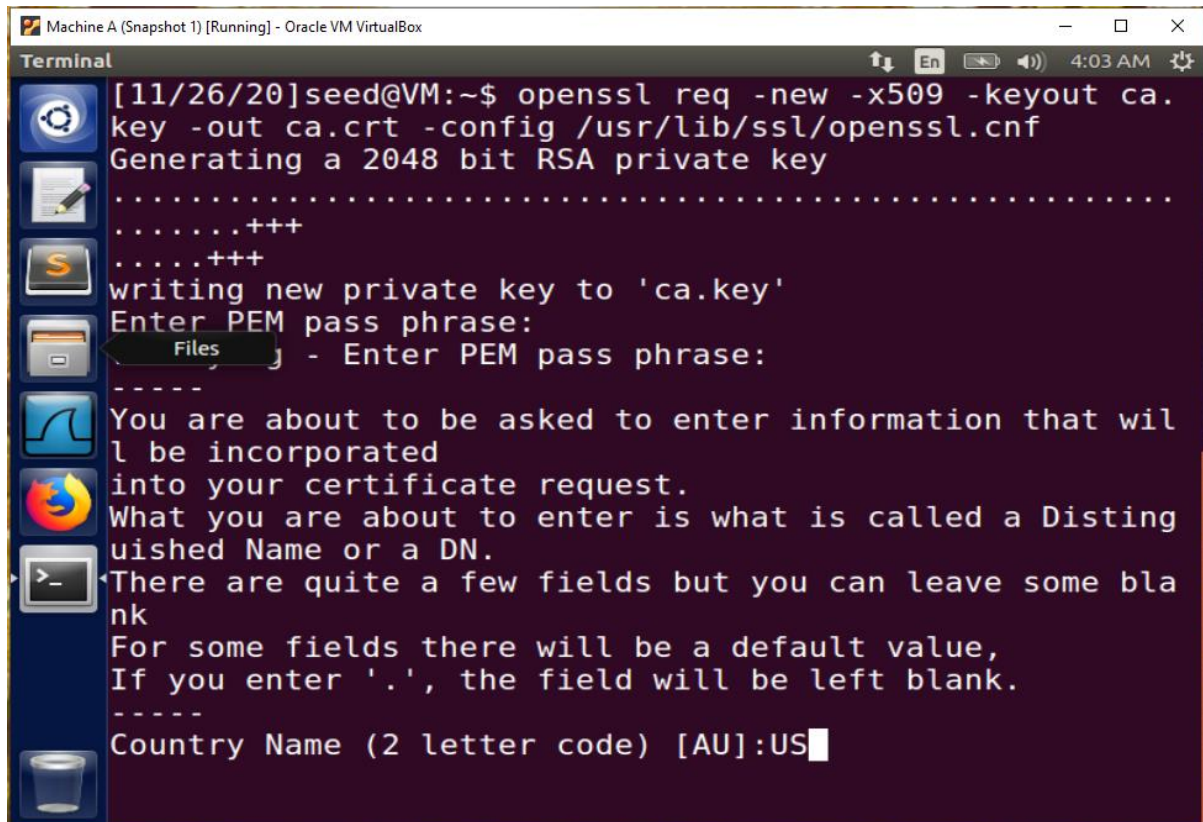


Public-Key Infrastructure (PKI) Lab

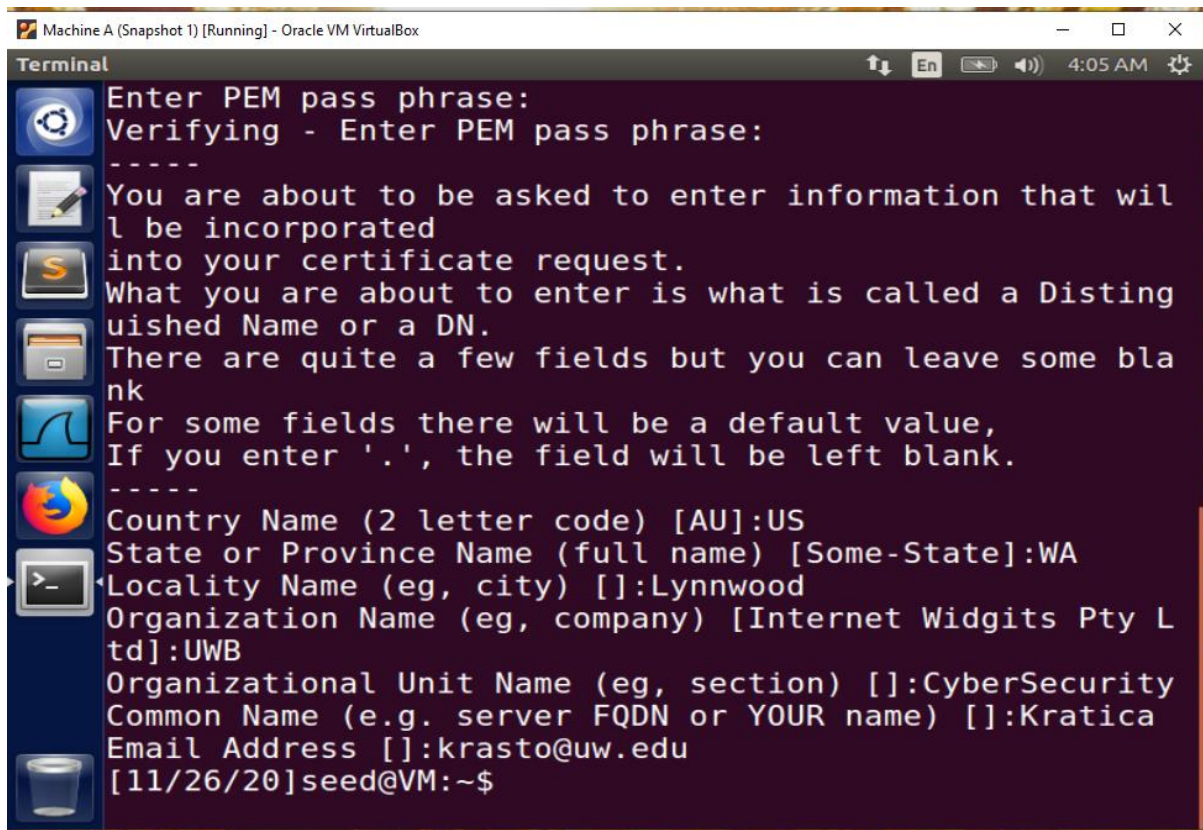
Task 1: Becoming a Certificate Authority (CA)

- To generate a self-signed certificate for CA. This CA is totally trusted, and its certificate will serve as the root certificate. After running the given command, the self-signed certificate for the CA can be generated.

`openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf`



```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
[11/26/20]seed@VM:~$ openssl req -new -x509 -keyout ca.
key -out ca.crt -config /usr/lib/ssl/openssl.cnf
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Files - Enter PEM pass phrase:
-----
You are about to be asked to enter information that wil
l be incorporated
into your certificate request.
What you are about to enter is what is called a Disting
uished Name or a DN.
There are quite a few fields but you can leave some bla
nk
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
```



Task 2: Creating a Certificate for SEEDPKILab2018.com

Step 1: Generate public/private key pair

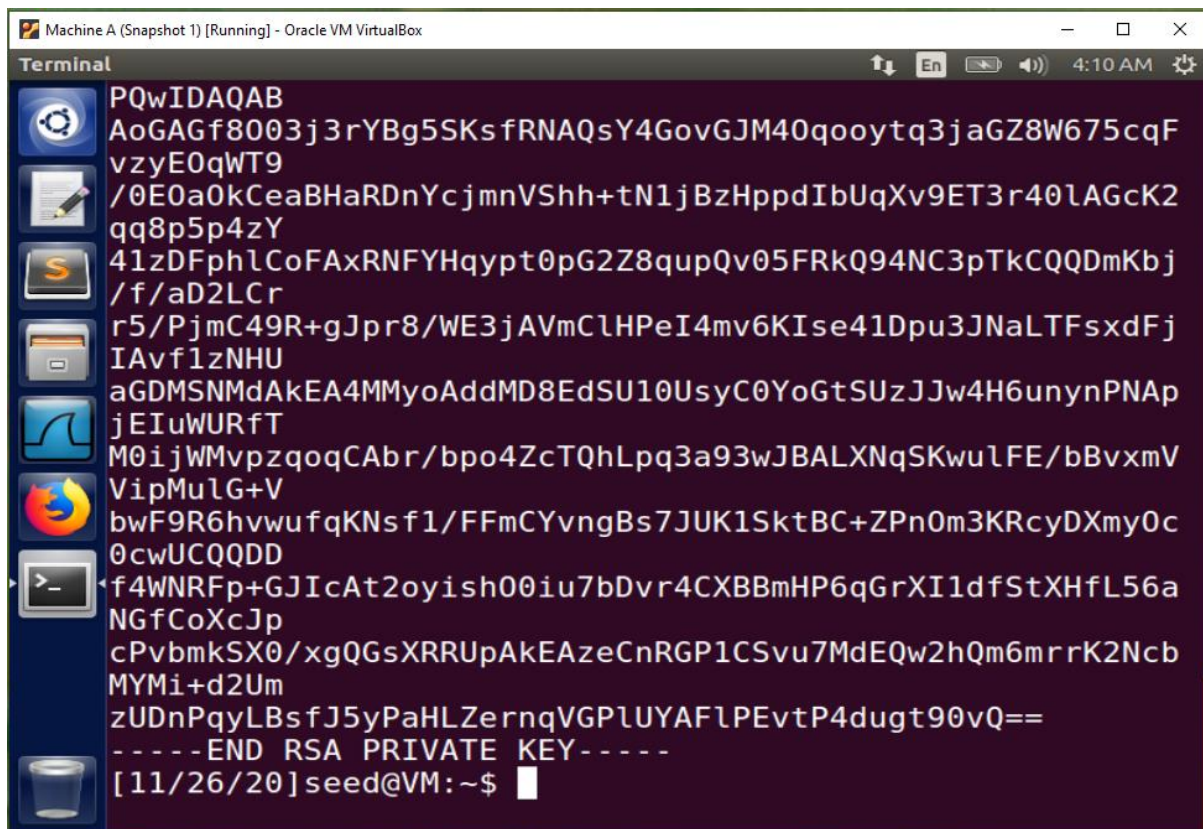
- The company is required to create its own public/private key pair. The command given below will generate an RSA key pair (both private and public keys). The private key password will be encrypted using AES-128 encryption algorithm.
`openssl genrsa -aes128 -out server.key 1024`
- The keys will be stored in the file server.key


```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
[11/26/20]seed@VM:~$ ls
android  ca.key  Documents  Music  source
bin      Customization  Downloads  Pictures  Templates
ca.crt   Desktop  lib        Public  Videos
[11/26/20]seed@VM:~$ openssl genrsa -aes128 -out server
.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
3070600896:error:28069065:lib(40):UI_set_result:result
too small:ui_lib.c:823:You must type in 4 to 1023 chara
cters
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
[11/26/20]seed@VM:~$ ls
android  Customization  lib  server.key
bin      Desktop        Music  source
ca.crt   Documents      Pictures  Templates
ca.key   Downloads      Public  Videos
[11/26/20]seed@VM:~$
```

- The key stored generated file server.key is an encrypted text file. Therefore, the content in the file are modulus and exponents. The following command will show the modulus and private exponent in file:

openssl rsa -in server.key -text

```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[11/26/20]seed@VM:~$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
 00:ca:14:01:2b:81:6a:9f:69:d4:bf:28:d4:0e:33:
 d7:7e:b9:5e:8b:f9:95:70:81:55:cb:fb:6c:36:cb:
 90:54:59:2c:10:41:41:aa:66:66:b5:6b:f0:29:e4:
 59:a4:74:29:b8:24:0d:cd:fb:99:2a:ca:e8:1f:d4:
 87:db:2c:fe:d6:e5:db:ee:74:a9:f1:94:cf:3b:03:
 42:55:1d:c6:fb:97:93:39:2e:6b:0a:04:82:5a:57:
 9b:82:b1:9f:de:05:12:e1:0b:ec:3d:71:d4:64:95:
 ee:5c:6d:39:18:85:ed:e9:fd:a2:c8:db:1b:19:d1:
 f8:ff:c3:61:1e:65:aa:4f:43
publicExponent: 65537 (0x10001)
privateExponent:
 19:ff:0e:d3:78:f7:ad:80:60:e5:22:ac:7d:13:40:
 42:c6:38:1a:8b:c6:24:ce:0e:aa:8a:32:b6:ad:e3:
 68:66:7c:5b:ae:f9:72:a1:6f:cf:21:0e:a9:64:fd:
 ff:41:0e:68:e9:02:79:a0:47:69:10:e7:61:c8:e6:
 9d:54:a1:87:eb:4d:d6:30:73:1e:9a:5d:21:b5:2a:
 5e:ff:44:4f:7a:f8:d2:50:06:70:ad:aa:ab:ca:79:
 a7:8c:d8:e3:5c:c3:16:98:65:0a:81:40:c5:13:45:
```



