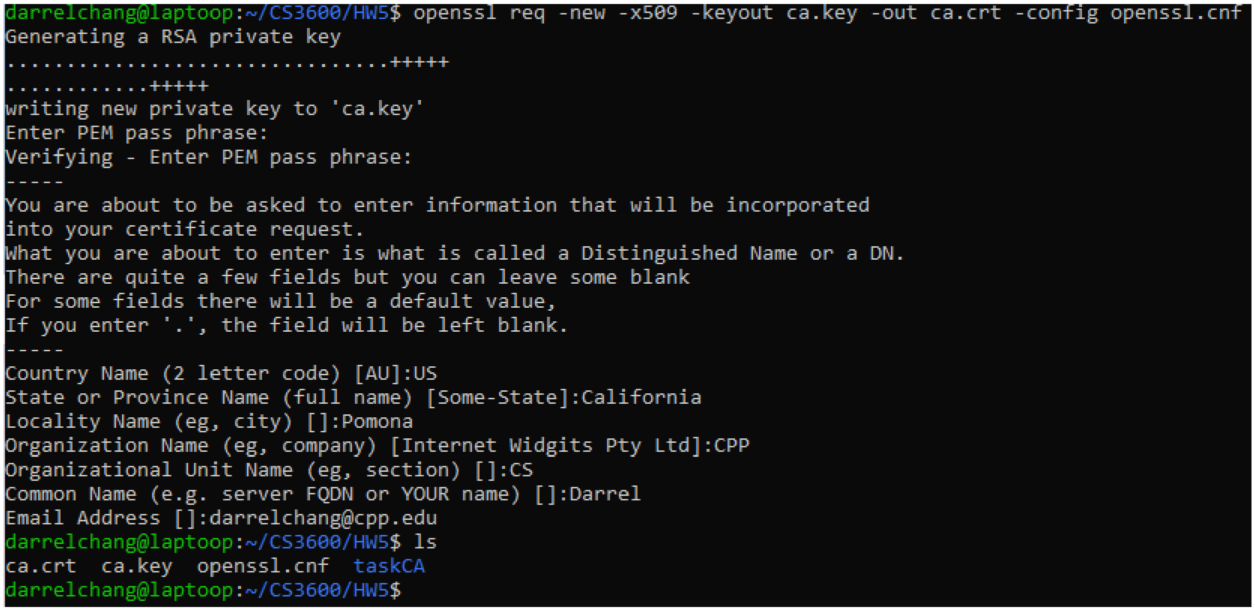
Darrel Chang
Tingting Chen
CS 4600
5/9/2024

Homework 5 PKI SEED Lab

Task 1: Becoming a Certificate Authority (CA)

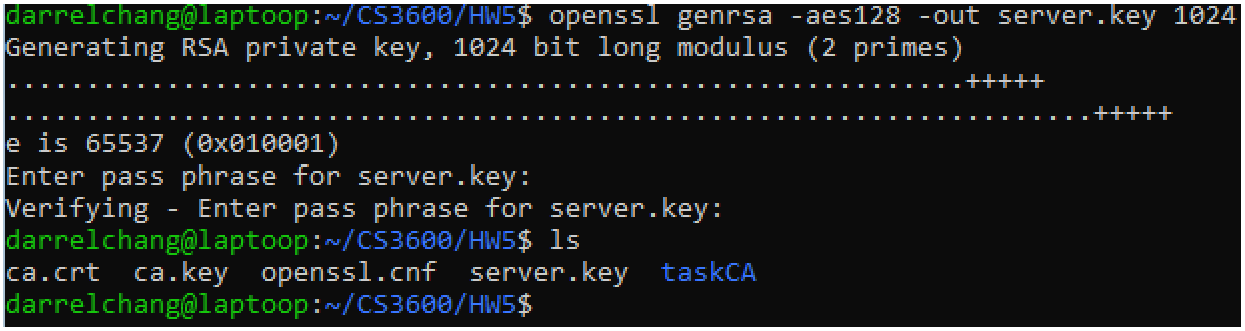


```
darrelchang@laptoop:~/CS3600/HW5$ openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf
Generating a RSA private key
...............................+++++
...........+++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about 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]:California
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CPP
Organizational Unit Name (eg, section) []:CS
Common Name (e.g. server FQDN or YOUR name) []:Darrel
Email Address []:darrelchang@cpp.edu
darrelchang@laptoop:~/CS3600/HW5$ ls
ca.crt  ca.key  openssl.cnf  taskCA
darrelchang@laptoop:~/CS3600/HW5$
```

After running the command, I create an RSA private key for my CA.key and then add info about my CA authority.

