Task 1 : Becoming a Certificate Authority (CA)

Step 1:

Copied the configuration file with command -

cp /usr/lib/ssl/openssl.cnf ./openssl.cnf

```
[10/06/22]seed@VM:~/.../task1$ cp /usr/lib/ssl/openssl.cnf ./openss
l.cnf
[10/06/22]seed@VM:~/.../task1$ ls
```

Step 2:

Created a directory named demoCA directory in the parent directory to store multiple files.

```
[10/06/22]seed@VM:~/.../task1$ mkdir ./demoCA
[10/06/22]seed@VM:~/.../task1$ ls
demoCA  openssl.cnf
[10/06/22]seed@VM:~/.../task1$
```

Step 3:

Creating multiple files such as certs, crl, etc,.

mkdir certs # Where the issued certs are kept

mkdir crl # Where the issued crl are kept

mkdir newcerts # Default place for new certs

touch index.txt # Database index file

echo “1000” > serial # The current serial number

```
[10/07/22]seed@VM:~/.../demoCA$ mkdir certs
[10/07/22]seed@VM:~/.../demoCA$ mkdir crl
[10/07/22]seed@VM:~/.../demoCA$ mkdir newcerts
[10/07/22]seed@VM:~/.../demoCA$ touch index.txt
[10/07/22]seed@VM:~/.../demoCA$ echo "1000" > serial
[10/07/22]seed@VM:~/.../demoCA$ ls
certs  crl  index.txt  newcerts  serial
[10/07/22]seed@VM:~/.../demoCA$
```

Step 4 -

Generating self-signed certificate with command –

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

```
[10/07/22]seed@VM:~/.../task1$ openssl req -new -x509 -keyout ca.ke
y -out ca.crt -config openssl.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 incorpo
rated
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]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Ritika
Email Address []:
[10/07/22]seed@VM:~/.../task1$
```

The output generated by the command is in two files as ca.key and ca.crt.
ca.key containing CA's private key and generated file is seen as:

```
[10/07/22]seed@VM:~/.../task1$ cat ca.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIHzu1uL4ZMKkCAggA
MBQGCCqGSIb3DQMHBAjiu+JBESmWlwSCBMhqaz6EFlhBMXGZe+lthAkF9EFHAKwg
AR34q293wnH054ARUsmFMgdOqU+Xi4PH1Btv4rJsoAGOqXM0ONsrtQAadAfQqbbk
IocptaY6bV0hePaGCv3WRoVq/klScWHtppxeyTlnNt+jLn4lOXkb7yDNZ7oxQ+aN
jCtPM6wiJrPJDS2oOrJElNq1BXkienNFmvcOIK5XN8LTNn0YomDCpo9KuSstDSyC
DOGGMUuYtbsoIOiBSTxaI3vG7o8sv8xatc4LOSxXhKz5OYMgXlMHcxX5t/1+EuTr
e+LkogNrkopHamxMFfTRIeEEAj6qaNo3nJOVtbn196UUQkaoNFDNZtlByM9u0fy8
yYqlFNVqbqiNCQNUM4kszaEwurcPRWZCKE7RYdEC9nN0agv09vcOgHwPI89pJdlg
9WGI9LmtpUhqyN6qF8D+y6PbzCk/v1hDJhuDqgBTIh1MVdkx6pga8ctKde/BzREi
mF8X+bLSRdWMNd3WZkrbuUJhWAVru5AIdV1V4BQKE0feSeneNNP6lPJtqd7IuSJu
8MkTvHsjjJ4Xvcj1f2PUWYwFRryt/pRWVxUfYdsUzEYrKRupvaq4Yt/4TToMpYXU
svXZYw++YDVEHQ2FqDfNxI/V08uGb1voF9PyNbM2qlEky7Z9JiMuqaPZFEmGR2kS
Ju9UjOlteP6vjCdwf4W4oVIbMp+d0gJl0myRPN3PGtv+VkI2Hele3OHcPiCx9Eil
yZS9Hti7Z1G9GbNjLlJDu+n2/6Buy0sD8wRg2VdAAZIUO2mbUyjkFCU1yRiC61E9
QdISsRdeagp2KNRl3KnJJOe3YGIT9q/MlOzwL0aUDtWSNi1g2zoC/u2xMmnb/P4+
zTdkHj3Ics0QpdXjgw1r2n47xcuyKIaxUk2G9kkJ2YK0UfxPrufo4UJCzxxJORse
Lz0LHIAS3P2WYz8+AuJD9KX6JMoXF4+32pH82wTyPB1bENohHvo0r0eiwzUZ8ryV
JXGHqUoMSgT3jnQl0EqwOwadZ94oYHgor1wJaRc7soRdF7i3YZ97xmfVYbeaDoyf
OgDDkEf/ZMjXb8tsW2iKaVP/1C3qqMJMr1HGQnGdfi2GfPg8wn3XpM3EnsXZl9Gy
08WtWXruhIBuQrA76gGTTabYcum+bLEUZcZborSQUXDa4ItLRUix6oBIg+t90JsK
TlsByRgLO91/BtCHkwFbJ1EtErr6U4EM//aE8q3/Oi7fAiBIOJ6vi2SFoHx3SAMR
tm6QoQKo8pWAM3KN8X4b5NJYnLC/NQ1BvxLYzmaRmvcnge24Gfa2ONa0fZpGXivH
N4eD8497h2ZjTwhMlj8fvo5F1jwrLUtMYsYQ4lh6AmgVV4eIkfBh+eARGINDUKkK
wtQae3VFhP7dAQtB452sW7p1bWscoZh1oa35xK8deaCYiz8jTUdvo9DWbk4I109E
PzbUEot0azxUX4B56MGWFubsjokH5sV+aj/f04J8xy6xKsWwMt7HTtq/4XbtRDA8
8u6psZUy15ULoF5tnB4xLtKOLY4ShwD/o/xtRgHZSPxCm5U2jCWThtJab4UdD5Vi
irqi3UD+nqWAXj9dWDMJSvx2bNodAD49vCcCrexX0cmMZdVp0SWpKSboptdZ3V8S
JGM=
-----END ENCRYPTED PRIVATE KEY-----
[10/07/22]seed@VM:~/.../task1$
```

ca.crt containing the public-key certificate and seen as below:

```
[10/07/22]seed@VM:~/.../task1$ cat ca.crt
-----BEGIN CERTIFICATE-----
MIIDfzCCAmegAwIBAgIJAN7PAzosTgIFMA0GCSqGSIb3DQEBCwUAMFYxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQxDzANBgNVBAMMBlJpdGlrYTAeFw0yMjEwMDcwNDMxMTBa
Fw0yMjExMDYwNDMxMTBaMFYxCzAJBgNVBAYTAkFVMRMwEQYDVQQIDApTb21lLVN0
YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBXaWRnaXRzIFB0eSBMdGQxDzANBgNVBAMM
BlJpdGlrYTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMztDeZGtNeW
JokQVtjnht+DQQpsNZEdpA/iGzl4RLdh4SZ2mvrDEMzw+jDoQHdrr1o82RZ4A5BH
qB1/cW/3/YSFHbFfGUh1ypg8dAtF8+LQVwKcq7712lDGr34kcq3jt5TppBChYohC
uCb20gTDwOX8ClpMkayeaaXWq7wuRZNUaUsnnxjqN6JYrj39wl6EfI0TiR2hUvMZ
WO2bR9Dv1lUb+TCnCqb8n24WhtSX/4Qr6otPNIyLXX11mcHv2ViAQpbdNVVYERj7
uAc9+1DkC0imzLDjRPsJSd9hBgt3Jo+ajUnLUb7/gPRrvL7BObdUaFAUVJ1J9nuB
Oiq0xaUIywMCAwEAAaNQME4wHQYDVR0OBBYEFAIgK+Y8ie6/fYhig9wTAW4i38dd
MB8GA1UdIwQYMBaAFAIgK+Y8ie6/fYhig9wTAW4i38ddMAwGA1UdEwQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAHVF9907Qh4w9q13Xk+rZ9EYseji9YGESld3qKx3
UKp8/MF5A3MRqVe1rg+zvgkJdsf7hAMUGUj9PNNW0qH8l17XCffO+wezjTW7/YkE
jLL29DlqIGyzdfhtSaaY6Sl8KKaTjhflEmZ4iqKI7qnDvBfiwopAmvYDW+BMlJKN
5d1UDUacb4HfnKdFs/Fq5biEyixpTe0gUNNgZkQMl/B48Wzm9IDmSEA88c65qOU/
SVGoM8hlnRFgJPZo1x6ZXni/svMJqVOiJH54anv2Sm+whVjGhTWujQN5ABfzkuJa
R3ZyC9gHT4VGP3yiJkrEZzLRSFhy+mvG2PXmJBfaiKMkpJQ=
-----END CERTIFICATE-----
[10/07/22]seed@VM:~/.../task1$
```

Task 2: Creating a certificate for SEEDPKILab2020.com

Step 1: Generate public/private key pair

Generating an RSA key pair with command -
openssl genrsa-aes128-out server.key 1024

```
[10/07/22]seed@VM:~/.../task1$ openssl genrsa -aes128 -out server.k
ey 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:
```

To see the actual content of the encoded text file we use command -

Openssl rsa -in server.key -text

```
[10/07/22]seed@VM:~/.../task1$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
    00:d7:f7:71:bf:fe:7b:0e:cb:cc:bd:9c:cf:7b:b8:
    f7:da:53:77:5a:50:a0:47:8f:2f:ed:bb:e2:d9:11:
    0a:1a:8a:f1:6f:31:f1:8d:cc:4f:83:07:c9:06:69:
    95:56:c0:64:bc:22:23:b9:0a:bc:41:ef:49:66:30:
    b6:cc:2c:92:7b:a1:c3:5a:e3:80:89:df:d7:f7:1f:
    6c:2c:a8:2f:74:89:c0:0d:87:d5:35:43:b2:40:c0:
    4f:f0:bd:cb:d9:fb:97:1e:43:46:c0:73:70:46:82:
    49:b5:17:2d:21:8c:62:53:13:2f:58:b0:2b:20:0c:
    17:b7:bb:2b:ea:88:97:91:11
publicExponent: 65537 (0x10001)
```

```
coefficient:
    00:80:0c:64:24:a9:7d:1b:c8:b1:4d:0a:d9:df:33:
    f8:40:3f:c5:92:44:c2:ef:b6:af:ca:99:9f:47:78:
    97:c1:d1:15:ba:2e:c5:7f:8e:e0:94:4d:12:61:5f:
    48:e2:25:1a:97:20:4e:71:3f:4e:c7:25:59:70:db:
    9f:f0:b8:14:9c
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDX93G//nsOy8y9nM97uPfaU3daUKBHjy/tu+LZEQoaivFvMfGN
zE+DB8kGaZVWwGS8IiO5CrxB70lmMLbMLJJ7ocNa44CJ39f3H2wsqC90icANh9U1
Q7JAwE/wvcvZ+5ceQ0bAc3BGgkm1Fy0hjGJTEy9YsCsgDBe3uyvqiJeREQIDAQAB
AoGBAJ0coX74JhPshWUHuBWcyYrmH7TvZLrKWybLAcvCTqRioiOvpLfwg4OiXUcx
9nem/WG0QGK0gaJB7xbNp54/asv5KUoSN/lp+dyghFHqLKQXCMkAIgXatRBAiWMX
JM0AjQ+/Lv9/tZZslSCmUFcfUrU63kCfb0dtP5jvkEL0p+yxAkEA/P2Qj6VXwwr2
whOAU/w51dJbr9Fuf4muiueRJ/UOmVhx9jukxjX99Y1KK6U434FFZkOORsvaqlFz
41UGWdwmnQJBANqJIVem9qw73I27HVe/EJukoHf580vyhFYI1iPnqh+X4pcd1UjY
VTiEso083kbGNr+wq5OSQT0+8ViF4SIVEAUCQQCJnGNFv2o9bXlLl8/13vyWkhXN
TN4OAZYveEY8tc97GeDjTw4RwsyhBHmTxAtrh6V4OsbpO3l1q1ieD2Gk/1sBAkA1
VgCgPuy3Tojewos/zi8EfaJ4hbV+lLr9sHxrGPBpc41m6m4SsFHkUGavl+p8BYZ5
NGo7wxdPFlybCVnQEqntAkEAgAxkJKl9G8ixTQrZ3zP4QD/FkkTC77avypmfR3iX
wdEVui7Ff47glE0SYV9I4iUalyBOcT9OxyVZcNuf8LgUnA==
-----END RSA PRIVATE KEY-----
[10/07/22]seed@VM:~/.../task1$
```

