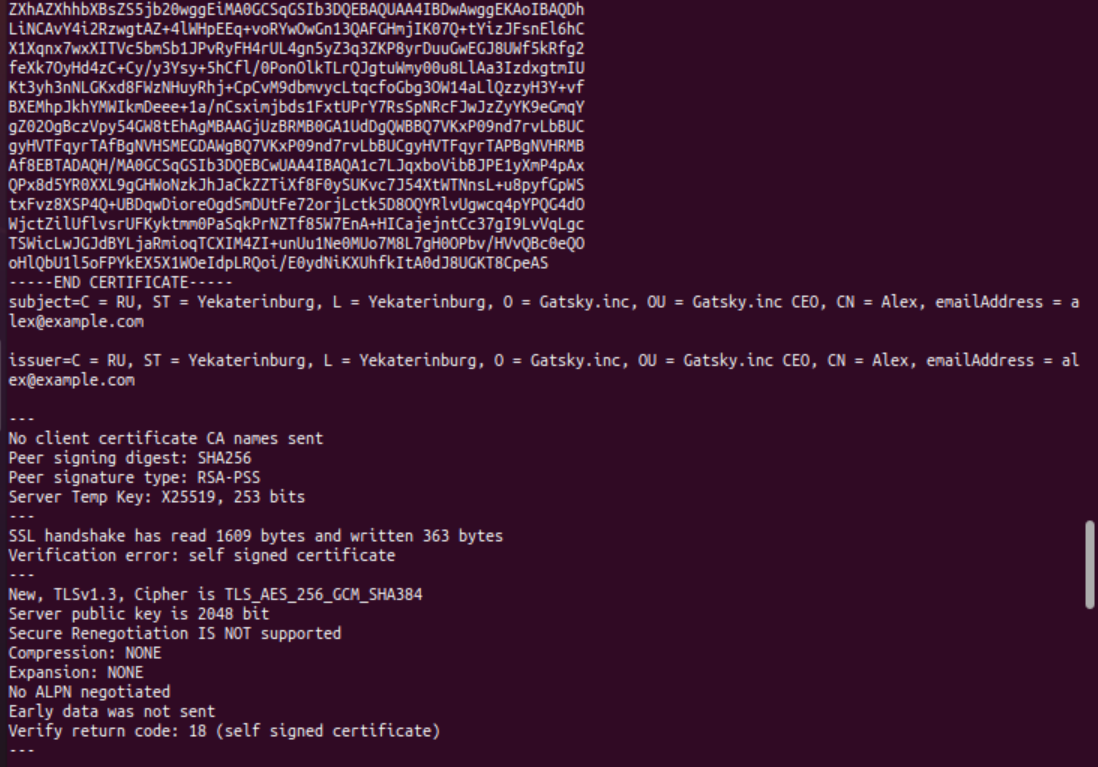
Далее нам необходимо создать файл с расширением .py, который запустит сервер. Его содержание указано ниже. Соединение будет производиться по порту 4443.

Далее происходит запуск только что написанного нами HTTPS сервера.

