Lab Report — Securing Apache Web Server with SSL/TLS

NB: I still can't install Ubuntu successfully . I have windows 10 older version. After trying every possible path, I couldn't solve the issue. So I created a virtual machine using an Ubuntu distribution inside Oracle VirtualBox.

Overview:

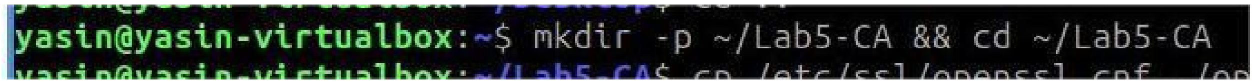
In this lab, I created my own CA and used it to issue certificates for my local Apache sites.

Tools Used

- **Ubuntu Linux**
- **Apache2 Web Server**
- **OpenSSL**
- **Firefox**
- **Terminal & Text Editor (nano)**

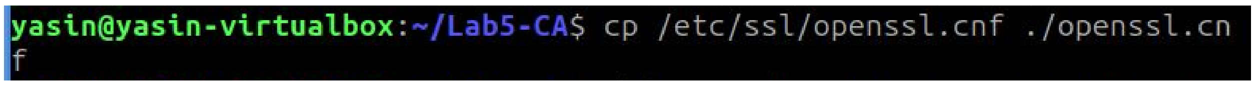
Step 1: Prepare the Workspace

I first created a folder to store all our certificate files:



```
yasin@yasin-virtualbox:~$ mkdir -p ~/Lab5-CA && cd ~/Lab5-CA
```

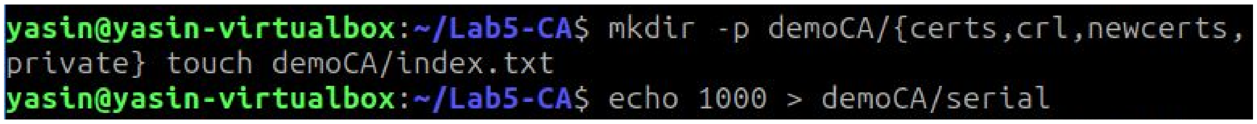
Then copied the OpenSSL config file:



```
yasin@yasin-virtualbox:~/Lab5-CA$ cp /etc/ssl/openssl.cnf ./openssl.cn
f
```

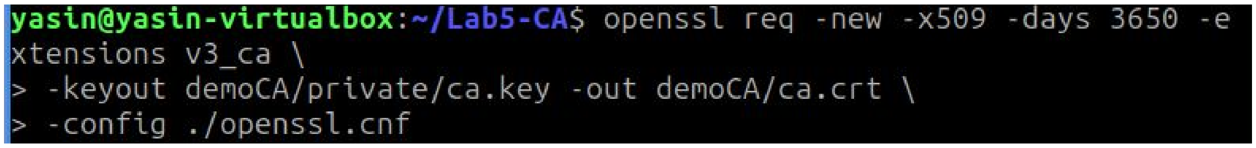
Step 2: Create a Root Certificate Authority (CA)

I created folders for my CA database:



```
yasin@yasin-virtualbox:~/Lab5-CA$ mkdir -p demoCA/{certs,crl,newcerts,
private} touch demoCA/index.txt
yasin@yasin-virtualbox:~/Lab5-CA$ echo 1000 > demoCA/serial
```