Task 2: Creating a Certificate for SEEDPKILab2020.com

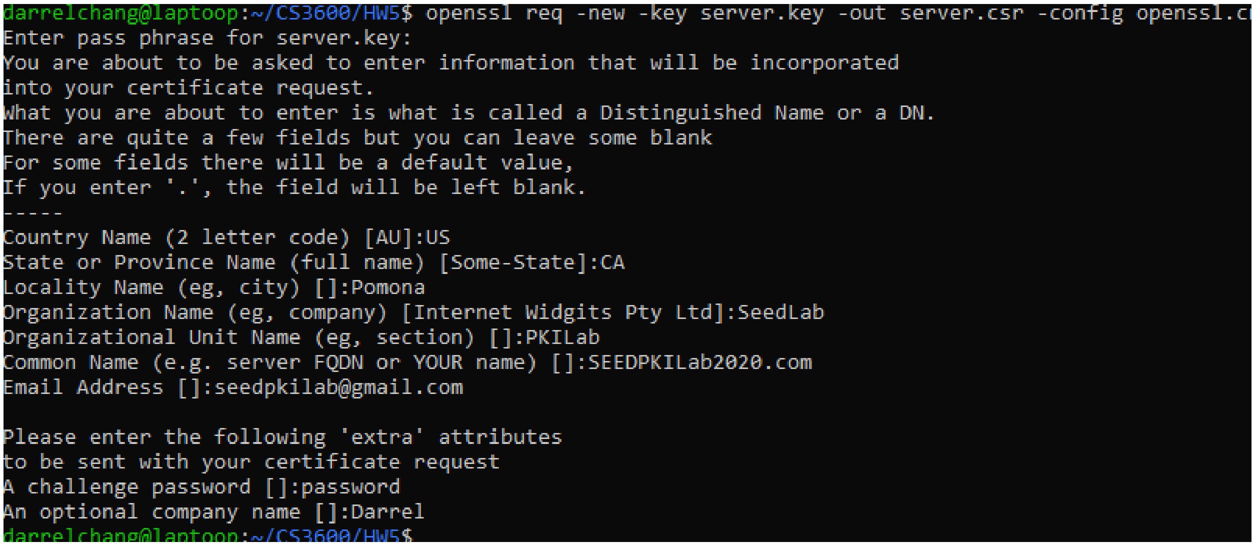
Step 1: Generate public/private key pair.



```
darrelchang@laptoop:~/CS3600/HW5$ openssl genrsa -aes128 -out server.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
............................................................+++++
....................................................................+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
darrelchang@laptoop:~/CS3600/HW5$ ls
ca.crt  ca.key  openssl.cnf  server.key  taskCA
darrelchang@laptoop:~/CS3600/HW5$
```

Here I generate a certificate after becoming a CA authority, which creates a signed server.key file

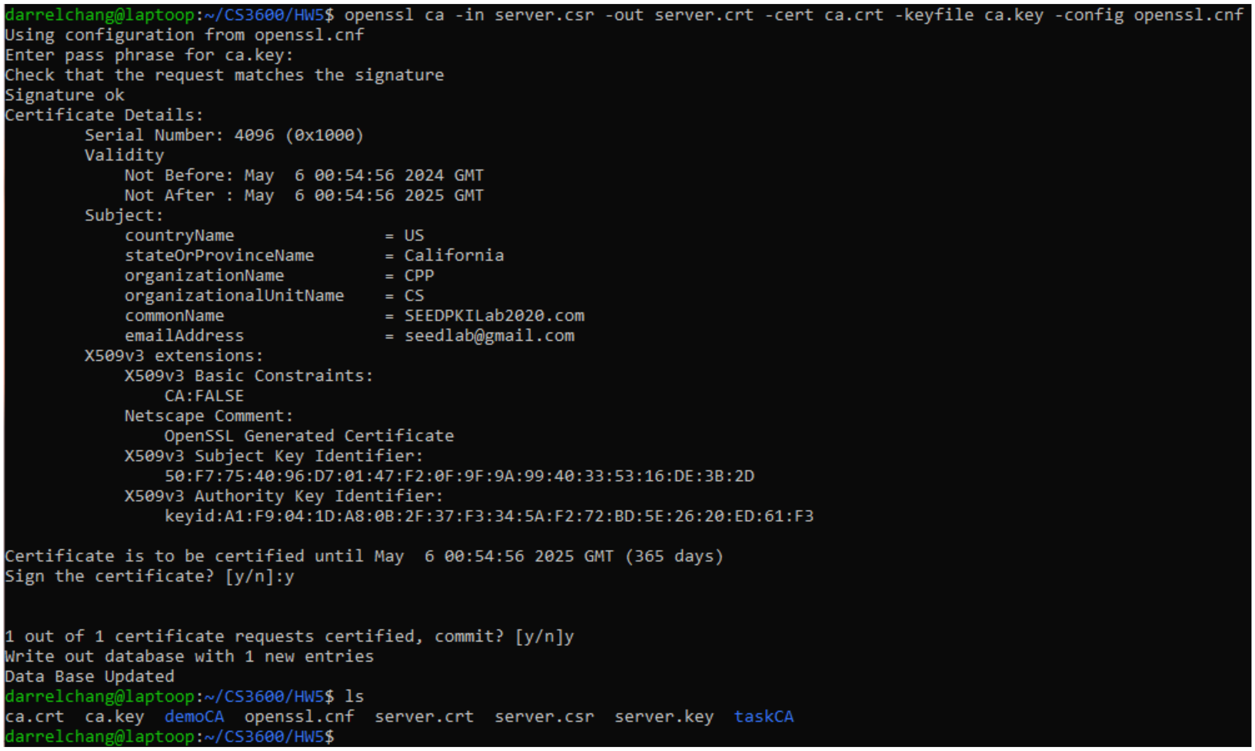
Step 2: Generate a Certificate Signing Request (CSR).



```
darrelchang@laptoop:~/CS3600/HW5$ openssl req -new -key server.key -out server.csr -config openssl.c
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]:CA
Locality Name (eg, city) []:Pomona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SeedLab
Organizational Unit Name (eg, section) []:PKILab
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2020.com
Email Address []:seedpkilab@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:Darrel
darrelchang@laptoop:~/CS3600/HW5$
```

