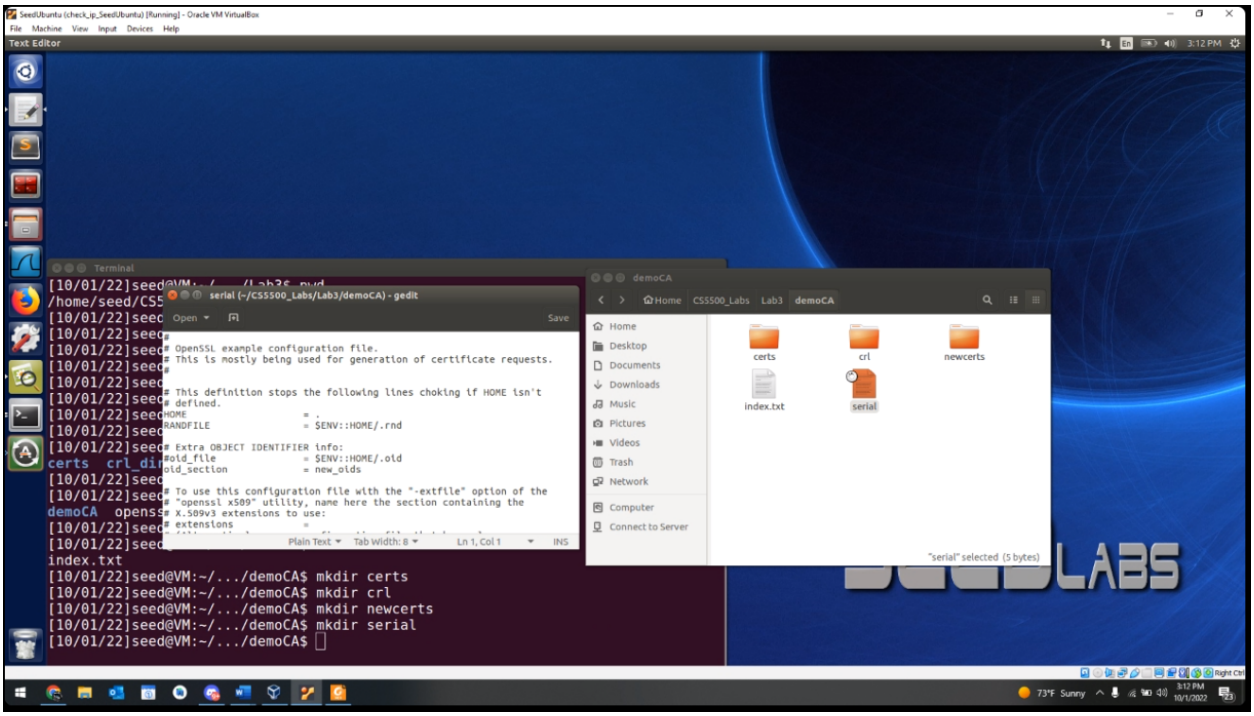
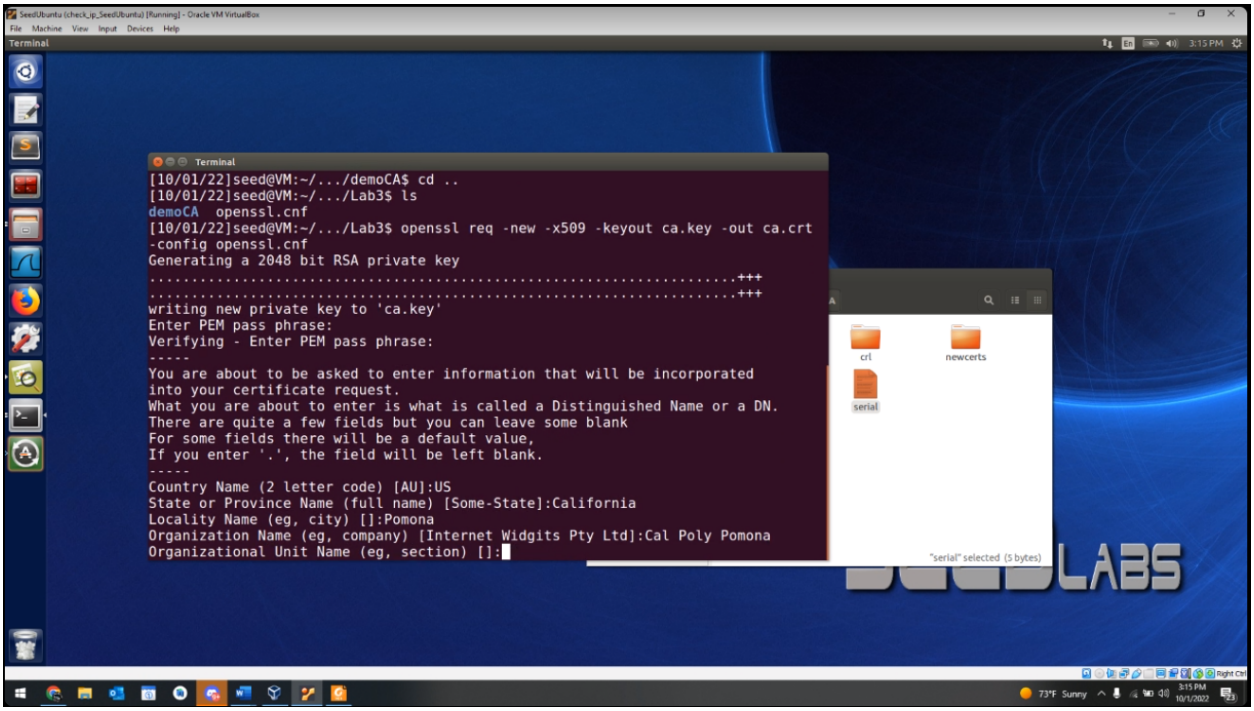
Group 8: Kevin Vo, Mason Godfrey, Thanh Le

CS 5500 – Lab 3: Public Key Infrastructure

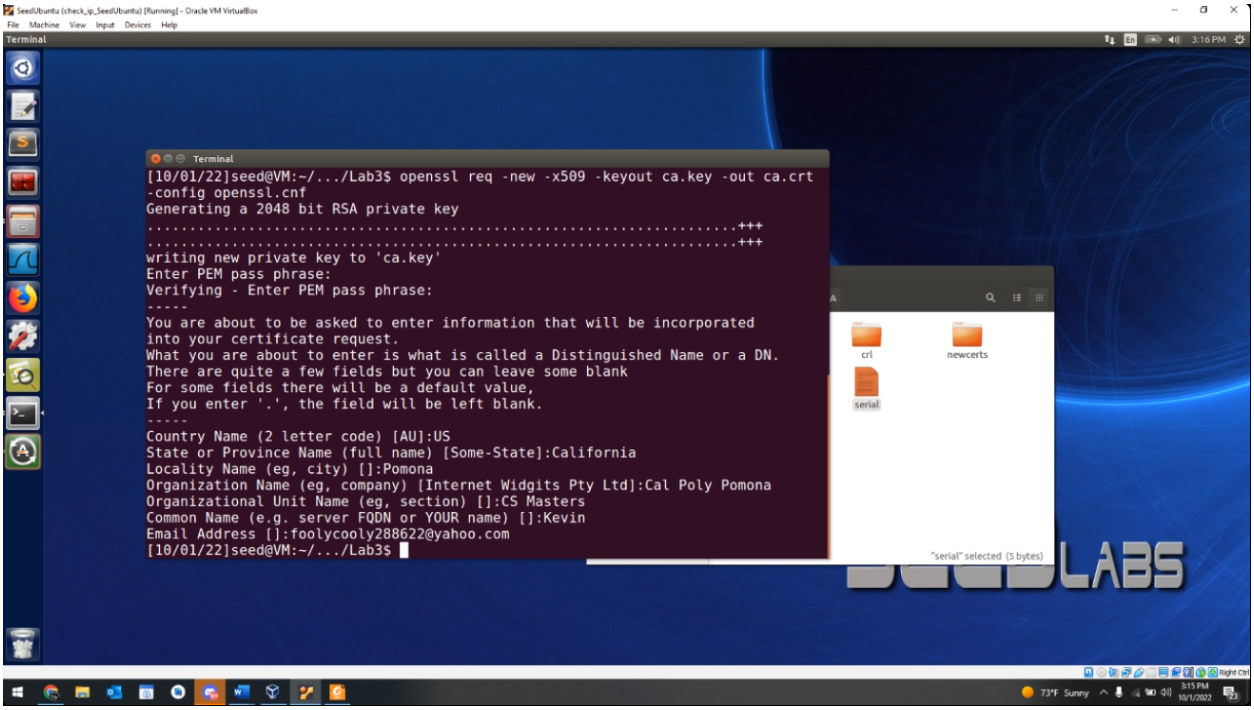
Task 1: Becoming a Certificate Authority

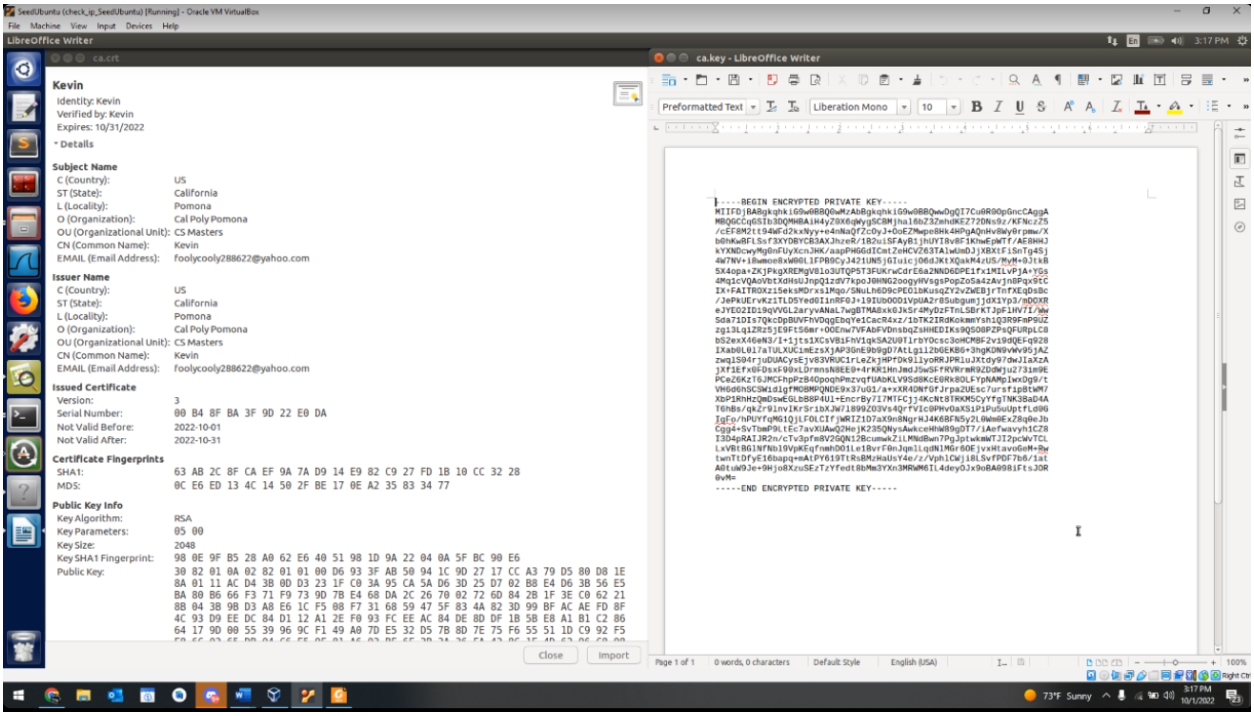


Here we created the proper directories that we were instructed to do. We were curious as the rest of the instructions did not have us use them beyond this point. It was only at the end of the lab that we were able to find auto-generated files from the instructions in this lab. Beyond this, our serial file had the necessary commands to proceed further.



At this point we took the liberty to fill out the Country Name, State, Locality Name, Organization Name, its unit, and so on. We felt this part of the lab is crucial as we learn by mistake that Common Name is a significant value.

