**APPLIED CRYPTOGRAPHY**

NAME: VISHWAS M

SRN: PES2UG20CS390

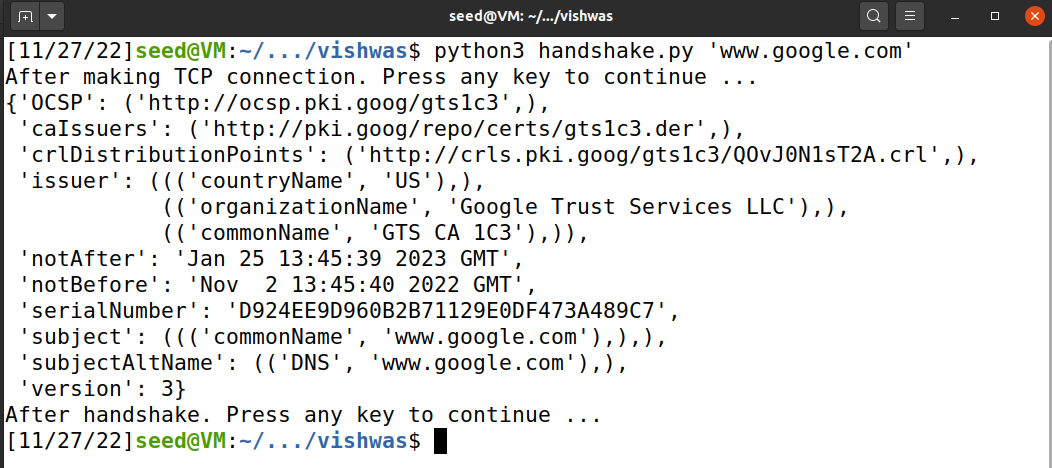
SEC: F

LAB: TLS Lab

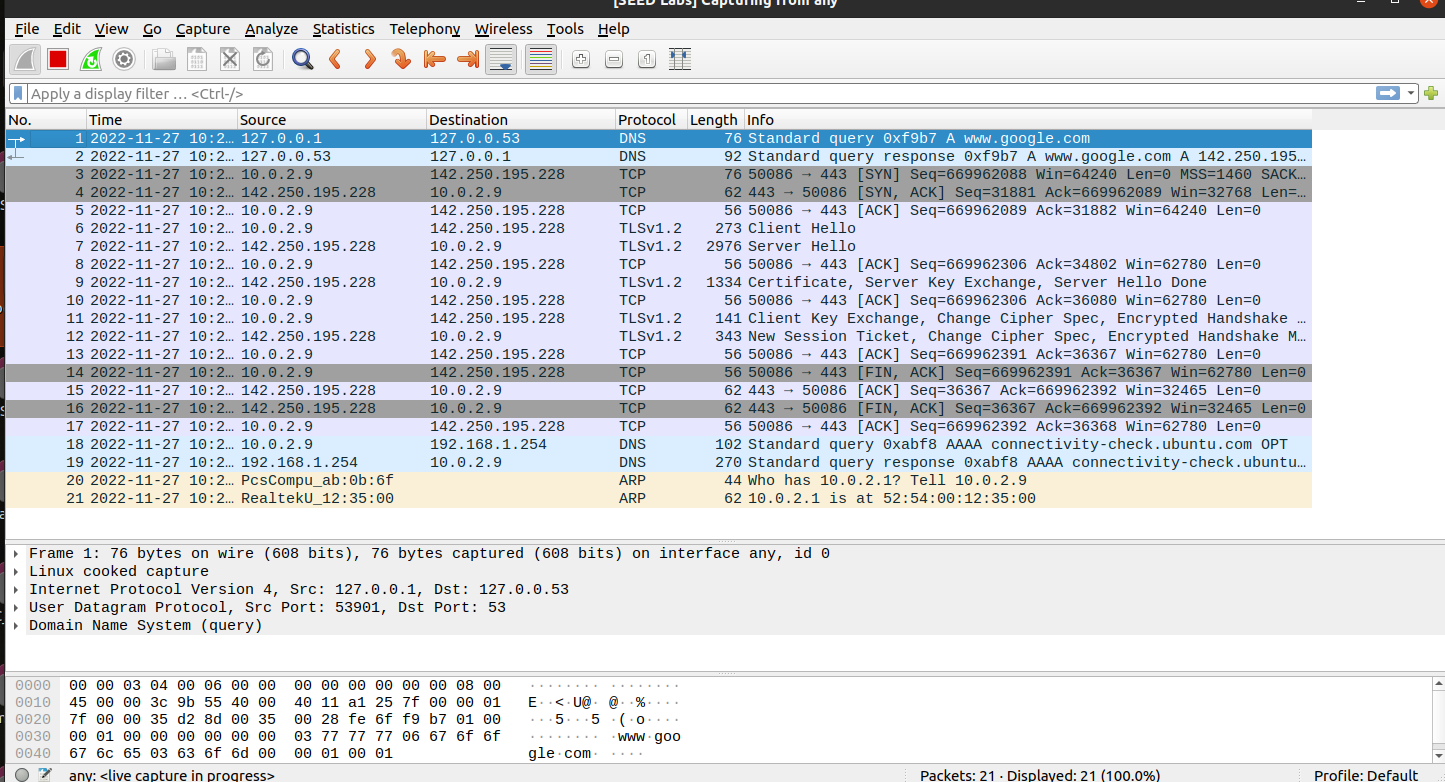
Task 1:

Task1.a:

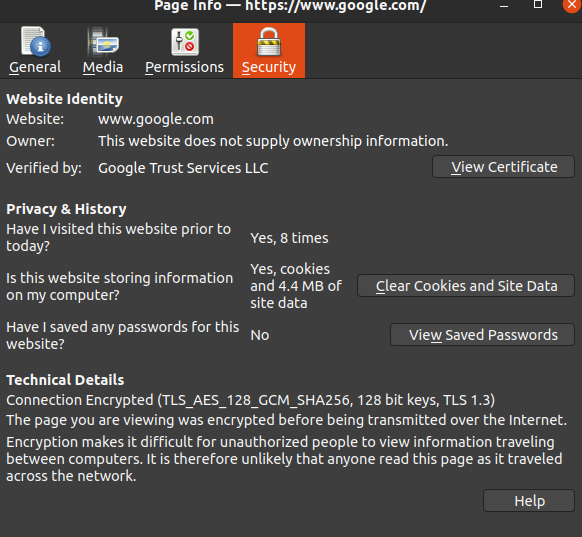
* Server Certificate from the program:



* Using Wireshark, we can see how TCP and TLS handshake are done. After TCP handshake, TLS handshake takes place and the certificates are shared between the client and the server. Then a new session key is shared between the client and server. The client will encrypt the messages and send those messages to server. Then server will decrypt the messages using the same shared key. After communication between client and server is finished TLS connection is closed and then TCP connection is closed.
* The Client initiates the TLS handshake by sending “Client Hello“ message as we can see in the below screenshot.



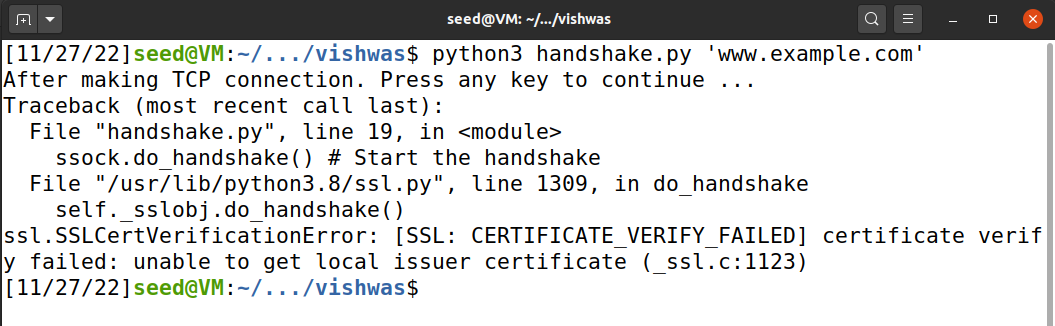
* We can see that AES encryption is used from the below screenshot.



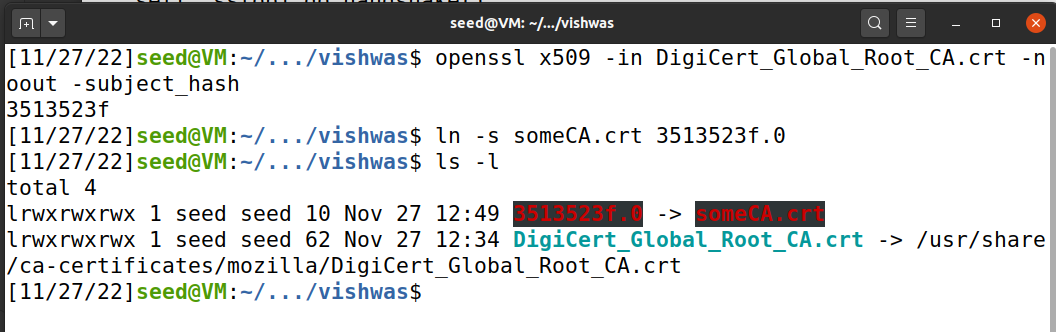
* “/etc/ssl/certs” is the directory of a folder which contains the list of all the certificates

Task 1.b: CA’s Certificate

After changing the cadir directory to ‘/home/seed/Desktop/certs/vishwas’, we get this error:

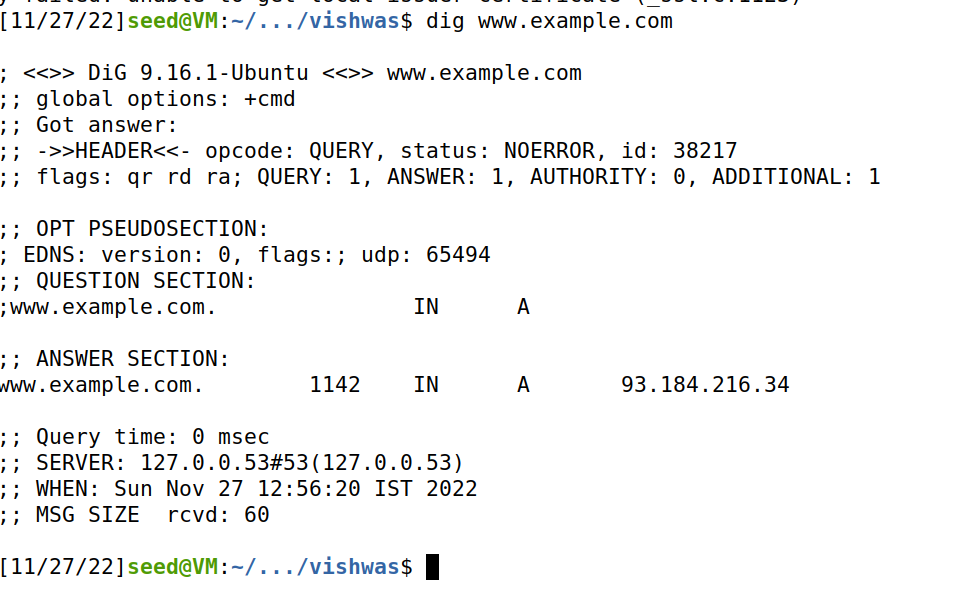


We have to change the name of the file that we copied:



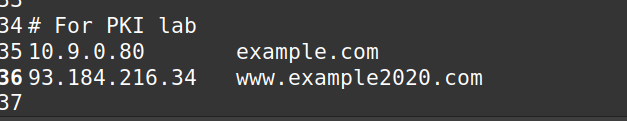
Task 1.c: Experiment with the hostname check

Step1: dig cmd



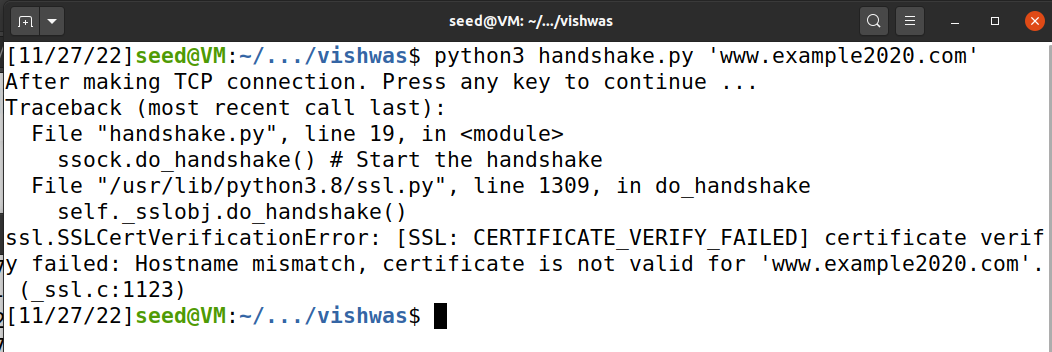
Step2:

Modifying /etc/hosts file

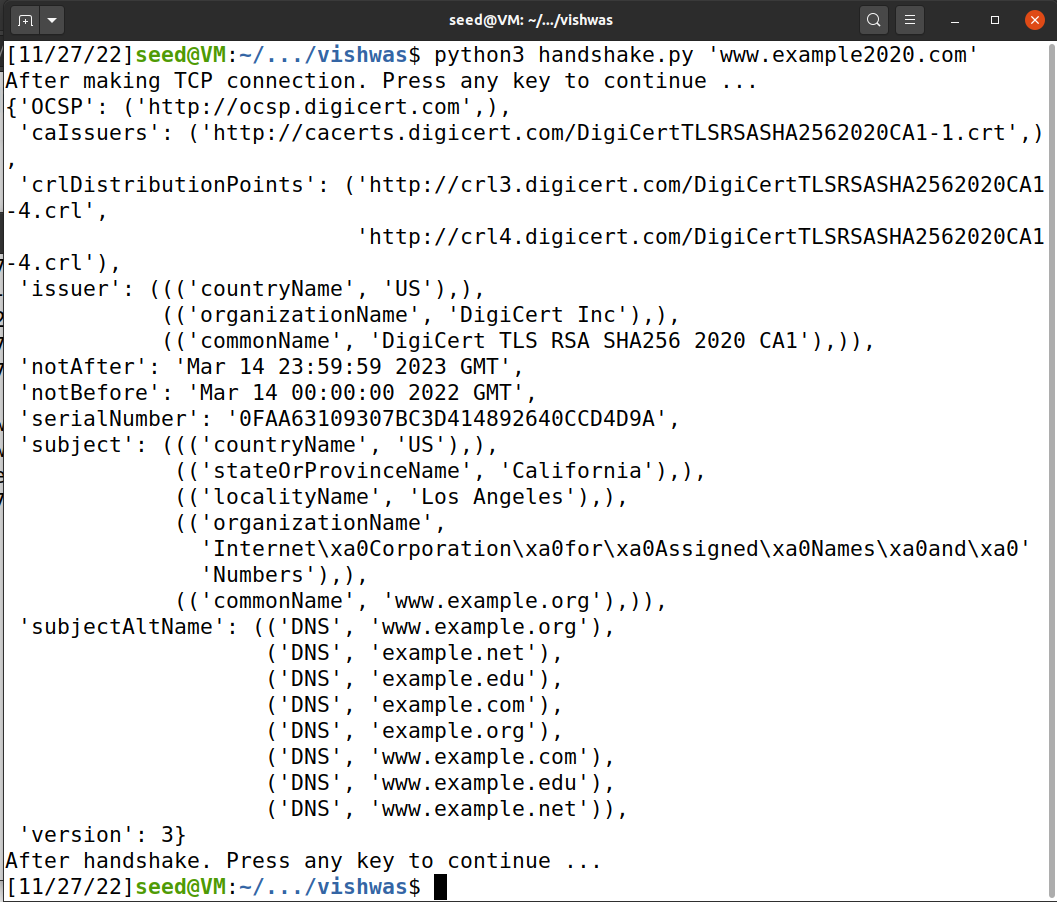


Step3:

When we change ‘Context.check\_hostname=True’, then:



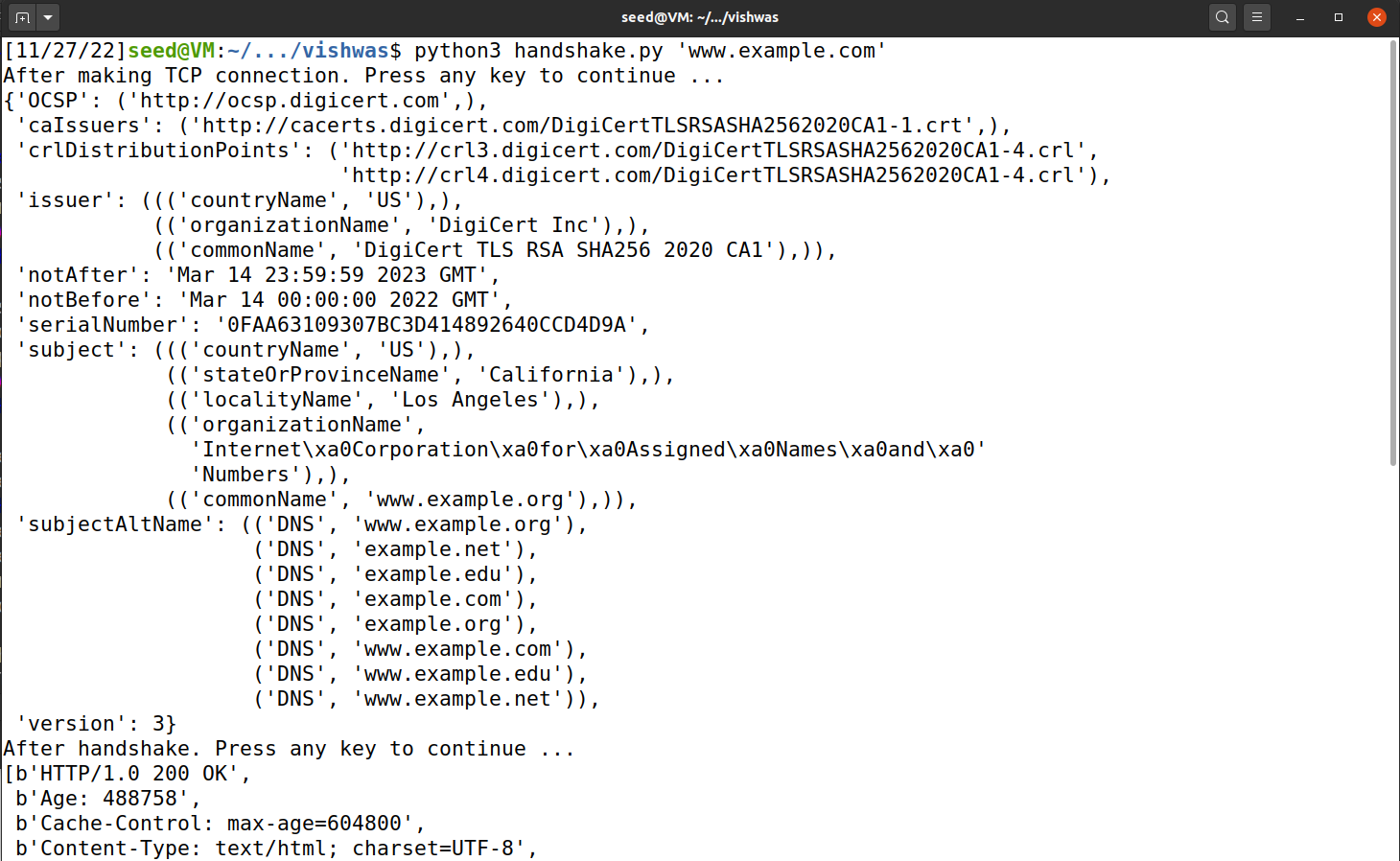
When we change ‘Context.check\_hostname=False’, then:

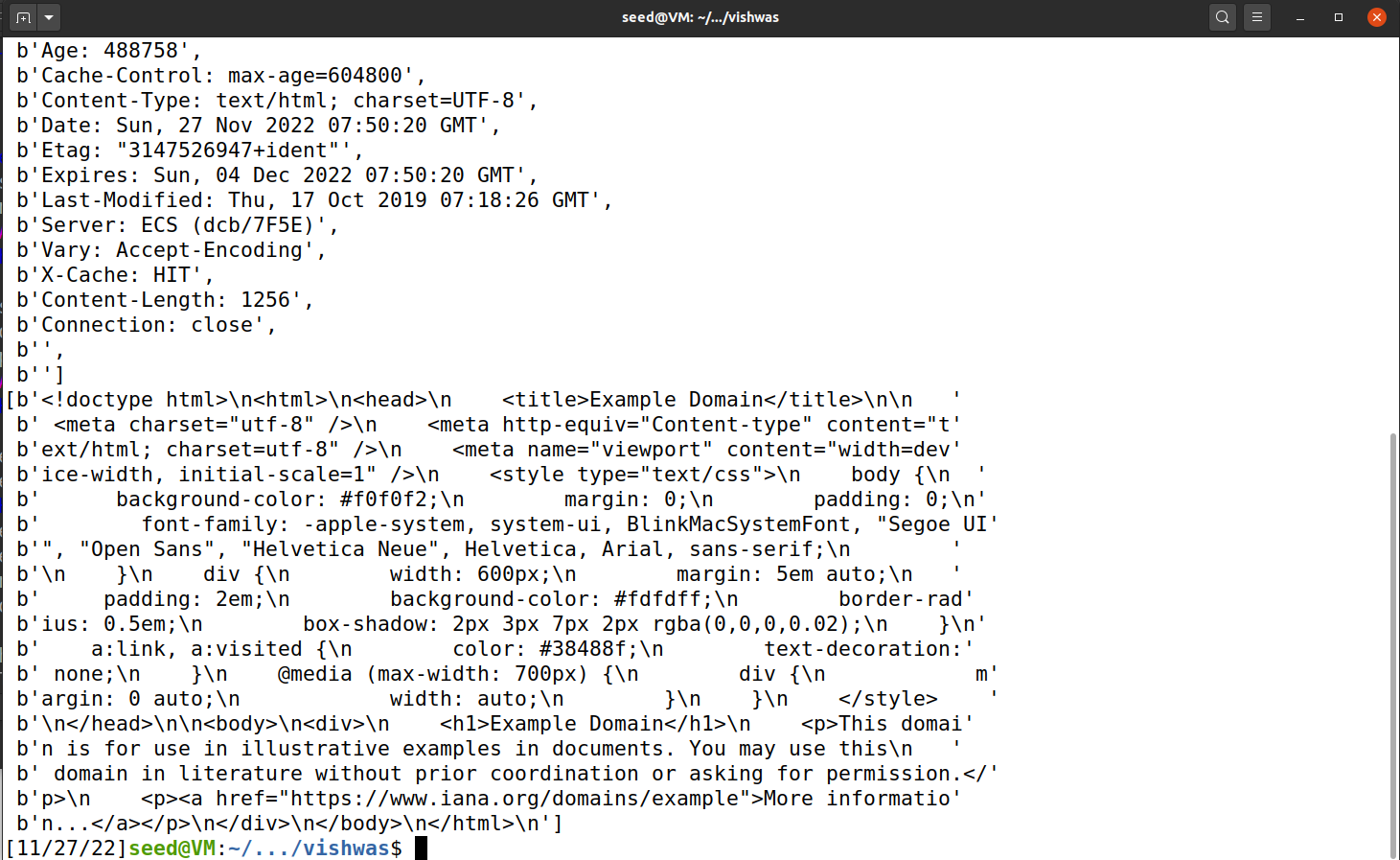


Based on the above observation we can conclude that it is necessary to check the hostname.

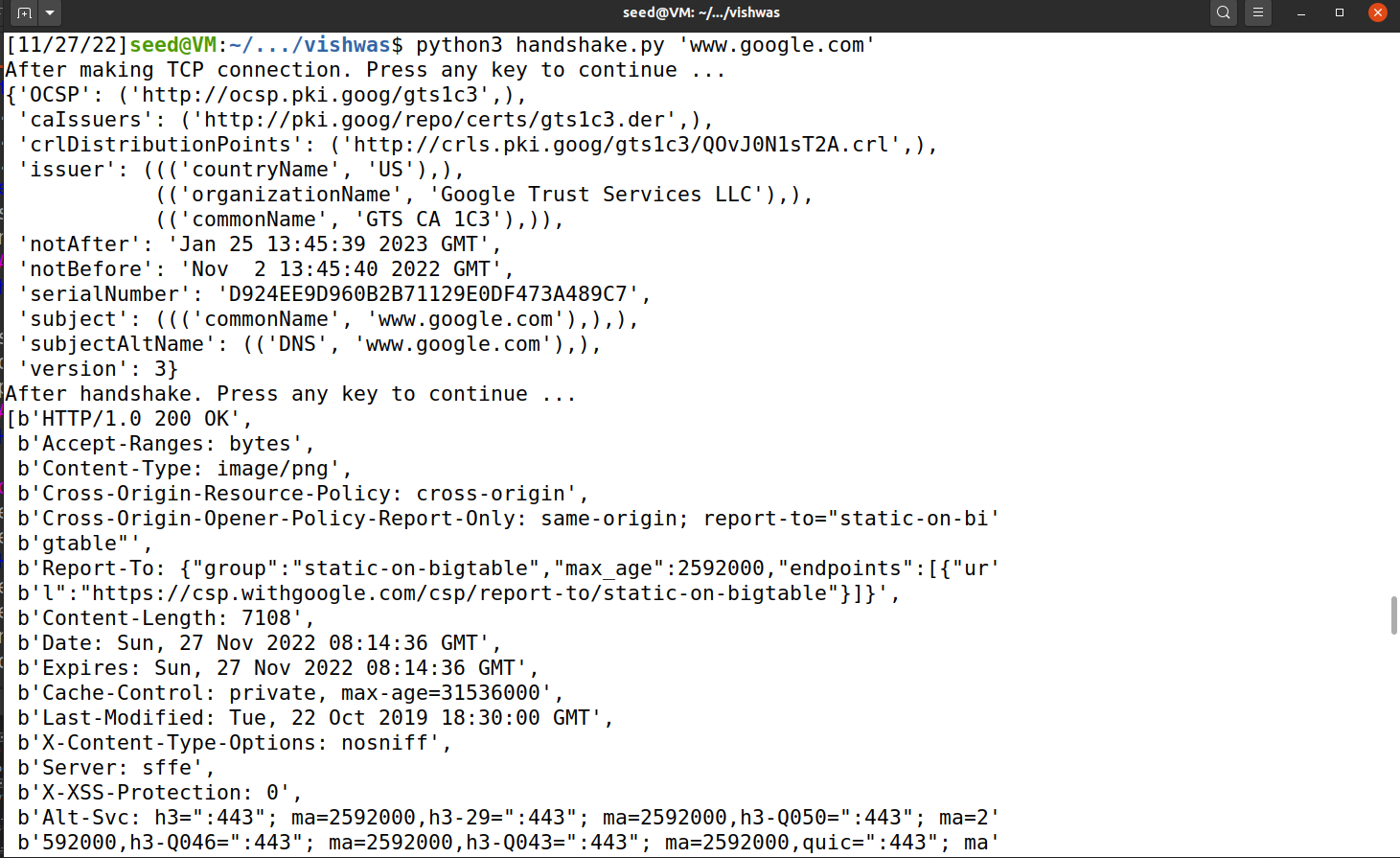
Task 1.d: Sending and Getting Data

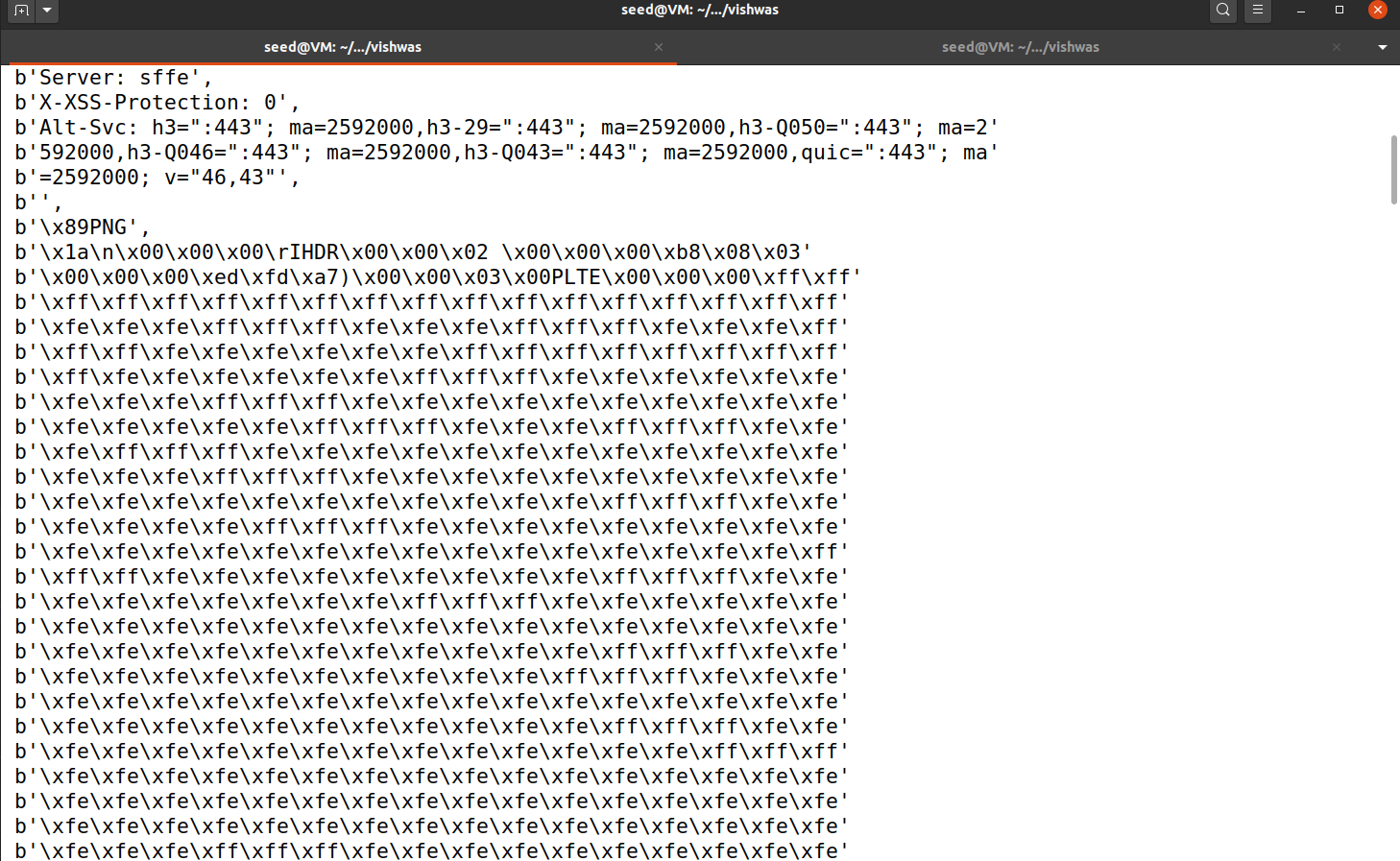
1. After adding the data sending/receiving code to client program.





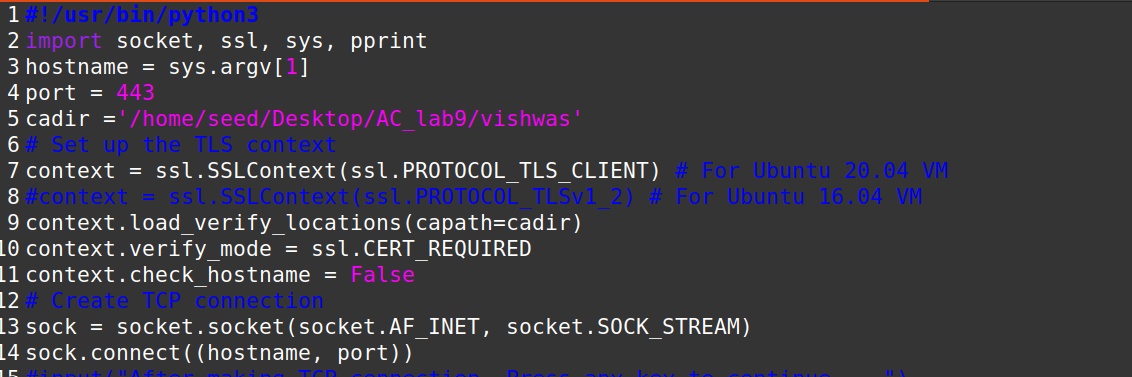
1. Modifying the HTTP GET request to get an image from www.google.com



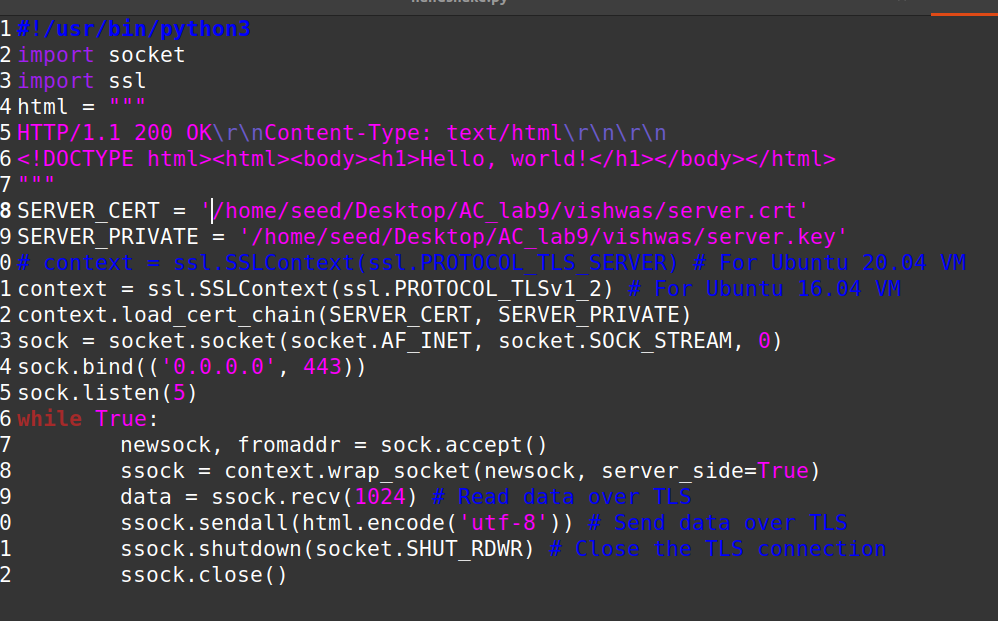


Task 2: TLS Server

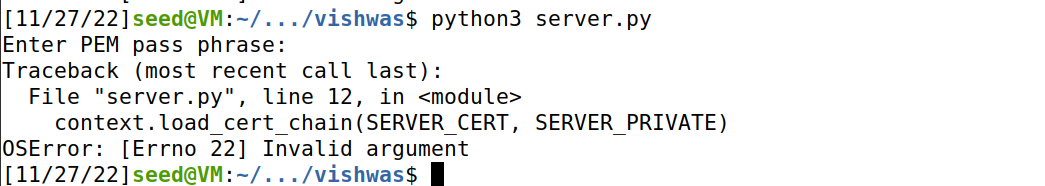
Handshake.py:



Server.py:

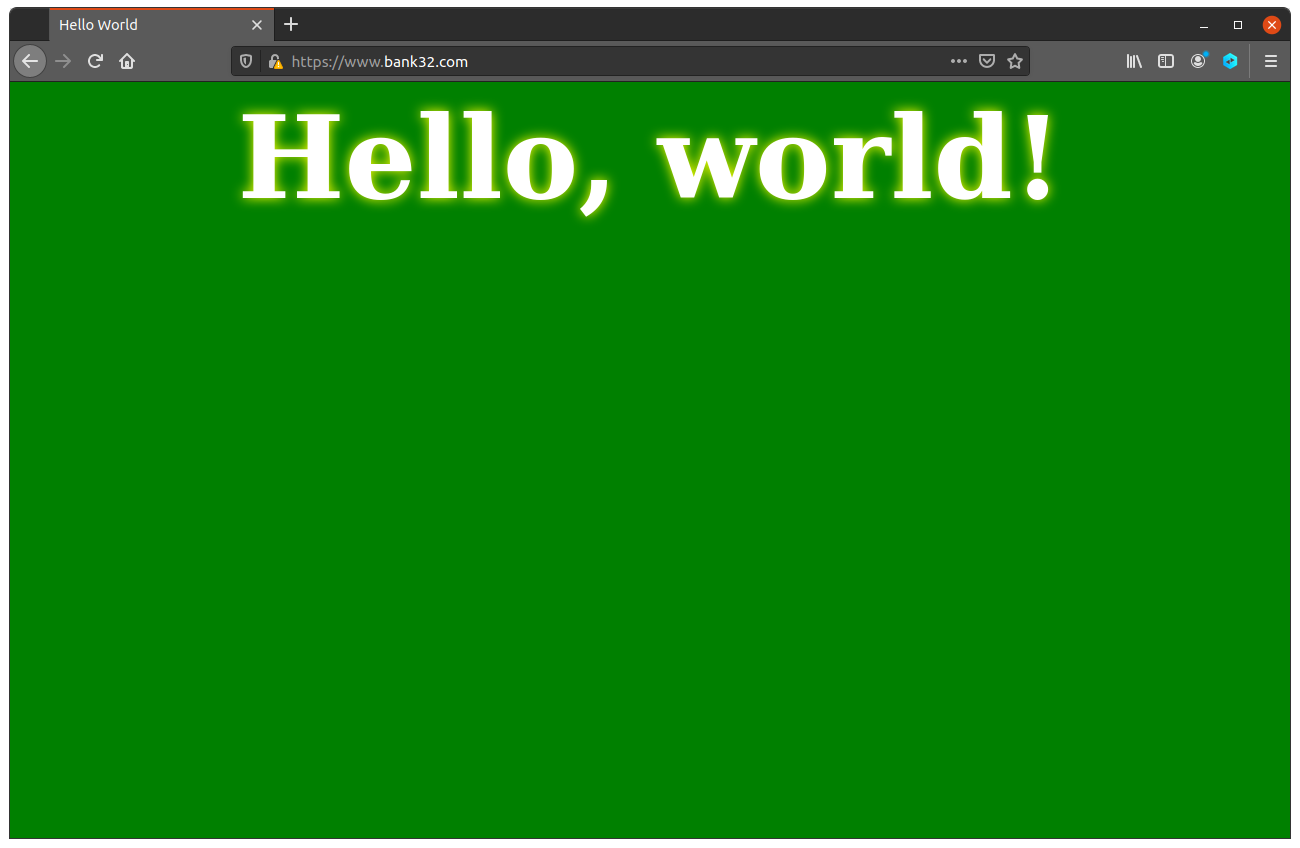


In the process of debugging the server.py program, we may encounter the following situations:



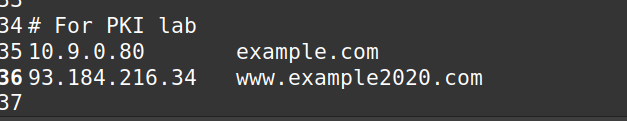
Task 2.b: Testing the server program using browsers

Since in the previous lab experiment, the root CA certificate created by yourself has been installed in the browser, you can directly test it. After running the server.py program, you will get the following results in the browser:

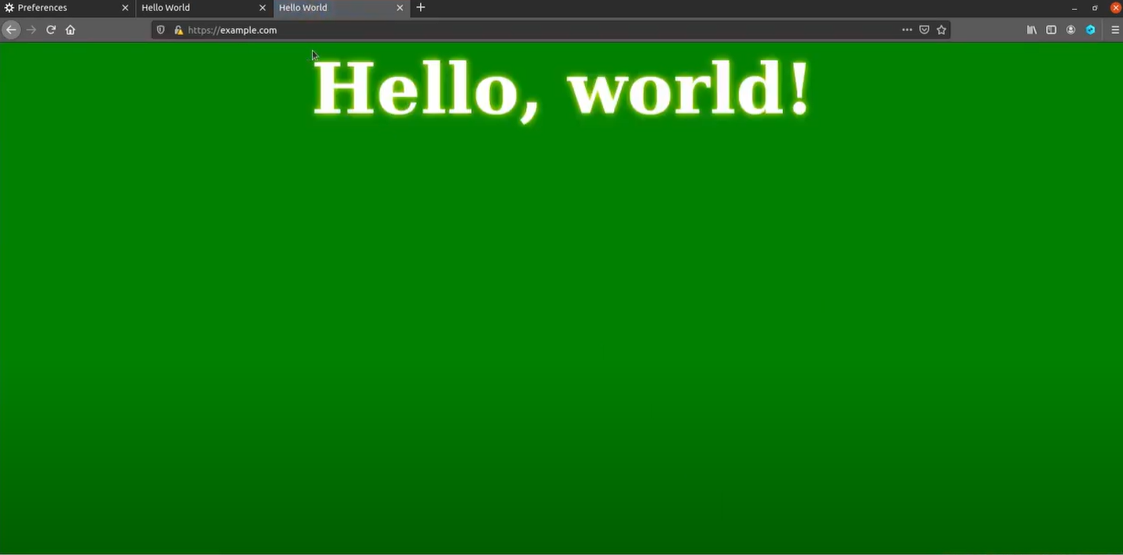


Task 2.c: Certificate with multiple names

Modifying the /etc/hosts file

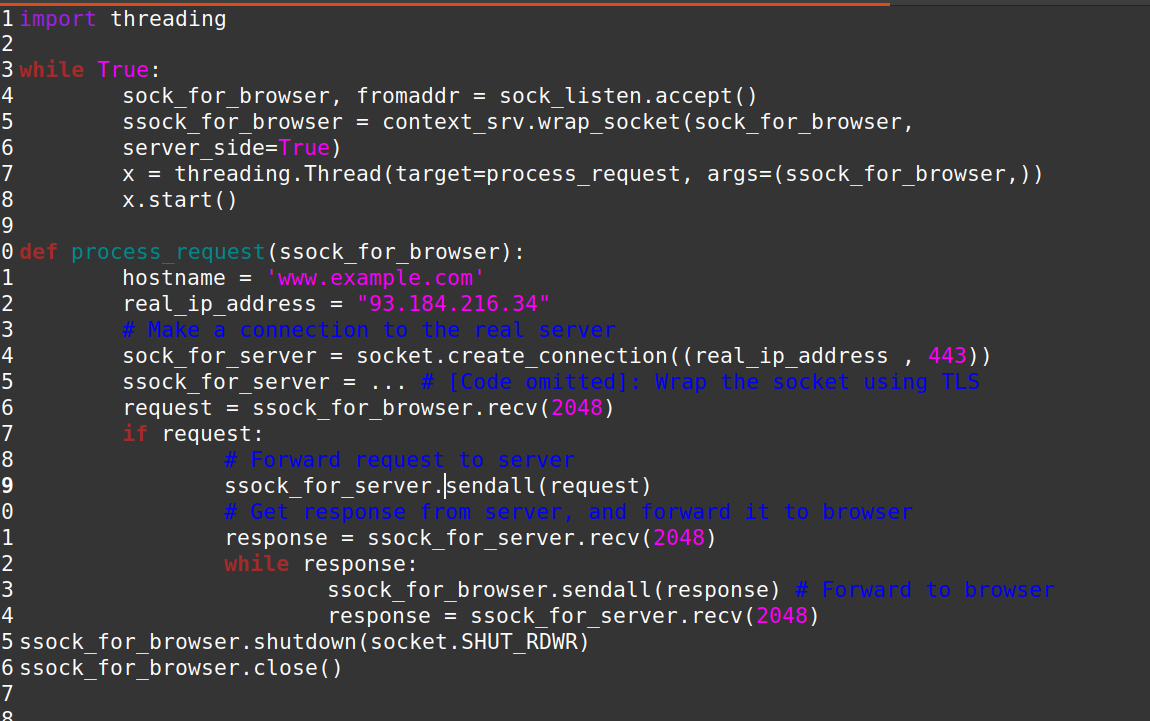


Checking in browser:

Since we have imported the private ca.crt certificate in the browser, and the server certificate we generated is issued by this ca.crt, it will pass the verification:

Of course, we can also test on other hosts, just need to configure a static IP in the etc/hosts file to make an IP modification.

Task 3: A Simple HTTPS Proxy



Since the hostname www.example.com is already mapped to the localhost, we cannot use this name

in the proxy code to connect to the real web server. We will get the IP address of this domain, and directly

use the IP address in our proxy code, instead of using the hostname.