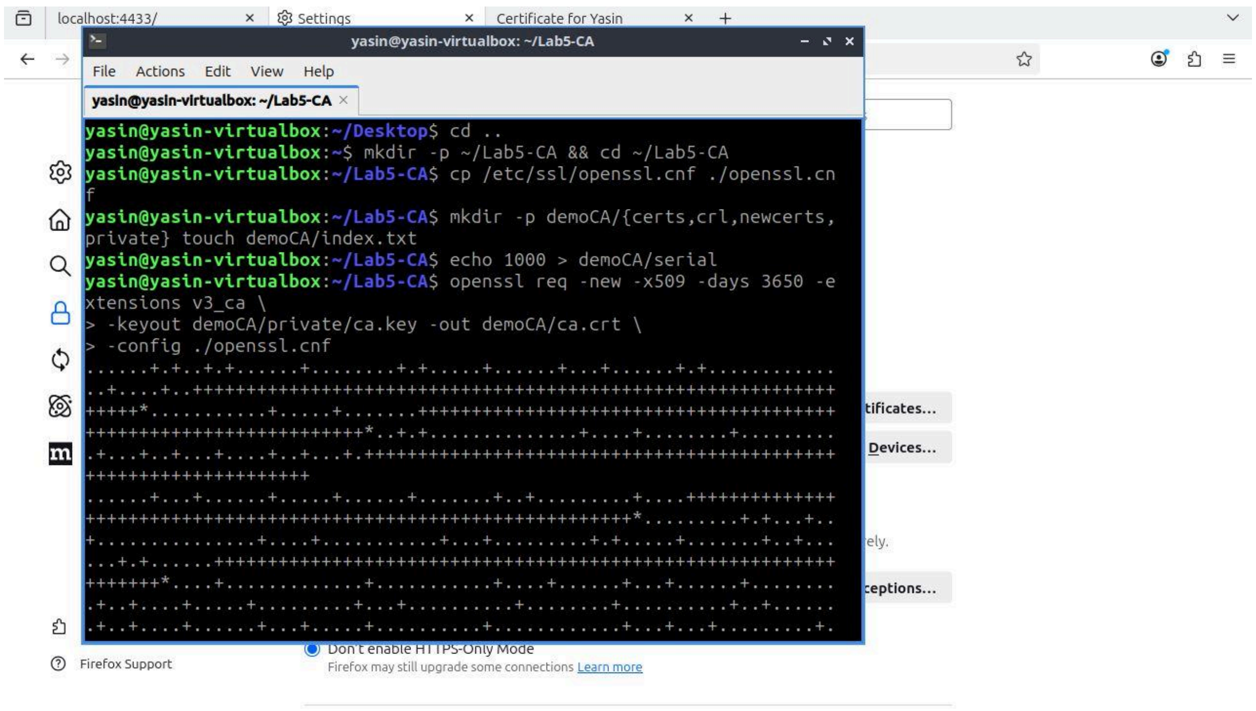
Then I generated a **self-signed CA certificate**:



```
yasin@yasin-virtualbox:~/Lab5-CA$ openssl req -new -x509 -days 3650 -e
xtensions v3_ca \
> -keyout demoCA/private/ca.key -out demoCA/ca.crt \
> -config ./openssl.cnf
```

This created two files:

- ➔ ca.key → private key (keep it secret)
- ➔ ca.crt → your CA certificate



```
yasin@yasin-virtualbox:~/Desktop$ cd ..
yasin@yasin-virtualbox:~$ mkdir -p ~/Lab5-CA && cd ~/Lab5-CA
yasin@yasin-virtualbox:~/Lab5-CA$ cp /etc/ssl/openssl.cnf ./openssl.cn
f
yasin@yasin-virtualbox:~/Lab5-CA$ mkdir -p demoCA/{certs,crl,newcerts,
private} touch demoCA/index.txt
yasin@yasin-virtualbox:~/Lab5-CA$ echo 1000 > demoCA/serial
yasin@yasin-virtualbox:~/Lab5-CA$ openssl req -new -x509 -days 3650 -e
xtensions v3_ca \
> -keyout demoCA/private/ca.key -out demoCA/ca.crt \
> -config ./openssl.cnf
.....+.+..+.+......+........+.+.....+......+...+......+.+............
..+....+..++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++*..........+.....+.......+++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++*..+.+..............+....+........+........
.+...+..+...+....+..+...+.++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++
......+...+......+.....+......+.......+..+.........+....++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++*.........+.+...+..
+..............+....+..........+...+.........+.+.....+.......+..+...
...+.+......++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++*....+.............+..........+....+......+...+.......+........
.+..+....+.....+.........+...+..........+.........+..........+..+.....
.+..+....+.......+...+.....+.........+..............+...+...+.........+.
```