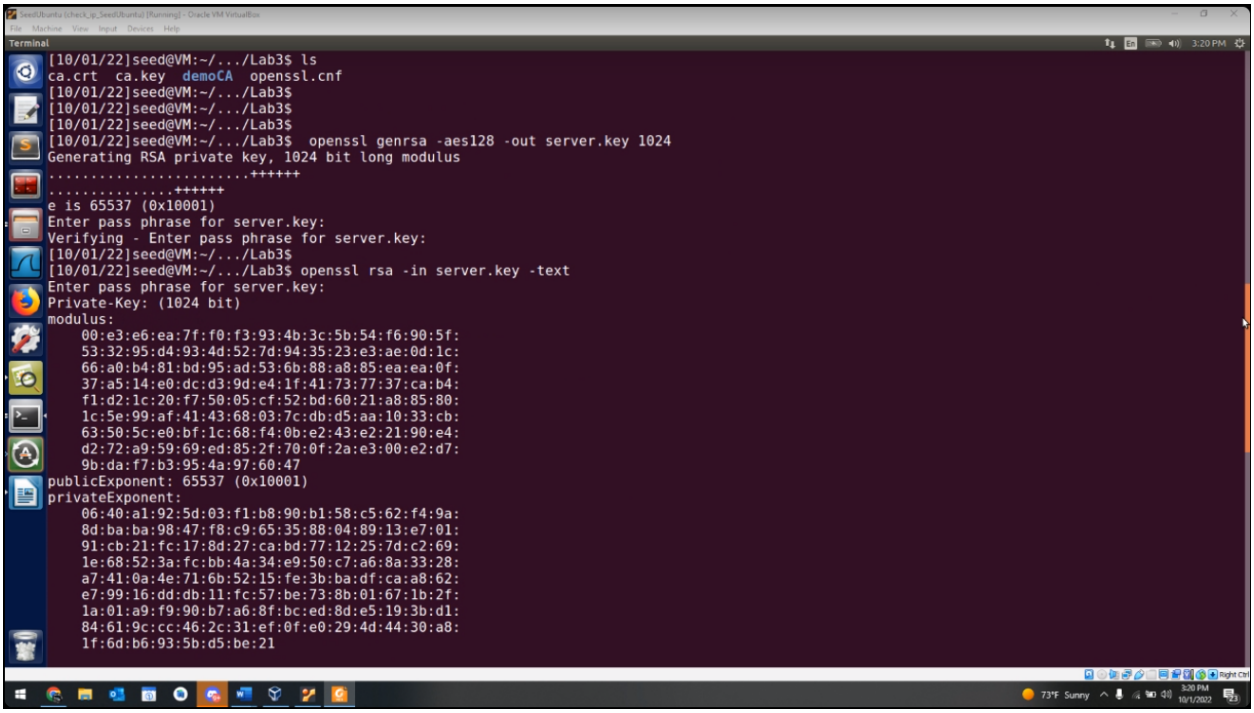




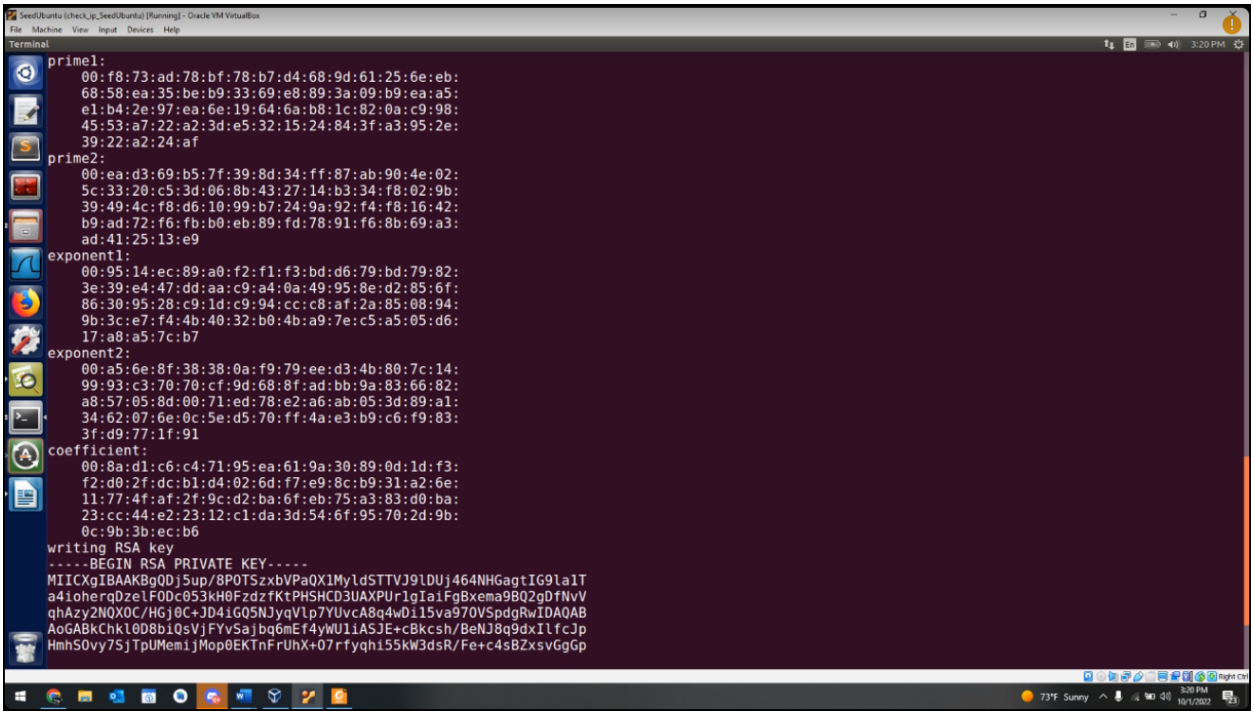
Here is the private key and our CA with our key within it.

Task 2: Creating a Certificate for SEEDPKILab2020.com

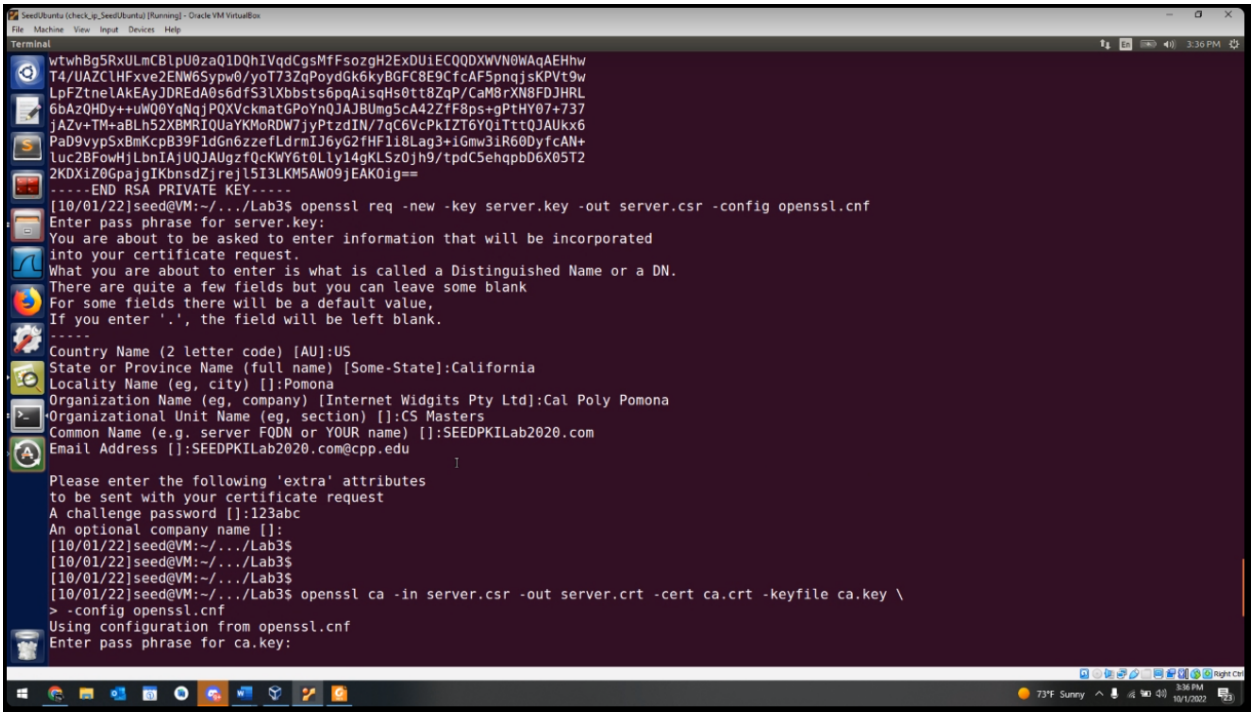
Step 1



Now, we complete a similar process for the server certificate, filling out the information needed to tie the certificate to a website.

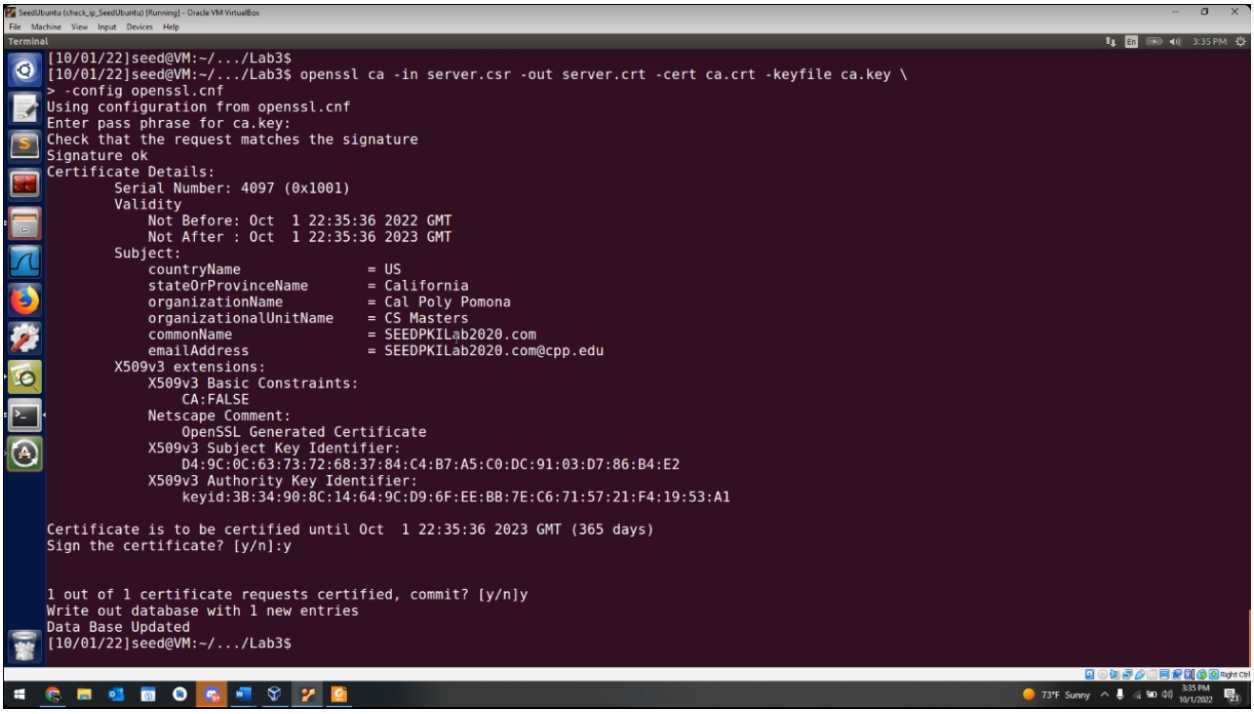


Step 2



We then request for the signature to be signed by our certificate authority to validate it.