Step 2: Generate a Certificate Signing Request (CSR)

Generating a certificate Signing Request(CSR) containing the company's public key. This CSR is sent to the CA who generates a certificate for the key (matches server's true identity). Used SEEDPKILab2020.com as a common name. Command used is -

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


```
coefficient:
    00:80:0c:64:24:a9:7d:1b:c8:b1:4d:0a:d9:df:33:
    f8:40:3f:c5:92:44:c2:ef:b6:af:ca:99:9f:47:78:
    97:c1:d1:15:ba:2e:c5:7f:8e:e0:94:4d:12:61:5f:
    48:e2:25:1a:97:20:4e:71:3f:4e:c7:25:59:70:db:
    9f:f0:b8:14:9c
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDX93G//nsOy8y9nM97uPfaU3daUKBHjy/tu+LZEQoaivFvMfGN
zE+DB8kGaZVWwGS8IiO5CrxB70lmMLbMLJJ7ocNa44CJ39f3H2wsqC90icANh9U1
Q7JAwE/wvcvZ+5ceQ0bAc3BGgkm1Fy0hjGJTEy9YsCsgDBe3uyvqiJeREQIDAQAB
AoGBAJ0coX74JhPshWUHuBWcyYrmH7TvZLrKWybLAcvCTqRioiOvpLfwg4OiXUcx
9nem/WG0QGK0gaJB7xbNp54/asv5KUoSN/lp+dyghFHqLKQXCMkAIgXatRBAiWMX
JM0AjQ+/Lv9/tZZslSCmUFcfUrU63kCfb0dtP5jvkEL0p+yxAkEA/P2Qj6VXwwr2
whOAU/w51dJbr9Fuf4muiueRJ/UOmVhx9jukxjX99Y1KK6U434FFZkOORsvaqlFz
41UGWdwmnQJBANqJIVem9qw73I27HVe/EJukoHf580vyhFYI1iPnqh+X4pcd1UjY
VTiEso083kbGNr+wq5OSQT0+8ViF4SIVEAUCQQCJnGNFv2o9bXlLl8/13vyWkhXN
TN4OAZYveEY8tc97GeDjTw4RwsyhBHmTxAtrh6V4OsbpO3l1q1ieD2Gk/1sBAkA1
VgCgPuy3Tojewos/zi8EfaJ4hbV+lLr9sHxrGPBpc41m6m4SsFHkUGavl+p8BYZ5
NGo7wxdPFlybCVnQEqntAkEAgAxkJKl9G8ixTQrZ3zP4QD/FkkTC77avypmfR3iX
wdEVui7Ff47glE0SYV9I4iUalyBOcT9OxyVZcNuf8LgUnA==
-----END RSA PRIVATE KEY-----
[10/07/22]seed@VM:~/.../task1$
```

```
[10/07/22]seed@VM:~/.../task1$ openssl req -new -key server.key -ou
t server.csr -config openssl.cnf
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorpo
rated
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:Computer Science
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2020.com
Email Address []:12345@gmail.edu

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:SysLab
[10/07/22]seed@VM:~/.../task1$
```

Step 3: Generating Certificates

In order to form a certificate CSR file needs to have CA's signature.
Signing signature(CA) to generate certificates by below command -

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

The requested signing certificate (server.csr) into an X509 certificate(server.crt) by using CA's key ca.crt and ca.key:

```
[10/07/22]seed@VM:~/.../task1$ 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: Oct  7 16:46:18 2022 GMT
            Not After : Oct  7 16:46:18 2023 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = CA
            organizationName          = CPP
            organizationalUnitName    = Computer Science
            commonName                = SEEDPKILab2020.com
            emailAddress              = 12345@gmail.edu
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                93:C4:22:E2:1E:7A:A9:4C:A6:0B:10:B9:2F:5C:3E:61:38:62:B7:19
            X509v3 Authority Key Identifier:
                keyid:7B:FB:43:22:D6:92:57:42:F9:75:7A:8E:E9:BC:89:40:57:63:6C:31

Certificate is to be certified until Oct  7 16:46:18 2023 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
[10/07/22]seed@VM:~/.../task1$
```

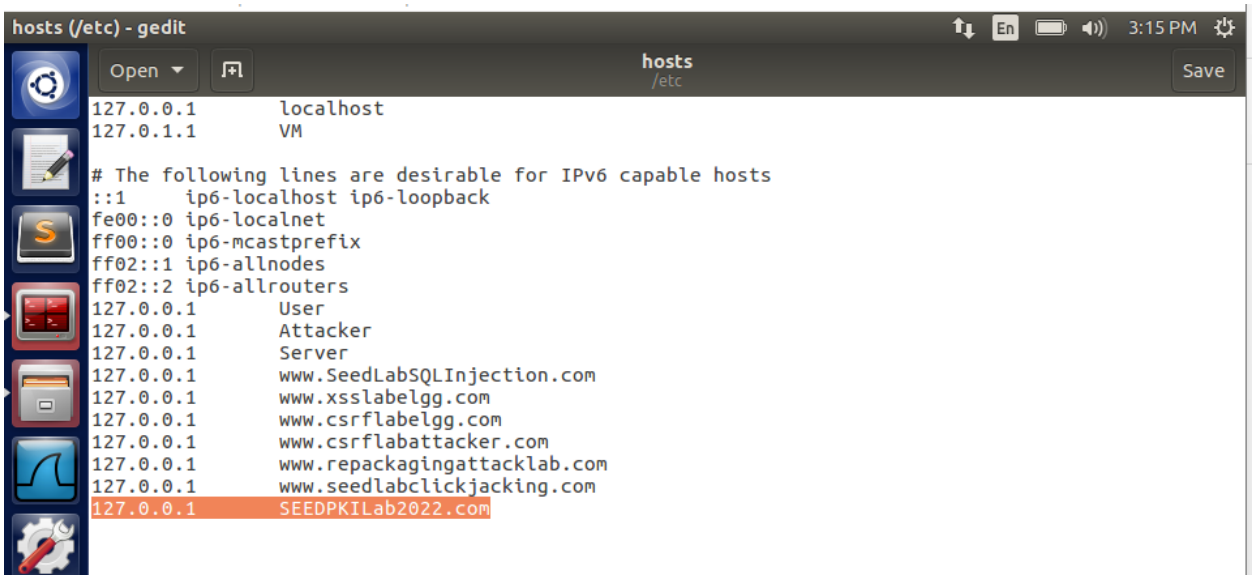
Task 3 - Deploying Certificate in an HTTPS Web Server

Step 1: Configuring DNS

I chose SEEDPKILab2022.com as the name of the website. To recognize my computer this name, added /etc/hosts the below line:

127.0.01 SEEDPKILab2022.com

S



Step 2: Configuring the web server

Launched the web server by combining the secret key and certificate into one file by following commands:

Combine the secret key and certificate into one file

cp server.key server.pem

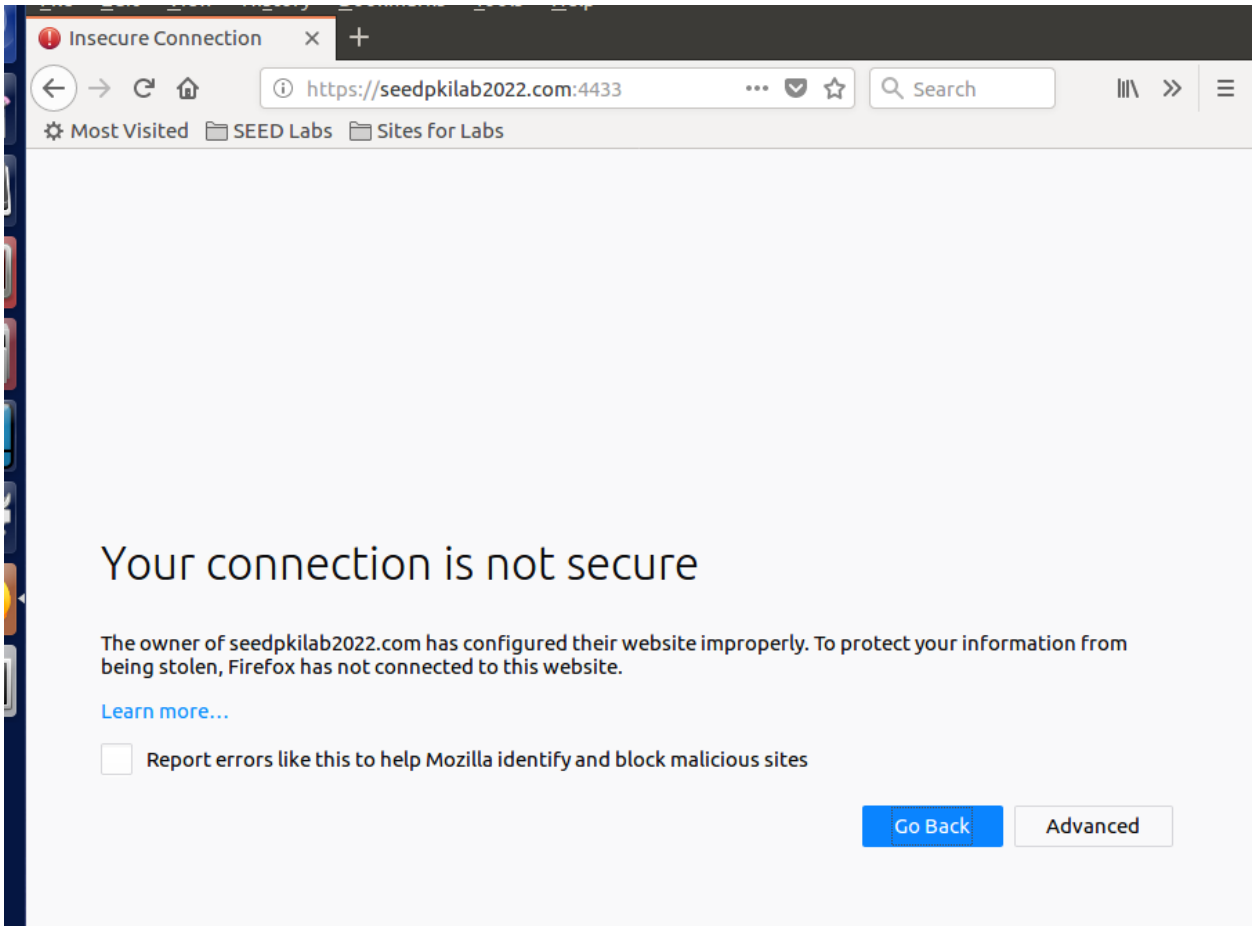
Cat server.crt >> server.pem

Launching the web server using server.pem

openssl s_server -cert server.pem -www

```
[10/07/22]seed@VM:~/.../lab3work$ cp server.key server.pem
[10/07/22]seed@VM:~/.../lab3work$ cat server.crt >> server.pem
[10/07/22]seed@VM:~/.../lab3work$ openssl s_server -cert server.pem
 -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

Now the URL: https://SEEDPKILab2022.com:4433 is entered in firefox new tab and error is shown as below.

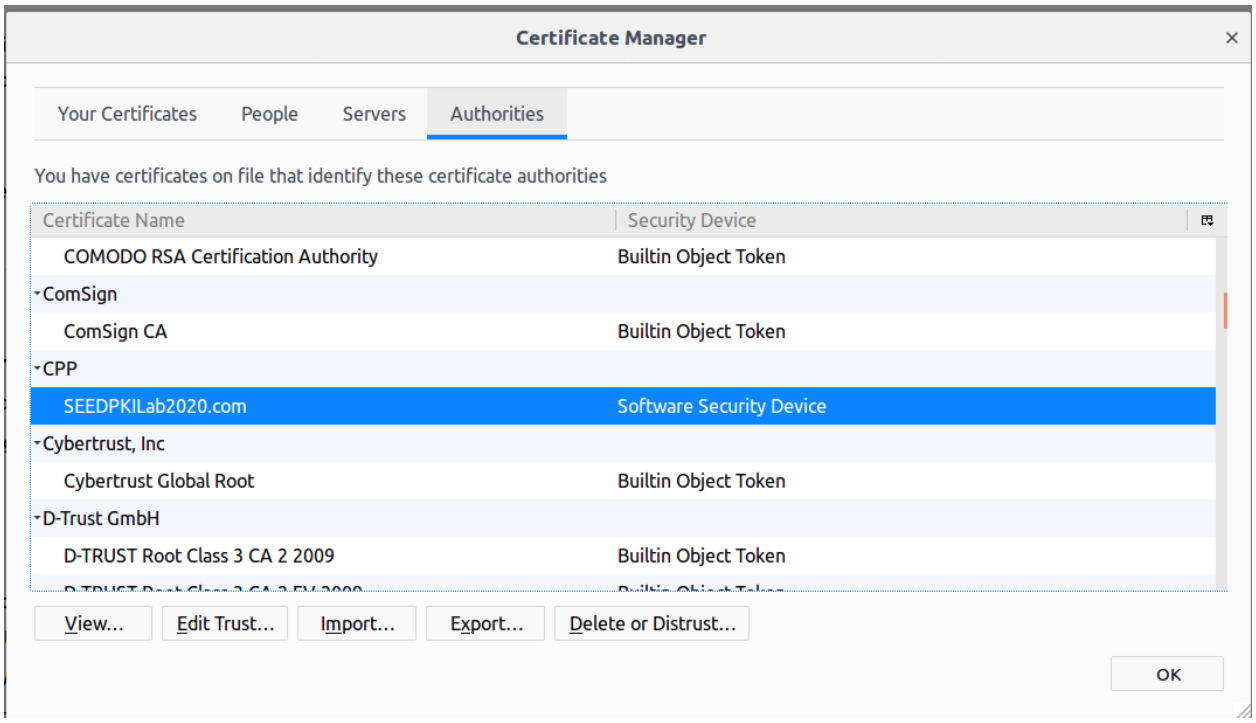


The above error is shown because the certificate of SEEDPKILab2022.com is signed by our own CA (i.e. using ca.crt) and this is not recognized by Firefox.

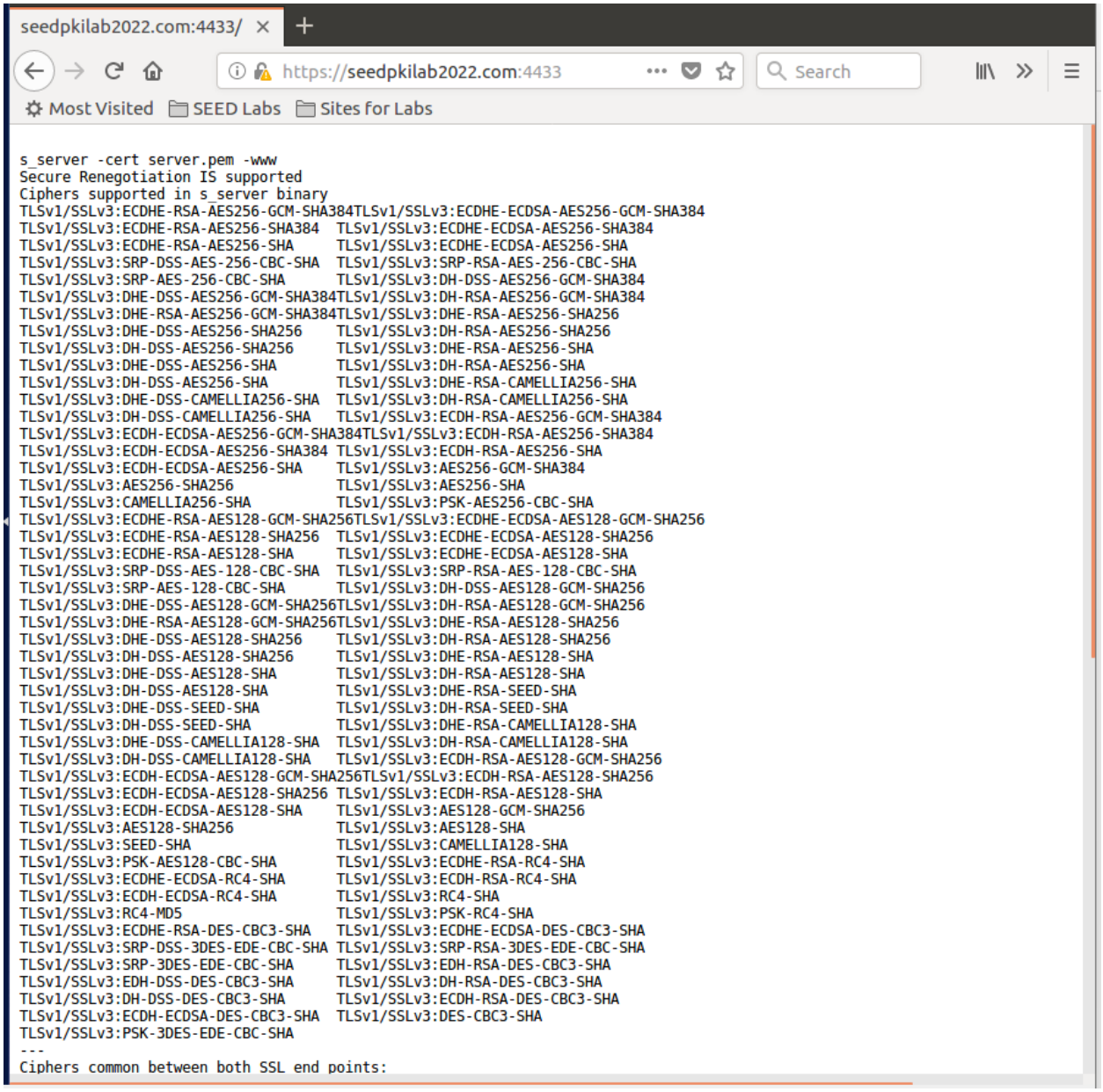
Step 3: Getting the browser to accept our CA certificate

In this step we are requesting Firefox browser to include our CA's certificate by importing ca.crt into Firefox settings by following steps:

Edit-> Preference-> Privacy & Security-> View Certificates.



Now when we refresh the Firefox tab of https://SEEDPKILab2022.com: 4433 we can see a running website.



Step 4 : Testing our HTTPs website

Here we change a single byte of server.pem file and restart the server.

server.pem

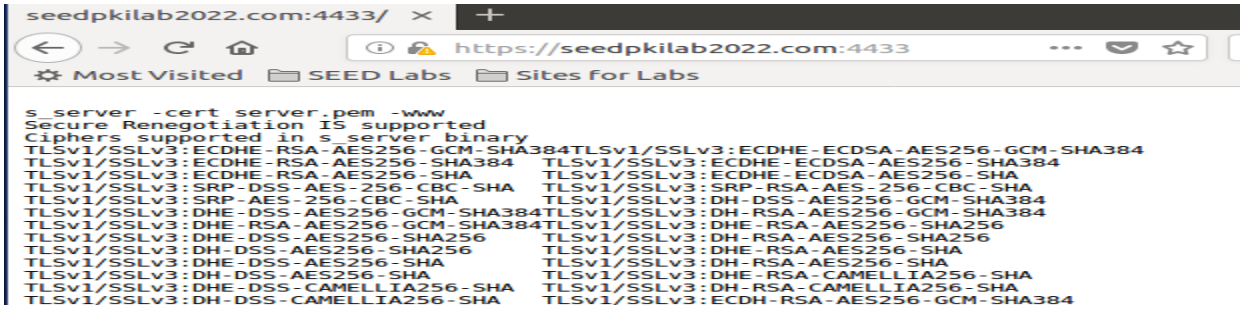
```
00000000 2D 2D 2D 2D 2D 42 45 47 49 4E 20 52 53 41 20 50 52 49 56 -----BEGIN RSA PRIV
00000013 41 54 45 20 4B 45 59 2D 2D 2D 2D 2D 0A 50 72 6F 63 2D 54 ATE KEY-----.Proc-T
00000026 79 70 65 3A 20 34 2C 45 4E 43 52 59 50 54 45 44 0A 44 45 ype: 4,ENCRYPTED.DE
00000039 4B 2D 49 6E 66 6F 3A 20 41 45 53 2D 31 32 38 2D 43 42 43 K-Info: AES-128-CBC
0000004c 2C 35 41 41 37 30 30 35 37 30 44 34 30 34 46 41 33 38 37 ,5AA700570D404FA387
0000005f 42 30 43 42 36 41 32 32 35 45 36 32 35 42 0A 0A 43 66 76 B0CB6A225E625B..Cfv
00000072 6B 70 64 4D 2F 7A 5A 77 4B 31 56 31 5A 4B 7A 78 36 65 45 kpdM/zZwK1V1ZKzx6eE
00000085 4F 79 5A 39 6D 68 57 70 37 52 6F 65 57 43 4D 36 41 54 44 OyZ9mhWp7RoeWCM6ATD
00000098 6F 43 62 37 77 58 55 66 59 77 6A 2F 6F 75 49 52 68 35 47 oCb7wXUfYwj/ouIRh5G
000000ab 66 6C 2F 69 0A 4C 50 64 61 4C 6A 4A 59 56 2F 2F 59 49 6B fl/i.LPdaLjJYV//YIk
000000be 67 54 68 77 2F 54 50 70 62 71 65 2B 2F 4C 53 4A 4A 6E 67 gThw/TPpbqe+/LSJJng
000000d1 57 69 6E 6A 4F 37 52 6D 64 4F 49 66 46 52 6F 6D 2B 39 68 WinjO7RmdOIfFRom+9h
000000e4 45 6C 4F 53 79 51 49 4C 77 69 43 79 0A 7A 79 50 35 31 64 ElOSyQILwiCy.zyP51d
000000f7 67 61 34 5A 6E 66 4D 63 51 6A 55 78 6C 53 7A 30 50 58 30 ga4ZnfMcQjUxlSz0PX0
0000010a 63 45 2F 69 32 6E 6A 6C 30 30 76 7A 61 73 66 54 72 75 45 cE/i2njl00vzasfTruE
```

The server is unable to start.

```
[10/07/22]seed@VM:~/.../lab3work$ openssl s_server -cert server.pem
 -www
Enter pass phrase for server.pem:
Using default temp DH parameters
error setting private key
3070928576:error:0B080074:x509 certificate routines:X509_check_priv
ate_key:key values mismatch:x509_cmp.c:340:
[10/07/22]seed@VM:~/.../lab3work$
```

Now trying to change at some other place in the server.pem file and reload the URL.
The server started reloading again and the URL is also working fine.

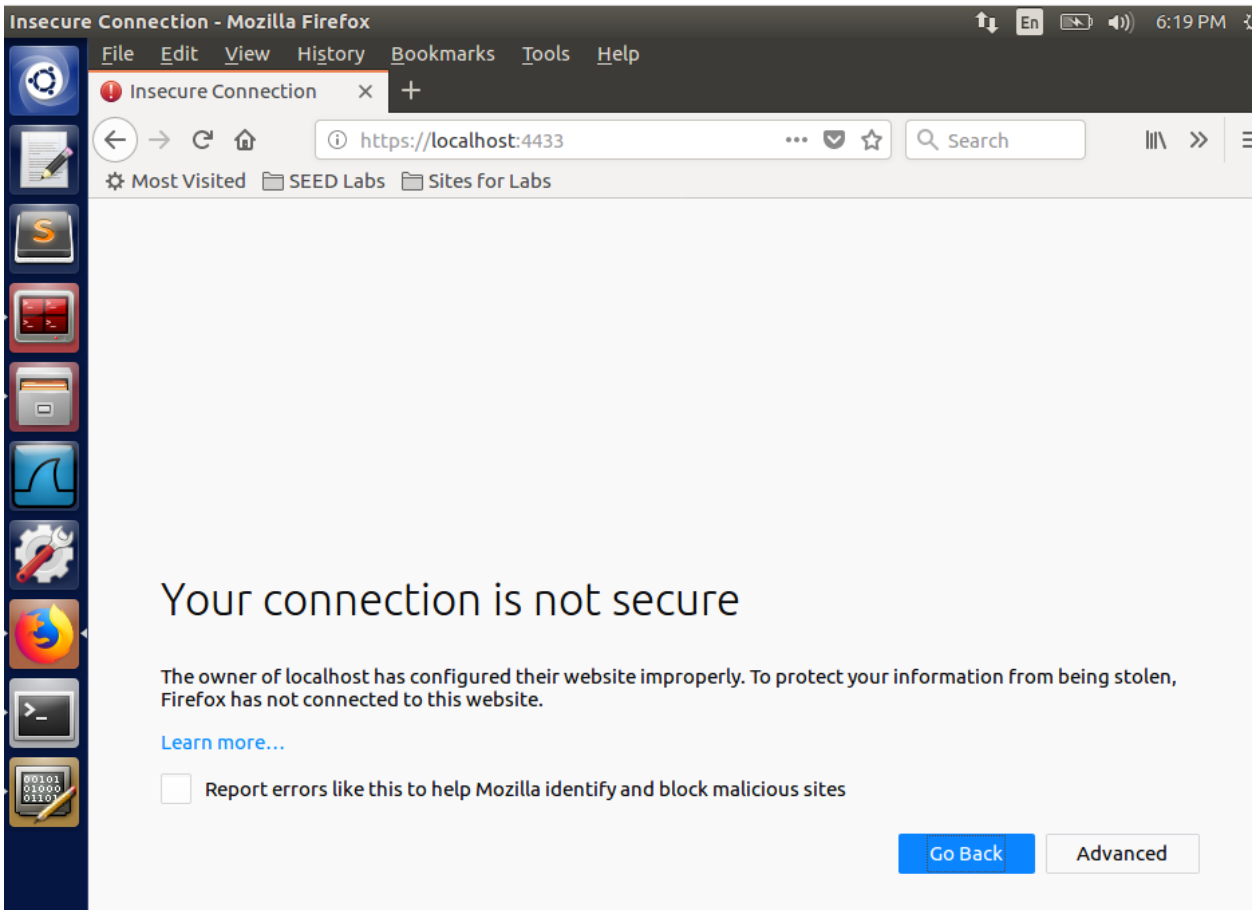
```
[10/07/22]seed@VM:~/.../lab3work$ openssl s_server -cert server.pem
 -www
Enter pass phrase for server.pem:
Using default temp DH parameters
error setting private key
3070928576:error:0B080074:x509 certificate routines:X509_check_priv
ate_key:key values mismatch:x509_cmp.c:340:
[10/07/22]seed@VM:~/.../lab3work$ openssl s_server -cert server.pem
 -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
```



We were able to access the web server although the single byte was changed again in another place because the web server does not have the facility to check the integrity of the file before running it. Hence we can say that as certain parts of the server.pem are not altered, the server would run fine.

Use of localhost:

When I try to browse https://localhost:4433/ it is reported as 'connection is not secure'.



Because the localhost has no certificate, and the certificate is bound to the domain as seedpkilab2022.com. Hence, we tried to attempt using the localhost address we get the warning as 'Your connection is not secure'.

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

In this task Apache server is used to host the website SEEDPKILab2022.com with certification key and server key. For this we first added below lines to default-ssl.conf file.

