Network Security - Assignment 4

2.1 Becoming a Certificate Authority (CA)

```
[11/21/20]seed@VM:~/demoCA$ ls
certs  crl  index.txt  newcerts  serial
[11/21/20]seed@VM:~/demoCA$
```

```
[11/21/20]seed@VM:~$ ls
android         demoCA     Downloads         host    openssl.cnf  source
bin             Desktop    examples.desktop  lib     Pictures     Templates
Customization   Documents  get-pip.py        Music   Public       Videos
```

```
[11/21/20]seed@VM:~$ openssl req -new -x509 -keyout ca.key -out ca.crt -conf
penssl.cnf
Generating a 2048 bit RSA private key
.....................+++
.....................................+++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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]:TEXAS
Locality Name (eg, city) []:HOUSTON
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NYU
Organizational Unit Name (eg, section) []:NS
Common Name (e.g. server FQDN or YOUR name) []:SEED
Email Address []:.
```

```
[11/21/20]seed@VM:~$ ls
android  Customization  Downloads         lib          Public
bin      demoCA         examples.desktop  Music        source
ca.crt   Desktop        get-pip.py        openssl.cnf  Templates
ca.key   Documents      host              Pictures     Videos
```

2.2 Creating a Certificate for SEEDPKILab2018.com

```
[11/21/20]seed@VM:~$ #Gen pub/private key
[11/21/20]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:
Verifying - Enter pass phrase for server.key:
[11/21/20]seed@VM:~$ ls
android  Customization  Downloads         lib          Public      Videos
bin      demoCA         examples.desktop  Music        server.key
ca.crt   Desktop        get-pip.py        openssl.cnf  source
ca.key   Documents      host              Pictures     Templates
[11/21/20]seed@VM:~$ sudo openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
