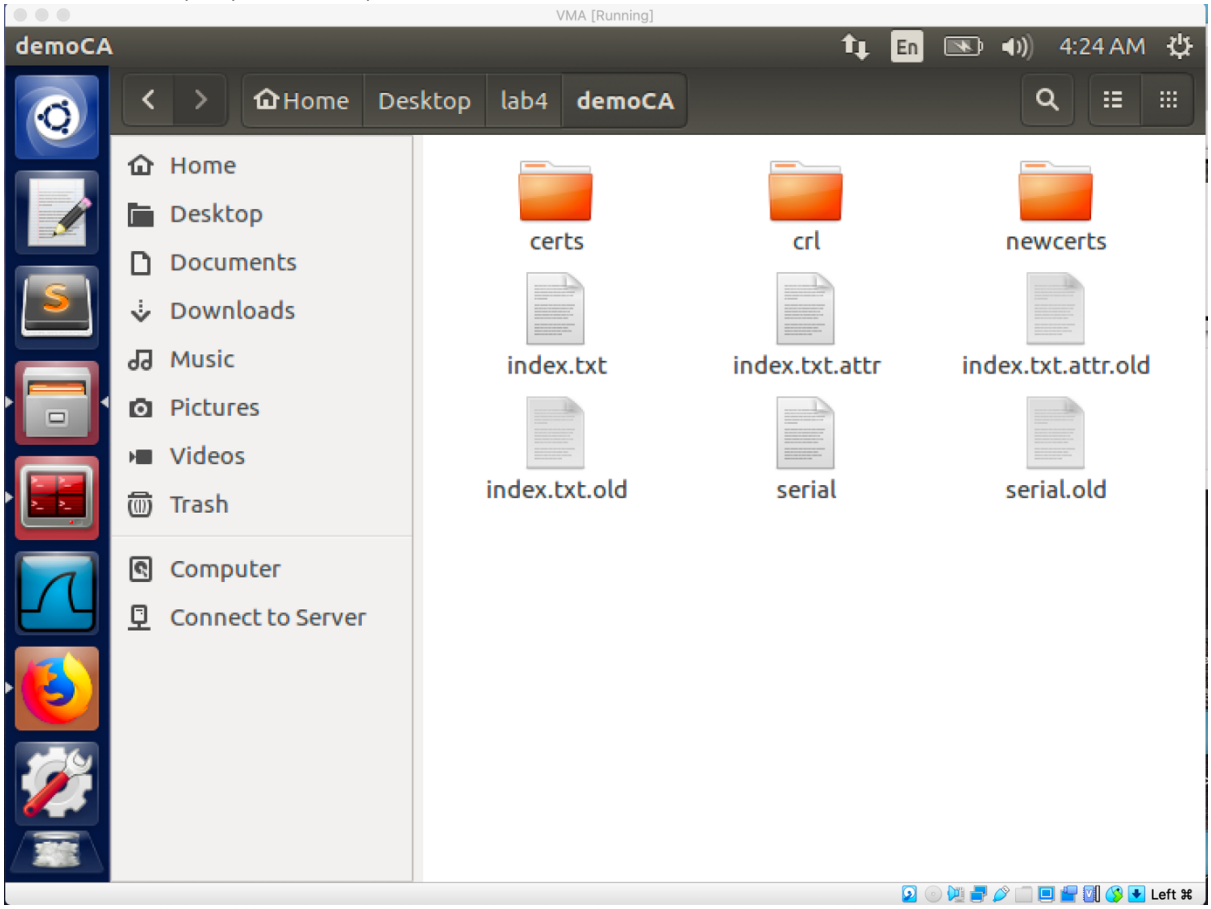
Task1: Becoming a Certificate Authority (CA)

In order to use OpenSSL to create certificates we need a config file which is openssl.conf, I make a copy of openssl.conf to my current directory which is '/home/seed/Desktop/lab4' and I named it by openssl1.com. I also create a sub-directory which is called demoCA, it contains certs, ctl, newcerts, index.txt and serial.



After create the directory, I use the command ' openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl1.cnf' to generate a self-signed certificate for the CA. Two file will be generated, ca.key contains the private key of the CA, ca.crt contains the public key of the CA.

```
-----
Country Name (2 letter code) [AU]:Singapore
string is too long, it needs to be less than  2 bytes long
Country Name (2 letter code) [AU]:SG
State or Province Name (full name) [Some-State]:Singapore
Locality Name (eg, city) []:singapore
Organization Name (eg, company) [Internet Widgits Pty Ltd]:aaron
Organizational Unit Name (eg, section) []:aaron
Common Name (e.g. server FQDN or YOUR name) []:zewen.com
Email Address []:www.zewen.com
[02/26/21]seed@VM:~/.../lab4$ ls
ca.crt  certs  index.txt  openssl1.cnf
ca.key  crl    newcerts   serial
```

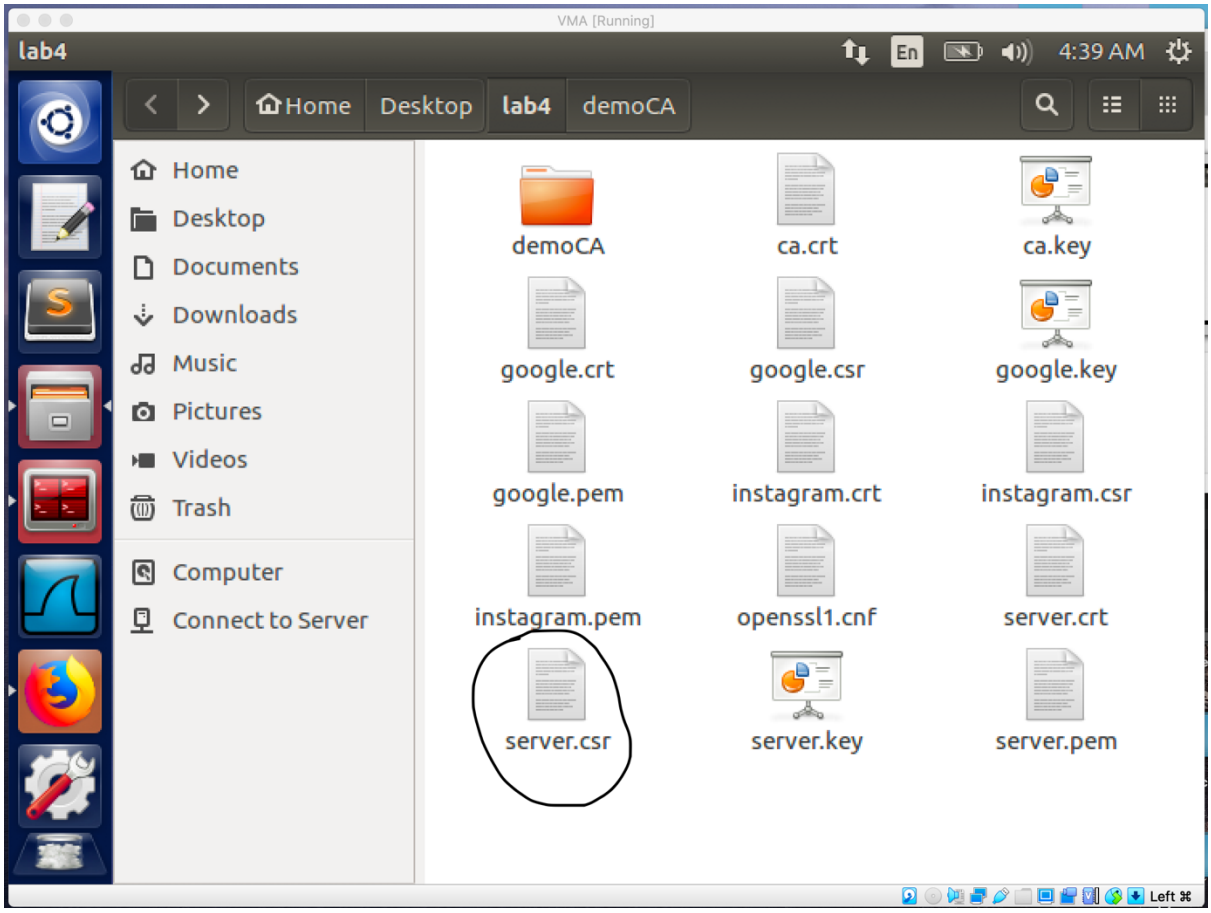
Task2: Creating a Certificate for SEEDPKILab2020.com

Step1:

I use command ‘ openssl genrsa -aes128 -out server.key 1024’ to generate a public/private key pair which encrypted by aes128 algorithm. A server.key file will be outputted. We can check the modulo, exponent by using command ‘openssl rsa -in server.key -text’.