With the server.key file I generate a CSR after adding information about it

Step 3: Generating Certificates

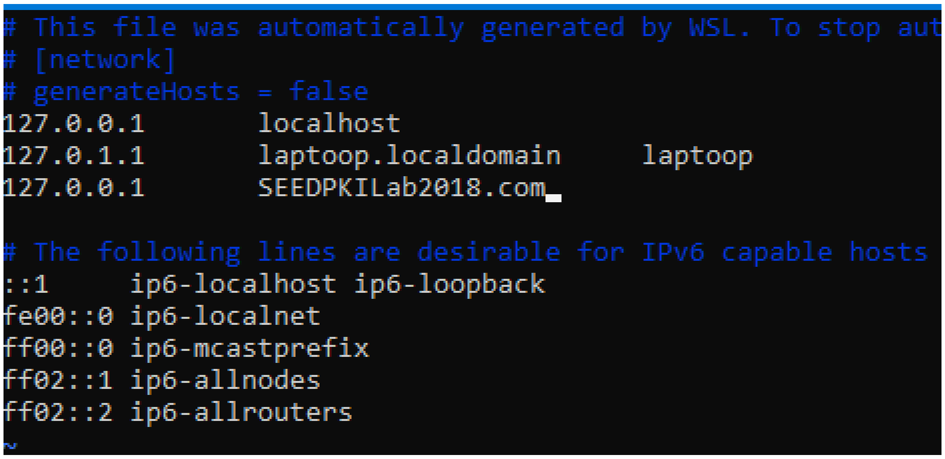


```
darrelchang@laptoop:~/CS3600/HW5$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: May  6 00:54:56 2024 GMT
            Not After : May  6 00:54:56 2025 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = California
            organizationName          = CPP
            organizationalUnitName    = CS
            commonName                = SEEDPKILab2020.com
            emailAddress              = seedlab@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                50:F7:75:40:96:D7:01:47:F2:0F:9F:9A:99:40:33:53:16:DE:3B:2D
            X509v3 Authority Key Identifier:
                keyid:A1:F9:04:1D:A8:0B:2F:37:F3:34:5A:F2:72:BD:5E:26:20:ED:61:F3

Certificate is to be certified until May  6 00:54:56 2025 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
darrelchang@laptoop:~/CS3600/HW5$ ls
ca.crt  ca.key  demoCA  openssl.cnf  server.crt  server.csr  server.key  taskCA
darrelchang@laptoop:~/CS3600/HW5$
```