```
# To open editor
cd /etc/apache2/sites-available
```

sudo gedit /etc/apache2/sites-available/default-ssl.conf
#Added following lines

Then copied the server and private key to the folder to have access with commands:

sudo mkdir /var/www/pki

sudo cp server.pem server.key server.crt /var/www/pki

```
[10/07/22]seed@VM:~/.../task4$ sudo cp server.pem server.crt /var/w
ww/pki
```

Testing Apache for errors and enabling SSL module, site and restarting Apache with following commands:

// Test the Apache configuration file for errors

sudo apachectl configtest

// Enable the SSL module

sudo a2enmod ssl

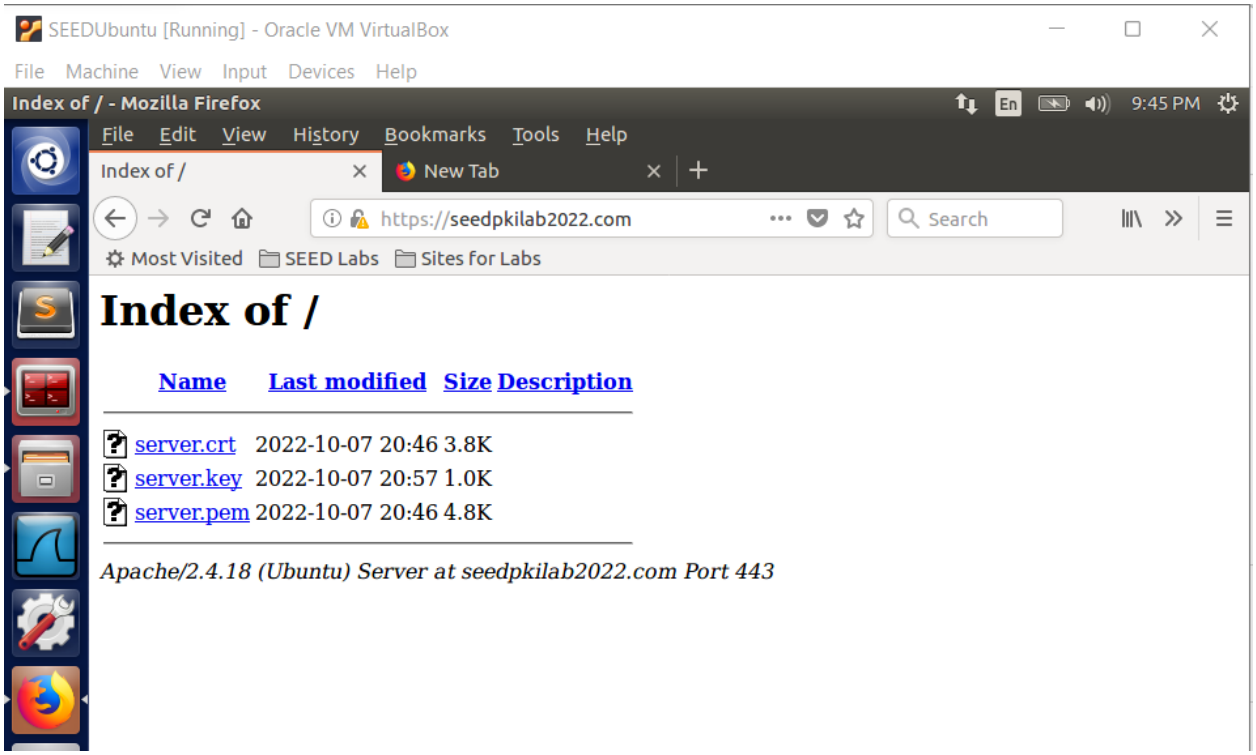
// Enable the site we have just edited

sudo a2ensite default-ssl

// Restart Apache : **sudo service apache2 restart**

```
[10/07/22]seed@VM:~/.../task4$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does
not exist
AH00558: apache2: Could not reliably determine the server's fully q
ualified domain name, using 127.0.1.1. Set the 'ServerName' directi
ve globally to suppress this message
Syntax OK
[10/07/22]seed@VM:~/.../task4$ 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
[10/07/22]seed@VM:~/.../task4$ sudo a2ensite default-ssl
Site default-ssl already enabled
[10/07/22]seed@VM:~/.../task4$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2022.com:443 (RSA):
 *****
```

Then Firefox opened https://seedpkilab2022.com once the Apache service restarted.



The website seedpkilab2022.com was loaded successfully including all signatures verified.

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

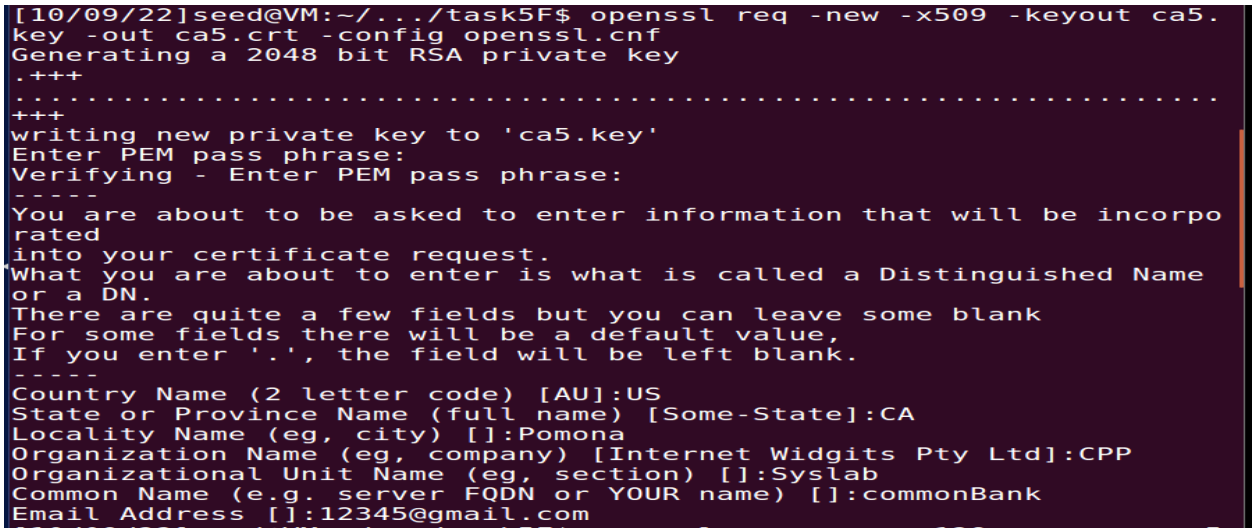
In this task we will see how we can defeat Man-In-The-Middle (MITM) attacks. We are using a social networking site called ‘twitter.com’ to complete this task.

At first MITM attack we tried with ‘Bank of America’ as the target site but because bank of america sites were so secure and utilizing total security couldn’t get successful notification from the victim VMs about the alarm. Whenever I tried and typed ‘https://bankofamerica.com’ it directly loaded successfully the correct site. I even tried changing names to bankamerica similar to bank of america and generated all keys and certificates of having as bankamerica but still didn’t work. So far based on my research I found Bank of America sites to be very secure. I even tried with SEEDPKILab2022 keys and CA’s.

When I took social networking site ‘Twitter’ as our target sites and applied same all keys and certificates signing of bank of bankamerica it worked. Below are the steps and screenshots attached:

Step -1: Setting up the malicious website.

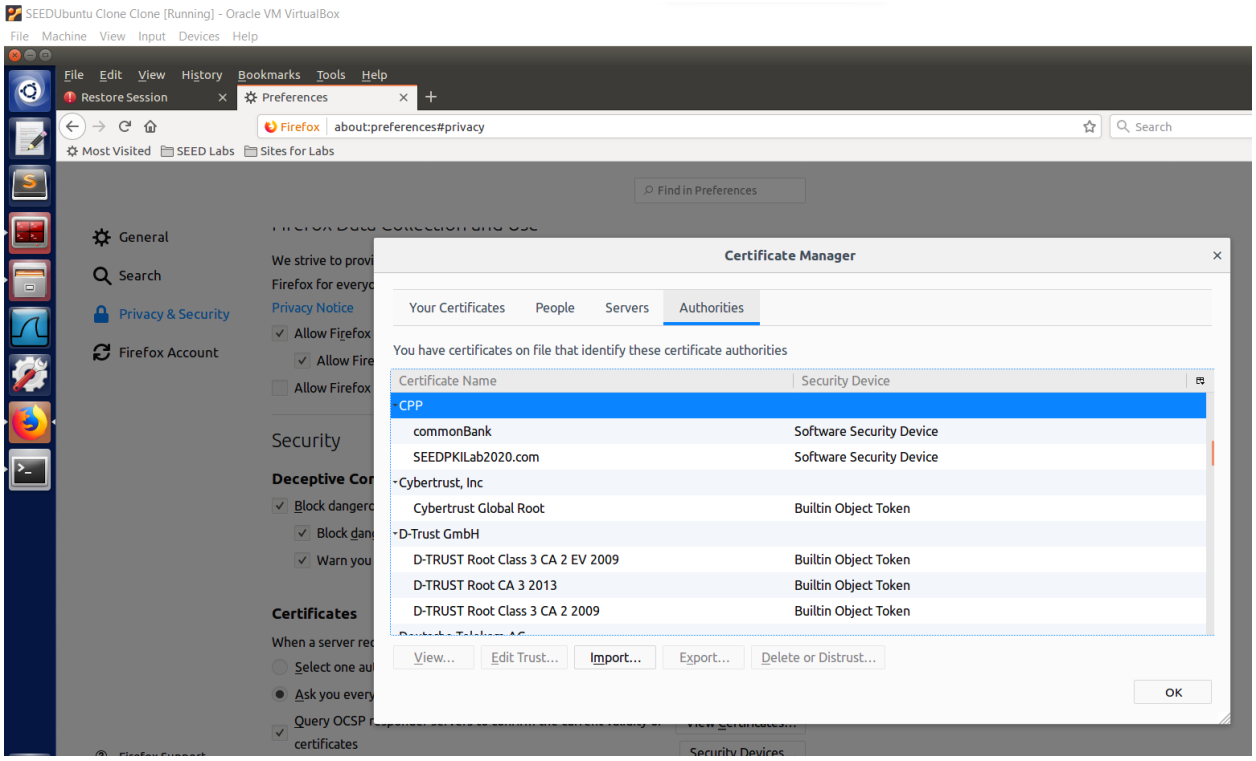
Generating key



