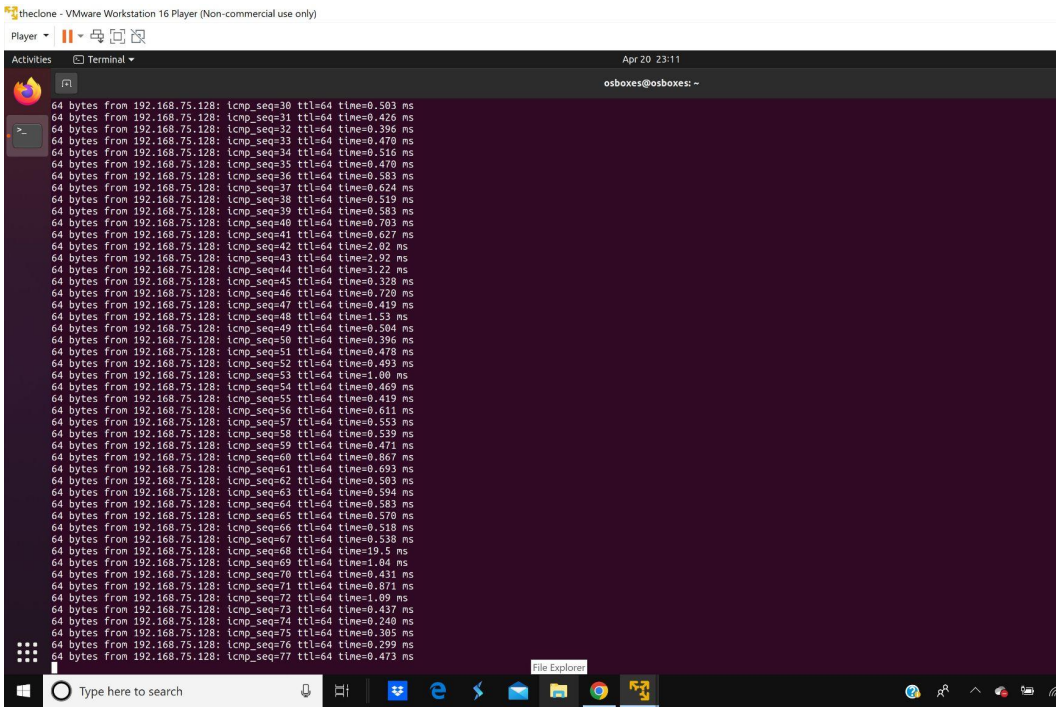
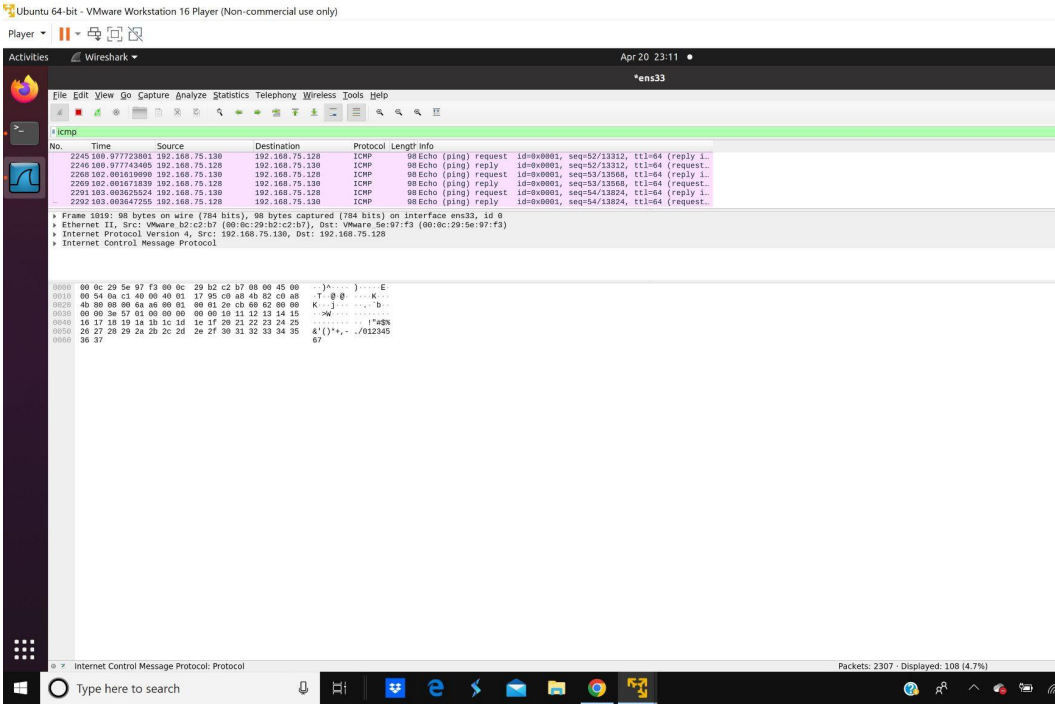
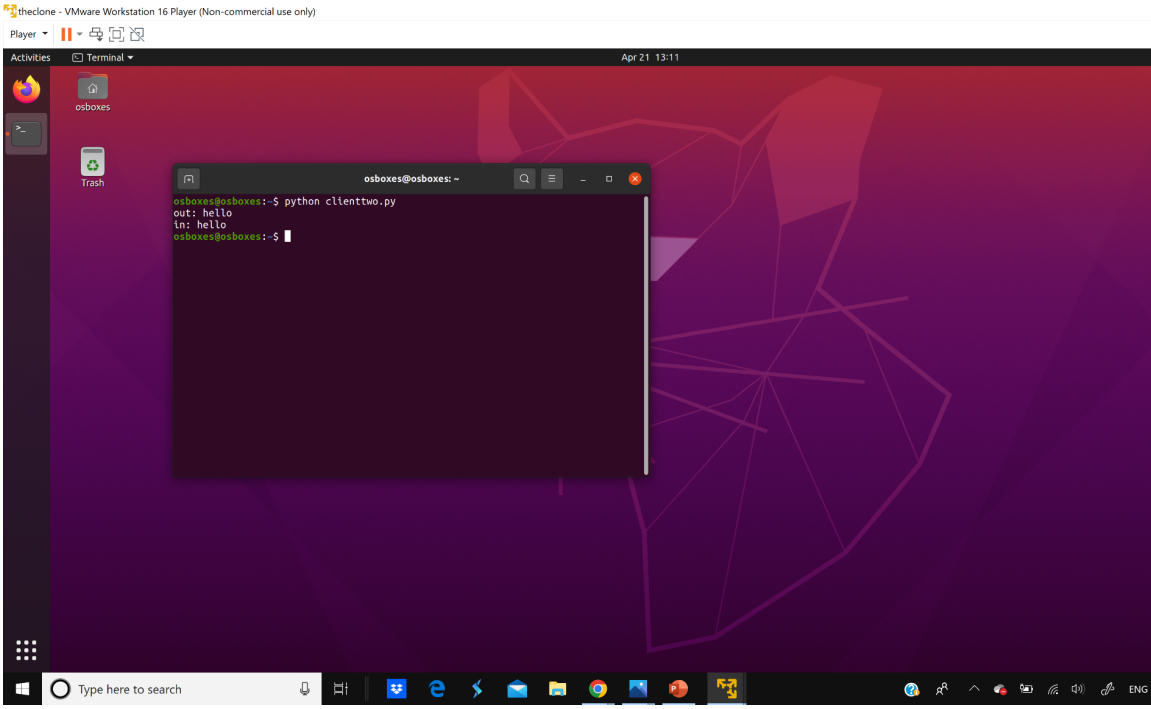
- The first thing I did was create another VM using the image we were given at the start of class. I initially tried to clone it but there wasn't an option for me.
- I then installed wireshark on my server VM using sudo apt update and sudo apt install wireshark
    - I confirmed that my nic was ens33 using ip addr show
    - then I confirmed that the two VMs had different IPs by going to wireshark and typing ICMP in the box and pinged from client to server
    - my original VM has an IP address of 192.168.75.128 while my clone one had 192.168.75.130
        - client perspective
        - 