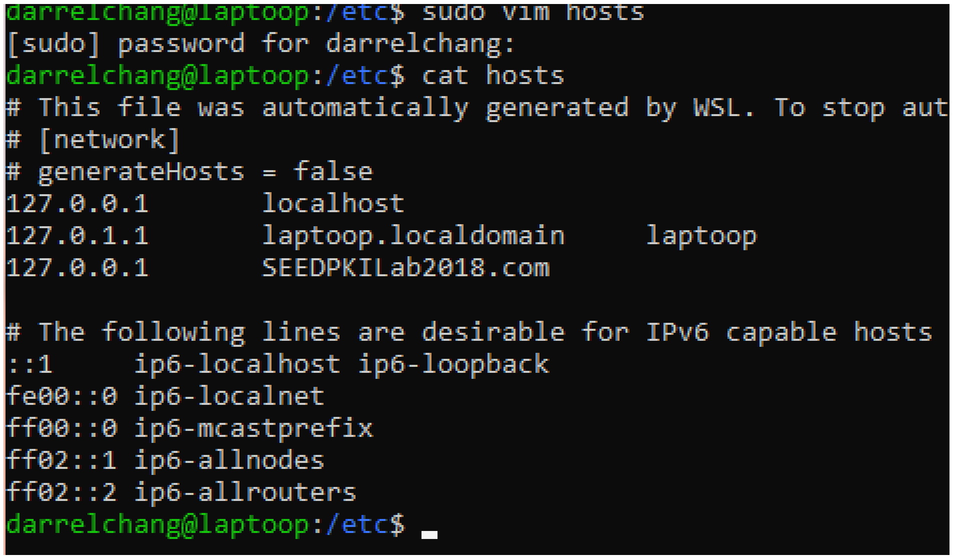
I then generate the certificate for the CSR that was created using my CA authority

Task 3: Deploying Certificate in an HTTPS Web Server

Step 1: Configuring DNS.
Added 127.0.0.1   SEEDPKILAB2018.com to /etc/hosts file using VIM

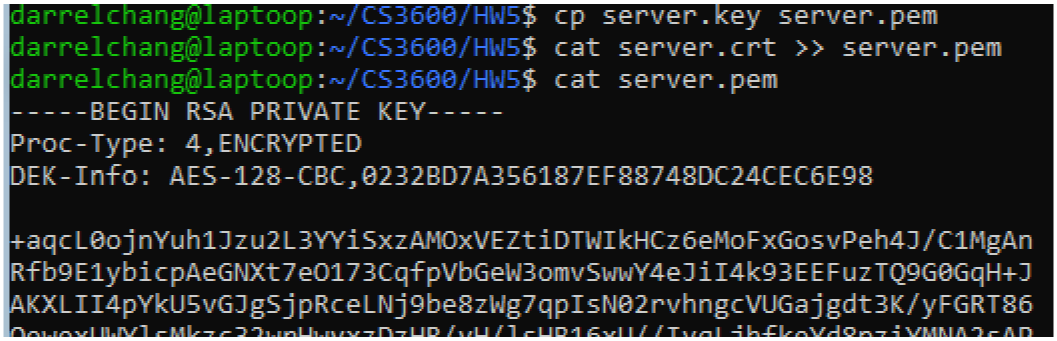


```
# This file was automatically generated by WSL. To stop aut
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       laptoop.localdomain     laptoop
127.0.0.1       SEEDPKILab2018.com

# 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
~
```



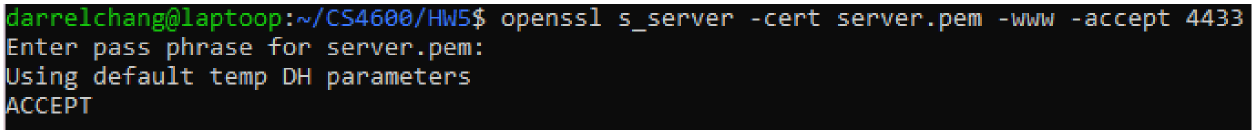
```
darrelchang@laptoop:/etc$ sudo vim hosts
[sudo] password for darrelchang:
darrelchang@laptoop:/etc$ cat hosts
# This file was automatically generated by WSL. To stop aut
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       laptoop.localdomain     laptoop
127.0.0.1       SEEDPKILab2018.com

# 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
darrelchang@laptoop:/etc$
```

I add a line to create a URL alias for my localhost address as the url: SEEDPKILab2018.com
Step 2 Configuring the web server