    00:cd:2a:7a:13:7a:bc:4e:e1:43:a8:1d:93:5b:01:
    56:2c:cd:0e:20:fa:c5:4d:09:ad:4e:5b:15:26:a9:
    49:4b:62:43:fc:c2:e7:2c:b3:ca:96:6f:7e:ad:84:
    85:70:48:8d:f8:8f:2b:f1:ea:56:73:ea:b1:71:a7:
    78:4e:1f:50:6a:4f:4a:1e:89:79:b2:1d:7e:ec:c0:
    82:60:36:19:2c:78:c3:a4:82:75:ba:2e:3d:65:21:
    16:a6:4e:9a:7d:c5:d3:7c:7c:34:20:0a:31:a0:72:
    7f:6c:c7:ec:0d
coefficient:
    00:9d:4c:ca:7a:aa:95:c7:46:c7:3f:81:69:a0:c7:
    fc:2c:57:9c:f2:8c:94:d2:d2:90:fc:3f:71:db:d5:
    a0:c9:11:a8:0e:16:58:71:5d:2d:8c:a2:2d:ec:9a:
    cc:66:13:ea:7e:52:65:91:55:55:d1:6c:ce:21:b1:
    99:92:87:e9:13
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDNKnoTerxO4UOoHZNbAVYszQ4g+sVNCa1OWxUmqUlLYkP8wucs
s8qWb36thIVwSI34jyvx6lZz6rFxp3hOH1BqT0oeiXmyHX7swIJgNhkseMOkgnW6
Lj1lIRamTpp9xdN8fDQgCjGgcu1PXz0YConTUaUPFgoyxWAy6VPUpJJK1wIDAQAB
AoGAEpE0iQzDe/QA9nbuvf6p+NX54FwxfN5SU9QEXwJVAsB63KYa273NqL2gXj2v
AdFfH859MKB/gHAN7PY1+gP1d/AEkDcxb1AG0gEF1uOzp1jCHiBzCrZQRibKsRbX
upGyO0EpAsKqri0iPI9jxJXF0f0oqU/vcrpvdGcVvA+rgSkCQQDq+BSL8RSxjGOA
XFLFZ6A/WlN2PQ07wUrQBQH1SJ7Qu0A1OJVPeN5H128UfOmYBJT6YExnBq1A/HZX
hWqWA2PjAkEA34eCCYxOHHp/lr2O/gdwqyA/HFBdd/0Y2KaaT9/9MkPrKDIlhqXn
KsP3RVdW6TOL0cfIfCKG8lFmOF3WuVR3fQJBANamty54mGzyUVvL+5l25z+45pIX
h+VU2WFbkcRgCSWqSifUfyJQys9akZFn44YKeYaZPnLH+LJrcDjYGtzcT/0CQQCn
Fqcotf2JojQPnSDvWOZqtM8YPiawU7wn7tm4zWl6EnUM9fdBIHCdciE7bx8jcQma
RaniyPhw2FH93n9sx+wNAkEAnUzKeqqVx0bHP4FpoMf8LFec8oyU0tKQ/D9x29Wg
yRGoDhZYcV0tjKIt7JrMZhPqflJlkVVV0WzOIbGZkofpEw==
-----END RSA PRIVATE KEY-----
[11/21/20]seed@VM:~$ clear
```

```
[11/21/20]seed@VM:~$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key \
> -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Nov 22 03:08:33 2020 GMT
            Not After : Nov 22 03:08:33 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = TEXAS