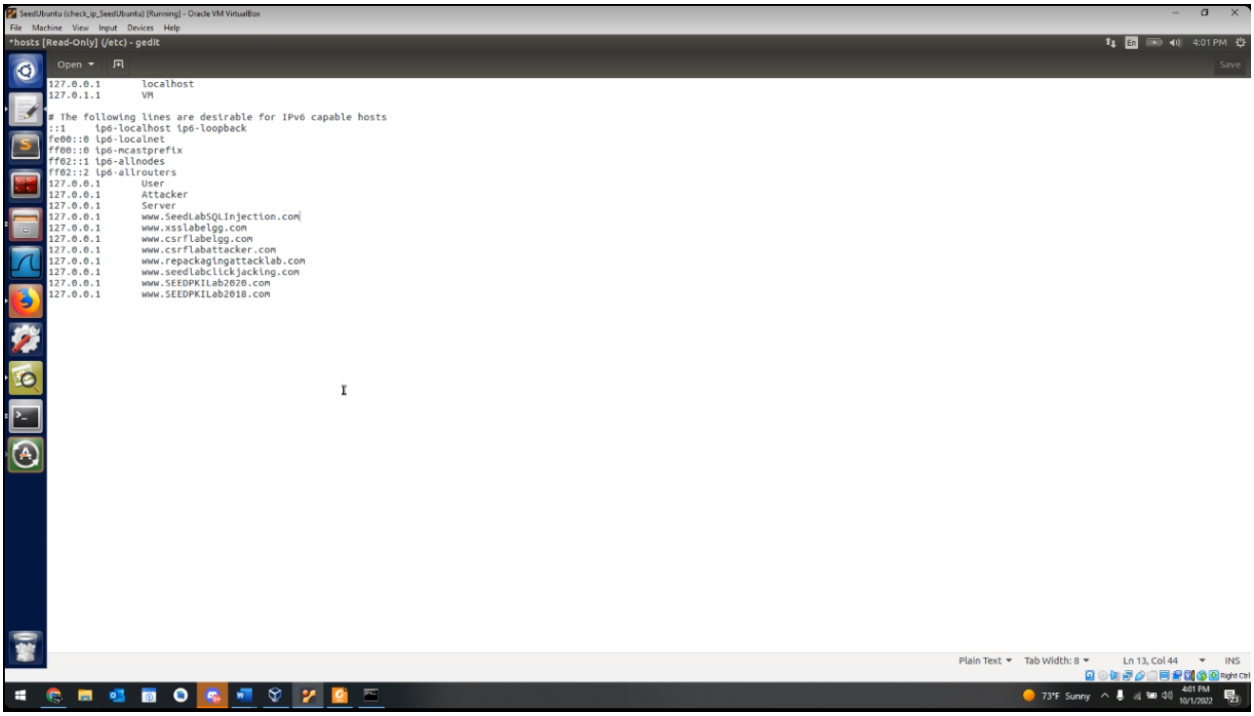
Step 3



We then generate the certificate itself by signing the certificate into our certificate authority.

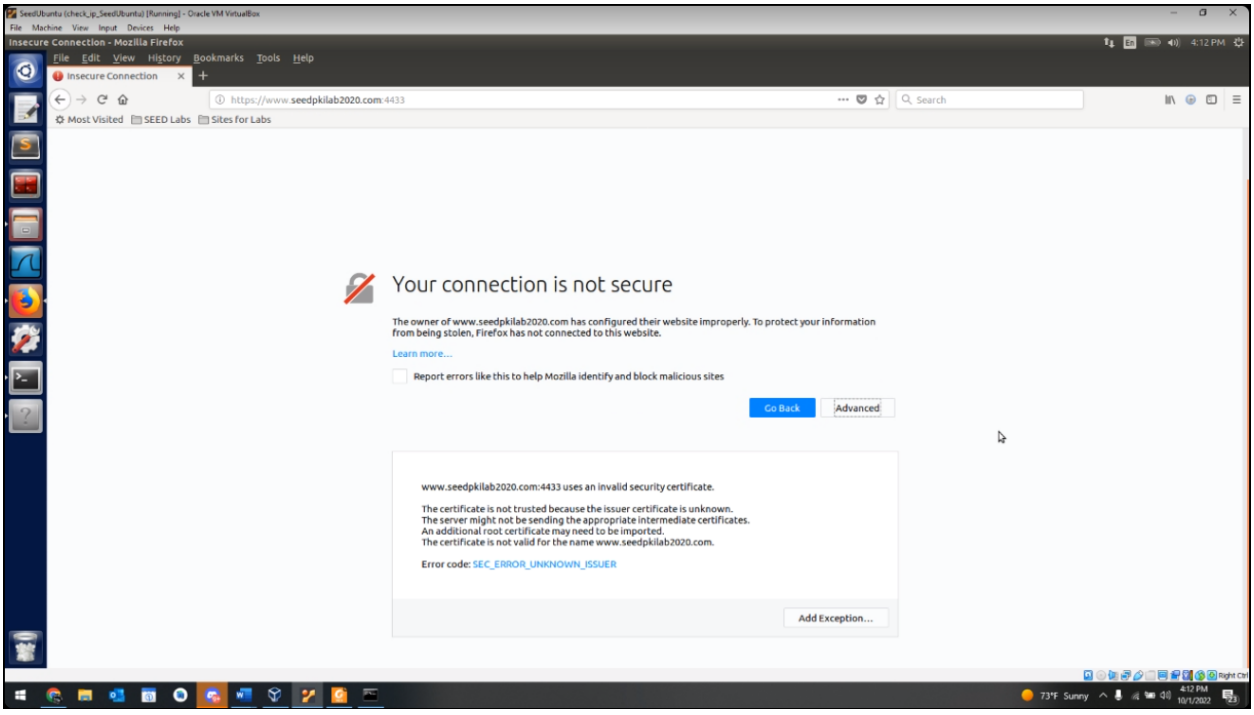
Task 3: Deploying Certificate in an HTTPS Web Server

Step 1



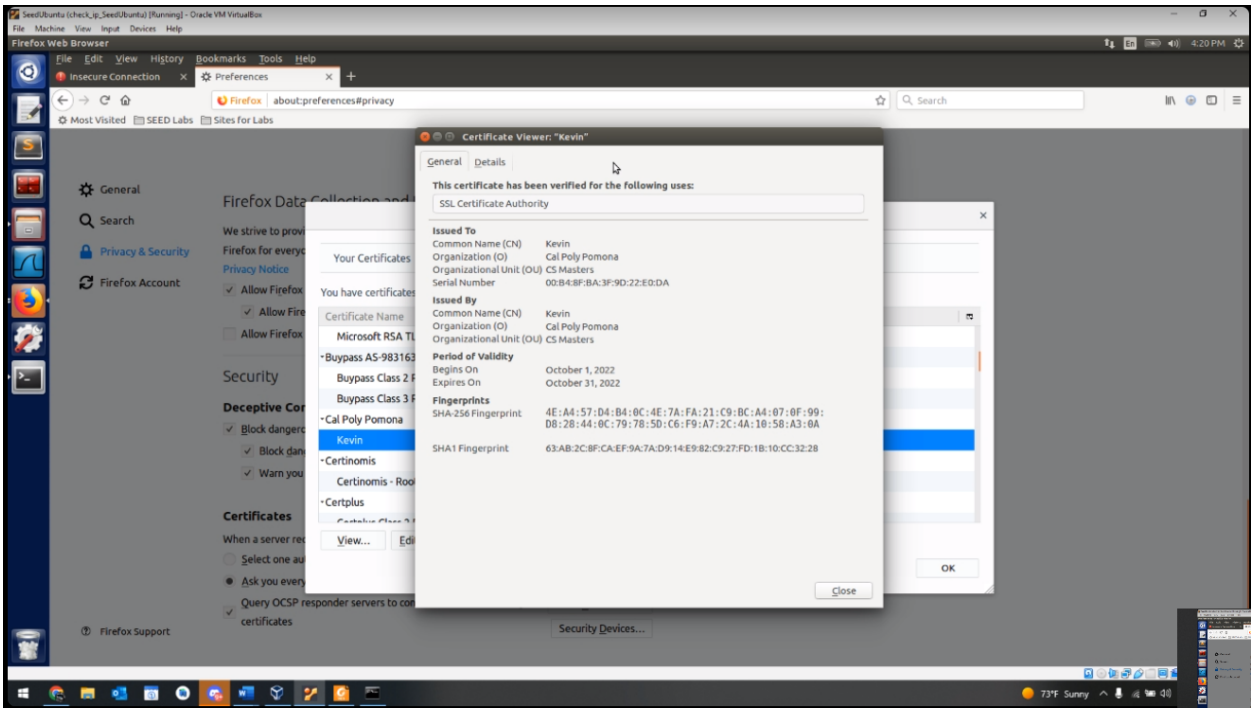
We included the required “www.SEEDPKILab2020.com” and “www.SEEDPKILab2018.com” just in case since the instructions were unclear and occasionally switches out these two URLs. This is done to have our computer recognize this URL/name.

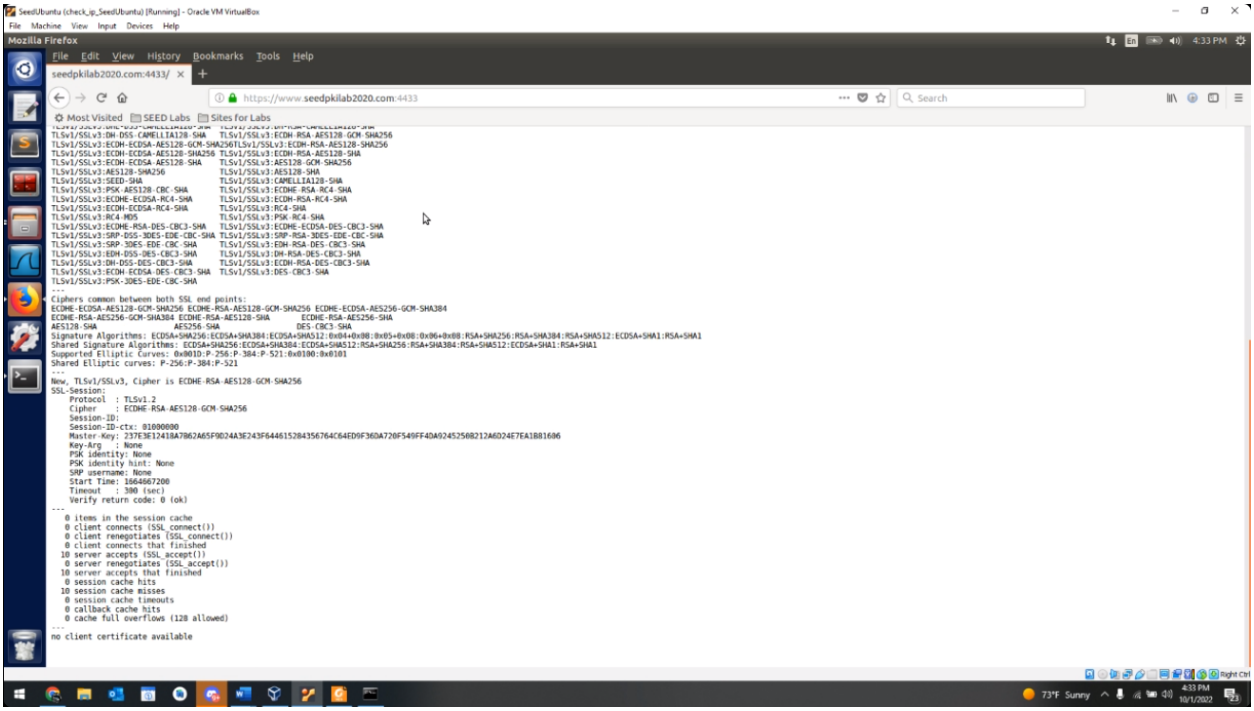
Step 2



This point it took us a while to figure out, but we had to include “www.” within the URL shown in the picture above for it to traverse to this page before you. It found that our connection was insecure since it doesn’t know our security certificate.

