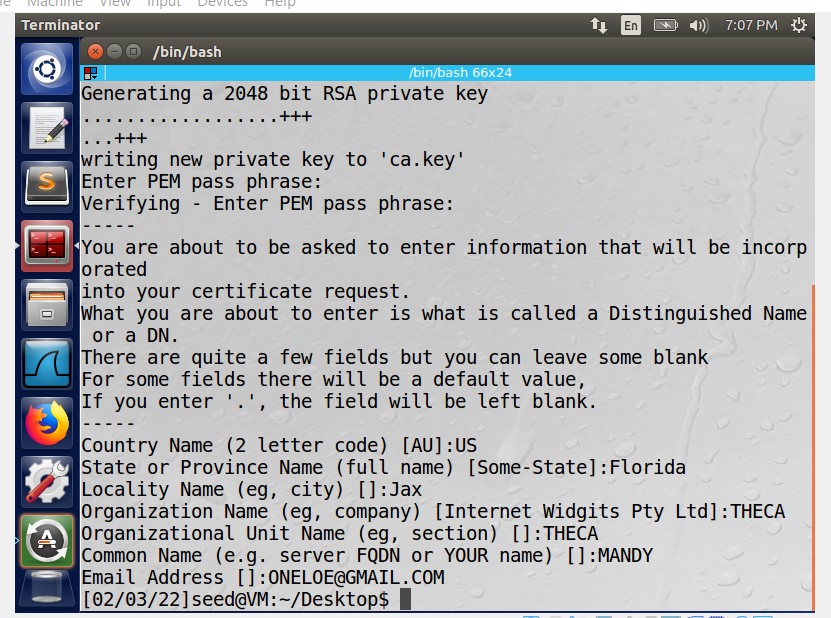
`**Project 1 - Public-Key Infrastructure (PKI) Lab**

Group Members: Trysten Southern and Manderas Kidane

2.1 Task 1: Becoming a Certificate Authority (CA)

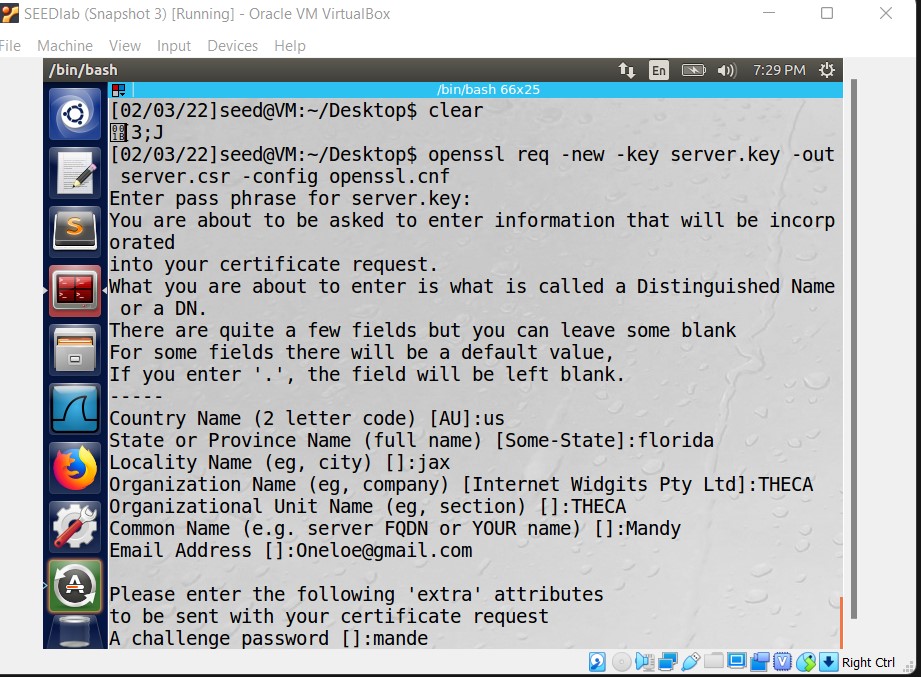


Using this command will generate a self-signed certificate for us (The CA), creating a ca.key and a ca.crt file which stores the key and certificate respectively.

