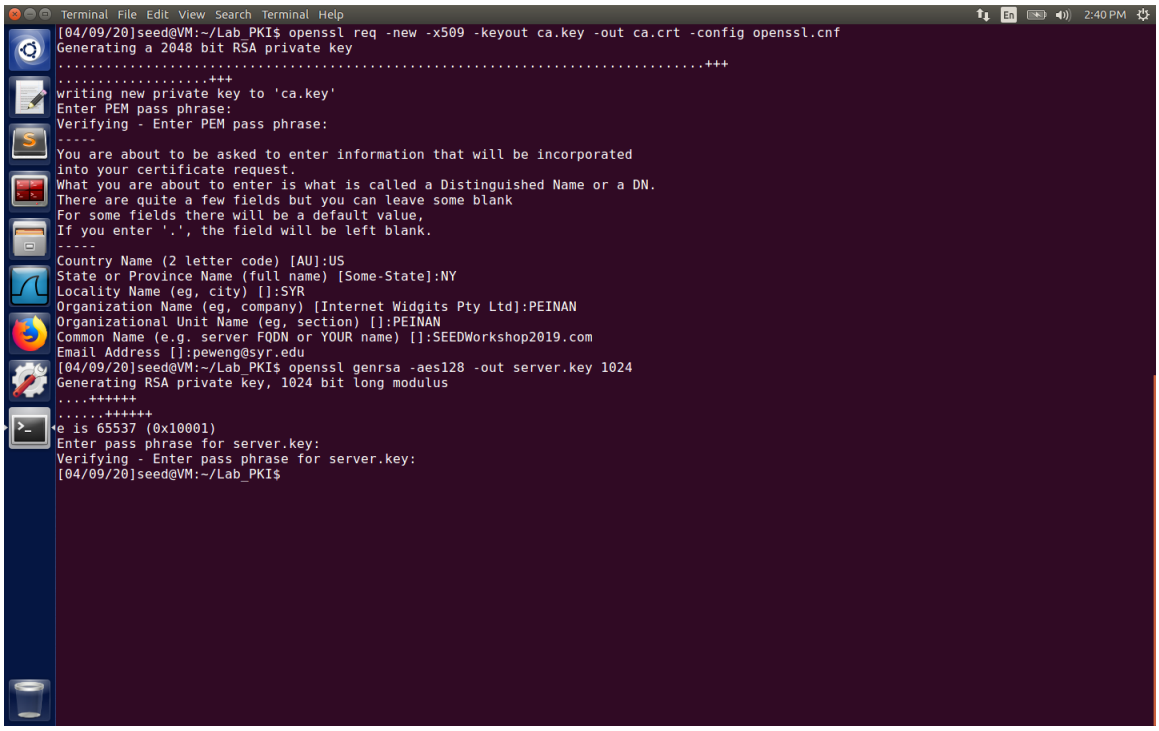
Public-Key Infrastructure (PKI) Lab

Task 1: Becoming a Certificate Authority (CA)

Task 2: Creating a Certificate for `SEEDPKILAB2018.com`

Certificate Authority (CA).

Generate public/private key pair.



```
[04/09/20]seed@VM:~/Lab_PKI$ 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) []:SYR
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PEINAN
Organizational Unit Name (eg, section) []:PEINAN
Common Name (e.g. server FQDN or YOUR name) []:SEEDWorkshop2019.com
Email Address []:peweng@syr.edu
[04/09/20]seed@VM:~/Lab_PKI$ 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:
[04/09/20]seed@VM:~/Lab_PKI$
```

Generate a Certificate Signing Request (CSR).

```
[04/09/20]seed@VM:~/Lab_PKI$ openssl req -new -key server.key -out server.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) []:SYR
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PEINAN
Organizational Unit Name (eg, section) []:PEINAN
Common Name (e.g. server FQDN or YOUR name) []:SEEDWorkshop2019.com
Email Address []:peweng@syr.edu

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:qwert970927
An optional company name []:PEINAN
[04/09/20]seed@VM:~/Lab_PKI$
```

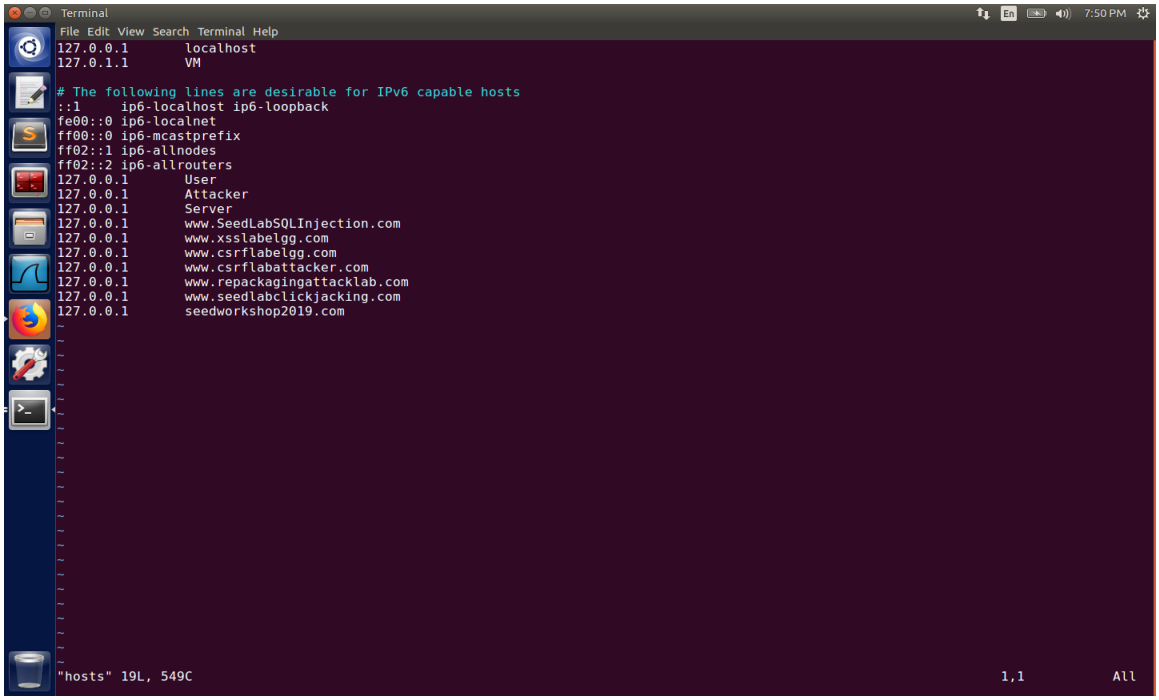
Generating Certificates.

```
[04/09/20]seed@VM:~/Lab_PKI$ 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: 4099 (0x1003)
        Validity
            Not Before: Apr  9 18:56:28 2020 GMT
            Not After : Apr  9 18:56:28 2021 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NY
            localityName              = SYR
            organizationName          = PEINAN
            organizationalUnitName    = PEINAN
            commonName                = SEEDWorkshop2019.com
            emailAddress              = peweng@syr.edu
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                87:42:9B:96:2F:9C:6F:26:2E:22:01:1D:0C:78:61:73:EC:A3:F7:64
            X509v3 Authority Key Identifier:
                keyid:EF:B1:14:57:37:93:03:E6:42:63:D6:56:58:ED:D0:C4:04:9C:E8:23

Certificate is to be certified until Apr  9 18:56:28 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
[04/09/20]seed@VM:~/Lab_PKI$
```

Task 3: Deploying Certificate in an HTTPS Web Server

Step 1: Configuring DNS

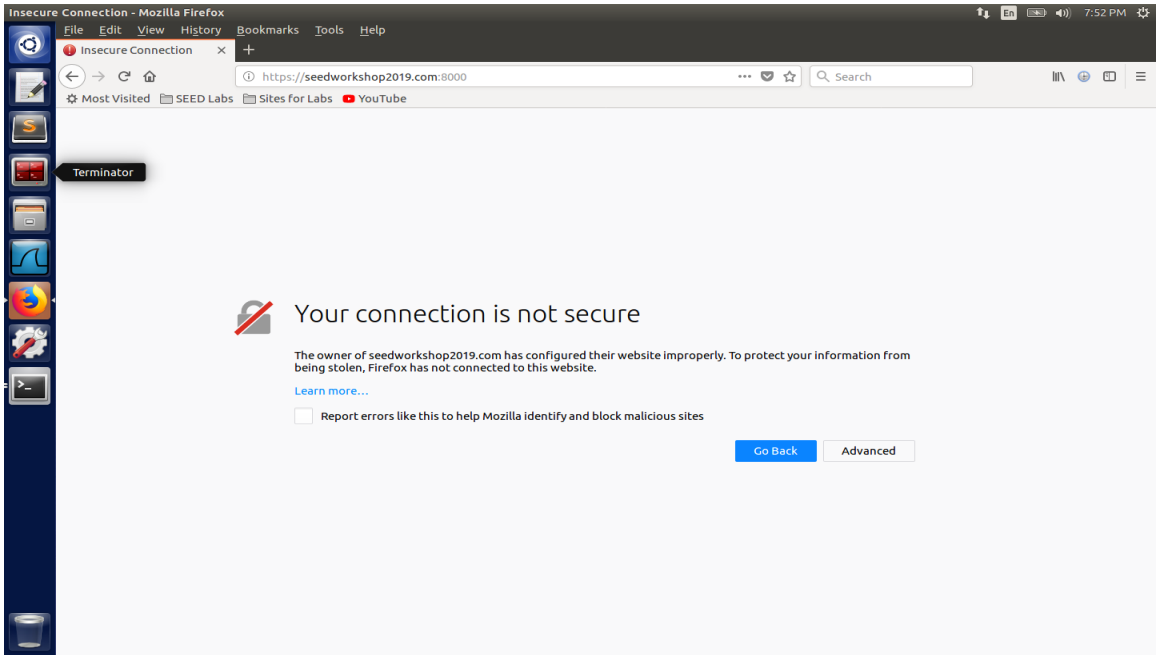
Add `seedworkshop2019.com` to `/etc/hosts`



Step 2: Configuring the web server

```
openssl s_server -cert server.pem -www -accept 8000
```

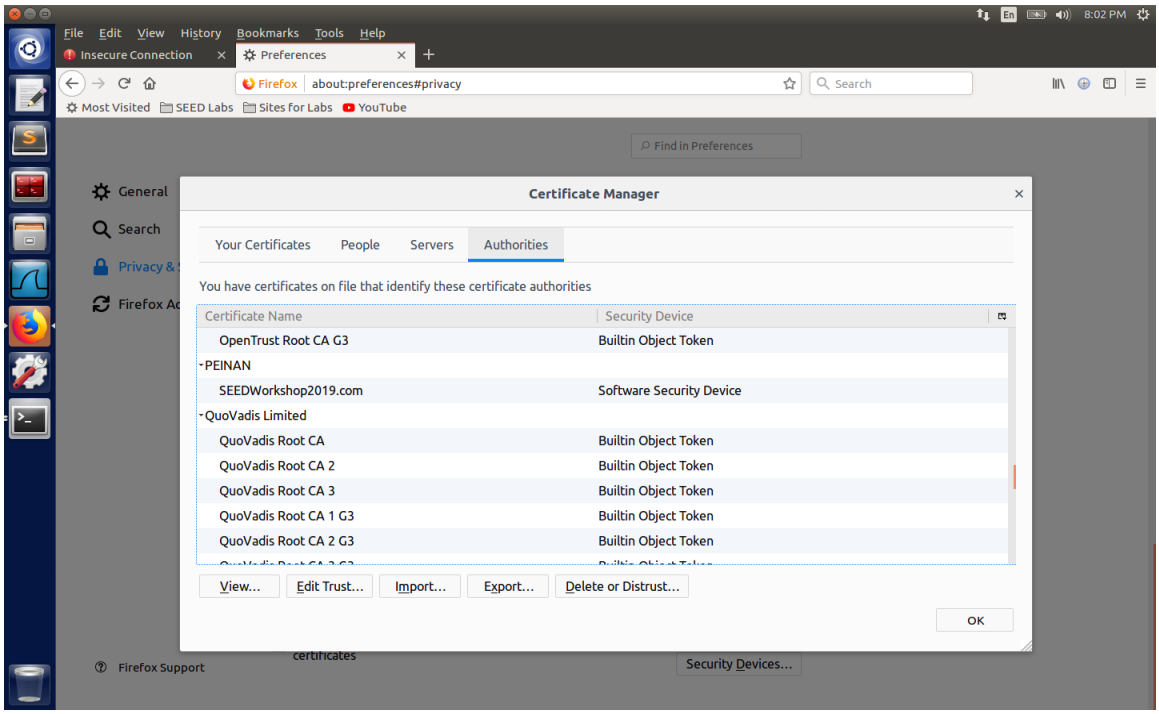
Test `seedworkshop2019.com`



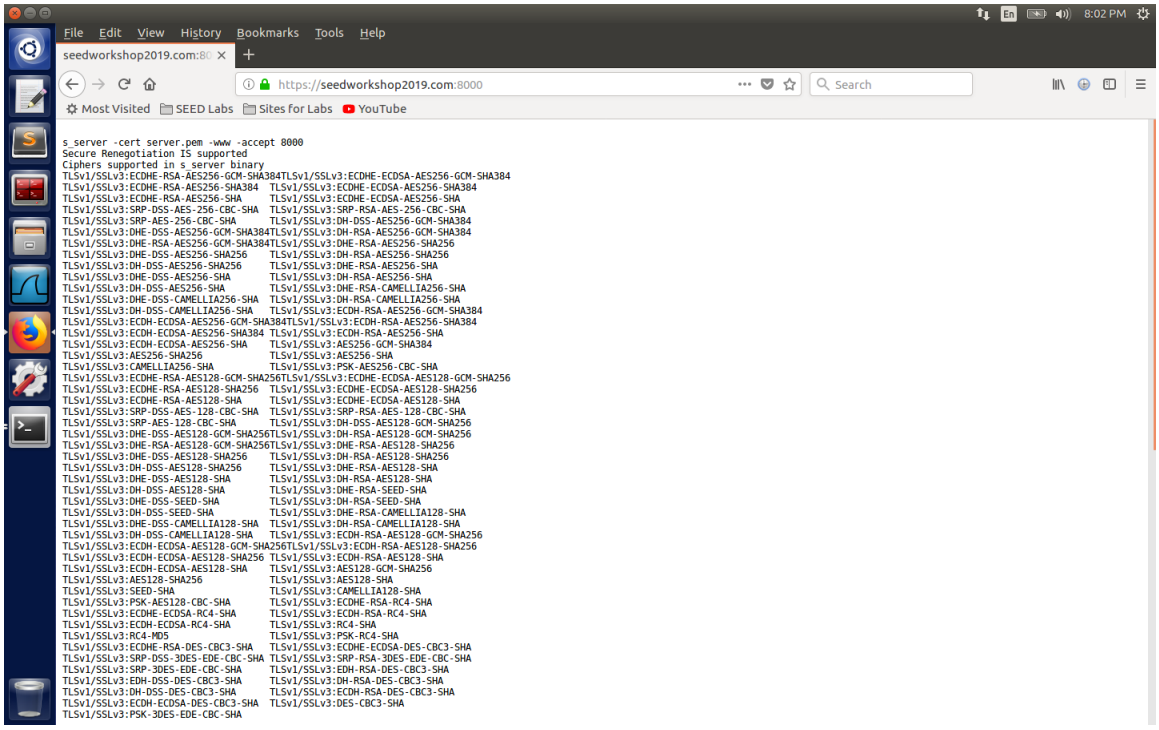
Because the issuer certificate is unknown, the certificate is not trusted and browser showed insecure connection.

Step 3: Getting the browser to accept our CA certificate

Import our `ca.crt` certificate into Firefox certificate manager.

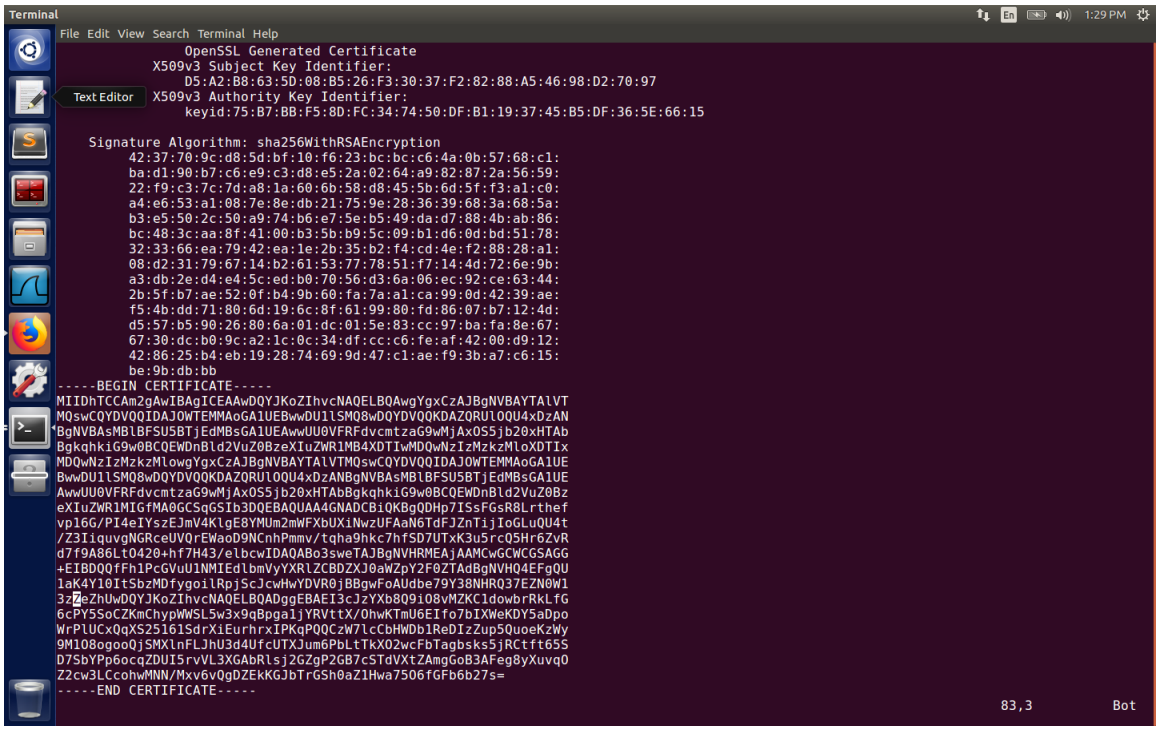


Step 4. Testing our HTTPS website

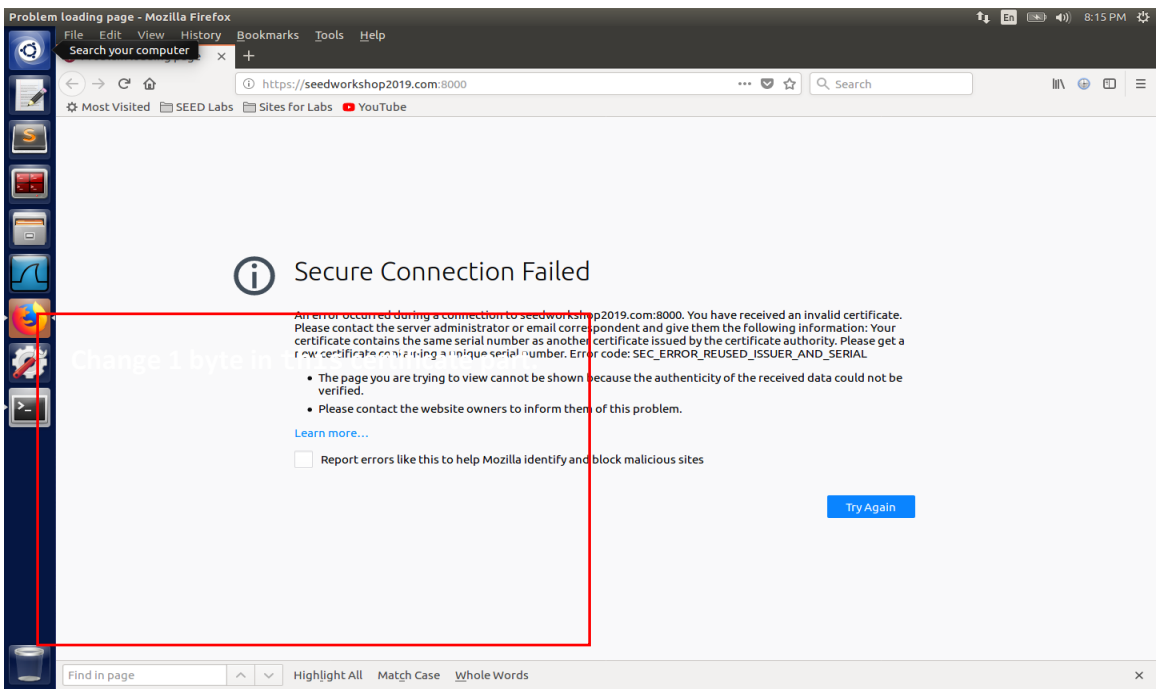


After importing our CA certificate, the browser will trust our website and load the page correctly.

1. Modify a single byte of `server.pem` and then restart the server, reload the URL.

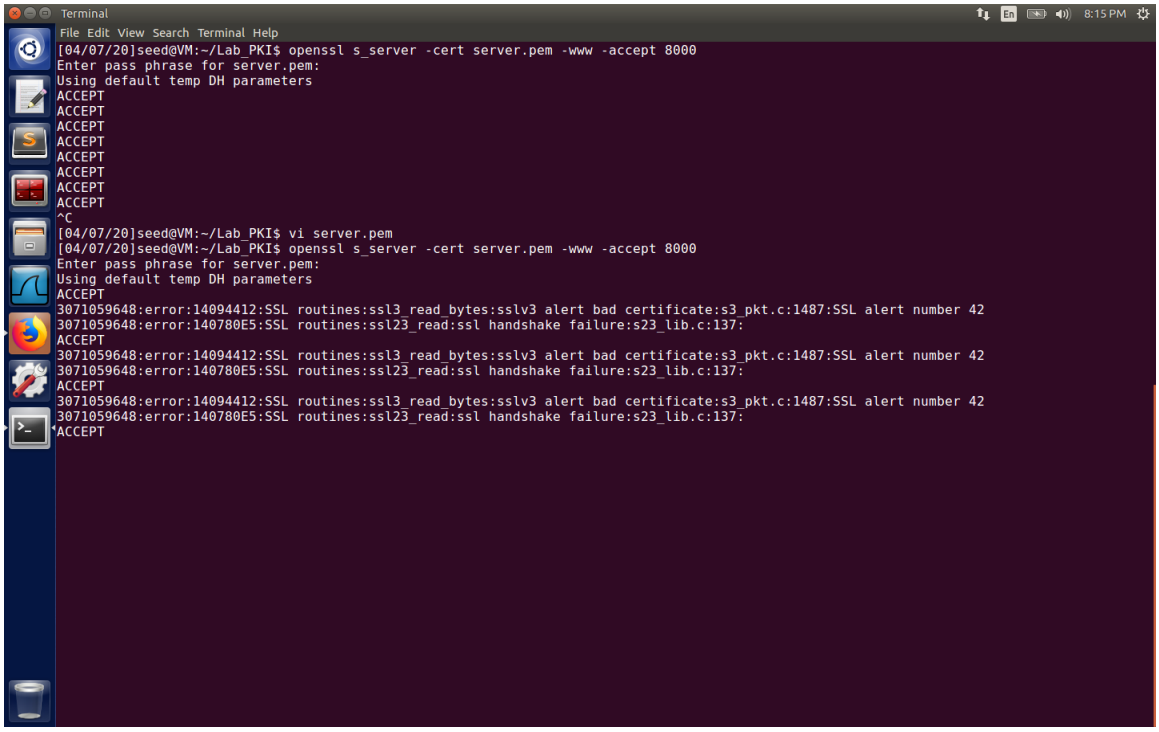


After I modified a signle byte of the certificate part of `server.pem`.



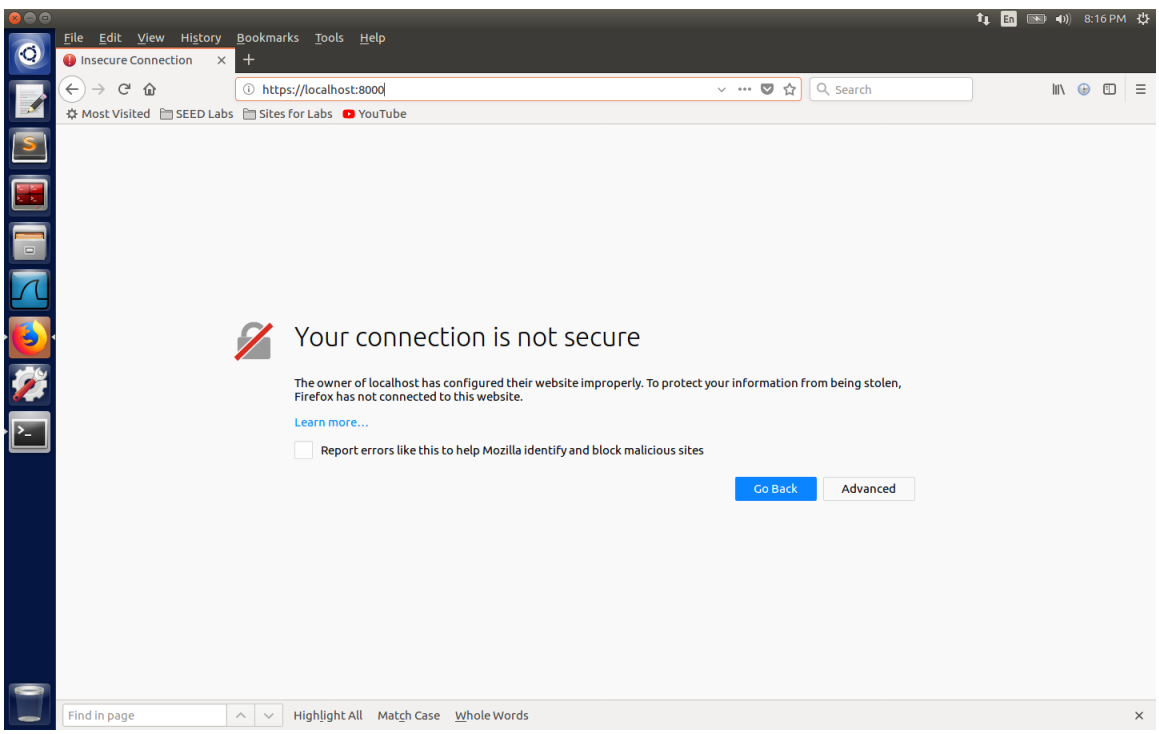
Unable to connect to `https://seekworkshop2019.com` because the certificate is incorrect.

On our web server console side, it showed `bad certificate`.



```
[04/07/20]seed@VM:~/Lab_PKI$ openssl s_server -cert server.pem -www -accept 8000
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
ACCEPT
ACCEPT
ACCEPT
ACCEPT
ACCEPT
ACCEPT
^C
[04/07/20]seed@VM:~/Lab_PKI$ vi server.pem
[04/07/20]seed@VM:~/Lab_PKI$ openssl s_server -cert server.pem -www -accept 8000
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
3071059648:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad certificate:s3_pkt.c:1487:SSL alert number 42
3071059648:error:140780E5:SSL routines:ssl23_read:ssl handshake failure:s23_lib.c:137:
ACCEPT
3071059648:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad certificate:s3_pkt.c:1487:SSL alert number 42
3071059648:error:140780E5:SSL routines:ssl23_read:ssl handshake failure:s23_lib.c:137:
ACCEPT
3071059648:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad certificate:s3_pkt.c:1487:SSL alert number 42
3071059648:error:140780E5:SSL routines:ssl23_read:ssl handshake failure:s23_lib.c:137:
ACCEPT
```

2. What happened if we use `https://localhost:8000` instead.



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

Unable to connect to `https://localhost:8000` because the certificate is matched based on the domain name instead of host address. Even if the host address of localhost and `seedworkshop2019.com` are both `127.0.0.1`, their domain name are different.

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

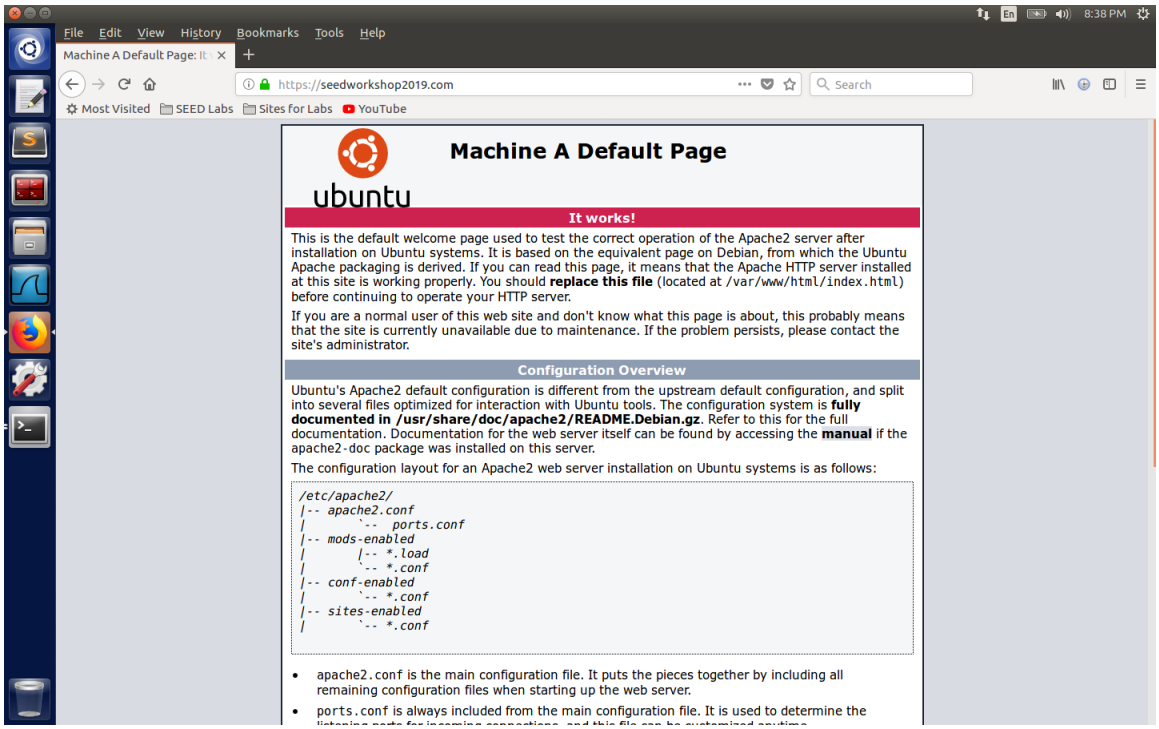
Configure our Apache server.

```
Terminal
File  Edit  View  Search  Terminal  Help
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabelgg.com
        DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabattacker.com
        DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.repackagingattacklab.com
        DocumentRoot /var/www/RepackagingAttack
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.seedlabclickjacking.com
        DocumentRoot /var/www/seedlabclickjacking
</VirtualHost>
<VirtualHOst *:443>
        ServerName seedworkshop2019.com
        DocumentRoot /var/www/html
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/server.pem
        SSLCertificateKeyFile /etc/apache2/ssl/server.key
</VirtualHost>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                    39,14         Bot
```

Enable our Apache server.

```
Terminal
File  Edit  View  Search  Terminal  Help
[04/07/20]seed@VM:.../apache2$ 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
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
[04/07/20]seed@VM:.../apache2$ 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
[04/07/20]seed@VM:.../apache2$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
[04/07/20]seed@VM:.../apache2$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for seedworkshop2019.com:443 (RSA): **********
[04/07/20]seed@VM:.../apache2$
```

Then test our website in our browser.



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

Step 1: Setting up the malicious website

```
        DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabattacker.com
        DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.repackagingattacklab.com
        DocumentRoot /var/www/RepackagingAttack
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.seedlabclickjacking.com
        DocumentRoot /var/www/seedlabclickjacking
</VirtualHost>
<VirtualHost *:443>
        ServerName seedworkshop2019.com
        DocumentRoot /var/www/html
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/server.pem
        SSLCertificateKeyFile /etc/apache2/ssl/server.key
</VirtualHost>
<VirtualHost *:443>
        ServerName google.com
        DocumentRoot /var/www/html
        DirectoryIndex index.html

        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/server.pem
        SSLCertificateKeyFile /etc/apache2/ssl/server.key
</VirtualHost>
                                                                42,9          Bot
```

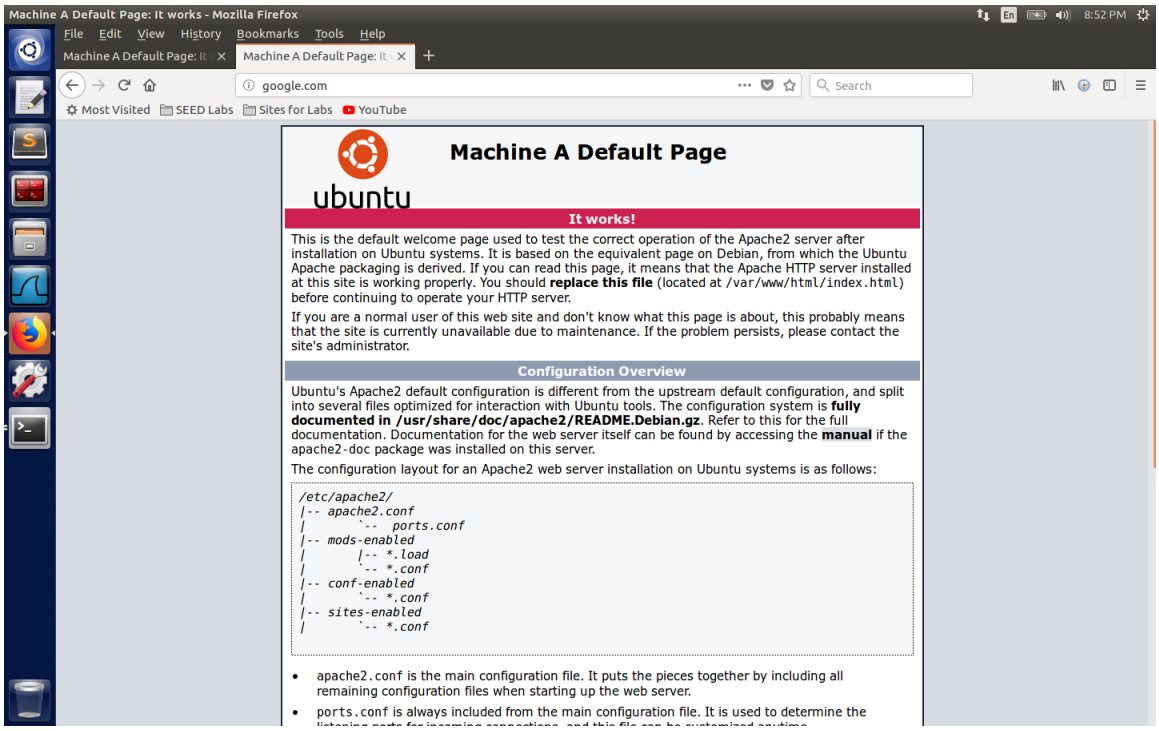
Step 2: Becoming the man in the middle