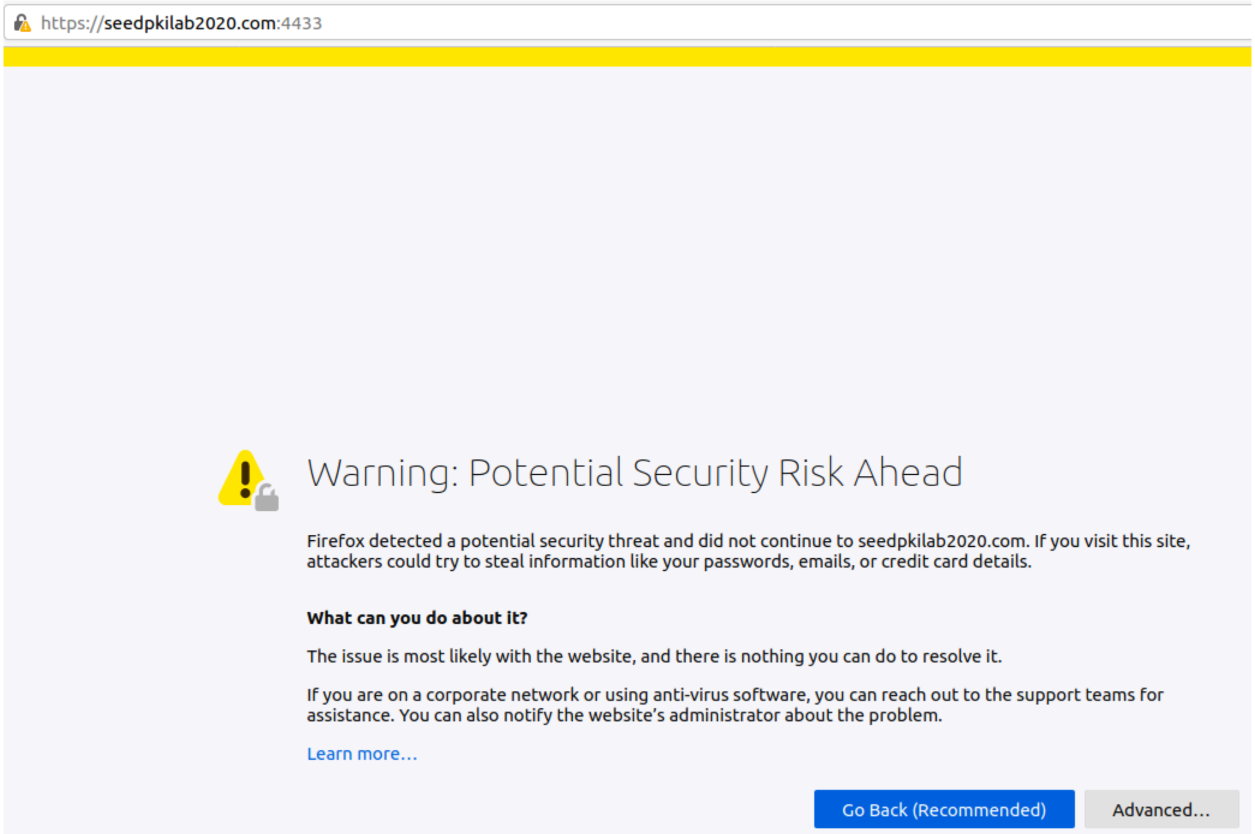


```
darrelchang@laptoop:~/CS3600/HW5$ cp server.key server.pem
darrelchang@laptoop:~/CS3600/HW5$ cat server.crt >> server.pem
darrelchang@laptoop:~/CS3600/HW5$ cat server.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,0232BD7A356187EF88748DC24CEC6E98

+aqcL0ojnYuh1Jzu2L3YYiSxzAMOxVEZtiDTWIkHCz6eMoFxGosvPeh4J/C1MgAn
Rfb9E1ybicpAeGNXt7eO173CqfpVbGeW3omvSwwY4eJiI4k93EEFuzTQ9G0GqH+J
AKXLII4pYkU5vGJgSjpRceLNj9be8zWg7qpIsN02rvhngcVUGajgdt3K/yFGRT86
```