            organizationName          = NYU
            organizationalUnitName    = NS
            commonName                = SEEDPKILab2018.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                77:18:7A:63:D3:AB:60:12:65:2D:3D:A4:52:BE:75:E0:9B:DC:0C:DB
            X509v3 Authority Key Identifier:
                keyid:CE:80:C0:FF:01:94:D7:D8:7A:A7:98:78:45:8C:CC:51:D4:0C:D1:FE

Certificate is to be certified until Nov 22 03:08:33 2021 GMT (365 days)
Sign the certificate? [y/n]:y
```

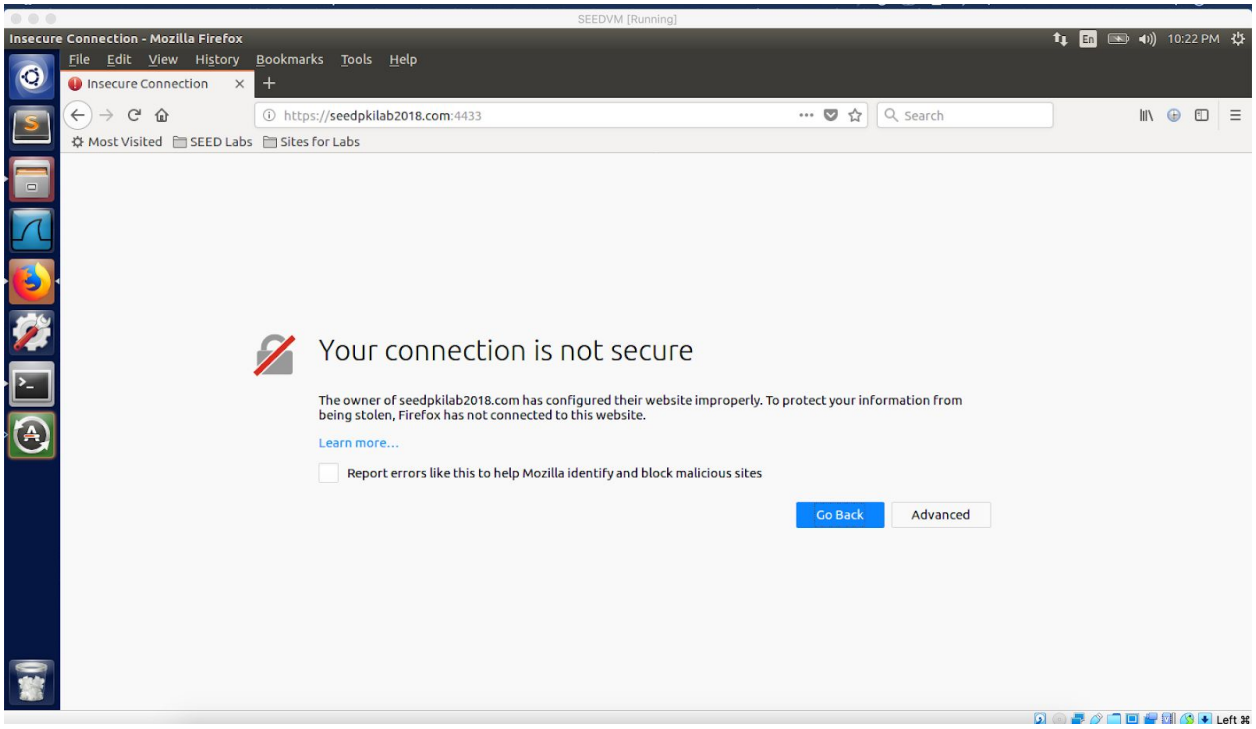
```
Certificate is to be certified until Nov 22 03:08:33 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[11/21/20]seed@VM:~$
```

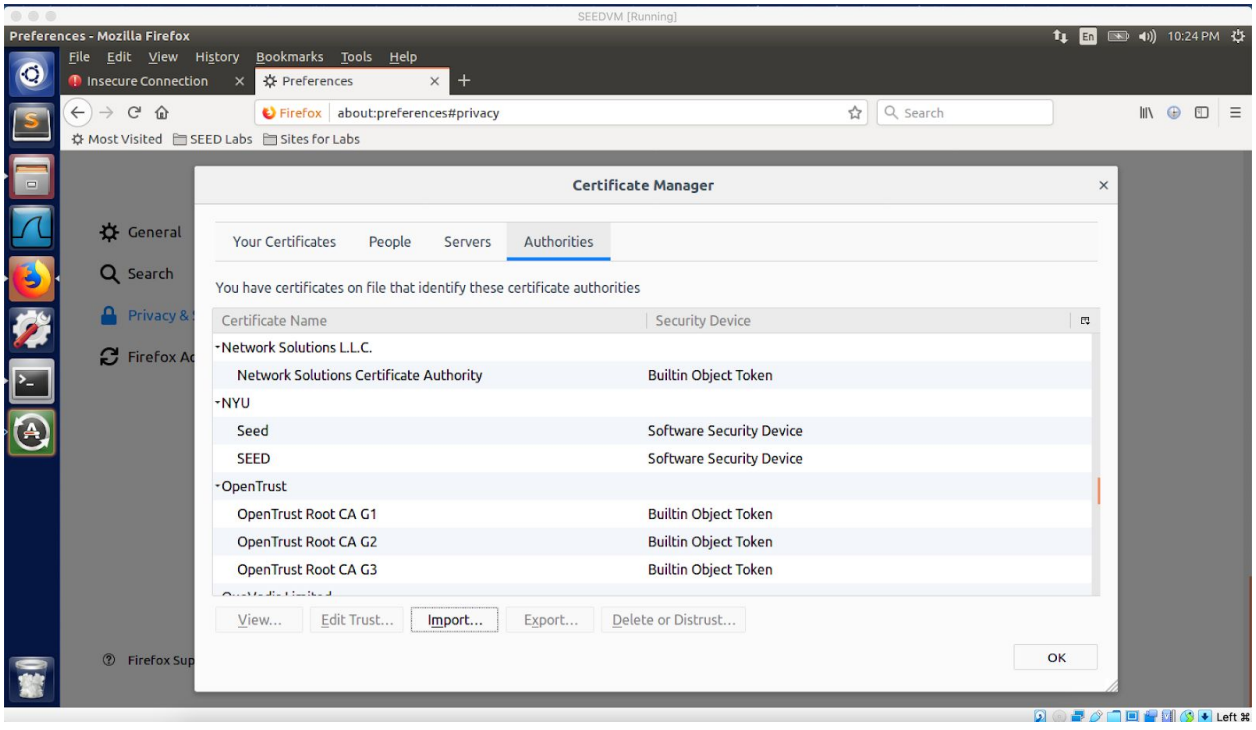
2.3 Deploying Certificate in an HTTPS Web Server

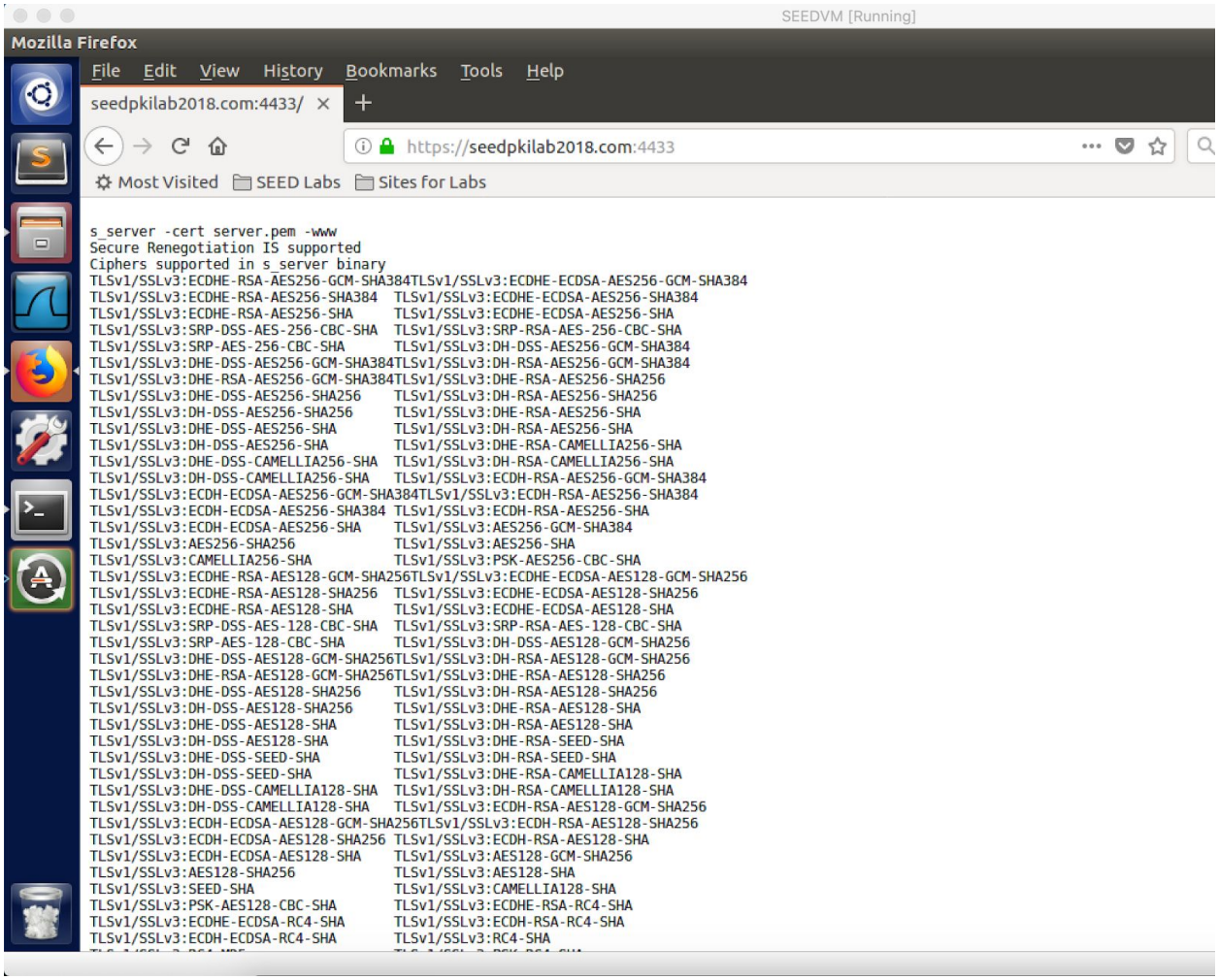
```
[11/21/20]seed@VM:~$ #deploying in https server
[11/21/20]seed@VM:~$ sudo vi /etc/hosts
[11/21/20]seed@VM:~$
```

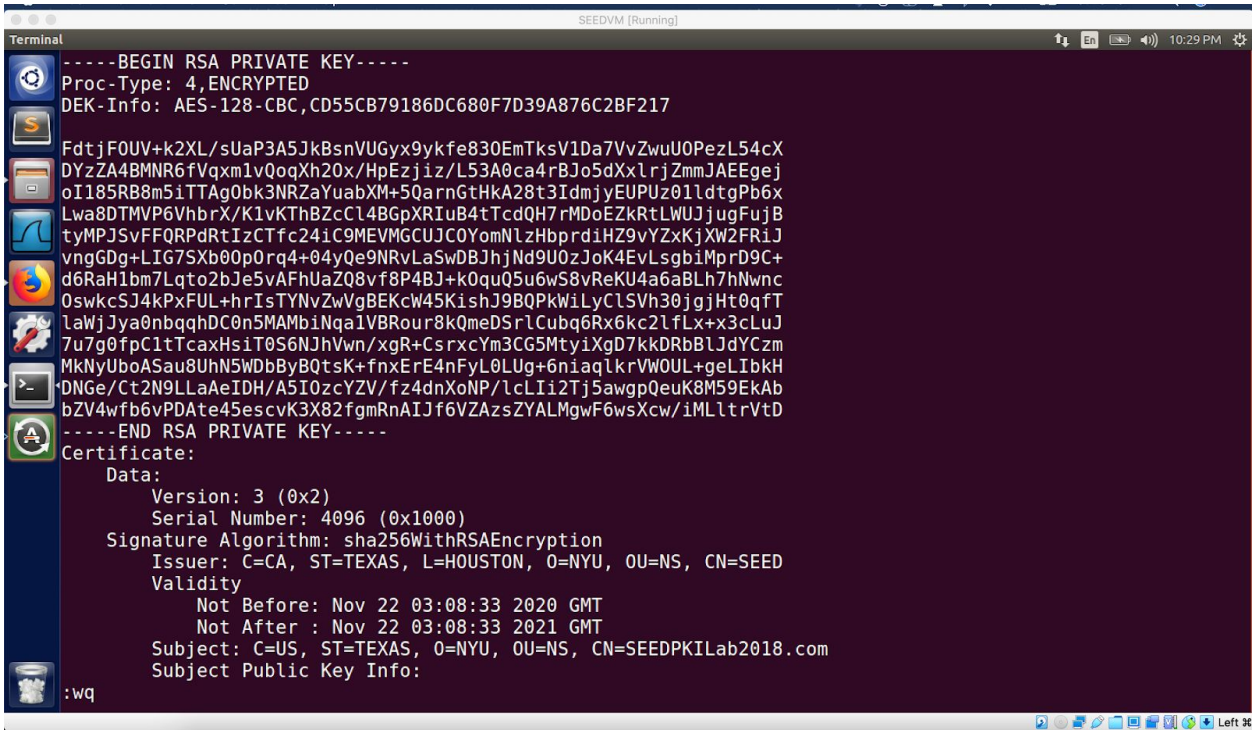
```
127.0.0.1       localhost
127.0.1.1       VM

# The following lines are desirable for IPv6 capable hosts
::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.csrflabattacker.com
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com
127.0.0.1       SEEDPKILab2018.com
~
```

```
[11/21/20]seed@VM:~$ #deploying in https server
[11/21/20]seed@VM:~$ sudo vi /etc/hosts
[11/21/20]seed@VM:~$ cp server.key server.pem
[11/21/20]seed@VM:~$ cat server.crt >> server.pem
[11/21/20]seed@VM:~$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```


SEEDVM [Running]
Insecure Connection - Mozilla Firefox
10:22 PM
File Edit View History Bookmarks Tools Help
Insecure Connection
https://seedpkilab2018.com:4433
Search
Most Visited SEED Labs Sites for Labs
Your connection is not secure
The owner of seedpkilab2018.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.
