Name: Bharath Karumudi
Lab: Public Key Infrastructure

Task 1: Becoming a Certificate Authority (CA)

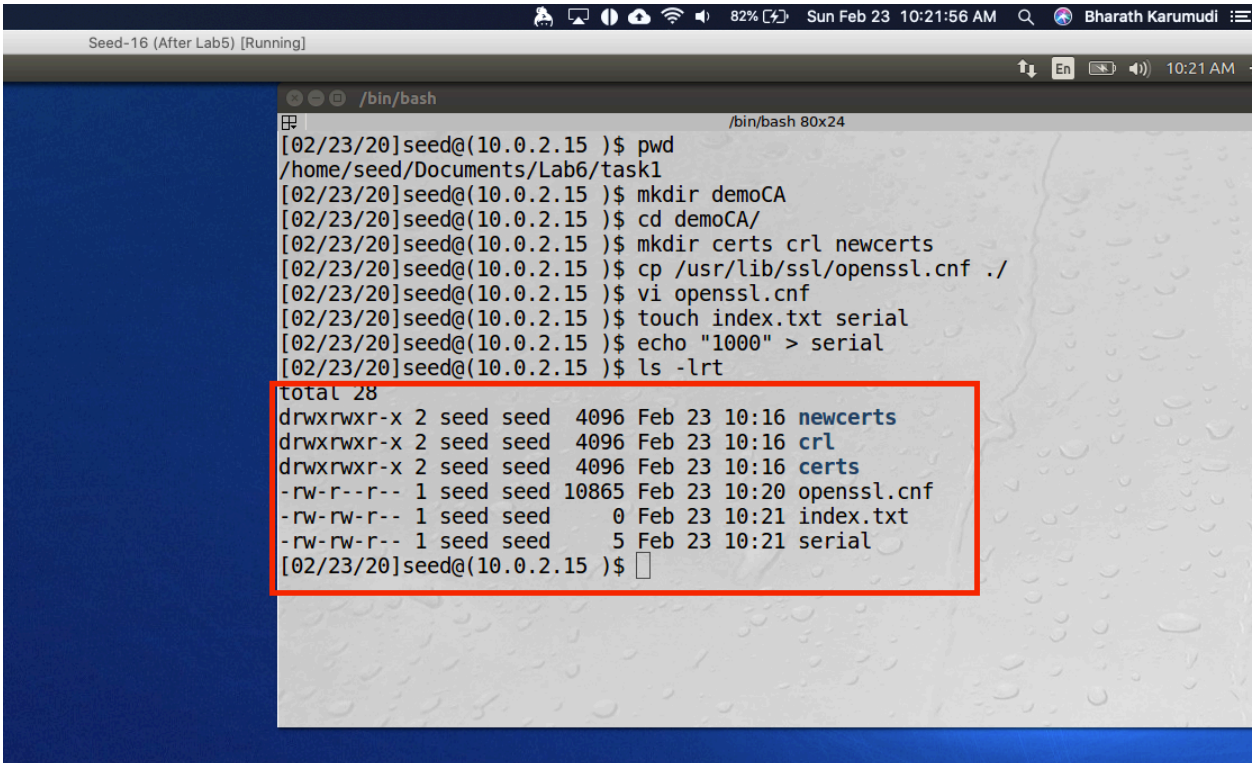


Fig1: Created the default directories and modied openssl.cnf file for demoCA

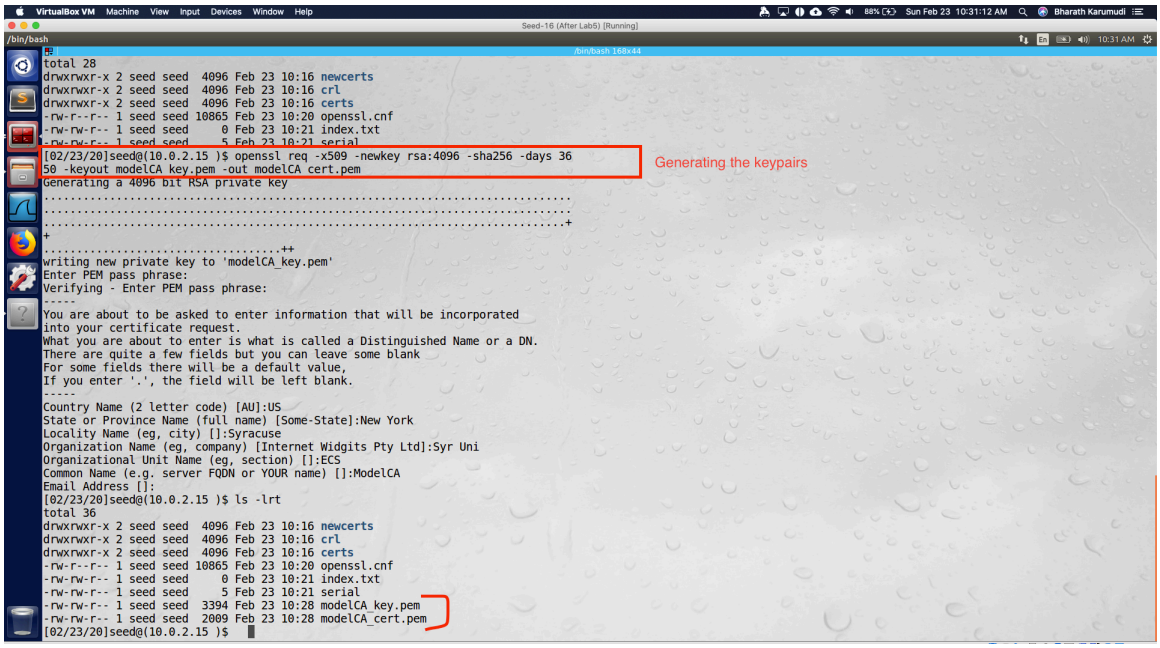


Fig2: Generated the keypairs and self-signed cert for the ModelCA

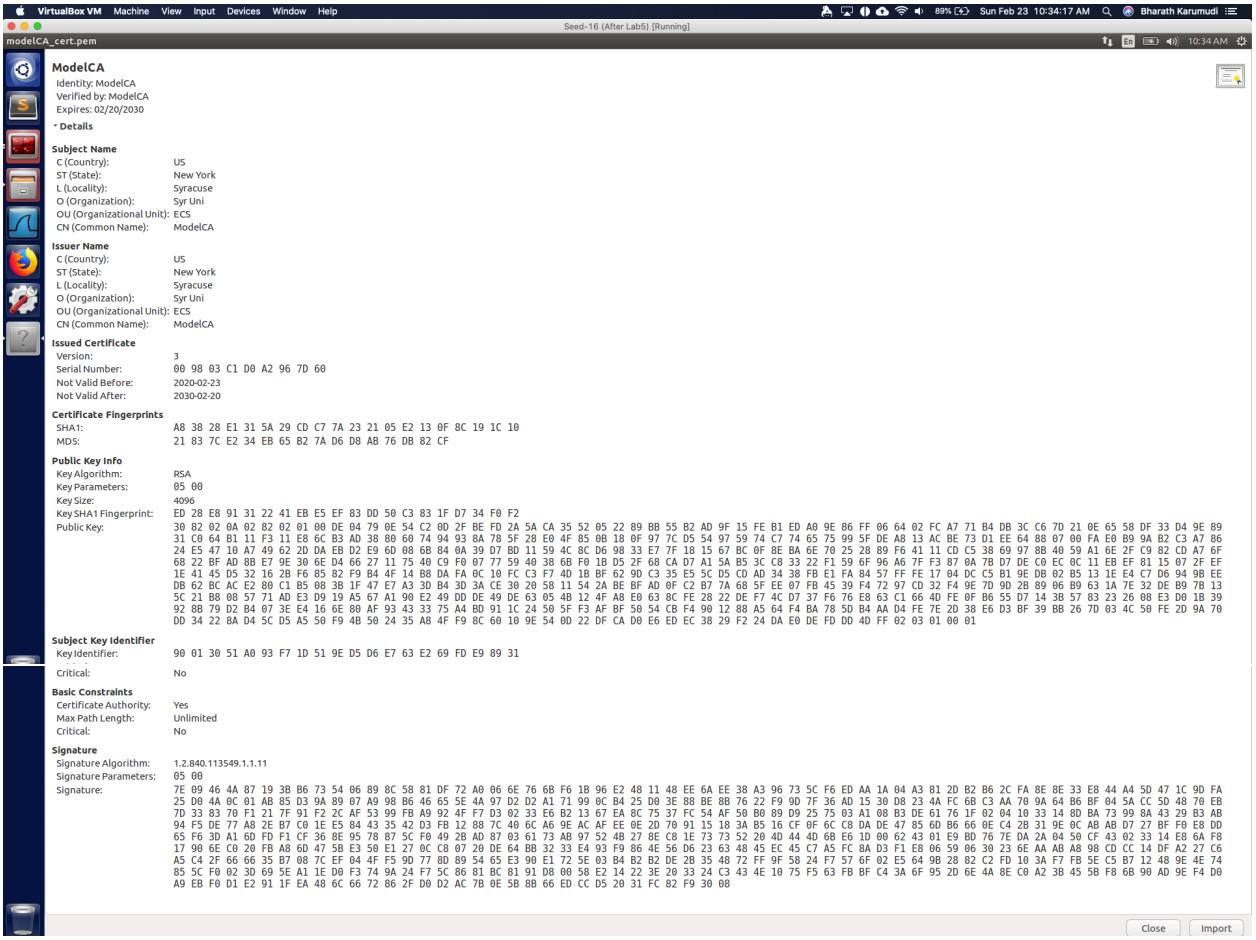


Fig3: Self Signed certificate of ModelCA

Observation: A Certificate Authority was established of name ModelCA and generated the public and private keys for the ModelCA and the self-signed public key certificate and became the root CA to vouch the other entities.