Add `google.com` to `/etc/hosts`

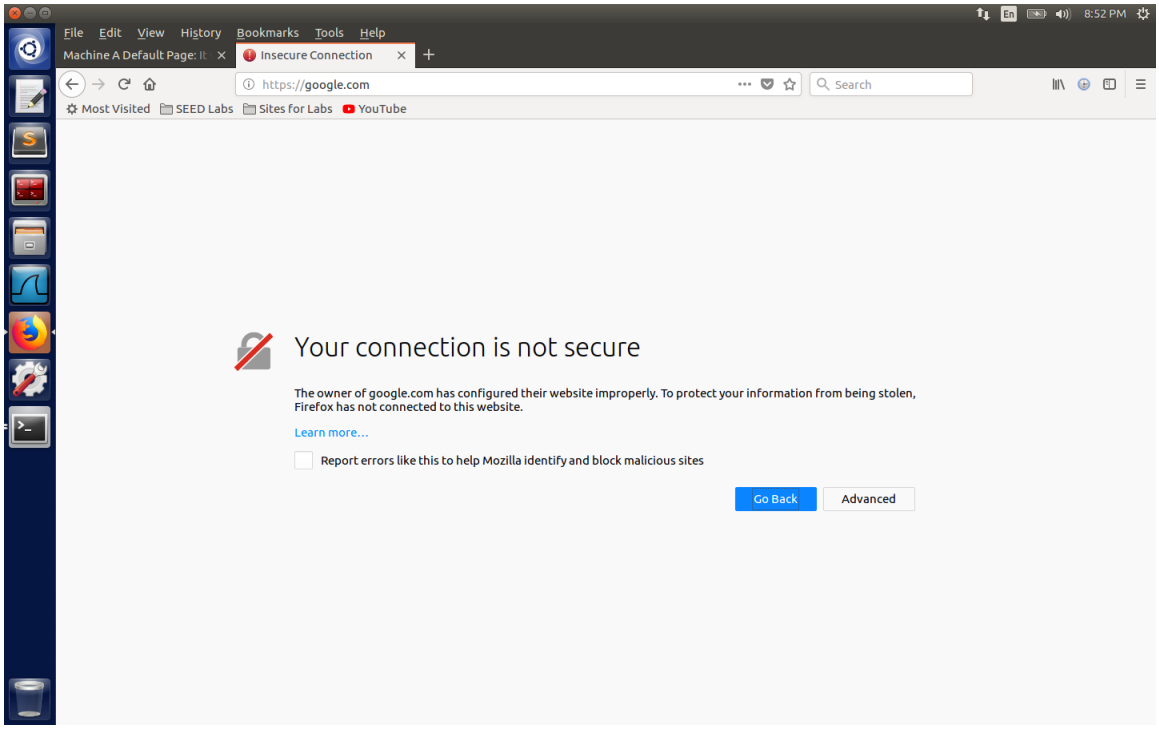
```
127.0.0.1       localhost
127.0.1.1       VM

# 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       seedworkshop2019.com
127.0.0.1       google.com
-- INSERT --                                                    20,21-27      All
```

Step 3: Browse the target website

If we just type `google.com` in our browser, we can see this page successfully on HTTP.



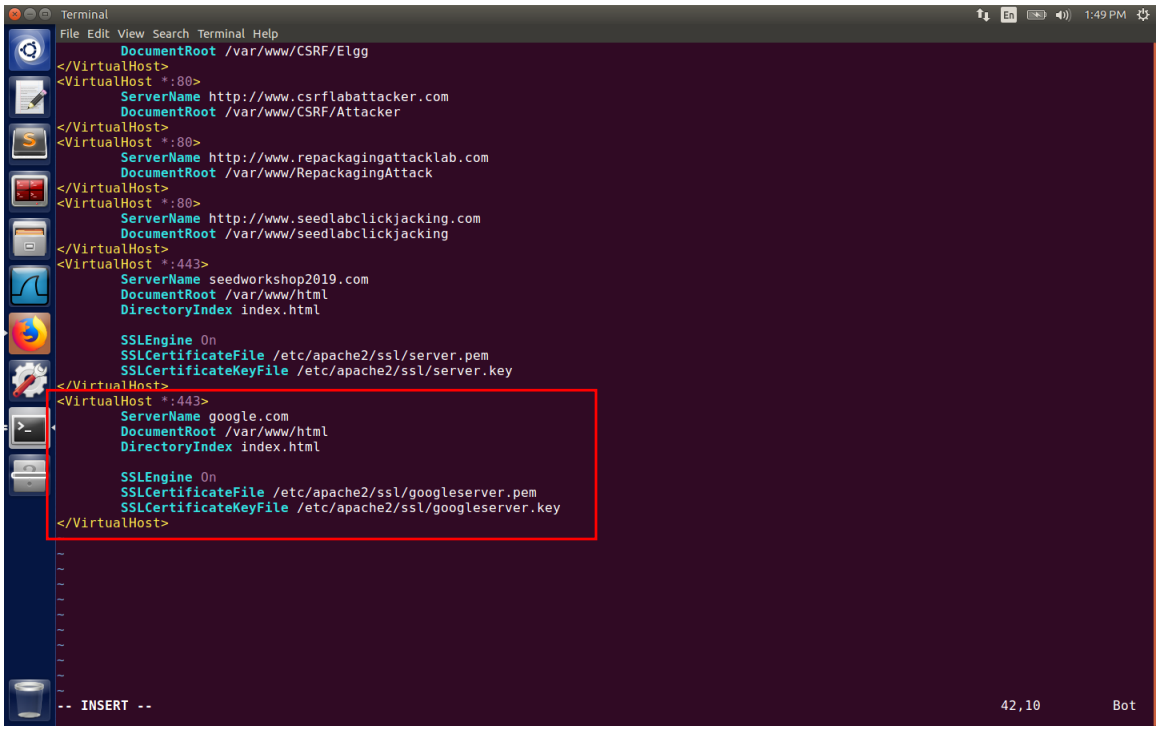
If we force to use HTTPS, it showed insecure connection.



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

In Task 5, we cannot browse `https://google.com`. This is because we configure this domain name in the configuration fire of Apache server. However, we still use the key generated for `seedworkshop2019.com` domain name. One key is generated only for one domain name, if we use the same key for another one, it could be an invalid certificate for that domain name.

So we need to generate another key file for our `google.com` and set the CommonName to `google.com`.



Then we could connect to our google.com successfully.