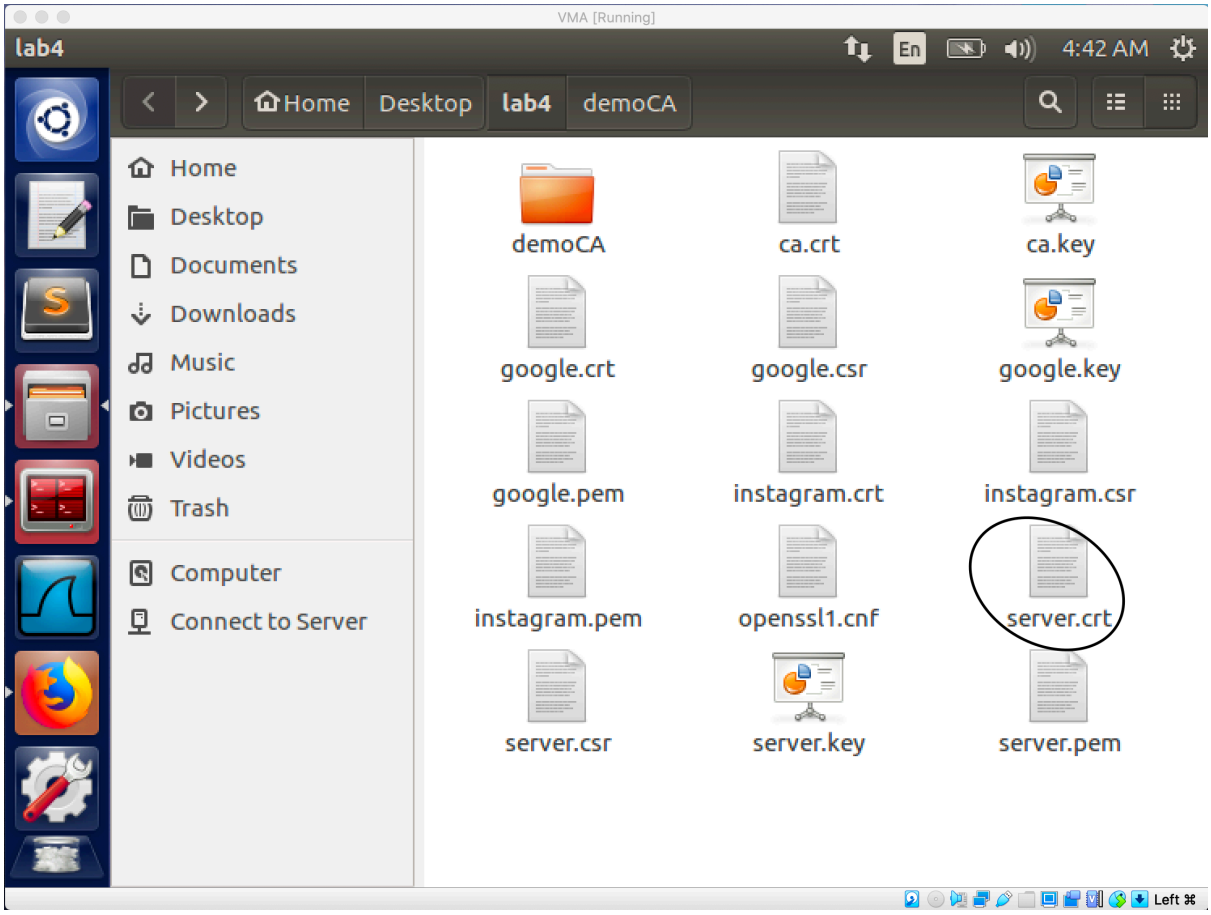
Step2:

I use command ‘openssl req -new -key server.key -out server.csr -config openssl1.cnf’ to generate a request to sign file for ‘SEEDPKILab2020.com’. A ‘server.csr’ file will be generated.



Step3:

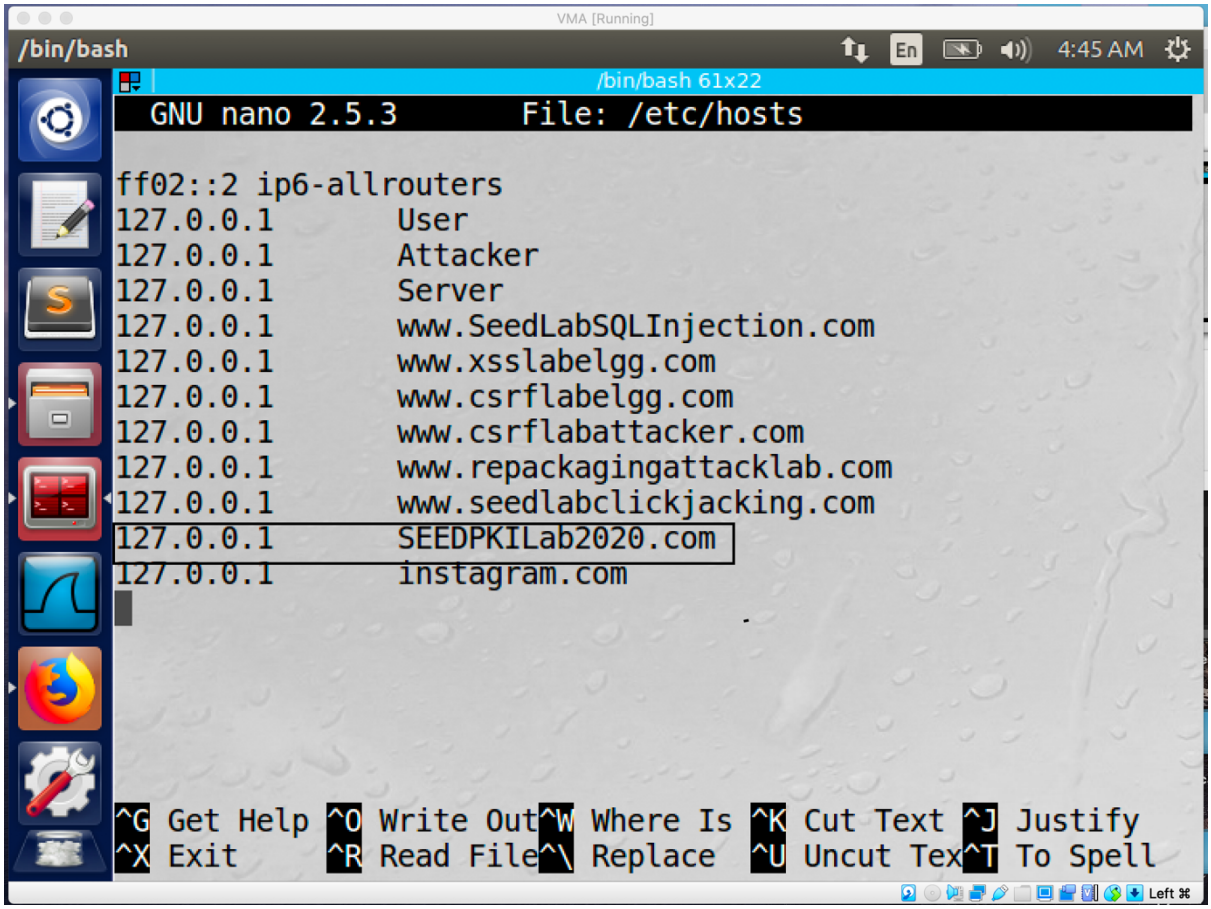
I use command ‘ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl1.cnf’ to send the csr file to the CA and ask for signing the certificate. A file called server.crt will be generated which is the signed certificate of our server.



