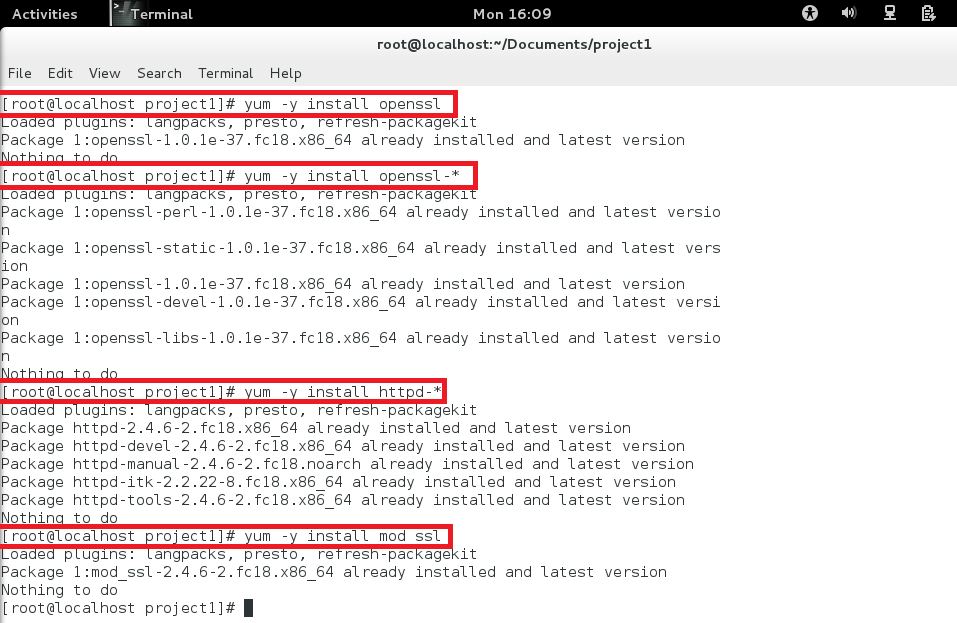
**Project 1: Public Key, Root CA, Certificates, Hash, MAC and SSL VPNs**