```
[10/09/22]seed@VM:~/.../task5F$ openssl genrsa -aes128 -out server5
.key 1024
Generating RSA private key, 1024 bit long modulus
..................++++++
.....................++++++
e is 65537 (0x10001)
Enter pass phrase for server5.key:
Verifying - Enter pass phrase for server5.key:
```

Signing certificates and then we imported the ca.crt. So that we can get the website on and display the message

```
[10/09/22]seed@VM:~/.../task5F$ openssl ca -in server5.csr -out ser
ver5.crt -cert ca5.crt -keyfile ca5.key -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca5.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Oct 10 02:03:31 2022 GMT
            Not After : Oct 10 02:03:31 2023 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = CA
            organizationName          = CPP
            organizationalUnitName    = SysLab
            commonName                = BankAmerica.com
            emailAddress              = 12345@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                B6:8C:36:D3:C7:3B:DB:40:34:D4:88:31:68:AF:85:3B:14:
39:3B:F8
            X509v3 Authority Key Identifier:
                keyid:15:31:3E:55:EB:B9:74:1F:9A:B8:B1:BE:03:BB:CB:
80:3D:2F:69:2E

Certificate is to be certified until Oct 10 02:03:31 2023 GMT (365
days)
Sign the certificate? [y/n]:y
```



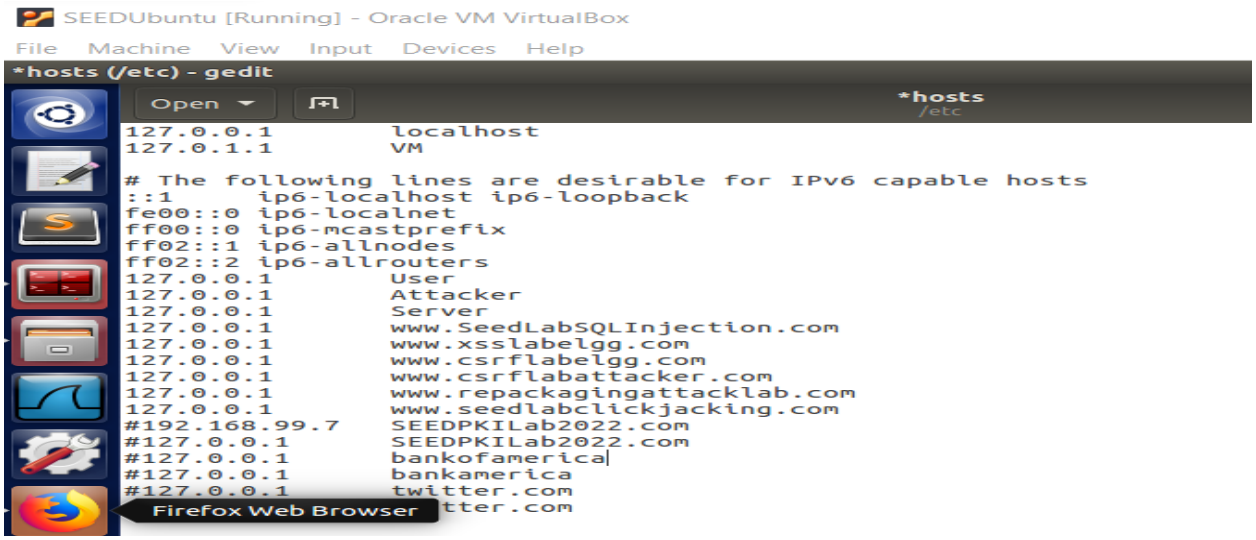
Above image we are importing the certificate ca.crt for bank of america.

Step 2: Becoming the man in the middle

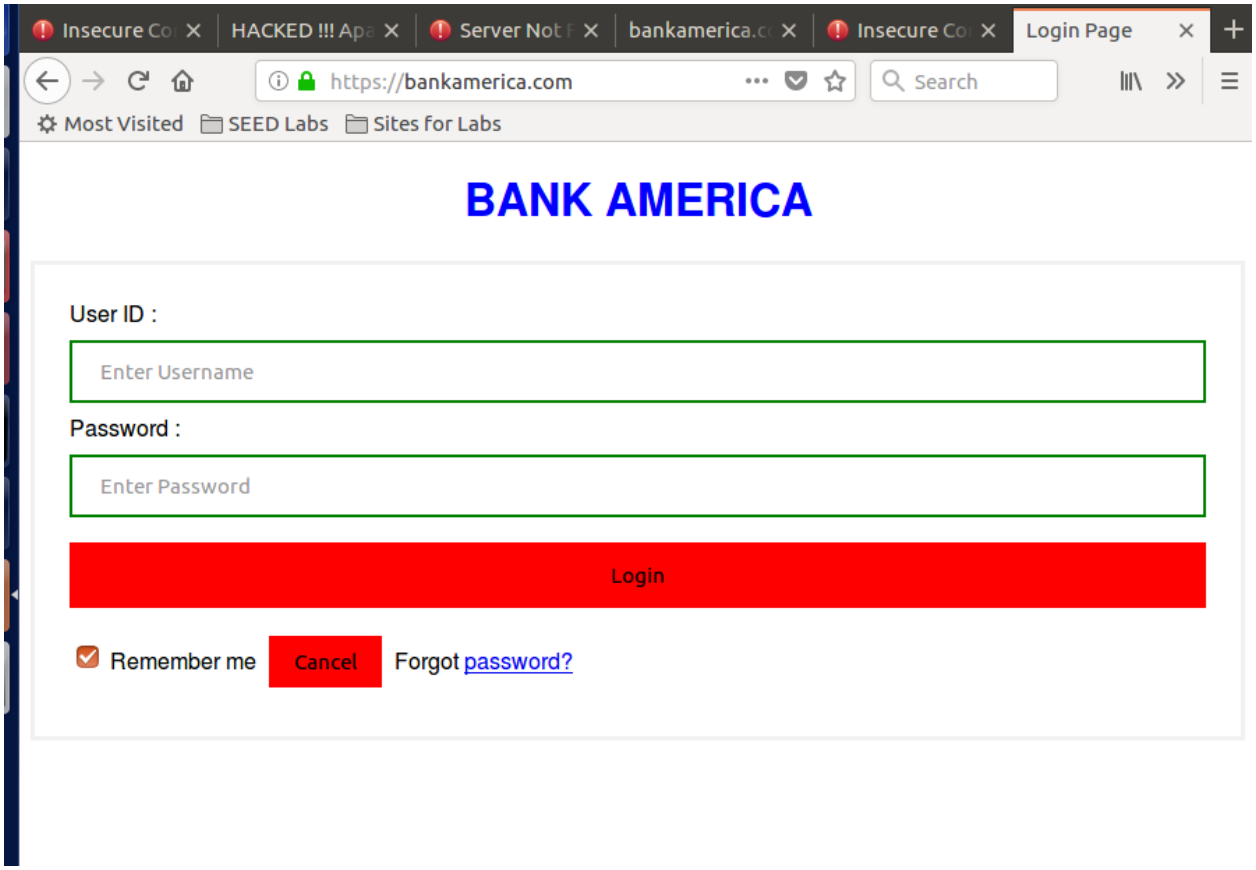
The man-in-the-middle is set with the following steps:

Add an entry to /etc/hosts (with root privilege) on the victim to simulate DNS cache poisoning.

As such, the modified would be as shown below:



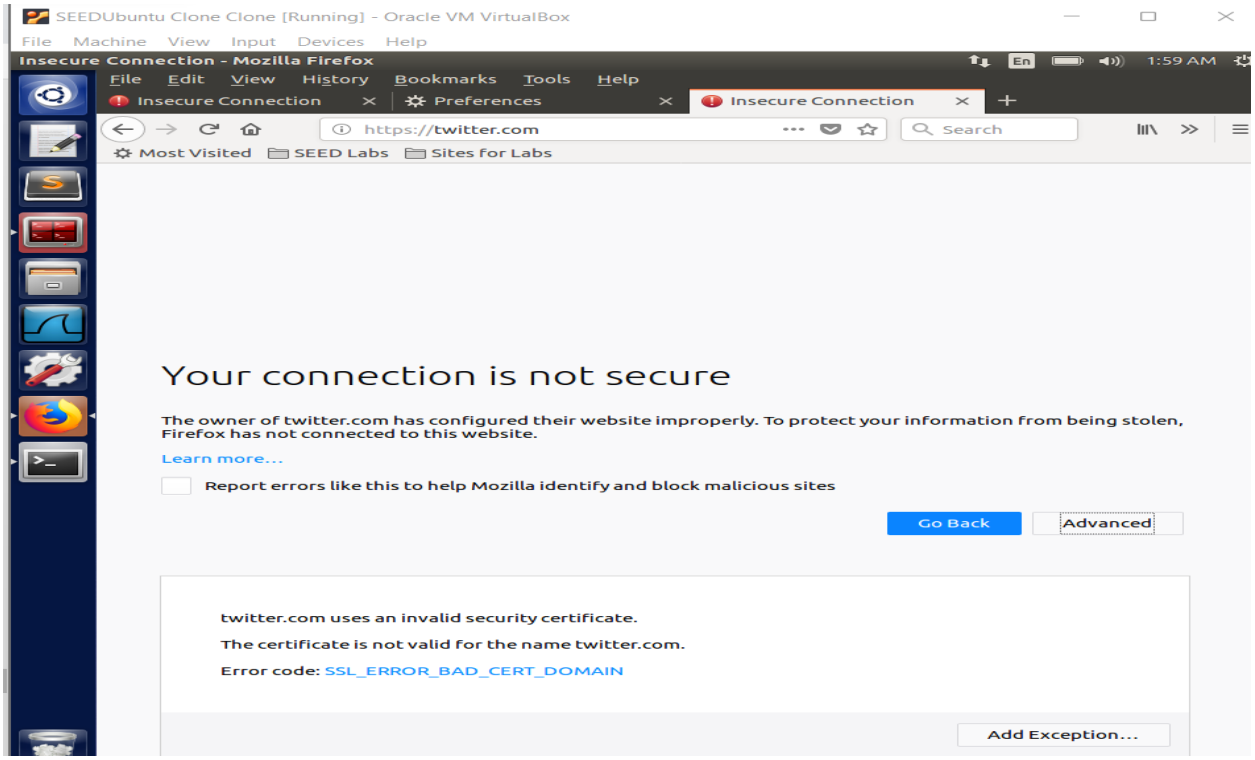
```
[10/09/22]seed@VM:~/.../task5F$ openssl req -new -key server5.key -
out server5.csr -config openssl.cnf
Enter pass phrase for server5.key:
You are about to be asked to enter information that will be incorpo
rated
into your certificate request.
What you are about to enter is what is called a Distinguished 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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:SysLab
Common Name (e.g. server FQDN or YOUR name) []:BankAmerica.com
Email Address []:12345@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:banking
```



The above website still uses the previous seepkilab2022 website domain. So now we will change the domain to be as bankofamerica.com.

Step 3: Browse the target website.

When the target site is accessed, an error saying that the connection is not secure is displayed as shown below.



After clicking Advanced, the error is caused by an invalid security certificate being used, with the error code of SSL_ERROR_BAD_CERT_DOMAIN. This is expected, as the certificate used is for SEEDPKILab2022.com, but the domain we are trying to access is twitter.com. Due to this misalignment, an error is flagged.

Extra work for trying Task -5 taking target site as Bank of America screenshots are below:

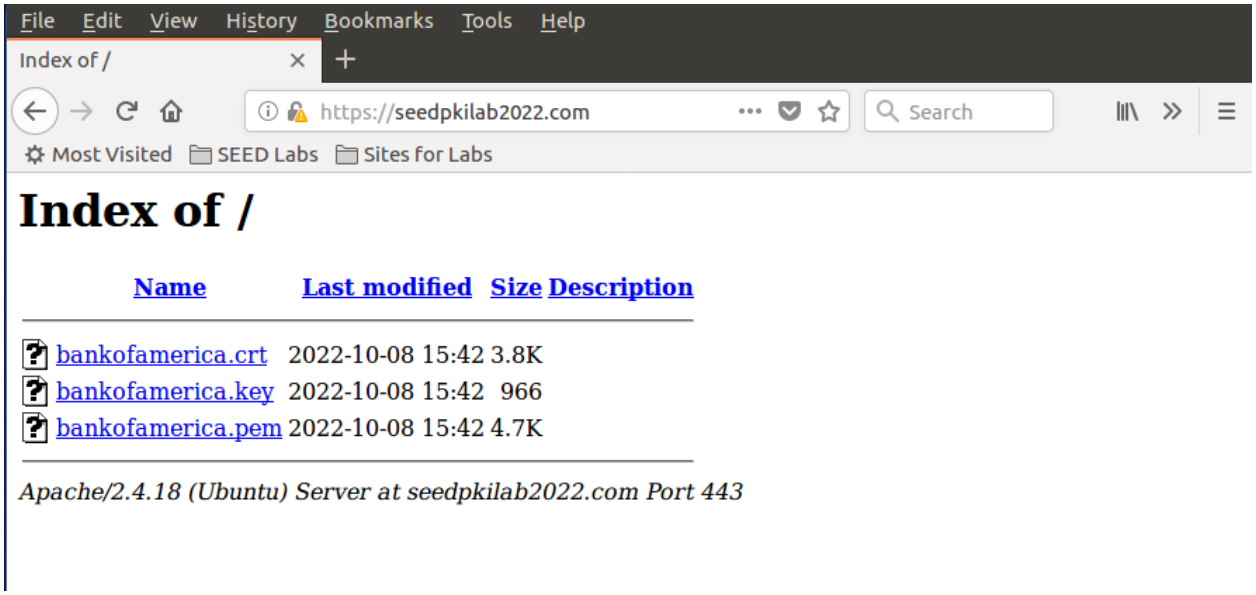
```
[10/08/22]seed@VM:~/.../task5$ openssl genrsa -aes128 -out bankofam
erica.key 1024
Generating RSA private key, 1024 bit long modulus
.............++++++
.......................++++++
e is 65537 (0x10001)
Enter pass phrase for bankofamerica.key:
Verifying - Enter pass phrase for bankofamerica.key:
```

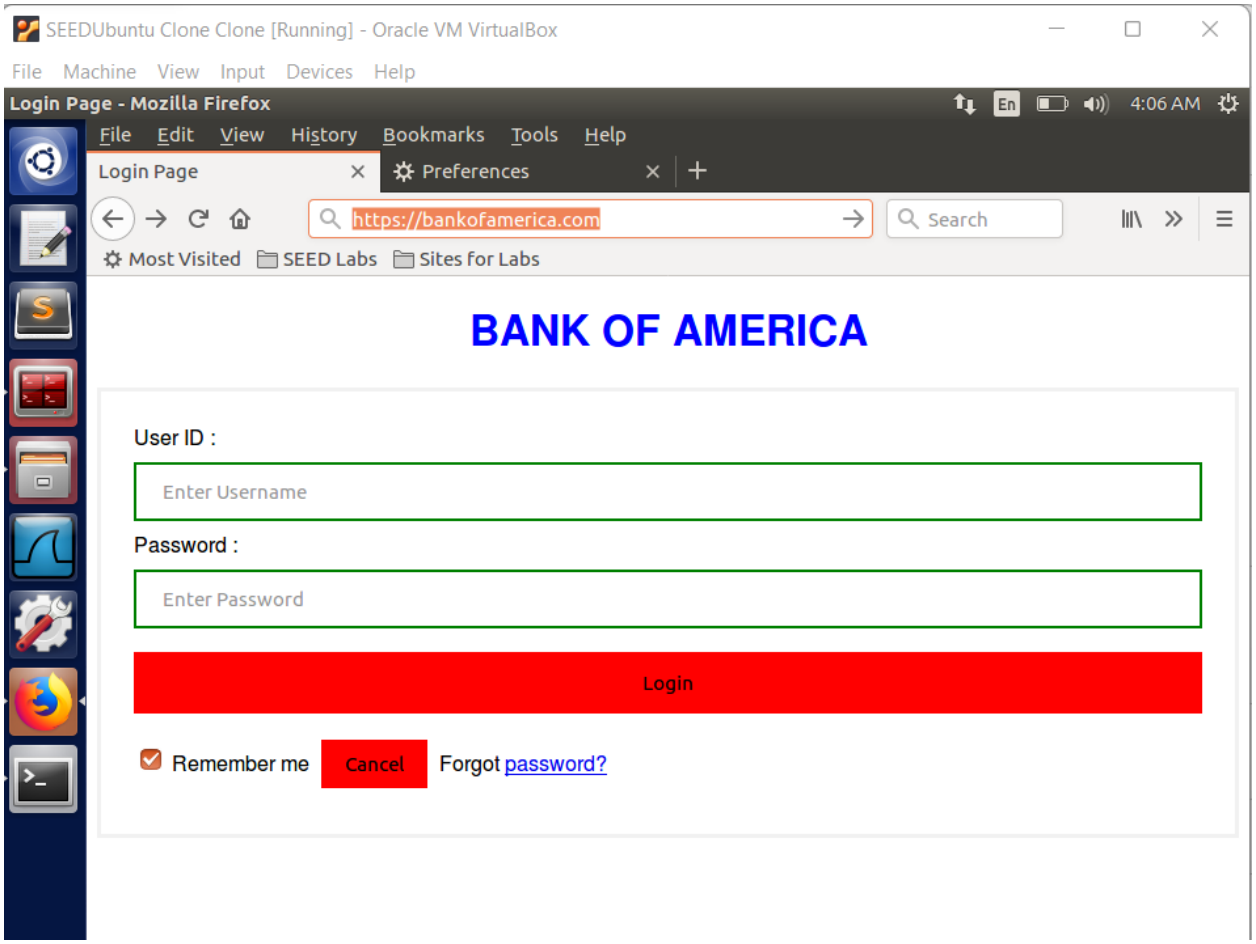
```
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[10/08/22]seed@VM:~/.../lab3work$ cp bankofamerica.key bankofameric
a.pem
[10/08/22]seed@VM:~/.../lab3work$ cat bankofamerica.crt >> bankofam
erica.pem
[10/08/22]seed@VM:~/.../lab3work$ sudo mkdir /var/www/bankofamerica
[10/08/22]seed@VM:~/.../lab3work$ sudo cp bankofamerica.crt bankofa
merica.key bankofamerica.pem /var/www/bankofamerica
[10/08/22]seed@VM:~/.../lab3work$ sudo gedit /etc/apache2/sites-ava
ilable/default-ssl.conf
```

