**Task 1**

To begin the project, a digital certificate was created using the openssl.cnf configuration file. The directories given in the SEED Lab manual were made and placed in the correct folders. The openssl.cnf file was moved to the appropriate location. In addition, a blank text file was created and named index.txt and a file named serial containing the integer 1000 was also made. These files were placed in the demoCA folder. Figure 1 displays the terminal prompts used to create the directories and the files in their appropriate folders. Then, a certificate was created using the command prompt shown in Figure 2. This results in the generation of an RSA private key. When asked for a PEM pass phrase “cis4360” was chosen and verified. The command prompt then asks for a country name, state, city, company, organization unit name, common name, and email seen in Figure 3.

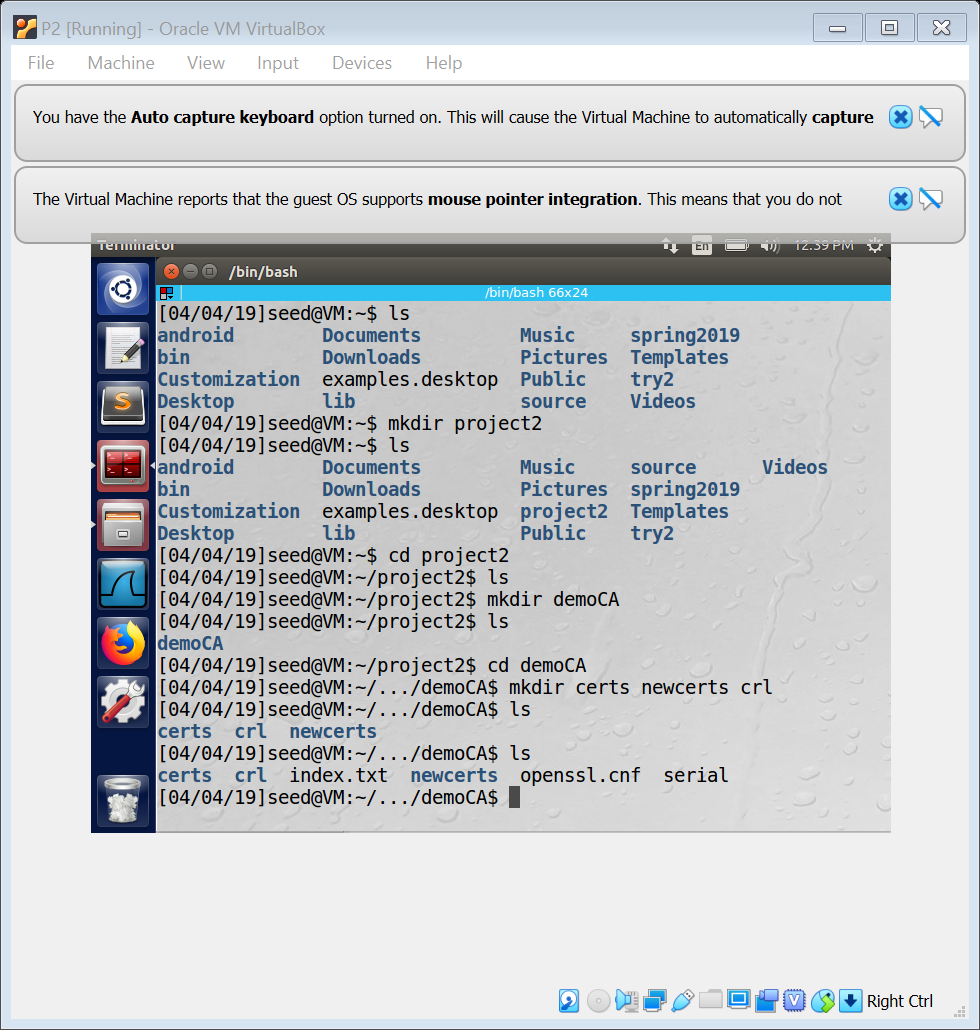


Figure 1. Creating directories.

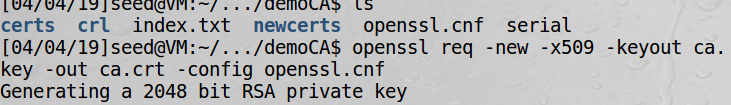


Figure 2. Create certificate authority

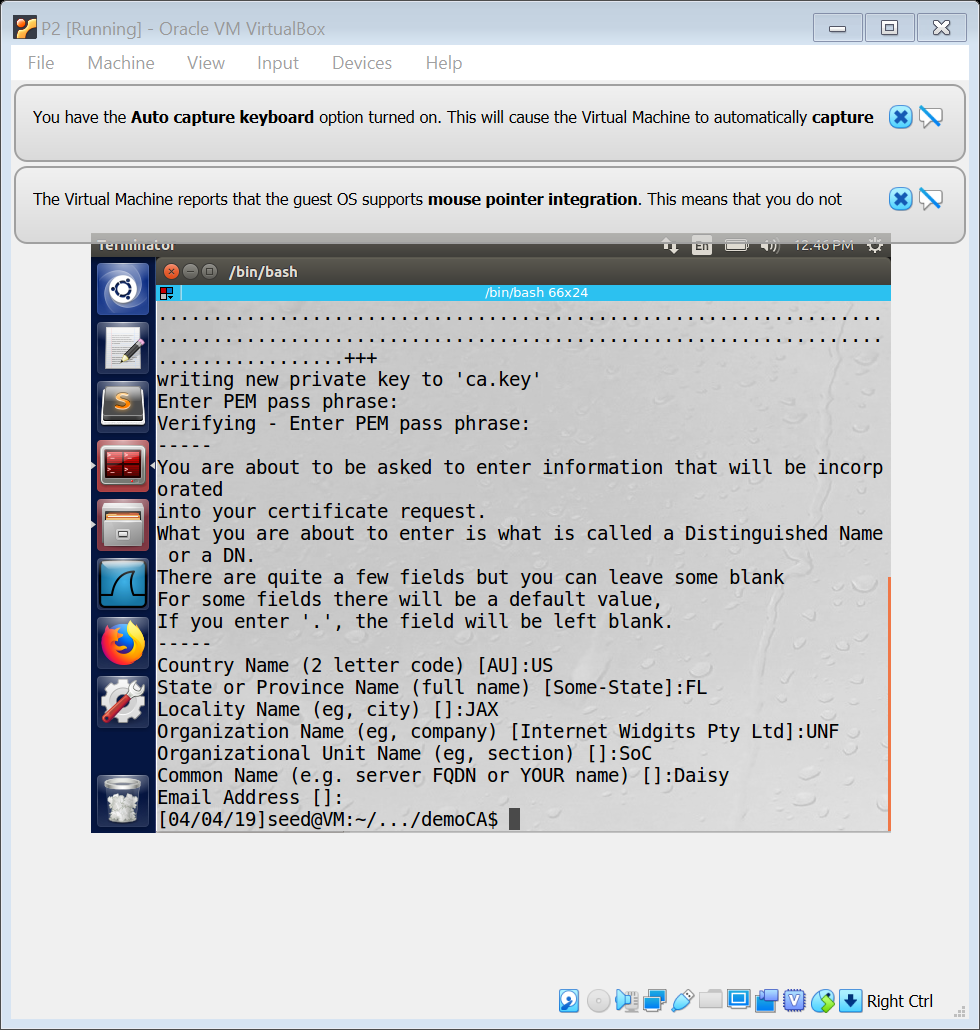


Figure 3. Create private key and fill out certificate.