Learn more...
Report errors like this to help Mozilla identify and block malicious sites
Go Back
Advanced
Left ⌘

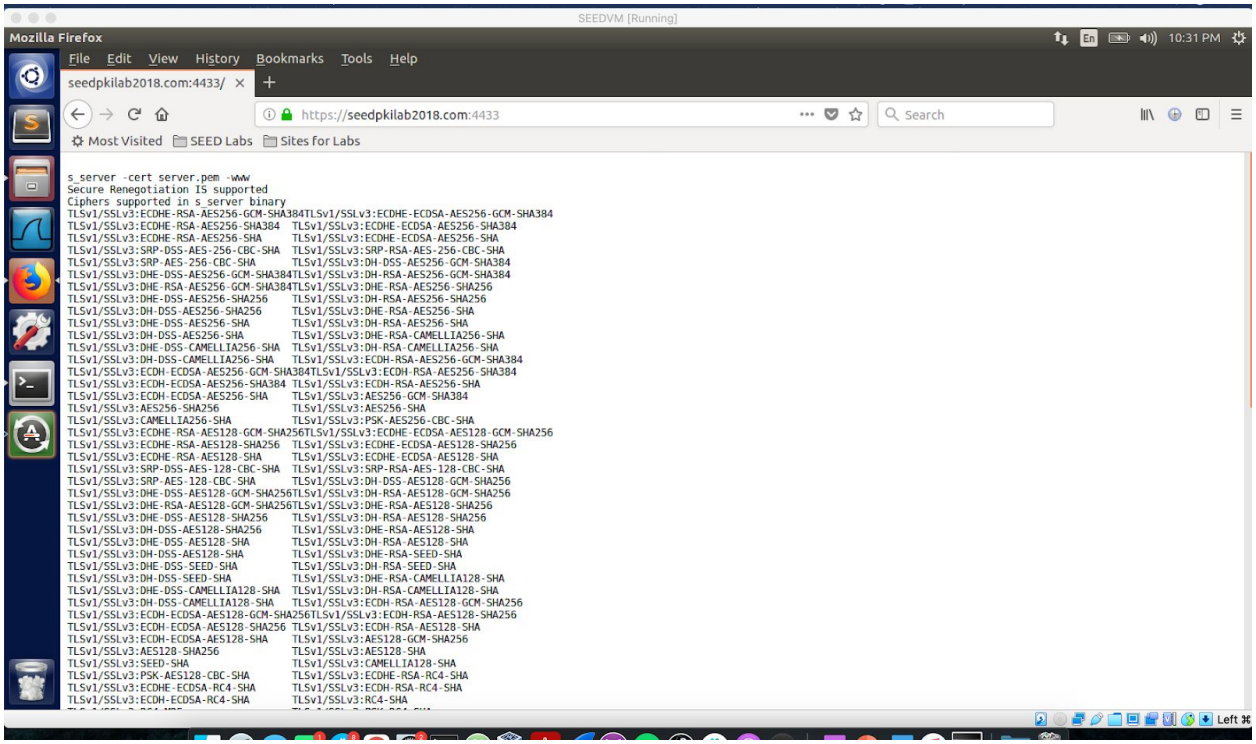

SEEDVM [Running]
Preferences - Mozilla Firefox
10:24 PM
File Edit View History Bookmarks Tools Help
Insecure Connection
Preferences
Firefox about:preferences#privacy
Search
Most Visited SEED Labs Sites for Labs
General
Search
Privacy &
Firefox Ac
Firefox Sup
Certificate Manager
Your Certificates People Servers Authorities
You have certificates on file that identify these certificate authorities
Certificate Name
Security Device
Network Solutions L.L.C.
Network Solutions Certificate Authority
Builtin Object Token
NYU
Seed
Software Security Device
SEED
Software Security Device
OpenTrust
OpenTrust Root CA G1
Builtin Object Token
OpenTrust Root CA G2
Builtin Object Token
OpenTrust Root CA G3
Builtin Object Token
View... Edit Trust... Import... Export... Delete or Distrust...