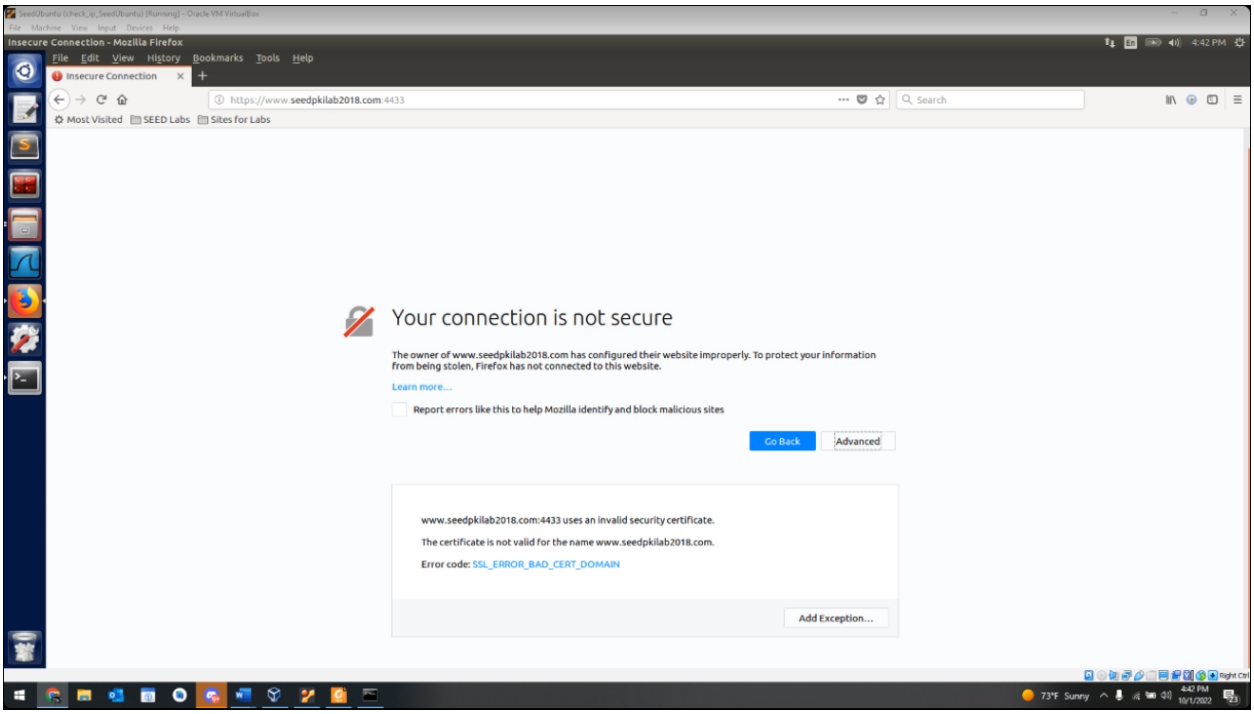
Step 3

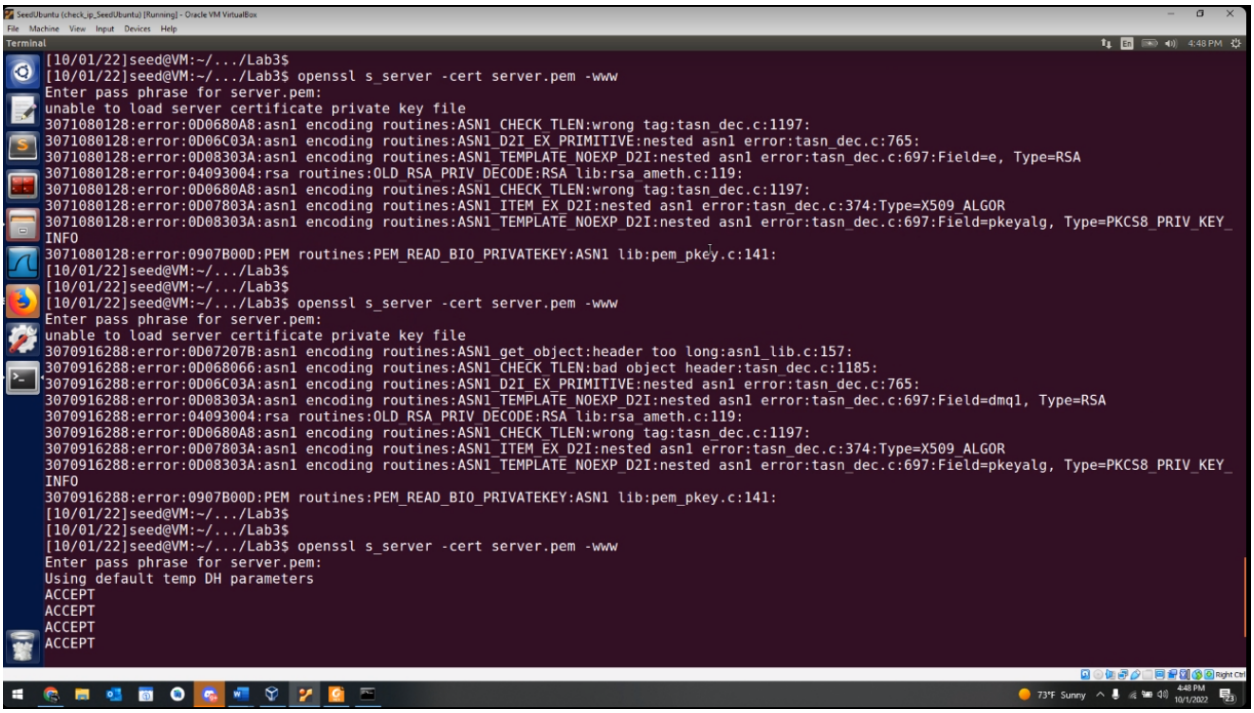




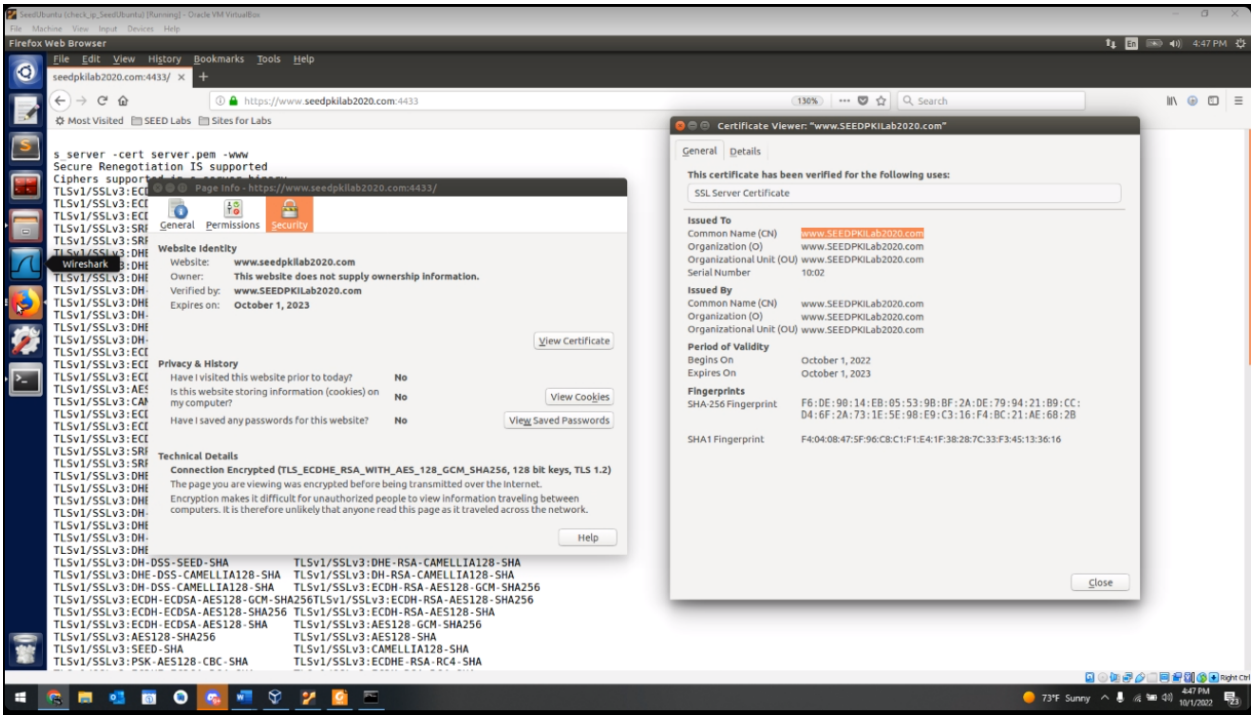
To counter the SEC_ERROR_UNKNOWN_ISSUER, we imported our own certificate into FireFox for it to display the contents of the image above.

Step 4

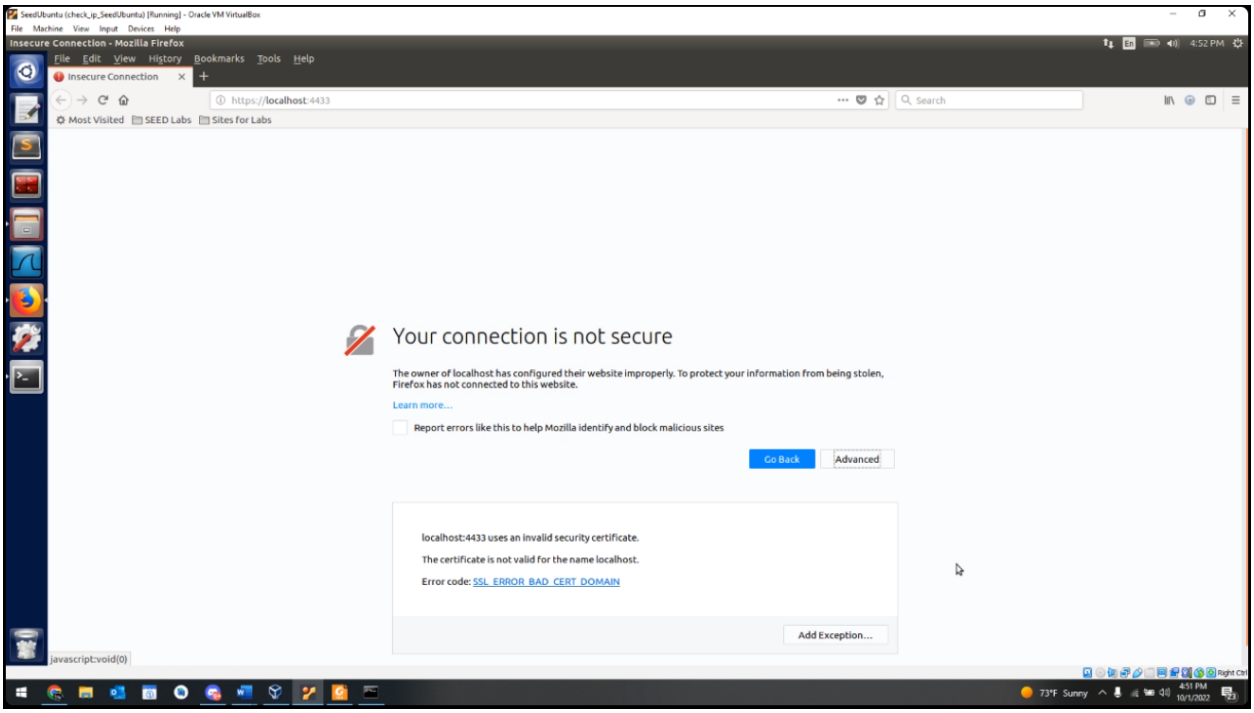




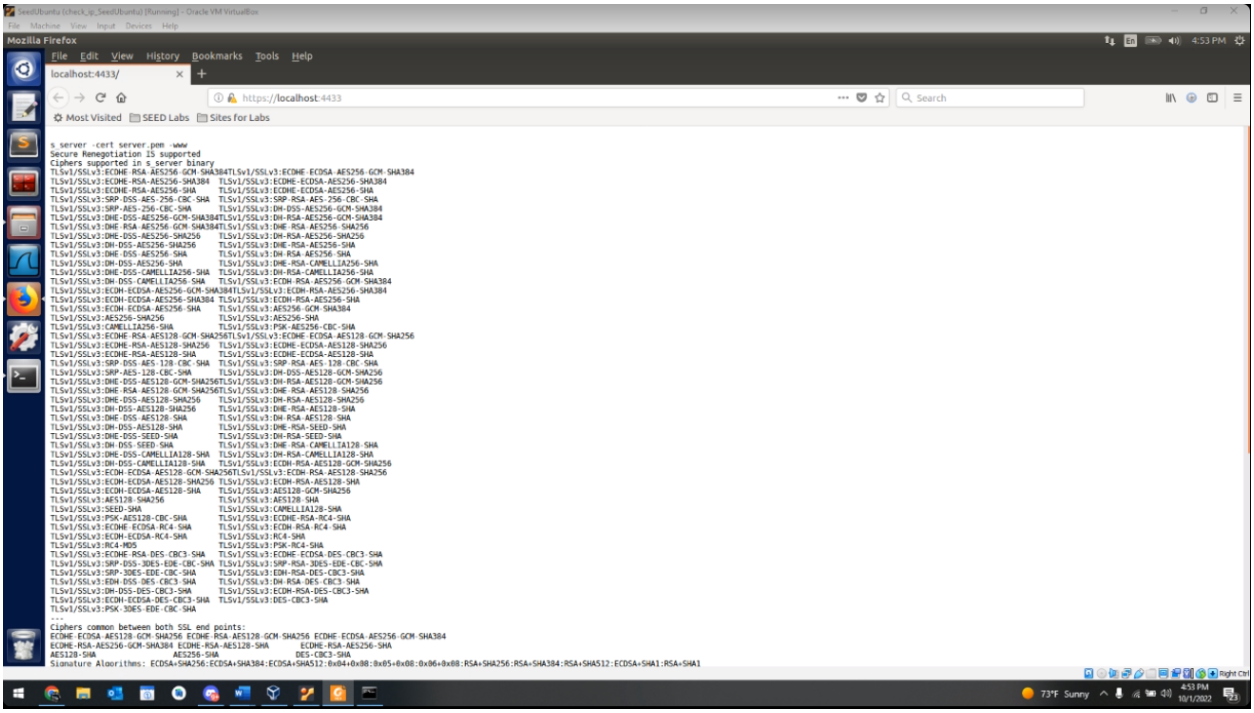
We found that Step 4 was most amusing since we discovered that altering certain characters within the server.pem file can have more weight than others. This can be seen in the image above as our initial change provided us an error when starting up the server with server.pem. However, when we went back and try to change a different character within it, the server worked.