**Task 2**

In this portion of the lab the objective is to create a certificate for a customer: SEEDPKILab2018.com. The first step is to create a public and private key for the company. In order to do so, the prompts in Figure 4 are used to encrypt the private key and store it as server.key. By running the second prompt the contents of the private key like the modulus and exponents are available. After this is completed, the certificate signing request for the company must be made. The request is generated by the prompt shown in Figure 5 which also shows the result. The pass phrase “cis4360” was entered for the server.key and then the country name, state, etc were entered. The next prompt allows the certificate to be signed. The challenge password of “cis4360” was entered and then there was a prompts to sign the certificate in which the letter y was entered to indicate y. The results are shown in Figure 6. Now, the web server can be configured using the s\_server command. Right before this is done the server key is copied to a file name server.pem and concatenated with the server.crt file. Then, the server is configure as shown in Figure 7. When access to the site was attempted, the image shown in Figure 8 is what was seen.

.

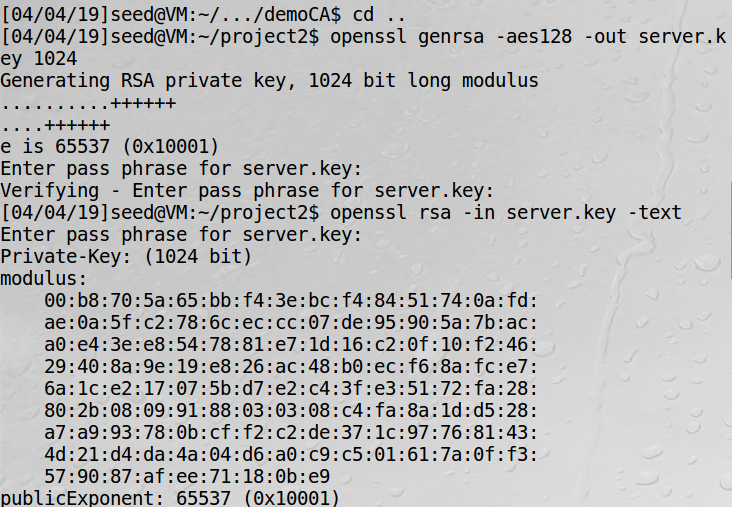


Figure 4. Encrypt private key.

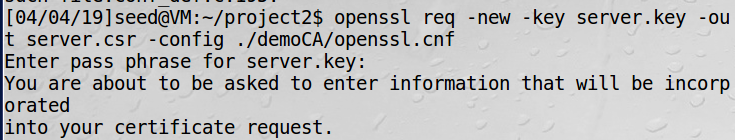


Figure 5. Generate certificate signing request.

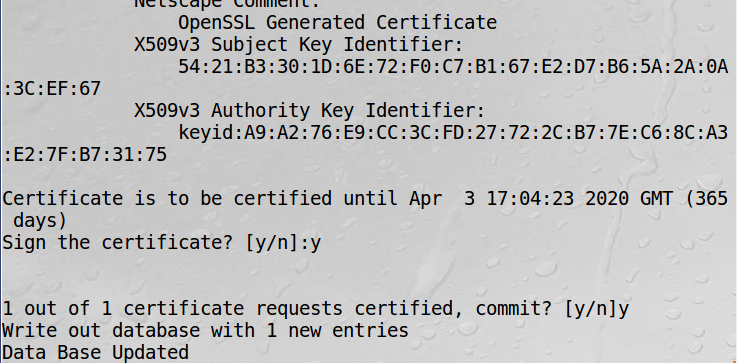


Figure 6. Signing the certificate.

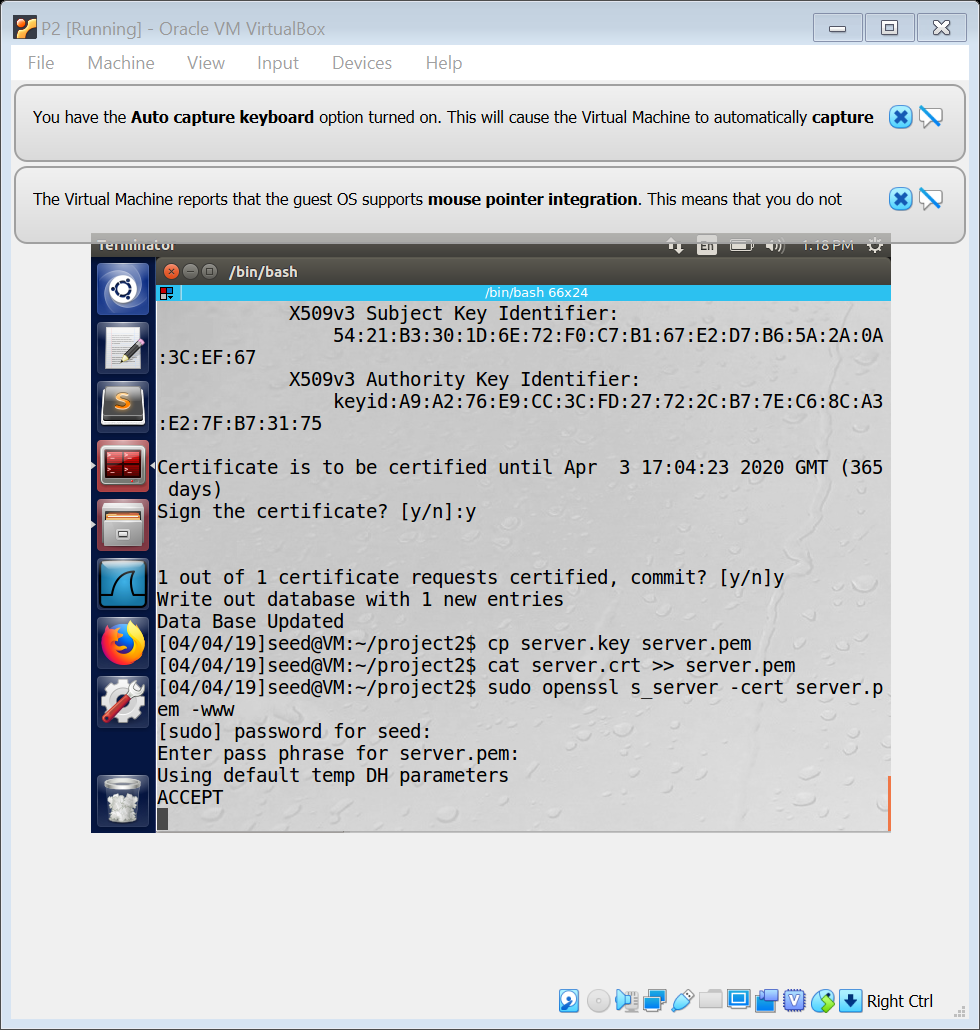


Figure 7. Using s\_server to grant access.