```
[10/08/22]seed@VM:~/.../task5$ openssl req -new -key bankofamerica.
key -out bankofamerica.csr -config openssl.cnf
Enter pass phrase for bankofamerica.key:
You are about to be asked to enter information that will be incorpo
rated
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:Banking
Common Name (e.g. server FQDN or YOUR name) []:Rit
Email Address []:12345@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:
[10/08/22]seed@VM:~/.../task5$
```

```
[10/08/22]seed@VM:~/.../lab3work$ openssl req -new -key bankofameri
ca.key -out bankofamerica.csr -config openssl.cnfEnter pass phrase
for bankofamerica.key:
You are about to be asked to enter information that will be incorpo
rated
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:Banking
Common Name (e.g. server FQDN or YOUR name) []:Rit
Email Address []:12345@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:
```

```
    <VirtualHost *:443>
        ServerName bankofamerica.com
        DocumentRoot /var/www/bankofamerica
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile /var/www/bankofamerica/bankofamerica.crt
        SSLCertificateKeyFile /var/www/bankofamerica/bankofamerica.pem
```




SEEDUbuntu Clone Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Login Page - Mozilla Firefox
4:06 AM
File Edit View History Bookmarks Tools Help
Login Page
Preferences
https://bankofamerica.com
Search
Most Visited
SEED Labs
Sites for Labs
BANK OF AMERICA
User ID :
Enter Username
Password :
Enter Password
Login
Remember me
Cancel
Forgot password?

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

As we saw in the above task, root CA was compromised by an attacker and also its private key was stolen. The attacker can also generate any arbitrary certificate using the CA's private key. So, in this current task we will see the consequences of compromised root CA by below steps:

Generating keys : Since, CSR file generated by anyone, therefore the attacker generates CSR file with its own key. So, the attacker can fool the victim into making a CSR file with a similar name.

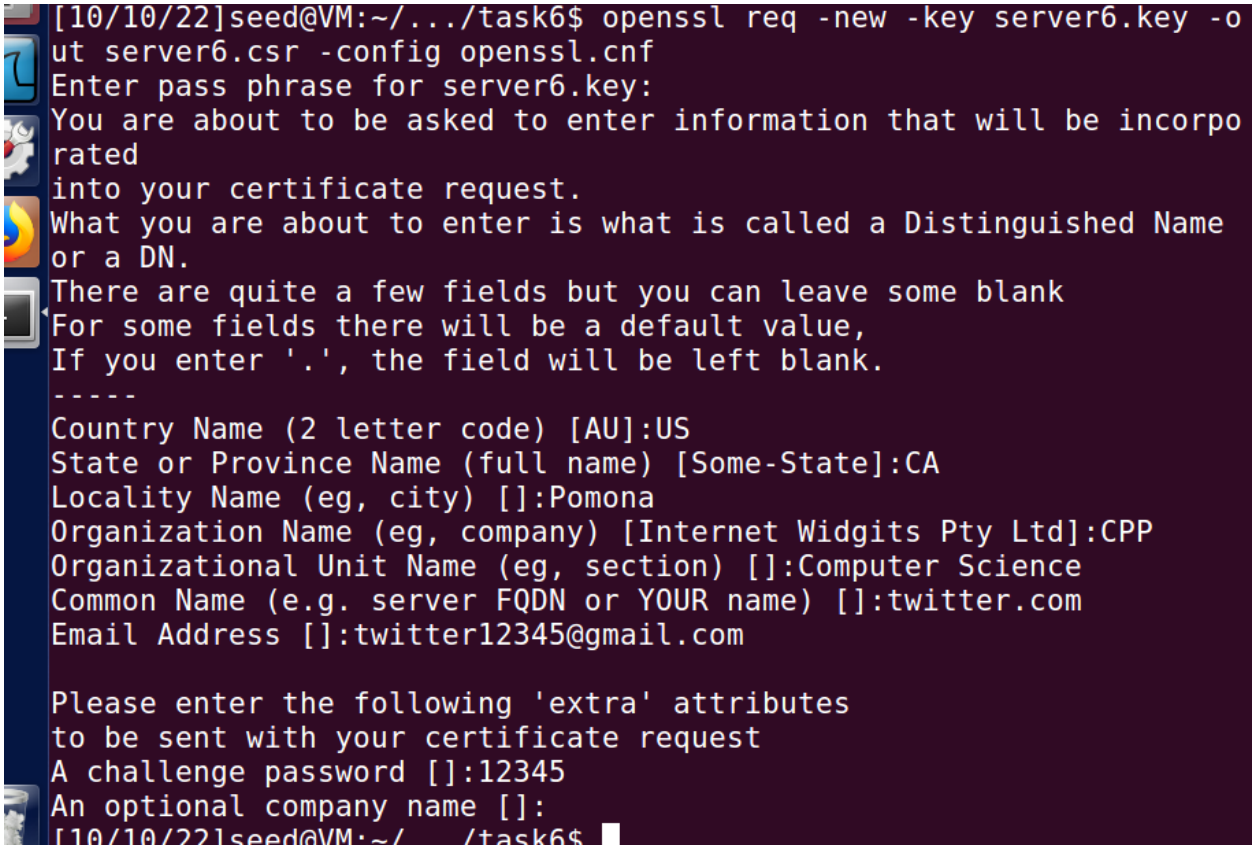
First the attacker generates their own private/public rsa keys:

```
[10/10/22]seed@VM:~/.../task6$ openssl genrsa -aes128 -out server6.
key 1024
Generating RSA private key, 1024 bit long modulus
.............++++++
...++++++
e is 65537 (0x10001)
Enter pass phrase for server6.key:
Verifying - Enter pass phrase for server6.key:
```

They then create a certificate signing request claiming they own: www.twitter.com

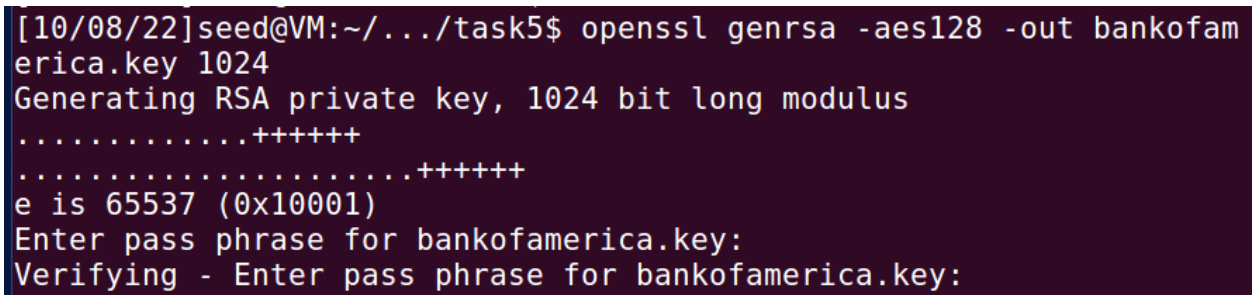
```
[10/10/22]seed@VM:~/.../task6$ openssl req -new -x509 -keyout ca6.k
ey -out ca6.crt -config openssl.cnf
Generating a 2048 bit RSA private key
.....................................+++
.....................................................+
++
writing new private key to 'ca6.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorpo
rated
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:Computer Science
Common Name (e.g. server FQDN or YOUR name) []:twitter.com
Email Address []:twitter12345@gmail.com
[10/10/22]seed@VM:~/.../task6$
```

.



For signing the certificates CA first verifies the owner but in this case the attacker already knows the private key. Therefore, the certificate is signed without any verification.
We tried to do with twitter but it did not work so we used bank of america.com

First the attacker generates their own private/public rsa keys:



They then create a certificate signing request claiming they own bank of america.com