Step 3: Create a Key and Certificate Request for the Server

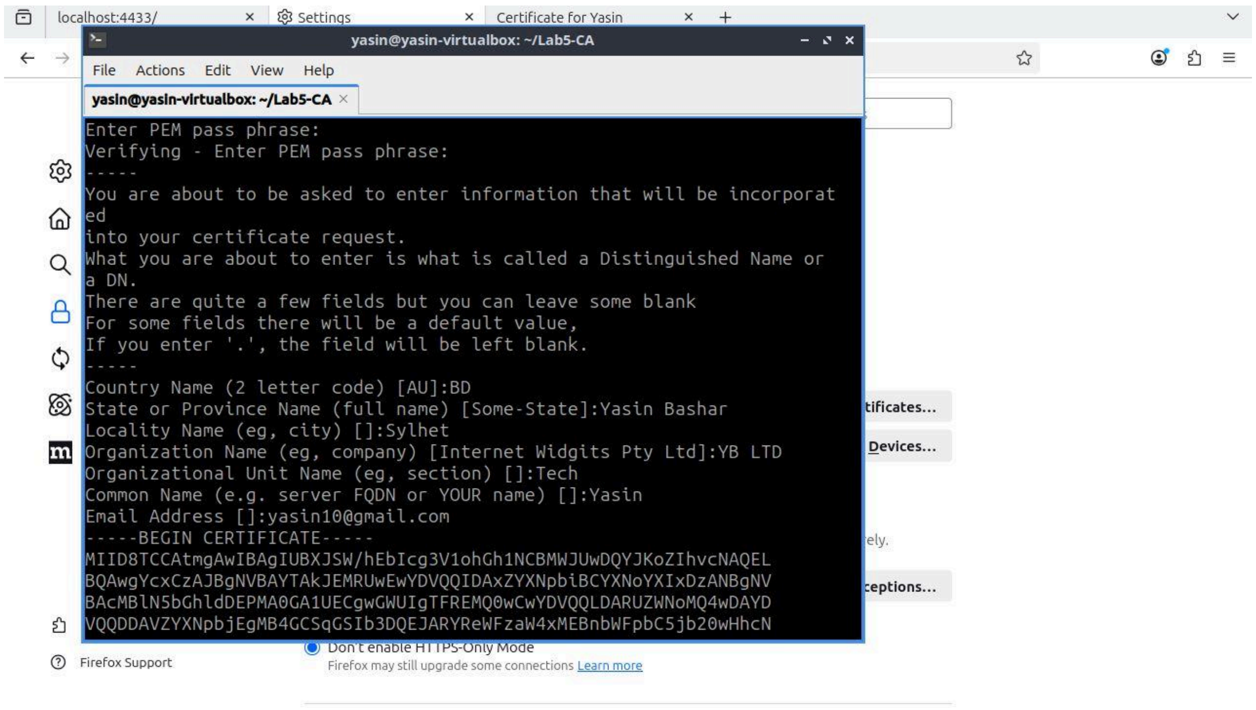
Next, I generated a private key for our website:

openssl genrsa -des3 -out server.key 2048

Then created a CSR (Certificate Signing Request):

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

This request will later be signed by our CA.



```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporat
ed
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]:BD
State or Province Name (full name) [Some-State]:Yasin Bashar
Locality Name (eg, city) []:Sylhet
Organization Name (eg, company) [Internet Widgits Pty Ltd]:YB LTD
Organizational Unit Name (eg, section) []:Tech
Common Name (e.g. server FQDN or YOUR name) []:Yasin
Email Address []:yasin10@gmail.com
-----BEGIN CERTIFICATE-----
MIID8TCCAtmgAwIBAgIUBXJSW/hEbIcg3V1ohGh1NCBMWJUwDQYJKoZIhvcNAQEL
BQAwgYcxCzAJBgNVBAYTAkJEMRUwEwYDVQQIDAxZYXNpbiBCYXNoYXIxDzANBgNV
BAcMBlN5bGhldDEPMA0GA1UECgwGWUIgTFREMQ0wCwYDVQQLDARUZWNoMQ4wDAYD
VQQDDAVZYXNpbjEgMB4GCSqGSIb3DQEJARYReWFzaW4xMEBnbWFpbC5jb20wHhcN
```

localhost:4433/
Settings
Certificate for Yasin
yasin@yasin-virtualbox: ~/Lab5-CA
File Actions Edit View Help
yasin@yasin-virtualbox: ~/Lab5-CA
Don't enable HTTPS-Only Mode
Firefox may still upgrade some connections Learn more
Firefox Support

Step 4: Add SAN (Subject Alternative Names)

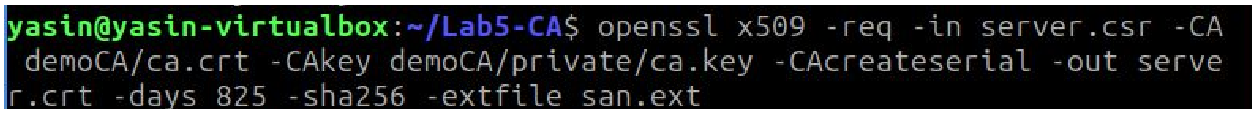
To make the certificate valid for multiple domains
I created a file named **san.ext:**

subjectAltName = DNS:demo.com, DNS:localhost, IP:127.0.0.1

This ensures browsers won't complain about mismatched hostnames.