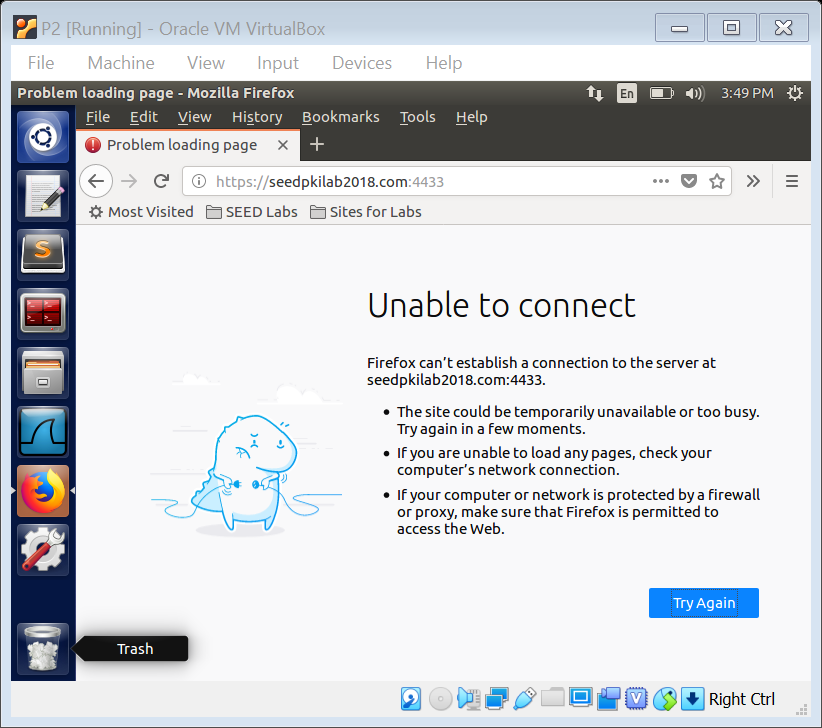


Figure 8. Results of Task 2.

**Task 3**

Task 3 focuses on setting up a website with the ultimate goal of secure web browsing. First, the DNS was configured by adding the website SEEDPKILab2018.com with the IP address 127.0.0.1 in the file named hosts. This was done with the sudo gedit command as seen in Figure 9 which produced the results seen in Figure 10. It was found that adding www caused difficulties, so it was omitted.

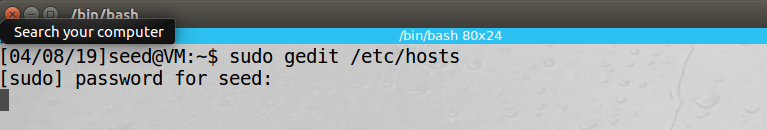


Figure 9. Use sudo gedit command to add website.

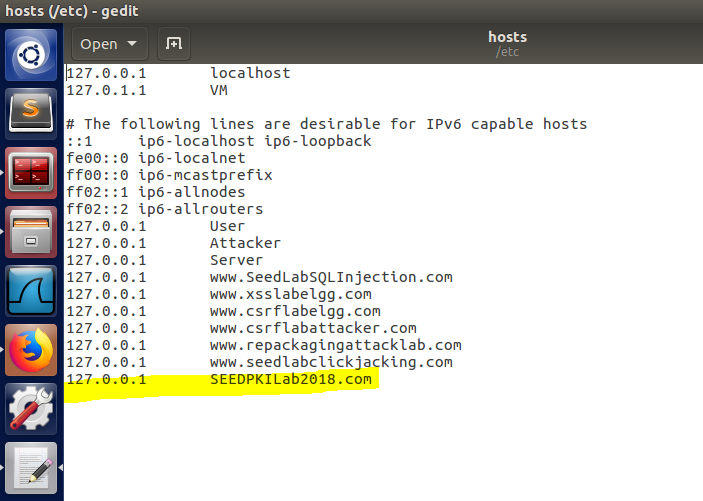


Figure 10. Website has been added to hosts file.

After this was completed the server.key file was copied to the server.pem file. The command cat was used to concatenate the server.crt with server.pem so the server key and certificate information were together in one file. The s\_server command was used and an attempt to access the website. The terminal prompts that were used are shown in Figure 11 and the page generated when accessing the website is shown in Figure 12.

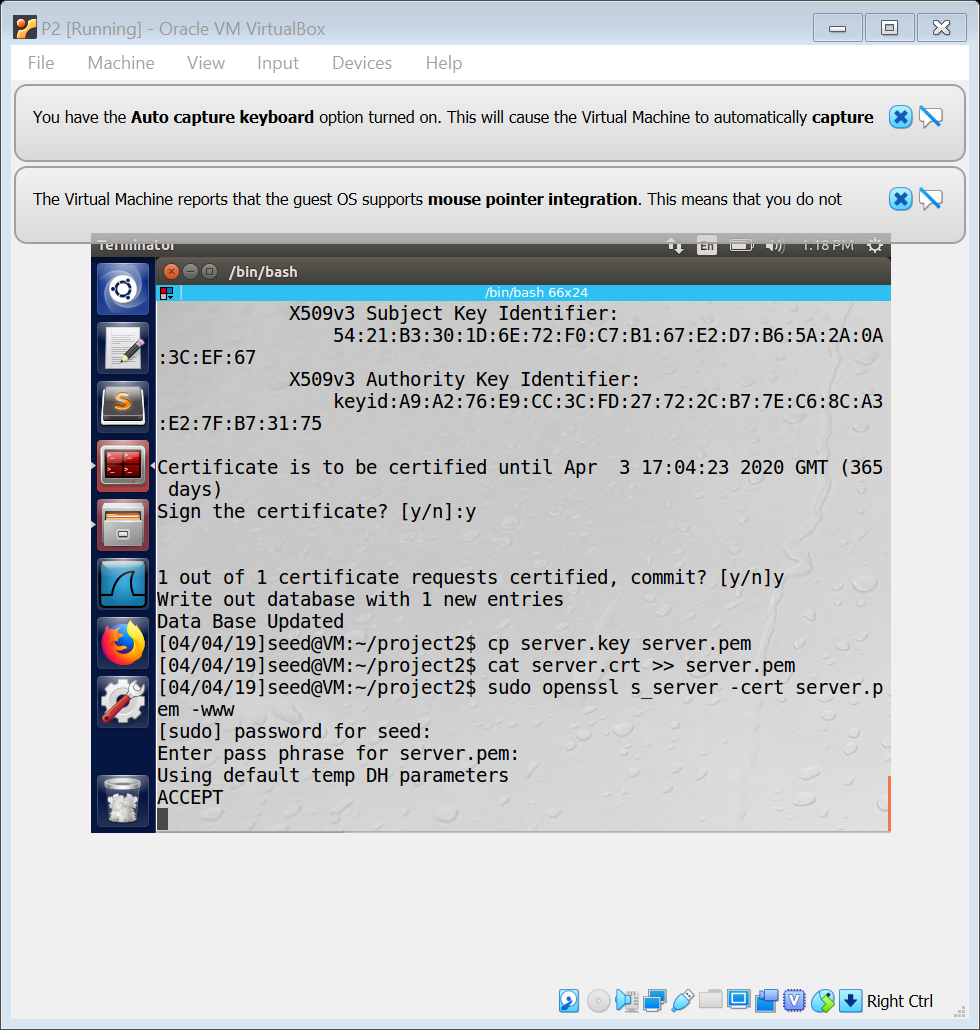


Figure 11. Concatenating and using s\_server command.