**TCSS 431: Network Security**

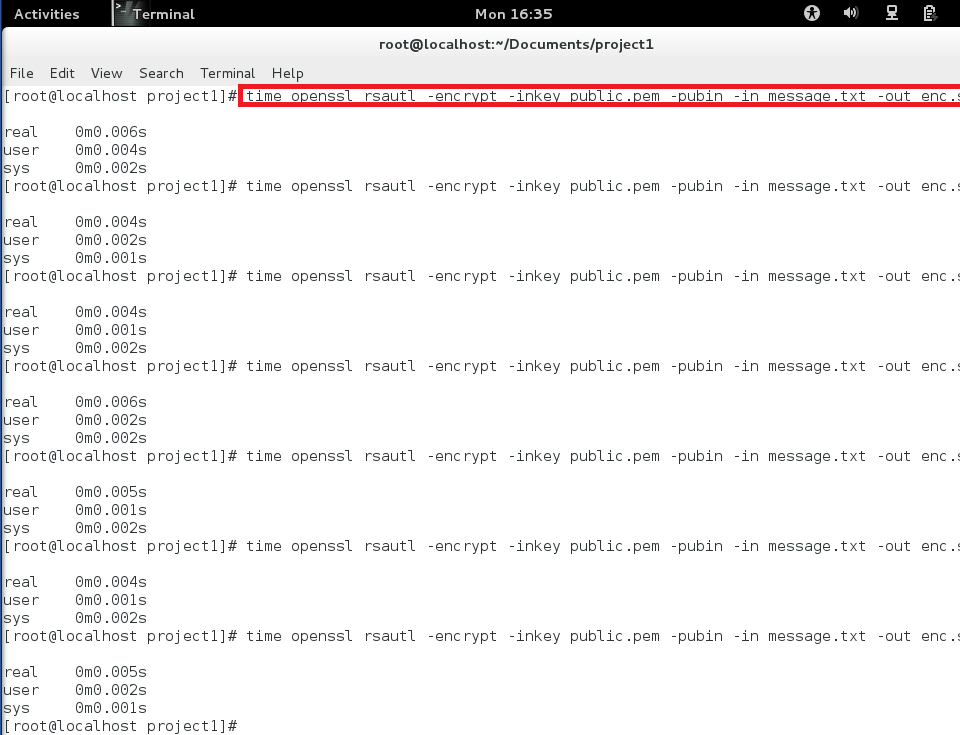
**Professor Wei Cheng**

**By Eduard Klimenko & Kyle Beveridge**

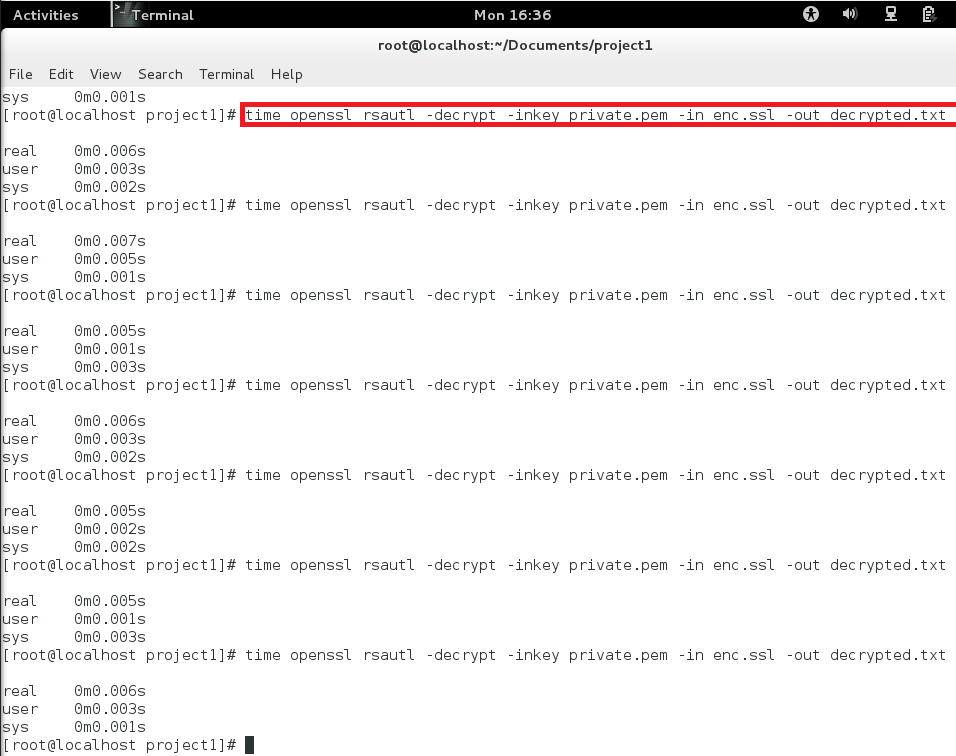
**Lab Preparation (2 points)**

****

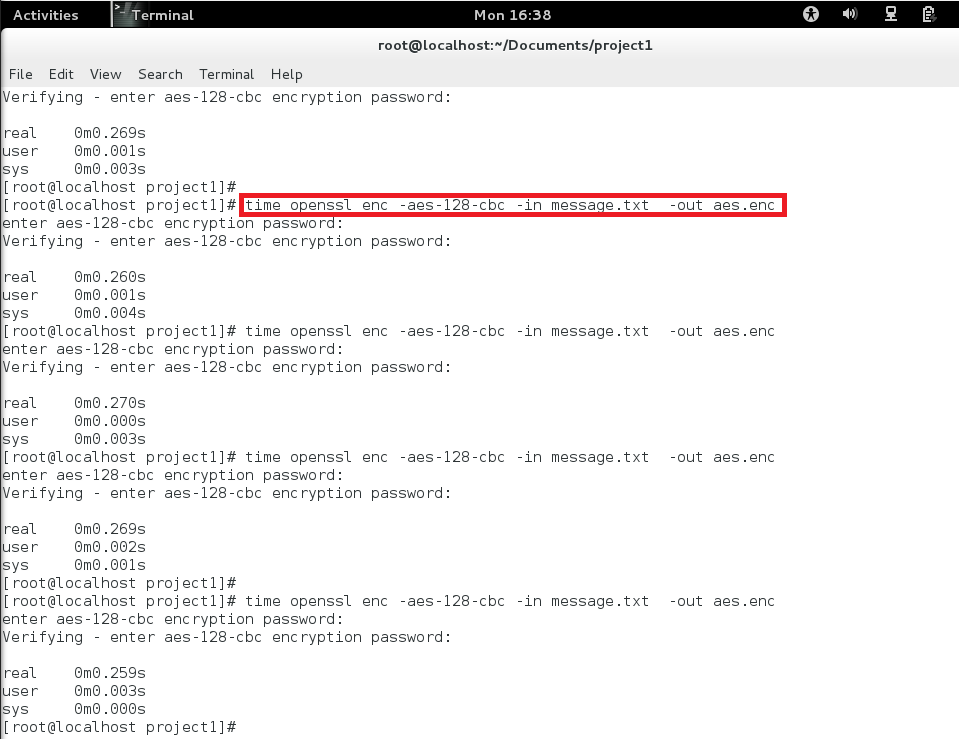
**Encrypt message.txt using the public key; save the output in message enc.ssl. (2 points)**

****

**2. Decrypt message enc.ssl using the private key. (2 points)**

****

**3. Encrypt message.txt using a 128-bit AES key. (2 points)**

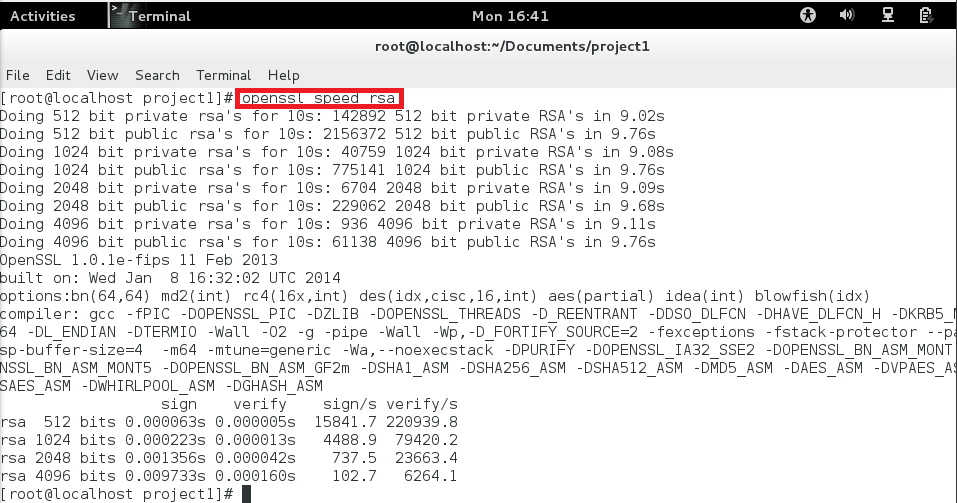
****

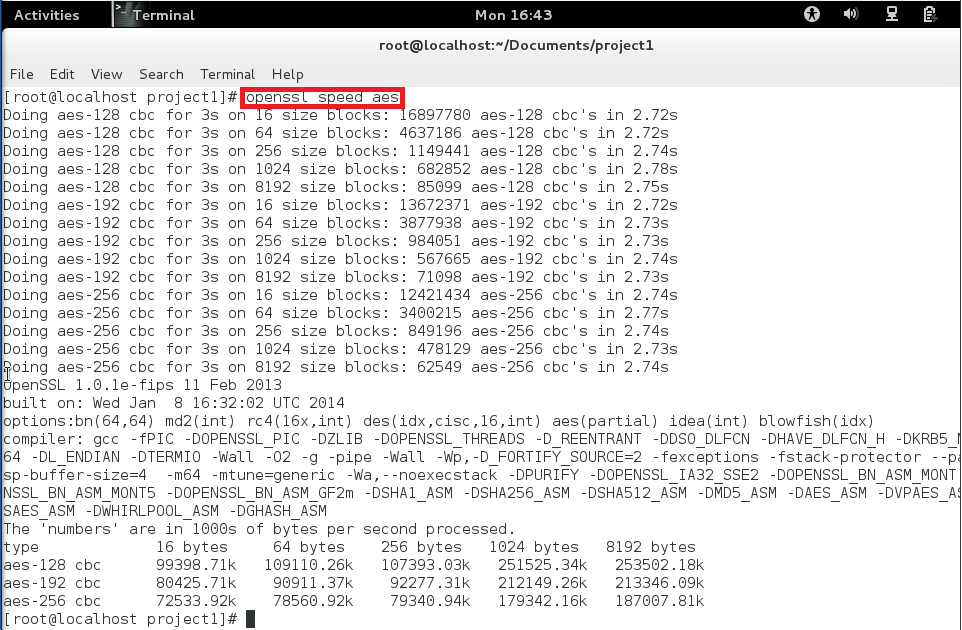
**Compare the time spent on each of the above operations, and describe your observations. If an operation is too fast, you may want to repeat it for many times, and then take an average. (5 points)**

Both RSA encryption and decryption took almost no time (3.4ms and 4ms respectively). The AES encryption took more time because the command required user input. However, running the speed tests with RSA and AES, we can clearly see that AES is faster.