OK
Left ⌘


SEEDVM [Running]
Mozilla Firefox
File Edit View History Bookmarks Tools Help
seedpkilab2018.com:4433/
https://seedpkilab2018.com:4433
Most Visited SEED Labs Sites for Labs
s_server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384 TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA
TLSv1/SSLv3:SRP-DSS-AES-256-CBC-SHA TLSv1/SSLv3:SRP-RSA-AES-256-CBC-SHA
TLSv1/SSLv3:SRP-AES-256-CBC-SHA TLSv1/SSLv3:DH-DSS-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-DSS-AES256-GCM-SHA384TLSv1/SSLv3:DH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:DHE-RSA-AES256-SHA256
TLSv1/SSLv3:DHE-DSS-AES256-SHA256 TLSv1/SSLv3:DH-RSA-AES256-SHA256
TLSv1/SSLv3:DH-DSS-AES256-SHA256 TLSv1/SSLv3:DHE-RSA-AES256-SHA
TLSv1/SSLv3:DHE-DSS-AES256-SHA TLSv1/SSLv3:DH-RSA-AES256-SHA
TLSv1/SSLv3:DH-DSS-AES256-SHA TLSv1/SSLv3:DHE-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA256-SHA TLSv1/SSLv3:DH-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA256-SHA TLSv1/SSLv3:ECDH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDH-RSA-AES256-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA384 TLSv1/SSLv3:ECDH-RSA-AES256-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA TLSv1/SSLv3:AES256-GCM-SHA384
TLSv1/SSLv3:AES256-SHA256 TLSv1/SSLv3:AES256-SHA
TLSv1/SSLv3:CAMELLIA256-SHA TLSv1/SSLv3:PSK-AES256-CBC-SHA
TLSv1/SSLv3:ECDHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA256 TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA
TLSv1/SSLv3:SRP-DSS-AES-128-CBC-SHA TLSv1/SSLv3:SRP-RSA-AES-128-CBC-SHA
TLSv1/SSLv3:SRP-AES-128-CBC-SHA TLSv1/SSLv3:DH-DSS-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-GCM-SHA256TLSv1/SSLv3:DH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:DHE-RSA-AES128-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-SHA256 TLSv1/SSLv3:DH-RSA-AES128-SHA256
TLSv1/SSLv3:DH-DSS-AES128-SHA256 TLSv1/SSLv3:DHE-RSA-AES128-SHA
TLSv1/SSLv3:DHE-DSS-AES128-SHA TLSv1/SSLv3:DH-RSA-AES128-SHA
TLSv1/SSLv3:DH-DSS-AES128-SHA TLSv1/SSLv3:DHE-RSA-SEED-SHA
TLSv1/SSLv3:DHE-DSS-SEED-SHA TLSv1/SSLv3:DH-RSA-SEED-SHA
TLSv1/SSLv3:DH-DSS-SEED-SHA TLSv1/SSLv3:DHE-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA128-SHA TLSv1/SSLv3:DH-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA128-SHA TLSv1/SSLv3:ECDH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDH-RSA-AES128-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA256 TLSv1/SSLv3:ECDH-RSA-AES128-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA TLSv1/SSLv3:AES128-GCM-SHA256
TLSv1/SSLv3:AES128-SHA256 TLSv1/SSLv3:AES128-SHA
TLSv1/SSLv3:SEED-SHA TLSv1/SSLv3:CAMELLIA128-SHA
TLSv1/SSLv3:PSK-AES128-CBC-SHA TLSv1/SSLv3:ECDHE-RSA-RC4-SHA
TLSv1/SSLv3:ECDHE-ECDSA-RC4-SHA TLSv1/SSLv3:ECDH-RSA-RC4-SHA
TLSv1/SSLv3:ECDH-ECDSA-RC4-SHA TLSv1/SSLv3:RC4-SHA



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.

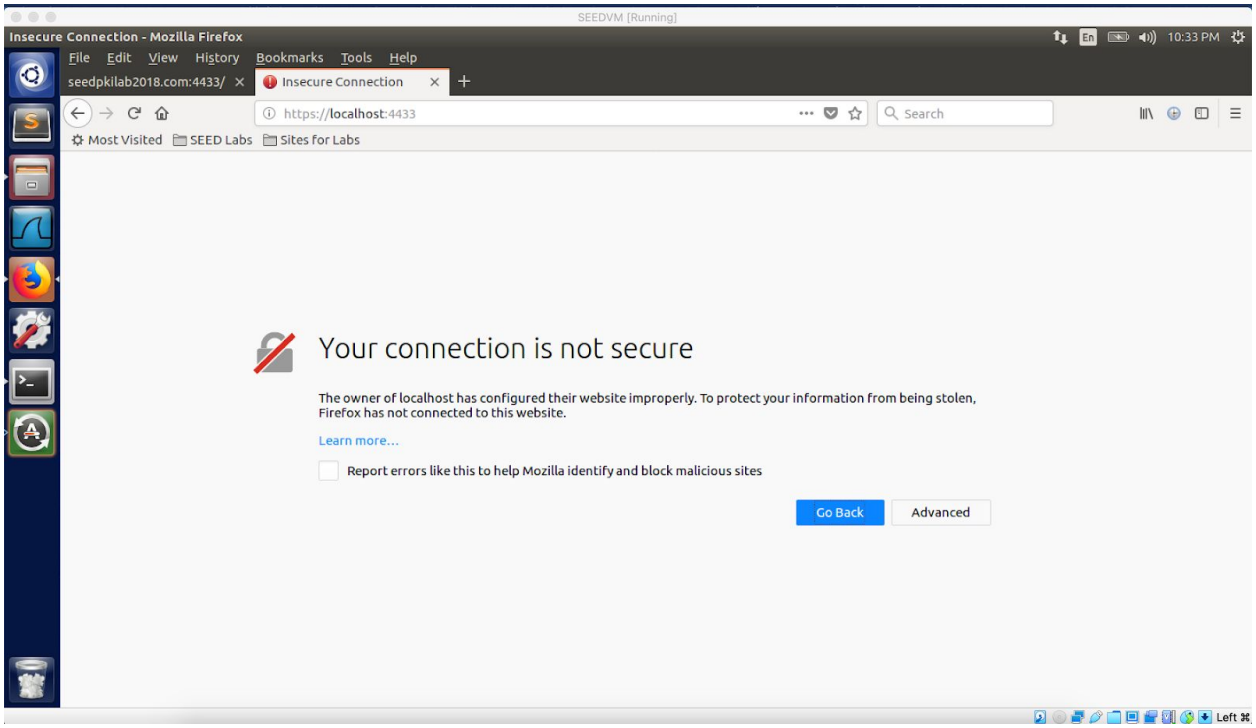
Ans: It works the same when I modify server.pem. Here I decided to change Country from US to CA.




```
[11/21/20]seed@VM:~$ #changing country to CA
[11/21/20]seed@VM:~$ sudo vi server.pem
[11/21/20]seed@VM:~$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
^C
[11/21/20]seed@VM:~$
```

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.