Step 5: Sign the Certificate with Our CA

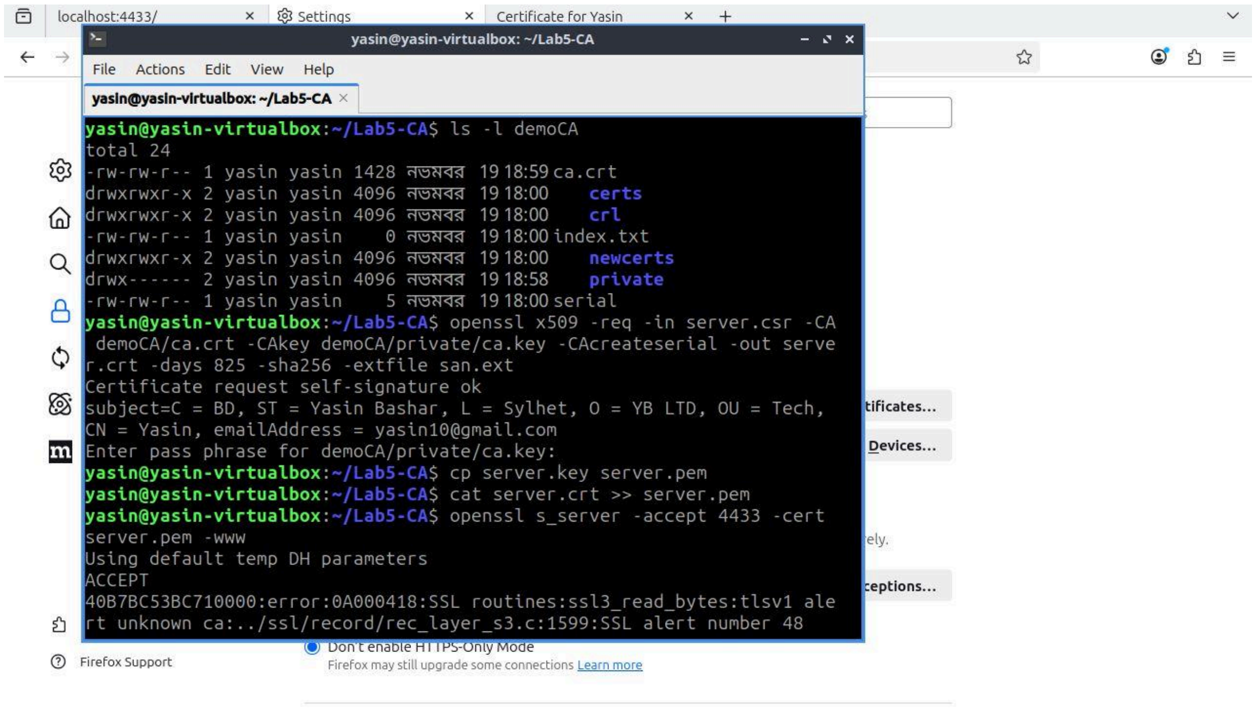
Use CA to sign the CSR and generate the actual certificate:



```
yasin@yasin-virtualbox:~/Lab5-CA$ openssl x509 -req -in server.csr -CA
 demoCA/ca.crt -CAkey demoCA/private/ca.key -CAcreateserial -out serve
r.crt -days 825 -sha256 -extfile san.ext
```

Now :

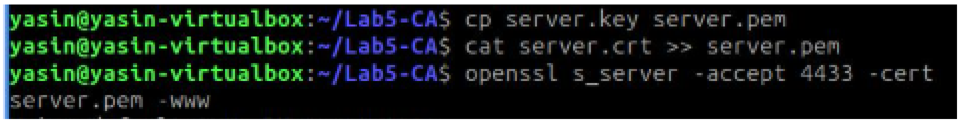
- server.crt → signed certificate
- server.key → server's private key



```
yasin@yasin-virtualbox:~/Lab5-CA$ ls -l demoCA
total 24
-rw-rw-r-- 1 yasin yasin 1428 নভেম্বর  19 18:59 ca.crt
drwxrwxr-x 2 yasin yasin 4096 নভেম্বর  19 18:00  certs
drwxrwxr-x 2 yasin yasin 4096 নভেম্বর  19 18:00  crl
-rw-rw-r-- 1 yasin yasin    0 নভেম্বর  19 18:00 index.txt
drwxrwxr-x 2 yasin yasin 4096 নভেম্বর  19 18:00  newcerts
drwx------ 2 yasin yasin 4096 নভেম্বর  19 18:58  private
-rw-rw-r-- 1 yasin yasin    5 নভেম্বর  19 18:00 serial
yasin@yasin-virtualbox:~/Lab5-CA$ openssl x509 -req -in server.csr -CA
 demoCA/ca.crt -CAkey demoCA/private/ca.key -CAcreateserial -out serve
r.crt -days 825 -sha256 -extfile san.ext
Certificate request self-signature ok
subject=C = BD, ST = Yasin Bashar, L = Sylhet, O = YB LTD, OU = Tech,
CN = Yasin, emailAddress = yasin10@gmail.com
Enter pass phrase for demoCA/private/ca.key:
yasin@yasin-virtualbox:~/Lab5-CA$ cp server.key server.pem
yasin@yasin-virtualbox:~/Lab5-CA$ cat server.crt >> server.pem
yasin@yasin-virtualbox:~/Lab5-CA$ openssl s_server -accept 4433 -cert
server.pem -www
Using default temp DH parameters
ACCEPT
40B7BC53BC710000:error:0A000418:SSL routines:ssl3_read_bytes:tlsv1 ale
rt unknown ca:../ssl/record/rec_layer_s3.c:1599:SSL alert number 48
```