Task3: Deploying Certificate in an HTTPs web server

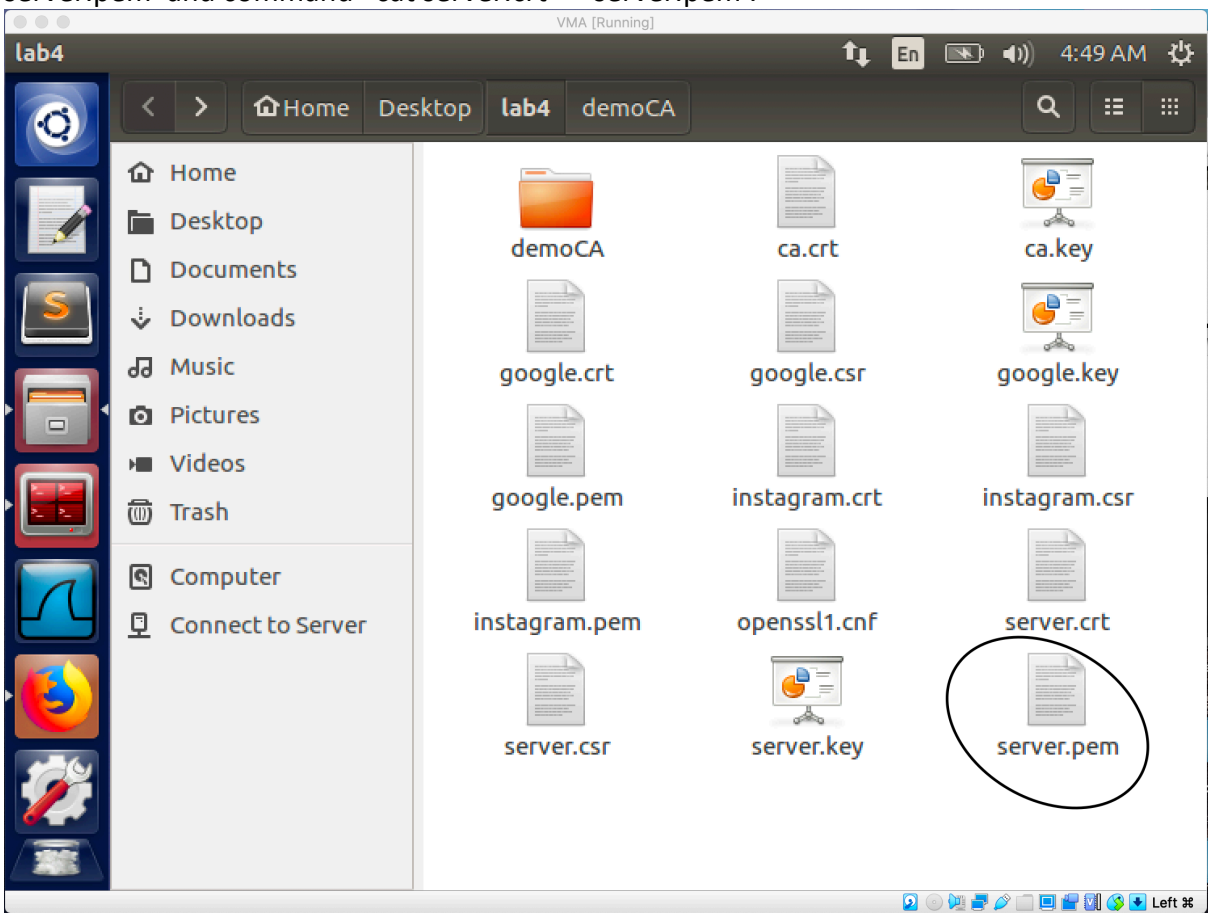
Step1:

We choose SEEDPKILab202.com as the name of our website. To get the VM recognize this name, I add the ‘ 127.0.0.1 SEEDPKILab2020.com’ entry to my ‘/etc/hosts’ file.



Step2: Configuring the web server

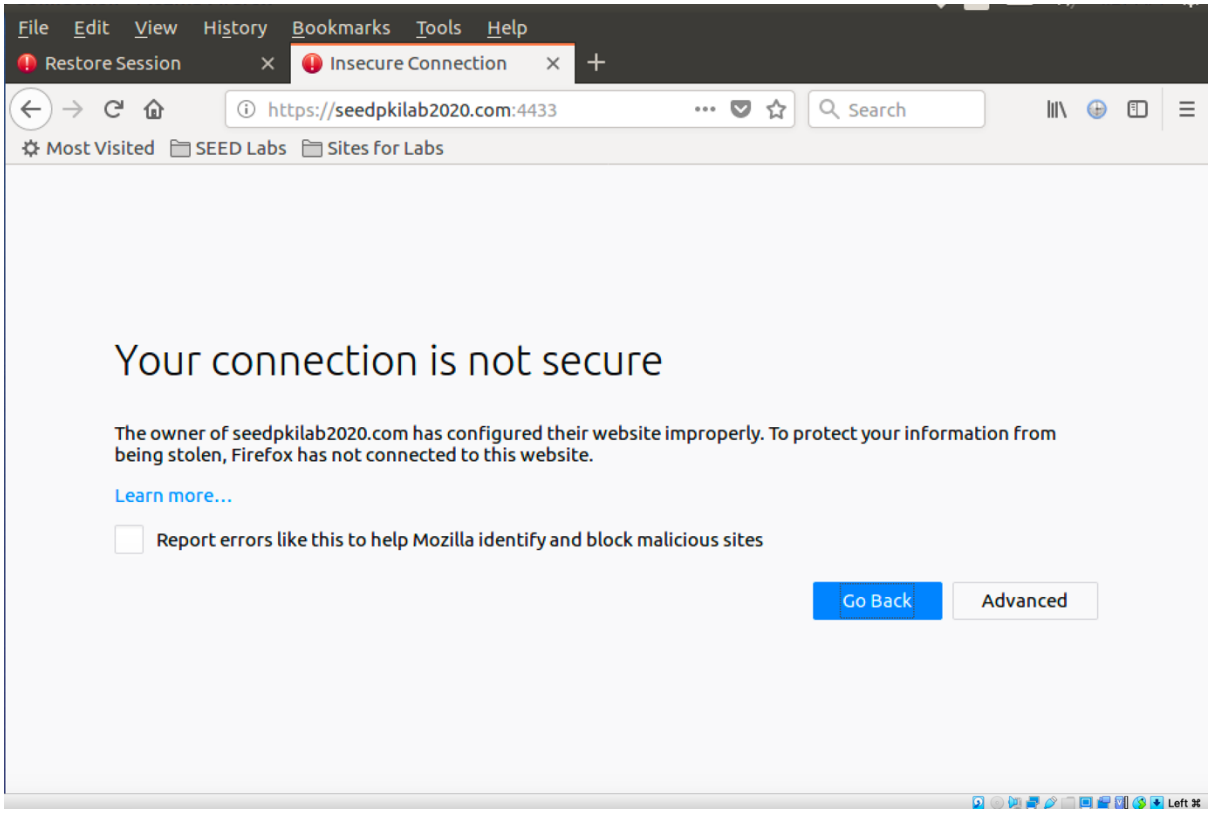
I combine the secret key and certificate into one file by using command ‘cp server.key server.pem’ and command ‘ cat server.crt >> server.pem’.



After that I use the command ‘ openssl s_server -cert server.pem -www’ to start a simple web server.