Ans: It doesn't work when I use localhost instead of SEEDPKILab2018.com:

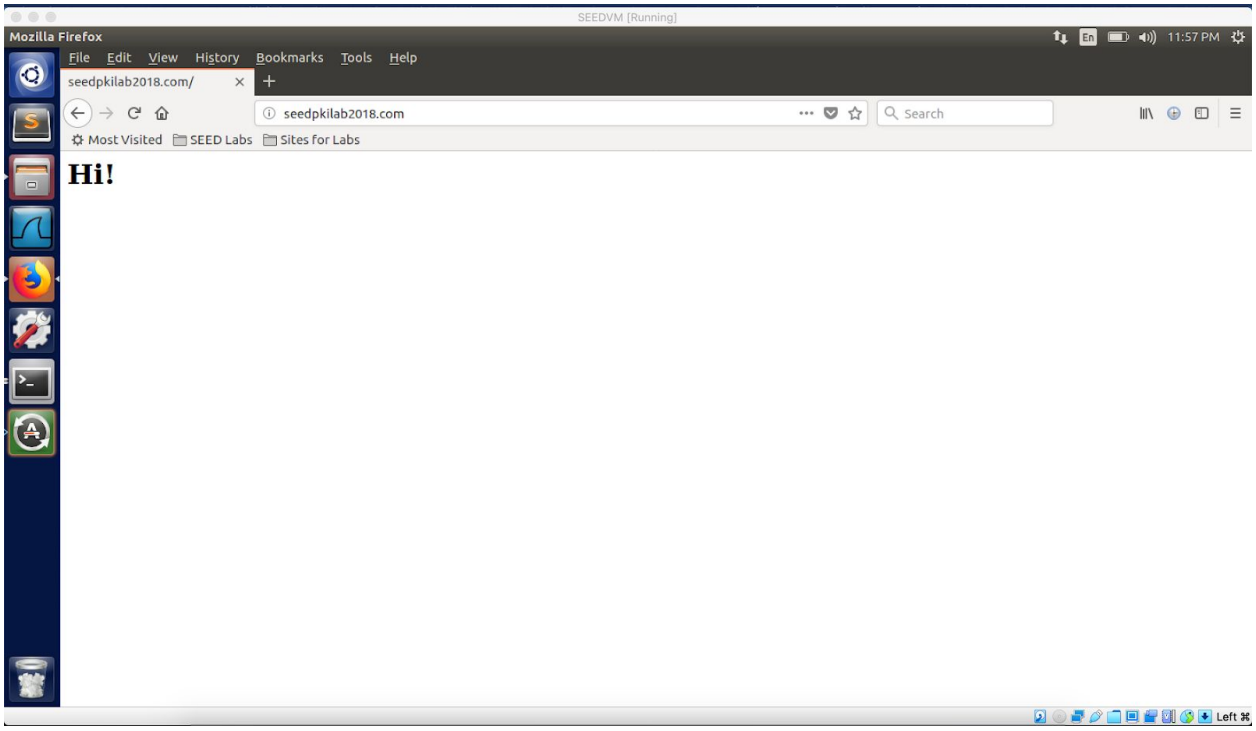


2.4 Deploying Certificate in an Apache-Based HTTPS Website

```
[11/21/20]seed@VM:~$ #Task 4: Deploying Certificate in an Apache-Based HTTPSWebsite
[11/21/20]seed@VM:~$ cp server.crt CERT.pem
[11/21/20]seed@VM:~$ cp sever.key KEY.pem
cp: cannot stat 'sever.key': No such file or directory
[11/21/20]seed@VM:~$ cp server.key KEY.pem
```

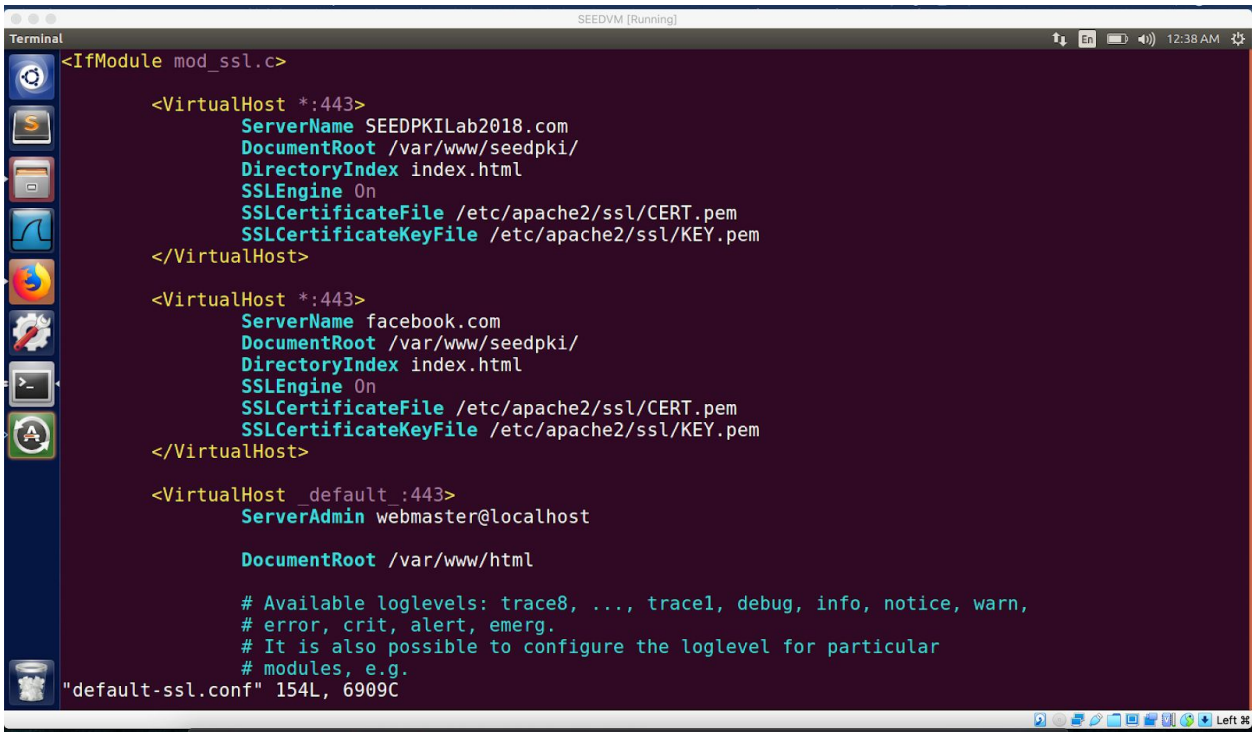
```
[11/21/20]seed@VM:.../sites-available$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1
. Set the 'ServerName' directive globally to suppress this message
Syntax OK
Trash
]seed@VM:.../sites-available$
```

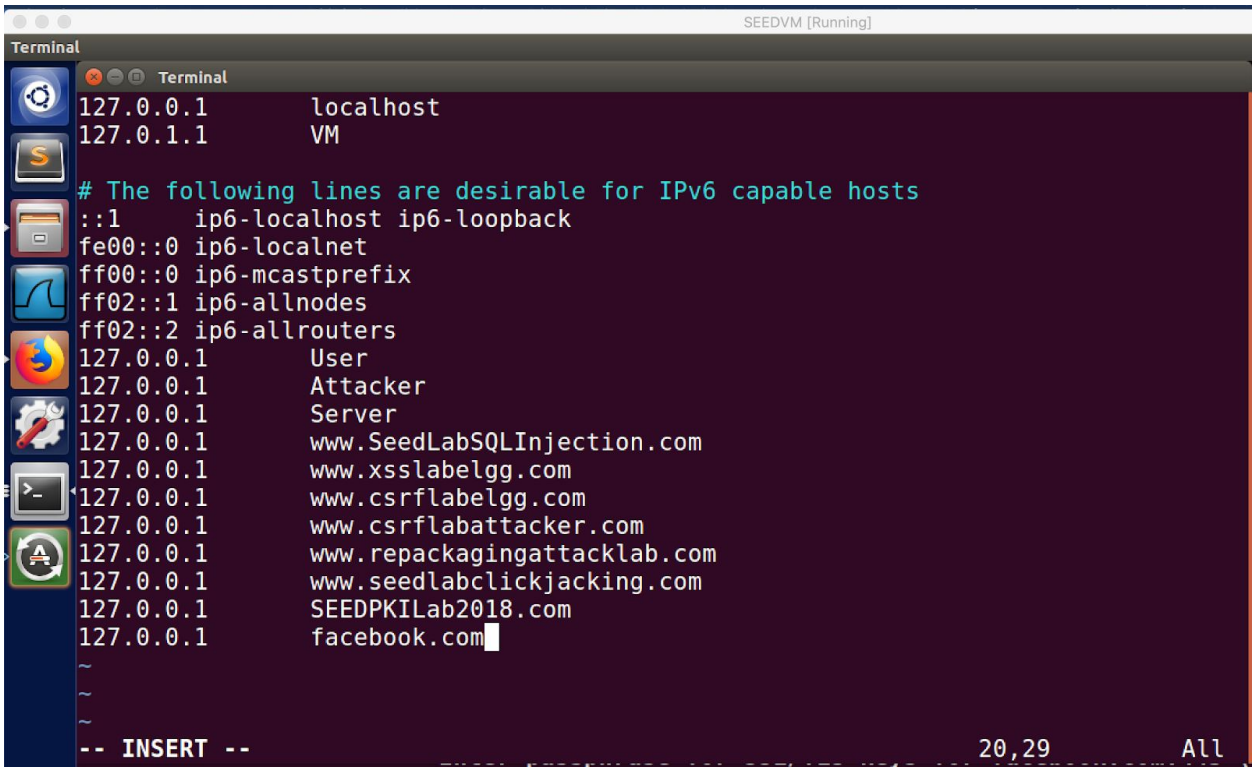