Explanation: Created the required default directories for the CA as shown in Fig1 and then generated the public and private key as shown in Fig2 for the ModelCA along with the self-signed public key certificate (shown in Fig3). With these will now be able to issue the certificates to the other entities.

Task 2: Creating a Certificate for SEEDPKILab2018.com

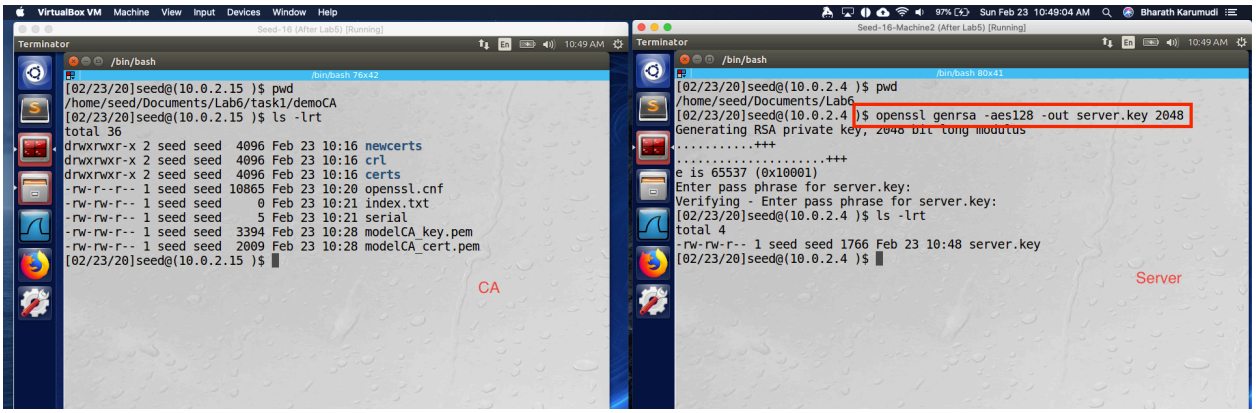
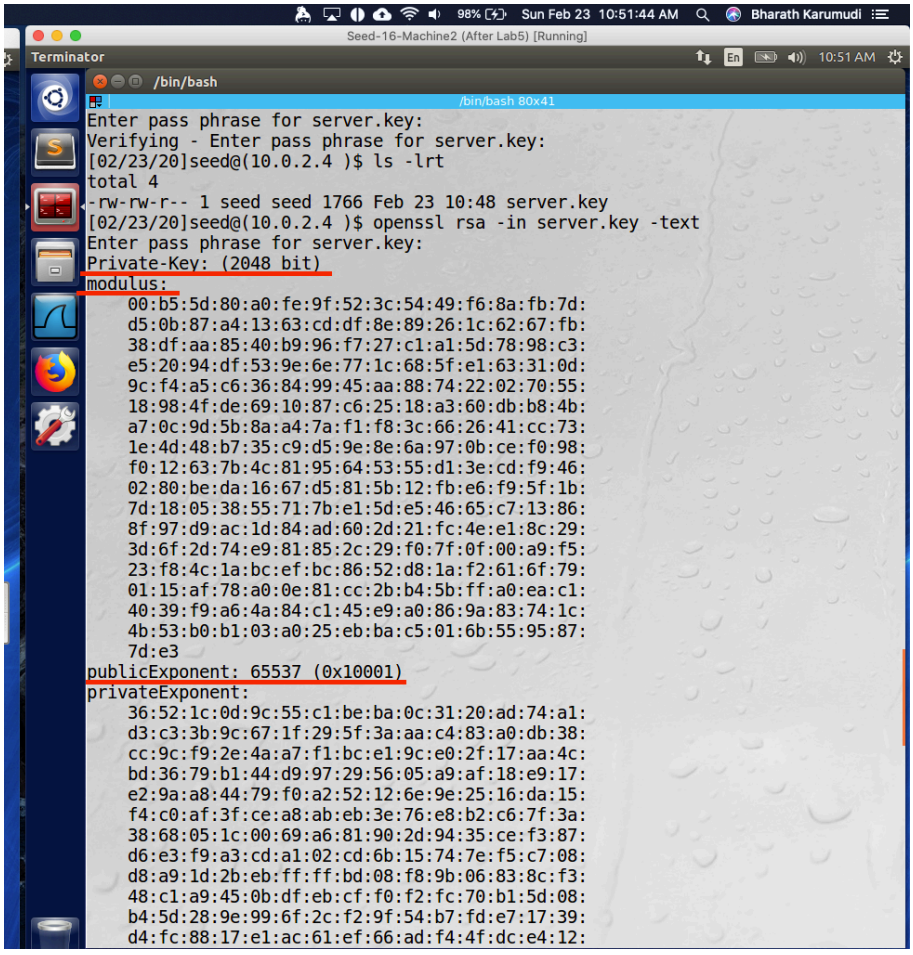