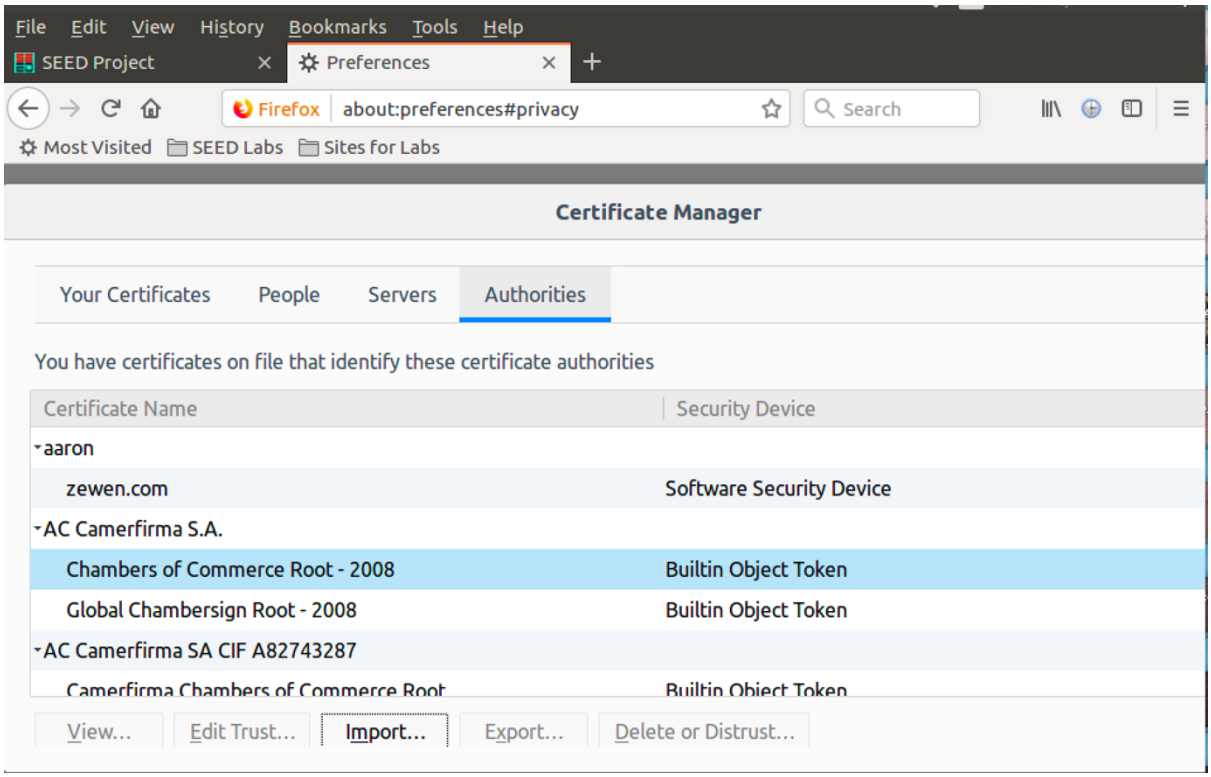
```
[02/27/21]seed@VM:~/.../lab4$ ls
ca.crt  ca.key  demoCA  openssl1.cnf  server.crt  server.csr  server.key
[02/27/21]seed@VM:~/.../lab4$ sudo nano /etc/hosts/
[02/27/21]seed@VM:~/.../lab4$ sudo nano /etc/hosts
[02/27/21]seed@VM:~/.../lab4$ cp server.key server.pem
[02/27/21]seed@VM:~/.../lab4$
[02/27/21]seed@VM:~/.../lab4$

[02/27/21]seed@VM:~/.../lab4$
[02/27/21]seed@VM:~/.../lab4$
[02/27/21]seed@VM:~/.../lab4$
[02/27/21]seed@VM:~/.../lab4$ cat server.crt >> server.pem
[02/27/21]seed@VM:~/.../lab4$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
ACCEPT
```

By searching the URL ‘ SEEDPKILab2020.com’ on Firefox I was blocked because the certificate is self-signed.



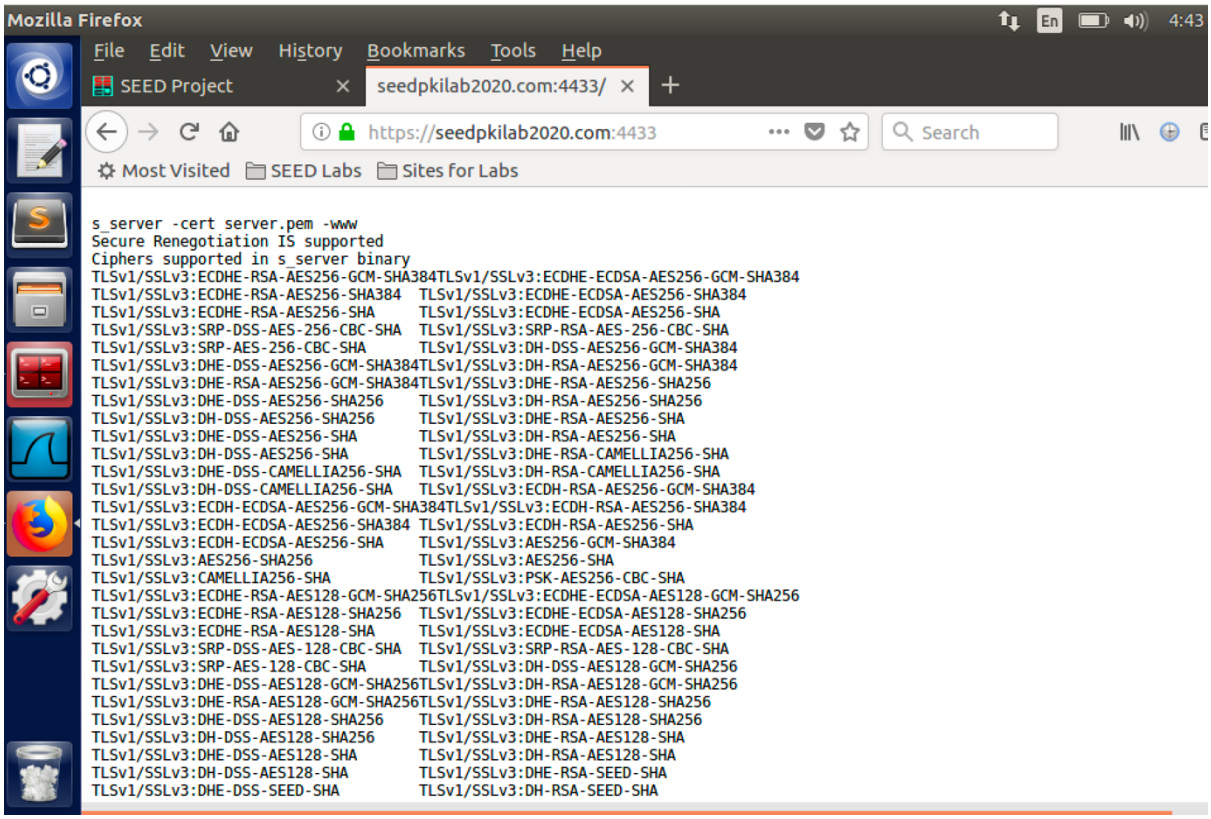
Step3: Getting the browser to accept our CA certificate

Since our self-signed is not include into our Firefox server, I manually add the ca.crt to the Firefox.