        - wireshark server perspective

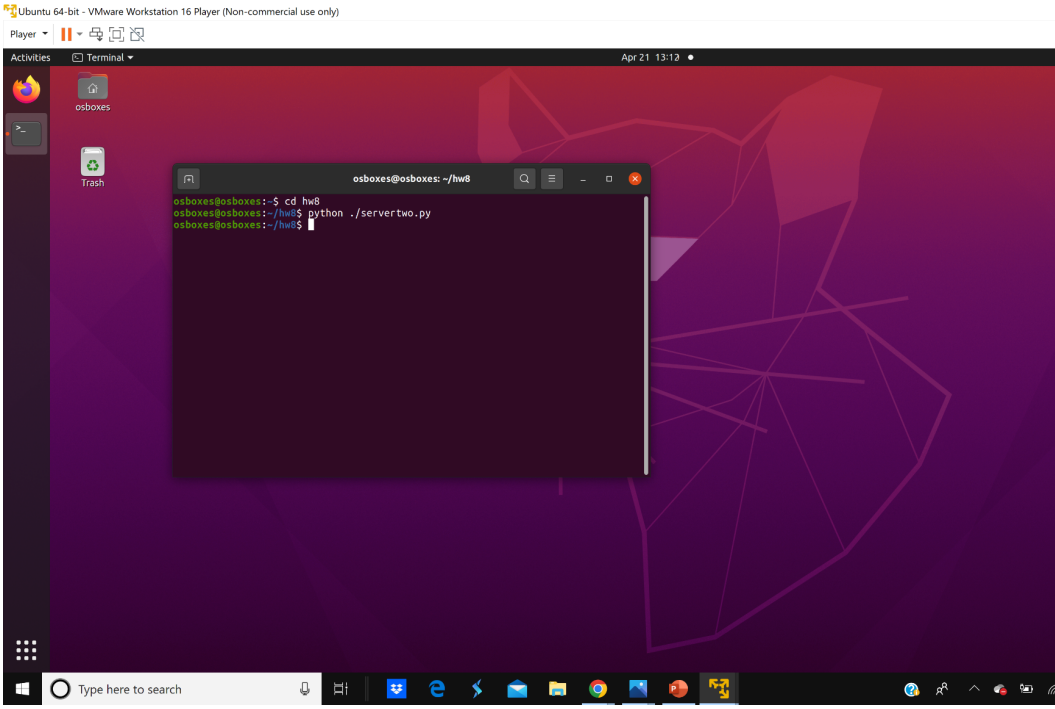
- 

- Next I aimed to create a server.py file on my main VM and a client.py file on my clone VM without ssl at first
  - I used the format of the example code given in the network powerpoint slides and ran into some minor issues like the fact that I had to encode and decode the message for the s.sendall() to work. Or that I had to use the server IP address for the host.
  - Eventually I got the code to work where first you run the server.py on my original VM, then run the client.py on the other. You type your message in the client and then it encodes it and sends it to the server where it receives it and sends it back. Then in the client server it receives it again and decodes it and prints it out