```
[10/08/22]seed@VM:~/.../lab3work$ openssl req -new -key bankofameri
ca.key -out bankofamerica.csr -config openssl.cnfEnter pass phrase
for bankofamerica.key:
You are about to be asked to enter information that will be incorpo
rated
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:Banking
Common Name (e.g. server FQDN or YOUR name) []:Rit
Email Address []:12345@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:
```

The attacker then uses the CA's private key to sign the csr into a crt.

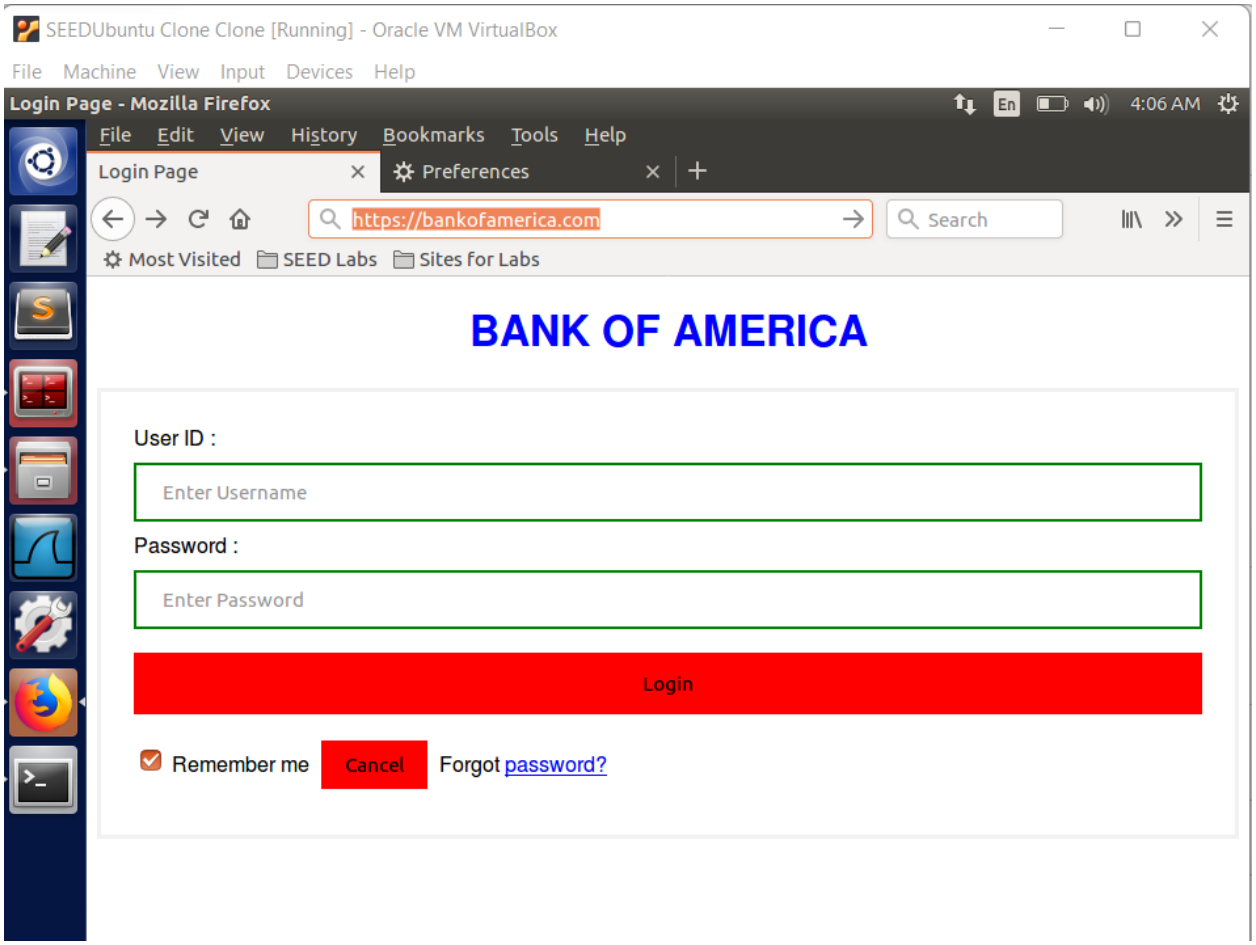
```
[10/09/22]seed@VM:~/.../task5F$ openssl ca -in server5.csr -out ser
ver5.crt -cert ca5.crt -keyfile ca5.key -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca5.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Oct 10 02:03:31 2022 GMT
            Not After : Oct 10 02:03:31 2023 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = CA
            organizationName          = CPP
            organizationalUnitName    = SysLab
            commonName                = BankAmerica.com
            emailAddress              = 12345@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                B6:8C:36:D3:C7:3B:DB:40:34:D4:88:31:68:AF:85:3B:14:
39:3B:F8
            X509v3 Authority Key Identifier:
                keyid:15:31:3E:55:EB:B9:74:1F:9A:B8:B1:BE:03:BB:CB:
80:3D:2F:69:2E

