INTERNET SECURITY LAB

PKI LAB

Akarsh Shetty Umesh Mudelkadi
317752264

Task 1: Becoming a Certificate Authority(CA)

Made the needful changes in configuration file.

```
[04/19/2019 22:43]Mudelkadi@VM1:~$ openssl req -new -x509 -keyout ca.key -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 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]:NY
Locality Name (eg, city) []:Syracuse
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ISec
Organizational Unit Name (eg, section) []:Spring
Common Name (e.g. server FQDN or YOUR name) []:Akarsh
Email Address []:ashettyu@syr.edu
[04/19/2019 22:44]Mudelkadi@VM1:~$
```

Ca.key file:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIGnvXXdYyQHQCAggA
MBQGCCqGSIb3DQMHBAiD5peVEbWGiwSCBMhJaukfiY8qAkwIiS7ICPBuEfIAU38g
3zAzukQU378vqXqH210SzY96sdg9pL1IqWZmT7yziLqy08JVRFG2z2ps8qO2Y+Oj
NzRjrtcD0hDL93/85QNlROszqgSz2i9r8hpnY+ZRA9YkDNR/q/eM2WZ6riuIGzv+
Ilc8NlkqR6zIFUsN+4RslM8KeNGoVp7cCso8UgJ3Ibgw4z3I2PP4oa8sTJ2fAMnK
FvGD3YOPOGABiTcz6gl6NwnJj/zSz08AKrzX/umQ0gN3aMct7HCdHGpw7+vreXyj
eDaVFFRNEeiy61wJYDoWfo1o0f6OtjAVQ8/C/YfxjUw7WT/wmkMu45ZqrYI+qlF/
j+RtpVpATRybl/TIdipwMcvYymmlEbmIvZfb0LeOCvH3oZzJambz0wMnvUiYGy05
3ZFSJ2xF7DSTbo/9HDfYqItuAd1v7/6vAiyWFM1RZx4pIcmDBY9snhcbfblsno05
grHszlTxonBRxOWrKRmrOydmJURDMto4Lhmky2AdBqMxRmb/mtSZ633BhFGXLyu1
KD+Uy7/x6u2nAAcoxpRel4MuN8VUPVtmlDs1fALyJqvR6ojPKqQWCM1Xlb+0aeYp
Pr4M9GPo0Ix92r2B0r71BQedjGBJP6Ag7jzduhkKeB20EpgJDm7rBTdbOXsnGVOI
Eboxk4O2C84brMABC5X/nsAAhXZvYMbP2ncrw1oxy6/fzkkBwHtQ1IpW+Zy0h6BE
PCkMvWQdvjYgaRYcC9UP2Sutug8Cx12AFtMrOFwniNFFdAFKKVS9sj7GlZ7FuCm2
/WwNeHw0U/MY2y1wdK2BI01NK0EMfSFgt3QoZvVd66aVYcqfnp1N5Z6cMFQP8G1z
zq+phkR0S81lTBUeWuJRVBSPVqkw17UIhMpvB64FwFJY3LErbjqHlzE8wQPabePq
GqY7gvpQa7j7la9BQ2oyl61SkQpB01GWzqX+egKYnUKorlubZMy6qyL1gRCsiXeU
/AVvS4qe6YT4nxqN6ZJvP0/tcLRAnDDtcQttbtKLmc0/Mw96Q4hHAClEQZ9rHm4l
Gc9DaTatUb9GzVQfT3uOOfI1LKefAdZoKLIhGzwWuNcuqafdcUEJF4dwg5afE3Fs
J92W2r20dgTsmxGN9qTaxdGpNOlTtmo7wnZvUHV5rWl4dM5ovs+gL2MUyRfC43Kc
6NZxmx2dI93Y50+JchifZa00DLoHNGy45epFCL4/AMxCzosMyVPeuNnRNRRsGZ7I
OUBvwNhi6MifG7gWiTBUxs/Ufmw5yPre4zxmXQ38Mrp0SYzL9k4+sgcUCPpDVdOH
/ZZtmZWcZBvaN6x+Nr5Sk80Ts63oHRlY44nSN2LoLzQqS8qAptc06EpdIU1qb4Cd
z469GQjoJsEXbjHi9cTxhrpbZZfyjS5ZTtXkhW72IhroyltUTW8+InNnFl2N5yUR
AT+y0r/9I6f8Wc5PCmxck+bQ7i+44Kp01qKjENgQEESqxTxqiwNRDwCkb5vFEJMr
rkSMo3W4zWy3CJLui73Ya45818u441XYqjn40dEupYfxD7zjrXX6UIh4KzouRFj4
diqaD8V+09P9P3pP+bvL3H7Np1dWhOklF6oyIuLfjzvLSJW4oNWOYpaEKB89U+81
```

Ca.crt file

```
-----BEGIN CERTIFICATE-----
MIID0TCCArmgAwIBAgIJAK7il8scwS3gMA0GCSqGSIb3DQEBCwUAMH8xCzAJBgNV
BAYTAlVTMQswCQYDVQQIDAJOWTERMA8GA1UEBwwIU3lyYWN1c2UxDTALBgNVBAoM
BElTZWMxDzANBgNVBAsMBlNwcmluZzEPMA0GA1UEAwwGQWthcnNoMR8wHQYJKoZI
hvcNAQkBFhBhc2hldHR5dUBzeXIuZWR1MB4XDTE5MDQyMDAyNDQ0NVoXDTE5MDUy
MDAyNDQ0NVowfzELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk5ZMREwDwYDVQQHDAhT
eXJhY3VzZTENMAsGA1UECgwESVNlYzEPMA0GA1UECwwGU3ByaW5nMQ8wDQYDVQQD
DAZBa2Fyc2gxHzAdBgkqhkiG9w0BCQEWEGFzaGV0dHl1QHN5ci5lZHUwggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDyJ+BRe8kT2FGS7yHKsvu16MSiXoz0
Cc6QQGUUm/wQUaDT33elb7MtowfP9bwHiVHakpeXI00un14ijsp80OBgCr7SxFCX
k5KwAeBiFY8pExaZEiH1ZmK09lBO+4X3WLEPLpjK7vgGiHULUQqpAr5bZYYVm4Mg
XursfNtQ1Dq29E9YhEi9/uSAjQxGX1FgRiRZ4YMf87z11t0Hg4FrkJ86sBEVF+fy
d1Lg9U7dOFQ7N9V9yySAmWT+zIwLlu807GP6atl5PT38J8bOofTopGJqAYHnzMy/
L/k0A+ezvVrlZGJ0azxAZfp2E2yLaYbZWX9OTDl1IONk1DOSydIDM/aNAgMBAAGj
UDBOMB0GA1UdDgQWBBTtQygkw5sM58F9NHeXH6/IuzxkxDAfBgNVHSMEGDAWgBTt
Qygkw5sM58F9NHeXH6/IuzxkxDAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUA
A4IBAQBJk3yF64UAVPyHuMcm4KpljWiSKhGkVHG9wdzB72eM+kp+bPKdM3eJDWyg
7U54R/eBw/lhvwOuSmjFLQzDZeCBvQDY3Gl4e+A7WtJGKKTdPHJ7HbBpb59cpbVl
HEeGcWi1cXM/cnz1tHuyRbVcbHmgnmW/QMZprI+ODo6SuFo3WvXaXooulpp9ot1i
W/LwbWIROT58UiPOFKGtra7IjZmeHIIwFiO7l9Qv1jlgRZjX7/fZBGlZEZiMgIpO
wK7CVptFOalQBhEwQqbutG2rv51IVh5CaxdRlv2IdcnS10R6IfxRNtgckfp1Rnu4
Snj4r7jtToRyn77Ur6o6vLFcMg9G
-----END CERTIFICATE-----
```

Task 2: Creating a certificate for SEEDPKILab2018.com.

```
[04/19/2019 22:47]Mudelkadi@VM1:~$ 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:
[04/19/2019 22:47]Mudelkadi@VM1:~$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
    00:9e:6a:ea:68:f0:e4:c6:fd:08:3c:c4:32:5c:4e:
    25:ce:27:42:40:b9:35:70:c4:7f:1c:d7:fb:46:f0:
    87:cf:89:5e:a3:1c:3d:37:5d:fe:02:9d:33:c6:e7:
    e4:f7:c5:b8:68:df:b4:c6:07:62:47:e9:9c:c9:d7:
    9b:3b:fe:d8:35:63:59:64:8a:13:4b:95:b2:5b:7a:
    c0:b5:ef:f2:77:fd:fb:de:b6:56:04:6b:65:02:c9:
    7b:8b:4f:be:36:6b:4a:71:38:01:bc:95:51:9d:2c:
    f5:ae:9e:50:46:39:9a:36:ed:5b:a3:3e:21:9b:4d:
    12:38:c0:28:01:7d:84:b0:53
publicExponent: 65537 (0x10001)
privateExponent:
    64:c0:98:df:2f:3e:41:74:52:55:db:6e:30:18:9f:
    a0:b2:85:47:86:c3:7f:39:e9:a2:1d:97:39:71:e6:
```

Observation: Generated public/private key pair using the following command:

openssl genrsa -aes128 -out server.key 1024

The keys are stored in server.key file.

```
[04/19/2019 23:14]Mudelkadi@VM1:~/PKI$ openssl req -new -key server.key -out serv
er.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]:NY
Locality Name (eg, city) []:Syracuse
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ISec
Organizational Unit Name (eg, section) []:Spring
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2018.com
Email Address []:Liverpool.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Liverpool
An optional company name []:Kop
[04/19/2019 23:18]Mudelkadi@VM1:~/PKI$
```

Observation: The above picture shows the generation of Certificate signing request.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4096 (0x1000)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=NY, L=Syracuse, O=ISec, OU=Spring, CN=Akarsh/emailAddres
s=ashettyu@syr.edu
        Validity
            Not Before: Apr 20 03:19:36 2019 GMT
            Not After : Apr 19 03:19:36 2020 GMT
        Subject: C=US, ST=NY, O=ISec, OU=Spring, CN=SEEDPKILab2018.com/emailAddre
ss=Liverpool.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:bf:36:a2:4c:45:24:e4:35:c8:ba:e9:ae:24:f8:
                    a7:6d:0f:bd:7f:ba:1b:79:34:b9:80:95:22:e2:3c:
                    42:27:a3:22:b9:23:b7:c0:af:38:93:4c:11:5a:f3:
                    0d:8c:a6:85:6d:85:71:c0:71:06:51:9d:70:5e:d9:
                    3b:23:4c:78:0a:91:4d:be:ab:7b:74:ea:81:ef:11:
                    02:57:fa:61:5f:76:b2:03:41:ae:69:85:91:57:e0:
                    37:6d:ef:dc:e1:04:dc:4b:42:f1:3e:3e:85:6e:46:
                    ba:e0:57:ca:87:e9:08:6d:78:bf:0c:ea:01:82:61:
                    51:01:aa:47:30:36:48:d7:f9
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
"server.crt" 71L, 3833C                                       1,1           Top
```

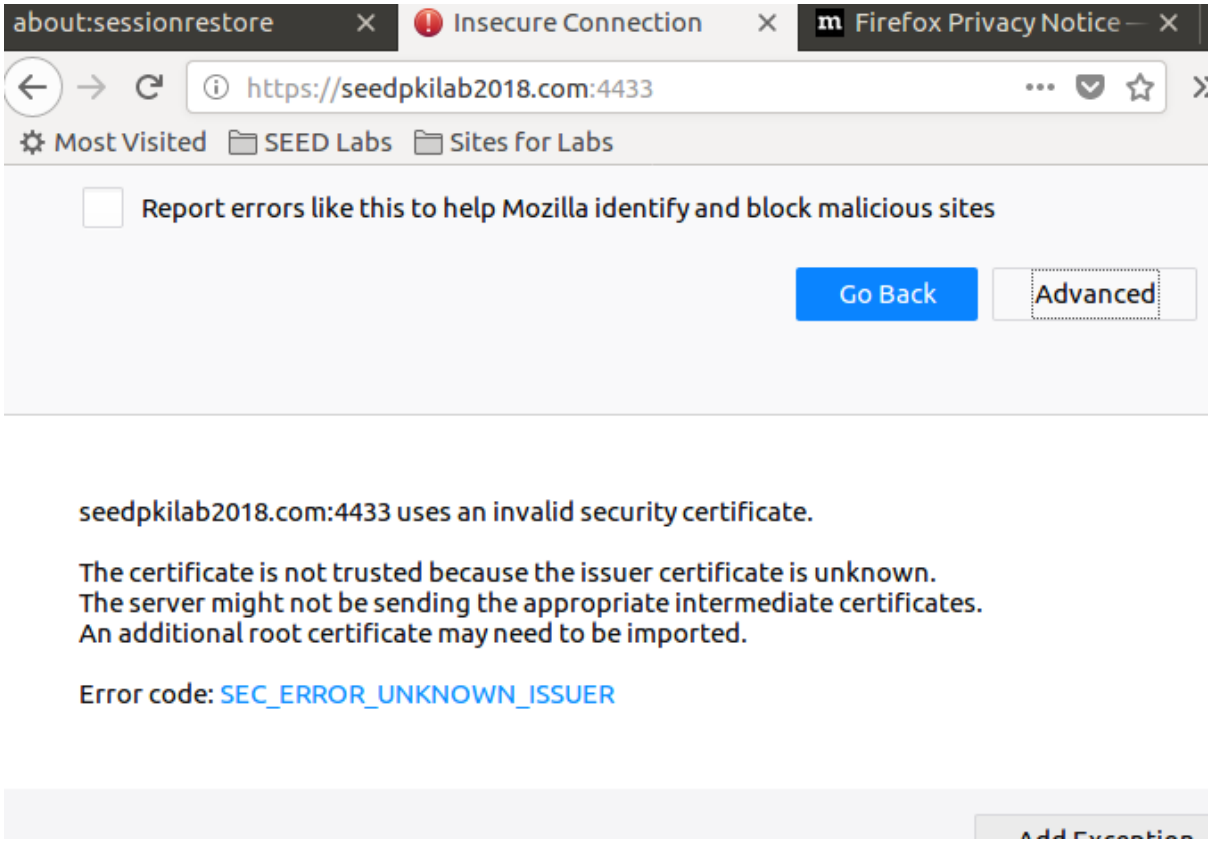
Observation: The CSR file is signed by CA's signature and its converted into .crt file by using command: openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf. The above picture is of the generated certificate.

Task 3: Deploying certificate in an HTTP Web Server.

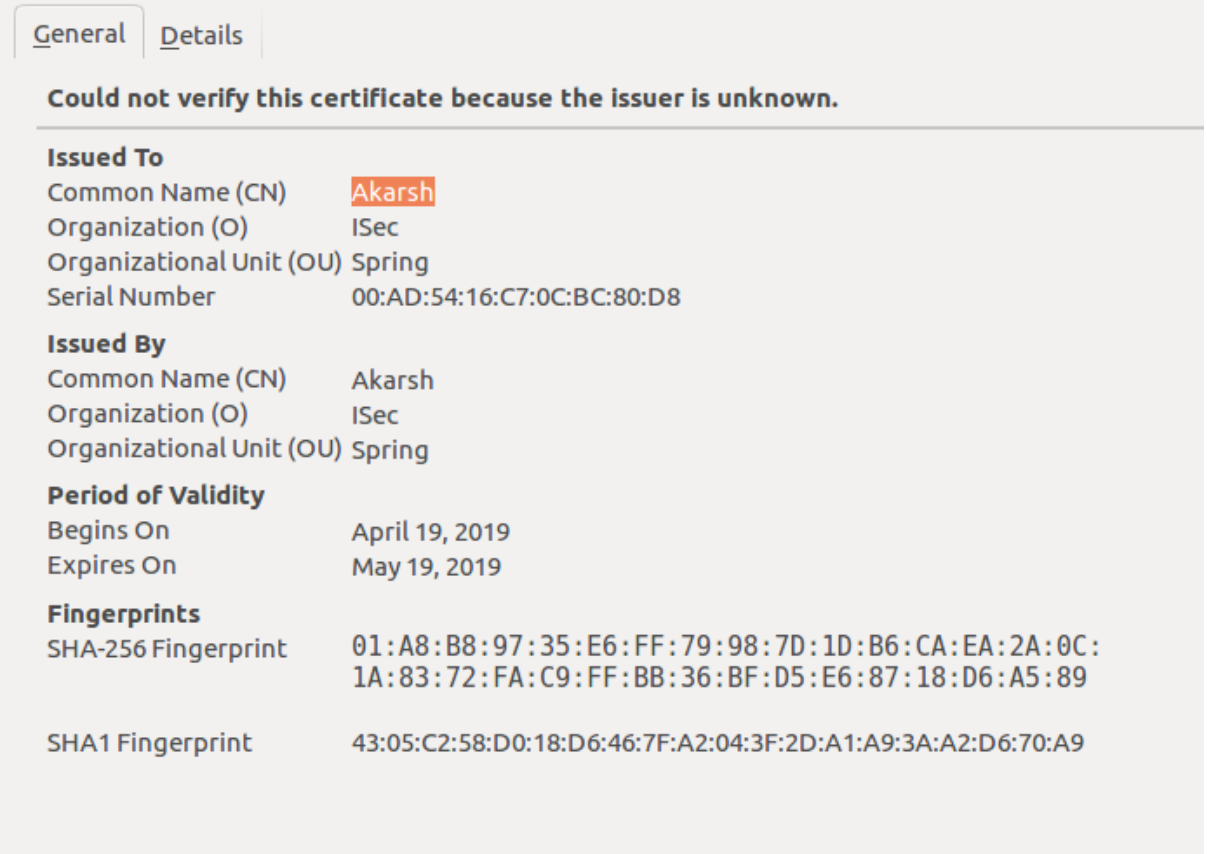
- Added SEEDPKILab2018.com in /etc/hosts file with IP 127.0.0.1.

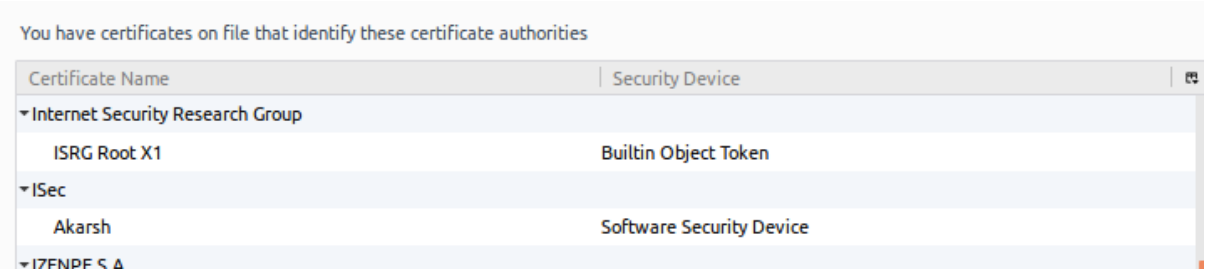
```
[04/19/2019 23:24]Mudelkadi@VM1:~/PKI$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

Started SEEDPKILab2018.com web server with the certificate generated in the previous task.

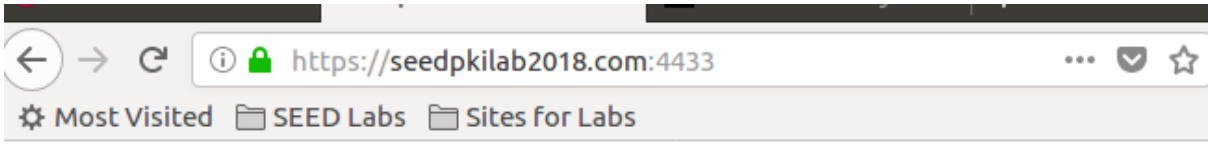


Observation: We get the above error when we try to access the web server we created. We get the error because the browser dosen't have the CA certificate for which our server's certificate is signed. Thus it throws an invalid security certificate error.





Observation: The above pictures shows the process of getting the browser accept our CA certificate.

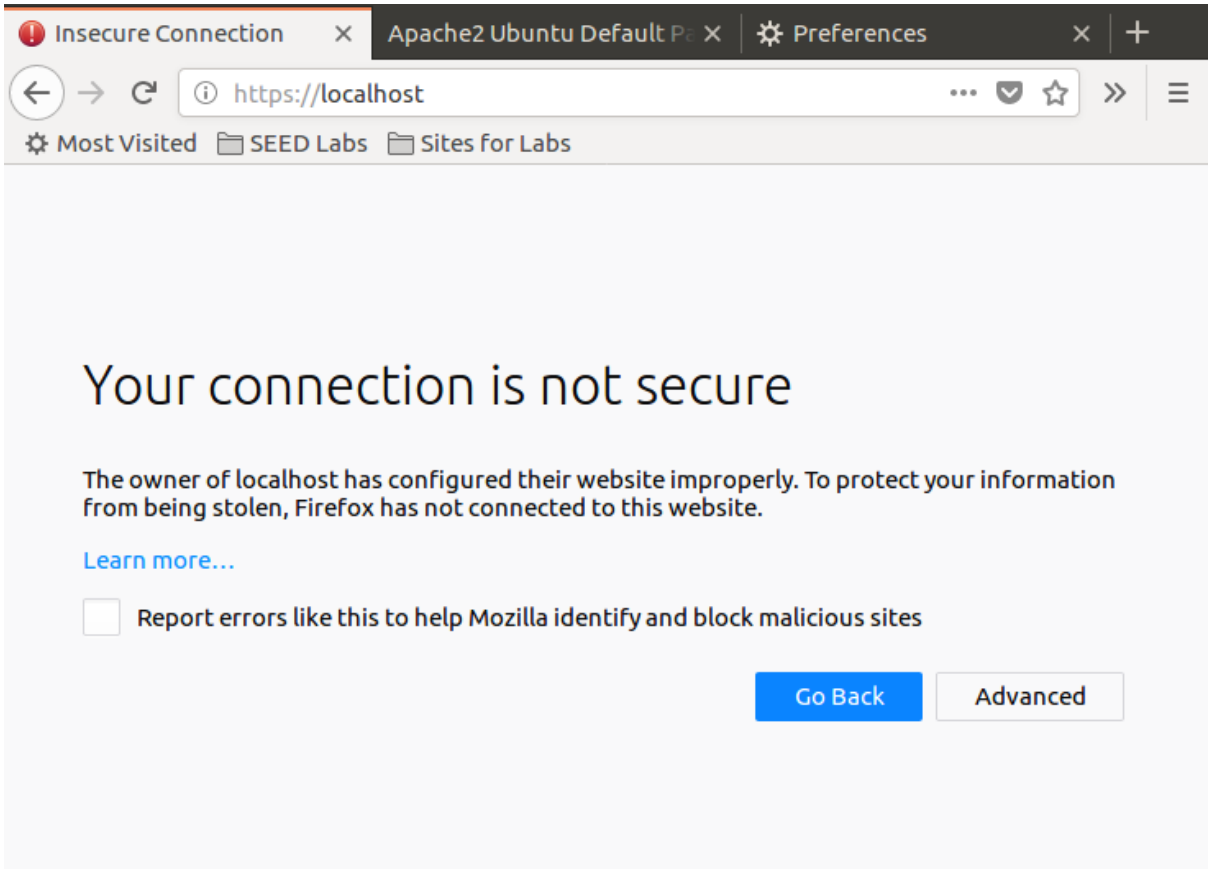



```
s_server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384   TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA      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
```

Observation: When our CA certificate is added to our browser, we get the above output. Therefore our web browser trusts its certificate.

```
[04/22/2019 22:47]Mudelkadi@VM1:~/PKI$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
unable to load server certificate private key file
3070686912:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong tag:tasn_d
ec.c:1197:
3070686912:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nested asn1 err
or:tasn_dec.c:374:Type=RSA
3070686912:error:04093004:rsa routines:OLD_RSA_PRIV_DECODE:RSA lib:rsa_ameth.c:11
9:
3070686912:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong tag:tasn_d
ec.c:1197:
3070686912:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nested asn1 err
or:tasn_dec.c:374:Type=PKCS8_PRIV_KEY_INFO
3070686912:error:0907B00D:PEM routines:PEM_READ_BIO_PRIVATEKEY:ASN1 lib:pem_pkey.
c:141:
```

Observation: When we try to modify a single byte in server.pem we get a decrypt error thus couldn't run the apache server.



Observation: We are unable to connect to https://localhost:443 as SEEDPKILab2018.com is used as domain name in the certificate not localhost.

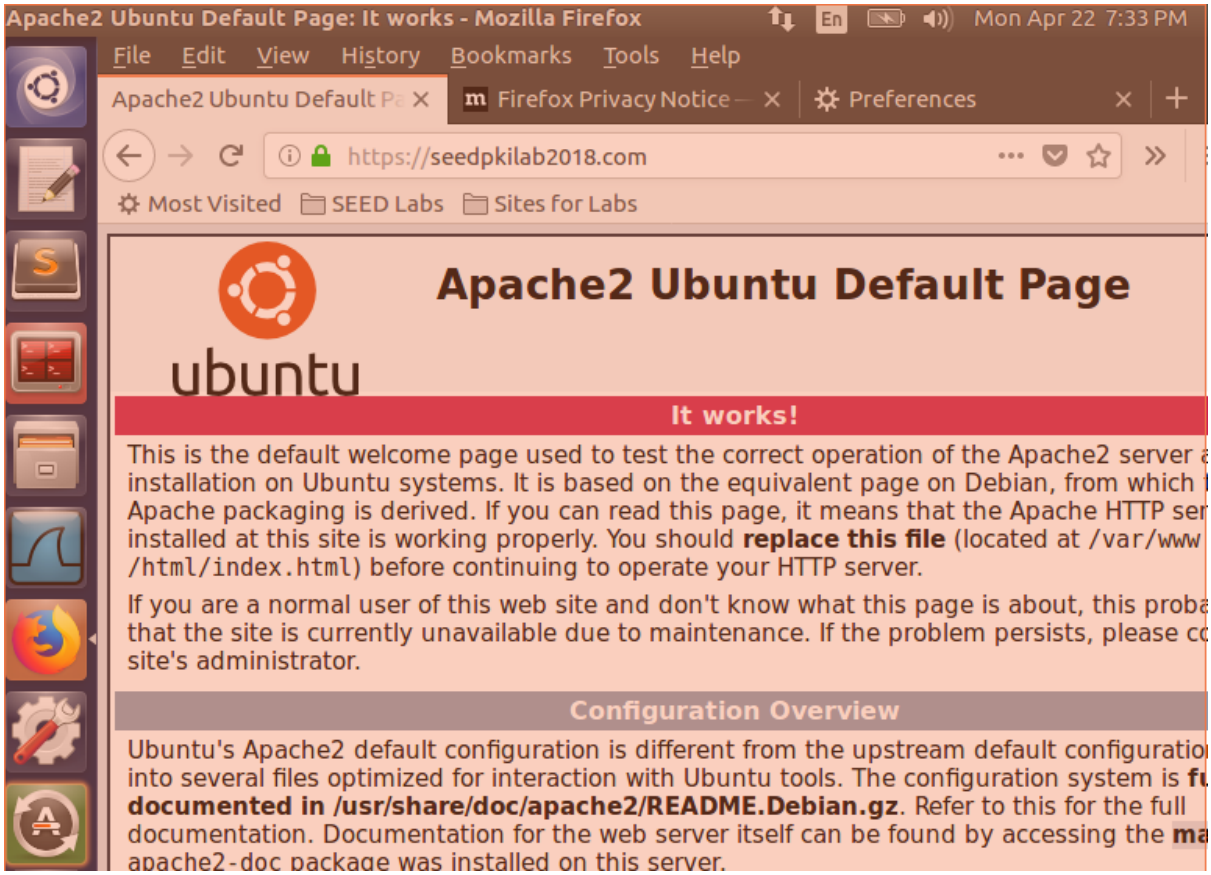
Task 4: Deploying Certificate in an Apache-Based HTTPS website.

```
<VirtualHost *:443>
        ServerName SEEDPKILab2018.com
        DocumentRoot /var/www/html
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile      /home/seed/PKI/server.crt
        SSLCertificateKeyFile   /home/seed/PKI/server.key
</VirtualHost>
```

Observation: Configured the default-ssl conf so that apache knows where directory where the websites certificate and key is known.

```
315  sudo vi default-ssl.conf
316  sudo apachectl configtest
317  sudo a2enmod ssl
318  sudo a2ensite default-ssl
319  sudo service apache2 restart
```

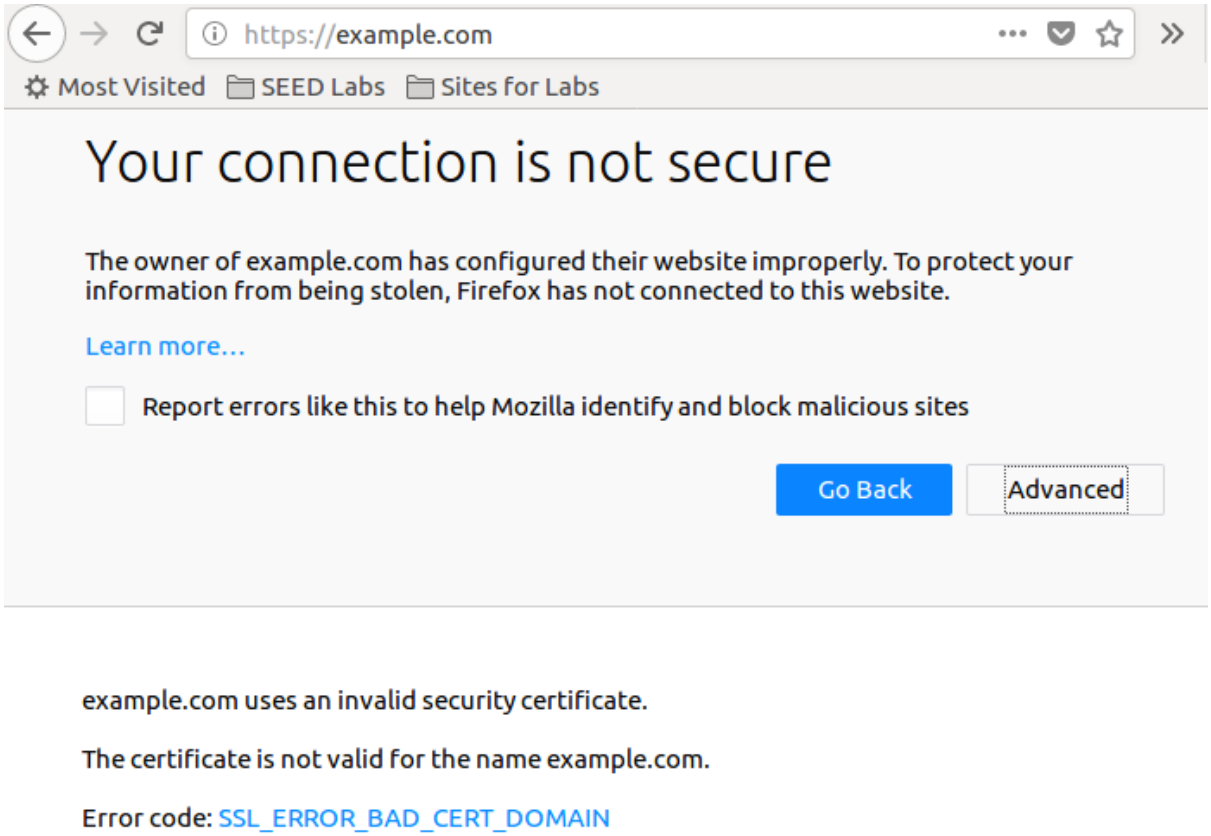
Observation: Executed the above last 4 commands to enable the SSL. The private key used for encrypting is given when prompted by restart command.



Observation: Successful http request to our web server.

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

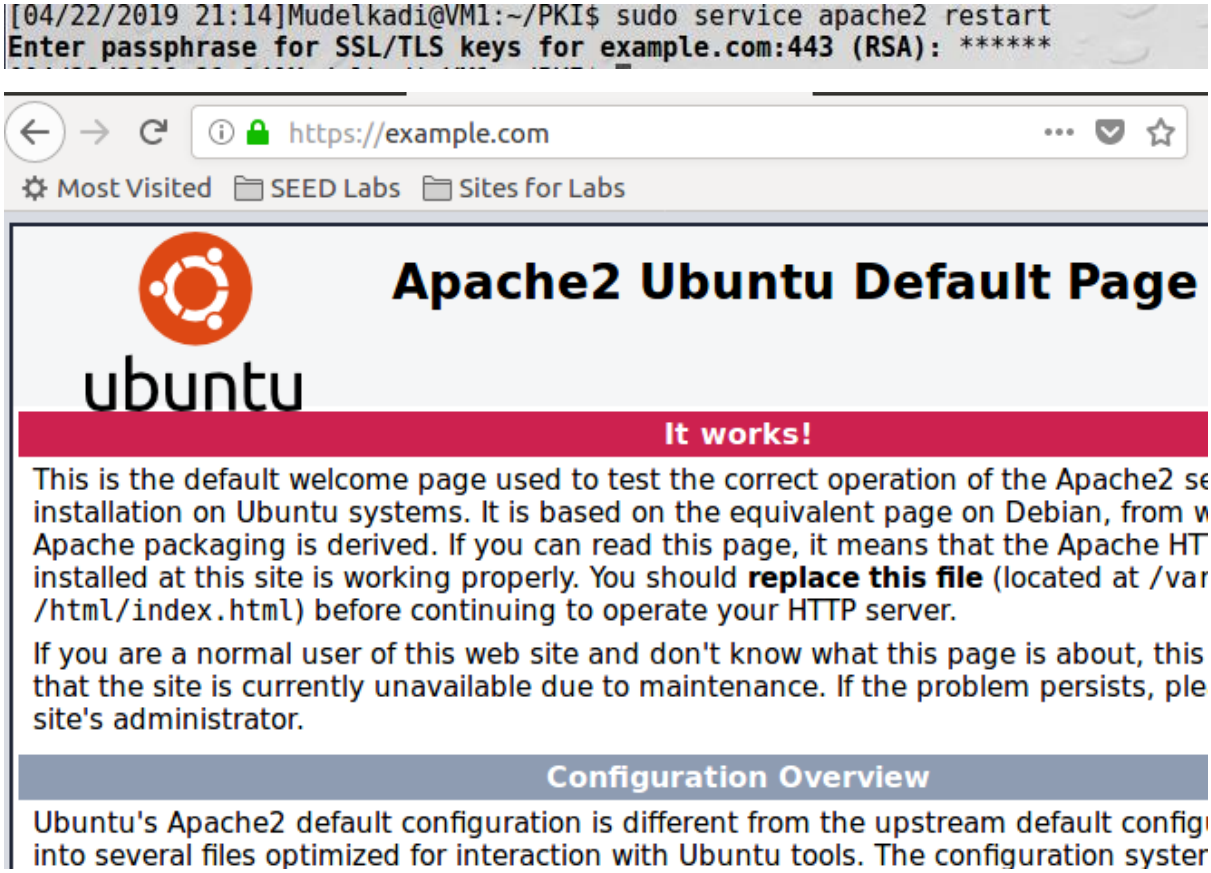
```
<VirtualHost *:443>
        ServerName example.com
        DocumentRoot /var/www/html
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile      /home/seed/PKI/server.crt
        SSLCertificateKeyFile   /home/seed/PKI/server.key
</VirtualHost>
```



Observation: Since the certificate is generated for SEEDPKILab2018, the above error is shown when we try to access example.com as it uses certificate and key of SEEDPKI and the domain(example.com) is not given while its certificate creation.

Task 6:

```
[04/22/2019 21:12]Mudelkadi@VM1:~/PKI$ openssl ca -in exmpserver.csr -out exmpser
ver.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: Apr 23 01:12:27 2019 GMT
            Not After : Apr 22 01:12:27 2020 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NY
            organizationName          = ISec
            organizationalUnitName    = ISec
            commonName                = example.com
            emailAddress              = bank.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
```



Observation: Since now we assume that we know the private key of CA, I created a server certificate for example.com by following steps in Task 3. Added example.com in /etc/hosts file. We already modified the default-ssl conf file for apache in task 5. We are successful in generating http request to example.com.