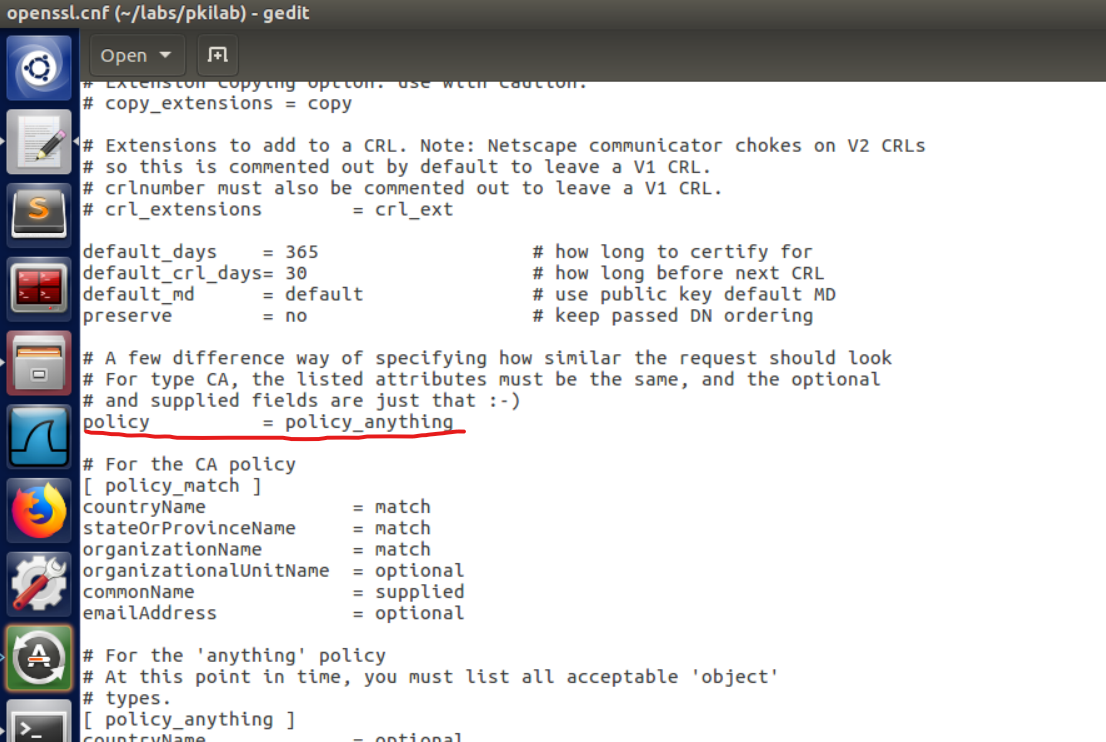


Our CA information is stored within the certification request

2.2 Task 2: Creating a Certificate for SEEDPKILab2020.com



We create a new certificate signing request for our website, SEEDPKILab2020.com



In order to for OpenSSL to generate a certificate, requests must match those of the CA. We can simply change this policy within the config to support anything (less restrictive)