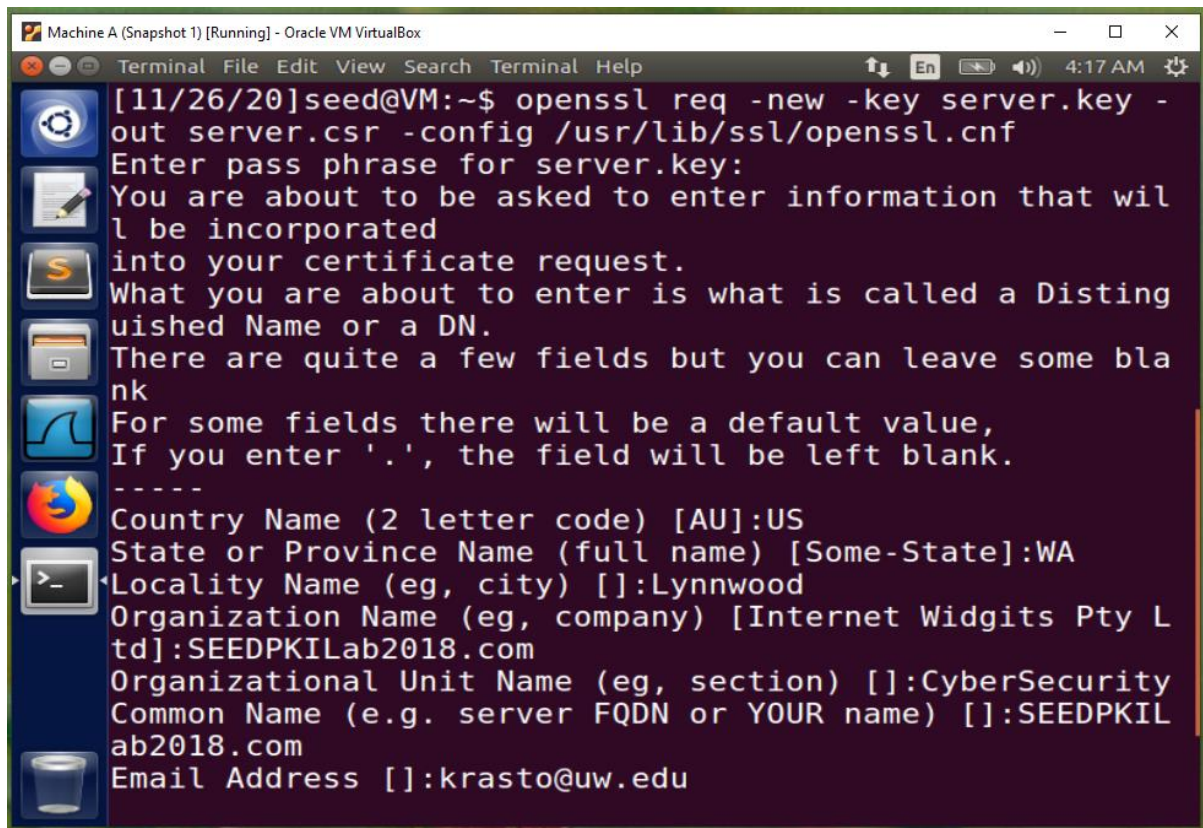
The screenshot shows a terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output displays the result of an RSA key generation process. It starts with a header "PQwIDAQAB", followed by a long base64-encoded string representing the public key. This is followed by another long base64-encoded string representing the private key. The output concludes with "-----END RSA PRIVATE KEY-----" and a timestamp "[11/26/20]seed@VM:~\$".

```
PQwIDAQAB
AoGAGf8003j3rYBg5SKsfRNAQsY4GovGJM40qooytq3jaGZ8W675cqF
vzyE0qWT9
/0E0a0kCeaBHaRDnYcjmVShh+tN1jBzHppdIbUqXv9ET3r40lAGcK2
qq8p5p4zY
41zDFphlCoFAxRNFYHqypt0pG2Z8qupQv05FRkQ94NC3pTkCQQDmKbj
/f/aD2LCr
r5/PjmC49R+gJpr8/WE3jAVmClHPeI4mv6KIse41Dpu3JNaLTFsxdFj
IAvf1zNHU
aGDMSNMdAKeA4MMyoAddMD8EdSU10UusyC0YoGtSUzJJw4H6unynPNap
jEIuWURfT
M0ijWMvpzqoqCAbr/bpo4ZcTQhLpq3a93wJBALXNqSKwu1FE/bBvxmV
VipMulG+V
bwF9R6hvwufqKNsf1/FFmCYvngBs7JUK1SktBC+ZPn0m3KRcyDXmyOc
0cwUCQQDD
f4WNRfp+GJIcAt2oyish00iu7bDvr4CXBBmHP6qGrXI1dfStXHfL56a
NGfCoXcJp
cPvbmksX0/xgQGsxRRUpAkeAzeCnRGP1CSvu7MdeQw2hQm6mrrK2Ncb
MYMi+d2Um
zUDnPqyLBsfJ5yPaHLZernqVGP1UYAF1PEvtP4dugt90vQ==
-----END RSA PRIVATE KEY-----
[11/26/20]seed@VM:~$
```

Step 2: Generate a Certificate Signing Request (CSR)

- Now the company has the server.key file. By using given command, the server.key file will generate the Certificate Signing Request (CSR) which contains SEEDPKILab2018.com public key.

openssl req -new -key server.key -out server.csr -config openssl.cnf



The screenshot shows a terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
[11/26/20]seed@VM:~$ openssl req -new -key server.key -out server.csr -config /usr/lib/ssl/openssl.cnf
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:WA
Locality Name (eg, city) []:Lynnwood
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SEEDPKILab2018.com
Organizational Unit Name (eg, section) []:CyberSecurity
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2018.com
Email Address []:krasto@uw.edu
```

Step 3: Generating Certificates

- Now to generate a certificate, the CSR file needs to have the CA's signature. Here I am using own trusted CA to generate certificates. The command given below will turn the certificate signing request (server.csr) into an X509 certificate (server.crt), using the CA's ca.crt and ca.key.
`openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf`
- I observed that the certificate has been generated successfully. Refer below snapshot:

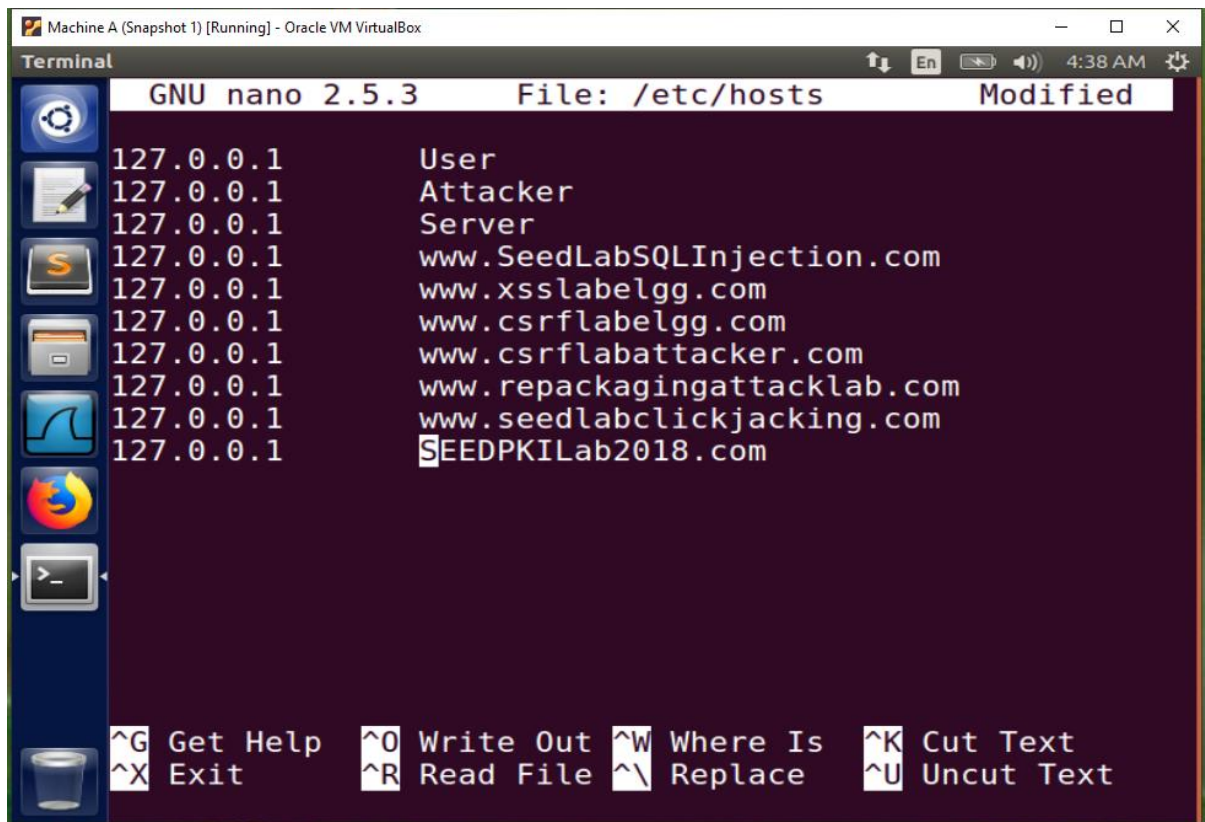
```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
countryName = US
stateOrProvinceName = WA
organizationName = SEEDPKILab2018.
com
organizationalUnitName = CyberSecurity
commonName = SEEDPKILab2018.
com
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
EC:98:DA:24:5D:E6:C6:74:44:62:3A:87:46:
7D:EC:82:A0:7E:47:51
X509v3 Authority Key Identifier:
keyid:56:DF:AB:AD:46:E8:76:18:8C:46:ED:
BA:3F:2D:34:DA:3F:C1:0D:BB
Certificate is to be certified until Nov 26 09:35:33 20
21 GMT (365 days)
Sign the certificate? [y/n]:
```

```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
EC:98:DA:24:5D:E6:C6:74:44:62:3A:87:46:
7D:EC:82:A0:7E:47:51
X509v3 Authority Key Identifier:
keyid:56:DF:AB:AD:46:E8:76:18:8C:46:ED:
BA:3F:2D:34:DA:3F:C1:0D:BB
Certificate is to be certified until Nov 26 09:35:33 20
21 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]
y
Write out database with 1 new entries
Data Base Updated
[11/26/20]seed@VM:~$
```

Task 3: Deploying Certificate in an HTTPSWeb Server

Step 1: Configuring DNS

Since SEEDPKILab2018.com website is being used in the experiment. Now configuring the entry of this website into /etc/hosts to localhost (127.0.0.1) as shown in below snapshot:



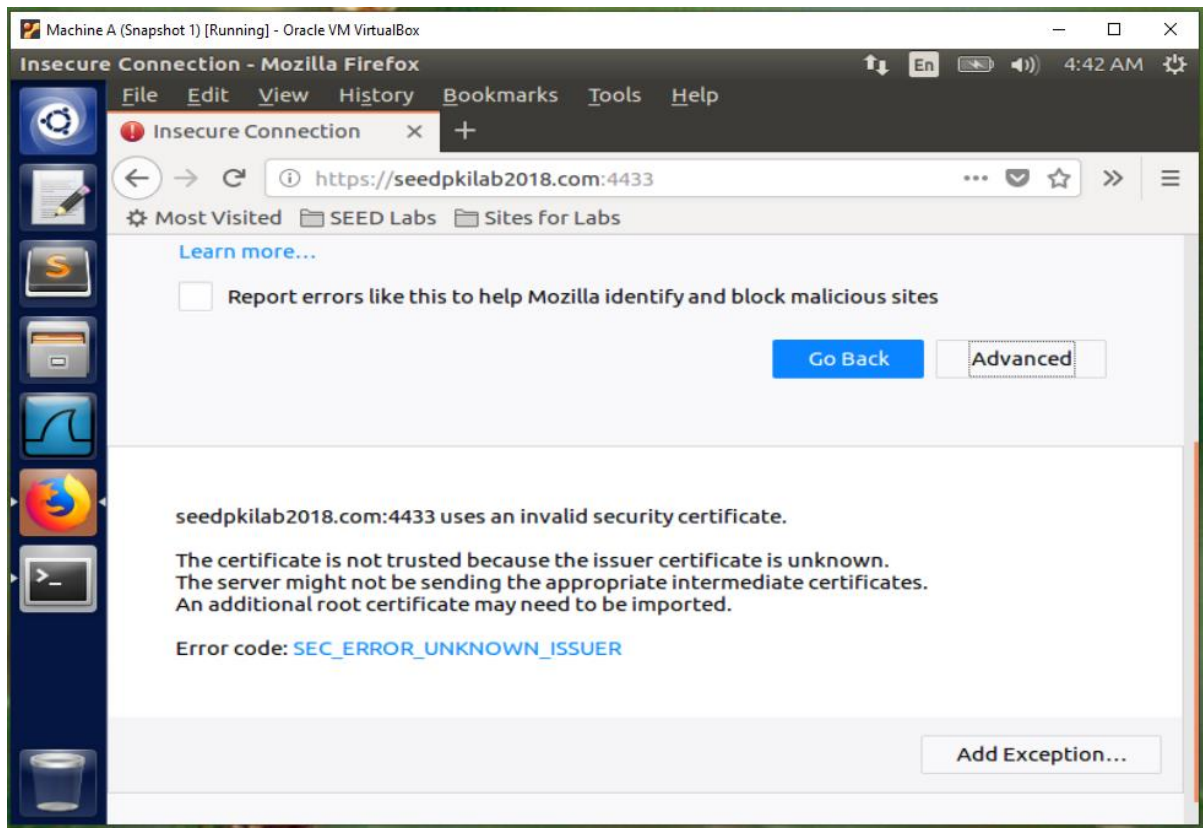
The screenshot shows a terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal is running GNU nano 2.5.3, editing the file /etc/hosts. The content of the file is as follows:

```
127.0.0.1      User
127.0.0.1      Attacker
127.0.0.1      Server
127.0.0.1      www.SeedLabSQLInjection.com
127.0.0.1      www.xsslabelgg.com
127.0.0.1      www.csrflabelgg.com
127.0.0.1      www.csrfabattacker.com
127.0.0.1      www.repackagingattacklab.com
127.0.0.1      www.seedlabclickjacking.com
127.0.0.1      SEEDPKILab2018.com
```

The terminal also shows a sidebar with various application icons and a bottom status bar with keyboard shortcuts: ^G Get Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\ Replace, ^K Cut Text, and ^U Uncut Text.

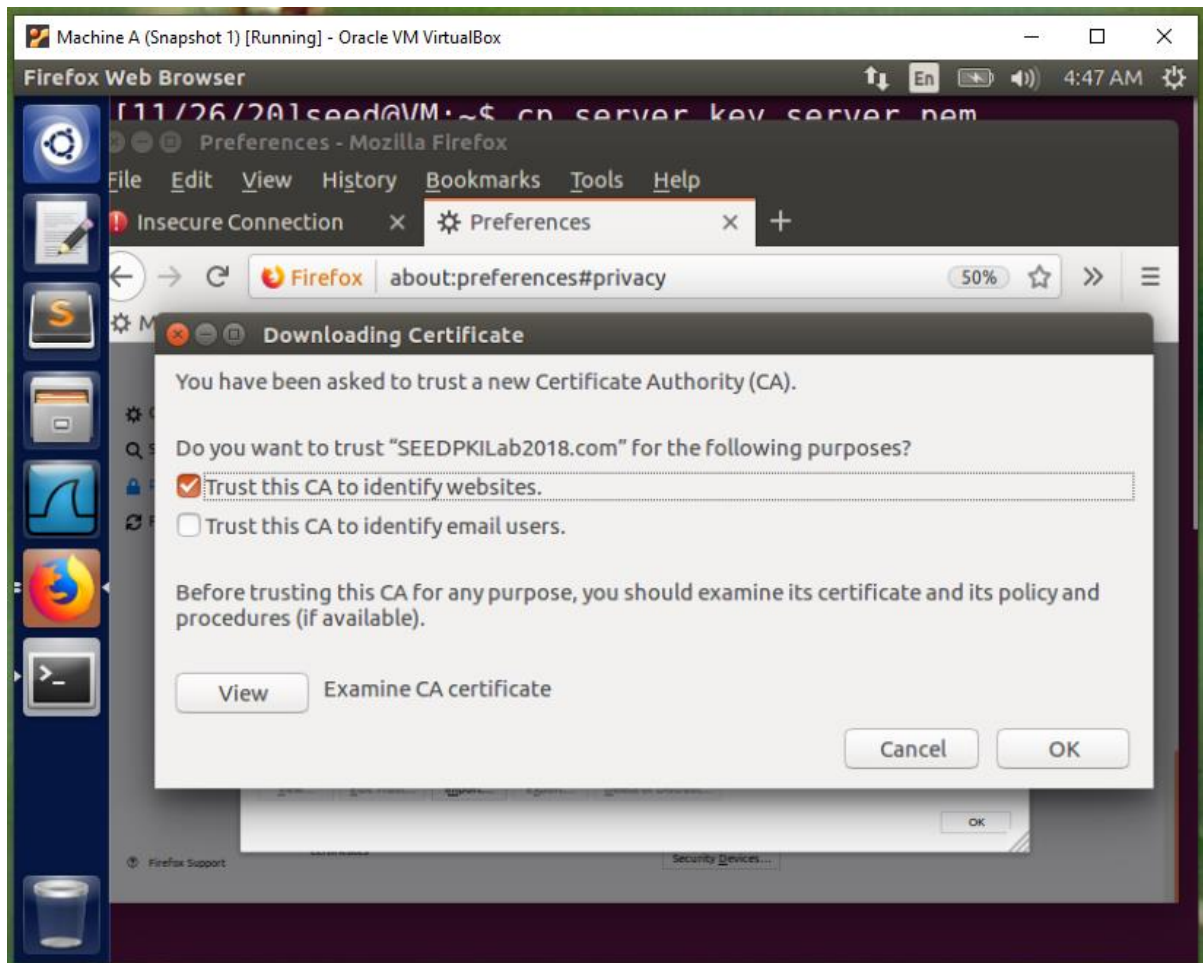
Step 2: Configuring the web server

- Firstly, combining the secret key and the certificate into one file by using given command:
`cp server.key server.pem`
`cat server.crt >> server.pem`
- By executing given OpenSSL command, the simple web server with generated certificate will launch.
`openssl s_server -cert server.pem -www`
- When I hit <https://seedpkilab2018.com:4433/> on Firefox browser, I observed error message.
"seedpkilab2018.com:4433 uses an invalid security certificate. The certificate is not trusted because the issuer certificate is unknown".
Refer below snapshot:



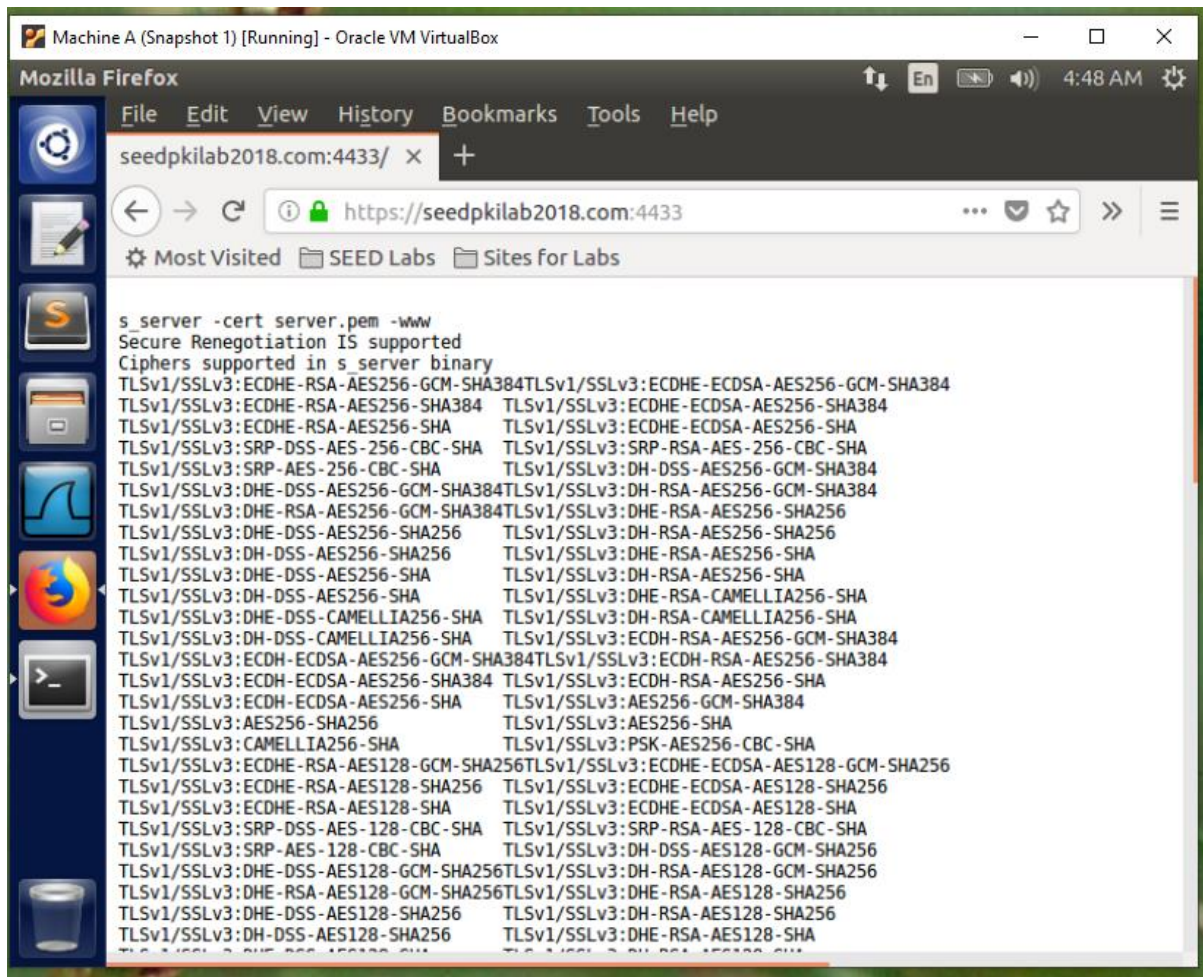
Step 3: Getting the browser to accept our CA certificate

- SEEDPKILab2018.com certificate is signed by CA (i.e., ca.crt), and Firefox does not accept this CA. So, we will manually add the ca.crt to the Firefox browser.
- Enabling "Trust this CA to identify websites" option while importing the certificate.
- Refer below snapshot showing ca.crt added to Firefox browser.



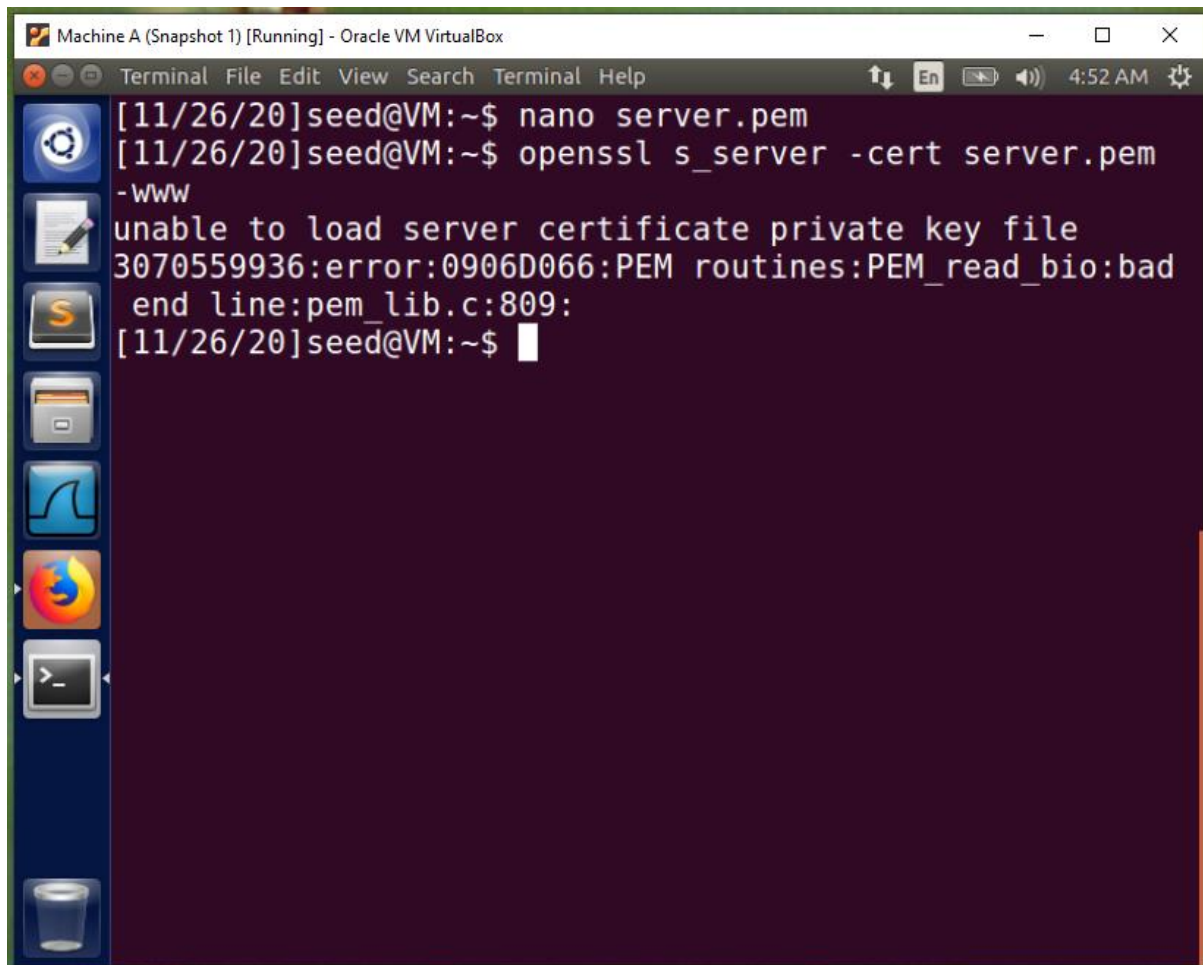
Step 4: Testing our HTTPS website

- After importing the CA certificate in Firefox browser. It will show the CA certificate in the certificate accepted list. When I browsed <https://SEEDPKILab2018.com:4433> website on Firefox browser, it didn't give error. Refer given snapshot:



Step 4: (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.

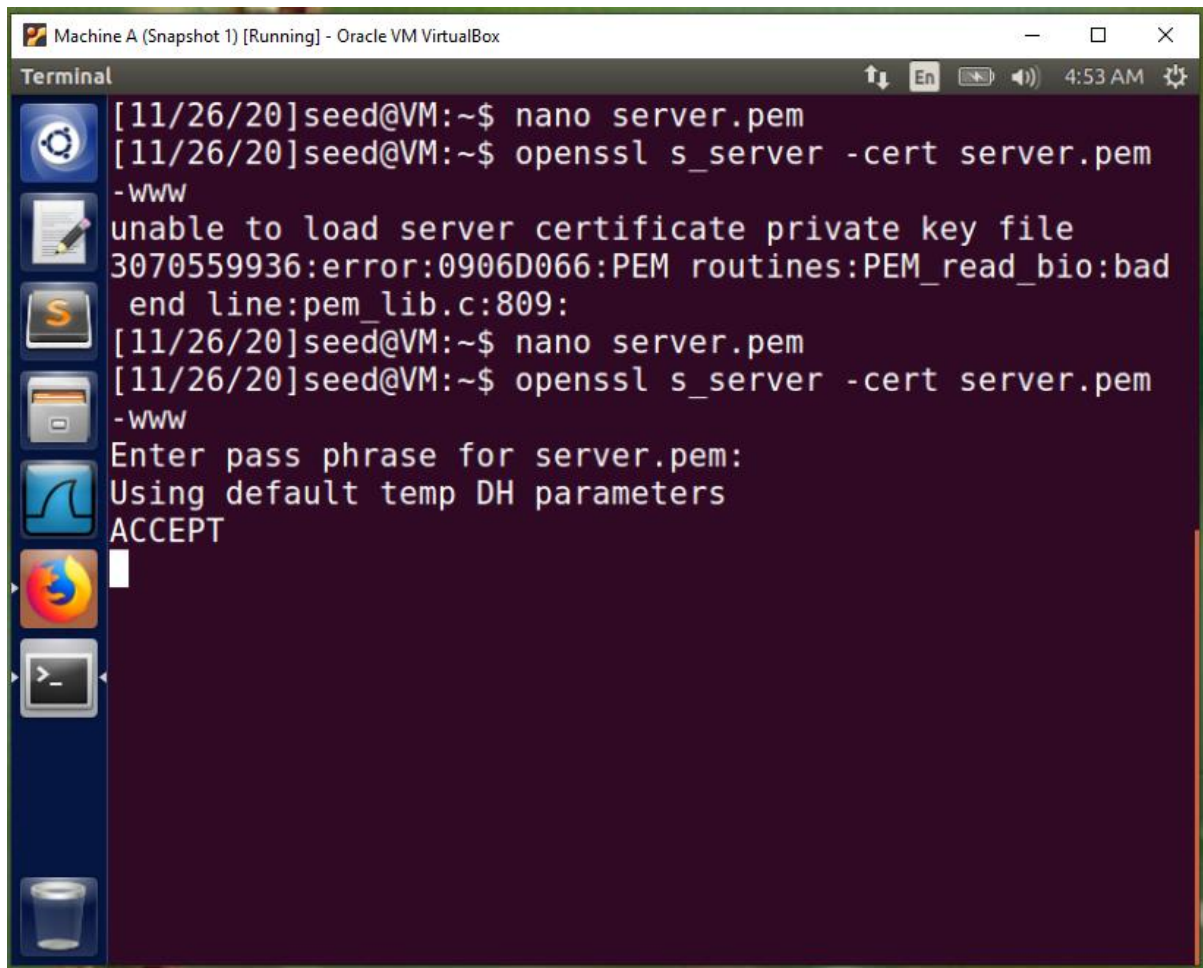
- After modifying the bytes in server.pem file, I restarted the server using below command:
openssl s_server -cert server.pem -www
- Observed error “unable to load server certificate private key file”. Hence,
<https://SEEDPKILab2018.com:4433> did not worked.



The screenshot shows a terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "Terminal", "File", "Edit", "View", "Search", "Terminal", and "Help". On the left is a vertical toolbar with icons for settings, a document, a terminal, a folder, a graph, a web browser, a terminal icon, and a trash can. The terminal output is as follows:

```
[11/26/20]seed@VM:~$ nano server.pem
[11/26/20]seed@VM:~$ openssl s_server -cert server.pem
- www
unable to load server certificate private key file
3070559936:error:0906D066:PEM routines:PEM_read_bio:bad
end line:pem_lib.c:809:
[11/26/20]seed@VM:~$
```

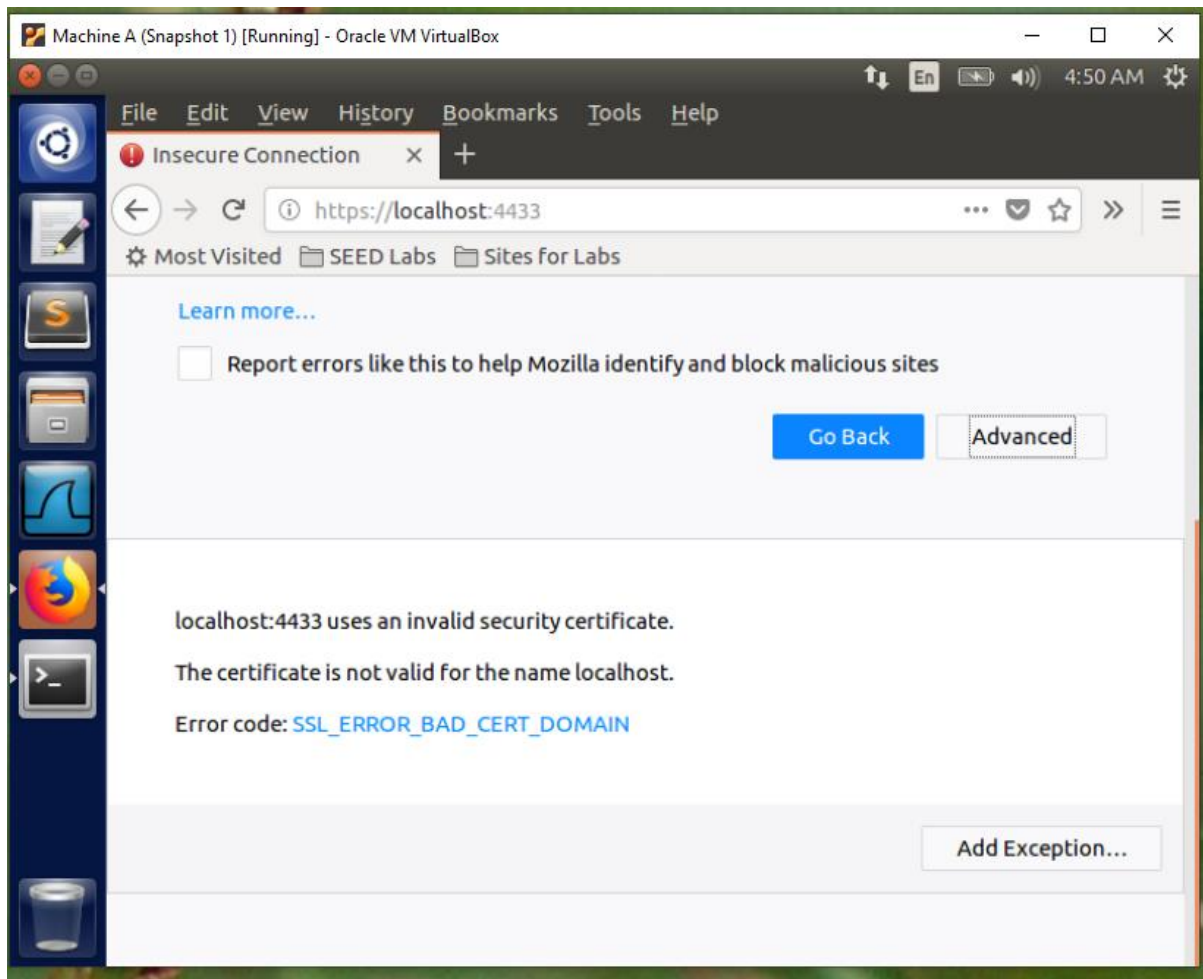
- Restoring back the original server.pem file. Then restarted the server using command `openssl s_server -cert server.pem -www`
- The given snapshot is showing ACCEPT which means that the server has been restarted successfully.



```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
[11/26/20]seed@VM:~$ nano server.pem
[11/26/20]seed@VM:~$ openssl s_server -cert server.pem -www
unable to load server certificate private key file
3070559936:error:0906D066:PEM routines:PEM_read_bio:bad
end line:pem_lib.c:809:
[11/26/20]seed@VM:~$ nano server.pem
[11/26/20]seed@VM:~$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

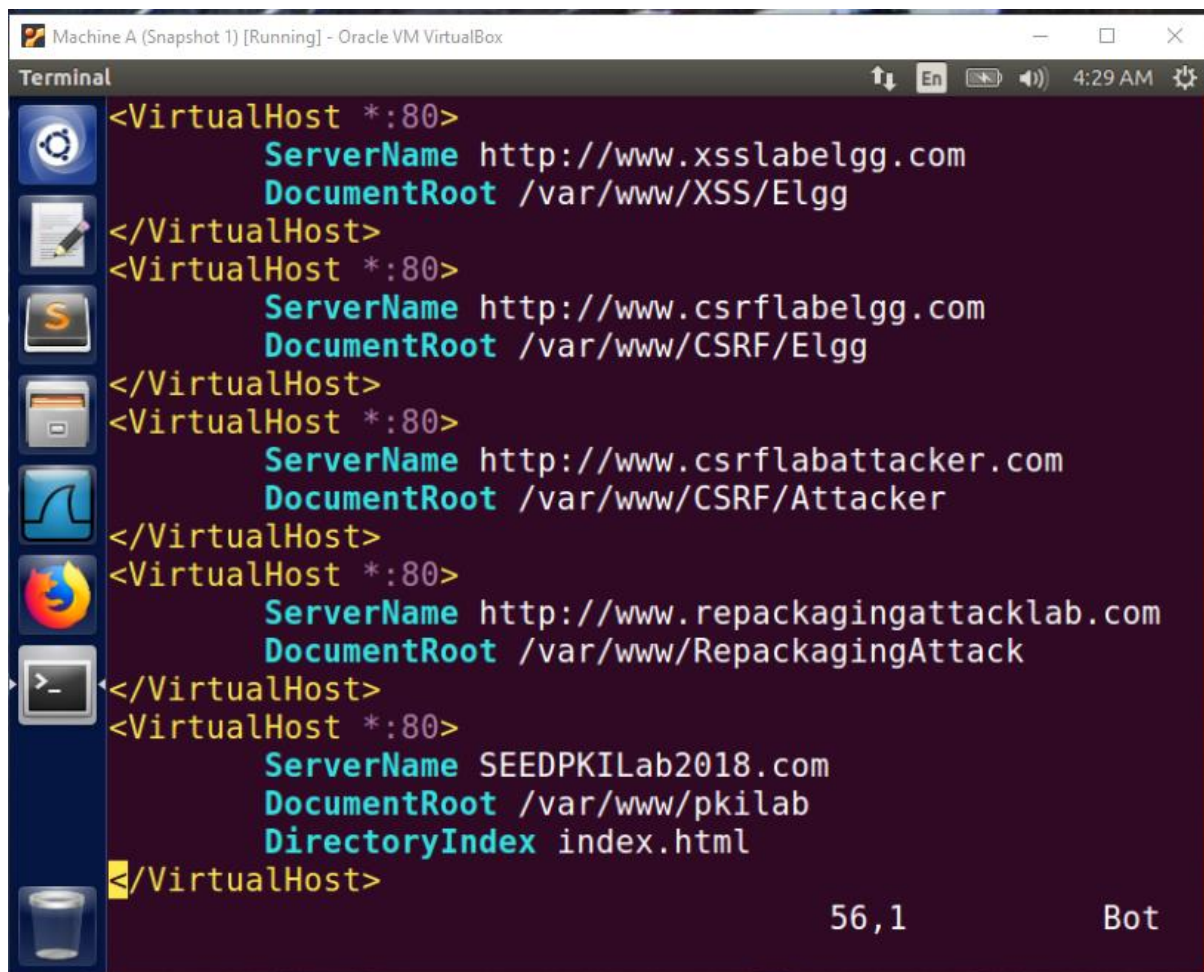
Step 4: (2) Since SEEDPKILab2018.com points to the localhost, if we use <https://localhost:4433> instead, we will be connecting to the same web server. Please do so, describe and explain your observations.