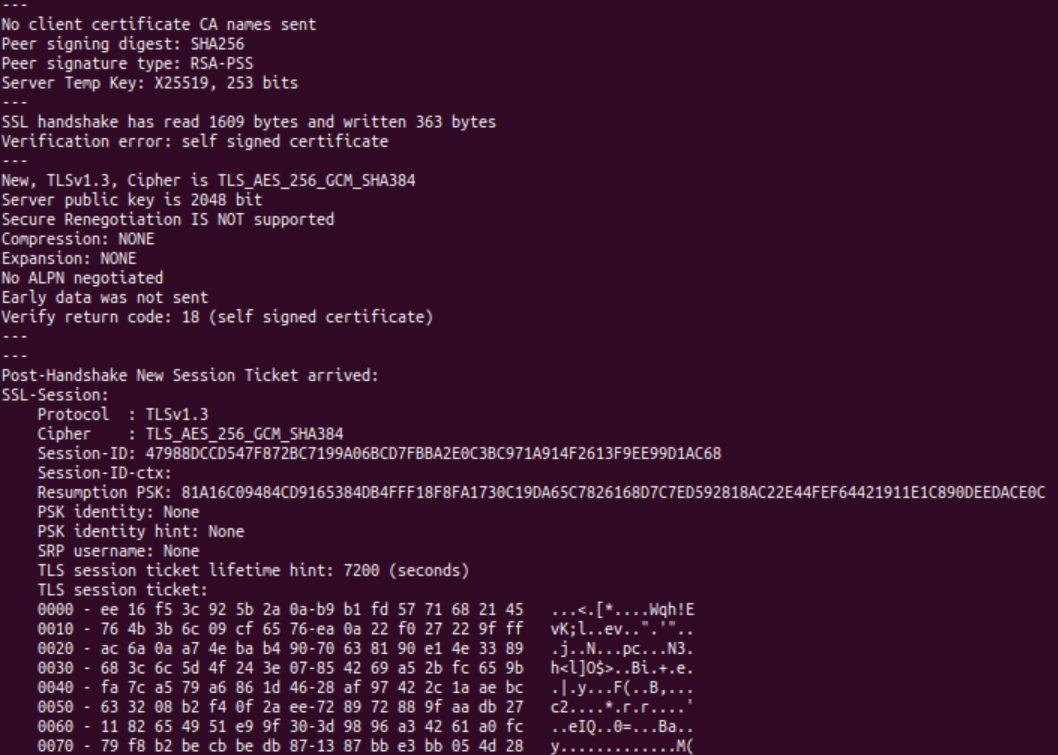


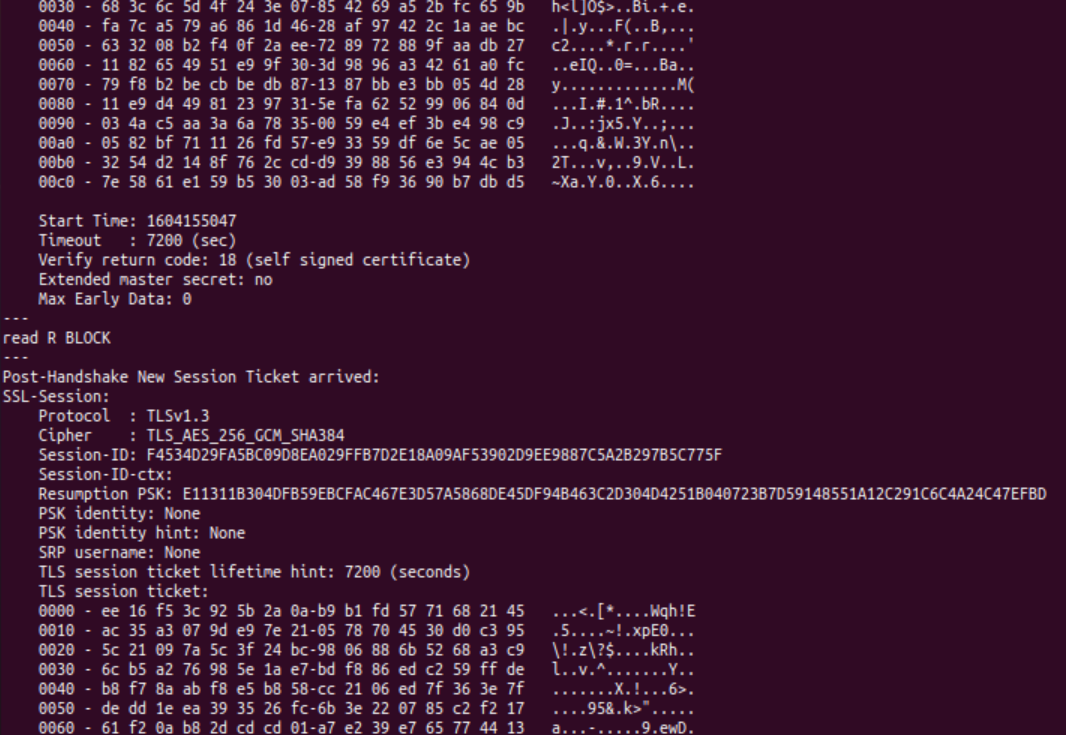
Для соединения с сервером используется команда, указанная на рисунке ниже. Сначала мы можем видеть ошибку верификации – ключ был сделан самому себе. Далее мы можем видеть саму подпись в форме объемного набора символов – начинается с BEGIN CERTIFICATE, и оканчивается на END CERTIFICATE.  