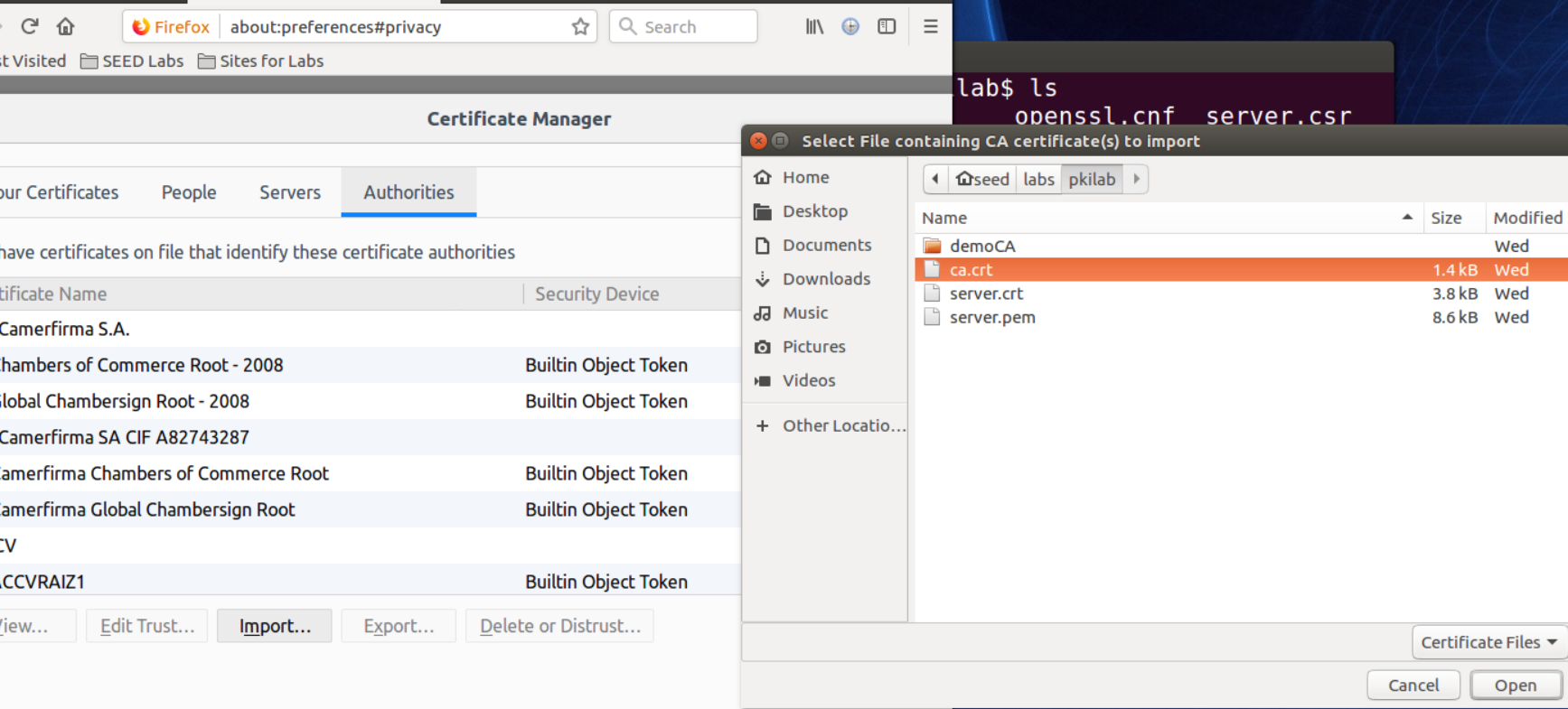
2.3 Task 3: Deploying Certificate in an HTTPS Web Server