Running web server



```
darrelchang@laptoop:~/CS4600/HW5$ openssl s_server -cert server.pem -www -accept 4433
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```



https://seedpkilab2020.com:4433

Warning: Potential Security Risk Ahead

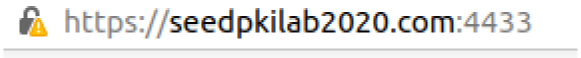
Firefox detected a potential security threat and did not continue to seedpkilab2020.com. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website’s administrator about the problem.

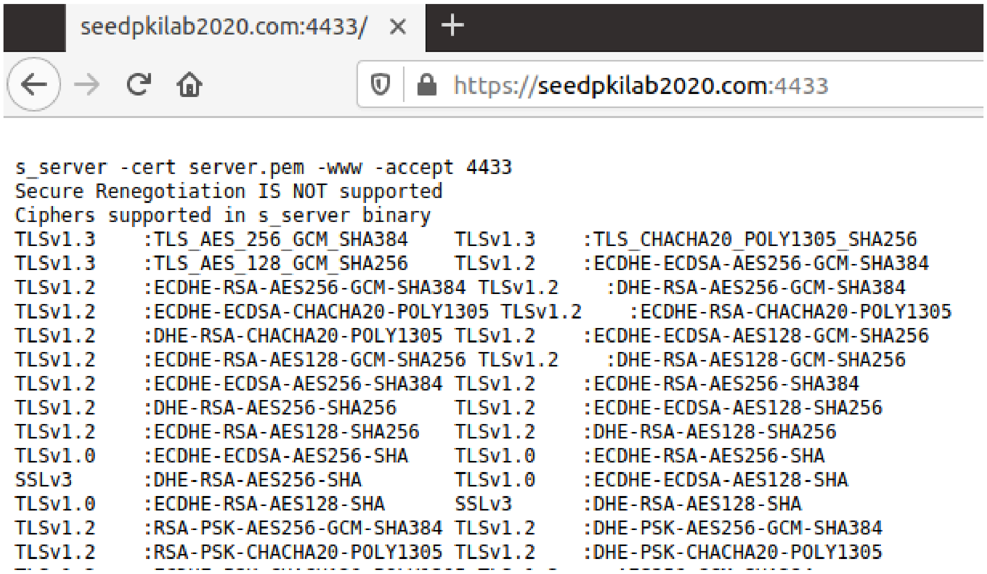
When we go to the url after ignoring the warnings, the connection is insecure



https://seedpkilab2020.com:4433

Step 3: Getting the browser to accept our CA certificate

When we connect after adding our CA authority to the browser, our connection is secure

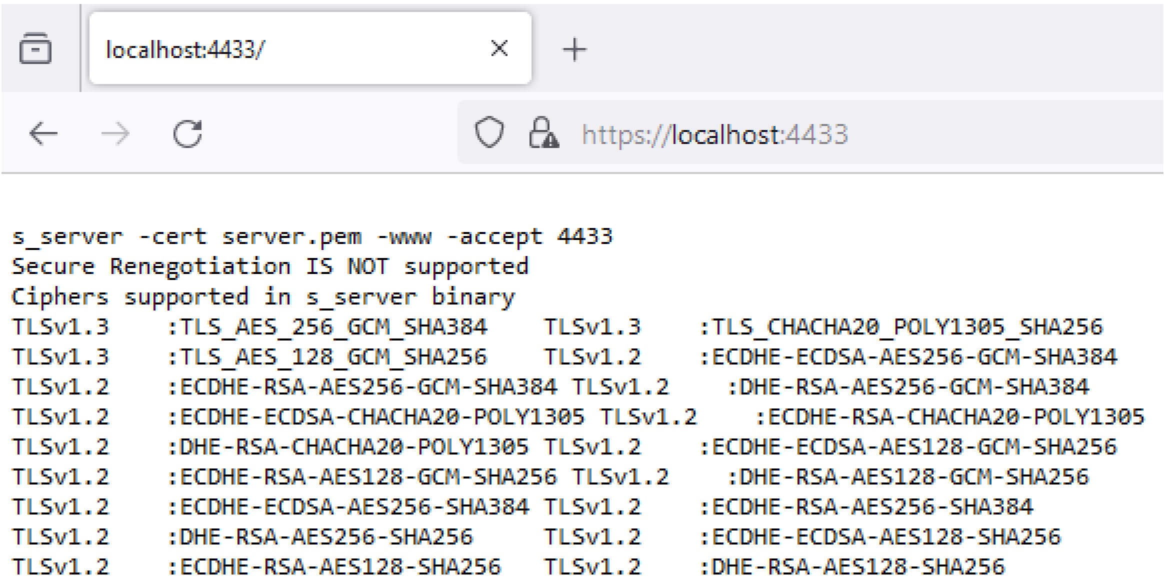


```
s_server -cert server.pem -www -accept 4433
Secure Renegotiation IS NOT supported
Ciphers supported in s_server binary
TLSv1.3    :TLS_AES_256_GCM_SHA384    TLSv1.3    :TLS_CHACHA20_POLY1305_SHA256
TLSv1.3    :TLS_AES_128_GCM_SHA256    TLSv1.2    :ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2    :DHE-RSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-RSA-CHACHA20-POLY1305
TLSv1.2    :DHE-RSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2    :DHE-RSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-ECDSA-AES256-SHA384 TLSv1.2    :ECDHE-RSA-AES256-SHA384
TLSv1.2    :DHE-RSA-AES256-SHA256     TLSv1.2    :ECDHE-ECDSA-AES128-SHA256
TLSv1.2    :ECDHE-RSA-AES128-SHA256   TLSv1.2    :DHE-RSA-AES128-SHA256
TLSv1.0    :ECDHE-ECDSA-AES256-SHA    TLSv1.0    :ECDHE-RSA-AES256-SHA
SSLv3      :DHE-RSA-AES256-SHA        TLSv1.0    :ECDHE-ECDSA-AES128-SHA
TLSv1.0    :ECDHE-RSA-AES128-SHA      SSLv3      :DHE-RSA-AES128-SHA
TLSv1.2    :RSA-PSK-AES256-GCM-SHA384 TLSv1.2    :DHE-PSK-AES256-GCM-SHA384
TLSv1.2    :RSA-PSK-CHACHA20-POLY1305 TLSv1.2    :DHE-PSK-CHACHA20-POLY1305
```

Step 4. Testing our HTTPS website

1. When changing just one byte of information, the website still looks the same. I'm not sure if it is supposed to be different or not.

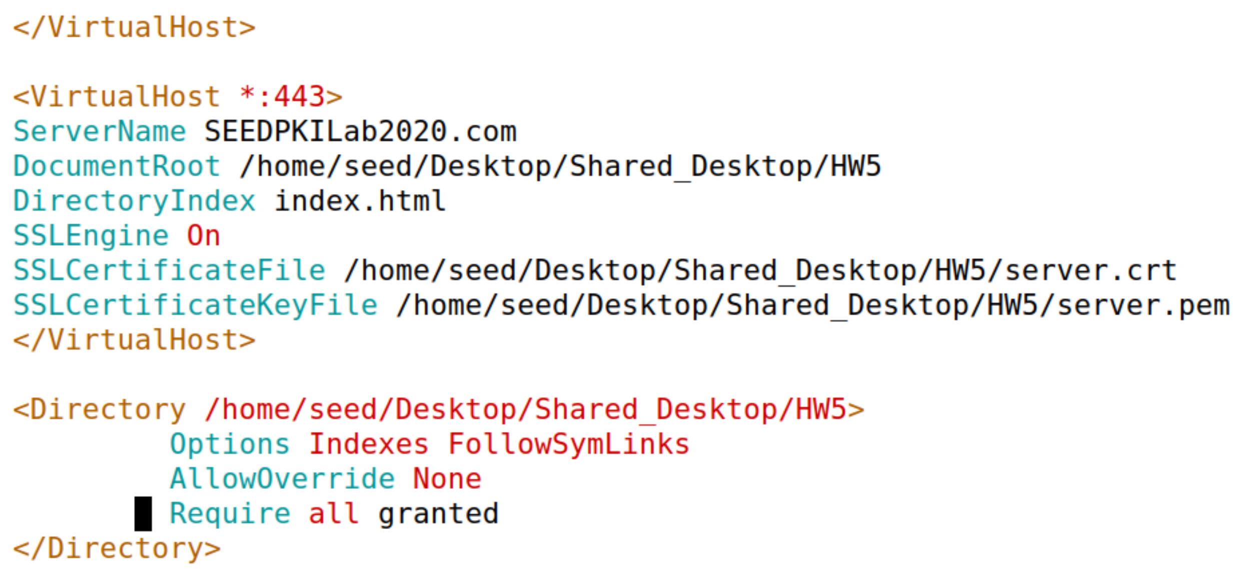
2. When connecting to localhost we get the same server connection



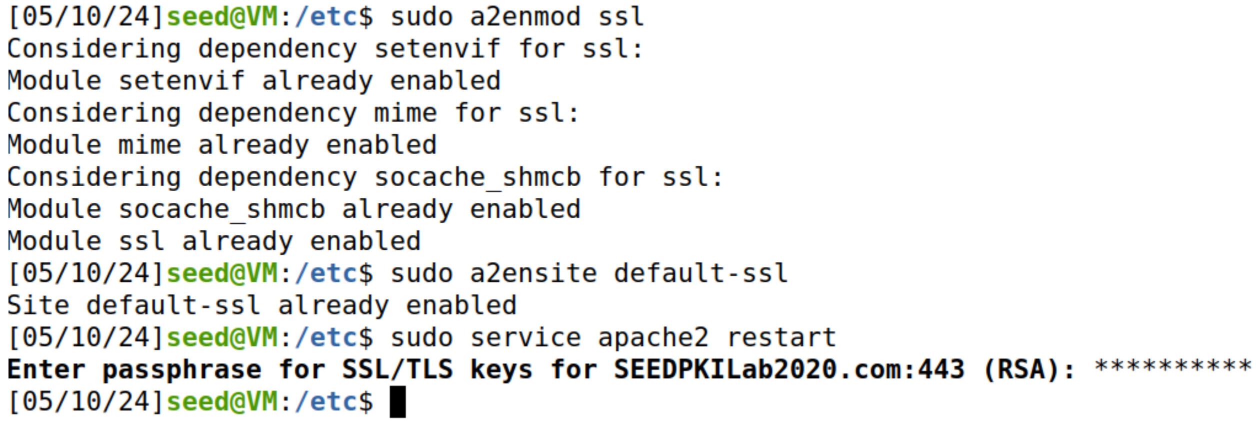
```
s_server -cert server.pem -www -accept 4433
Secure Renegotiation IS NOT supported
Ciphers supported in s_server binary
TLSv1.3    :TLS_AES_256_GCM_SHA384    TLSv1.3    :TLS_CHACHA20_POLY1305_SHA256
TLSv1.3    :TLS_AES_128_GCM_SHA256    TLSv1.2    :ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2    :DHE-RSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-RSA-CHACHA20-POLY1305
TLSv1.2    :DHE-RSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2    :DHE-RSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-ECDSA-AES256-SHA384 TLSv1.2    :ECDHE-RSA-AES256-SHA384
TLSv1.2    :DHE-RSA-AES256-SHA256     TLSv1.2    :ECDHE-ECDSA-AES128-SHA256
TLSv1.2    :ECDHE-RSA-AES128-SHA256   TLSv1.2    :DHE-RSA-AES128-SHA256
```

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

I add a new block to /etc/apache2/sites-available/default-ssl.conf using vim, in order to have apache recognize my server files and certificates

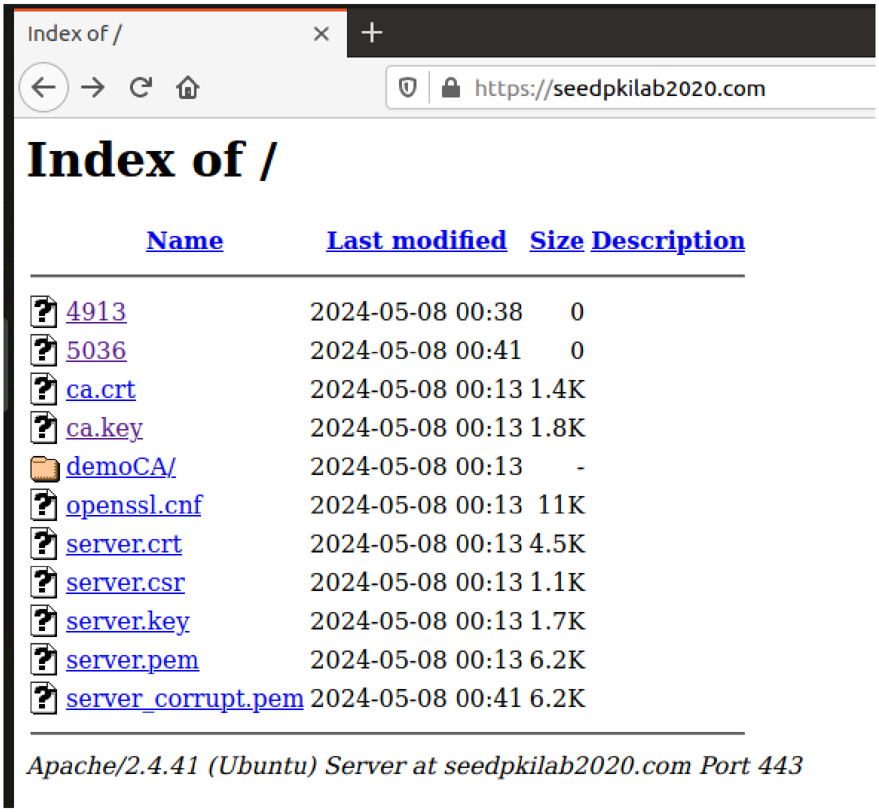


```
</VirtualHost>

<VirtualHost *:443>
ServerName SEEDPKILab2020.com
DocumentRoot /home/seed/Desktop/Shared_Desktop/HW5
DirectoryIndex index.html
SSLEngine On
SSLCertificateFile /home/seed/Desktop/Shared_Desktop/HW5/server.crt
SSLCertificateKeyFile /home/seed/Desktop/Shared_Desktop/HW5/server.pem
</VirtualHost>

<Directory /home/seed/Desktop/Shared_Desktop/HW5>
         Options Indexes FollowSymLinks
         AllowOverride None
         Require all granted
</Directory>
```



```
[05/10/24]seed@VM:/etc$ 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
[05/10/24]seed@VM:/etc$ sudo a2ensite default-ssl
Site default-ssl already enabled
[05/10/24]seed@VM:/etc$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): **********
[05/10/24]seed@VM:/etc$
```

After running apache2 restart, the url I set in the default.conf file (SEEDPKILab2020.com) directs me to my website. Since I did not specify an index.html, it just shows the directory of the server and the files within it



Index of /

Name	Last modified	Size	Description
4913	2024-05-08 00:38	0	
5036	2024-05-08 00:41	0	
ca.crt	2024-05-08 00:13	1.4K	
ca.key	2024-05-08 00:13	1.8K	
demoCA/	2024-05-08 00:13	-	
openssl.cnf	2024-05-08 00:13	11K	
server.crt	2024-05-08 00:13	4.5K	
server.csr	2024-05-08 00:13	1.1K	
server.key	2024-05-08 00:13	1.7K	
server.pem	2024-05-08 00:13	6.2K	
server_corrupt.pem	2024-05-08 00:41	6.2K	

Apache/2.4.41 (Ubuntu) Server at seedpkilab2020.com Port 443