After browsing <https://localhost:4433> on Firefox browser, I observed that the browser gave Bad cert domain "The certificate is not valid for the name localhost". The certificate is tied to SEEDPKILab2018.com not to localhost. Refer below snapshot showing error message:



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

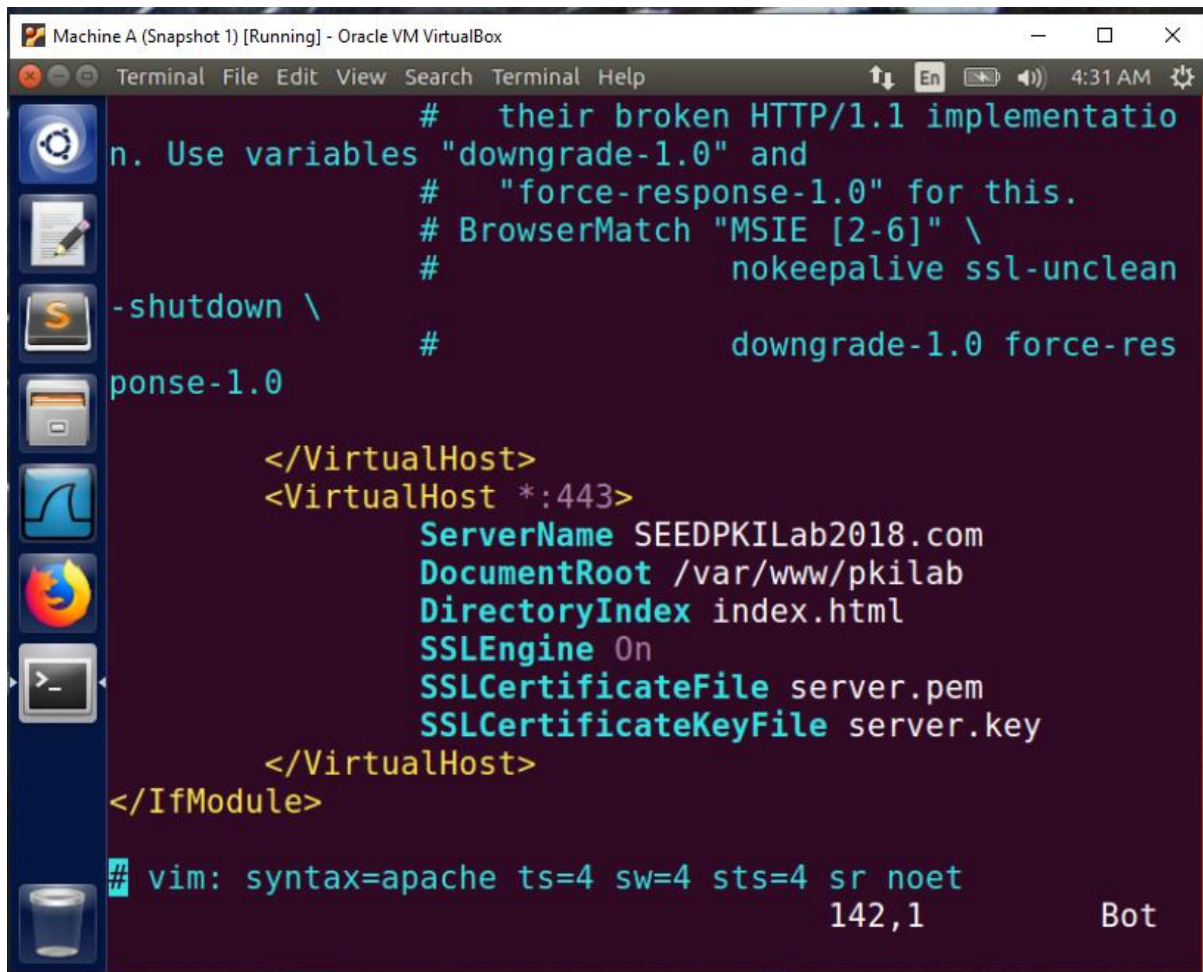
- Configuring an Apache server to build an HTTPS website i.e SEEDPKIlab2018.com, so it knows where to get the private key and certificates.
- An Apache server can host multiple websites simultaneously. The directory where a website's files are stored needs to be identified via its VirtualHost file. Adding HTTP website entry in /etc/apache2/sites-available directory and 000-default.conf file.



```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
4:29 AM Bot

<VirtualHost *:80>
    ServerName http://www.xsslabelgg.com
    DocumentRoot /var/www/XSS/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrflabelgg.com
    DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrflabattacker.com
    DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.repackagingattacklab.com
    DocumentRoot /var/www/RepackagingAttack
</VirtualHost>
<VirtualHost *:80>
    ServerName SEEDPKILab2018.com
    DocumentRoot /var/www/pkilab
    DirectoryIndex index.html
</VirtualHost>
```

- Now adding HTTPS website's virtual host entry in default-ssl.conf file of /etc/apache2/sites-available directory.

A screenshot of a VirtualBox terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal shows the configuration of an Apache virtual host for SSL. The configuration includes comments about HTTP/1.1 implementation, variables for downgrade and force-response, and a BrowserMatch rule for MSIE. The virtual host is named *:443 and has a ServerName of SEEDPKILab2018.com, DocumentRoot of /var/www/pkilab, and DirectoryIndex of index.html. The SSLEngine is set to On, and the SSLCertificateFile and SSLCertificateKeyFile are specified as server.pem and server.key respectively. The terminal also shows the end of the configuration block and a vim status line at the bottom.

```
# their broken HTTP/1.1 implementation
n. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.
# BrowserMatch "MSIE [2-6]" \
#         nokeepalive ssl-unclean
-shutdown \
#         downgrade-1.0 force-res
ponse-1.0

</VirtualHost>
<VirtualHost *:443>
    ServerName SEEDPKILab2018.com
    DocumentRoot /var/www/pkilab
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile server.pem
    SSLCertificateKeyFile server.key
</VirtualHost>
</IfModule>

vim: syntax=apache ts=4 sw=4 sts=4 sr noet
142,1 Bot
```

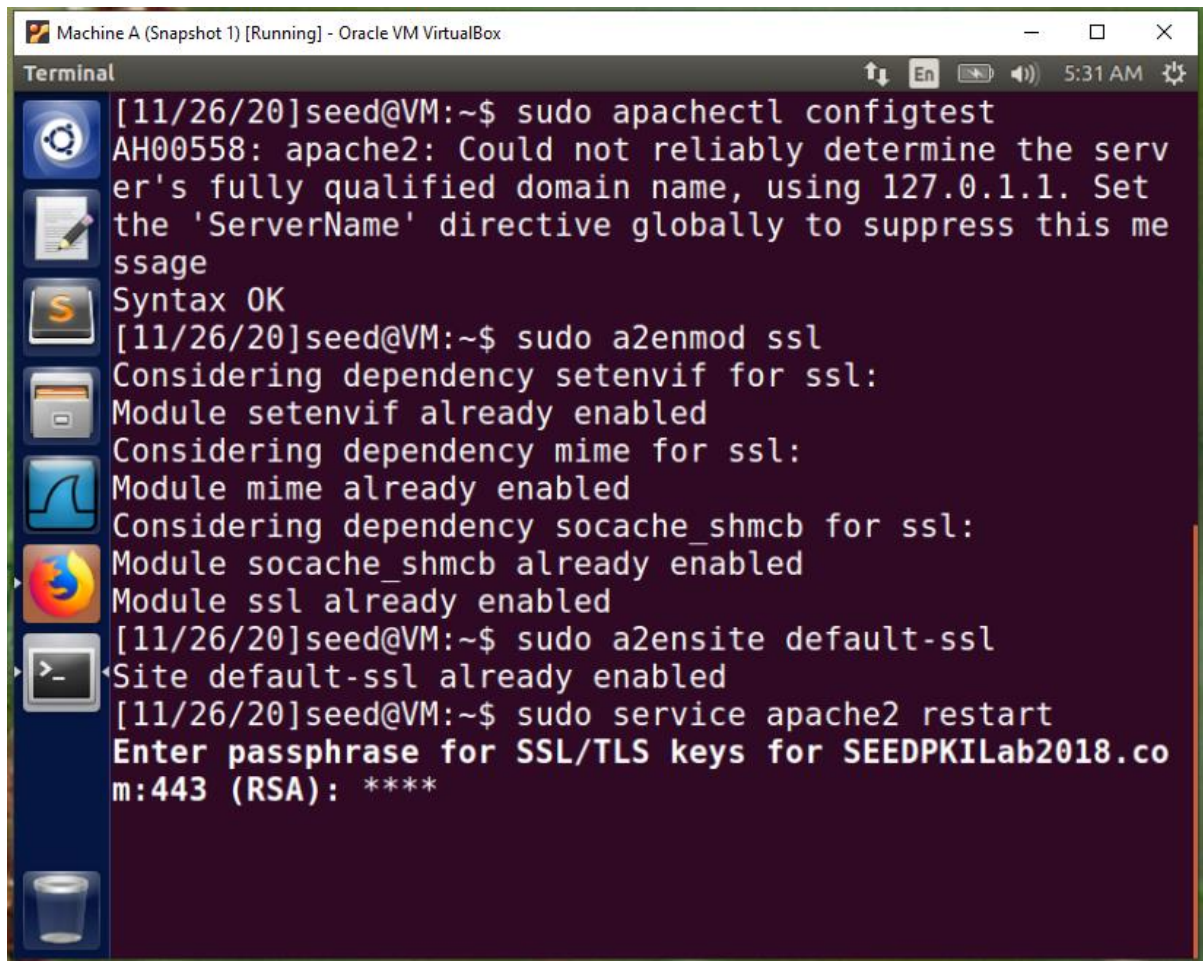
- After modifying both the files in /etc/apache2/sites-available folder, executed below commands to enable SSL. So, that the traffic between the browser and the server will be encrypted while browsing the site.

sudo apachectl configtest

sudo a2enmod ssl

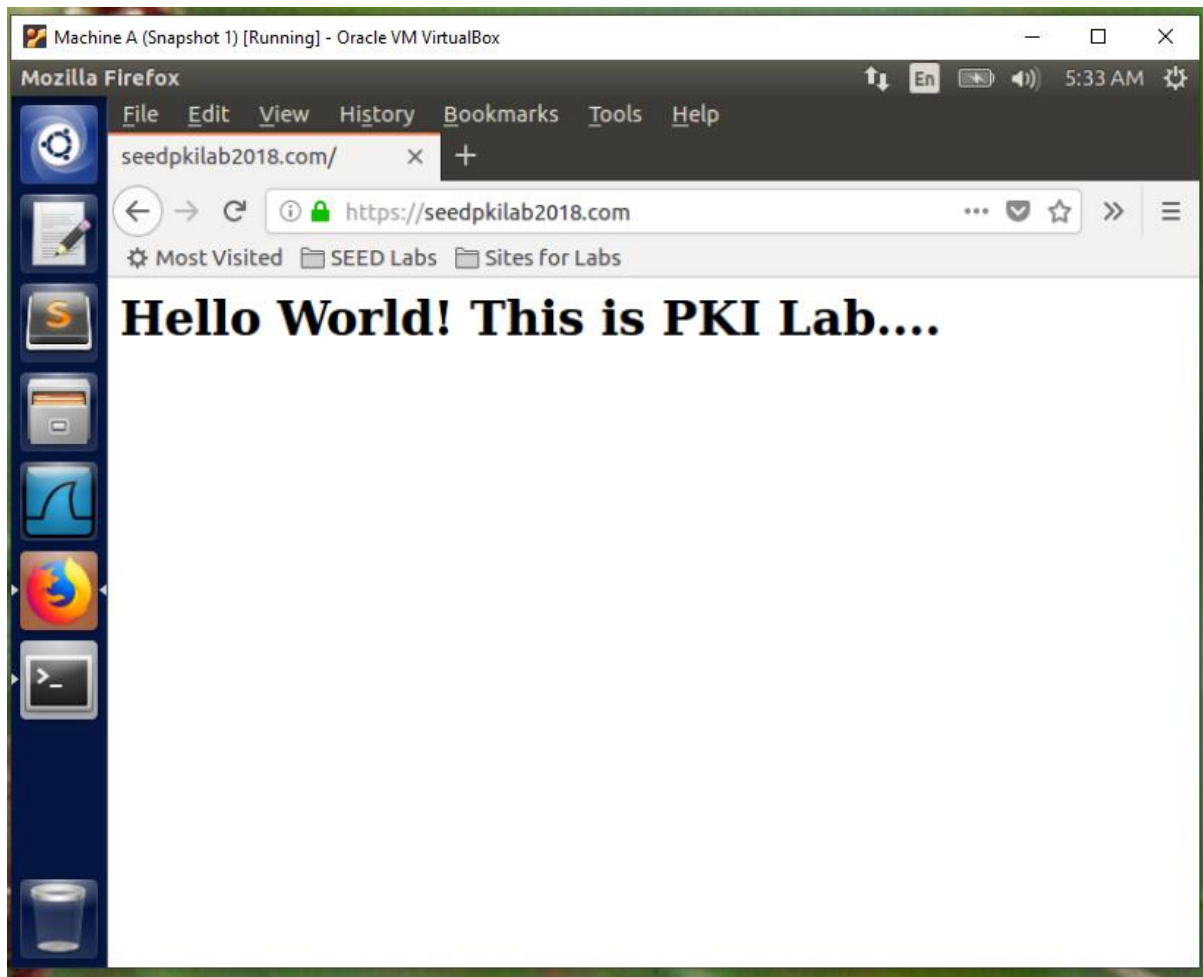
sudo a2ensite default-ssl

sudo service apache2 restart



```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
[11/26/20]seed@VM:~$ sudo apachectl configtest
AH00558: apache2: Could not reliably determine the serv
er's fully qualified domain name, using 127.0.1.1. Set
the 'ServerName' directive globally to suppress this me
ssage
Syntax OK
[11/26/20]seed@VM:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
[11/26/20]seed@VM:~$ sudo a2ensite default-ssl
Site default-ssl already enabled
[11/26/20]seed@VM:~$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2018.co
m:443 (RSA): ****
```

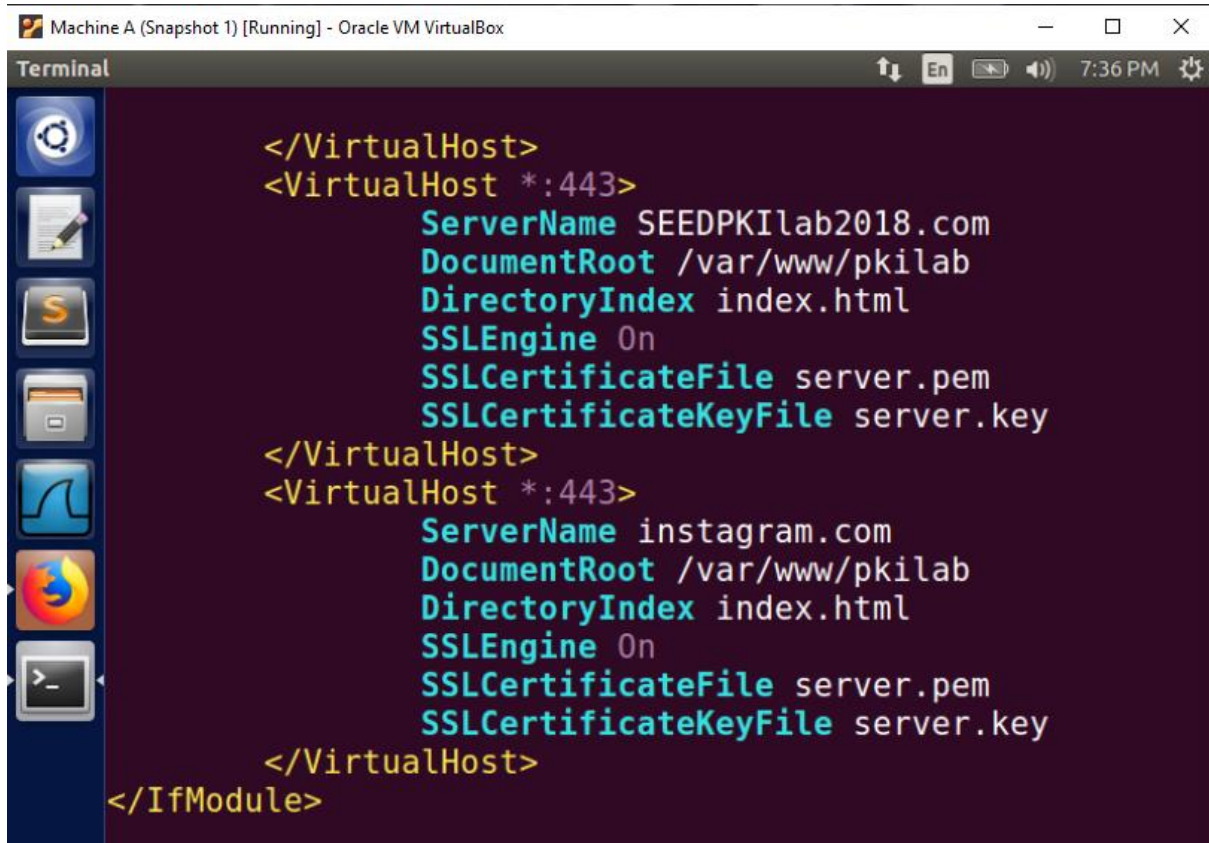
- Now again browse the website [https://seedpkilab2018.com: 443](https://seedpkilab2018.com:443) on Firefox browser and the site was successfully browsed. Refer below snapshot:



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

Step 1: Setting up the malicious website

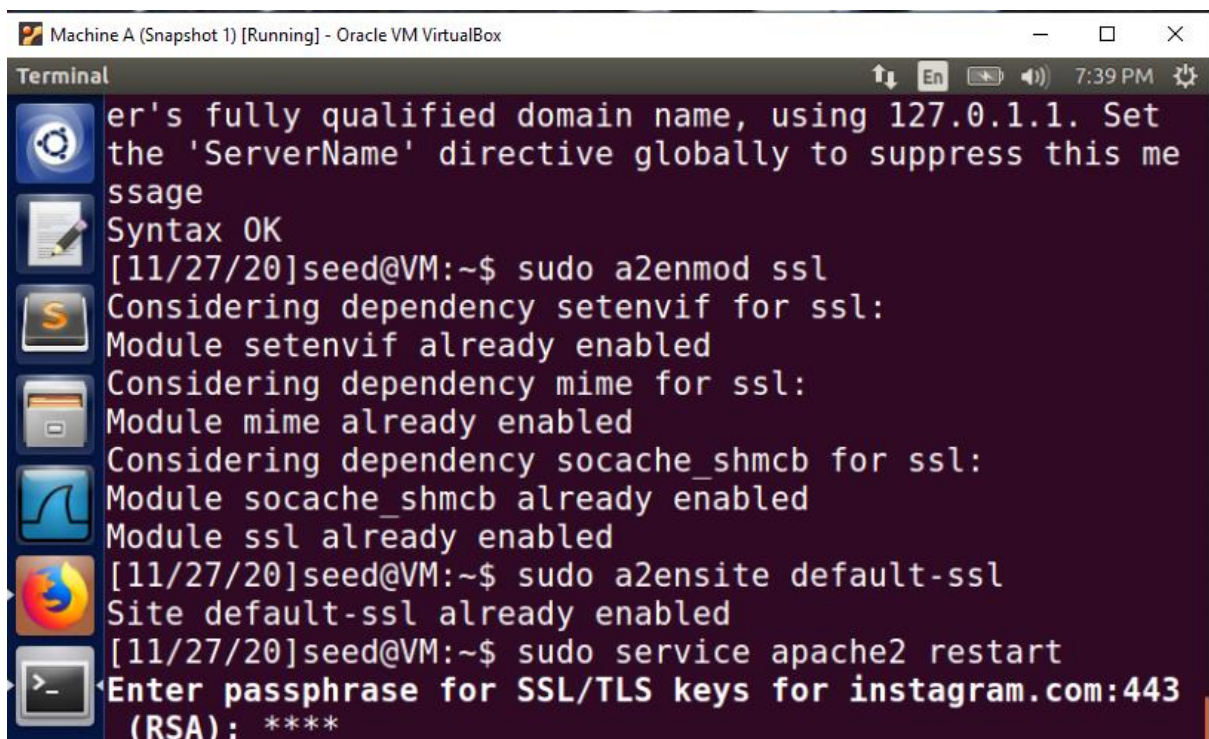
- I have used target website as Instagram.com. Therefore, made virtual host entry in `/etc/apache2/sites-available/default-ssl.conf`



```
</VirtualHost>
<VirtualHost *:443>
    ServerName SEEDPKIlab2018.com
    DocumentRoot /var/www/pkilab
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile server.pem
    SSLCertificateKeyFile server.key
</VirtualHost>
<VirtualHost *:443>
    ServerName instagram.com
    DocumentRoot /var/www/pkilab
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile server.pem
    SSLCertificateKeyFile server.key
</VirtualHost>
</IfModule>
```

- After adding virtual host, restarted apache server using below commands:

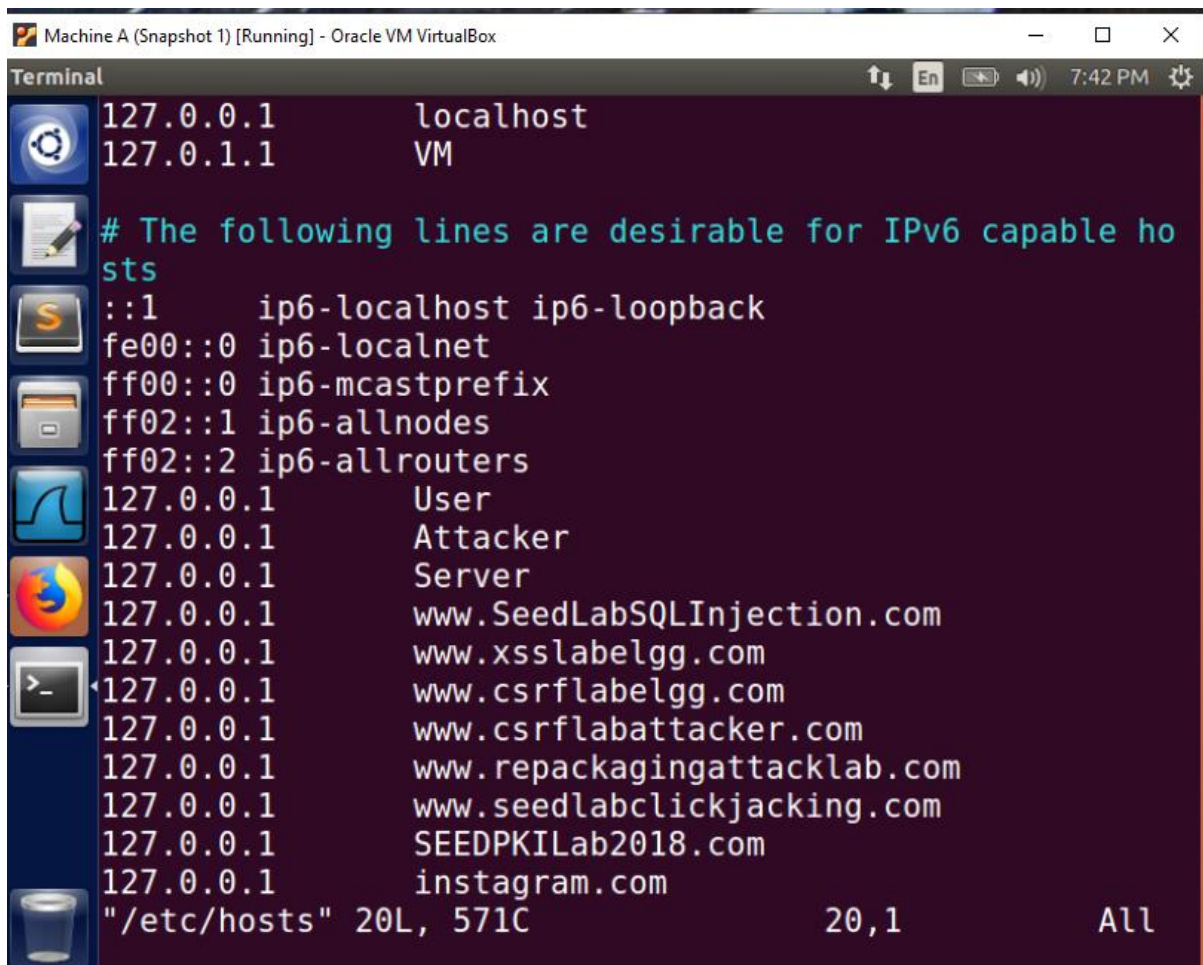
```
sudo apachectl configtest
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo service apache2 restart
```



```
er's fully qualified domain name, using 127.0.1.1. Set
the 'ServerName' directive globally to suppress this me
ssage
Syntax OK
[11/27/20]seed@VM:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
[11/27/20]seed@VM:~$ sudo a2ensite default-ssl
Site default-ssl already enabled
[11/27/20]seed@VM:~$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for instagram.com:443
(RSA): ****
```

Step 2: Becoming the man in the middle

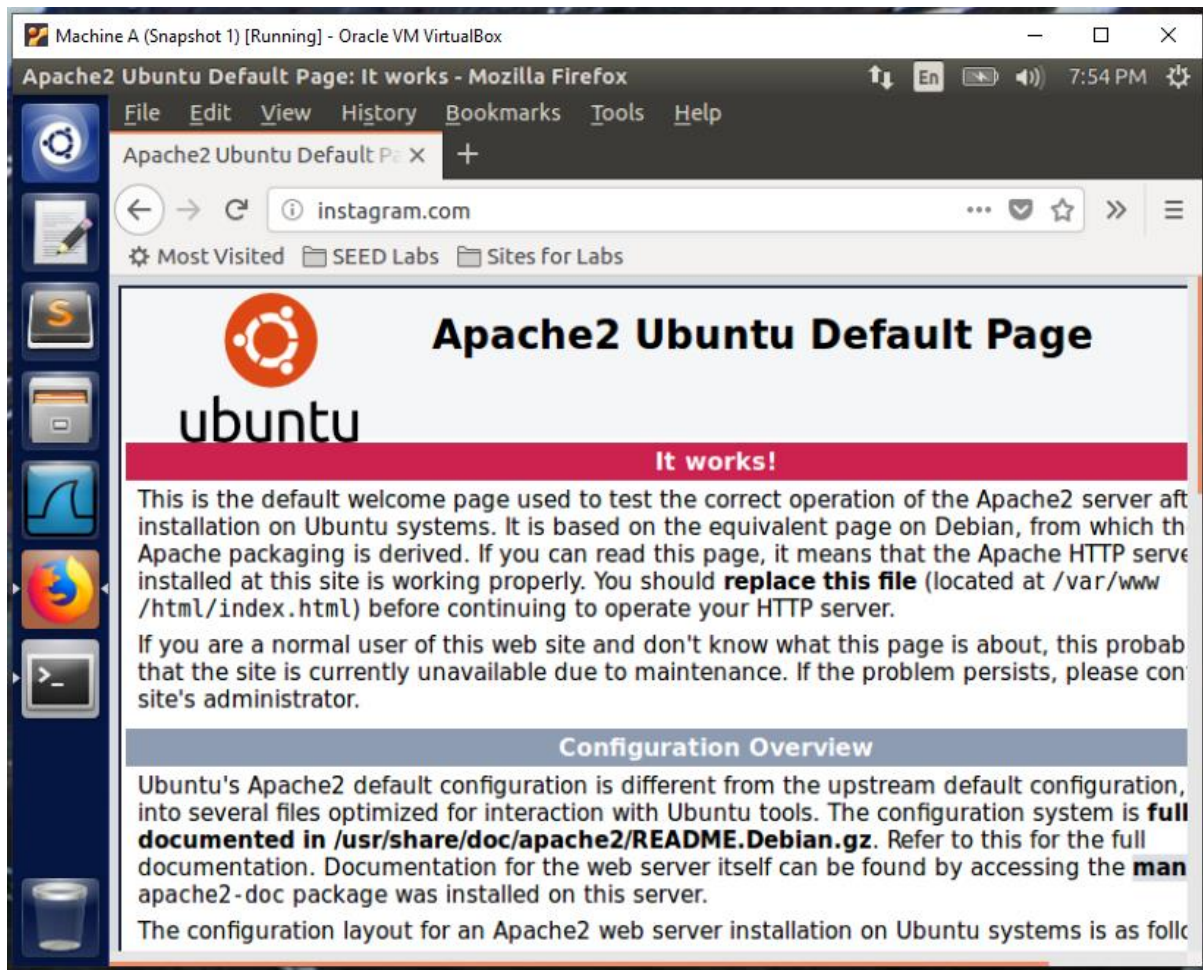
- Modified the host entry in /etc/hosts of victim's machine by emulating the result of DNS cache poisoning attack. Added Instagram.com entry as shown in snapshot:



```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
127.0.0.1      localhost
127.0.1.1      VM
# The following lines are desirable for IPv6 capable ho
sts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
127.0.0.1      User
127.0.0.1      Attacker
127.0.0.1      Server
127.0.0.1      www.SeedLabSQLInjection.com
127.0.0.1      www.xsslabelgg.com
127.0.0.1      www.csrflabelgg.com
127.0.0.1      www.csrfabattacker.com
127.0.0.1      www.repackagingattacklab.com
127.0.0.1      www.seedlabclickjacking.com
127.0.0.1      SEEDPKILab2018.com
127.0.0.1      instagram.com
"/etc/hosts" 20L, 571C                20,1                All
```

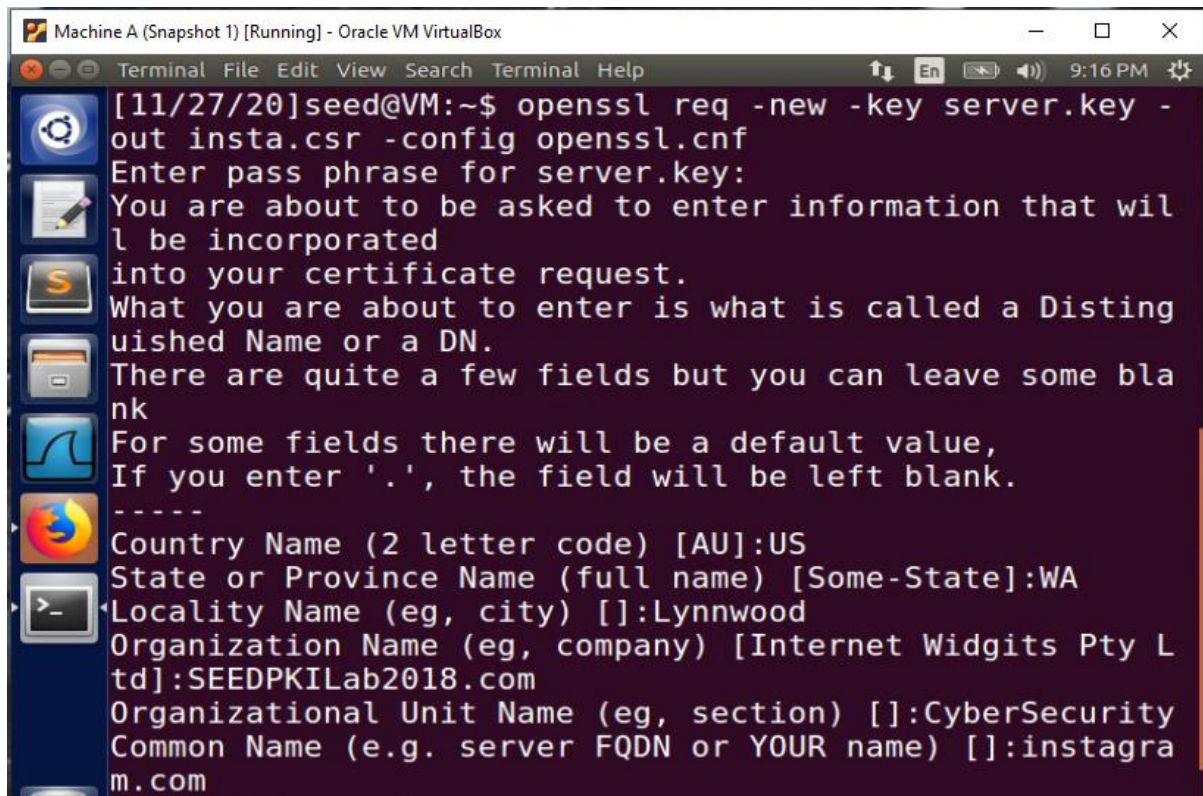
Step 3: Browse the target website

- After restarting the apache server using command:
`sudo service apache2 restart`
- When I browsed <https://instagram.com> on Firefox browser, observed the page of "apache2 ubuntu default page". Refer below snapshot:



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

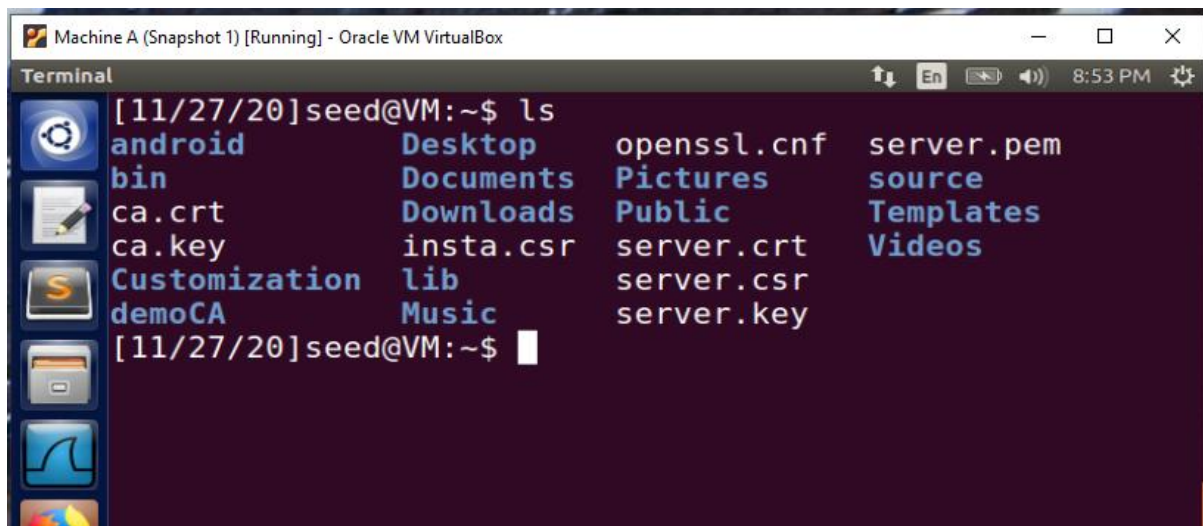
- Generating certificate signing request (csr) for instagram.com by using given command:
`openssl req -new -key server.key -out insta.csr -config openssl.cnf`



The screenshot shows a terminal window titled "Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
[11/27/20]seed@VM:~$ openssl req -new -key server.key -out insta.csr -config openssl.cnf
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:WA
Locality Name (eg, city) []:Lynnwood
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SEEDPKILab2018.com
Organizational Unit Name (eg, section) []:CyberSecurity
Common Name (e.g. server FQDN or YOUR name) []:instagram.com
```

- Certificate signing request (csr) has been generated → insta.csr



The screenshot shows the same terminal window with the command `ls` executed. The output lists the files and directories in the current directory:

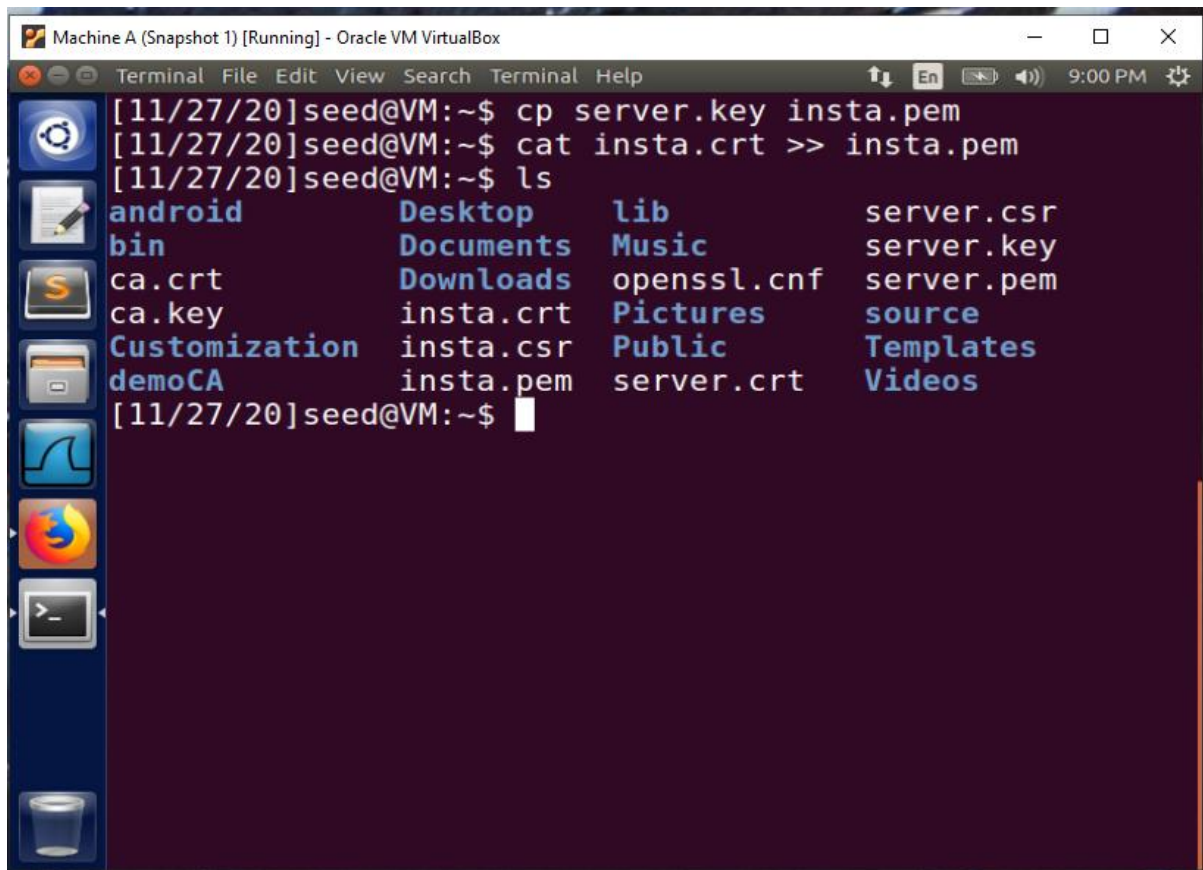
```
[11/27/20]seed@VM:~$ ls
android      Desktop      openssl.cnf  server.pem
bin          Documents   Pictures     source
ca.crt       Downloads   Public       Templates
ca.key       insta.csr   server.crt   Videos
Customization lib          server.csr
demoCA       Music       server.key
```

- Now generating the CA certificate for Instagram.com using following command:
`openssl ca -in instagram.csr -out instagram.crt -cert ca.crt -keyfile ca.key -config openssl.cnf`