Step 6: Test HTTPS with OpenSSL

Before using Apache, i tested certificate with OpenSSL's built-in web server:

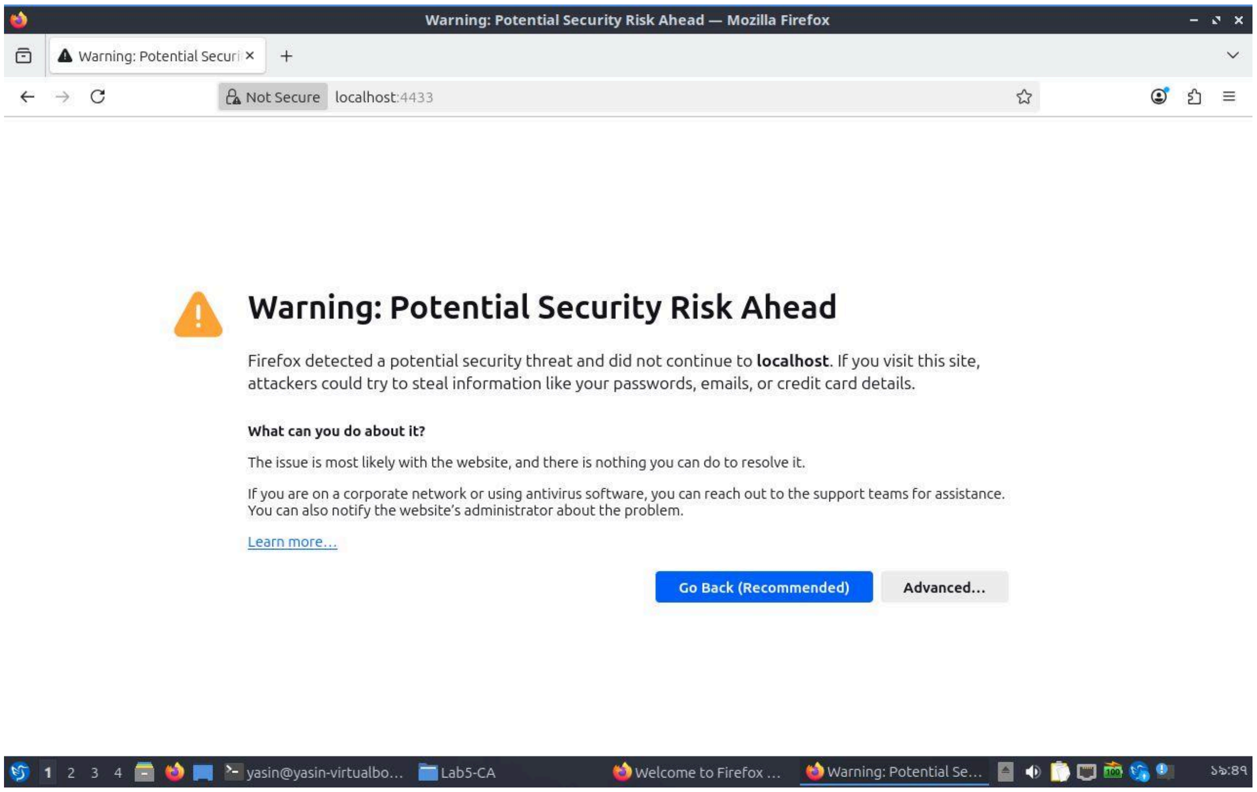


```
yasin@yasin-virtualbox:~/Lab5-CA$ cp server.key server.pem
yasin@yasin-virtualbox:~/Lab5-CA$ cat server.crt >> server.pem
yasin@yasin-virtualbox:~/Lab5-CA$ openssl s_server -accept 4433 -cert
server.pem -www
```

Then opened a browser and went to:

https://localhost:4433

At first, it showed a warning because the browser didn't yet trust our CA.