Above is evidence that we did what was asked of us to have our server running and that our connection is secure.



However, when we tried it under localhost, it was not secure as shown above.



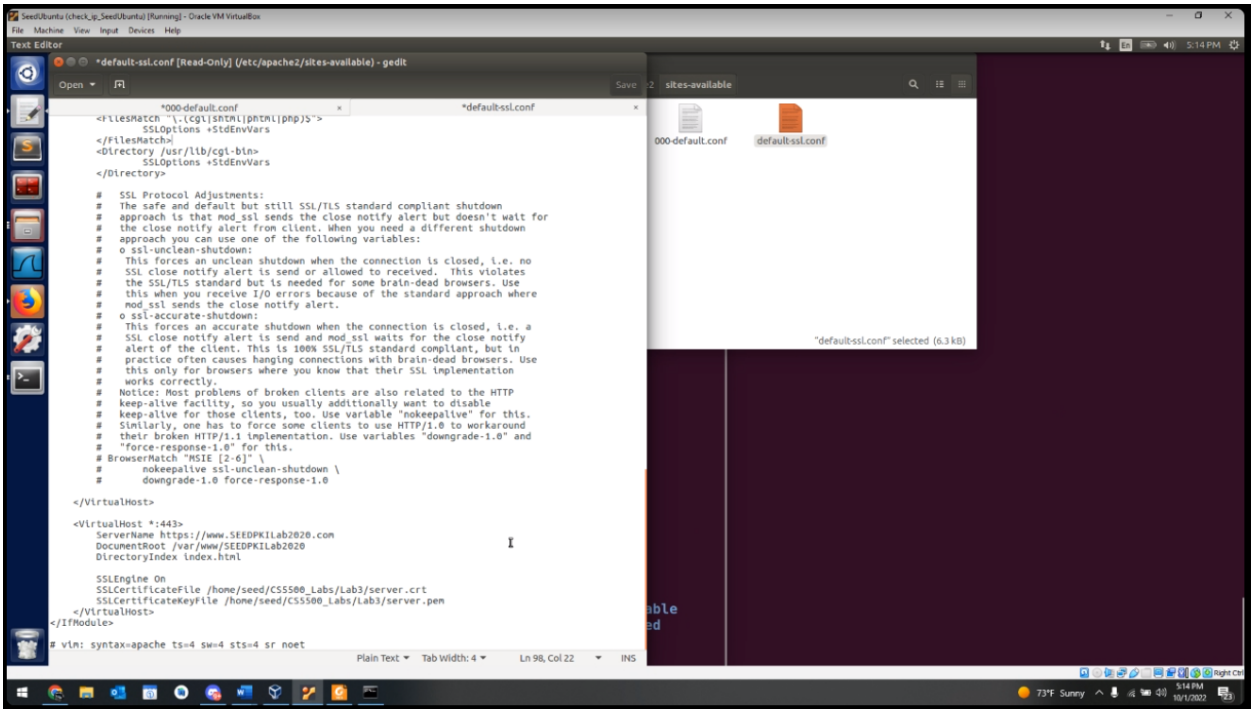
Regardless, after bypassing the warning, we were able to visit the site and see that the contents were the same.

1. Modify a single byte of server.pem, and restart the server, and reload the URL. What do you observe? Make sure you restore the original server.pem afterward. Note: the server

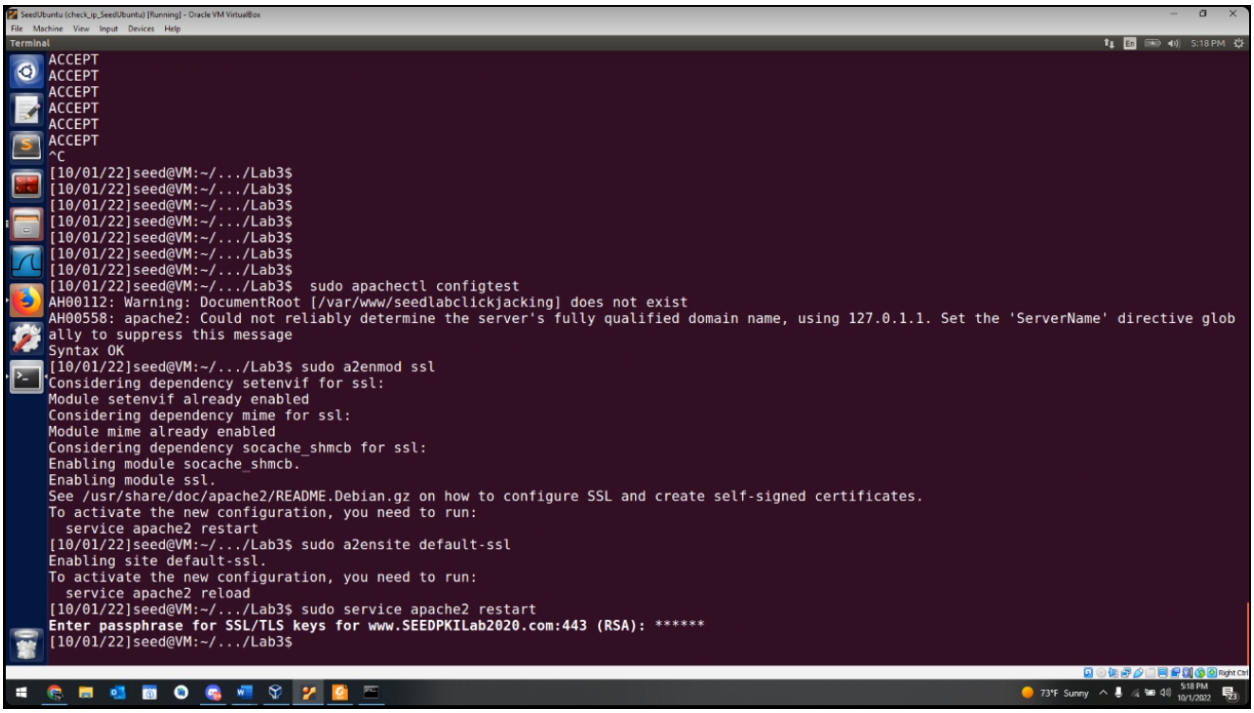
may not be able to restart if certain places of server.pem is corrupted; in that case, choose another place to modify.
    a. Since we never applied the certificate to the 2018 site, modifying a byte of server.pem didn't affect it. While the server didn't start at first due to corruption, we were able to modify a new byte that didn't seem to affect the 2020 site's stability after restarting the server.
2. Since SEEDPKILab2020.com points to the localhost, if we use https://localhost:4433 instead, we will be connecting to the same web server. Please do so, describe and explain your observations.
    a. When we connected via the localhost, we were told that our connection was insecure but were able to connect after bypassing the security warning.

Task 4: Deploying Certificate in an Apache-Based HTTPS Website

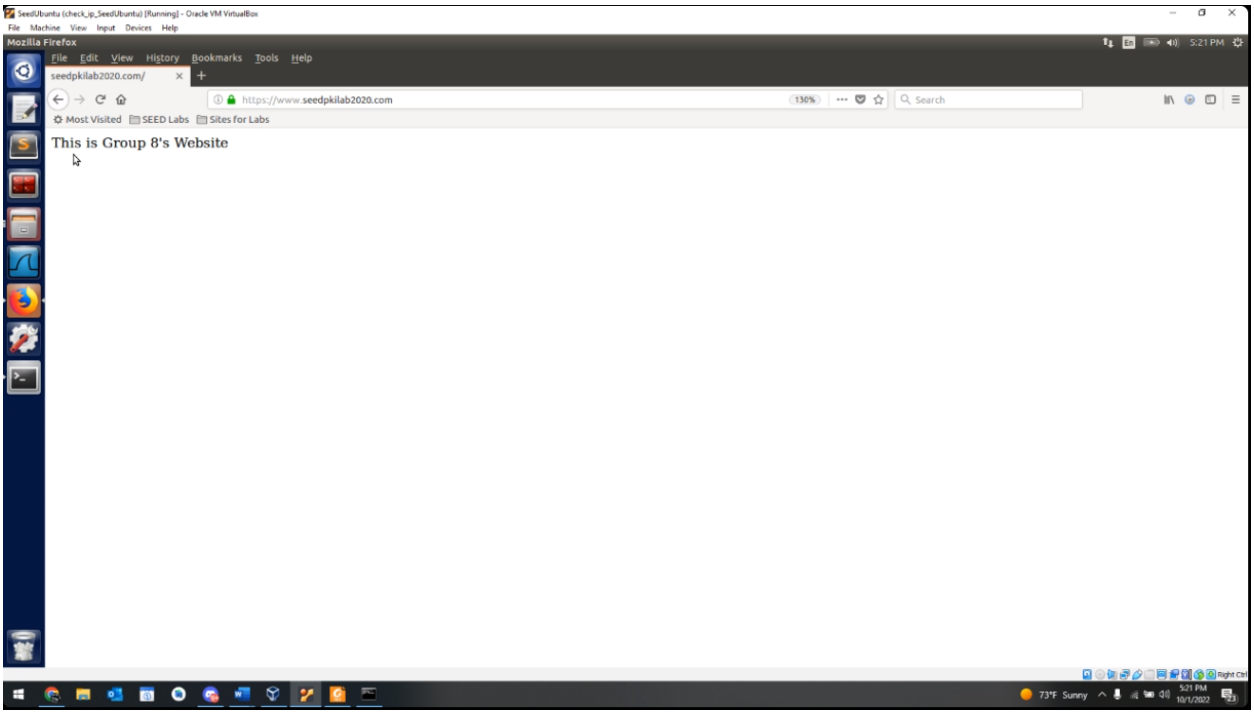
1. Please use the above example as guidance to set up an HTTPS server for SEEDPKILab2020.com. Please describe the steps that you have taken, the contents that you add to Apache's configuration file, and the screenshots of the final outcome showing that you can successfully browse the HTTPS site.
    a. We modified default-ssl.conf to add in a new entry, as this is how HTTPS connections can be generated. We created a new folder for the DocumentRoot path, and placed an index.html file within that had a short message about our group (group 8). We also linked our key and certification files inside the default-ssl.conf.



We wrote the website SSL information into default-ssl.conf, ensuring that the site with the expected URL is generated and that the contents of the website are from the index file.