**Please describe whether your observations are similar to those from the outputs of the speed command. (2 points)**

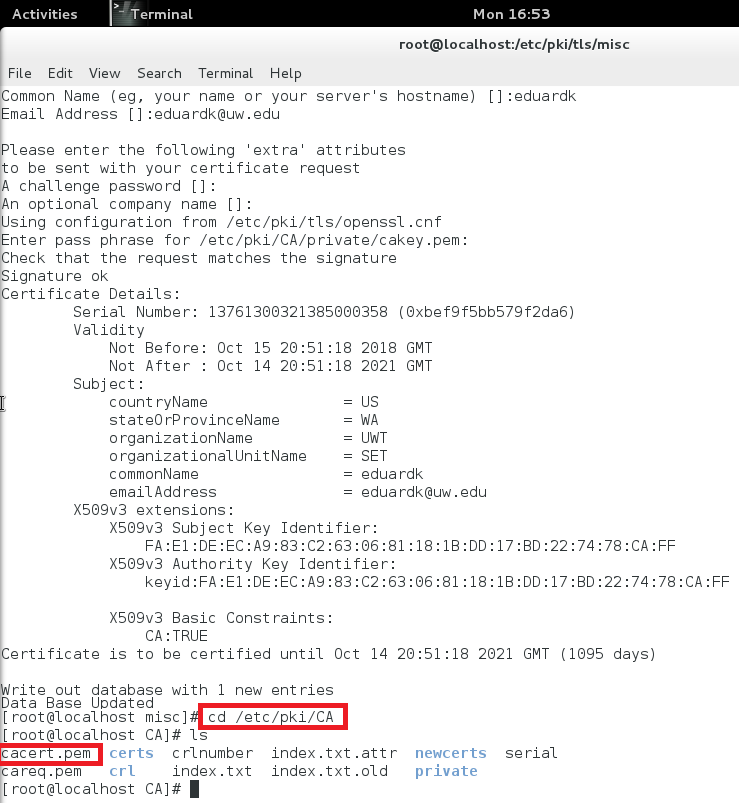
Aes seemed faster when we were encrypting the file, and the results below confirm that it is indeed much faster than RSA.

****

****

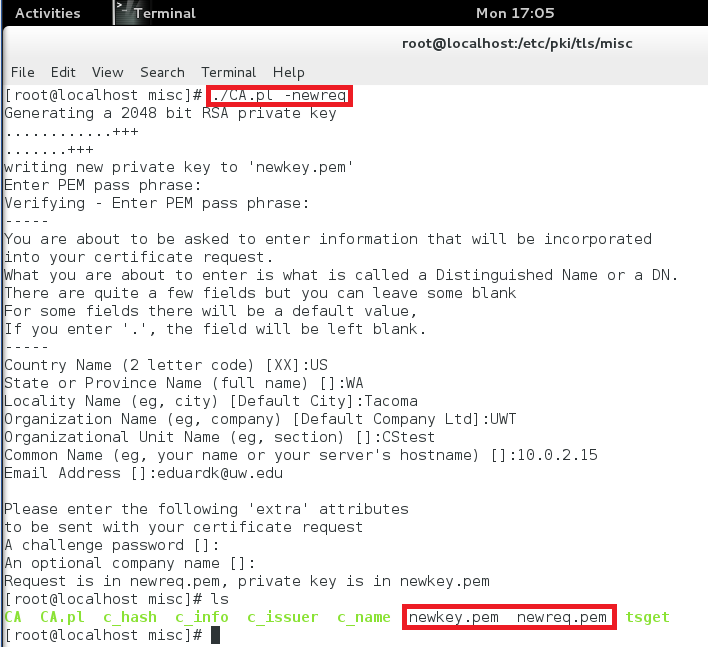
**Create a new CA ./CA.pl -newca**

**This creates a new root certificate in the directory */etc/pki/CA*. The new root certificate file is named as *cacert.pem*. (3 points)**

****

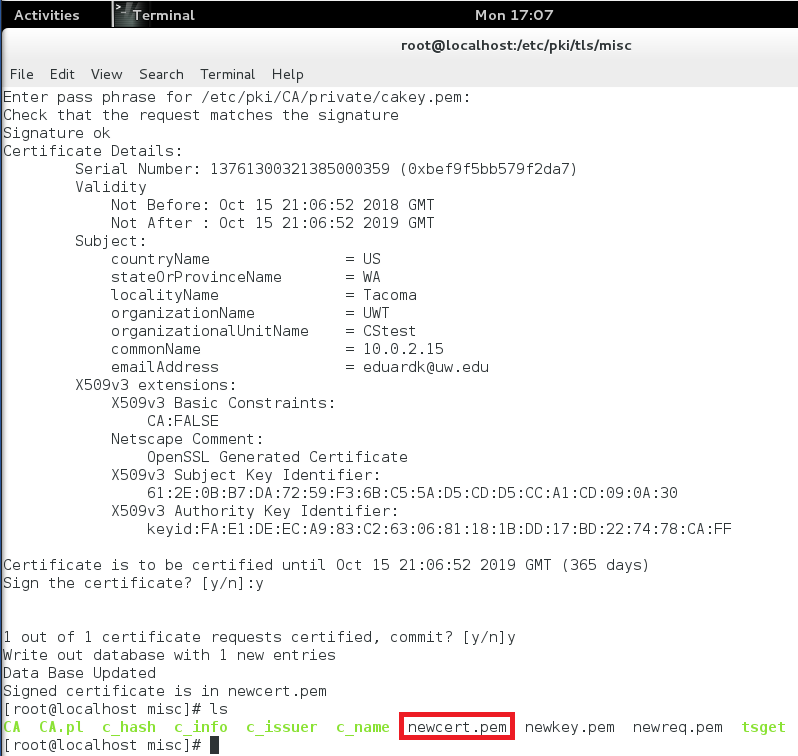
**Generate a new certificate request ./CA.pl -newreq**