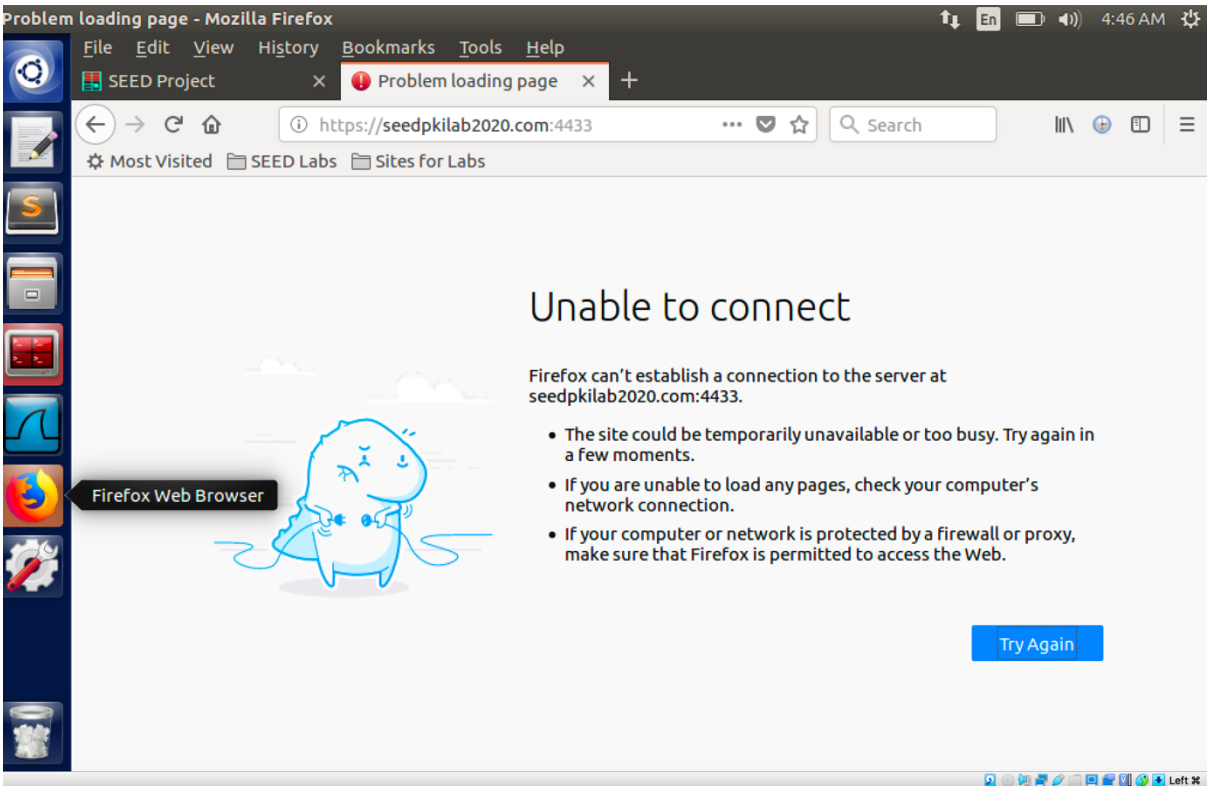


Step4: Testing our HTTPS website

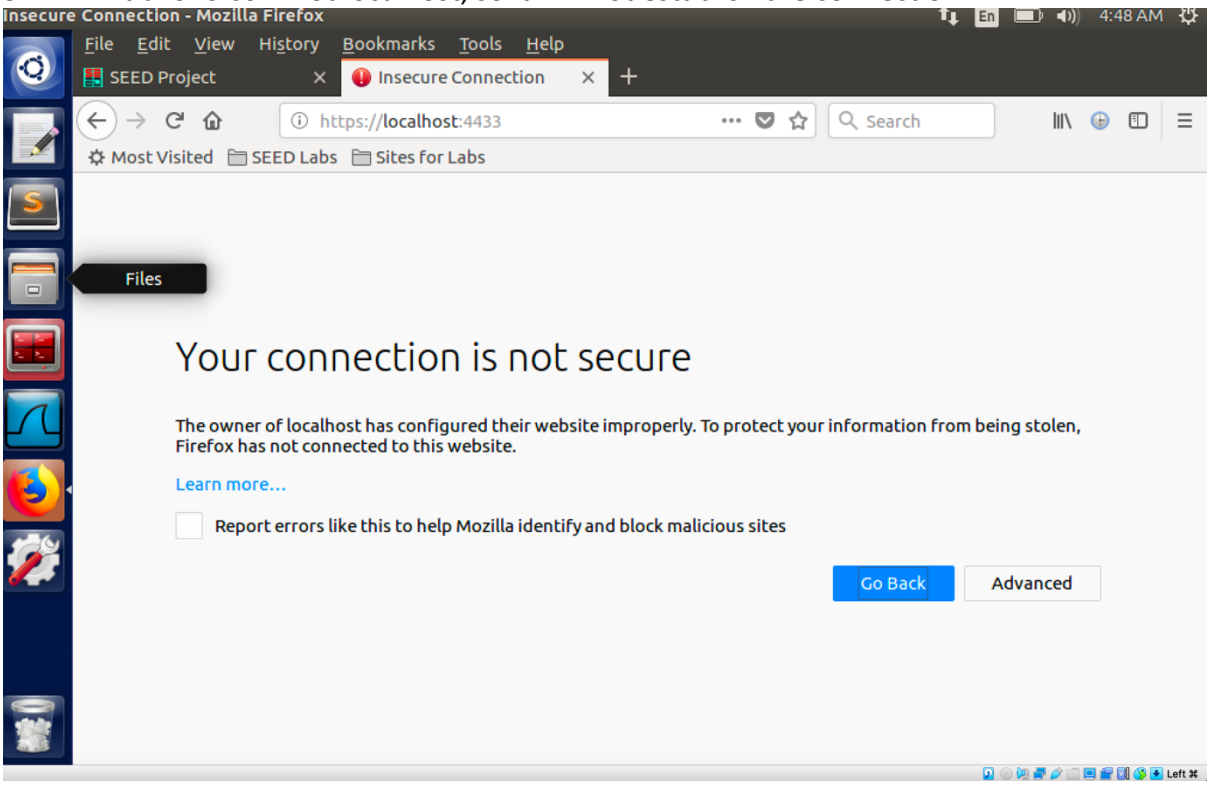
After adding the ca.crt to the Firefox server, I am able to visit the website.



- When I modify one byte of the server.pem file, I am not able to connect to the website. It is because the server.pem is corrupted so the web browser cannot authenticate the server's certificate.

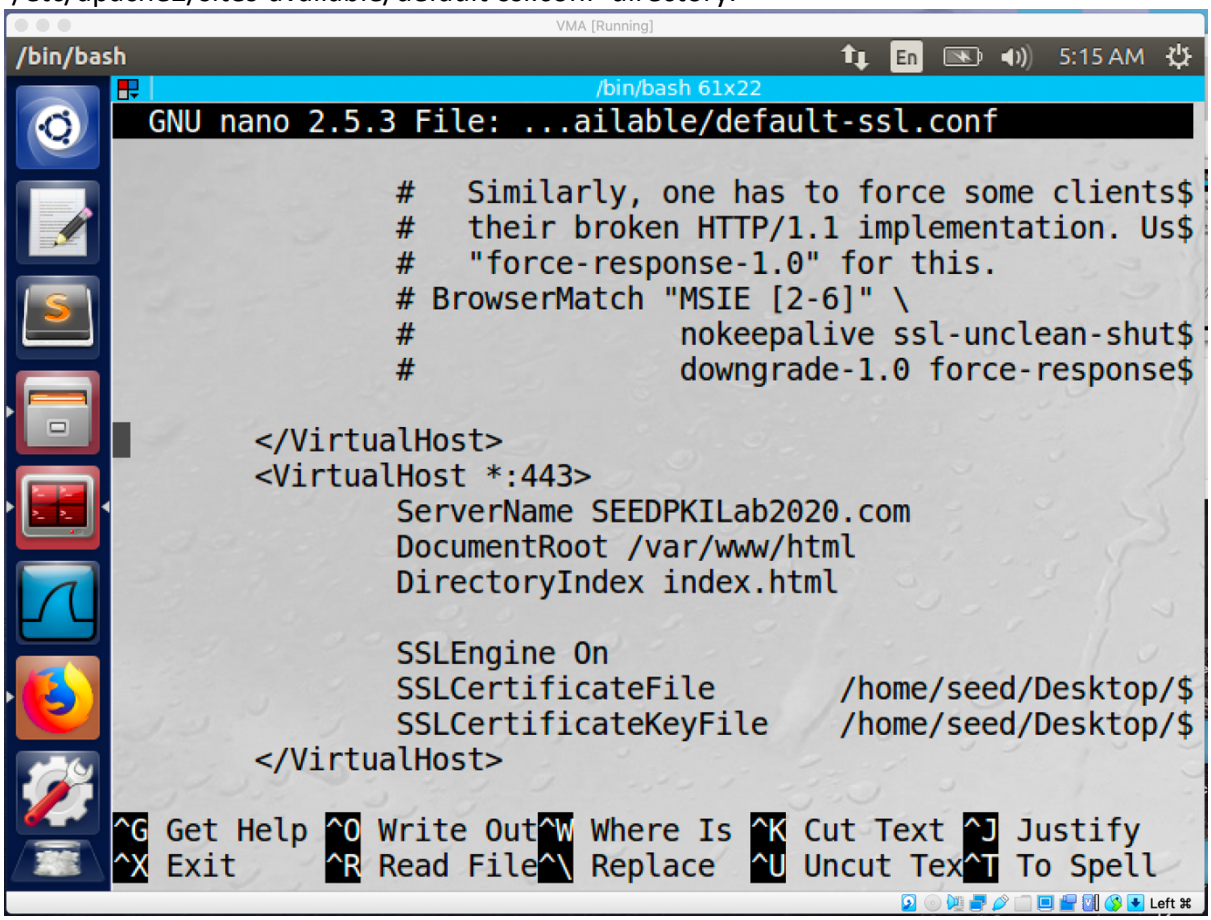


- If I change the URL to ‘localhost:4433’ I am not able to connect to the website, since Firefox will check the CN in the certificate, it will find out the CN is SEEDPKILab2020.com not localhost, so it will not establish the connection.