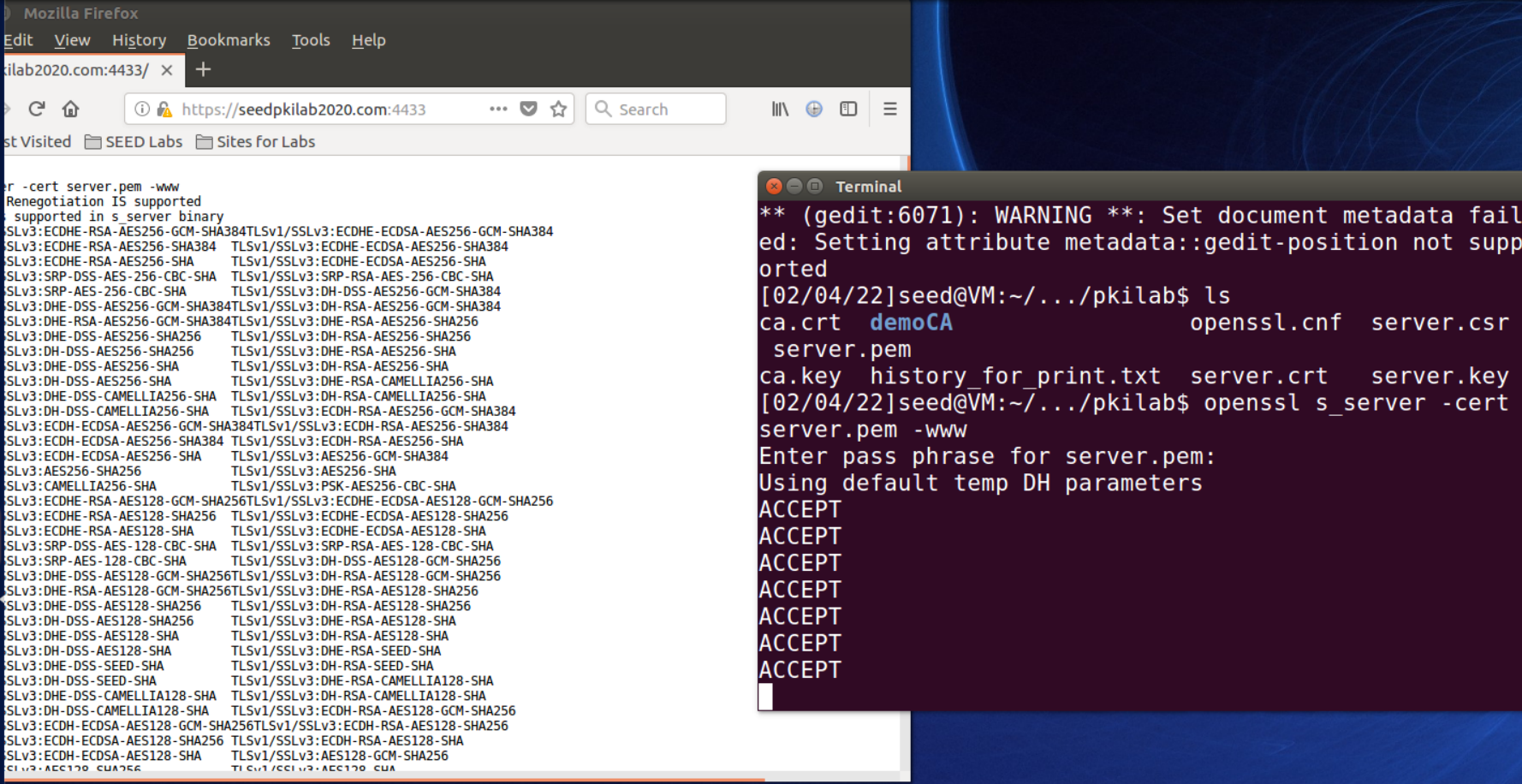
Must use the sudo gedit command to make an entry to the /etc/hosts file, as it is otherwise restricted



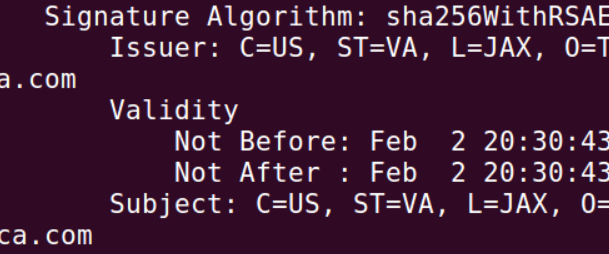
Using these two commands will combine the secret key and certificate



Ca.crt must be an authorized CA certificate in firefox, or else warnings will display when accessing the web page as the certificate issuer is unknown to firefox.

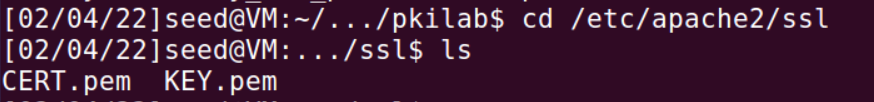
****

Within the https://seedpkilab2020.com:4433 web page, the cipher used, master key, protocol, encryption algorithms, as well as current session ID is visible.