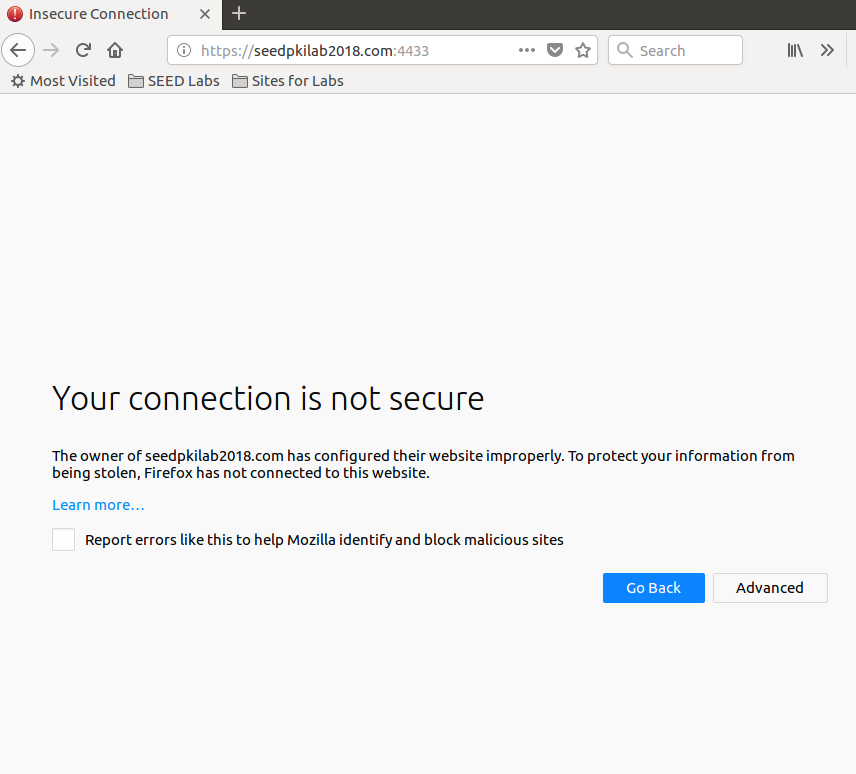


Figure 12. Attempt to access website in Firefox.

When trying to access the site, there is a prompt that warns of an insecure connection. It will connect but access to the page s unavailable because Firefox cannot verify the certificate. In order to use the self-signed certificate that was created in this lab, the certificate must be uploaded to the browser. Under Firefox’s preferences, there is a security section where the certificates can be viewed. The ca.crt file created earlier can be imported to the list of certificates seen in Figure 13 and has the option to be trusted shown in Figure 14. Finally, the server is rebooted with the openssl commands shown in Figure 15.

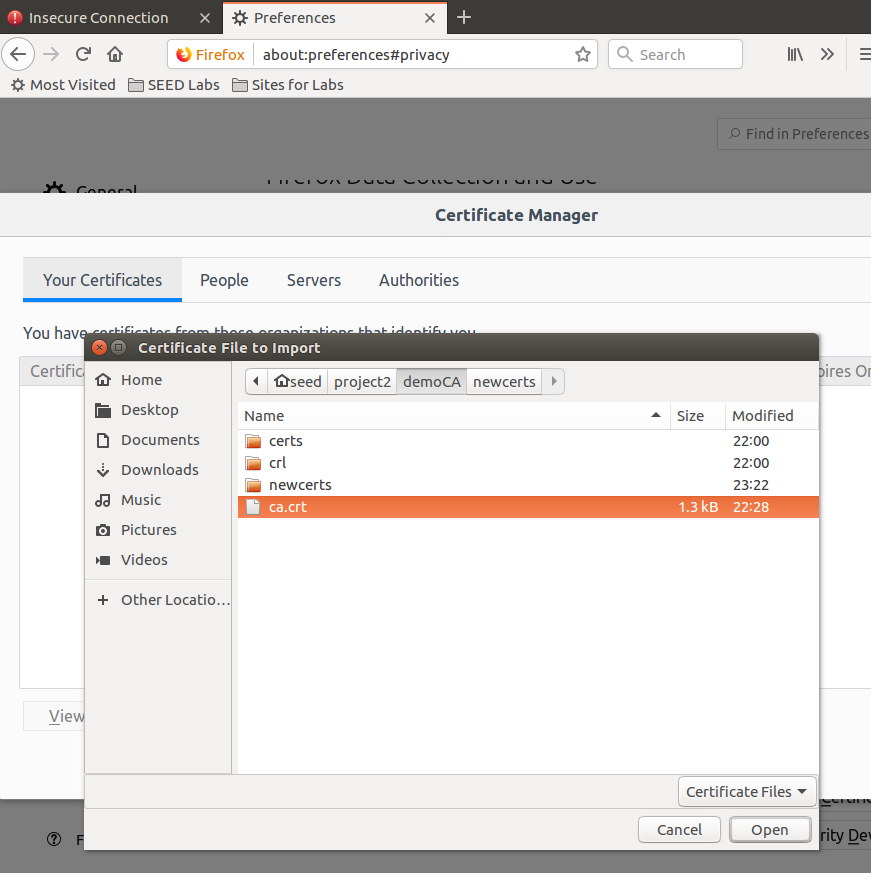


Figure 13. Adding ca.crt file to Firefox certificates.

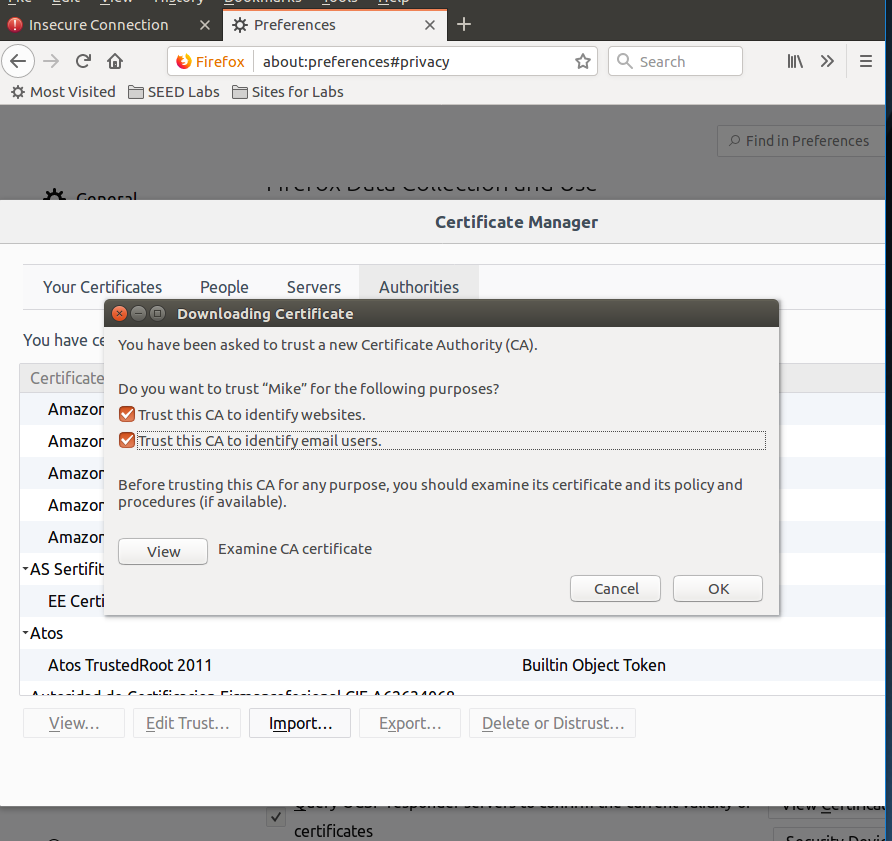


Figure 14. Choosing options to trust the self-signed certificate.