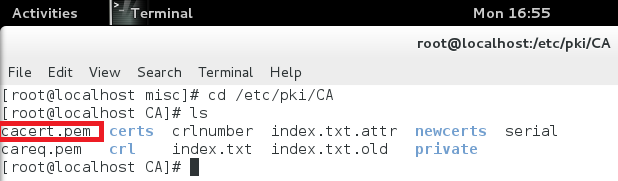
**The generated certificate request and private key in are stored newreq.pem and newkey.pem, respectively. (3 points)**

****

**Sign the certificate request ./CA.pl -sign**

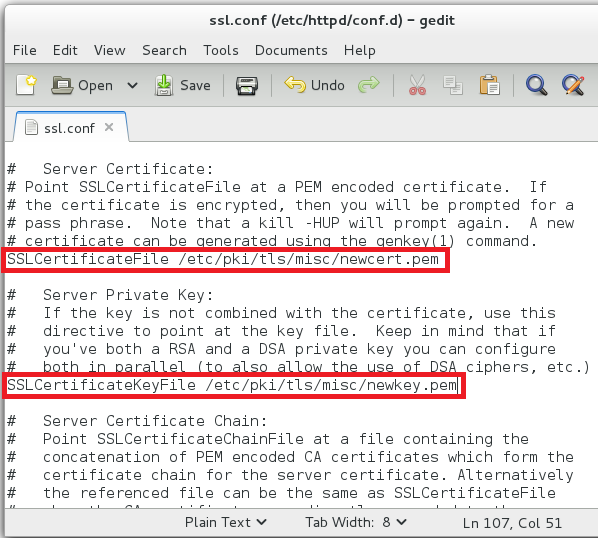
**Now, under the */etc/pki/tls/misc/* directory, there will be the certificate, newcert.pem, and the private key newreq.pem (encrypted) and newkey.pem (unencrypted). In addition, the root certificate file (named as *cacert.pem*) is located in the */etc/pki/CA* directory. (6 points)**

****

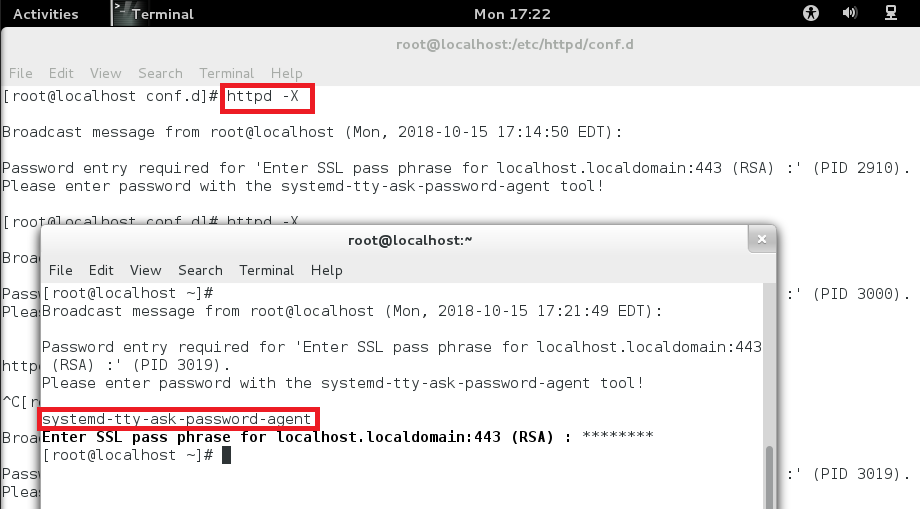
****

**Modify the Apache server configuration**

**On the FC Linux server, go the /etc/httpd/conf.d directory. There is a file named ssl.conf. edit this file and modify the two places: (4 points)**

****

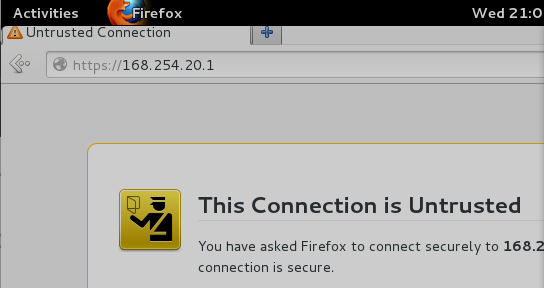
**Were asked to provide a key? Which key should you provide? (4 points)**

****

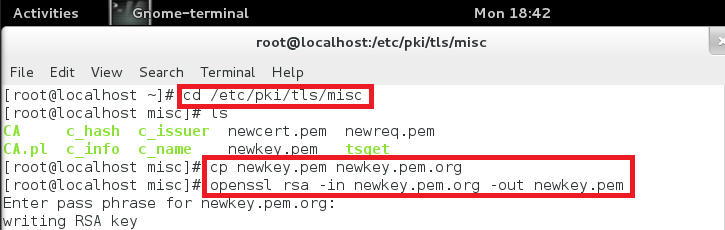
**Test an SSL/TLS connection Open a browser, type the** [**https://server\_IP**](https://server_ip)**.**

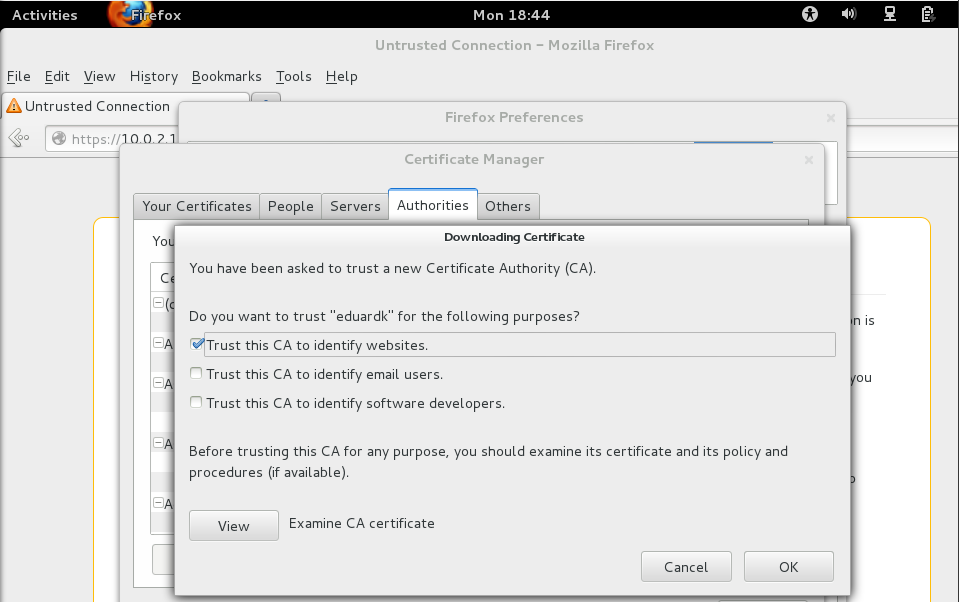
**Did you receive a warning message? (2 points)**

Yes, we got a warning stating that the connection was untrustworthy.