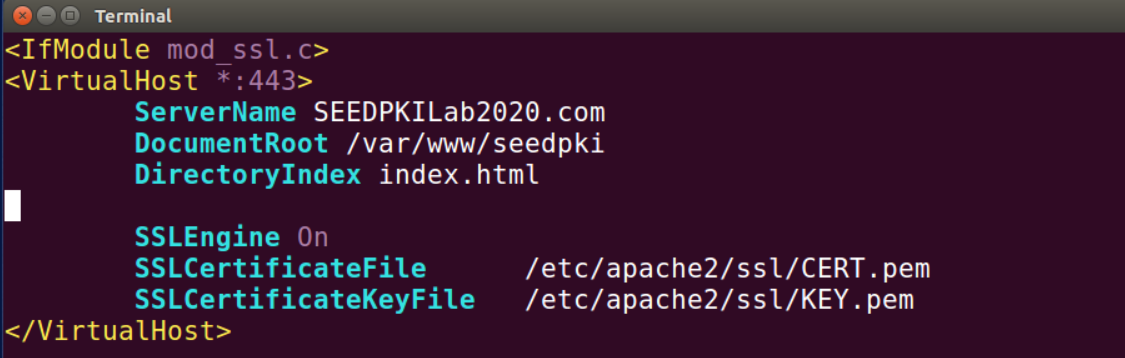


When modifying the server.pem file, the master key changed, though the web page still was able to run. When connecting to <https://localhost:4433>, a message displays stating that the web page is improperly set up.

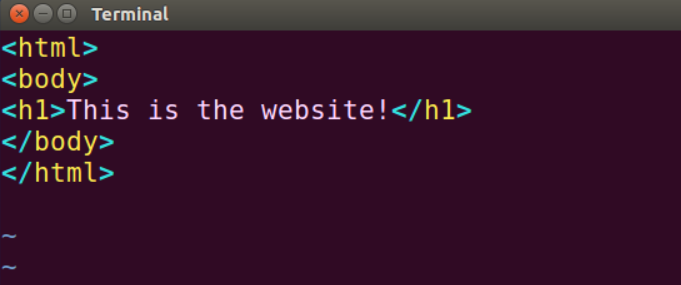
2.4 Task 4: Deploying Certificate in an Apache-Based HTTPS Website



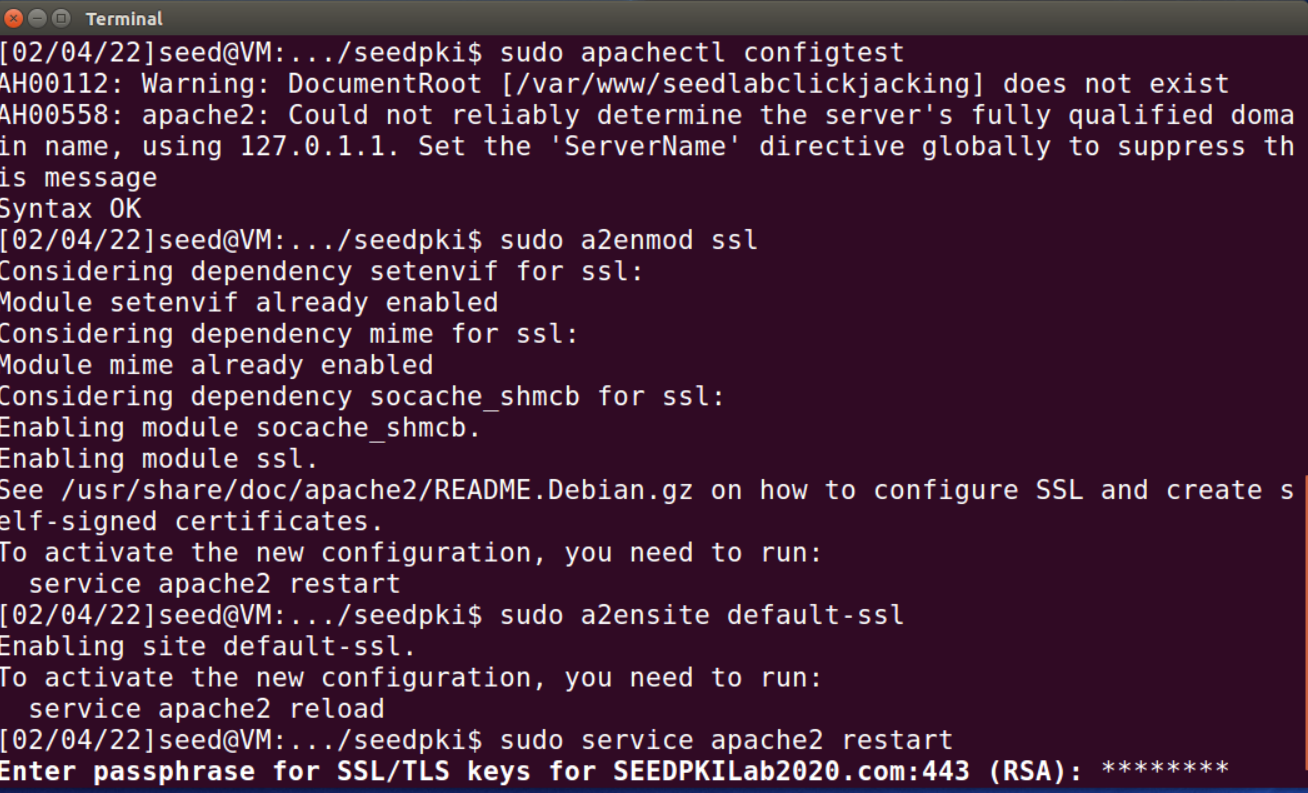
After using the cp command to copy server.crt to CERT.pem and server.key to KEY.pem, we move these to the apache ssl directory