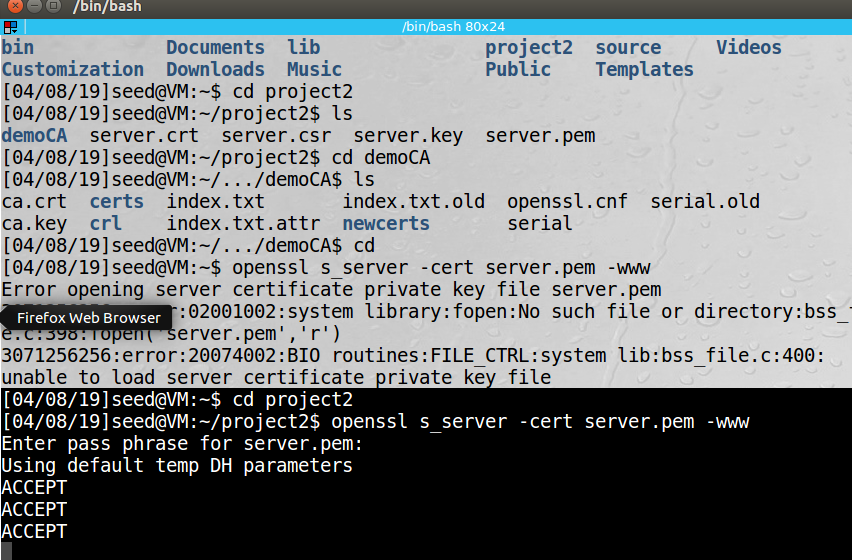


Figure 15. Use openssl to restart server.

Now, when the website is accessed the screen displayed in Figure 16 appears which shows tha the connected is labeled secure. The sudo gedit command shown in Figure 17 was implemented to open and modify a single byte in the server.pem file. In the file the line the S was removed from the line: "Subject Public Key Info:" demonstrated in Figure 18. This in turn removed the "Ciphers common between both SSL endpoints:" section. The file before is shown in Figure 19 and the results of the modification are given in Figure 20.

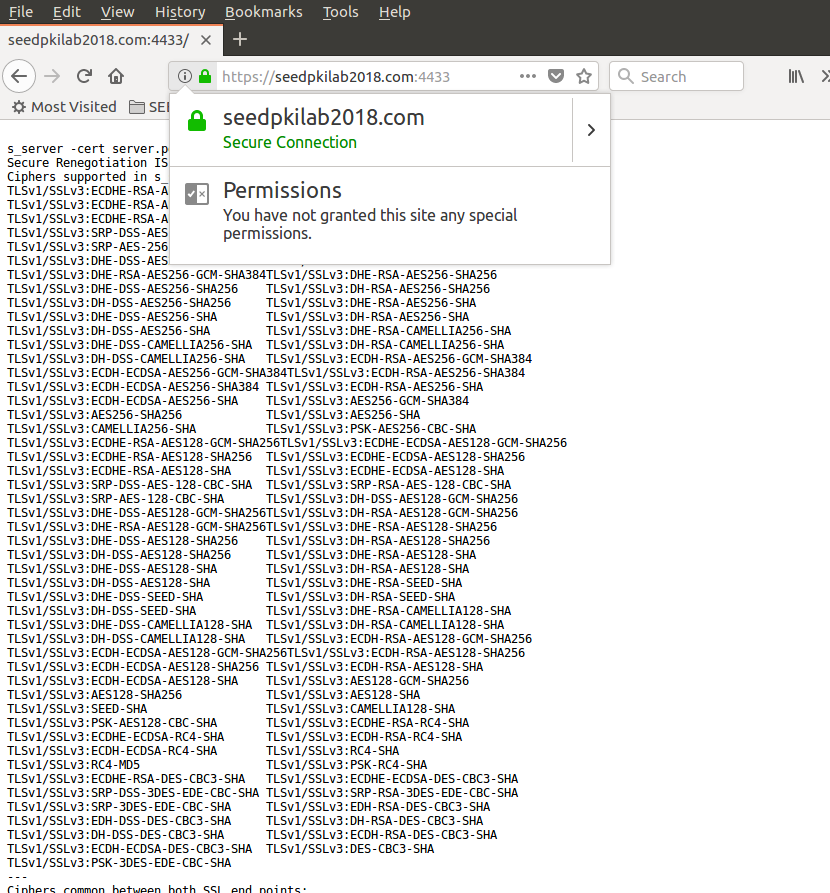


Figure 16. The website appears as a secure connection.

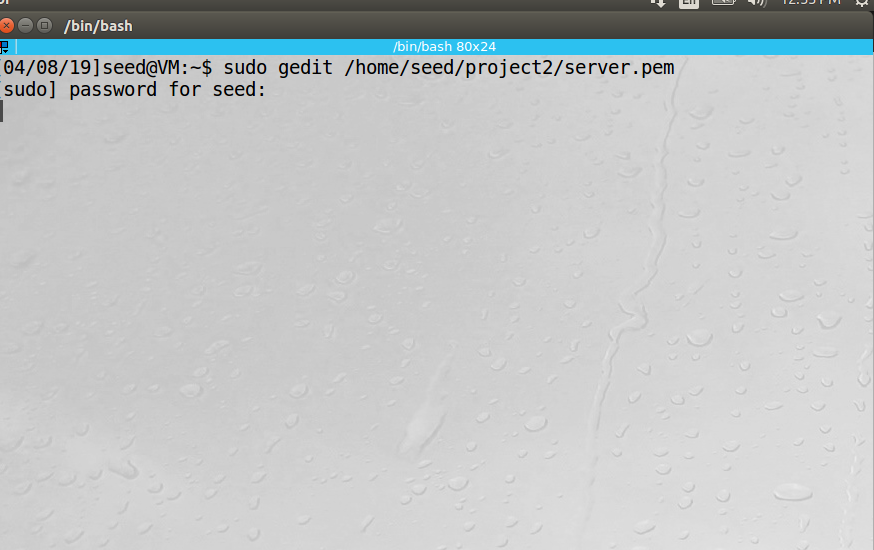


Figure 17. Use sudo gedit to mosify server.pem.

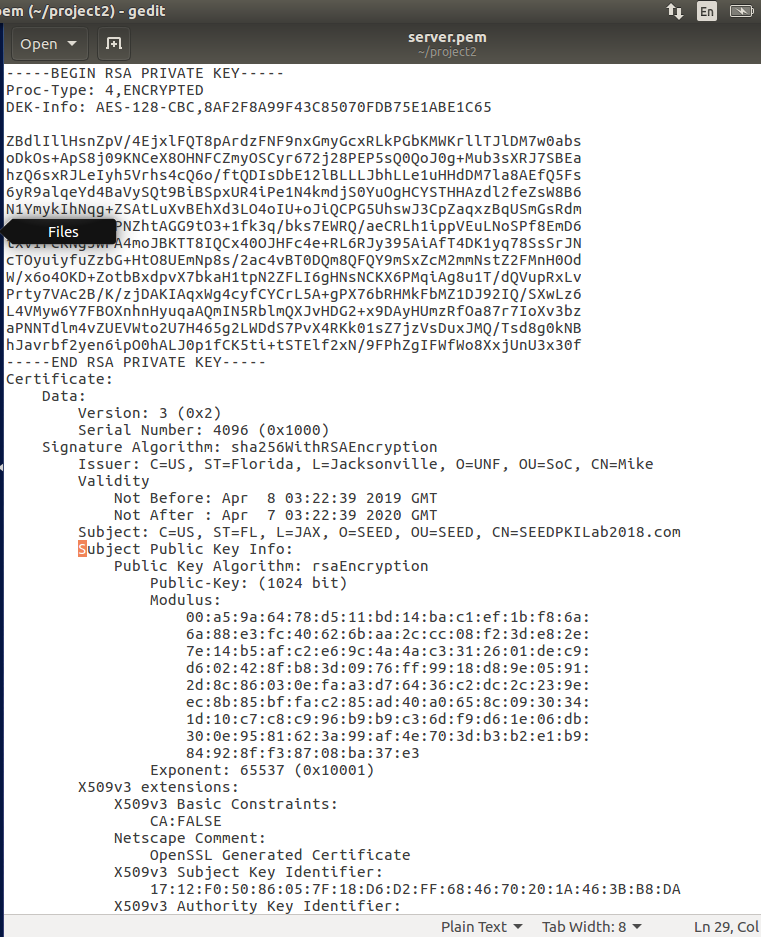


Figure 18. Modification of server.pem file.