```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
Subject:
  countryName           = US
  stateOrProvinceName   = WA
  organizationName       = SEEDPKILab2018.
com
  organizationalUnitName = CyberSecurity
  commonName             = instagram.com
X509v3 extensions:
X509v3 Basic Constraints:
  CA:FALSE
Netscape Comment:
  OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
  EC:98:DA:24:5D:E6:C6:74:44:62:3A:87:46:
7D:EC:82:A0:7E:47:51
X509v3 Authority Key Identifier:
  keyid:56:DF:AB:AD:46:E8:76:18:8C:46:ED:
BA:3F:2D:34:DA:3F:C1:0D:BB
Certificate is to be certified until Nov 28 02:16:52 20
21 GMT (365 days)
Sign the certificate? [y/n]:
```

```
Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminal
X509v3 extensions:
X509v3 Basic Constraints:
  CA:FALSE
Netscape Comment:
  OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
  EC:98:DA:24:5D:E6:C6:74:44:62:3A:87:46:
7D:EC:82:A0:7E:47:51
X509v3 Authority Key Identifier:
  keyid:56:DF:AB:AD:46:E8:76:18:8C:46:ED:
BA:3F:2D:34:DA:3F:C1:0D:BB
Certificate is to be certified until Nov 28 02:16:52 20
21 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]
y
Write out database with 1 new entries
Data Base Updated
[11/27/20]seed@VM:~$
```

- Configuring the web server by combining the secret key and certificate in one file for Instagram.com



Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox

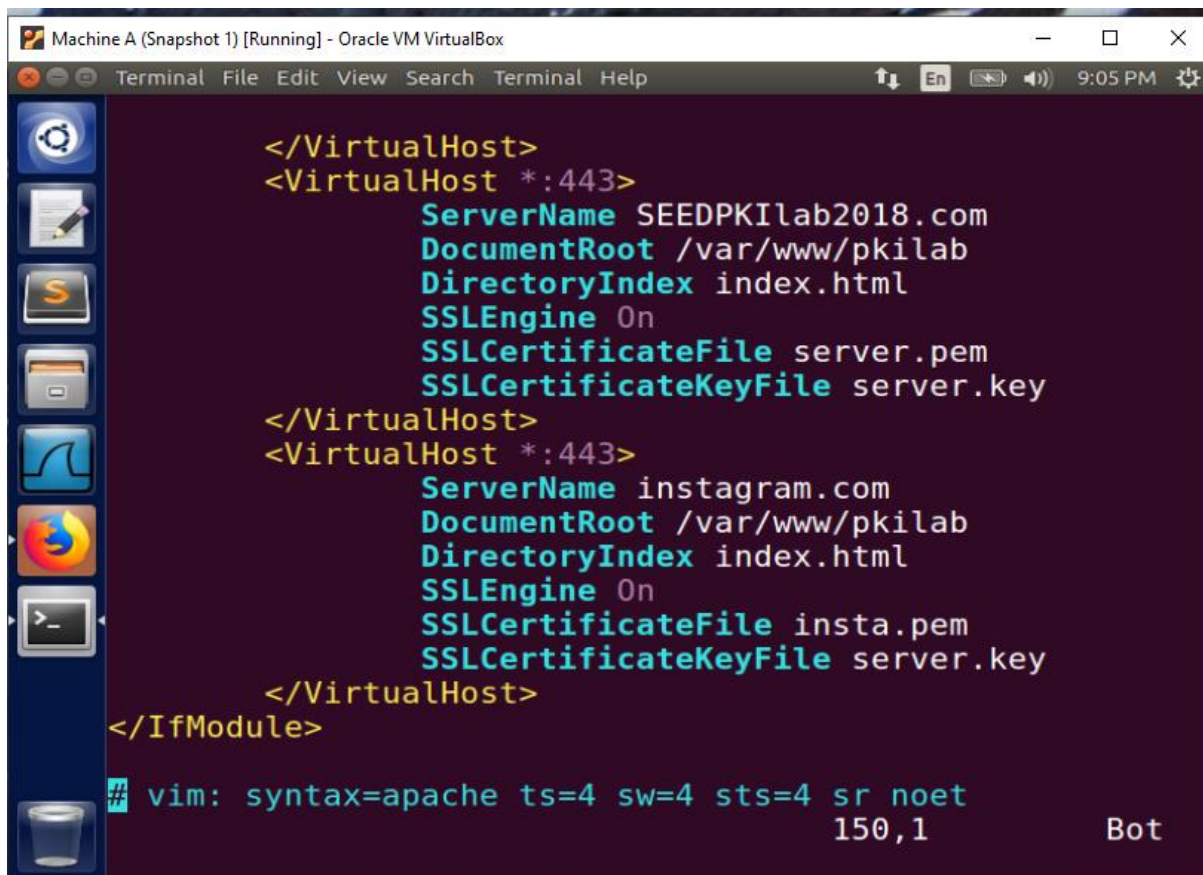
```
Terminal File Edit View Search Terminal Help 9:00 PM
```

[11/27/20]seed@VM:~\$ cp server.key insta.pem
[11/27/20]seed@VM:~\$ cat insta.crt >> insta.pem
[11/27/20]seed@VM:~\$ ls

android	Desktop	lib	server.csr
bin	Documents	Music	server.key
ca.crt	Downloads	openssl.cnf	server.pem
ca.key	insta.crt	Pictures	source
Customization	insta.csr	Public	Templates
demoCA	insta.pem	server.crt	Videos

[11/27/20]seed@VM:~\$

- Modified server certificate from server.pem to insta.pem

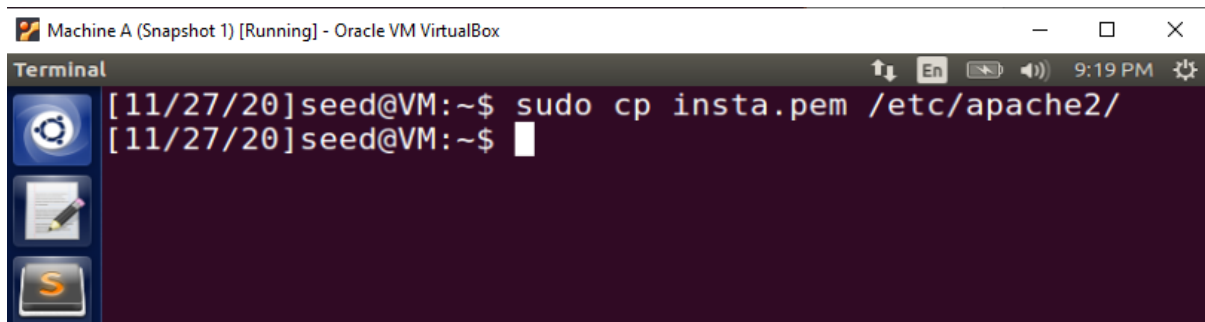


Machine A (Snapshot 1) [Running] - Oracle VM VirtualBox

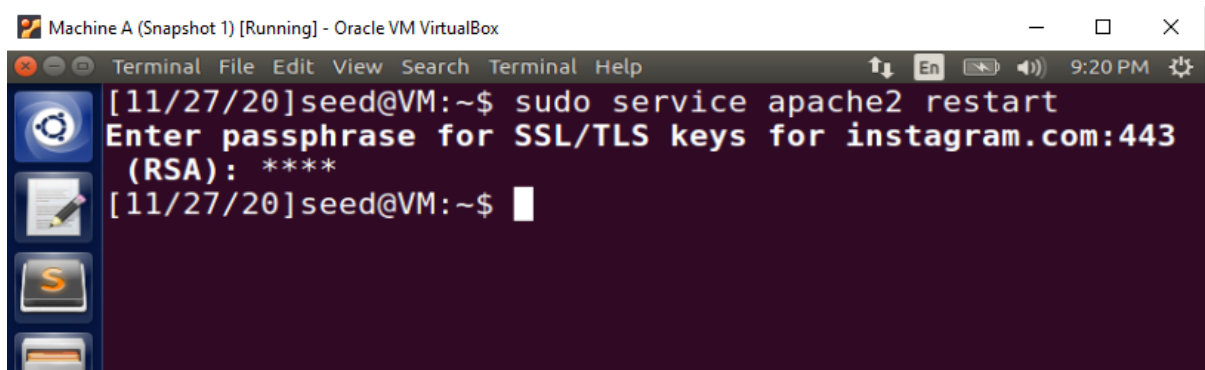
```
Terminal File Edit View Search Terminal Help 9:05 PM
```

```
</VirtualHost>  
<VirtualHost *:443>  
    ServerName SEEDPKIlab2018.com  
    DocumentRoot /var/www/pkilab  
    DirectoryIndex index.html  
    SSLEngine On  
    SSLCertificateFile server.pem  
    SSLCertificateKeyFile server.key  
</VirtualHost>  
<VirtualHost *:443>  
    ServerName instagram.com  
    DocumentRoot /var/www/pkilab  
    DirectoryIndex index.html  
    SSLEngine On  
    SSLCertificateFile insta.pem  
    SSLCertificateKeyFile server.key  
</VirtualHost>  
</IfModule>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
150,1 Bot
```

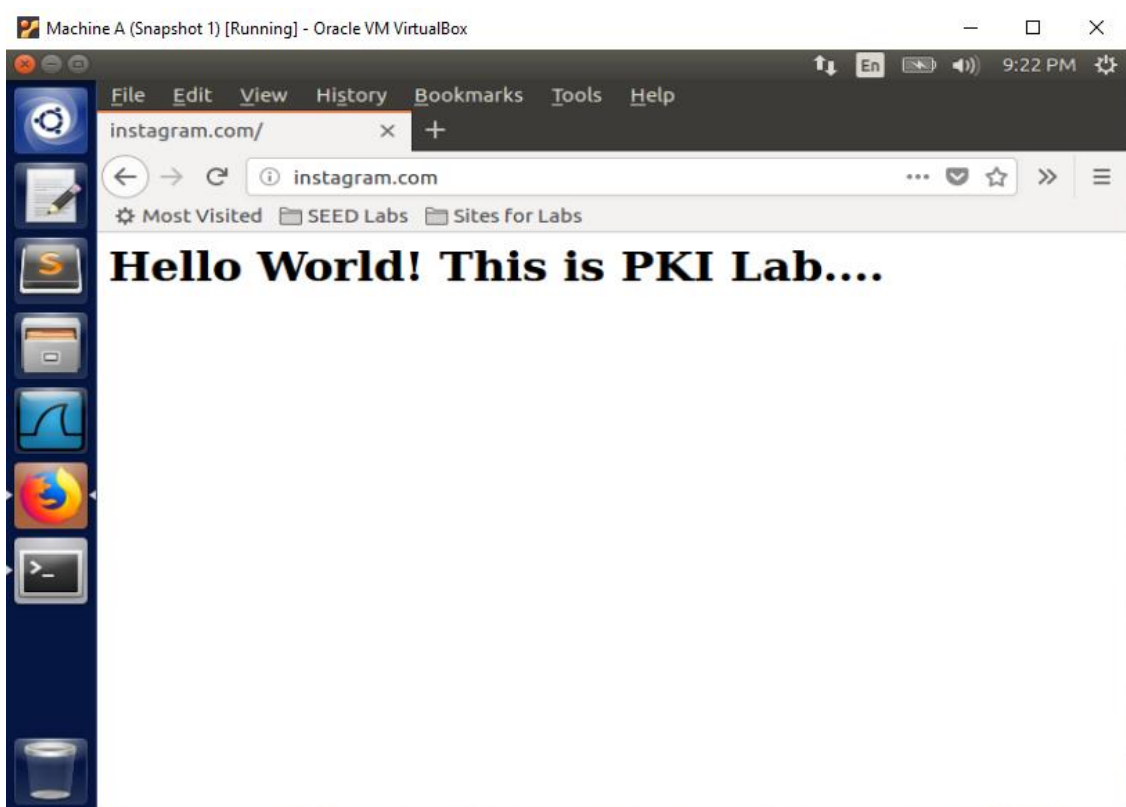
- Copying insta.pem file in /etc/apache2/ directory.



- Now restart the apache server using given command:
service apache2 restart



- After browsing <https://instagram.com> on Firefox browser, I observed that MITM attack was successful. Refer below snapshot of Firefox browser showing the MITM attack:



In overall lab, I found MITM attack was very interesting and surprising.