Fig1: Generated the public and private key pair for server



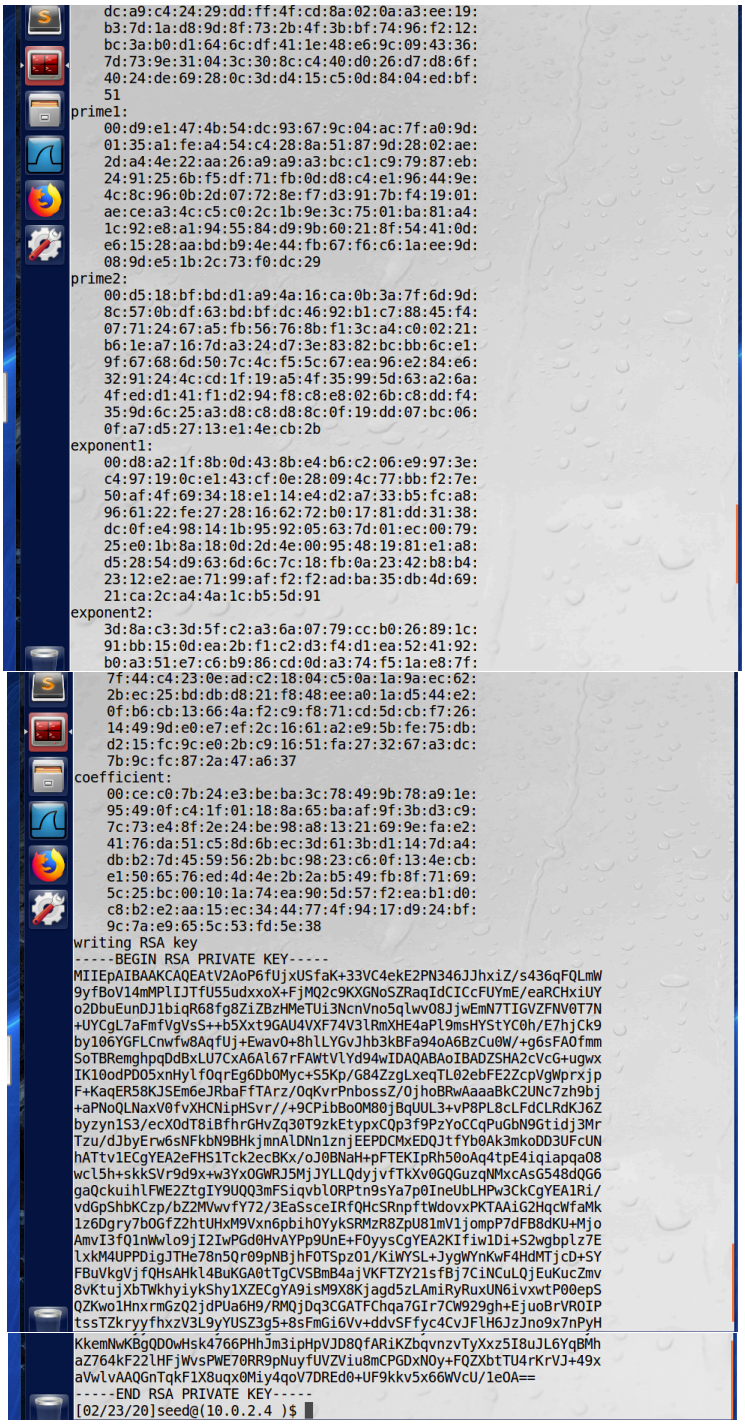


Fig2: Showing the private key contents

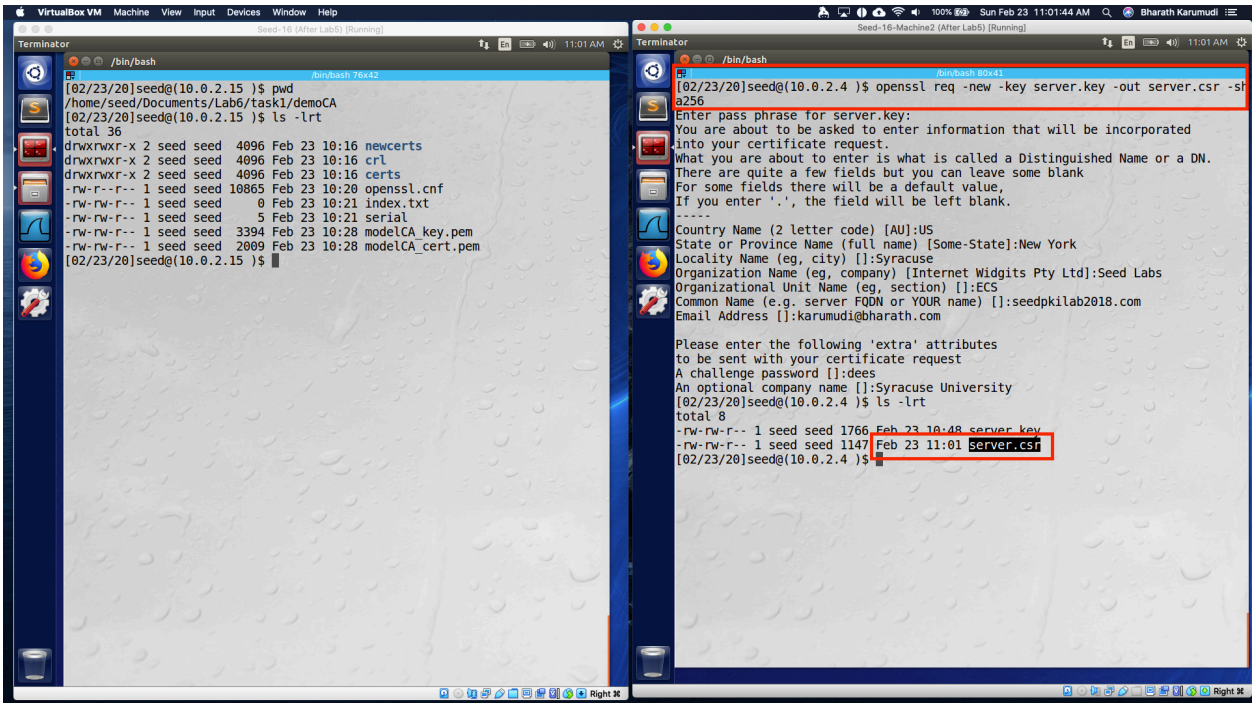


Fig3: Generated the Certificate Signing Request (csr)

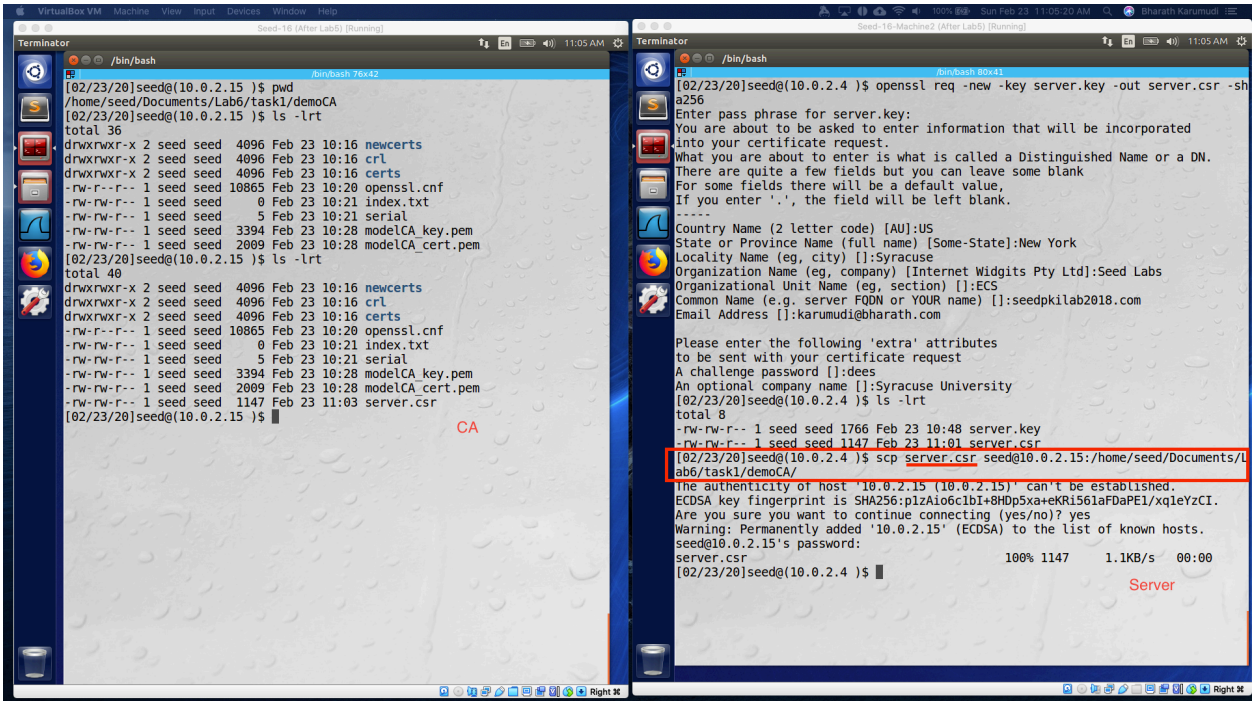


Fig4: Giving the csr to CA to sign over scp. In real-world it will be shared in their website.