```
[11/21/20]seed@VM:.../www$ cd seedpki/
[11/21/20]seed@VM:.../seedpki$ ls
index.html
[11/21/20]seed@VM:.../seedpki$ vi index.html
[11/21/20]seed@VM:.../seedpki$ cd
[11/21/20]seed@VM:~$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified doma
in name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
[11/21/20]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
[11/21/20]seed@VM:~$ sudo a2ensite default-ssl
Site default-ssl already enabled
[11/21/20]seed@VM:~$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2018.com:443 (RSA): ****
[11/21/20]seed@VM:~$
```



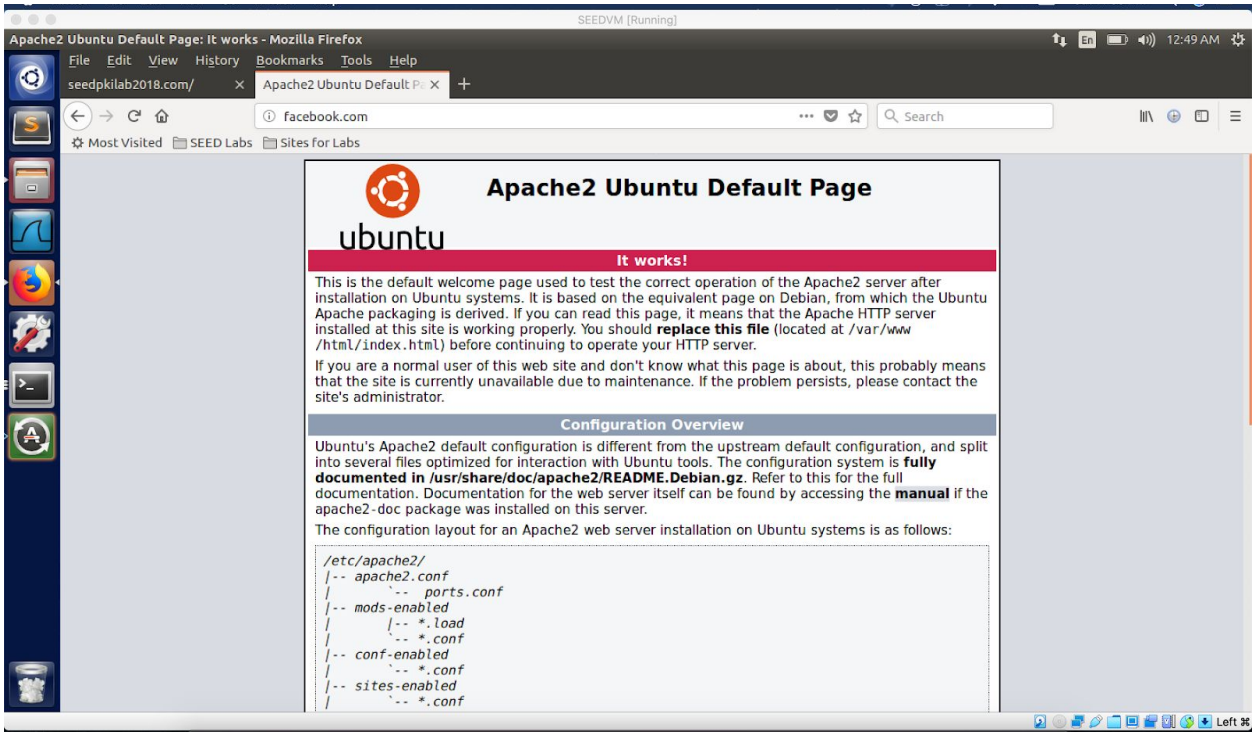
2.5 Launching a Man-In-The-Middle Attack

I choose facebook.com as the website for this attack.