****

**Install the root CA for Firefox on the Linux server (5 points)**

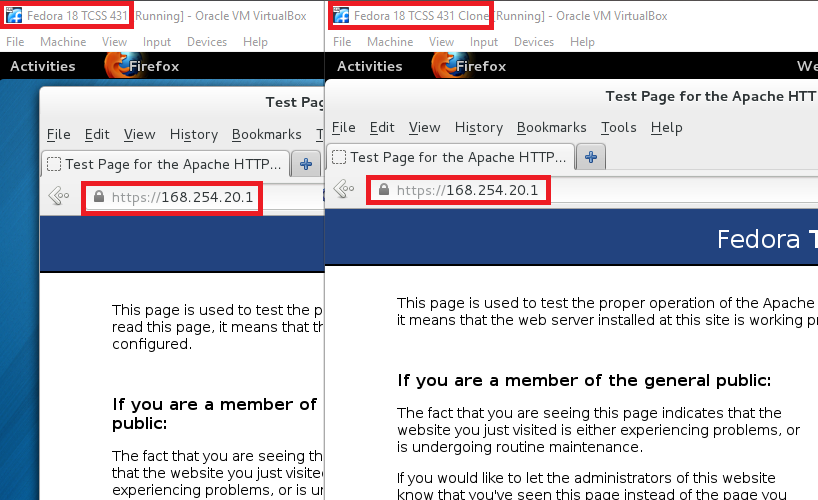
****

****

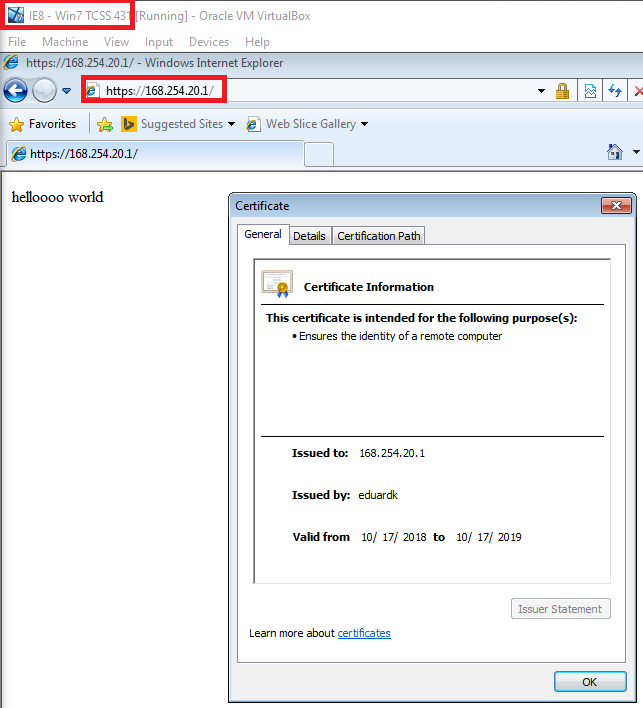
**Explain why you did not receive the warning message? (2 points)**

We installed the server’s CA into our browser, so when we connected to the server, the browser had our CA so it didn’t see the site as untrustworthy.

**Install the root CA for Firefox on a Linux client (5 points)**

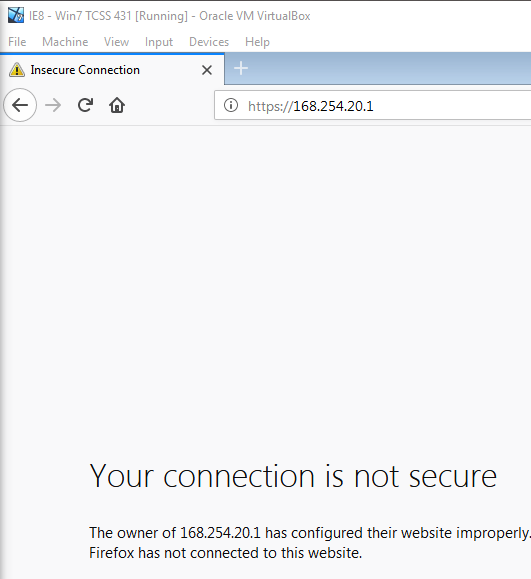
****

**Once you have downloaded the certificate to the Windows machine, double click the certificate to install it. Follow the default option should be okay. (5 points)**

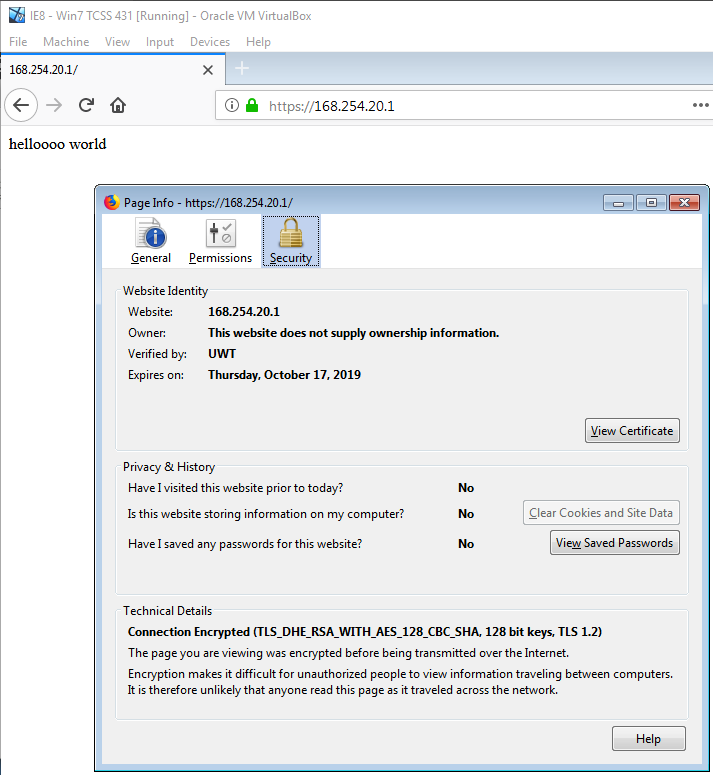
****

**Test an SSL/TLS connection using Firefox. Explain what happened. Why? (2 points)**

The connection was not secure as expected. We did not install the CA on firefox so it deemed the connection untrustworthy.

****

**Install the root CA for Firefox on Windows (3 points)**

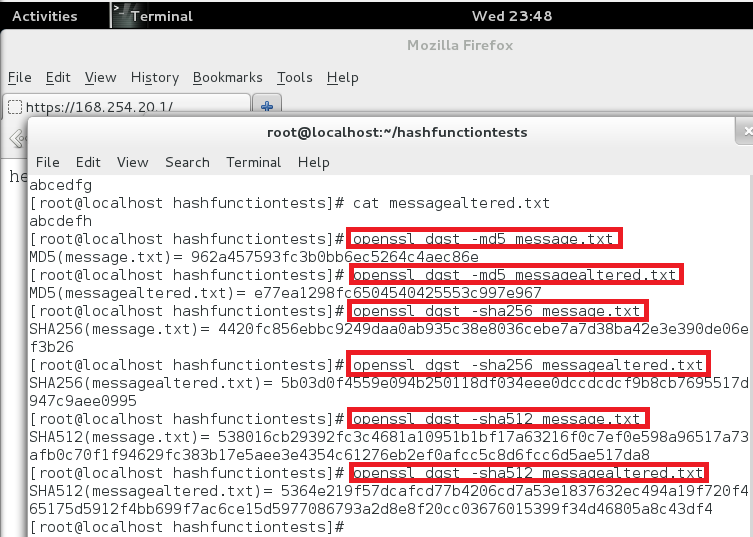
****

**Explain what happened. Why? (2 points)**

We tried connecting to the server on client machines using different browsers and operating systems. Initially, the connection was not secure because we did not install the root CA on the new systems. After we installed the CA, the site was verified. We also noticed that each browser needs its own CA (installing on IE doesn’t mean it will be secure on Firefox).

**Please observe whether the results are similar or not. Please describe your observations in the project report. (4 points)**

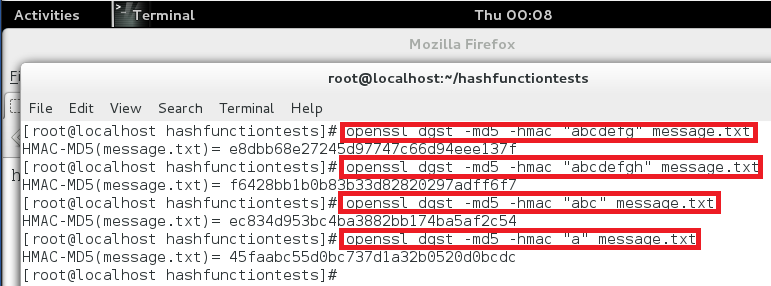
Small changes to message content seems to result in very different hashes. We only changed one letter but the output was very different.

****

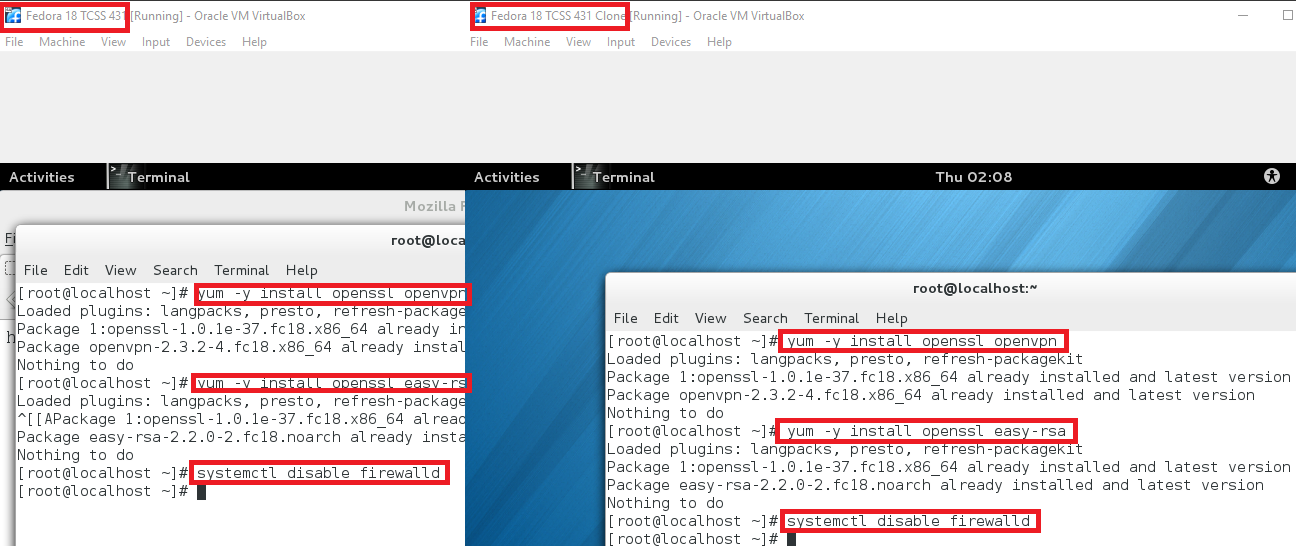
**Please generate a keyed hash using HMAC-MD5 and HMC-SHA512 for any file that you choose. Please try several keys with different length. (3 points)**

**Do we have to use a key with a fixed size in HMAC? If so, what is the key size? If not, why? (2 points)**

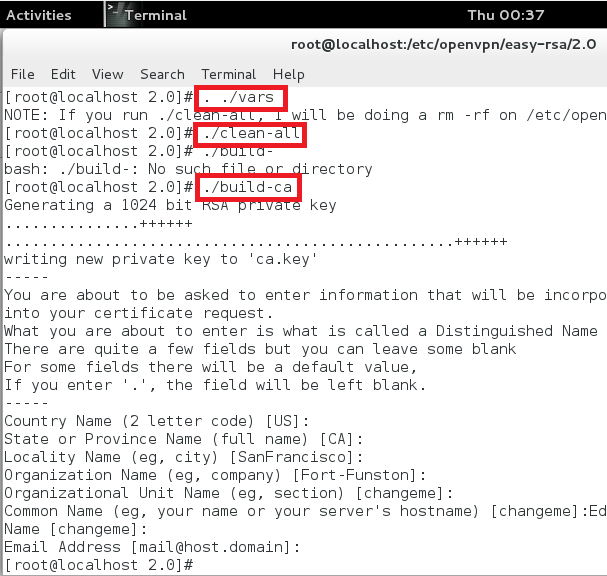
No fixed size is required, since HMAC maps any size key to a fixed output size (uses hashing algorithm).

****

**Determine your VPN server and client machines. On both FC server and FC client: (2 points)**

****

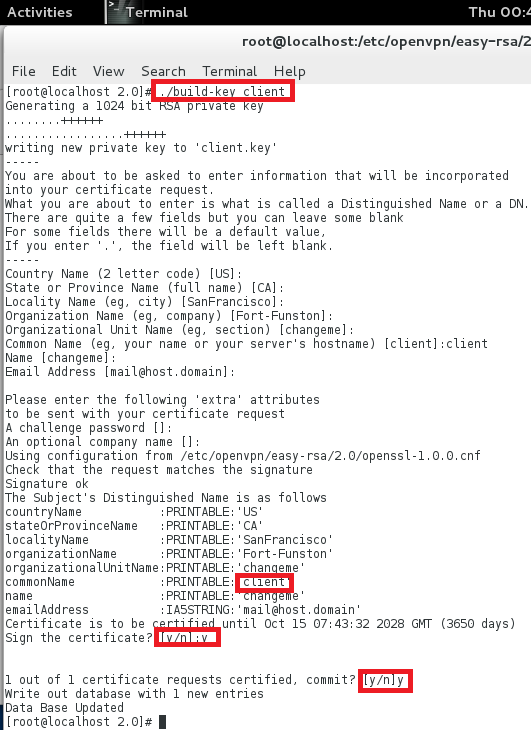
**Build Certificate Authority (CA) (2 points)**

****

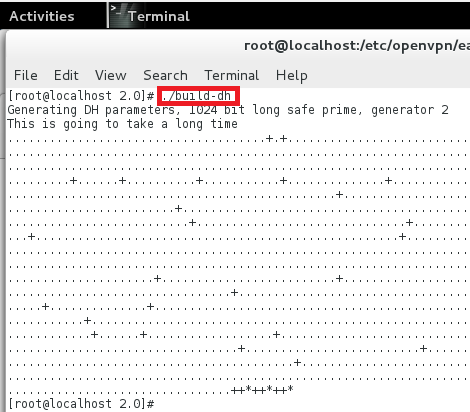
**We first will generate a certificate and private key for the server: (2 points)**

****

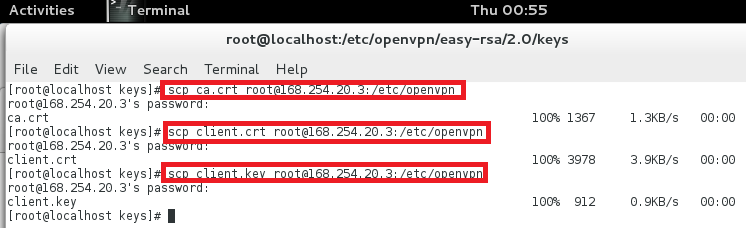
**Secondly, generate certificates & keys for a client. Generating client certificates is very similar to the previous step: (2 points)**

****

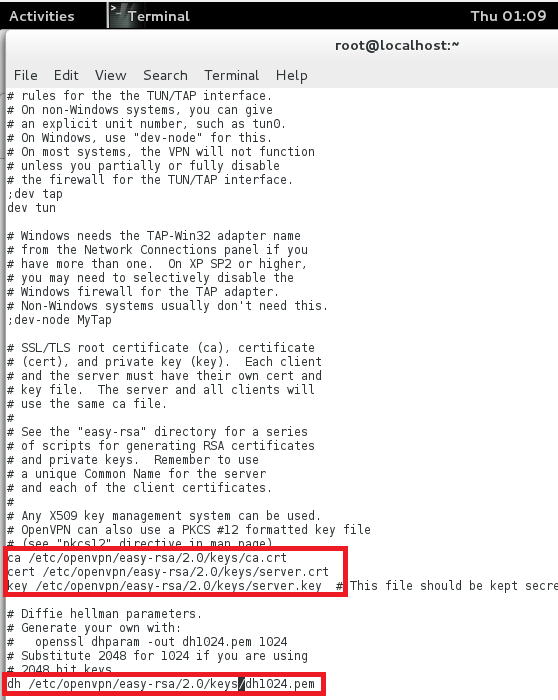
**Thirdly, generate Diffie Hellman parameters.** [**Diffie Hellman**](http://www.rsasecurity.com/rsalabs/node.asp?id=2248) **parameters must be generated for the OpenVPN server: (2 points)**

****

***scp ca.crt root@client\_IP\_address:/etc/openvpn* (2 points)**

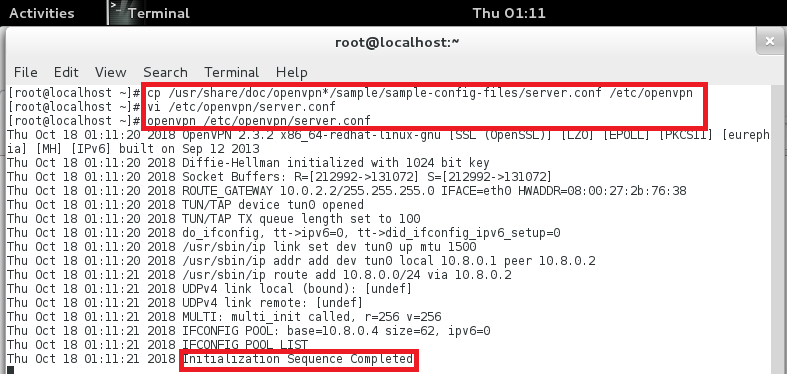
****

**You need to add the right path information for each of the 4 parameters (4 points)**

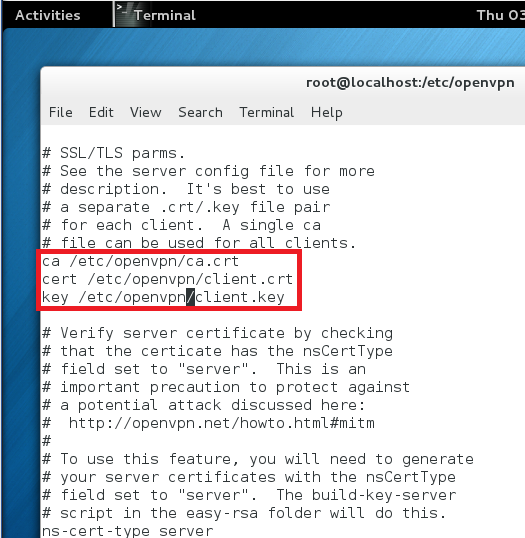
****

**Now, you can start the VPN server via: (2 points)**

***openvpn /etc/openvpn/server.conf***

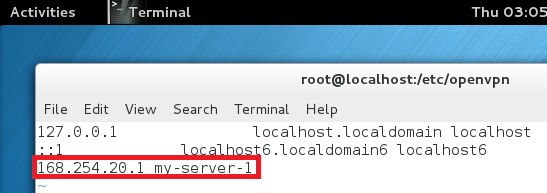
****

**Modify the following 3 parameters (these 3 files were copied from the server in the previous section 6.2): (4 points)**