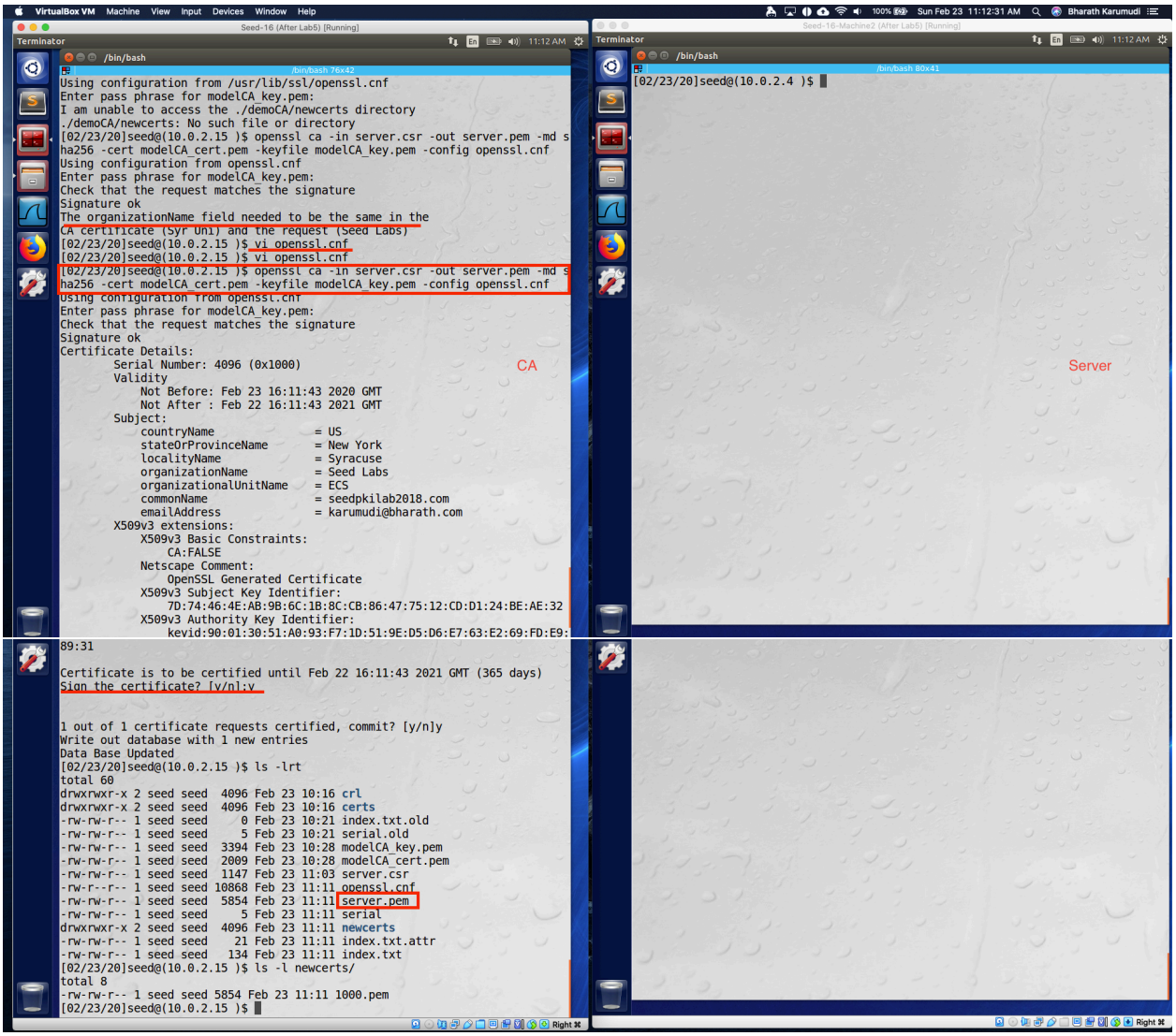


Fig5: Signed the csr and generated the certificate for the server

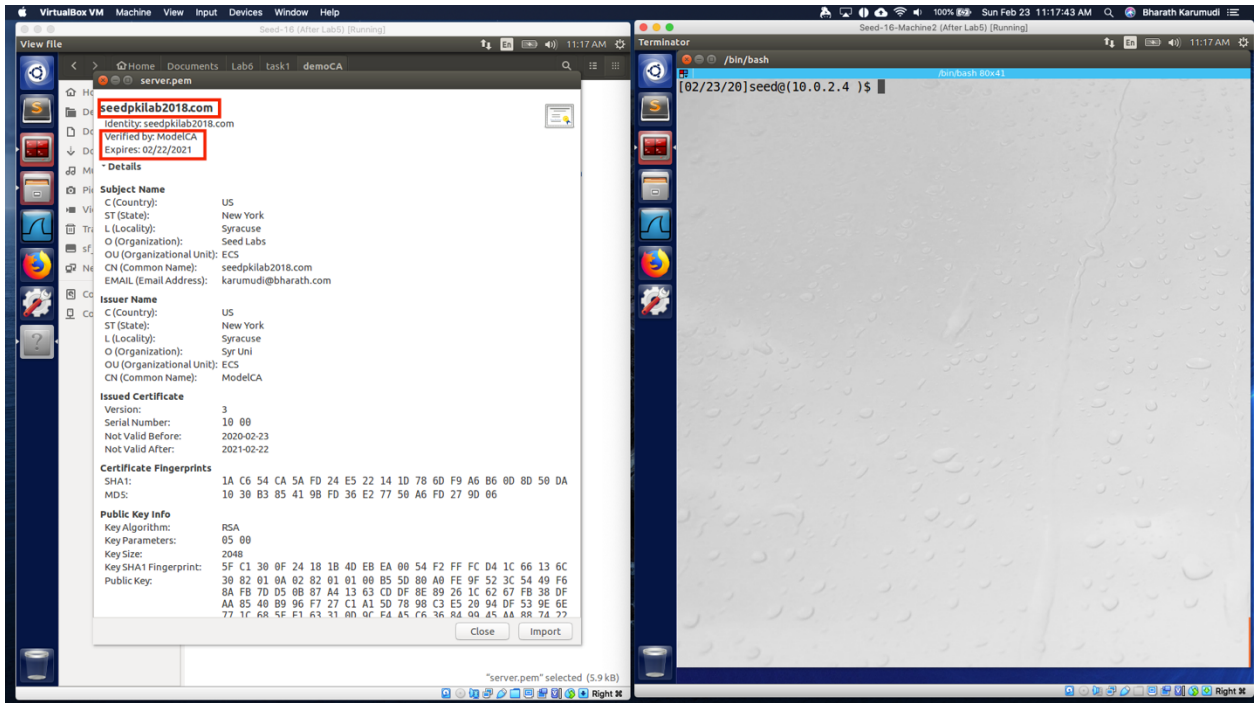


Fig6: View of the Signed and generated certificate by our ModelCA

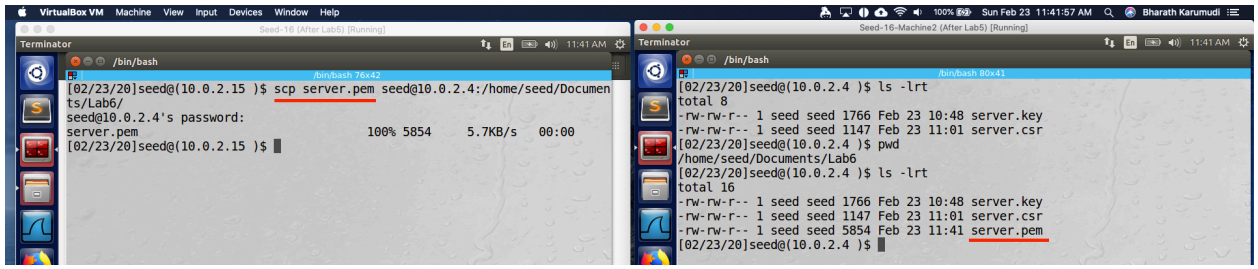


Fig7: Delivered the signed certificate to requestor (here over scp)

Observation: Generated the key pairs for the server and using them generated the Certificate Signing request for website (seedpkilab2018.com) and requested the ModelCA to sign and provide the Certificate.

Explanation: As a requestor, we first need the public and private key pair to establish a TLS/SSL connection on our website, so generated a keypair using openssl with RSA 2014 as shown in Fig1 and the generated private key is as shown in Fig2, it will have modulus, exponents and prime numbers that are used to generate the private key. Once the keypair is generated, created a Certificate Signing Request with all the details provided and sent to CA over scp as shown in Fig3 and Fig4.
As a CA, verified the requestor and signed the csr with ModelCA private key as shown in Fig 5 and issued a certificate to seedpkilab2018.com as shown in Fig6 and delivered to the requestor.

Task 3: Deploying Certificate in an HTTPS Web Server

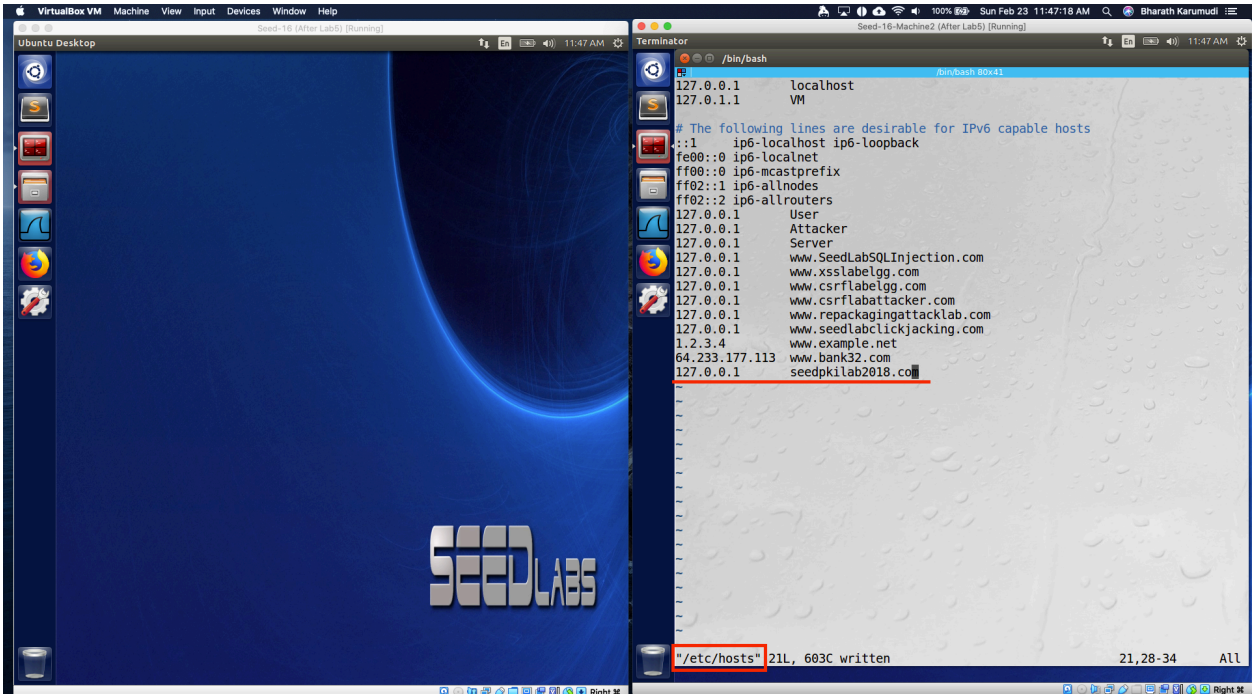


Fig1: Modified the /etc/hosts files by adding an entry for seedpkilab2018.com

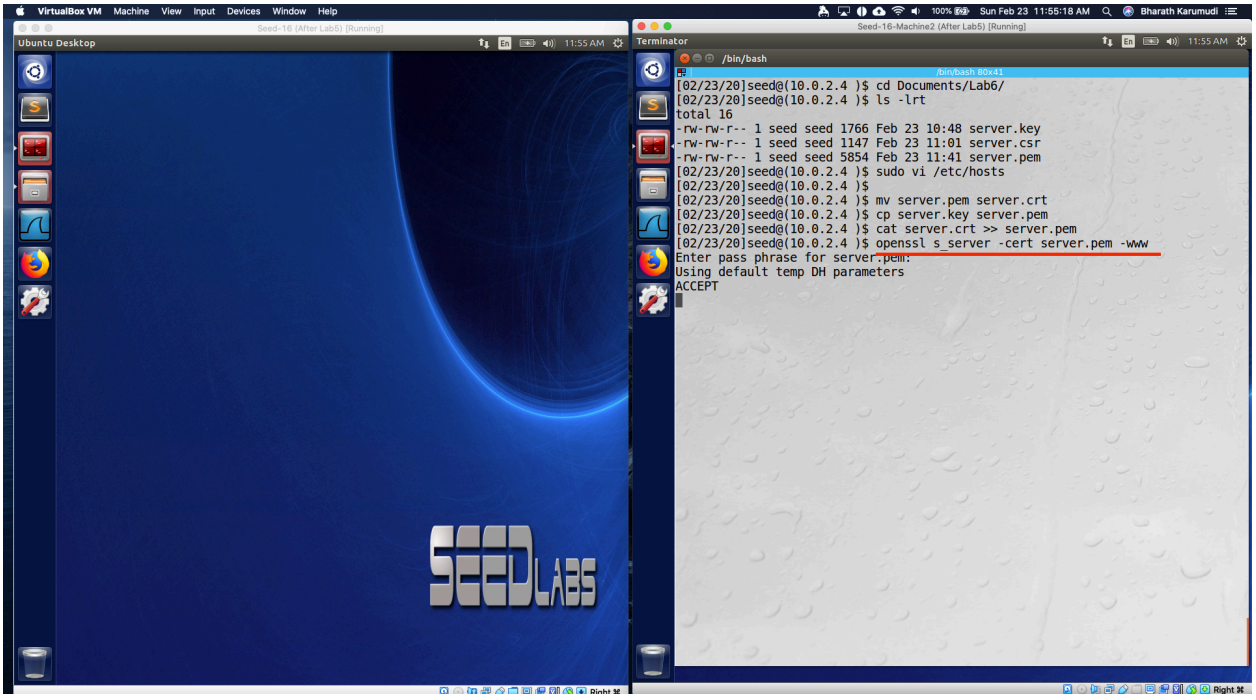


Fig2: Launching the server using openssl utility

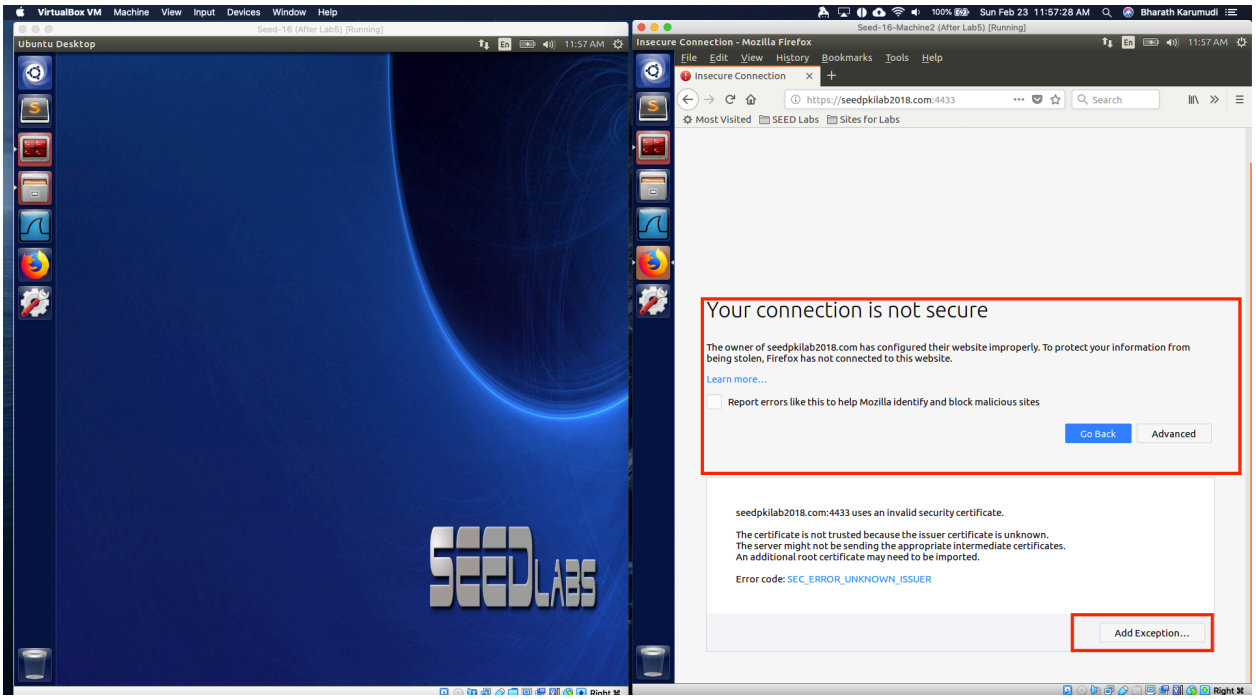


Fig3: Accessing the website using Firefox and can see the exceptions

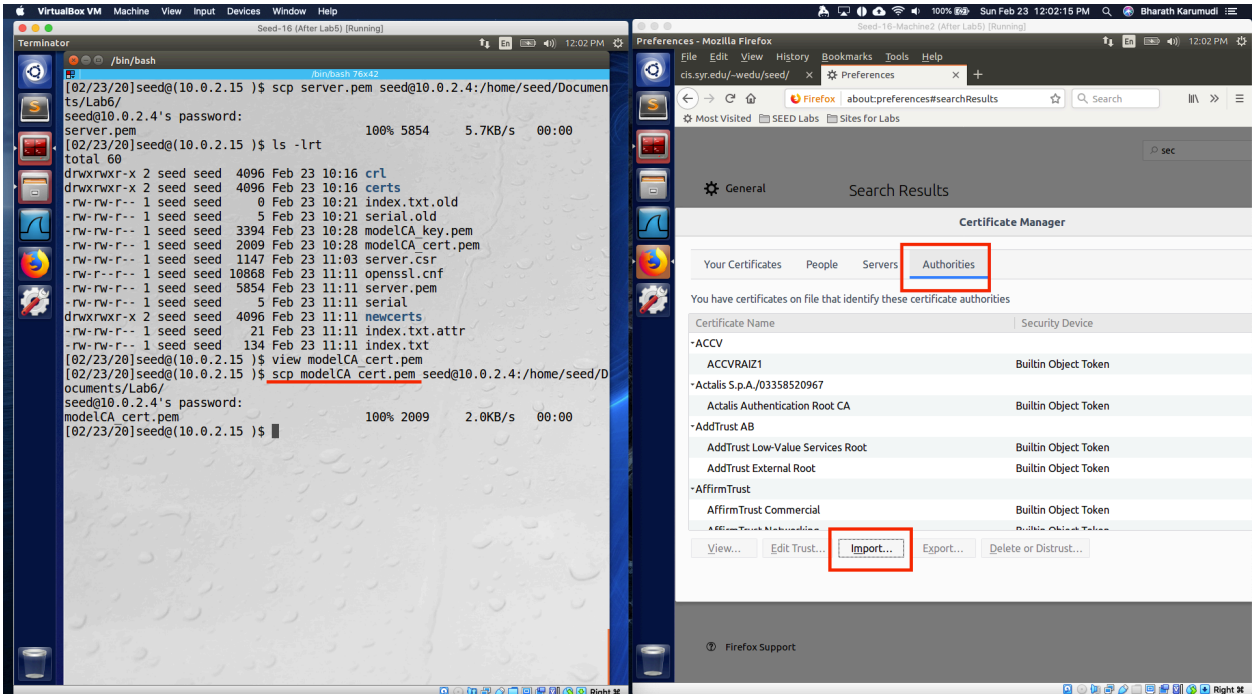


Fig4: CA shared the cert (left) and Firefox Certificate manager (right)

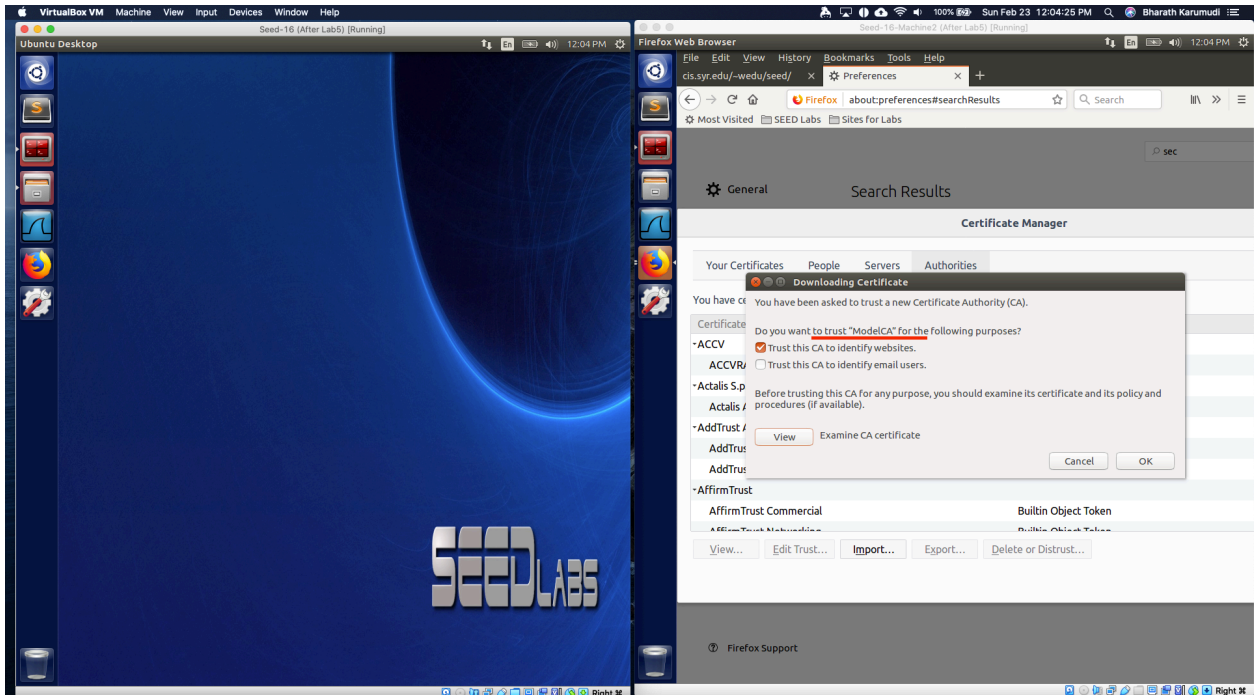


Fig5: Importing the ModelCA cert to Firefox

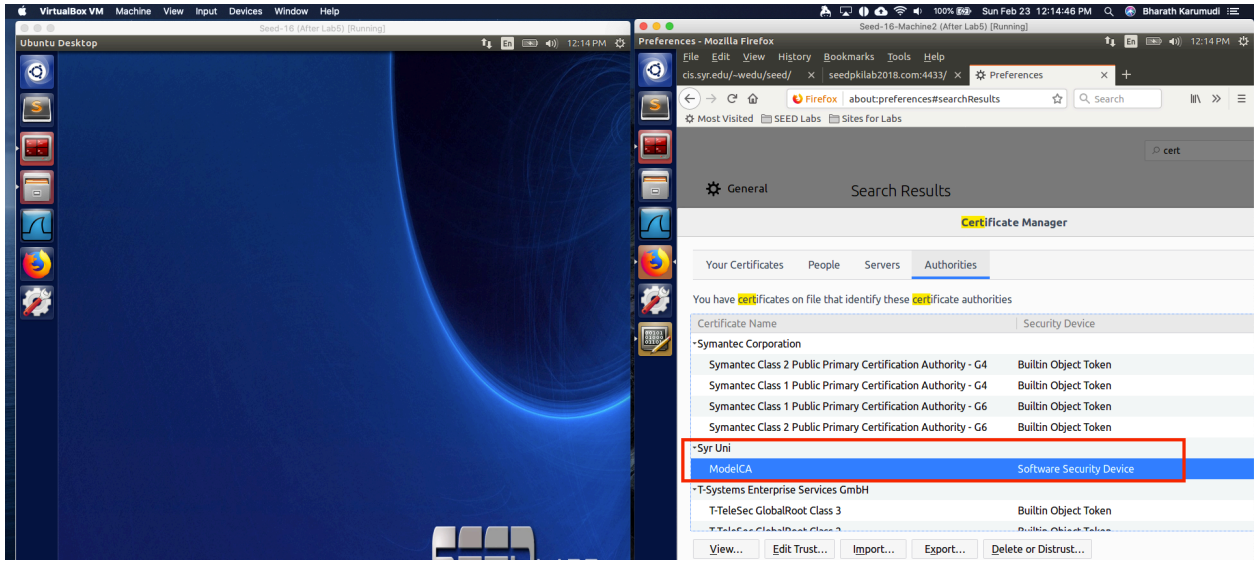


Fig6: Syr Uni ModelCA is now added to trusted CA

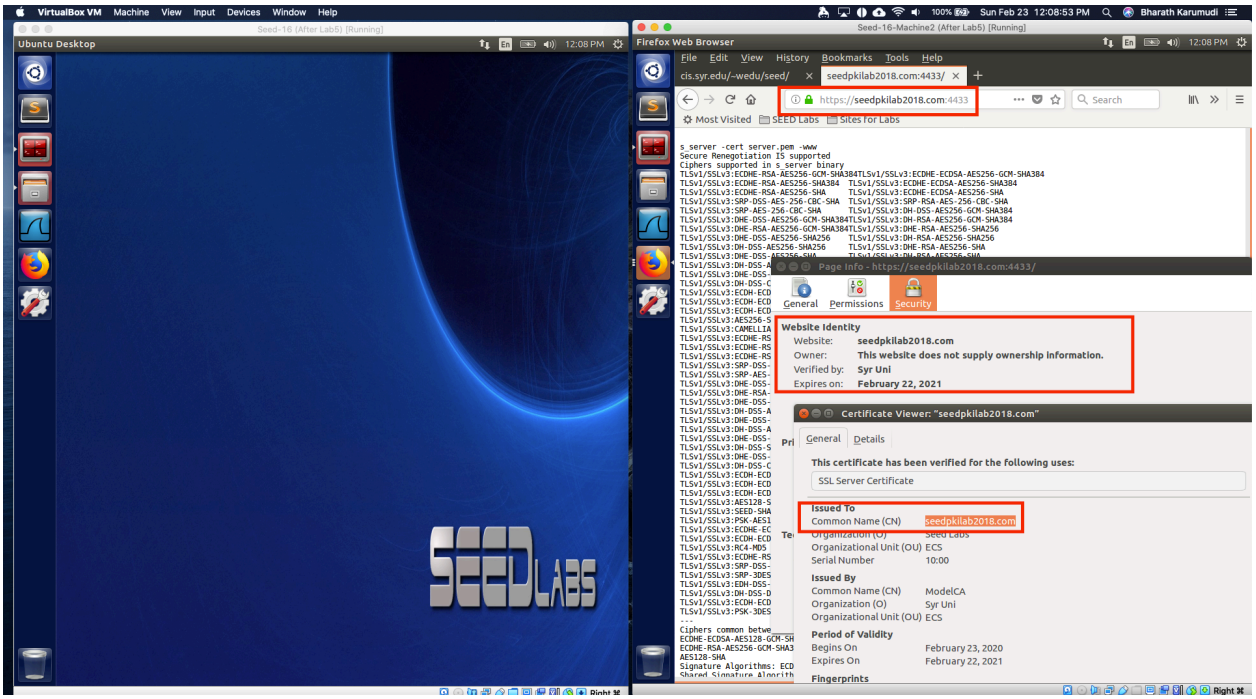


Fig7: The browser now trusted the website certificate

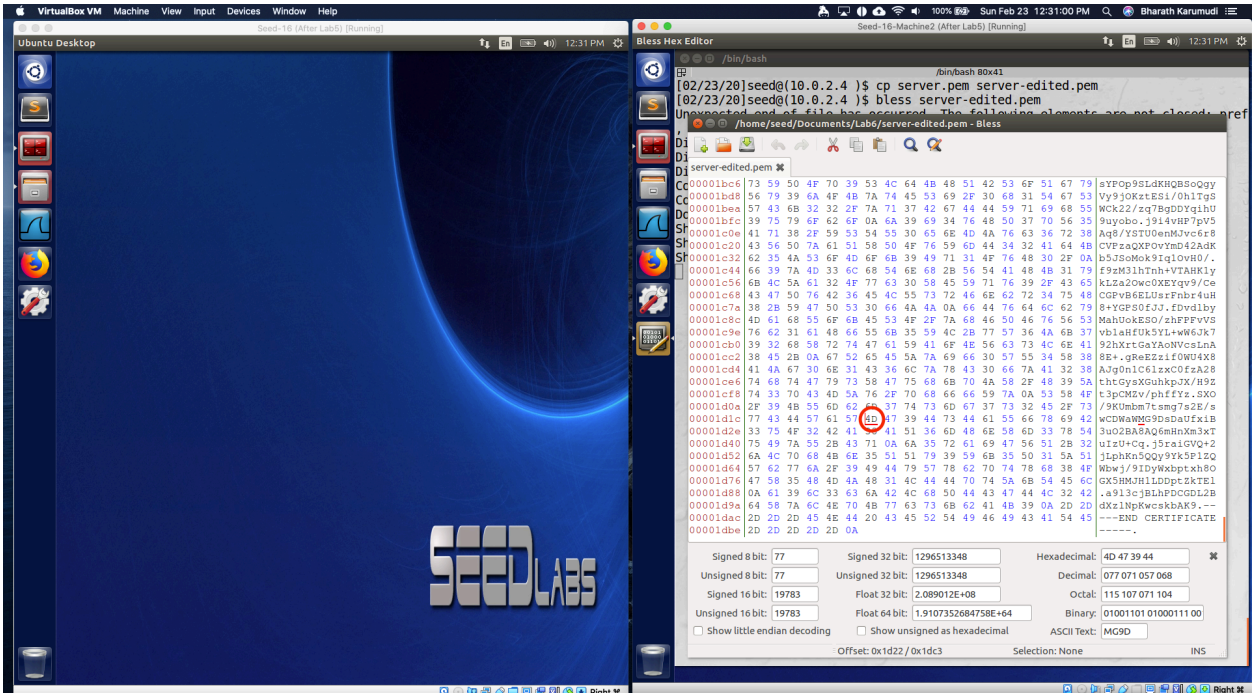


Fig8: Modifying the highlighted byte from 4D to 4F

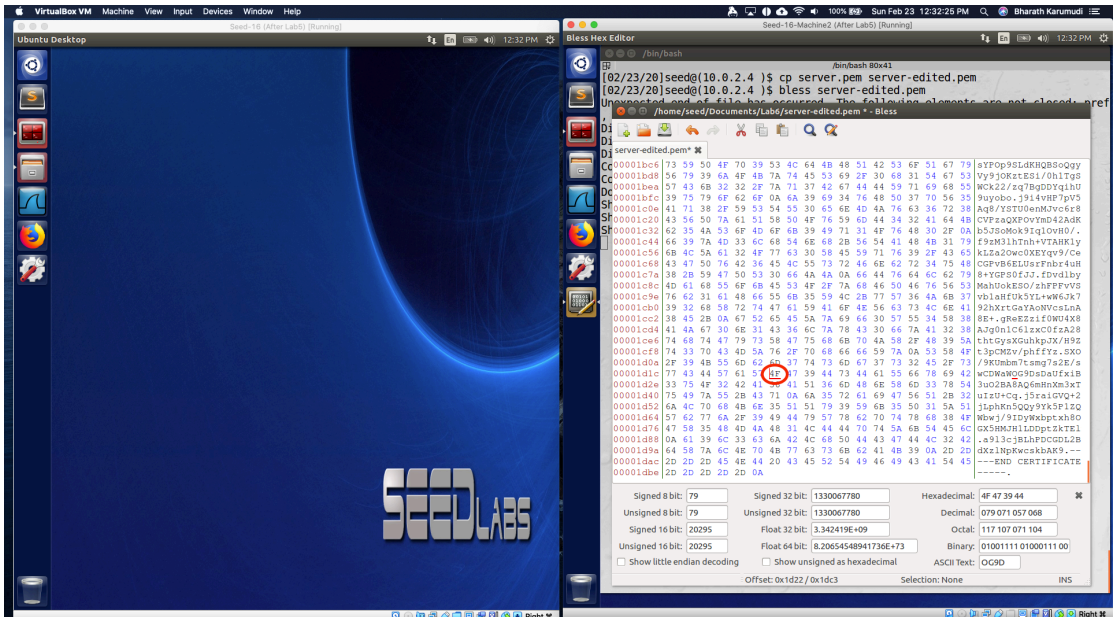


Fig9: Modified and Saved the cert as server-edited.pem

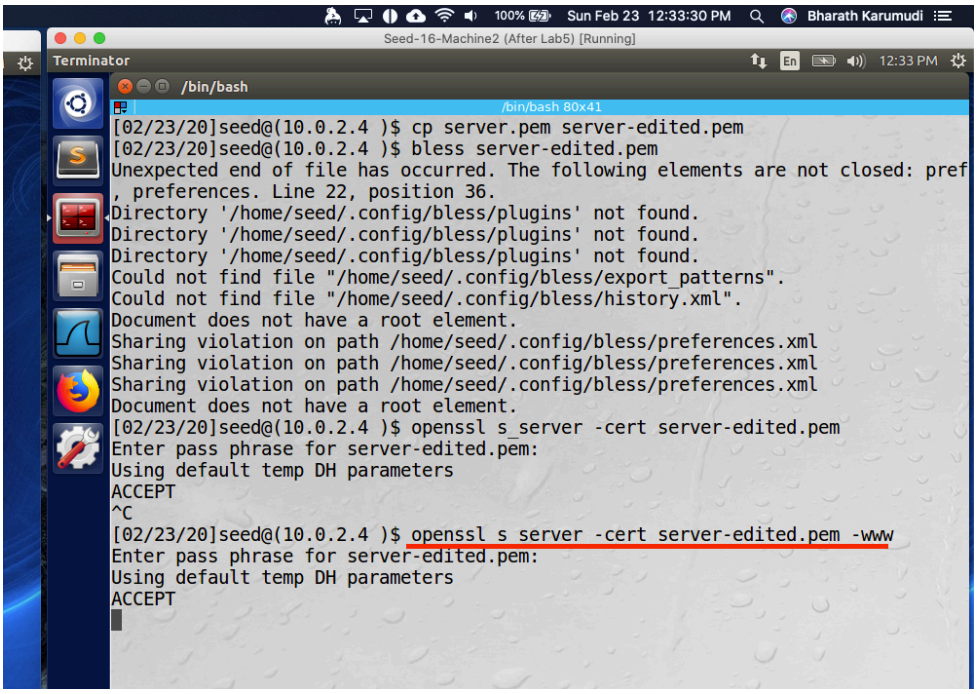


Fig10: Starting the server with modified cert

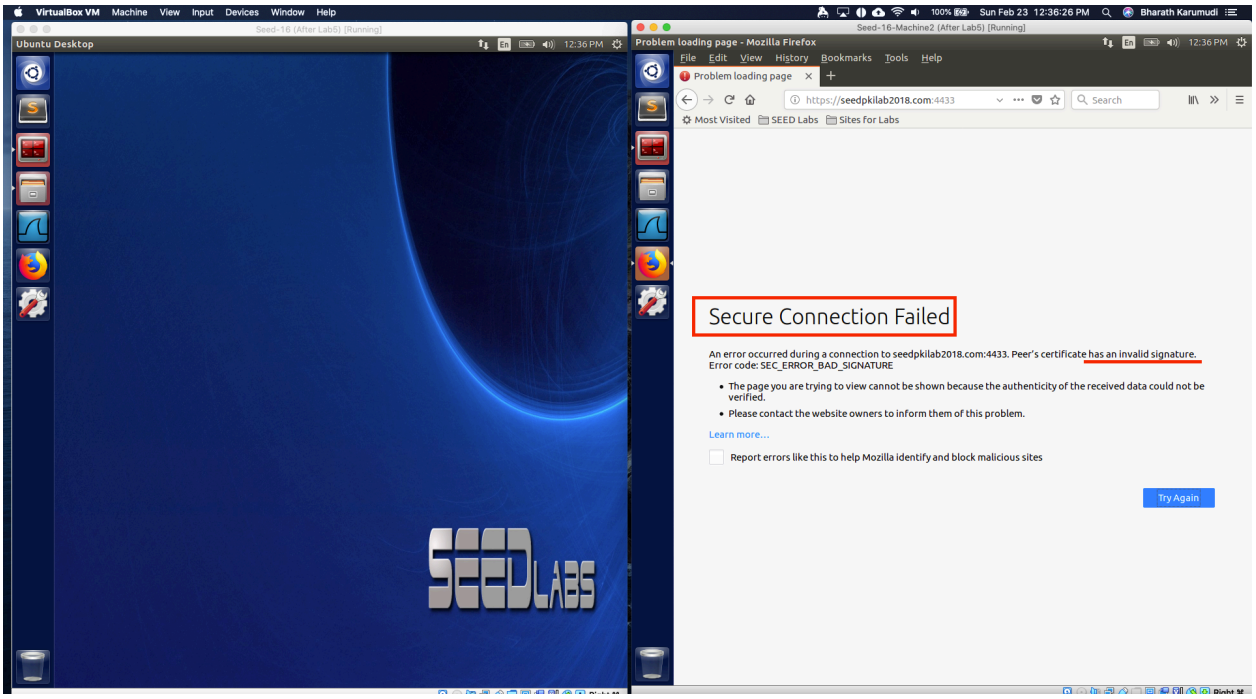


Fig11: Connection to the server failed with Invalid Signature

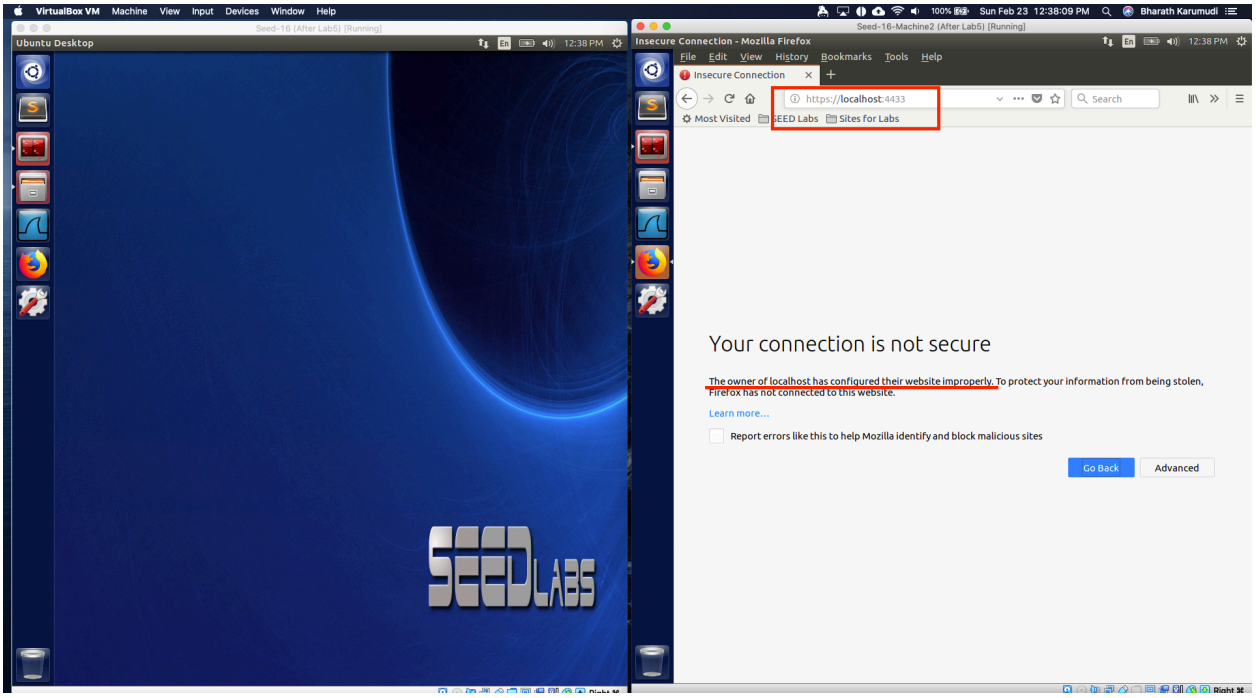


Fig12: Error when used on localhost instead of cert issued CN

Observation: The CA issued a signed certificate and then the certificate was used for the seedpkilab2018.com, but as the CA is not a default trusted one by Firefox, we have added the CA's certificate to the trust store. Once the CA certificate is added, the browser was able to trust the website certificate. Also, when modified the certificate by single byte the cert is no

longer valid and failed signature validation and also when used on localhost domain, it failed because the certificate was issued to a different Common Name (CN).

Explanation:

- Modified the /etc/hosts file and added an entry for the seedpkilab2018.com with localhost IP address.
- Received the signed certificate from CA and using openssl s_server launched a server to debug and from the browser when access the site https:// seedpkilab2018.com:4433 the Firefox issued a warning as the signed CA is not known.
- Received the CA Self-signed certificate and then imported the cert to the Firefox trust store and once added the Firefox was able to able to accept the website certificate without any errors or warnings.
- Modified the server.pem by a single byte from 4D to 4F as shown in Fig8 and Fig9 and saved the cert as server-edited.pem and launched the openssl s_server utility with the modified cert. When accessed the site from the browser, the Firefox was unable to validate the signature and failed to establish the connection (Fig 11).
- Also, when accessed the website with https://localhost:4433 domain, the Firefox failed to establish as the certificate is issued to different CN (Fig 12). This is a security measure to protect against Common Name attacks.