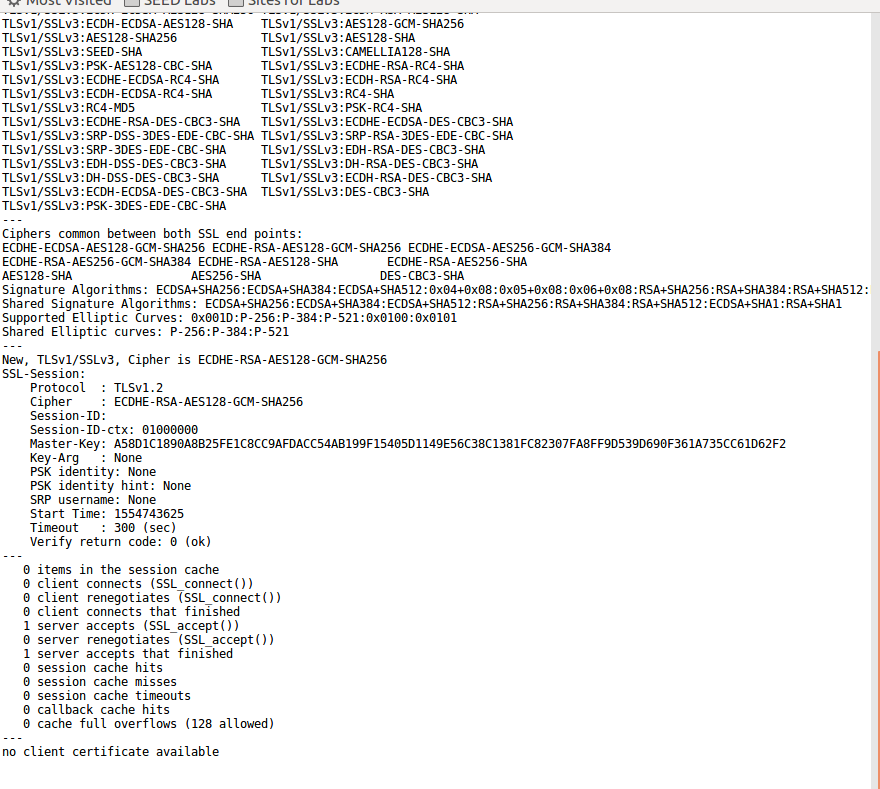




Figure 19. The section of text appears in file.

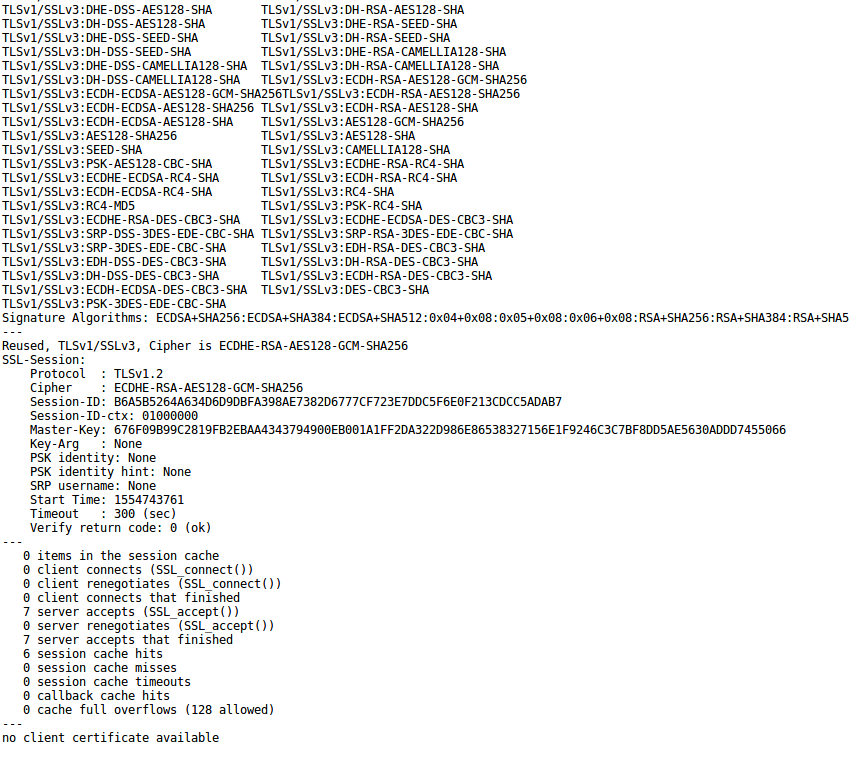




Figure 20. The text does not appear where it was previously.

When the wesite is accessed at this point, the following screen appears shown in Figure 21.

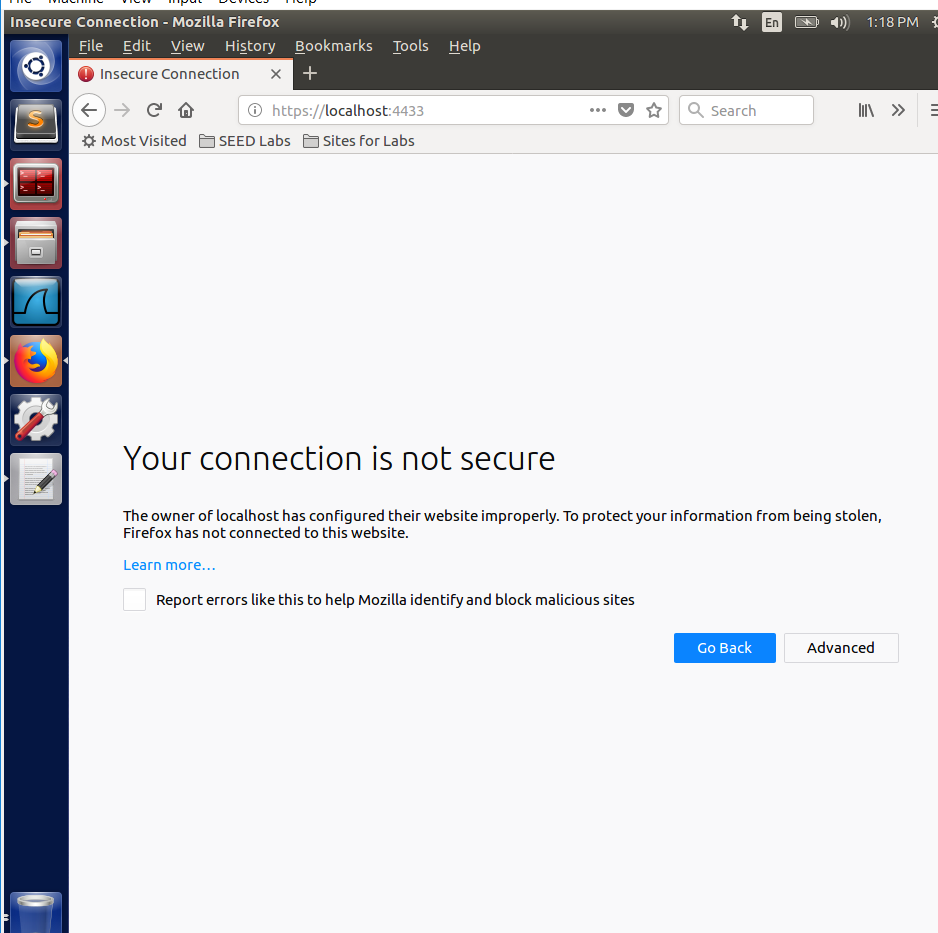


Figure 21. The connection is said to be insecure after file modification.