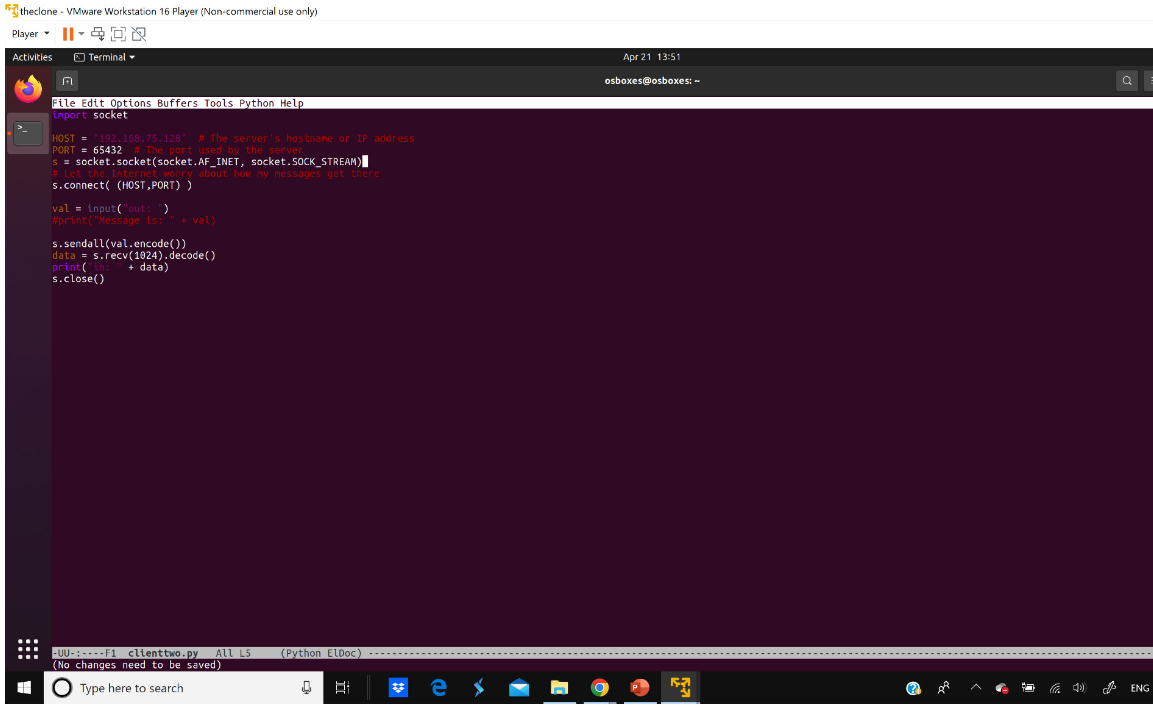
- this is my client run on my clone VM

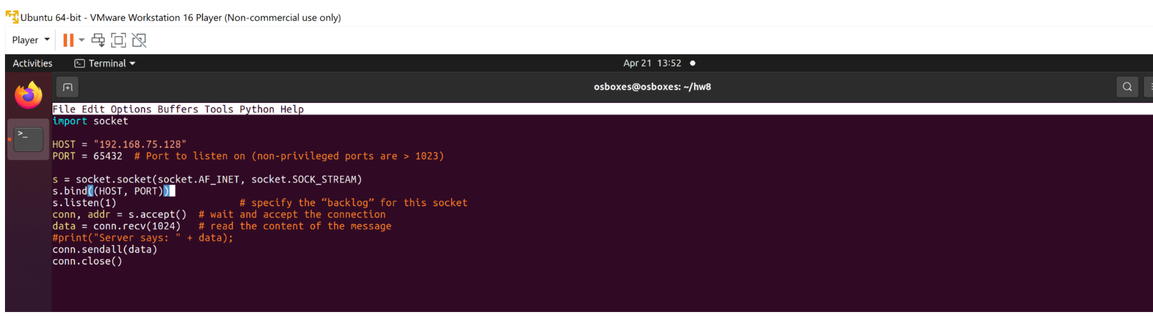


- this is my server run on my original VM

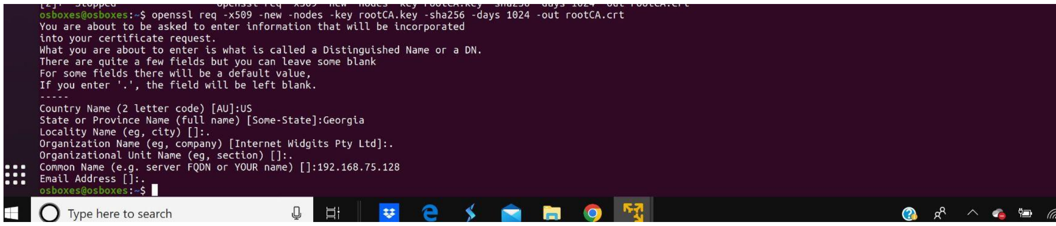


- This is the source code for client and server

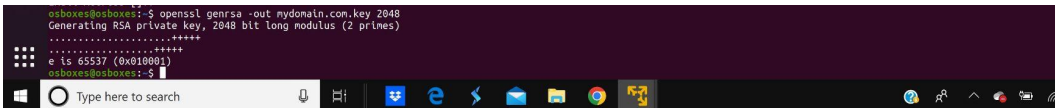
- 

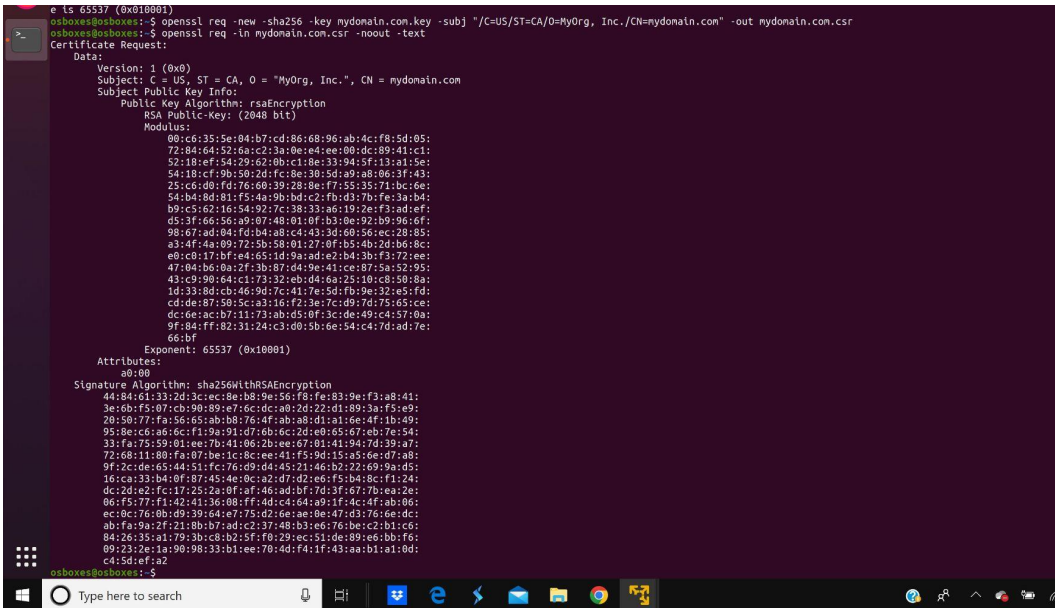
- 