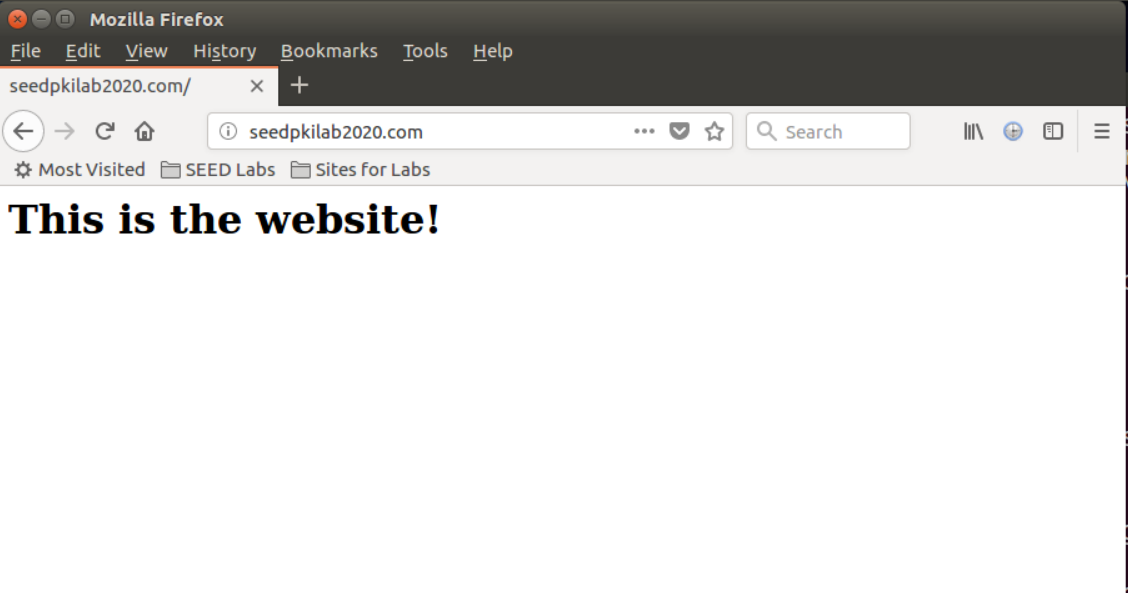
An entry to the sites-available config file allows apache to know where to find certificates for our hosted website.



Within the /var/www/seedpki file the index.html file utilizes html to display text on the website



After configuring the necessary files, commands as “sudo apachectl configtest”, “sudo a2enmod ssl” and “sudo a2ensite default-ssl” are ran to set up the web page “sudo a2ensite default-ssl” to ready the configuration and “sudo service apache2 restart” to complete. Once SSL/TLS keys are input, the page is good to go!



Here is the finished website within the https server. This is the text displayed from our index.html file!