****

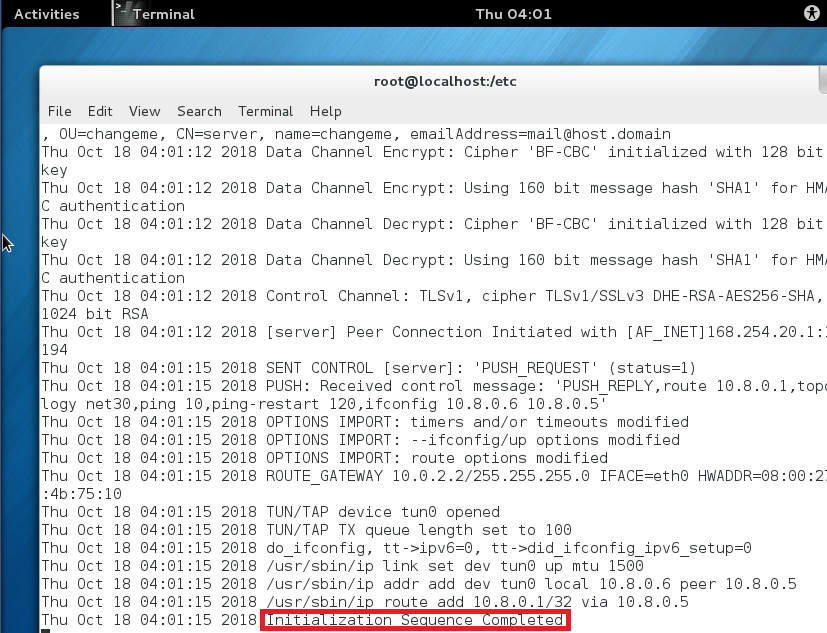
**You also need to modify the /etc/hosts file by adding one entry: (3 points)**

**IP\_address\_of\_your \_FC\_server my-server-1**

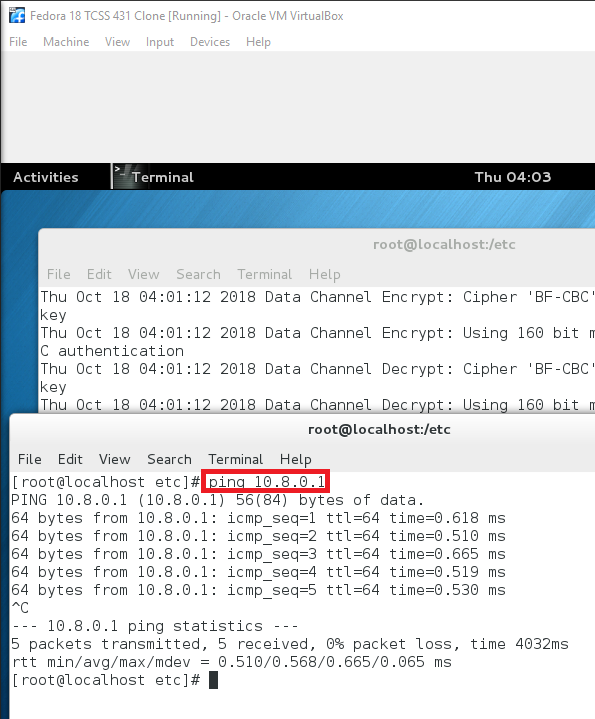
****

**Now, you can start the VPN client via:**

***openvpn /etc/openvpn/client.conf***

****

**If the ping succeeds, congratulations! You now have a functioning SSL based VPN. (3 points)**

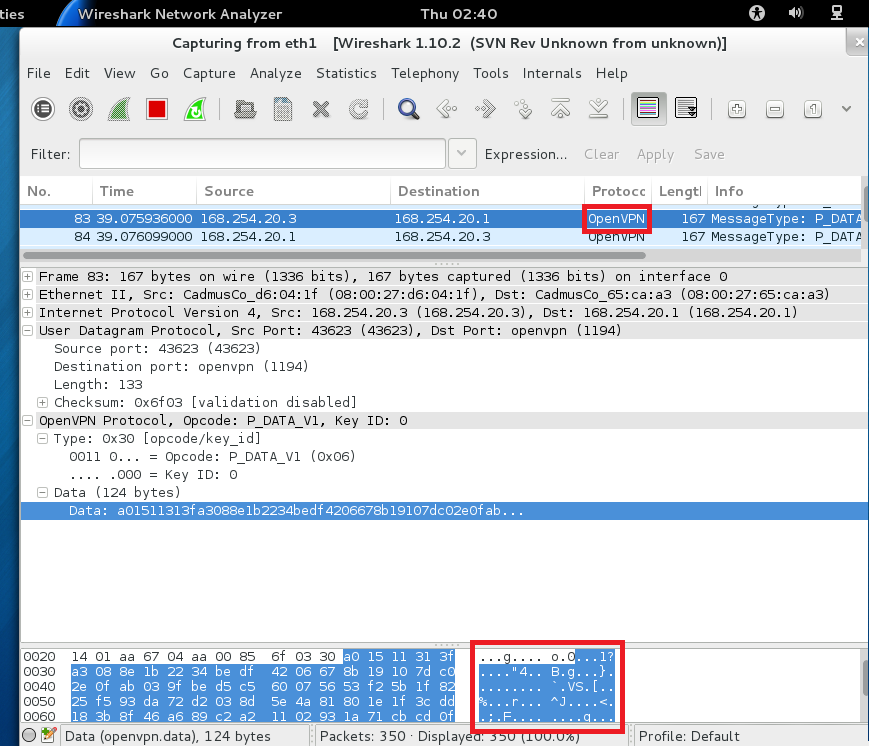
****

**Check whether the ping packets are encrypted. (2 points)**

They are encrypted because the data is not in plaintext.

**In your words, explain what does this setup do? (2 points)**

It creates a connection between a server and client using the OpenVPN protocol, where the contents of the messages is encrypted.

****