Certificate is to be certified until Oct 10 02:03:31 2023 GMT (365
days)
Sign the certificate? [y/n]:y
```

The attacker then modifies the apache/web server configuration to use the new valid certificate

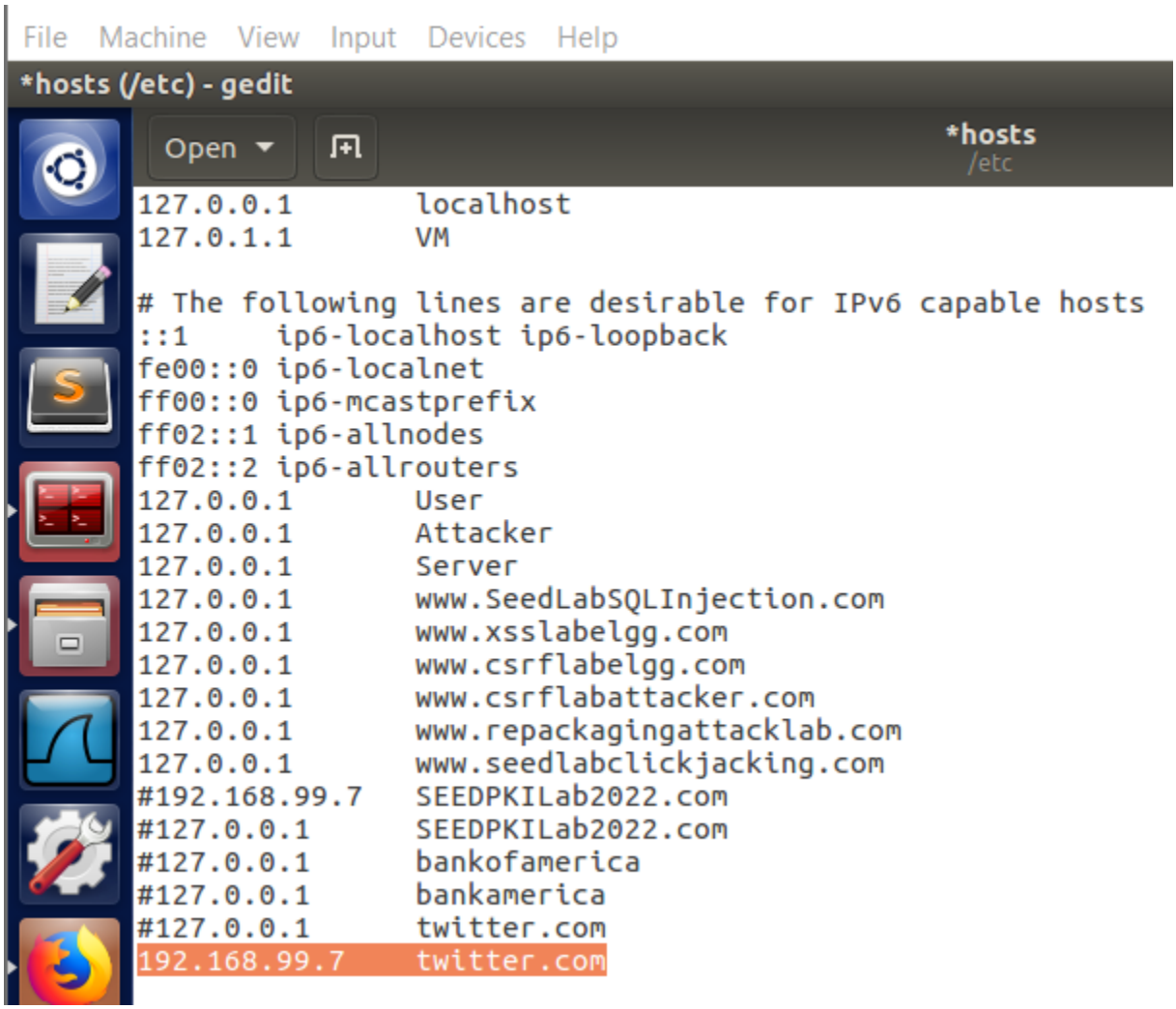
```
    <VirtualHost *:443>
        ServerName bankofamerica.com
        DocumentRoot /var/www/bankofamerica
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile /var/www/bankofamerica/bankofamerica.crt
        SSLCertificateKeyFile /var/www/bankofamerica/bankofamerica.pem
```

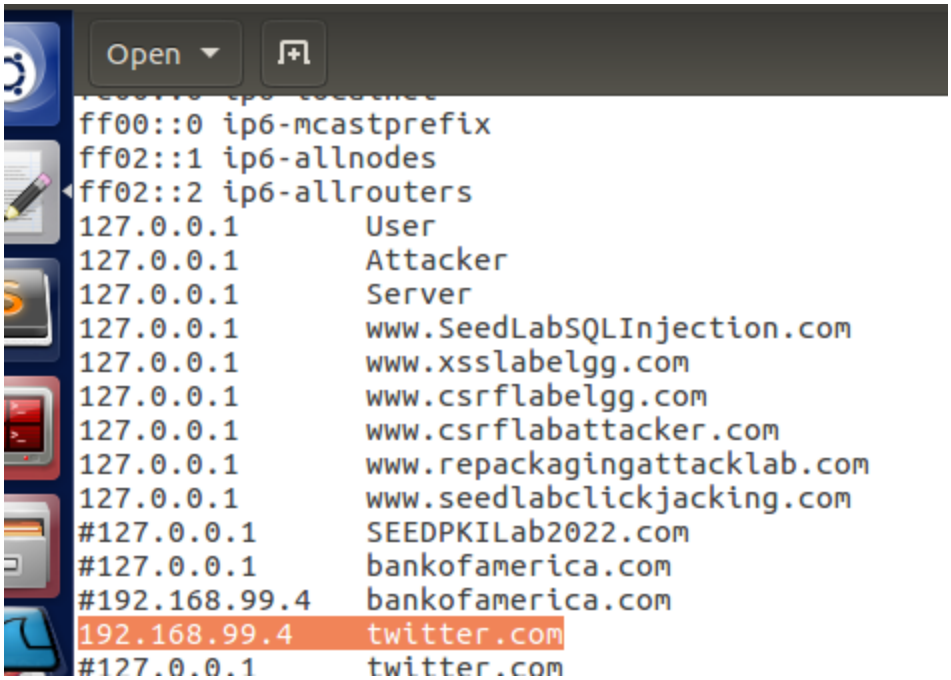
Now when the targeted user visits bank of america they are directed to the attackers website and firefox displays that the certificate is valid. We can see that the certificate is signed by bank of america which is the name of our CA, perhaps poorly named for this example.

To make the attack more convincing, the attacker can download the html for the spoofed website and serve it on their malicious server.
We can see in the screenshot below that the website is verified by our compromised CA named bank. The target site can then be accessed on the victim without alerting the browser, as seen below.

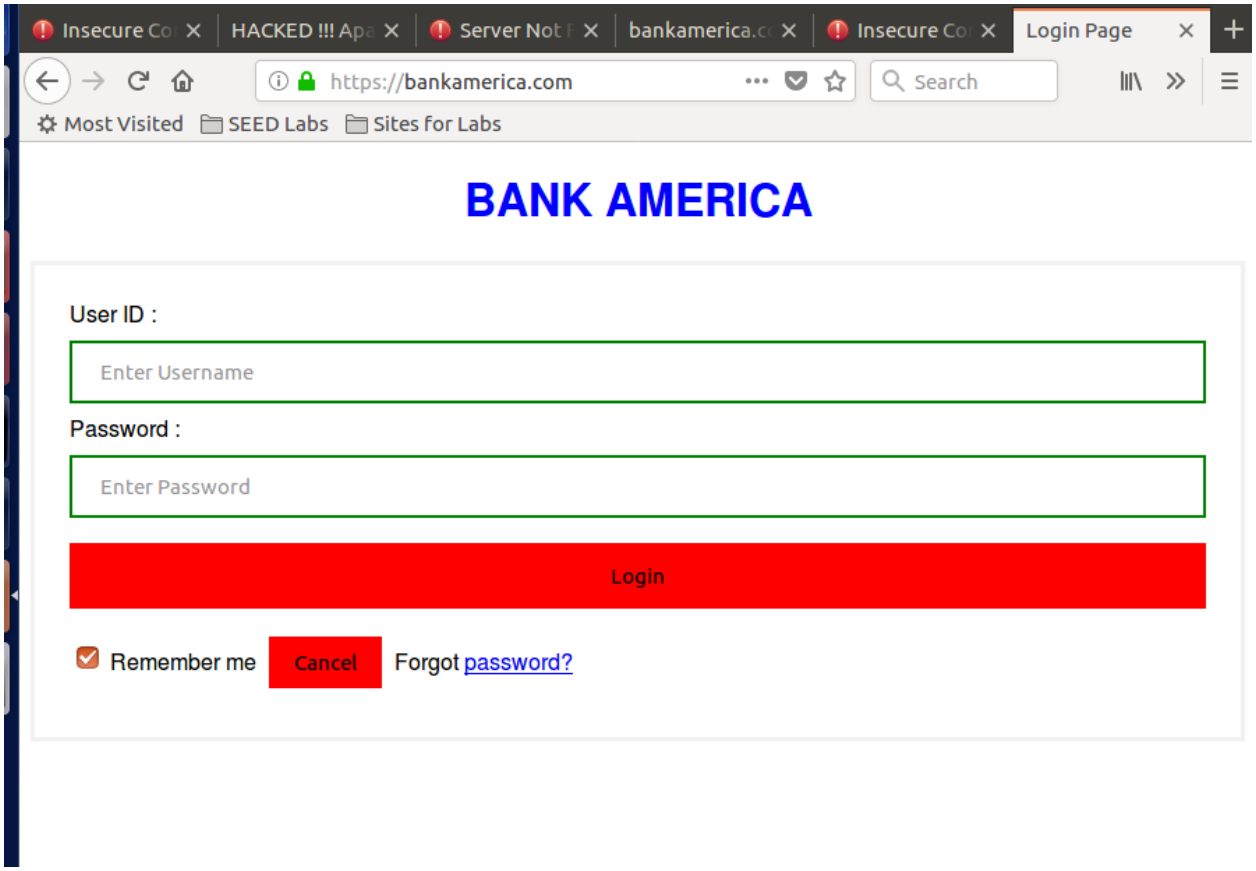


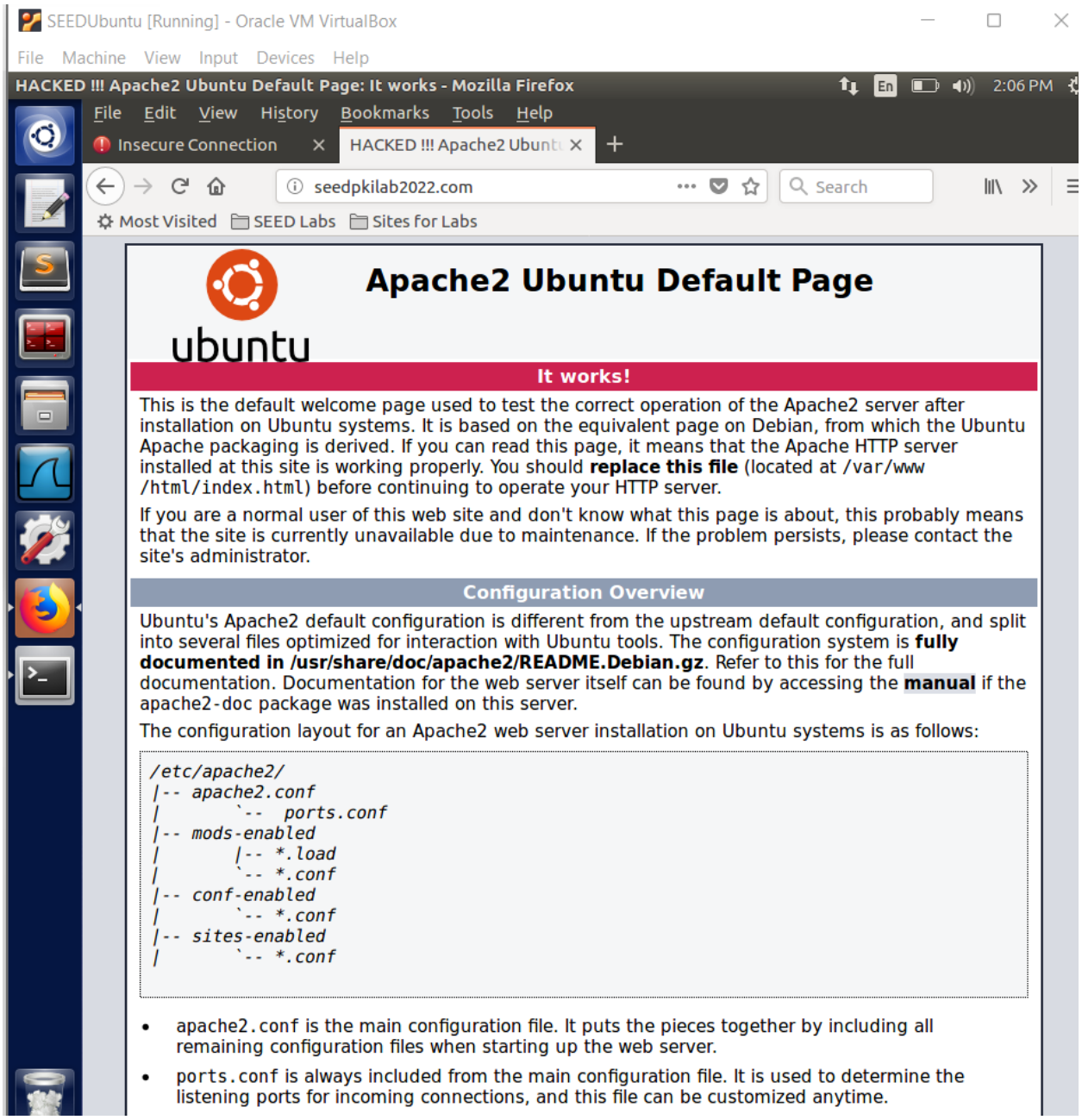
As seen above, the content can be modified on both HTTP and HTTPS web servers. Through this, attackers can possibly trick the victim to key in sensitive information, such as login credentials or credit card details. These details can then be used by the attacker for other sites to steal their private information and/or money.

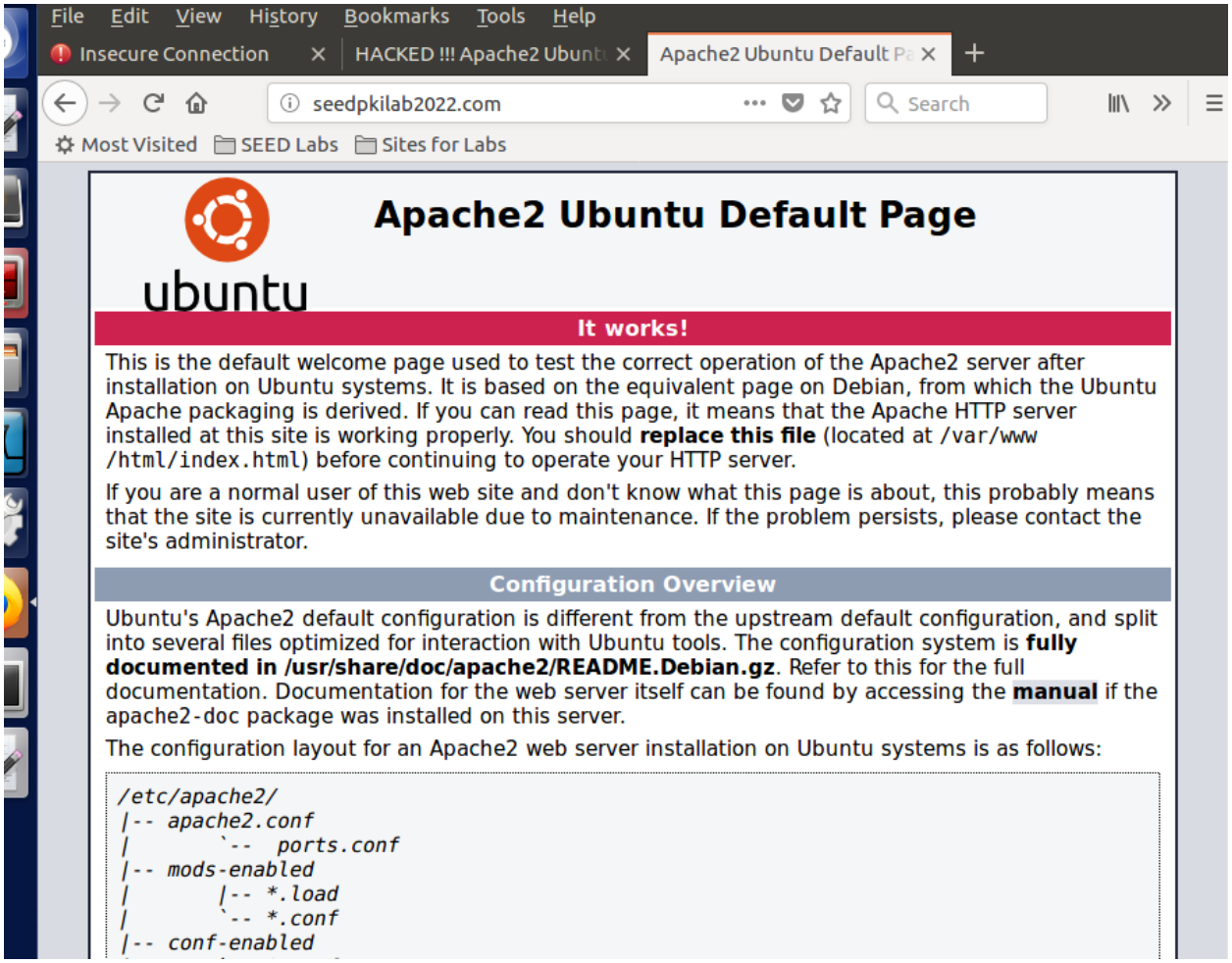

File Machine View Input Devices Help
*hosts (/etc) - gedit
Open
*hosts
/etc
127.0.0.1 localhost
127.0.1.1 VM
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 User
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
#192.168.99.7 SEEDPKILab2022.com
#127.0.0.1 SEEDPKILab2022.com
#127.0.0.1 bankofamerica
#127.0.0.1 bankamerica
#127.0.0.1 twitter.com
192.168.99.7 twitter.com



This is when I tried with apache index.html file




SEEDUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
HACKED !!! Apache2 Ubuntu Default Page: It works - Mozilla Firefox
2:06 PM
File Edit View History Bookmarks Tools Help
Insecure Connection
HACKED !!! Apache2 Ubunt
seedpkilab2022.com
Search
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 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/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 probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.


File Edit View History Bookmarks Tools Help
Insecure Connection
HACKED !!! Apache2 Ubunt
Apache2 Ubuntu Default Pa
seedpkilab2022.com
Search
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 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www /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 probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled