UNC CHARLOTTE

College of Computing and Informatics

Department of Software and Information Systems

Enterprise and Infrastructure Protection
ITIS 6230

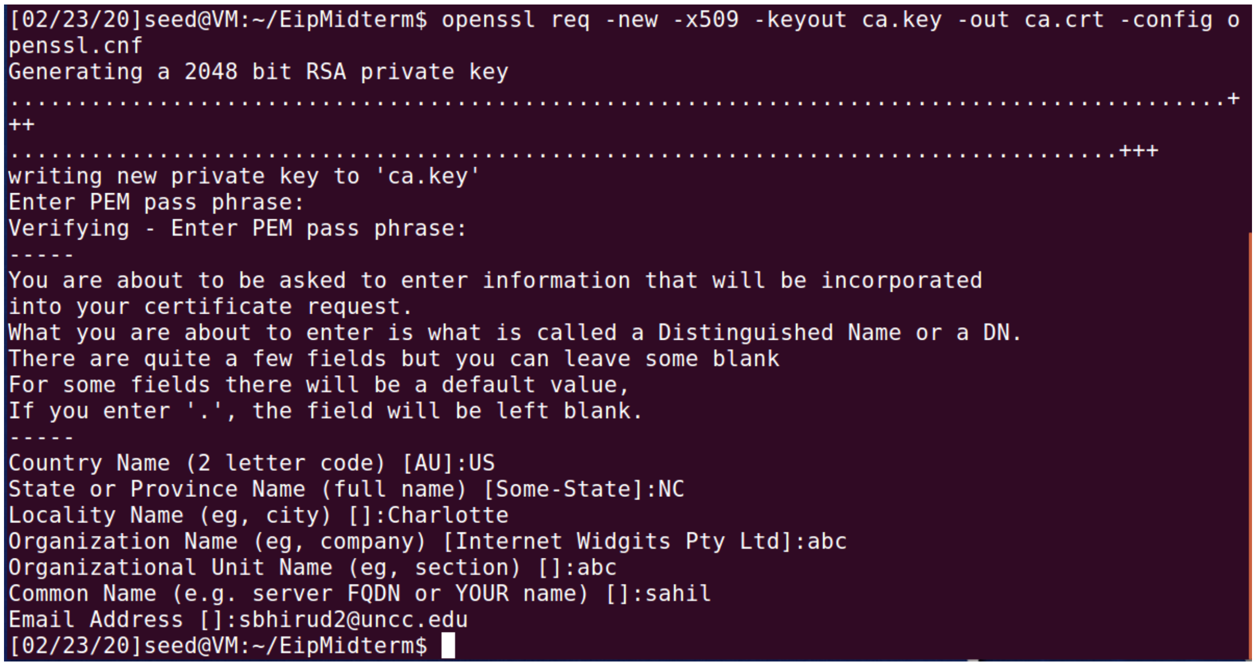
MIDTERM PROJECT

PKI LAB

SAHIL BHIRUD - 801138029

This lab was focused on providing hands-on experience on Public Key Infrastructure (PKI). It provided a better understanding of how PKI works, how it is used to protect the Web and how Man-In-The-Middle Attacks can be defeated by PKI.

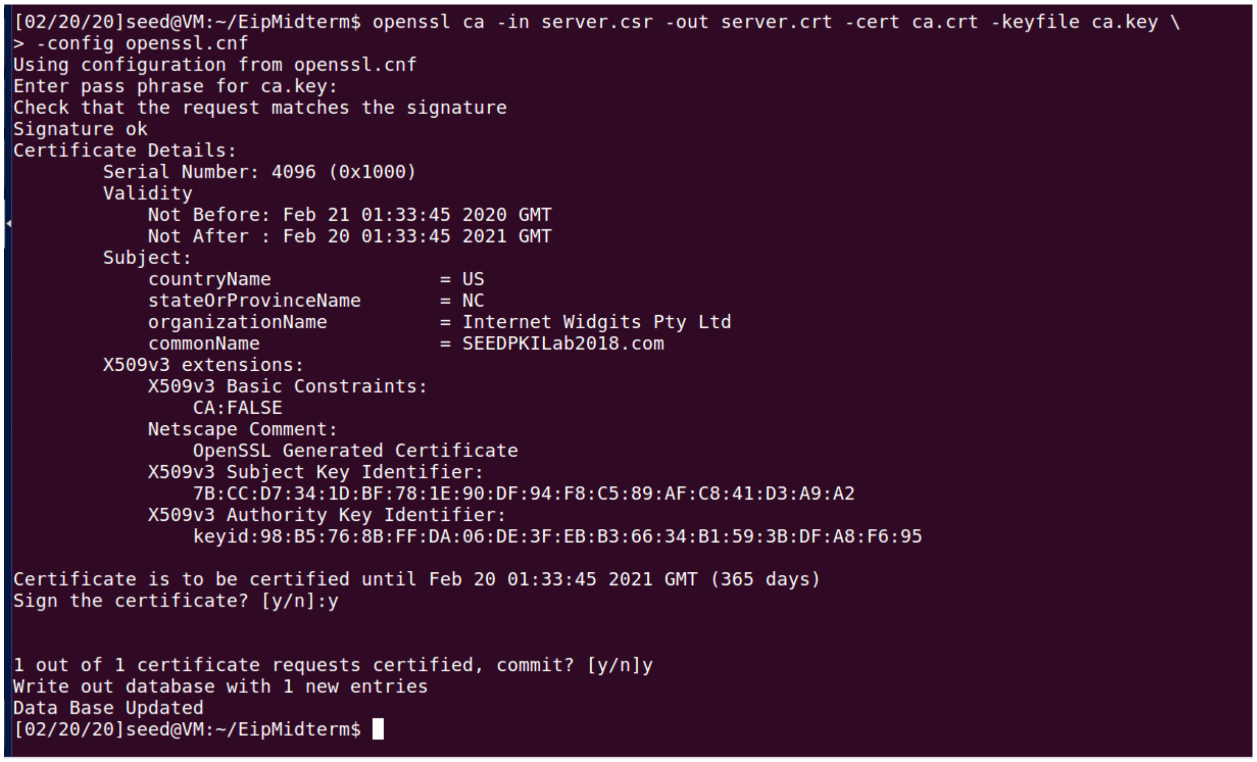
In **task 1**, I had to become a Certification Authority (CA). The CA uses the ***openssl.conf*** file to issue certificates for other servers. I had to create the respective directories as mentioned in the ***openssl.conf*** file and then run the command ***openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf*** which created the CA's key and public key certificate.



```
[02/23/20]seed@VM:~/EipMidterm$ openssl req -new -x509 -keyout ca.key -out ca.crt -config o
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]:NC
Locality Name (eg, city) []:Charlotte
Organization Name (eg, company) [Internet Widgits Pty Ltd]:abc
Organizational Unit Name (eg, section) []:abc
Common Name (e.g. server FQDN or YOUR name) []:sahil
Email Address []:sbhirud2@uncc.edu
[02/23/20]seed@VM:~/EipMidterm$ 
```

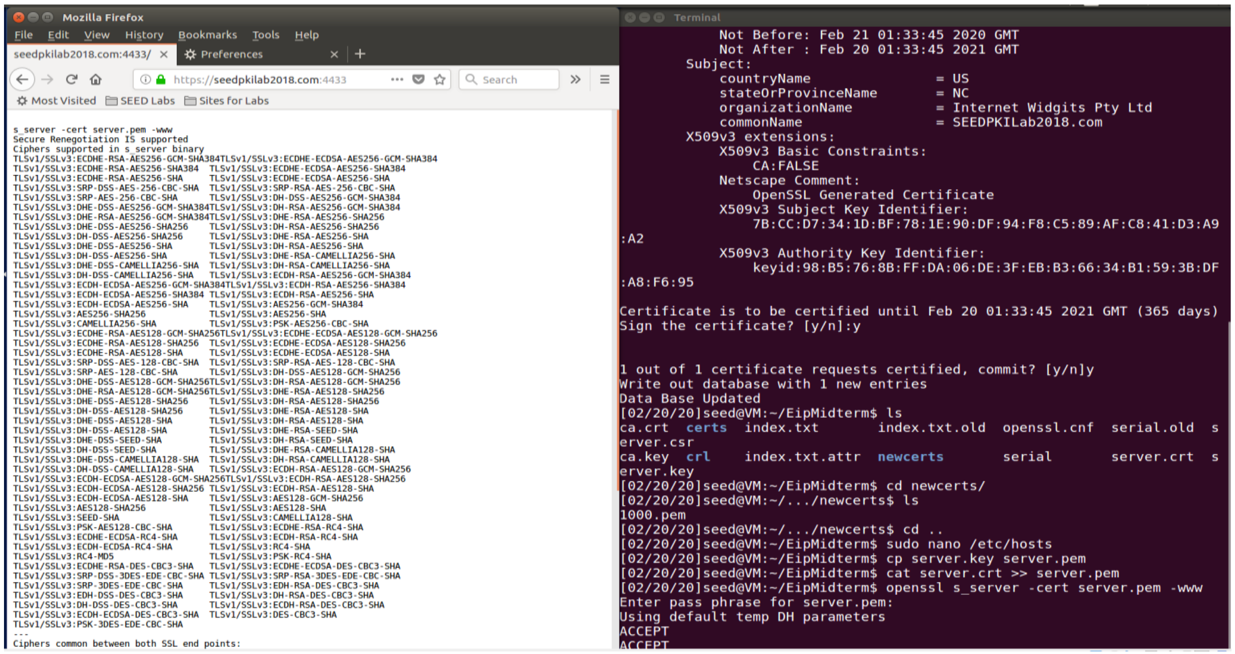
In **task 2**, I created a certificate for **SEEDPKILab2018.com** by performing the following step as mentioned in the SEED Documentation:

1. Generated the public/private key pair for it.
2. Generated a Certificate Signing Request (CSR) – It is a document which includes company's public key which is sent to the CA who generates a certificate for it once it authenticates the server.
3. Generated the certificate.



```
[02/20/20]seed@VM:~/EipMidterm$ 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: Feb 21 01:33:45 2020 GMT
            Not After : Feb 20 01:33:45 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NC
            organizationName          = Internet Widgits Pty Ltd
            commonName                = SEEDPKILab2018.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                7B:CC:D7:34:1D:BF:78:1E:90:DF:94:F8:C5:89:AF:C8:41:D3:A9:A2
            X509v3 Authority Key Identifier:
                keyid:98:B5:76:8B:FF:DA:06:DE:3F:EB:B3:66:34:B1:59:3B:DF:A8:F6:95

Certificate is to be certified until Feb 20 01:33:45 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
[02/20/20]seed@VM:~/EipMidterm$
```

Next, in **task 3**, I deployed the certificate in Apache Web Server, which is openssl's built in web server. Firstly, I had to configure the DNS to recognize the name of our website by editing the ***/etc/hosts*** file. Then, I configured the web server for the certificate generated in Task 2. While accessing the URL, I received an error form the browser saying Invalid Security Certificate. Next, I manually added the CA's certificate in the browser. Finally, I tested the website ***https://SEEDPKILab2018.com:4433*** where 4433 is the port number on which the server is listening. The website displayed a list of ciphers supported by openssl.

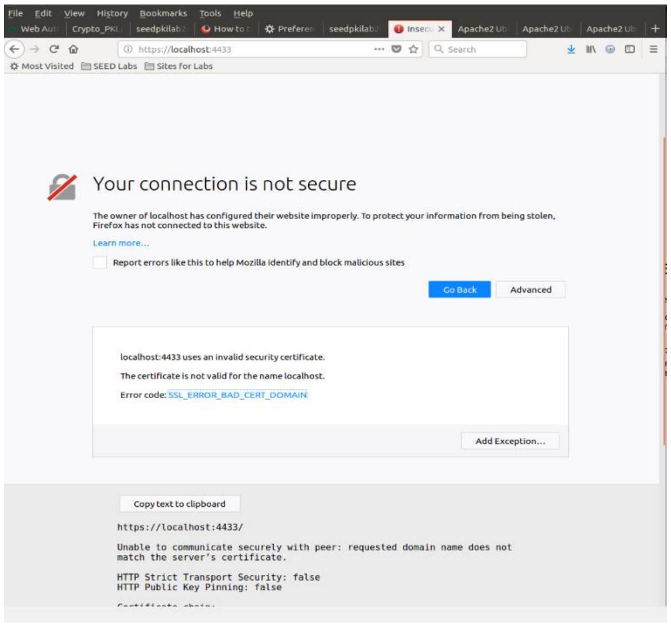


```
s_server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384  TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA     TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA
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
TLSv1/SSLv3:RC4-MD5                  TLSv1/SSLv3:PSK-RC4-SHA
TLSv1/SSLv3:ECDHE-RSA-DES-CBC3-SHA   TLSv1/SSLv3:ECDHE-ECDSA-DES-CBC3-SHA
TLSv1/SSLv3:SRP-DSS-3DES-EDE-CBC-SHA TLSv1/SSLv3:SRP-RSA-3DES-EDE-CBC-SHA
TLSv1/SSLv3:SRP-3DES-EDE-CBC-SHA     TLSv1/SSLv3:EDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:EDH-DSS-DES-CBC3-SHA     TLSv1/SSLv3:DH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:DH-DSS-DES-CBC3-SHA      TLSv1/SSLv3:ECDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:ECDH-ECDSA-DES-CBC3-SHA  TLSv1/SSLv3:DES-CBC3-SHA
TLSv1/SSLv3:PSK-3DES-EDE-CBC-SHA
---
Ciphers common between both SSL end points:
```

```
            Not Before: Feb 21 01:33:45 2020 GMT
            Not After : Feb 20 01:33:45 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NC
            organizationName          = Internet Widgits Pty Ltd
            commonName                = SEEDPKILab2018.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                7B:CC:D7:34:1D:BF:78:1E:90:DF:94:F8:C5:89:AF:C8:41:D3:A9
:A2
            X509v3 Authority Key Identifier:
                keyid:98:B5:76:8B:FF:DA:06:DE:3F:EB:B3:66:34:B1:59:3B:DF
:A8:F6:95

Certificate is to be certified until Feb 20 01:33:45 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
[02/20/20]seed@VM:~/EipMidterm$ ls
ca.crt  certs  index.txt       index.txt.old  openssl.cnf  serial.old   s
erver.csr
ca.key  crl    index.txt.attr  newcerts       serial       server.crt   s
erver.key
[02/20/20]seed@VM:~/EipMidterm$ cd newcerts/
[02/20/20]seed@VM:~/.../newcerts$ ls
1000.pem
[02/20/20]seed@VM:~/.../newcerts$ cd ..
[02/20/20]seed@VM:~/EipMidterm$ sudo nano /etc/hosts
[02/20/20]seed@VM:~/EipMidterm$ cp server.key server.pem
[02/20/20]seed@VM:~/EipMidterm$ cat server.crt >> server.pem
[02/20/20]seed@VM:~/EipMidterm$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
```

When a single byte of *server.pem* is changed, the browser shows an insecure connection warning as the certificate has changed and the browser does not have this certificate in its trusted list of certificates.

Also, if we try to access https://localhost:4433, the browser is not able to communicate securely with the peer since the requested domain name does not match the server's certificate.



Your connection is not secure

The owner of localhost 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

localhost:4433 uses an invalid security certificate.

The certificate is not valid for the name localhost.

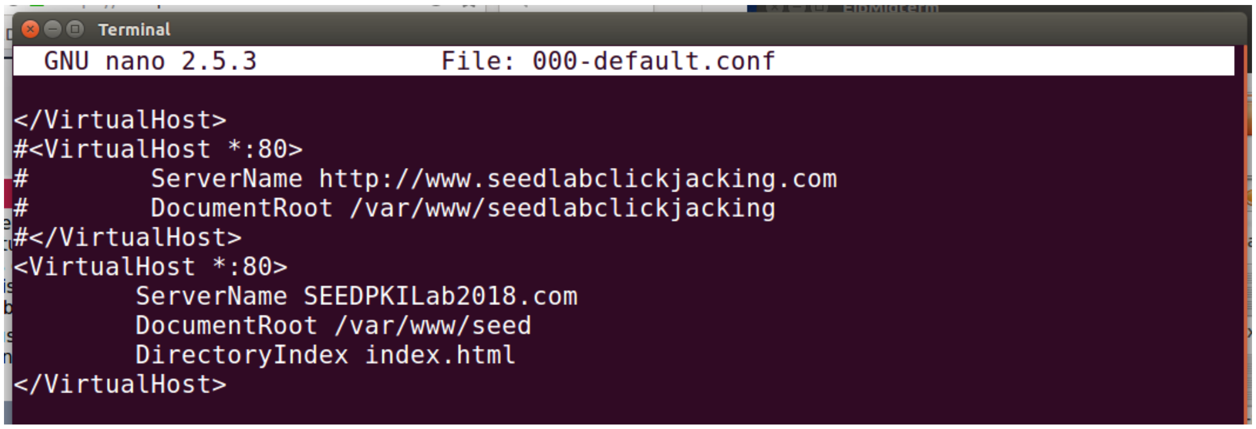
Error code: SSL_ERROR_BAD_CERT_DOMAIN

```
https://localhost:4433/

Unable to communicate securely with peer: requested domain name does not
match the server's certificate.

HTTP Strict Transport Security: false
HTTP Public Key Pinning: false
```

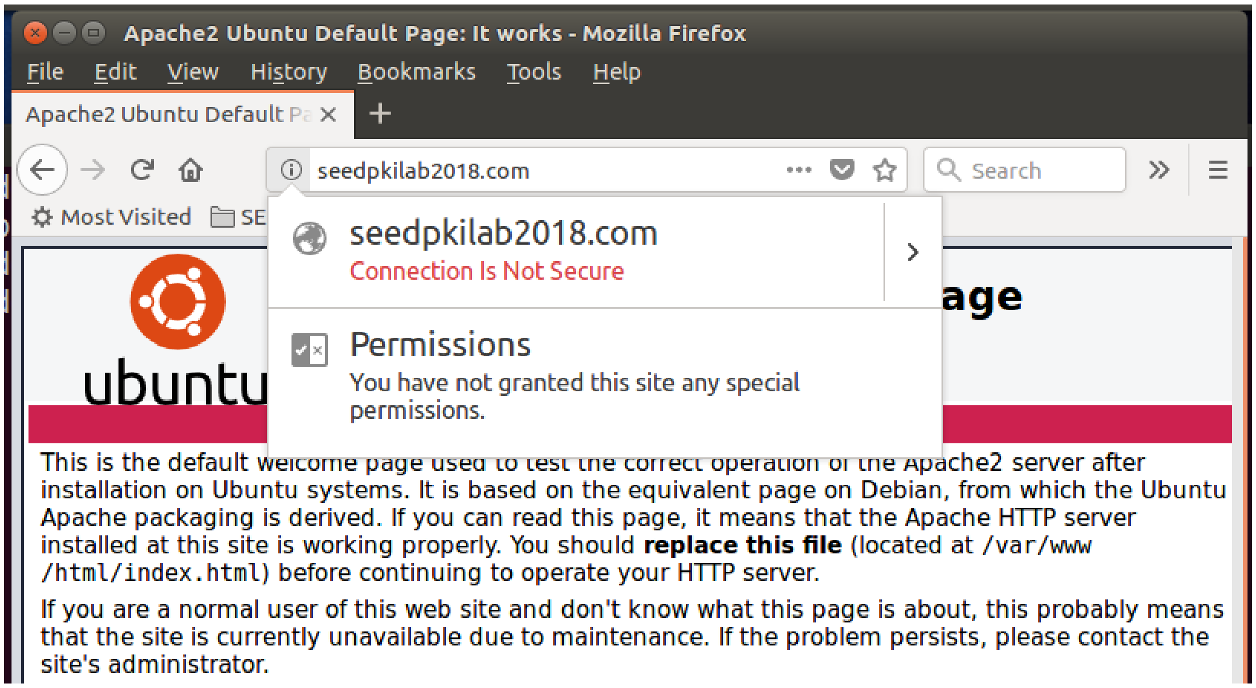
In Task 4, I set up an HTTPS website on the Apache server. I just needed to configure the Apache server to locate the private key and certificate. This configuration (shown below) was done in

/etc/apache2/sites-available/000-default.conf for HTTP website and ***/etc/apache2/sites-available/default-ssl.conf*** for HTTPS websites.

After configuring the server, I tested the configuration file for errors, enabled SSL module, enabled the site and restarted Apache server.



```
 GNU nano 2.5.3             File: 000-default.conf

</VirtualHost>
#<VirtualHost *:80>
#        ServerName http://www.seedlabclickjacking.com
#        DocumentRoot /var/www/seedlabclickjacking
#</VirtualHost>
<VirtualHost *:80>
        ServerName SEEDPKILab2018.com
        DocumentRoot /var/www/seed
        DirectoryIndex index.html
</VirtualHost>
```

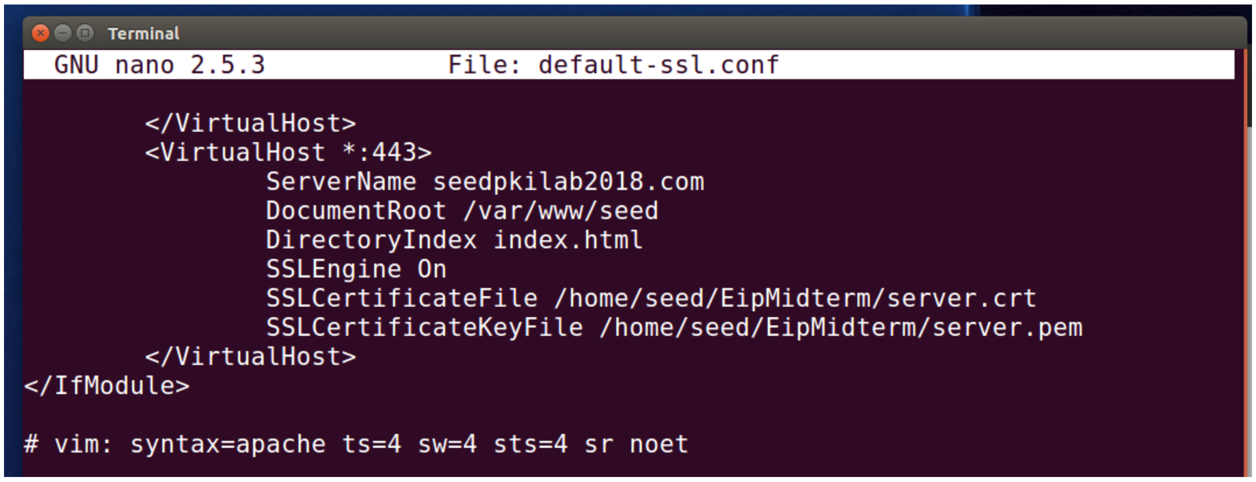


seedpkilab2018.com
Connection Is Not Secure

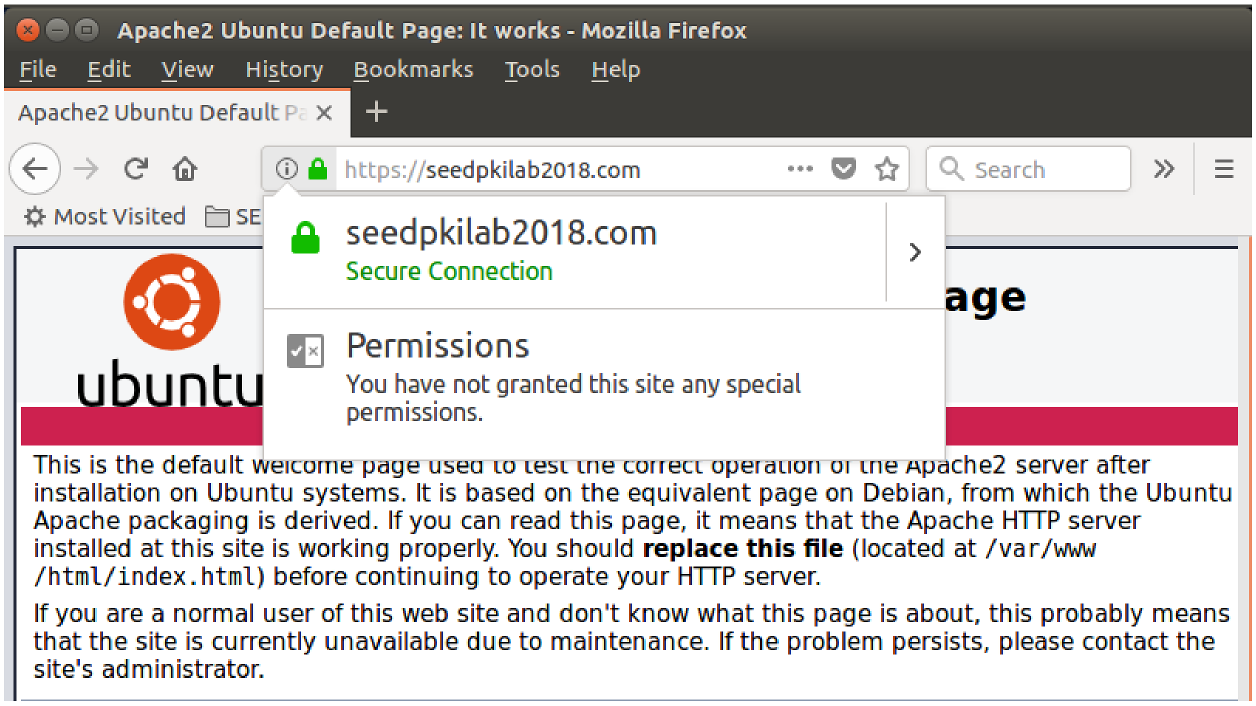
Permissions
You have not granted this site any special permissions.

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.



```
GNU nano 2.5.3                File: default-ssl.conf

        </VirtualHost>
        <VirtualHost *:443>
                ServerName seedpkilab2018.com
                DocumentRoot /var/www/seed
                DirectoryIndex index.html
                SSLEngine On
                SSLCertificateFile /home/seed/EipMidterm/server.crt
                SSLCertificateKeyFile /home/seed/EipMidterm/server.pem
        </VirtualHost>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```



Apache2 Ubuntu Default Page: It works - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache2 Ubuntu Default Pa
https://seedpkilab2018.com
Search
Most Visited
SE
ubuntu
age

seedpkilab2018.com
Secure Connection

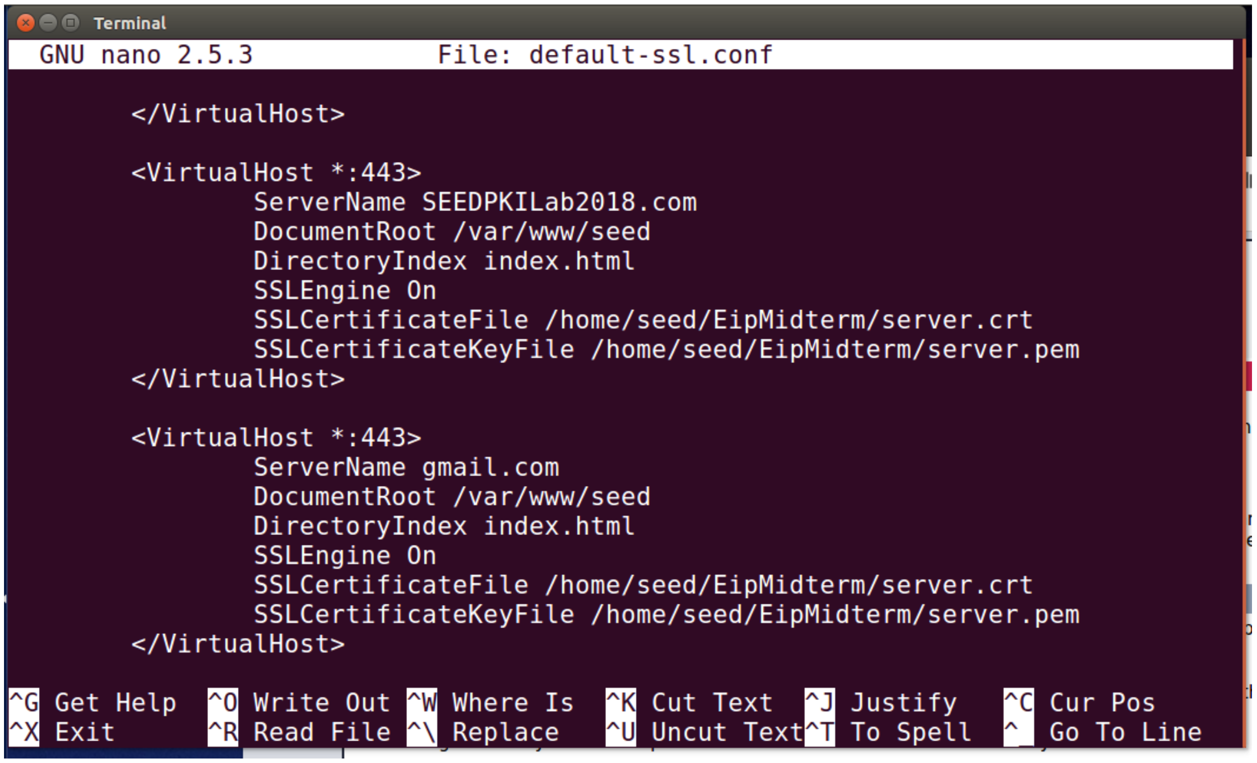
Permissions
You have not granted this site any special permissions.

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.

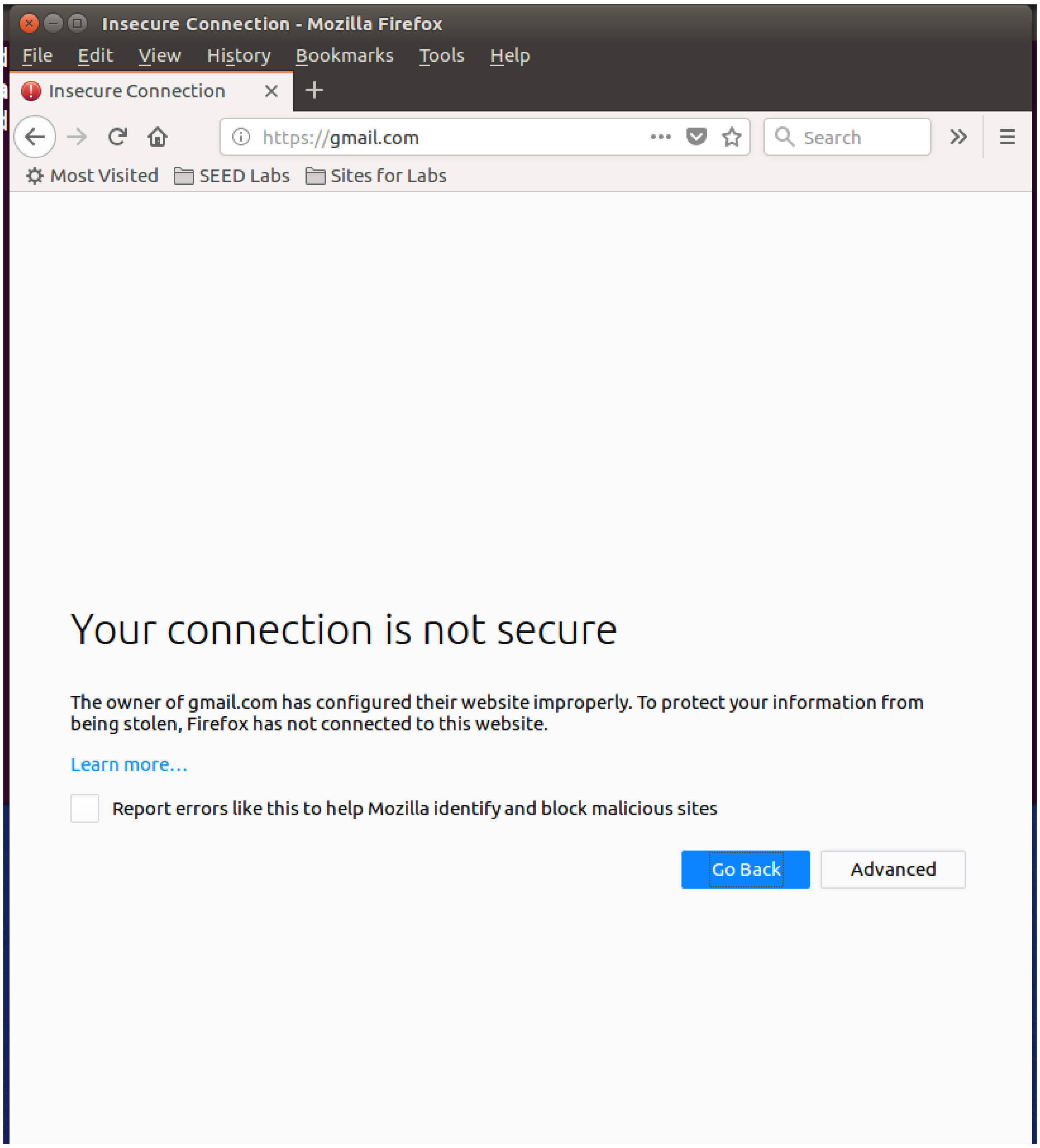
In **task 5**, I launched a MITM attack in which I had to perform the following steps:

1. Setting up the malicious website – I used the Apache server to impersonate **gmail.com** by adding a VirtualHost entry in the Apache's SSL configuration file.



```
  GNU nano 2.5.3              File: default-ssl.conf

        </VirtualHost>

        <VirtualHost *:443>
                ServerName SEEDPKILab2018.com
                DocumentRoot /var/www/seed
                DirectoryIndex index.html
                SSLEngine On
                SSLCertificateFile /home/seed/EipMidterm/server.crt
                SSLCertificateKeyFile /home/seed/EipMidterm/server.pem
        </VirtualHost>

        <VirtualHost *:443>
                ServerName gmail.com
                DocumentRoot /var/www/seed
                DirectoryIndex index.html
                SSLEngine On
                SSLCertificateFile /home/seed/EipMidterm/server.crt
                SSLCertificateKeyFile /home/seed/EipMidterm/server.pem
        </VirtualHost>

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

2. Becoming the man in the middle – I manually poisoned the DNS cache of the user by modifying ***/etc/hosts*** and adding an entry for **gmail.com**
3. Browse the target website – I visited the website and the browser showed an error saying that the connection is not secure i.e. PKI is saving the victim from visiting the malicious website.



Your connection is not secure

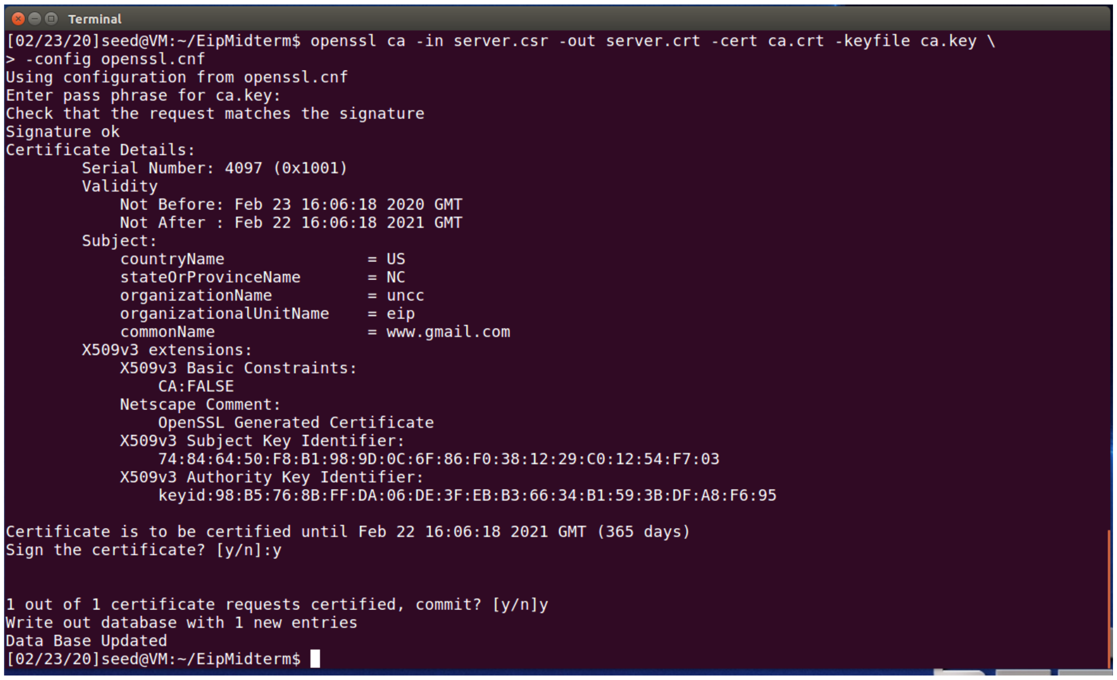
The owner of gmail.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

Learn more…

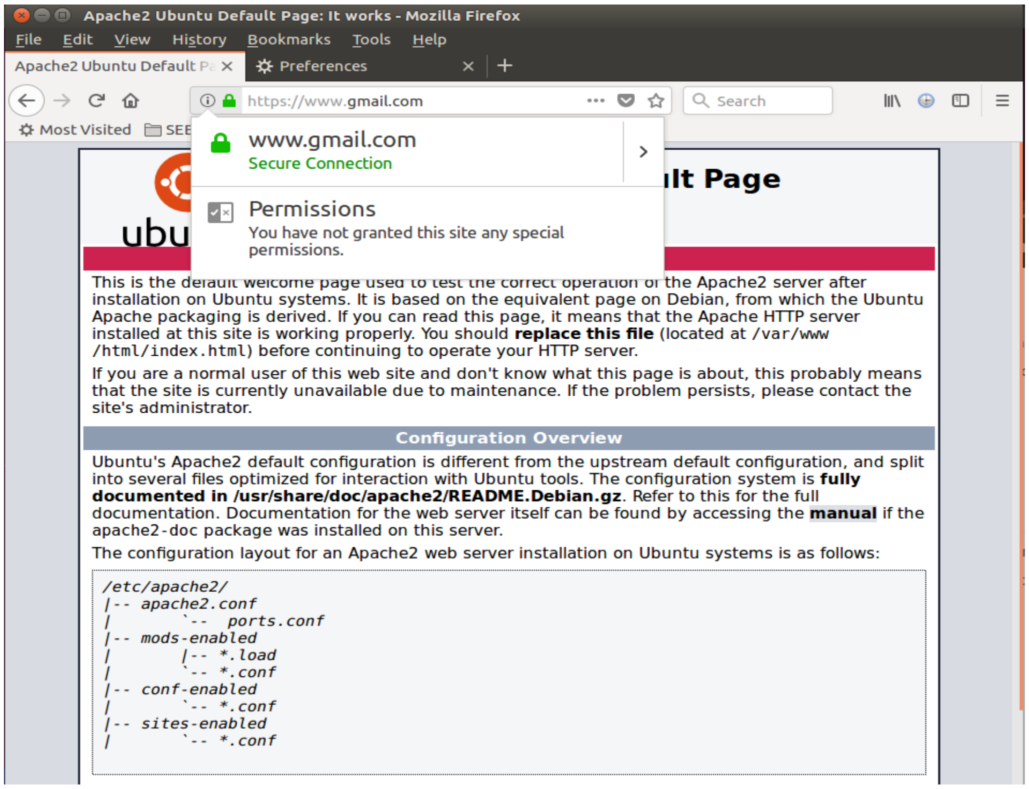
Report errors like this to help Mozilla identify and block malicious sites

In task 6, assuming that the root CA has been compromised, the attacker can now generate any certificate using the CA's private key. So, he generates a certificate which matches the domain name of the malicious server and performs the MITM attack as described in Task 5. The result (shown below) shows that the victim can land on the malicious website without him/her knowing that it is a malicious one.

Therefore, I inferred that if an attacker gains access to the root CA then he/she can bypass the security provided to the users by PKI.



```
[02/23/20]seed@VM:~/EipMidterm$ 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: 4097 (0x1001)
        Validity
            Not Before: Feb 23 16:06:18 2020 GMT
            Not After : Feb 22 16:06:18 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NC
            organizationName          = uncc
            organizationalUnitName    = eip
            commonName                = www.gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                74:84:64:50:F8:B1:98:9D:0C:6F:86:F0:38:12:29:C0:12:54:F7:03
            X509v3 Authority Key Identifier:
                keyid:98:B5:76:8B:FF:DA:06:DE:3F:EB:B3:66:34:B1:59:3B:DF:A8:F6:95

Certificate is to be certified until Feb 22 16:06:18 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
[02/23/20]seed@VM:~/EipMidterm$
```


Apache2 Ubuntu Default Page: It works - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache2 Ubuntu Default P
Preferences
https://www.gmail.com
Search
Most Visited
SEE
www.gmail.com
Secure Connection
Permissions
You have not granted this site any special permissions.
ult Page
ubu
Configuration Overview

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.

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
```