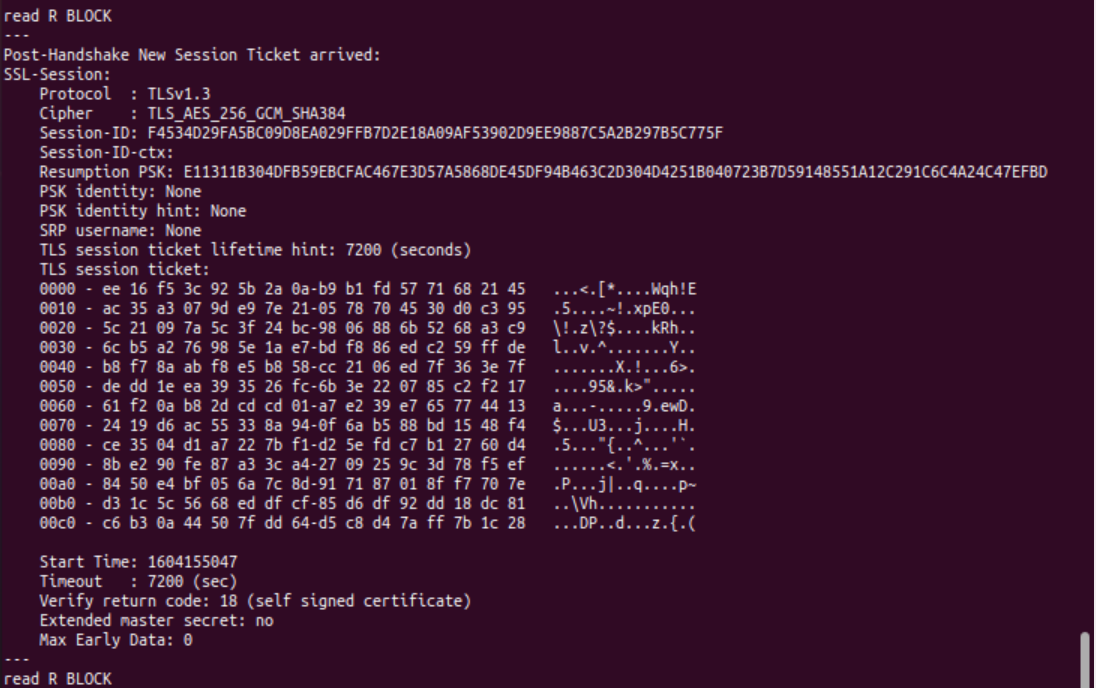
Ниже мы наблюдаем данные из сертификатов. Мы видим новую версию протокола – TSL v1.3, а также формат подписи – SHA256. Шифр: TSL\_AES\_256\_GCM\_SHA384. Сжатие данных отсутствует.



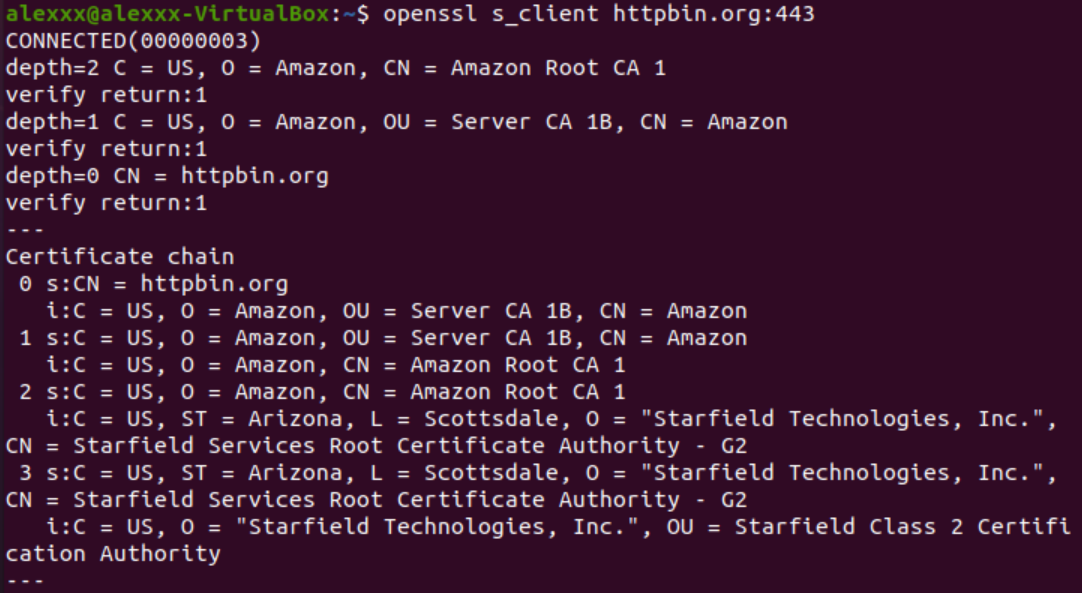
На этом этапе генерируется сеансовый ключ. Данные текущей SSL/TLS сессии используются для обмена ключом для симметричного шифрования. Ключи меняются каждую новую сессию.

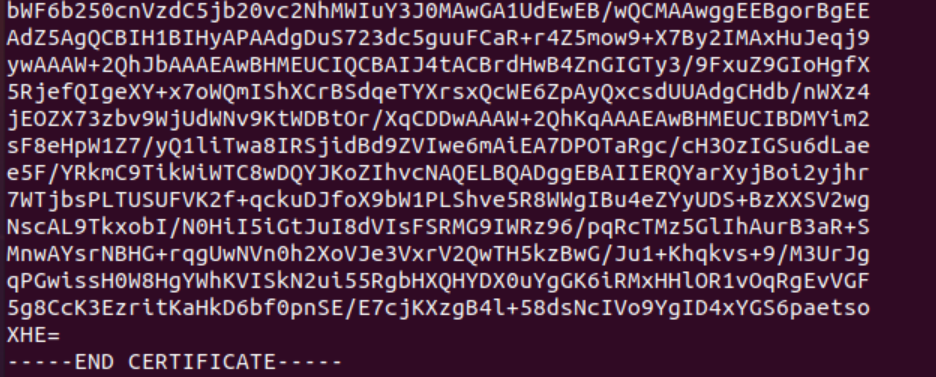




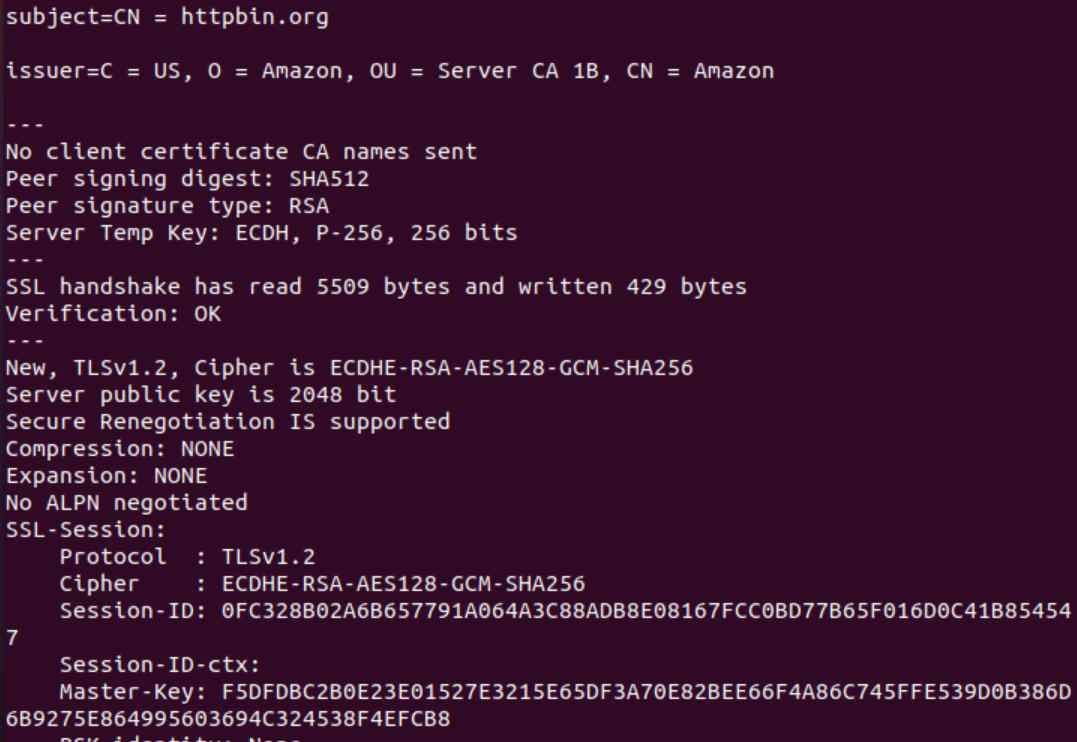


**Задание 3**

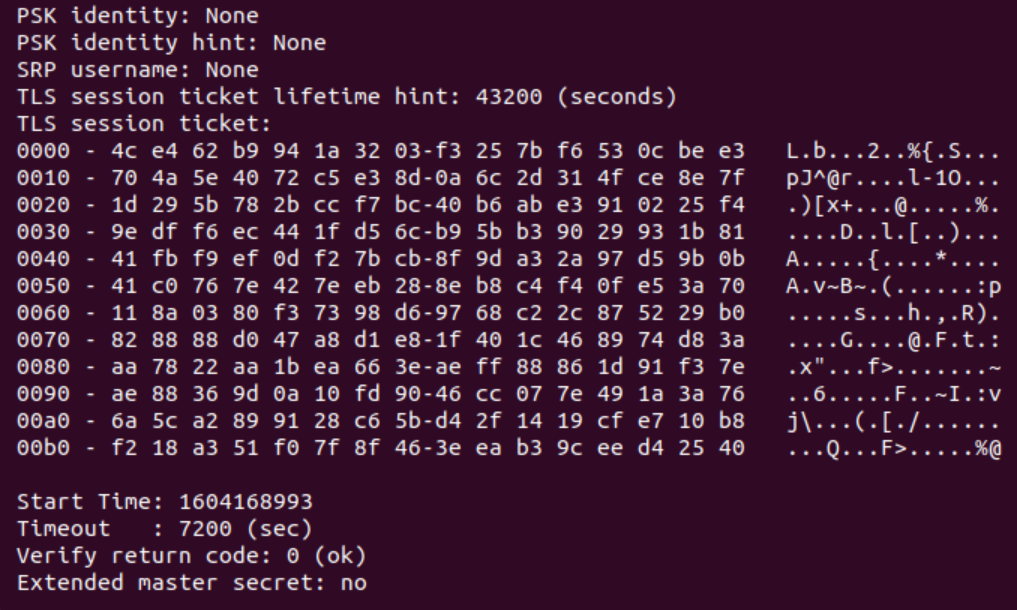
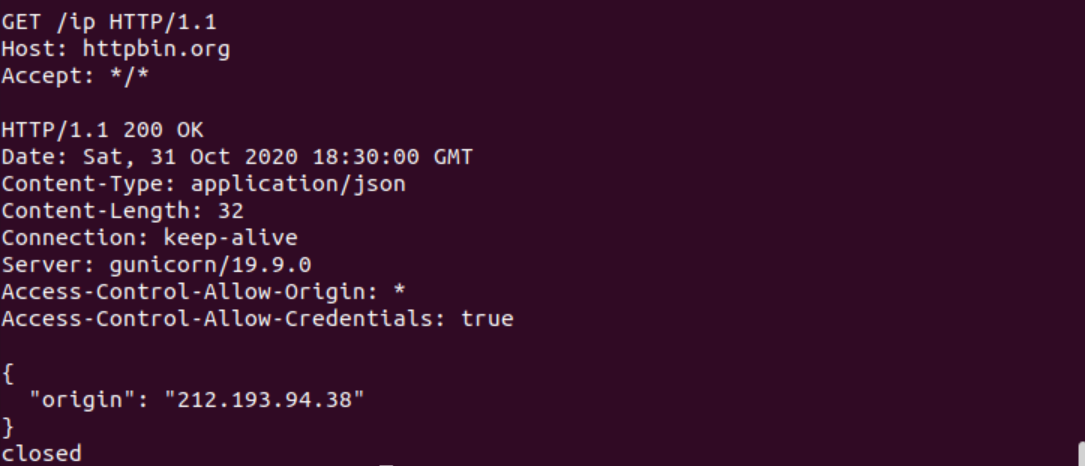
Идет подключение к серверу по 443 порту, а также обмен сертификатами с использованием утилиты openssl. Сервер зарегистрирован в соединенных штатах, организация – амазон. Штат Аризона.



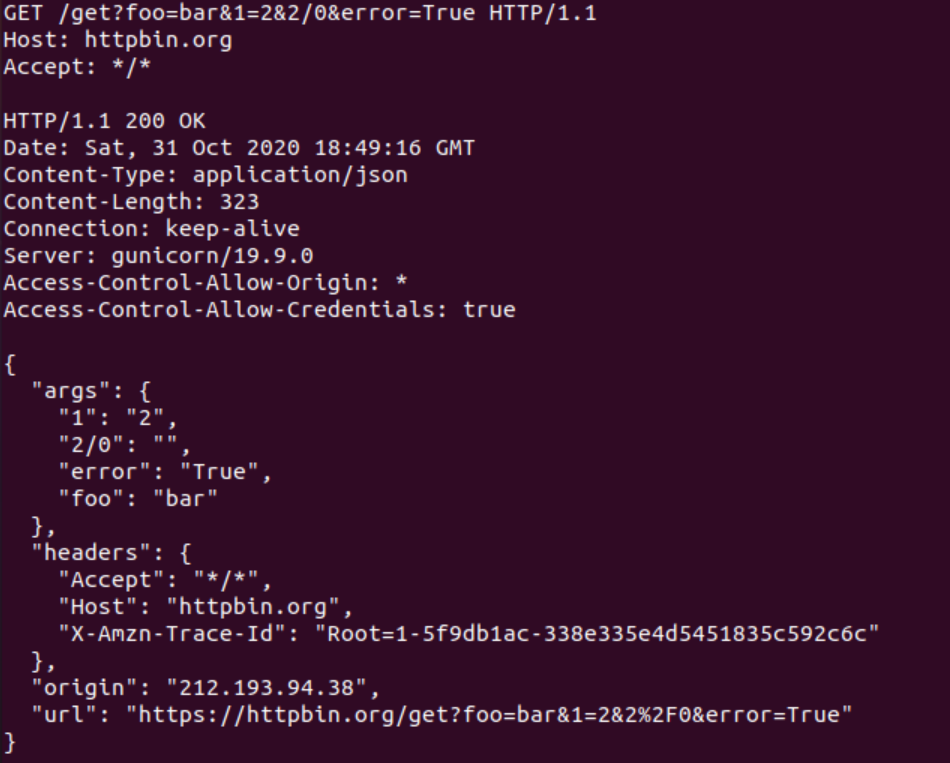
Мы можем заметить, что шифрование ведется по протоколу TLSv1.2 при формате подписи SHA512 RSA.



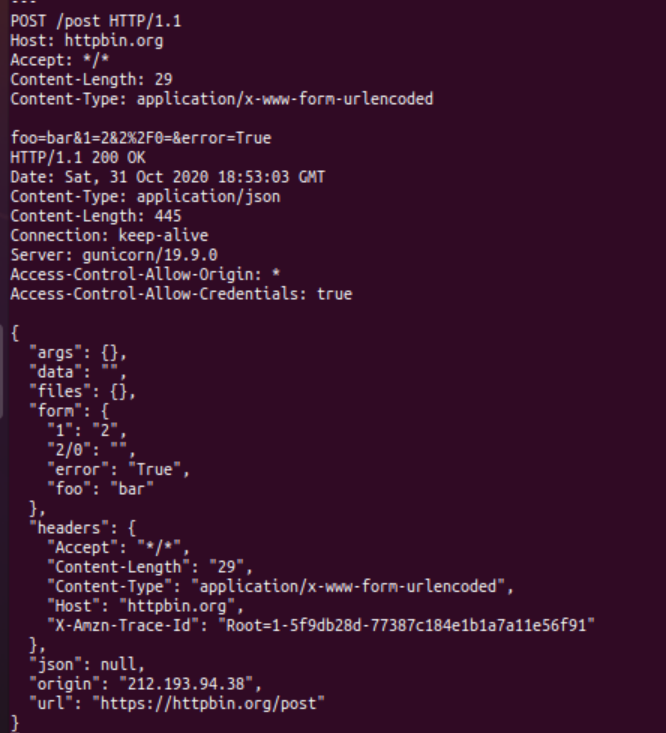
Данные текущей SSL/TLS сессии используются для обмена ключом для симметричного шифрования. Ключи меняются каждую новую сессию.

Далее мы можем видеть текст запроса, который нам необходимо отправить при подключении, а ниже ответ сервера. Код состояния в результате выполнения запроса – 200, что говорит нам о его успешном выполнении. Результат выполнения запроса – данные в формате json, предоставляющие нам ip адрес. Длина ответа – 32 символа.  


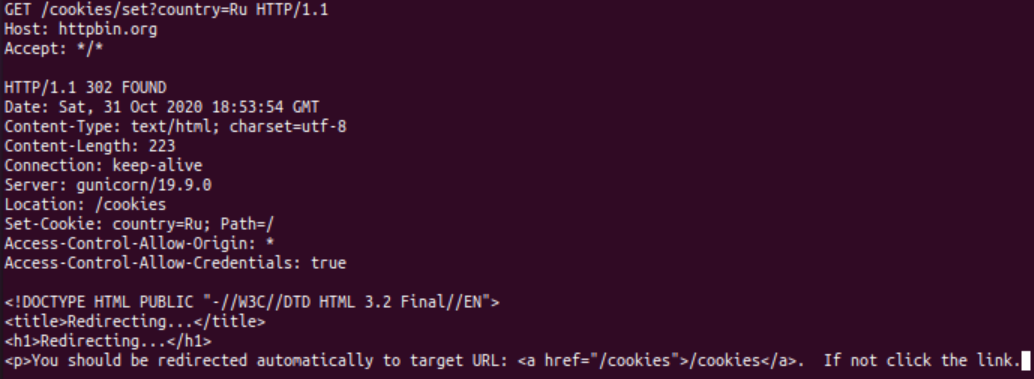
Далее отправим на сервер запрос с методом get. Текст запроса и его результат представлены на рисунке ниже. Результат выполнения запроса представлен с состоянием 200, что означает, что запрос был выполнен успешно. В результате запроса нам был передан ответ в формате json – аргументы, переданные нами серверу.



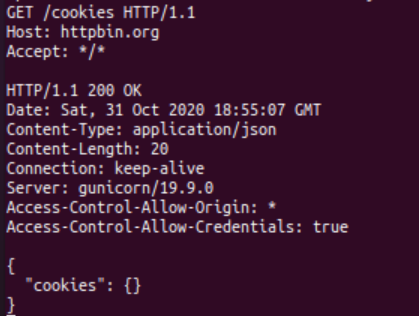
Далее отправим на сервер запрос методом POST. Сам запрос и результат его выполнения – на рисунке ниже. В поле Content-Length вводим длину запроса в символах, в нашем случае это 29, включая символ переноса. Результат выполнения запроса представлен на рисунке 12. Завершился с кодом 200.



Далее мы отправляем запрос на установку куки. Запрос и результат его выполнения предоставлены на рисунке ниже. Как мы можем видеть, мы получили ответ с кодом состояния 302 – код, сигнализирующий о том, что запрошенный ресурс временно перемещен на другую страницу. Результат выполнения запроса – перенаправление на другую страницу.



Далее нам нужно посмотреть список установленных куки. Для этого нам необходимо отправить соответствующий запрос. Текст запроса и результат его выполнения – на рисунке ниже. Как мы можем видеть, запрос был выполнен успешно, в результате запроса нам был передан ответ в формате json, из которого мы можем заключить, что список куки пуст.



Далее нам нужно отправить запрос на перенаправление. Запрос и результат его выполнения представлены на рисунке ниже. Результат выполнения говорит нам о том, что ресурс не найден. Это мы можем заключить исходя из кода состояния 404.