**Task 4**

In this task, the certificate created in the previous task was implemented to enable HTTPS protocol for the website www.SEEDPKILab2018.com. This must be done to configure the Apache server so that it is able to retrieve the private key and certificate necessary for verification. First, the command sudo gedit was applied to the 000-default.conf file in /etc/apache2/sites-available using the terminal as seen in Figure 22. This provides the VirtualHost with entry for the HTTP site. Then, sudo gedit was applied to the file default-ssl.conf in the sites-available as well shown in Figure 23 which is also required to add the VirtualHost entry for the HTTPS site. The figures show where the files for the site is stored under DocumentRoot and the website’s name under ServerName.

In the terminal, the command: sudo apachectl configtest to check the VirtualHost entries entered previously to make sure the directory paths to the server key and certificate are correct and functional. The command: sudo a2enmod ssl was used to enable the ssl module. To activate the site the following command was typed into the terminal: sudo a2ensite default-ssl. At last, the server was restarted by typing sudo service apache2 restart in the terminal. If typed correctly and everything is properly configured, the terminal should prompt for the passphrase for SSL/TLS keys for the site which was set as: cis4360. All the commands listed were ran and shown in Figure 24. Figure 25 presents what is seen when the site is properly booted up and operational.

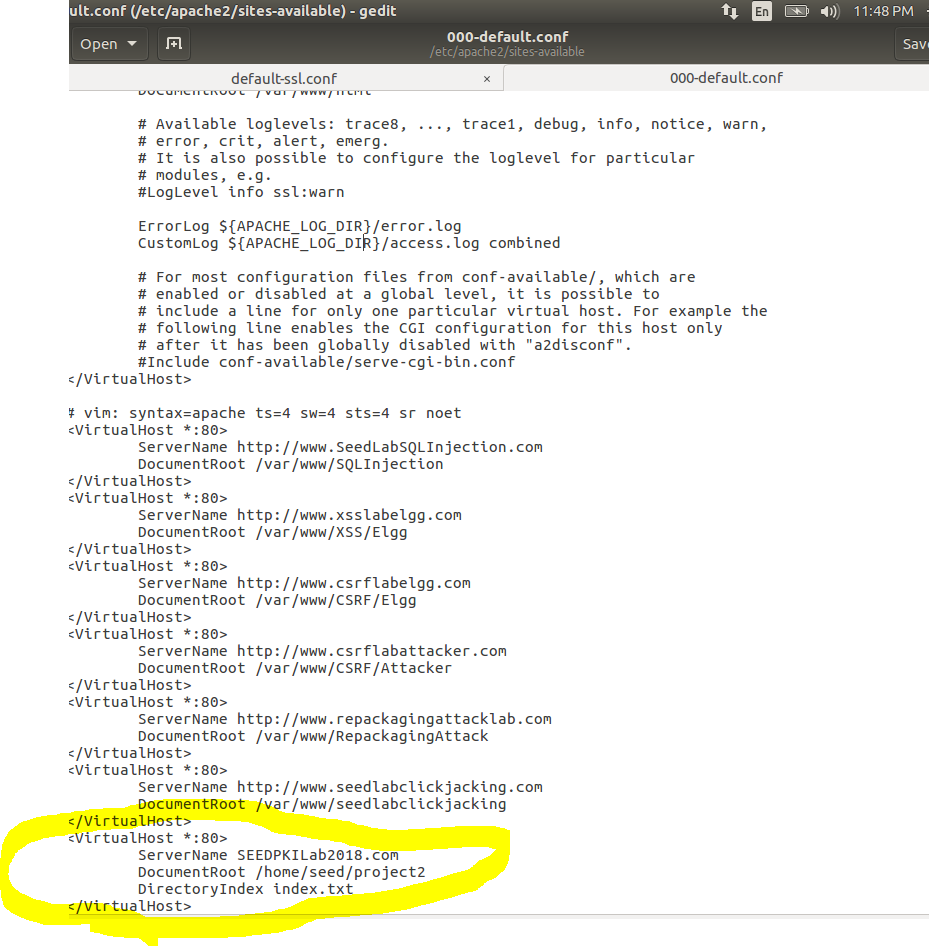


Figure 22. Using file 000-default.conf to gain entry to HTTP site.

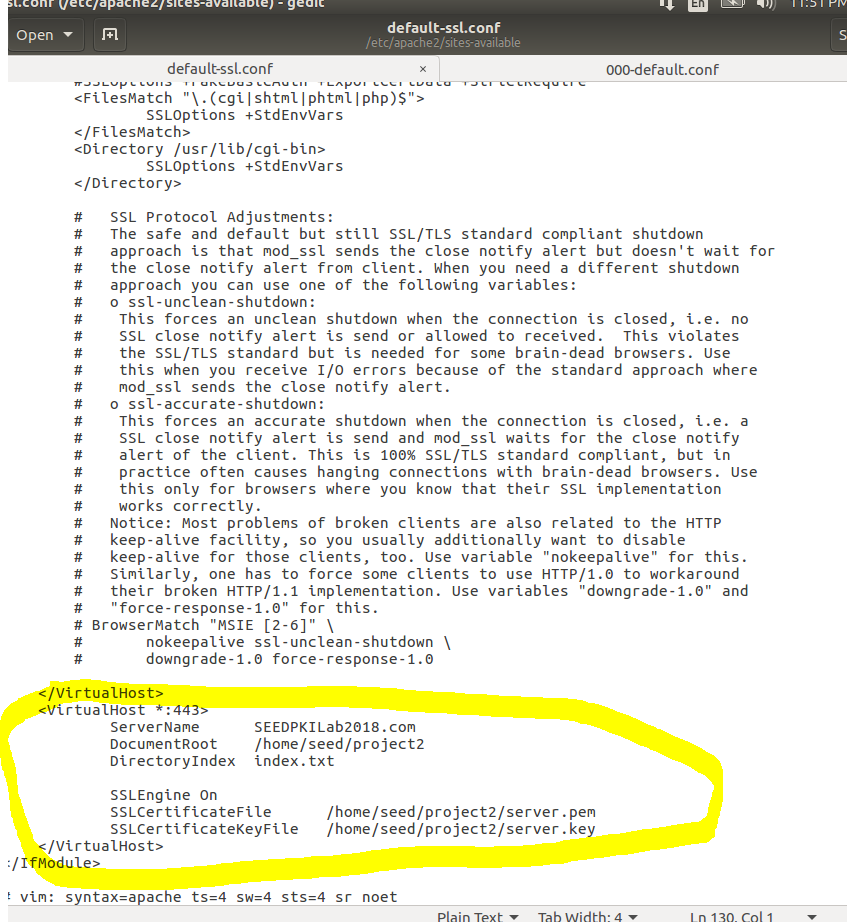


Figure 23. Using file default-ssl.conf to gain entry to HTTP site.