- Now I decided to create the PKI. I mainly followed the given tutorial and did it on the server side.
  - first I did openssl genrsa -des3 -out rootCA.key 4096 to create the root key
    - 

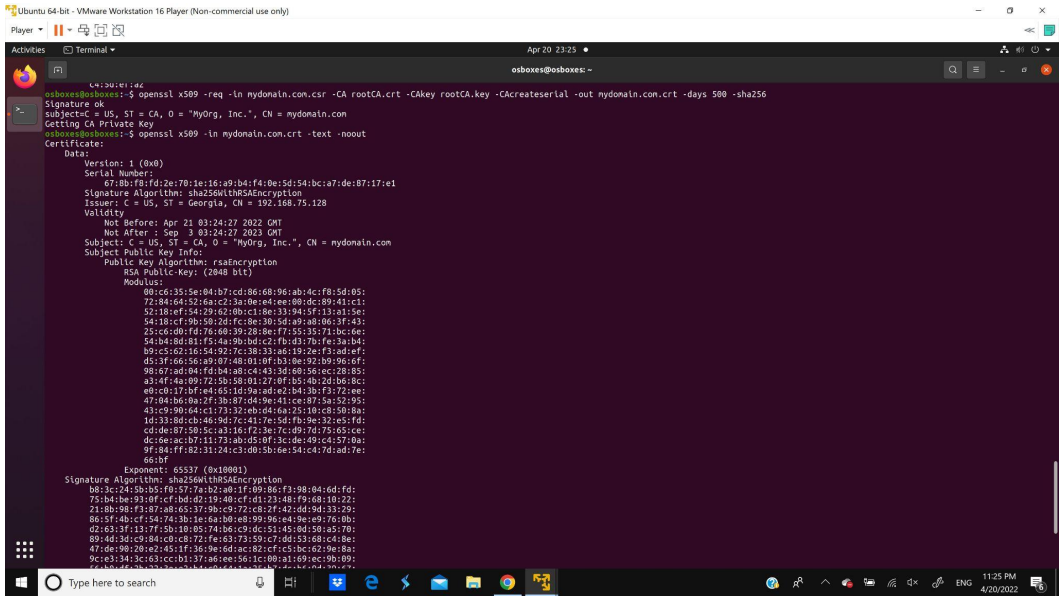
  - then i did openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.crt to create and sign the root certificate
    - 

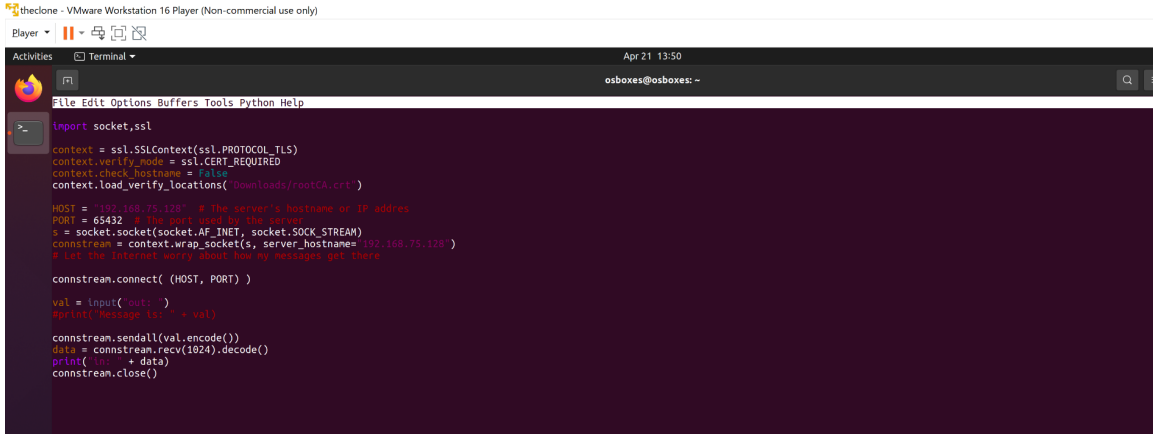
    - for common name I made sure to put the server IP address

- then i did openssl genrsa -out mydomain.com.key 2048 to create the certificate key
  - 

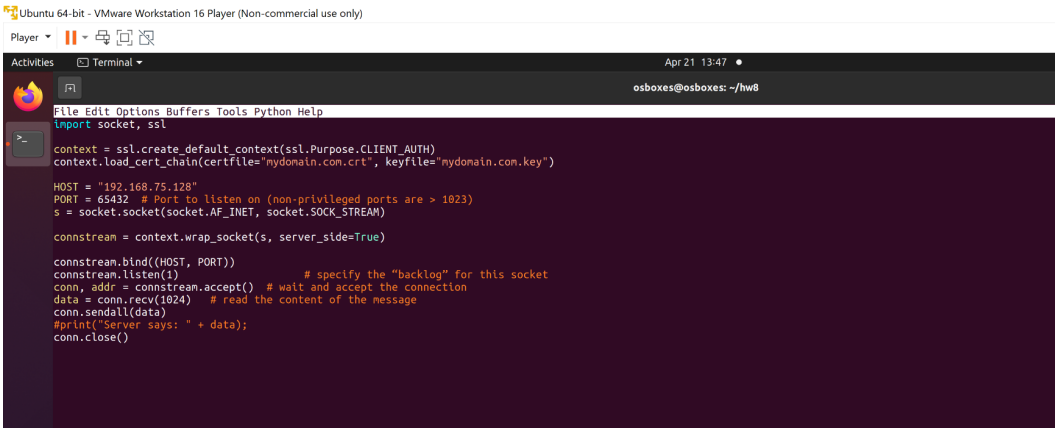
- Then I did the one liner version openssl req -new -sha256 -key mydomain.com.key -subj "/C=US/ST=CA/O=MyOrg, Inc./CN=mydomain.com" -out mydomain.com.csr in order to create the csr
- I did openssl req -in mydomain.com.csr -noout -text to certify the csr content
  - 

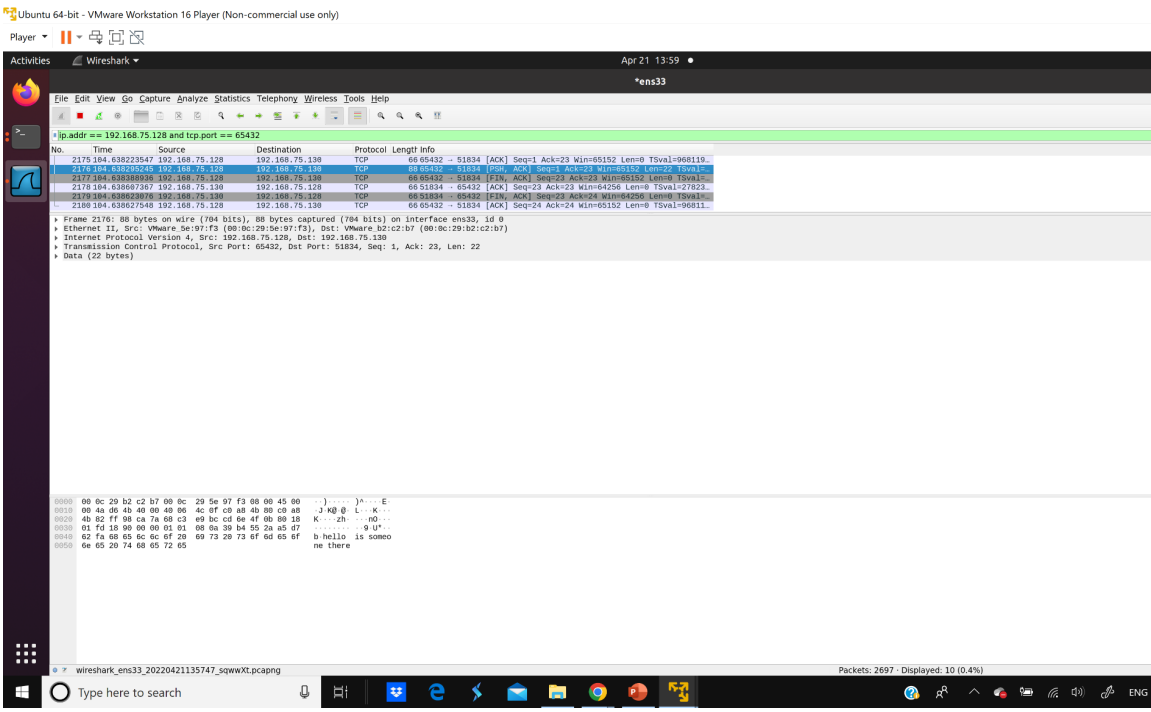
- then I made the certificate using mydomain csr and the CA Root key with openssl x509 -req -in mydomain.com.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out mydomain.com.crt -days 500 -sha256
- Finally I verified the certificates content with openssl x509 -in mydomain.com.crt -text -noout