The browser detects the common name mismatch so the html page doesn't show:



2.6 Launching a Man-In-The-Middle Attack with a Compromised CA

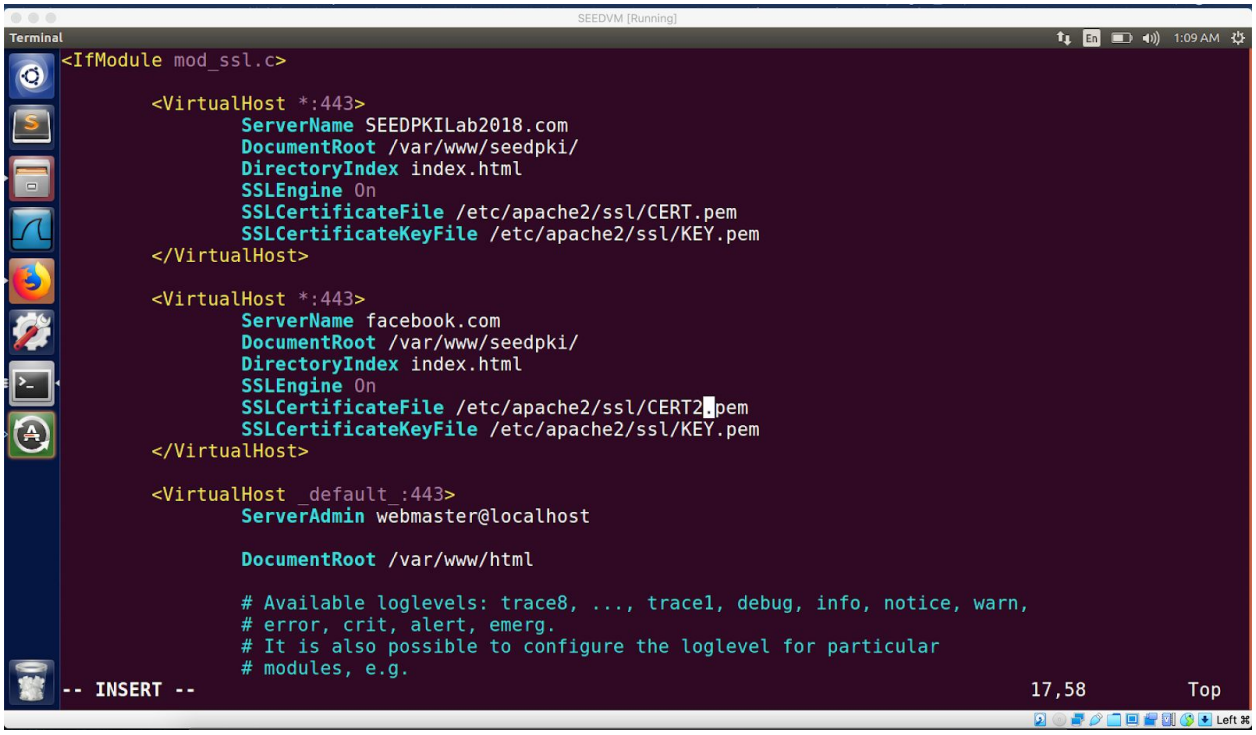
```
[11/22/20]seed@VM:~$ #Task 6
[11/22/20]seed@VM:~$ #Generating cert using ca pvt key
[11/22/20]seed@VM:~$ $ openssl req -new -key server.key -out facebook.csr -config openssl.cnf
$: command not found
[11/22/20]seed@VM:~$ openssl req -new -key server.key -out facebook.csr -config openssl.cnf
Enter 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]:TEXAS
Locality Name (eg, city) []:HOUSTON
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NYU
Organizational Unit Name (eg, section) []:SEED
Common Name (e.g. server FQDN or YOUR name) []:facebook.com
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pass
An optional company name []:pass
[11/22/20]seed@VM:~$
```

openssl ca -in facebook.csr -out facebook.crt -cert ca.crt -keyfile ca.key \
-config openssl.cnf

```
[11/22/20]seed@VM:~$ openssl ca -in facebook.csr -out facebook.crt -cert ca.crt -keyfile ca.key \
> -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Nov 22 06:01:32 2020 GMT
            Not After : Nov 22 06:01:32 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = TEXAS
            organizationName          = NYU
            organizationalUnitName    = SEED
            commonName                = facebook.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                77:18:7A:63:D3:AB:60:12:65:2D:3D:A4:52:BE:75:E0:9B:DC:0C:DB
            X509v3 Authority Key Identifier:
                keyid:CE:80:C0:FF:01:94:D7:D8:7A:A7:98:78:45:8C:CC:51:D4:0C:D1:FE

Certificate is to be certified until Nov 22 06:01:32 2021 GMT (365 days)
Sign the certificate? [y/n]:y
```

```
Certificate is to be certified until Nov 22 06:01:32 2021 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
[11/22/20]seed@VM:~$
```

cp server.key facebook.pem
cat facebook.crt >> facebook.pem



The attack is successful this time!