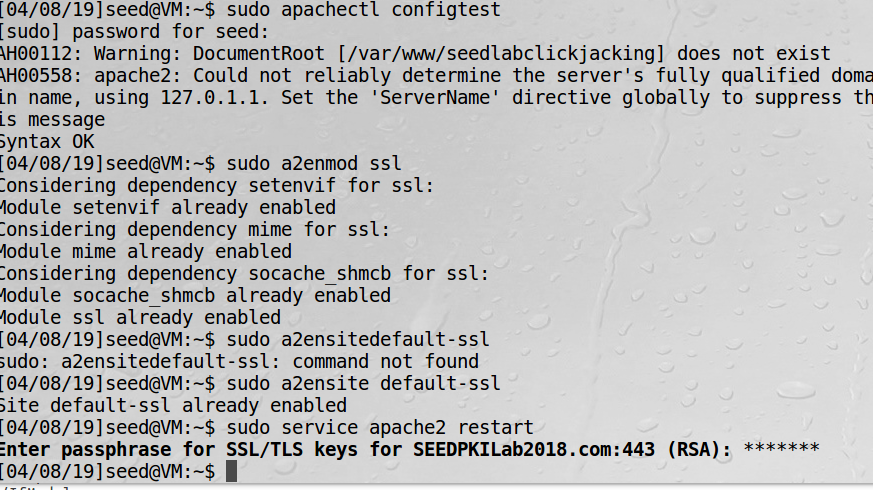


Figure 24. Commands implemented to enable SSL.

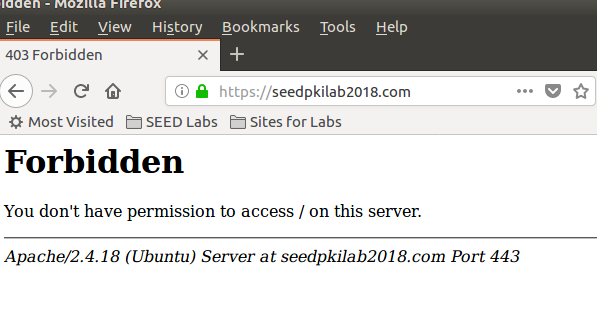


Figure 25. The result given by Task 4.

**Task 5**

This task focused on carrying out a Man-In-The-Middle attack. To begin, another VirtualHost entry in the file 000-default.conf where the ServerName is changed to example.com as directed in the lab manual and shown in Figure 26. Figure 27 depicts the file default-ssl.conf, where the same was done for the HTTPS server; the ServerName was changed to example.com. The command shown in Figure 28, was used to make sure the IP address of example.com is indeed different than what it was changed to. In order to simulate a DNS attack, the IP routing of example.com was set to 127.0.0.1 as done in Figure 29. Figure 30 and 31 present the differences between the two websites.

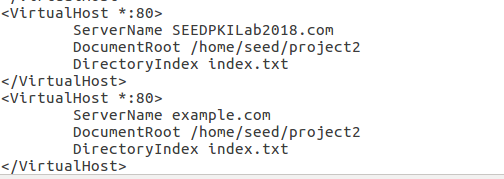


Figure 26. Adding example.com as VirtualHost.

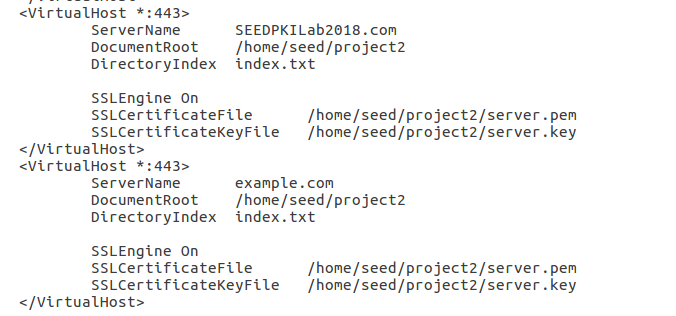


Figure 27. Change ServerName to example.com.

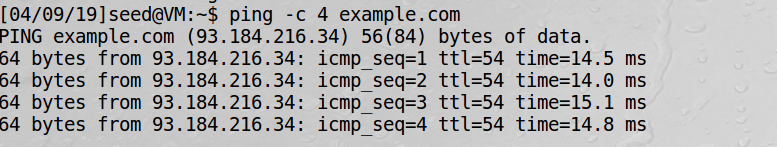


Figure 28. Use ping command to verify IP address.

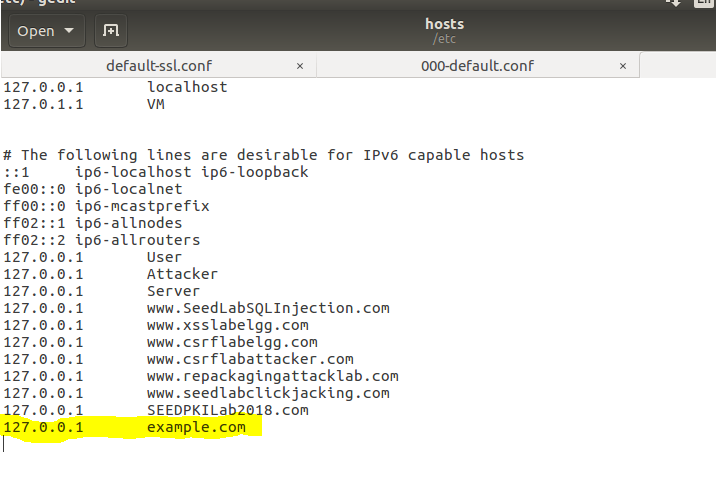


Figure 29. Route example.com to IP address.

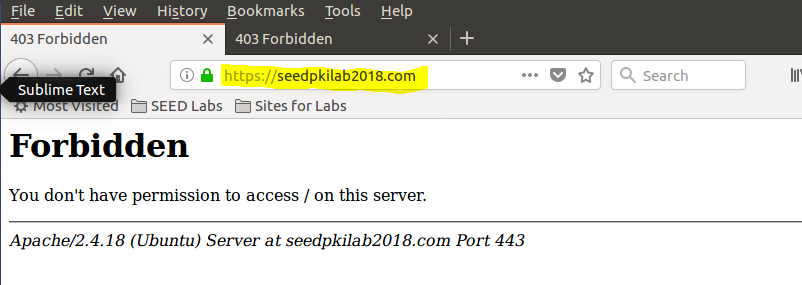


Figure 30.

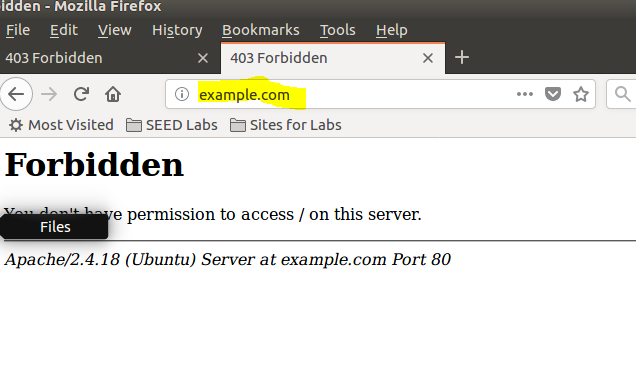


Figure 31.

**Task 6**