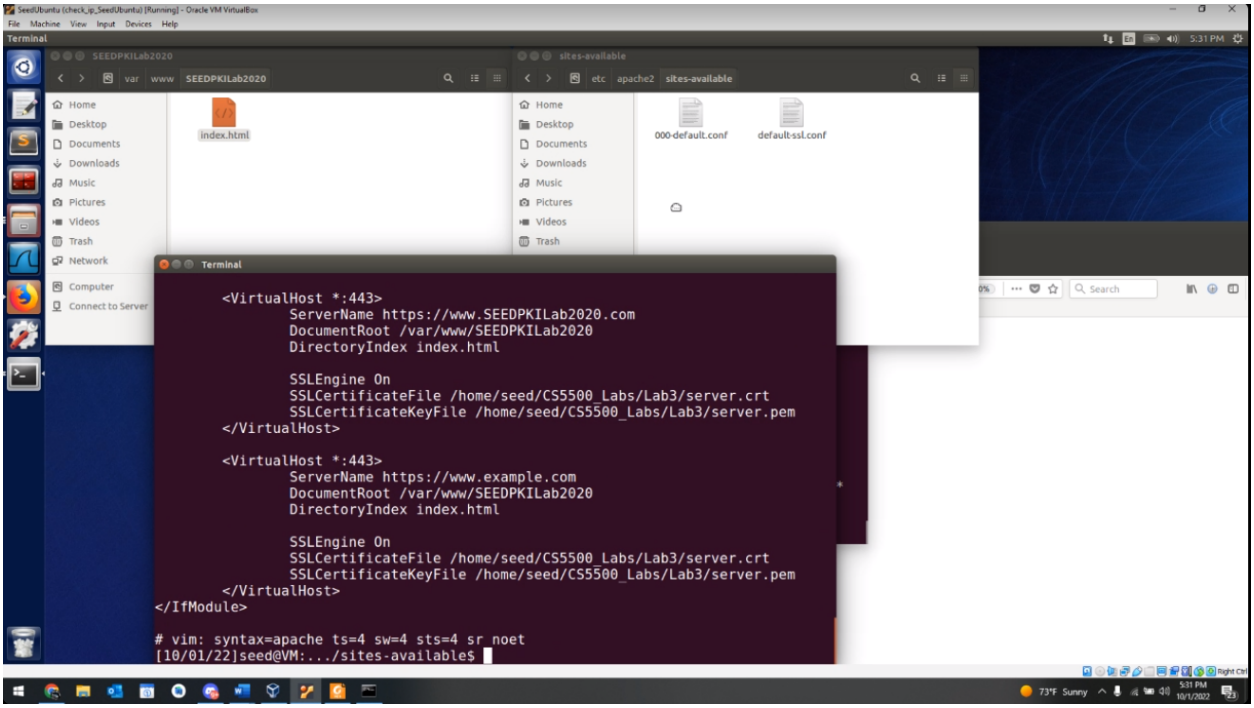
We prepared Apache to have it host the websites locally.



We went to the website, and found that it was being hosted locally and had the expected index contents.

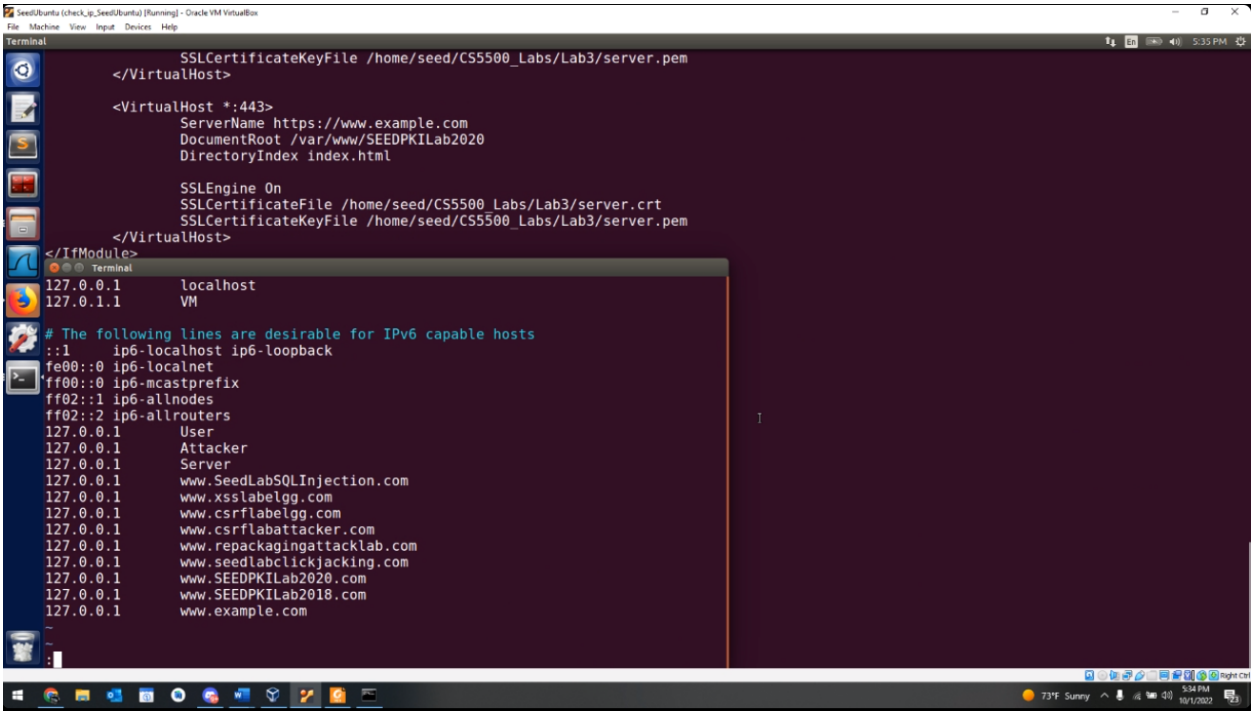
Task 5: Launching a Man-In-The-Middle Attack

Step 1



We modified a configuration file to tell Apache that example.com should have the same contents as the other website.

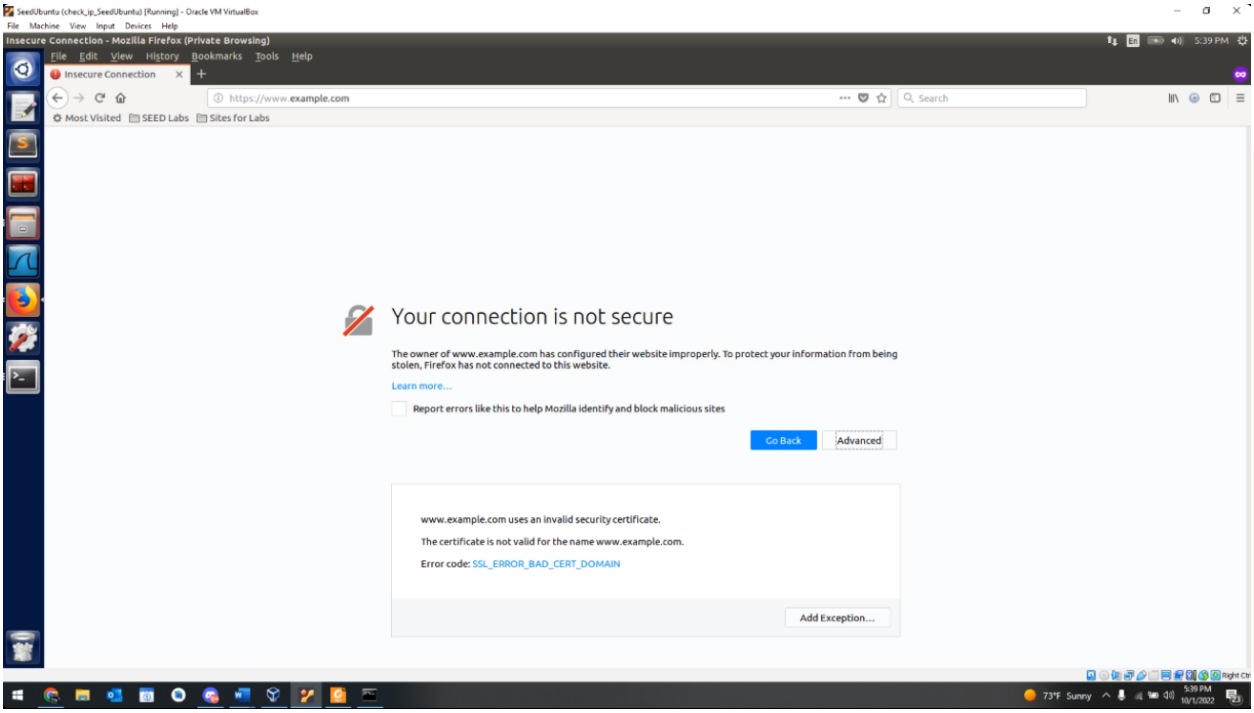
Step 2



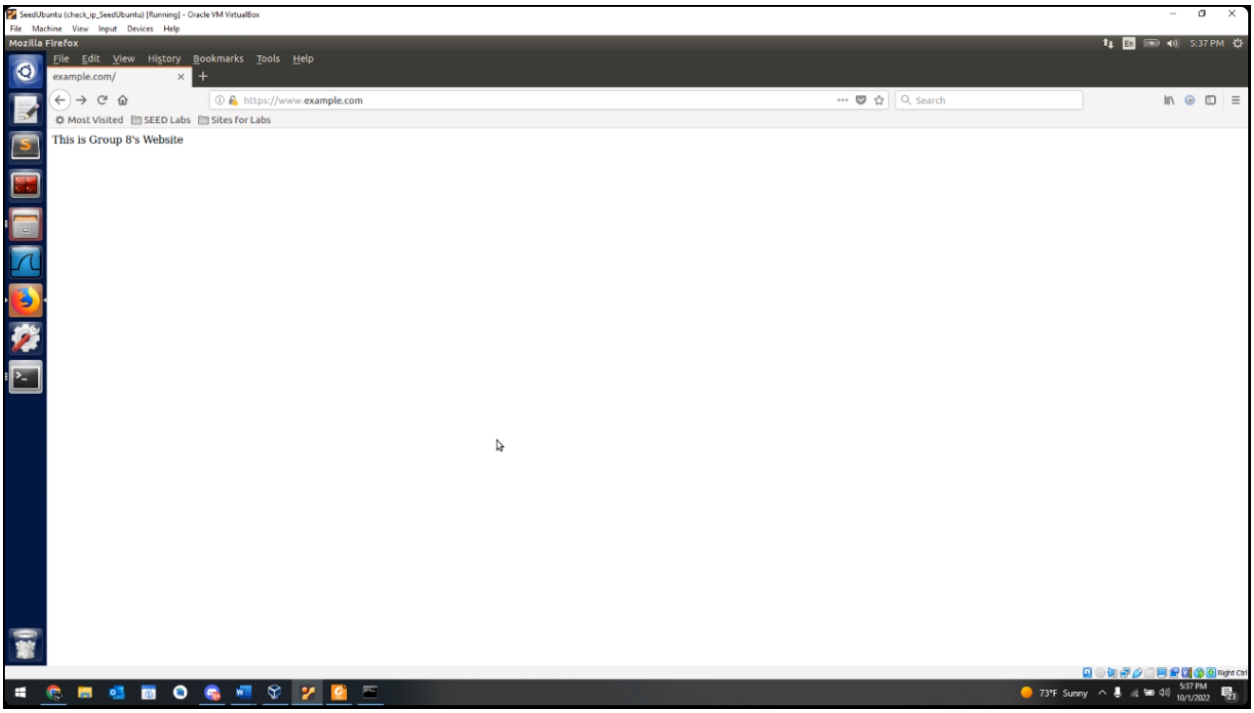
We overwrote the DNS information for example.com to redirect locally.

Step 3

1. With everything set up, now visit the target real website, and see what your browser would say. Please explain what you have observed.
    a. Firefox realized that the certificate associated with our attack was incorrect, and gave a warning that the connection was not secure. However, when we bypassed this warning we were allowed to connect to example.com which showed us the malicious message.

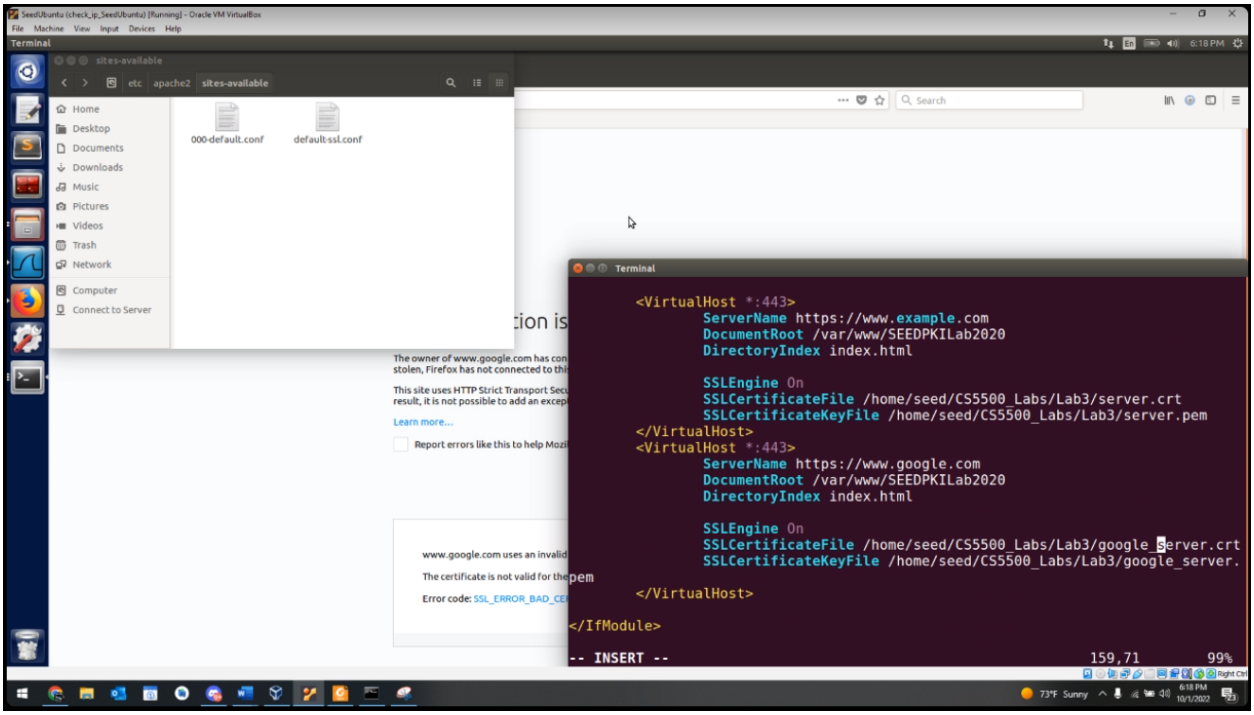


We were told that the certificate is invalid (it was signed for the incorrect site, not for example.com).