- 

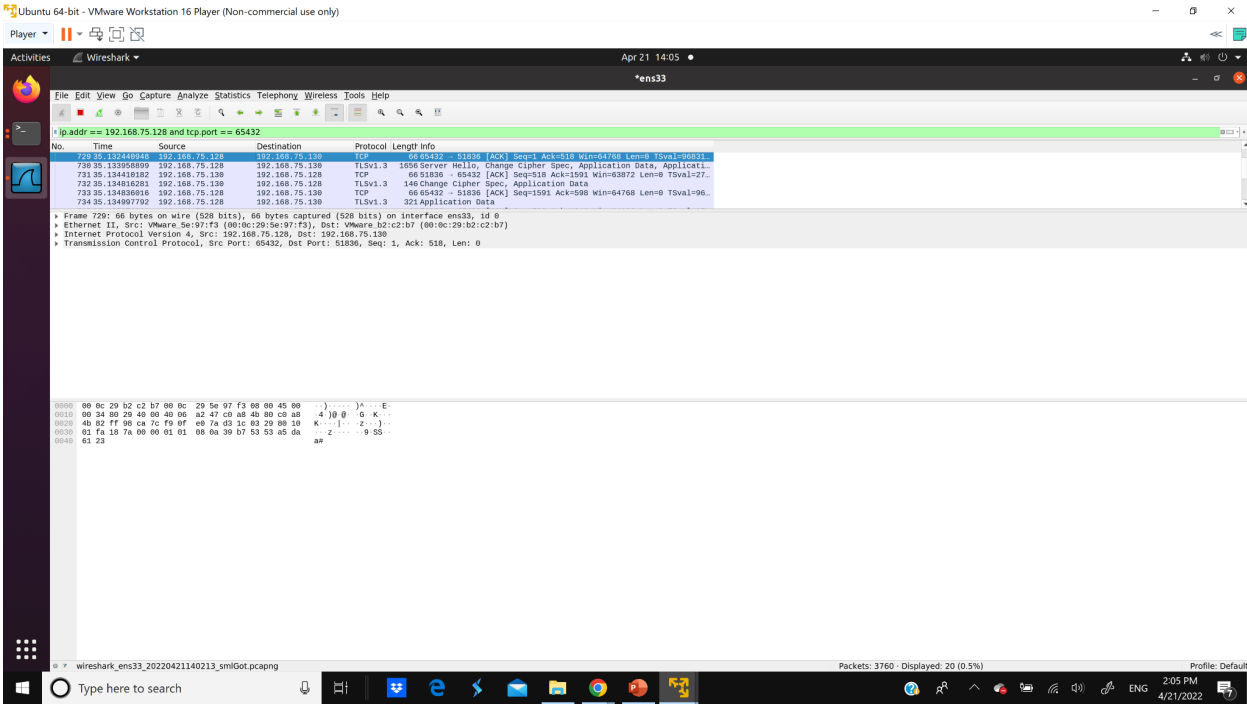
  - After doing all of this I hardcopied my rootCA.crt to my client VM
- I now have the PKI set up so now I can go ahead and wrap my socket with SSL
  - for client I constructed my context by
    - context = ssl.SSLContext(ssl.PROTOCOL_TLS)
    - context.verify_mode = ssl.CERT_REQUIRED
    - context.check_hostname = False
    - context.load_verify_locations("Downloads/rootCA.crt")
  - then I wrapped my socket with context.wrap_socket(s, server_hostname=192.168.75.128)
  - 

  - I really only needed the rootCA.crt for the client side of the PKI
  - for server I created my context by
    - context = ssl.create_default_context(ssl.Purpose.CLIENT_AUTH)
    - context.load_cert_chain(certfile="mydomain.com.crt", keyfile="mydomain.com.key")
    - I then wrapped my socket around using
      - connstream = context.wrap_socket(s, server_side=True)

-
-
- 

  - After doing this the program runs the exact same as before
- I then decided to verify that the encryption works by using wireshark on the server first with the unprotected no ssl version
  - I filtered it with ip.addr == 192.168.75.128 and tcp.port == 65432
  - 

  - as you can see the plaintext is visible with no ssl
- Now I did the same for the ssl wrapped version and when I checked through them all, I can find no plaintext that had the message, which means that it was encrypted

- 

- If you modify the server.key I believe the client can't connect to the server and you get the certificate verify failed error code because the server.key doesn't match
  - 

- If you change the CA file on the client side, I would assume that It would not work as well because the rootCA.crt does not correspond with the one on the server side
  -
-
-