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

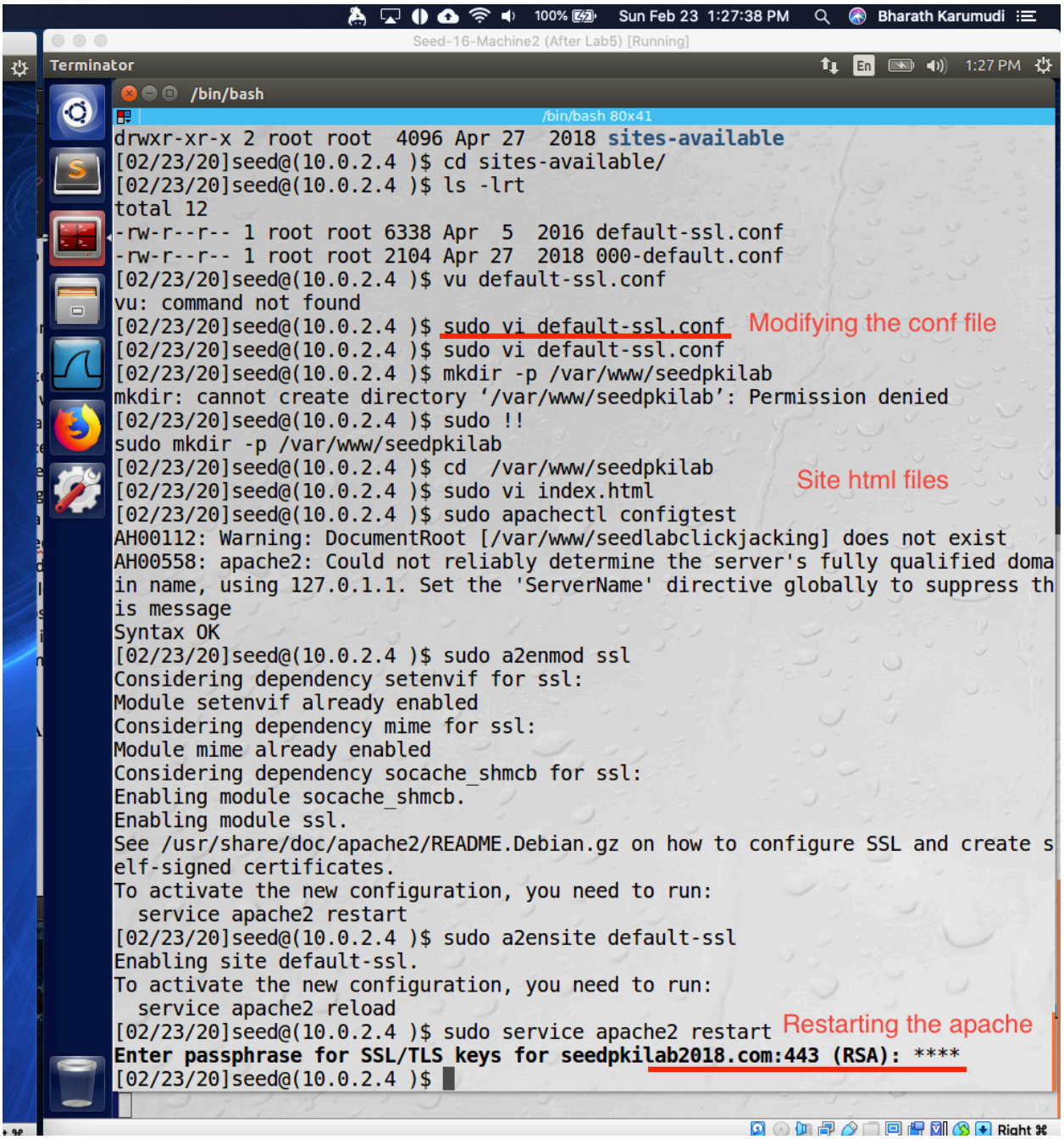


Fig1: Modifications to apache

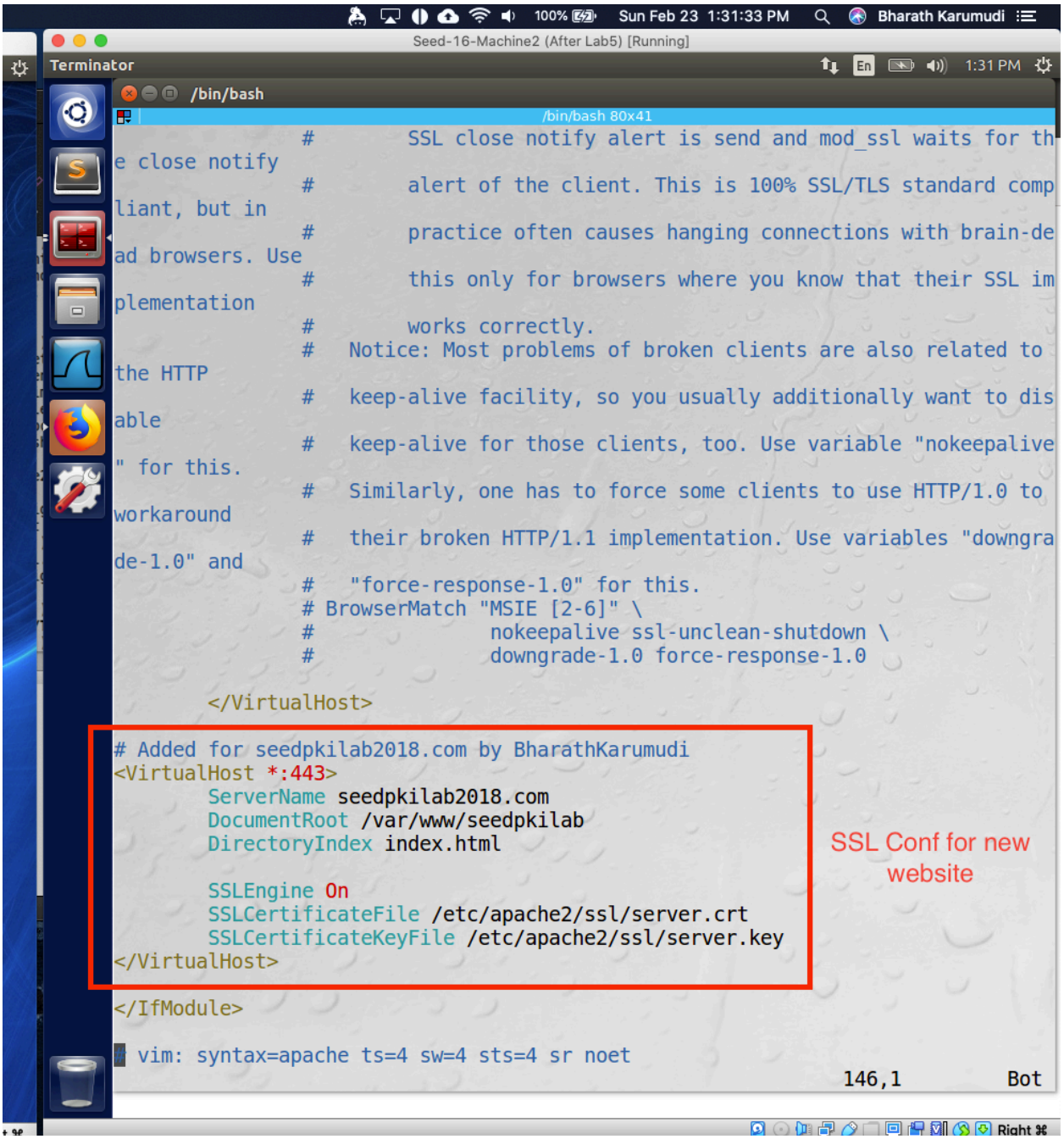


Fig2: default-ssl.conf file modifications

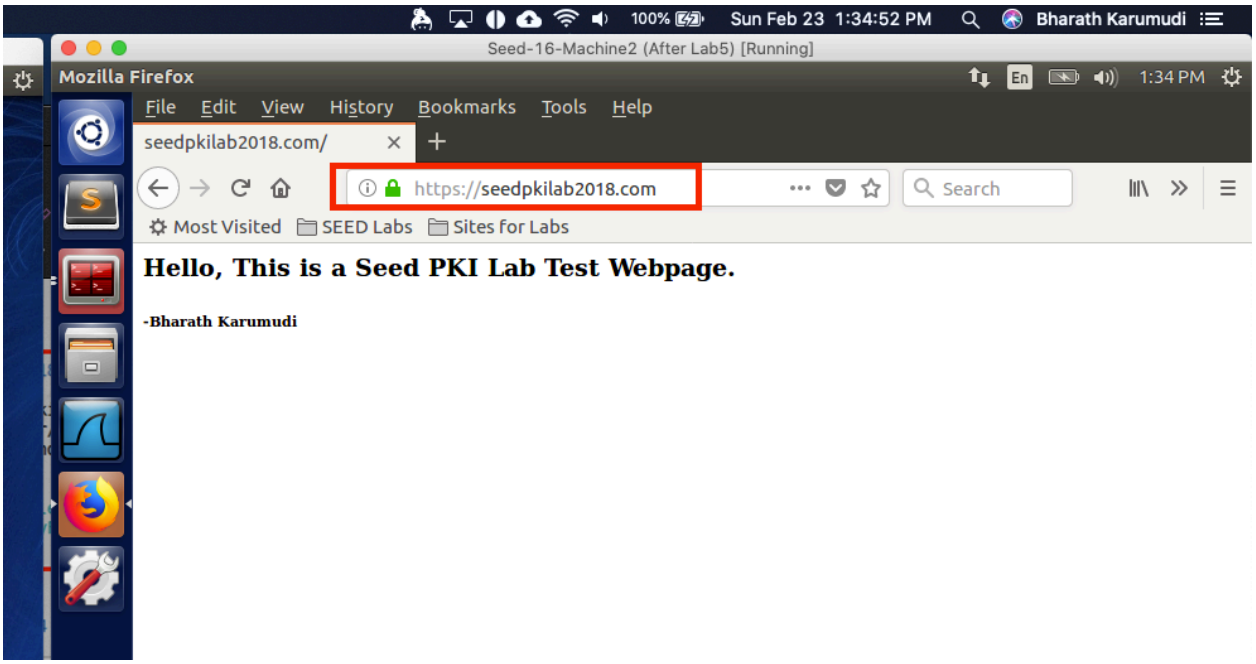


Fig3: Website running on SSL and apache

Observation: Modified the apache server to host the new website seedpkilab2018.com and configured the SSL details in default-ssl.conf file. Once restarted the apache, the site is now running on HTTPS without content.

Explanation:

1. Modified the default-ssl.conf file under /etc/apache2/sites-enabled/ to add the new virtualhost to listen on port 443. Also configured the site private key and CA signed certificate as shown in Fig1 and Fig2.
2. Created the site home as /var/www/seedpkilab/ and created a index.html
3. Ran the following commands to test the configuration and also to enable the SSL modules:
    a. sudo apachectl configtest
    b. sudo a2enmod ssl
    c. sudo a2ensite default-ssl
4. Finally restarted the apache service to make the configurations effective and on restart apache prompted the site private key (server.key) password and entered and started the service.
5. From browser accessed the https://seedpkilab2018.com and the site came up on SSL with our content (Fig3). As the CA self-signed cert is already on the Firefox trust store, the Firefox didn't issue any warnings or errors. The connection now uses encrypted communications between Client and Server.

Thus, configured the CA issued Certificate on Apache Webserver.