Task4: Deploying Certificate in an Apache-Based HTTPS Website

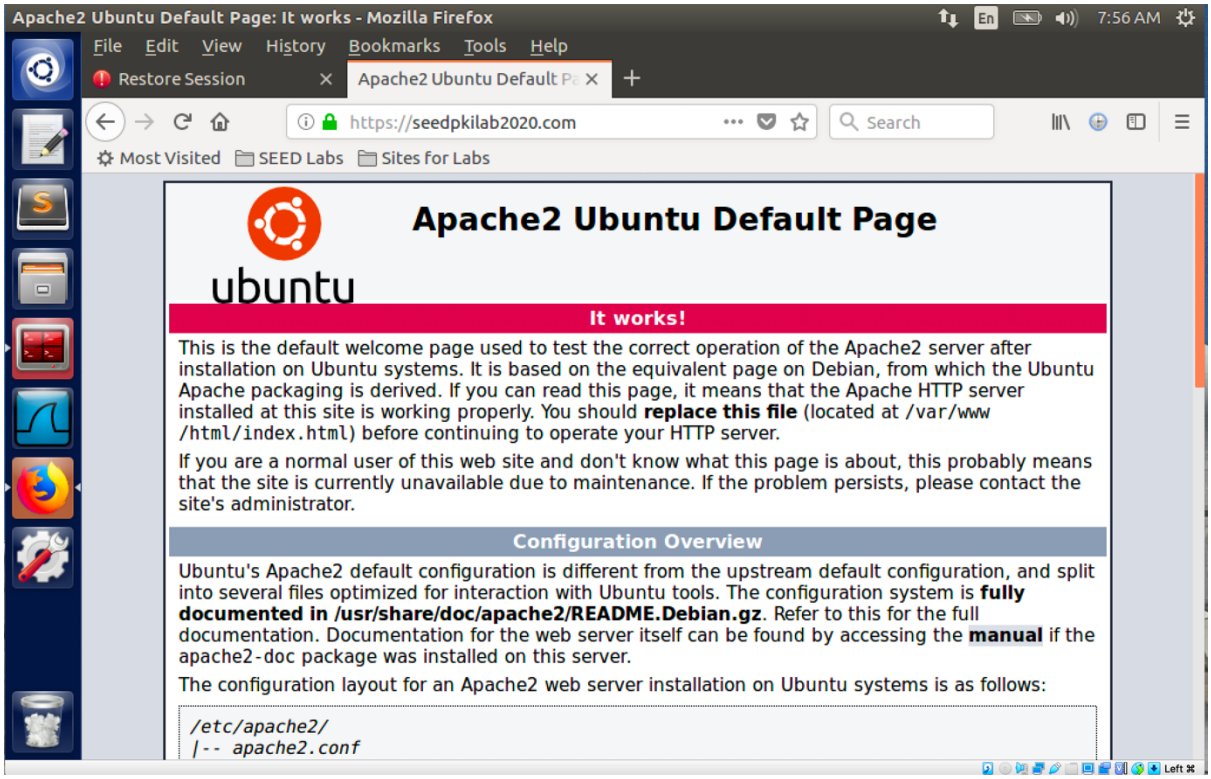
In order to add a HTTPS website to the server, I add a HTTPS web server to the '/etc/apache2/sites-available/default-ssl.conf' directory.



After add the server to default-ssl.conf file, I run server command to restart the service.

```
// Test the Apache configuration file for errors
$ sudo apachectl configtest

// Enable the SSL module
$ sudo a2enmod ssl

// Enable the site we have just edited
$ sudo a2ensite default-ssl

// Restart Apache
$ sudo service apache2 restart
```

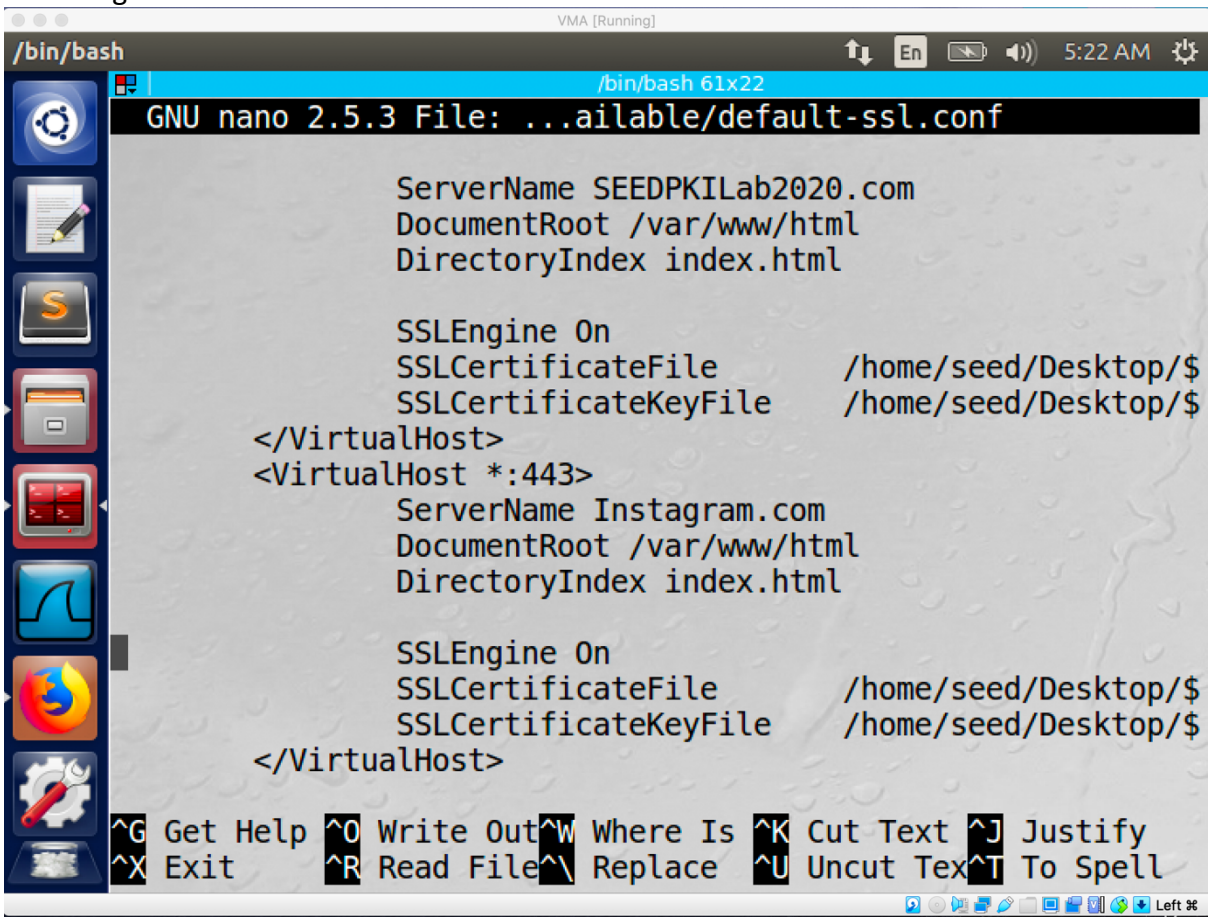
Then I am able to visit the SEEDPKILab2020.com under https request.