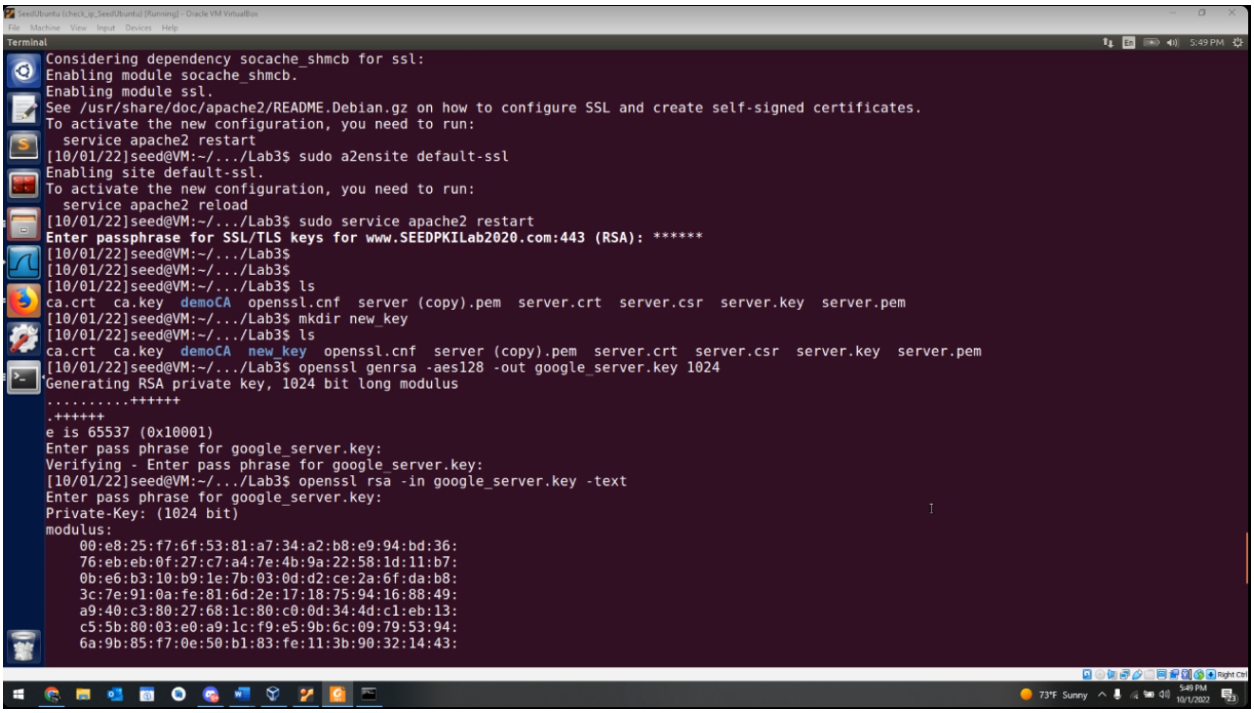


We were able to ignore the warning and view the site's contents, which have been overwritten on our own.

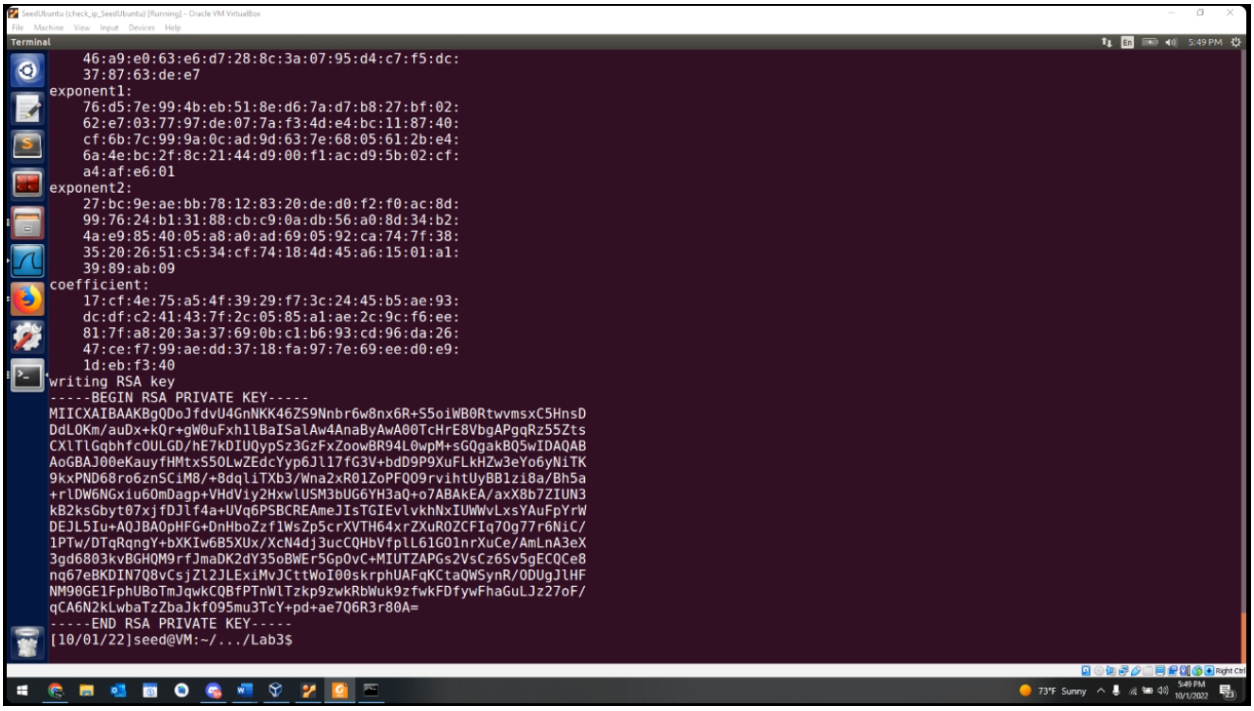
Task 6: Launching a Man-In-The-Middle Attack with a Compromised CA

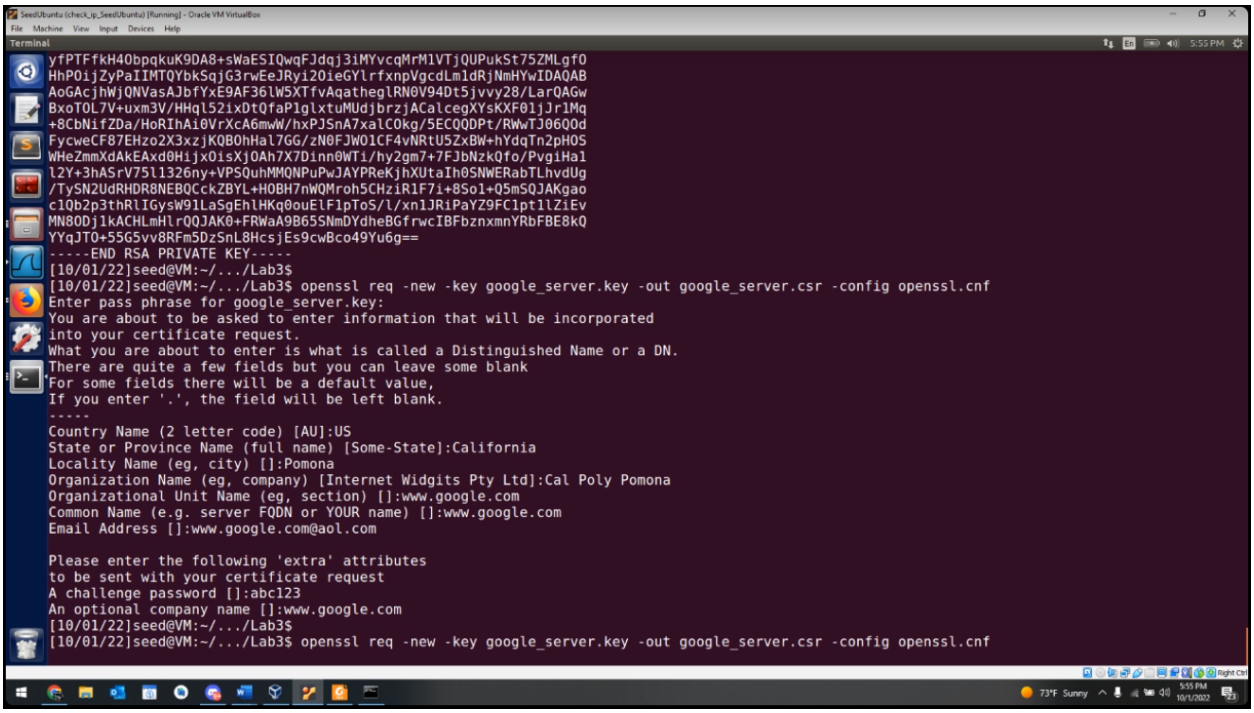


First, we added an additional host to the default-ssl configuration file to tell apache to redirect a local file.

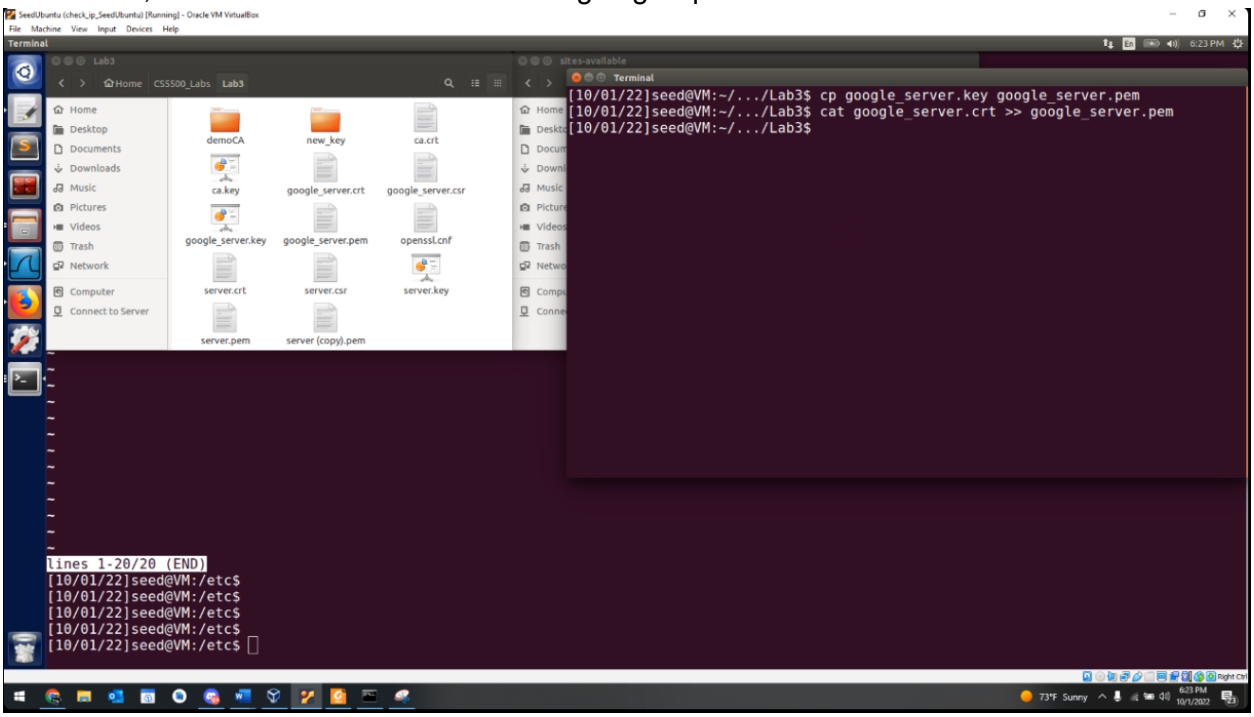


Next, we generated a new key for google so we could have a certificate that tied directly to the spoofed google site.