Warning: Potential Security Risk Ahead — Mozilla Firefox

Not Secure localhost:4433

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **localhost**. 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 antivirus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

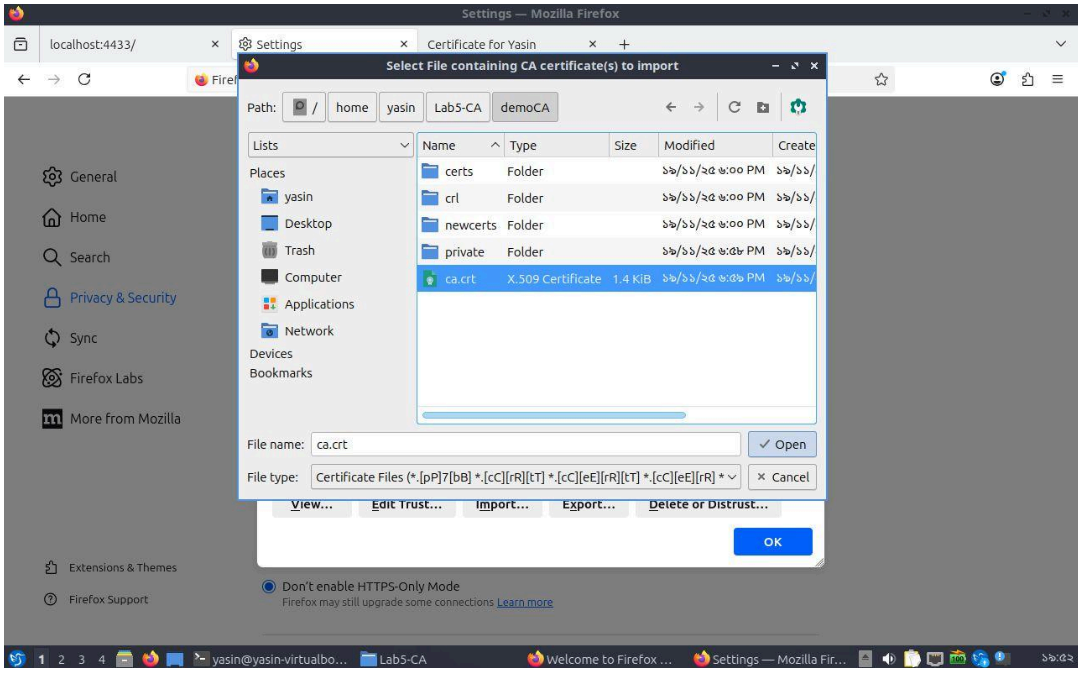
Learn more...

Step 7: Import Root CA into Browser

We added our CA to Firefox so it trusts our certificates:

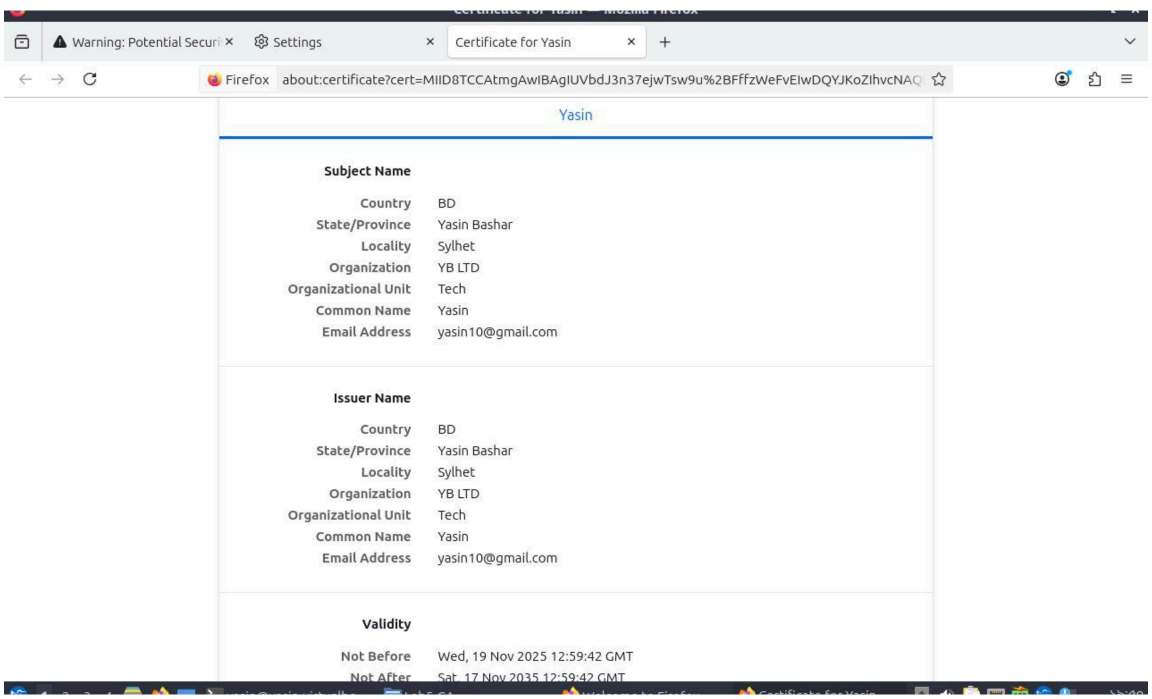
In Firefox:

1. Go to **Settings** → **Privacy & Security** → **Certificates** → **View Certificates**



2. Click **Import…** and select demoCA/ca.crt
3. Check **Trust this CA to identify websites**

Certificate details in Firefox browser:


Warning: Potential Securi
Settings
Certificate for Yasin
Firefox about:certificate?cert=MIID8TCCAtmgAwIBAgIUVbdJ3n37ejwTsw9u%2BFffzWeFvEIwDQYJKoZIhvcNAQ
Yasin

Subject Name

Country	BD
State/Province	Yasin Bashar
Locality	Sylhet
Organization	YB LTD
Organizational Unit	Tech
Common Name	Yasin
Email Address	yasin10@gmail.com

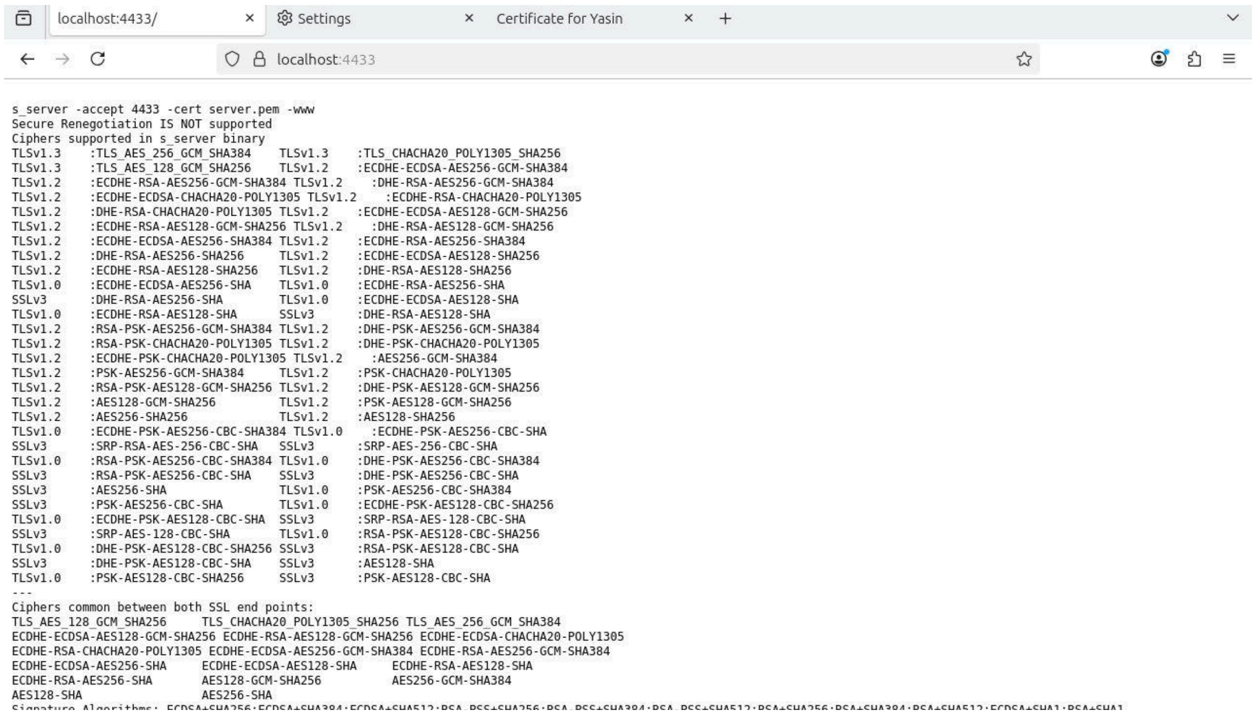
Issuer Name

Country	BD
State/Province	Yasin Bashar
Locality	Sylhet
Organization	YB LTD
Organizational Unit	Tech
Common Name	Yasin
Email Address	yasin10@gmail.com