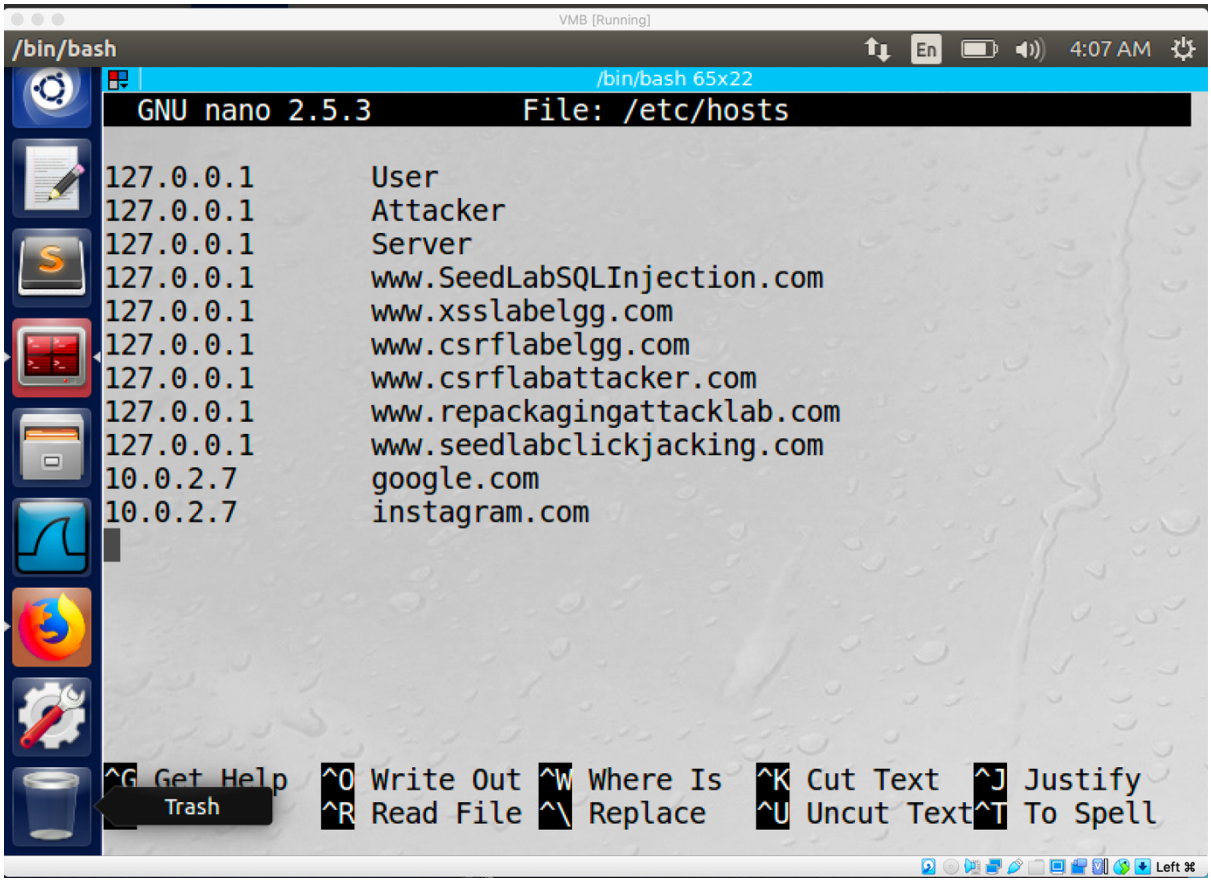
Task5: Man-In-The-Middle Attack

Step1: Setting up the malicious website.

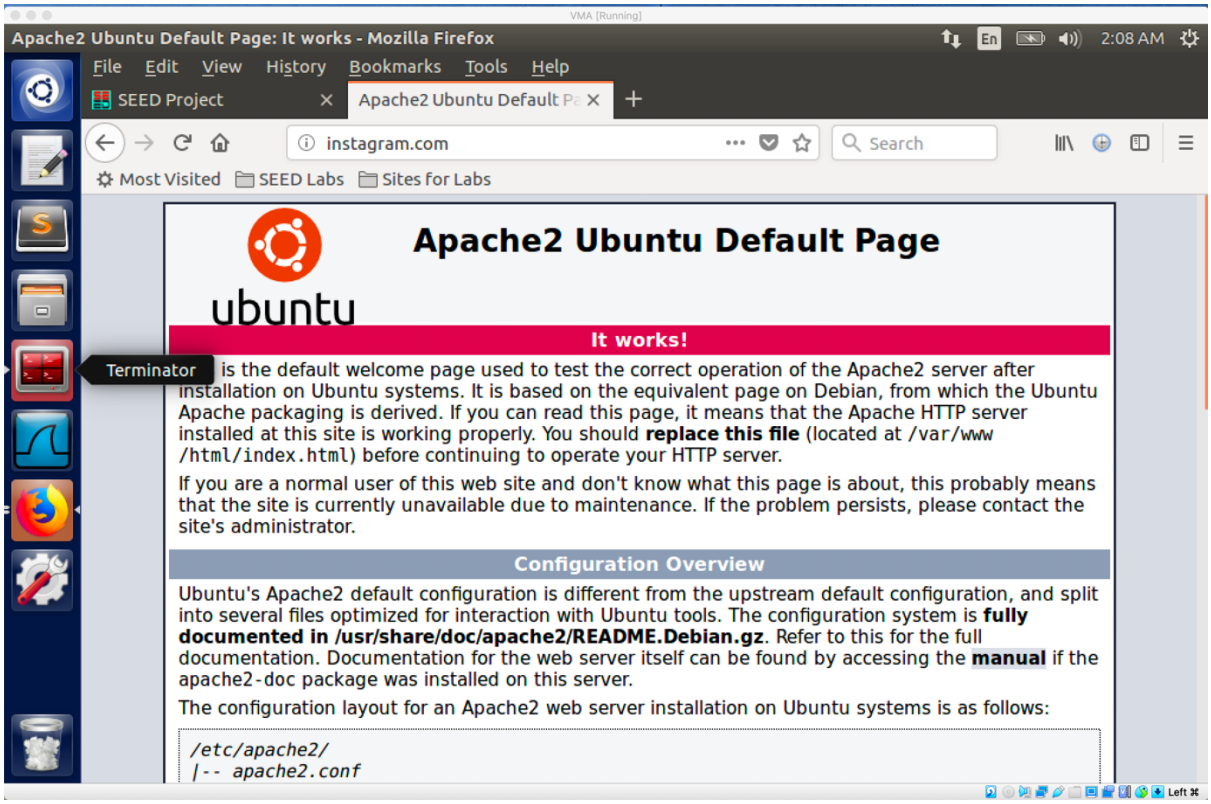
Followed the task 4 procedure, I add another HTTPS web server to the default-ssl.conf file for Instagram.com.



Step2:
After that I add the entry '10.0.2.7 instagram.com' to another vm. The 10.0.2.7 Is the ip address of the man-in-the-middle.