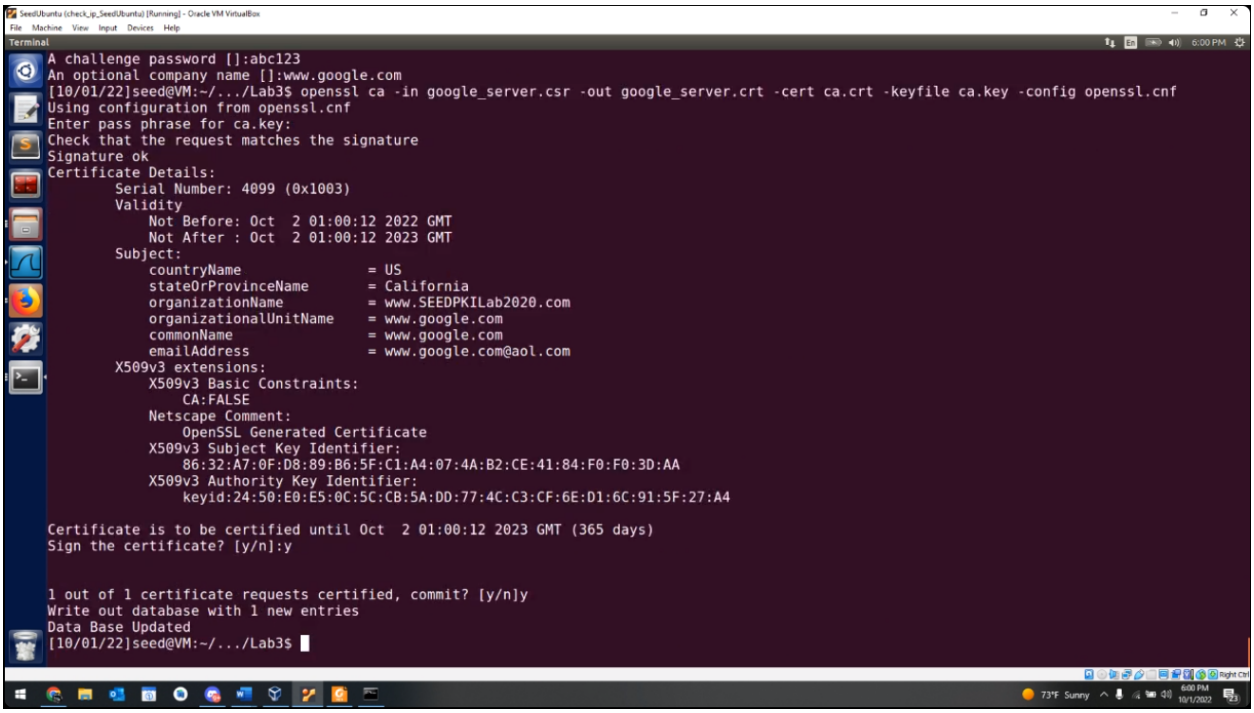




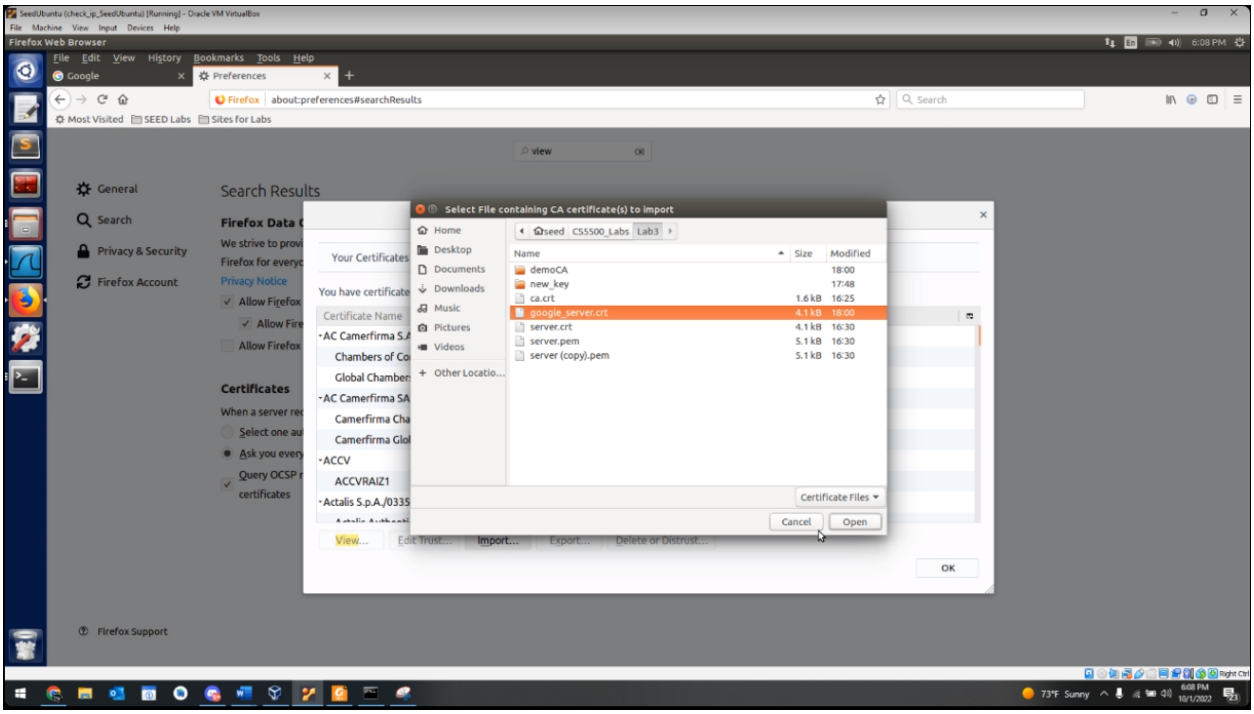
Afterwards, we created the new certificate signing request.



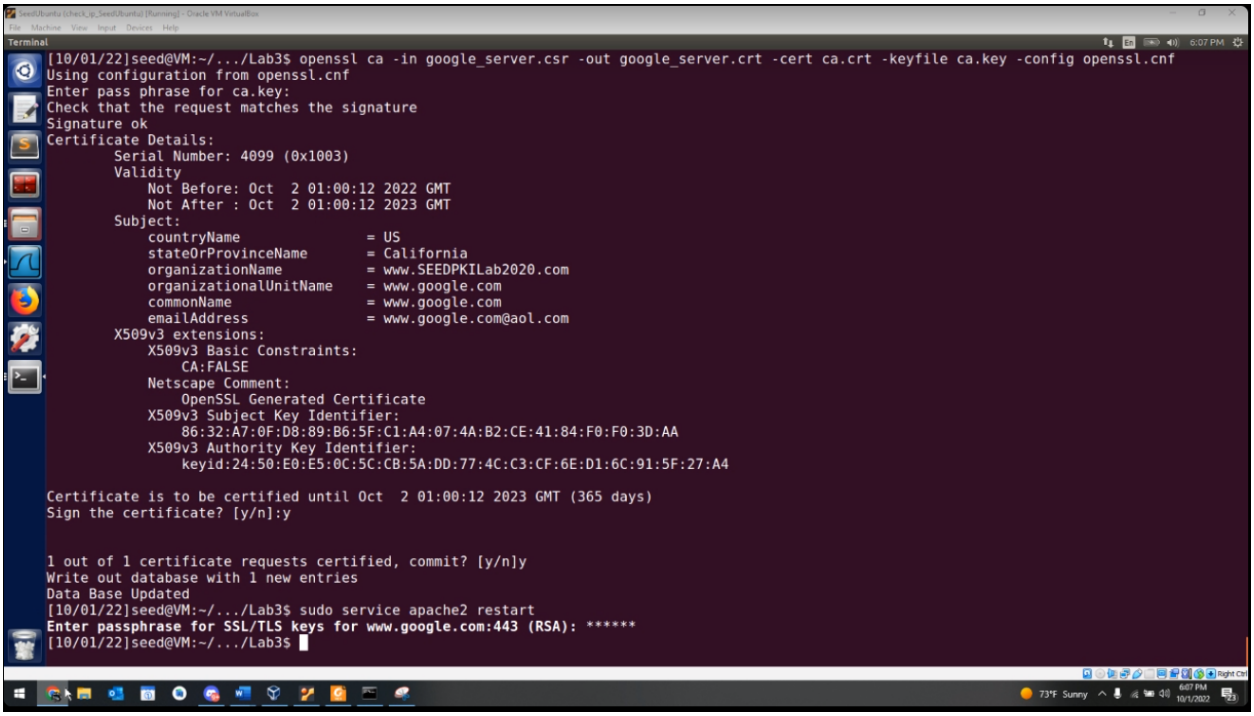
We then placed the key and certificate into a pem file.



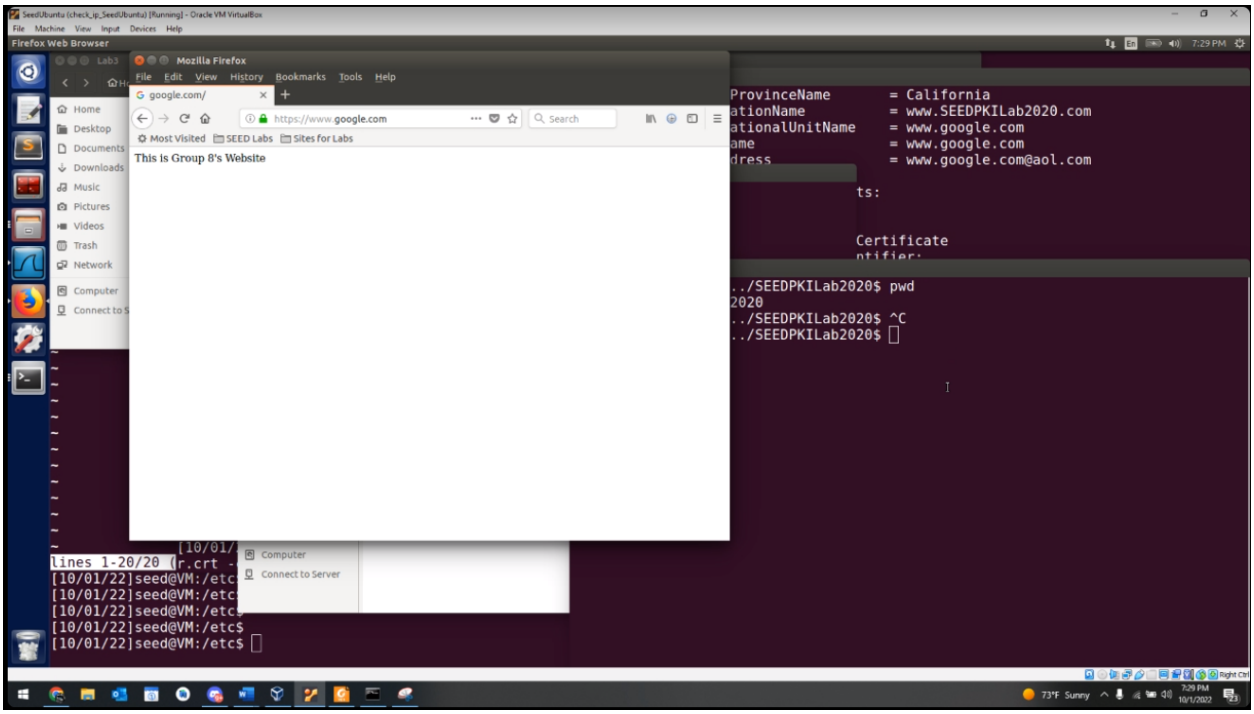
We then signed the certificate, creating a new entry in the ca certificate that had google's spoofed information.



We updated the certificate within firefox to ensure it was accurate.



We then restarted apache to ensure the changes would be reflected.



Finally, when we went to google.com we were told we had a secure connection even though the site was compromised.