Validity

Not Before	Wed, 19 Nov 2025 12:59:42 GMT
Not After	Sat, 17 Nov 2035 12:59:42 GMT

After that, refreshing the site showed a **padlock icon** — meaning HTTPS is working and trusted.



```
s_server -accept 4433 -cert server.pem -www
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
TLSv1.2    :ECDHE-PSK-CHACHA20-POLY1305 TLSv1.2    :AES256-GCM-SHA384
TLSv1.2    :PSK-AES256-GCM-SHA384     TLSv1.2    :PSK-CHACHA20-POLY1305
TLSv1.2    :RSA-PSK-AES128-GCM-SHA256 TLSv1.2    :DHE-PSK-AES128-GCM-SHA256
TLSv1.2    :AES128-GCM-SHA256         TLSv1.2    :PSK-AES128-GCM-SHA256
TLSv1.2    :AES256-SHA256             TLSv1.2    :AES128-SHA256
TLSv1.0    :ECDHE-PSK-AES256-CBC-SHA384 TLSv1.0    :ECDHE-PSK-AES256-CBC-SHA
SSLv3      :SRP-RSA-AES-256-CBC-SHA   SSLv3      :SRP-AES-256-CBC-SHA
TLSv1.0    :RSA-PSK-AES256-CBC-SHA384 TLSv1.0    :DHE-PSK-AES256-CBC-SHA384
SSLv3      :RSA-PSK-AES256-CBC-SHA    SSLv3      :DHE-PSK-AES256-CBC-SHA
SSLv3      :AES256-SHA                TLSv1.0    :PSK-AES256-CBC-SHA384
SSLv3      :PSK-AES256-CBC-SHA        TLSv1.0    :ECDHE-PSK-AES128-CBC-SHA256
TLSv1.0    :ECDHE-PSK-AES128-CBC-SHA  SSLv3      :SRP-RSA-AES-128-CBC-SHA
SSLv3      :SRP-AES-128-CBC-SHA       TLSv1.0    :RSA-PSK-AES128-CBC-SHA256
TLSv1.0    :DHE-PSK-AES128-CBC-SHA256 SSLv3      :RSA-PSK-AES128-CBC-SHA
SSLv3      :DHE-PSK-AES128-CBC-SHA    SSLv3      :AES128-SHA
TLSv1.0    :PSK-AES128-CBC-SHA256     SSLv3      :PSK-AES128-CBC-SHA
---
Ciphers common between both SSL end points:
TLS_AES_128_GCM_SHA256     TLS_CHACHA20_POLY1305_SHA256 TLS_AES_256_GCM_SHA384
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA     ECDHE-ECDSA-AES128-SHA     ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES256-SHA       AES128-GCM-SHA256          AES256-GCM-SHA384
AES128-SHA                 AES256-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
```

Step 8: Configure Apache for HTTPS

We copied the certificate and key to system folders:

```
sudo cp server.crt /etc/ssl/certs/demo_com.crt
```

```
sudo cp server.key /etc/ssl/private/demo_com.key
```

Enabled SSL module:

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

Then edited the Apache virtual host file:

```
sudo nano /etc/apache2/sites-available/demo.com.conf
```

Added this block:

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerAdmin admin@example.com
    ServerName example.com
    DocumentRoot /var/www/example.com/html

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/example_com.crt
    SSLCertificateKeyFile /etc/ssl/private/example_com.key

    ErrorLog ${APACHE_LOG_DIR}/example_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/example_ssl_access.log combined
</VirtualHost>
</IfModule>
```

Tested configuration:

```
sudo apache2ctl configtest
sudo systemctl restart apache2
```

Step 9: Verify HTTPS Website

Finally, we opened:

https://example.com

and saw the page load successfully with a **secure padlock** in the address bar.

Completed Tasks:

1. Created a self-signed CA.
2. Generated a server certificate.
3. Validated HTTPS connectivity using OpenSSL.
4. Added the CA to the browser’s trusted store.
5. Configured the Apache server to enable HTTPS.

Observations

➔ HTTPS encrypts communication, making it safer than HTTP.

➔ Self-signed CAs can be used for learning and local development.

➔ Once the CA is imported, browsers treat our certificates as trusted.

➔ Apache needs SSL enabled (`mod_ssl`) to handle HTTPS requests.

➔ The padlock icon proves encryption and certificate trust.