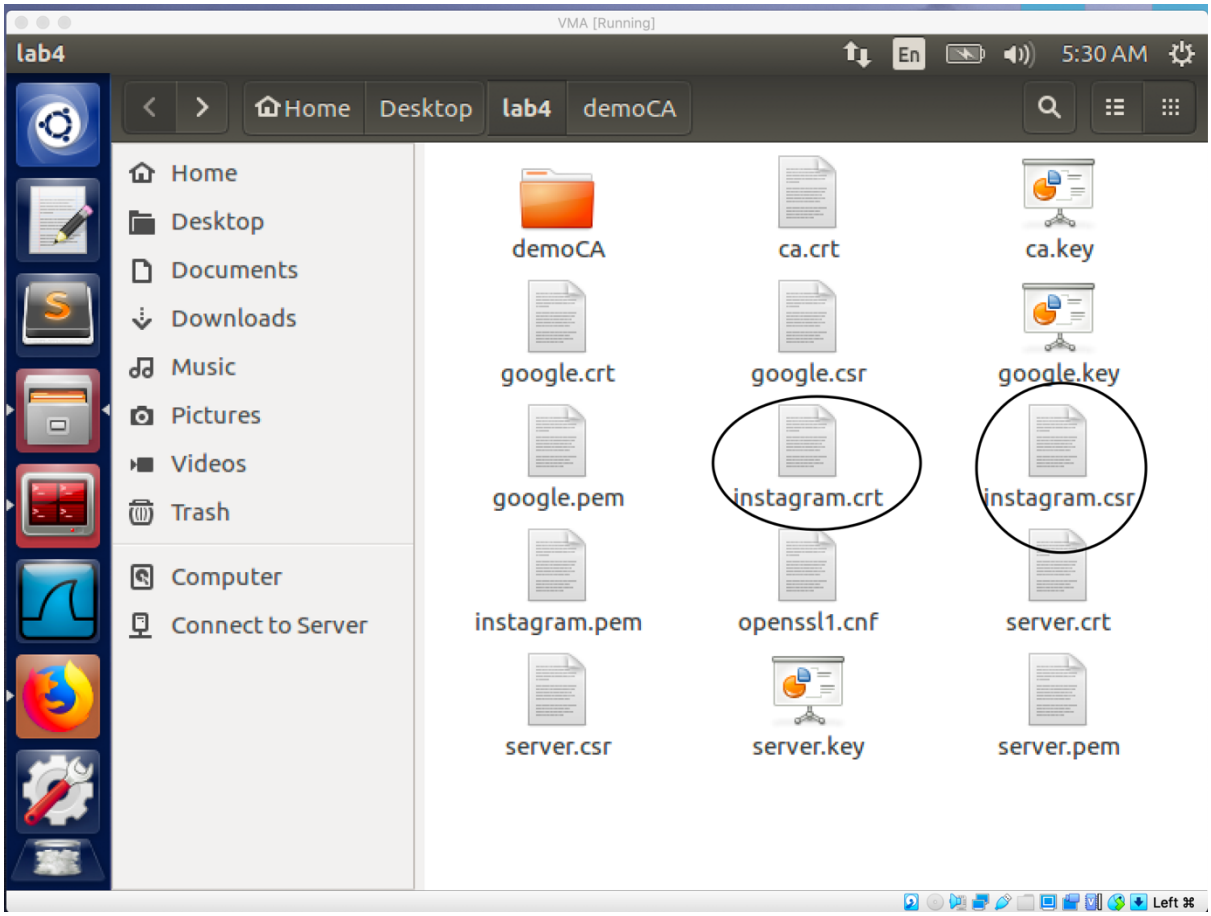
Step3: Browse the target website
By searching the URL Instagram.com on the victim machine, I will go to the malicious website.



Task6: Launching a Man-In-The-Middle Attack with a Compromised CA

Step1:

I using my CA to signed a certificate for Instagram.com by using command ‘openssl req -new -key server.key -out instagram.csr -config openssl1.cnf’ and command ‘openssl ca -in instagram.csr -out instagram.crt -cert ca.crt -keyfile ca.key -config openssl1.cnf’.



Step2:

I send the ca.crt file to the victim machine by using command ‘scp seed@10.0.2.7:/home/seed/Desktop/lab4/ca.crt /home/seed/Desktop’

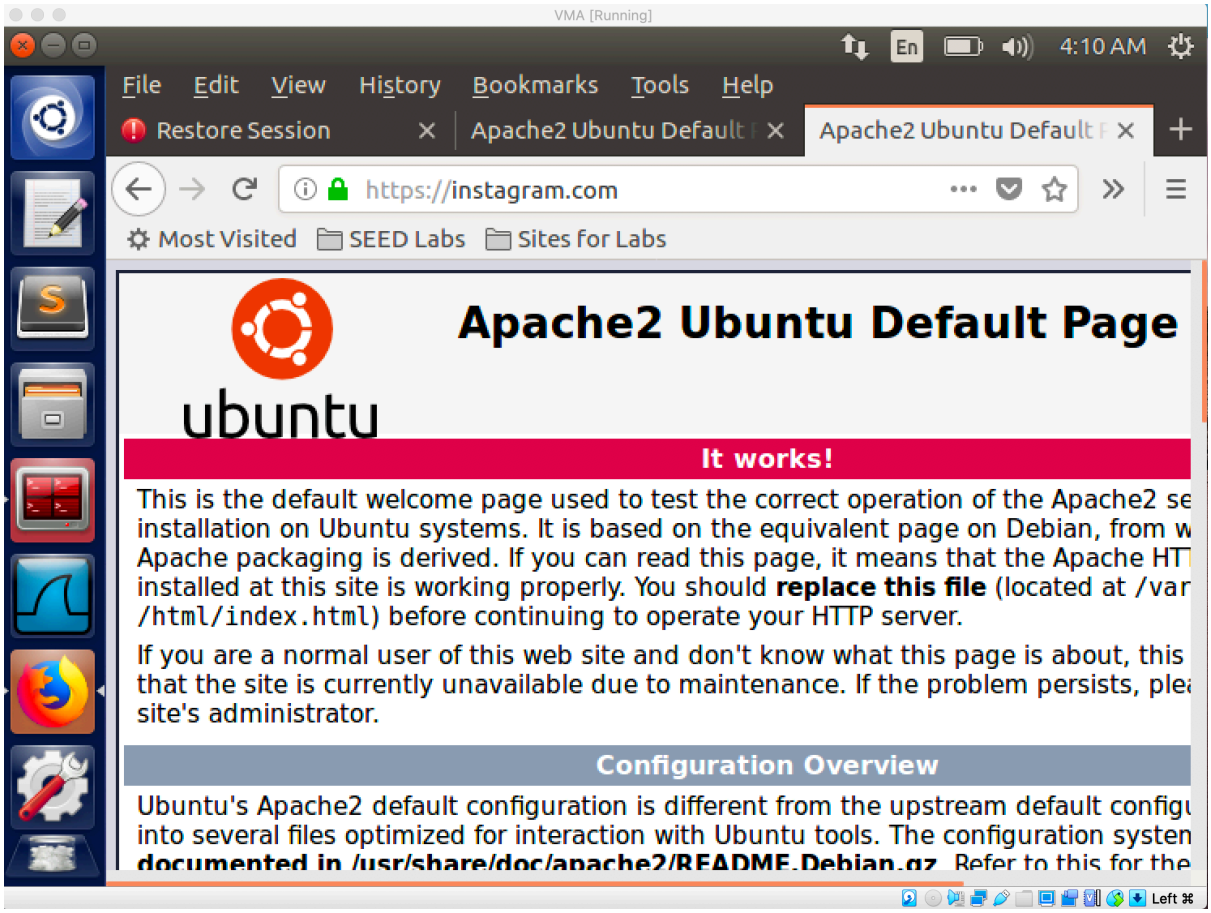
Step3:

I add the ca.crt file to the truested ca list of Firefox.

Certificate Name	Security Device
aaron	
zewen.com	Software Security Device
AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token

Step4:

After configuration, I am able to visit the target website under https request at the victim machine. Which means, if the DNS attack is successful I am able to set up the man-in-the-middle attack.


VMA [Running]
4:10 AM
File Edit View History Bookmarks Tools Help
Restore Session
Apache2 Ubuntu Default
Apache2 Ubuntu Default
https://instagram.com
Most Visited
SEED Labs
Sites for Labs
Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 se
installation on Ubuntu systems. It is based on the equivalent page on Debian, from w
Apache packaging is derived. If you can read this page, it means that the Apache HT
installed at this site is working properly. You should replace this file (located at /var
/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this
that the site is currently unavailable due to maintenance. If the problem persists, plea
site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configu
into several files optimized for interaction with Ubuntu tools. The configuration systen
documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the
Left ⌘