

**CSE487: Cybersecurity, Law and Ethics**

**[Summer 2022] Section: 03**

**Securing a networked system with Public Key Infrastructure Implementing Transport Layer Security on HTTP for https://connection**

**Project Report**

**Submitted to:**

Rashedul Amin Tuhin

Senior Lecturer

Department of Computer Science & Engineering

East West University

**Submitted by:**

|  |  |
| --- | --- |
| **Student ID** | **Student Name** |
| 2019-1-60-068 | Md. Shahadat Anik Sheikh |
| 2019-1-60-036 | Hasib Ar Rafiul Fahim |
| 2018-3-60-088 | Rashik Buksh Rafsan |

**Project Explanation:**

**Requirements:**

1. Configuration of Certification Authority AcmeCA
2. Configuration of the Web Server
3. Firewall configuration to allow necessary ports (53, 80, 443) only
4. DNS configuration for www.oursecureserver.com
5. CSR Configuration and Generation for the www. oursecureserver.com
6. Certification process (Verification and Certificate Generation from CSR)
7. Installation of the signed SSL certificate in the server
8. Making the system trust Acme-RootCA
9. Implementation of a simple file uploading page in the server.
10. Verifying the security of the connection by inspection (the padlock icon), and with Wireshark from another client.
11. Revoke the certificate issued to acmesecureserver.com from the AcmeCA and distribute the first CRL.
12. Verifying the revocation of previous certificate from the CRL (no padlock icon).

**Creating the Website:**

First, we created a simple website which uploads file. The files are static not stored anywhere.

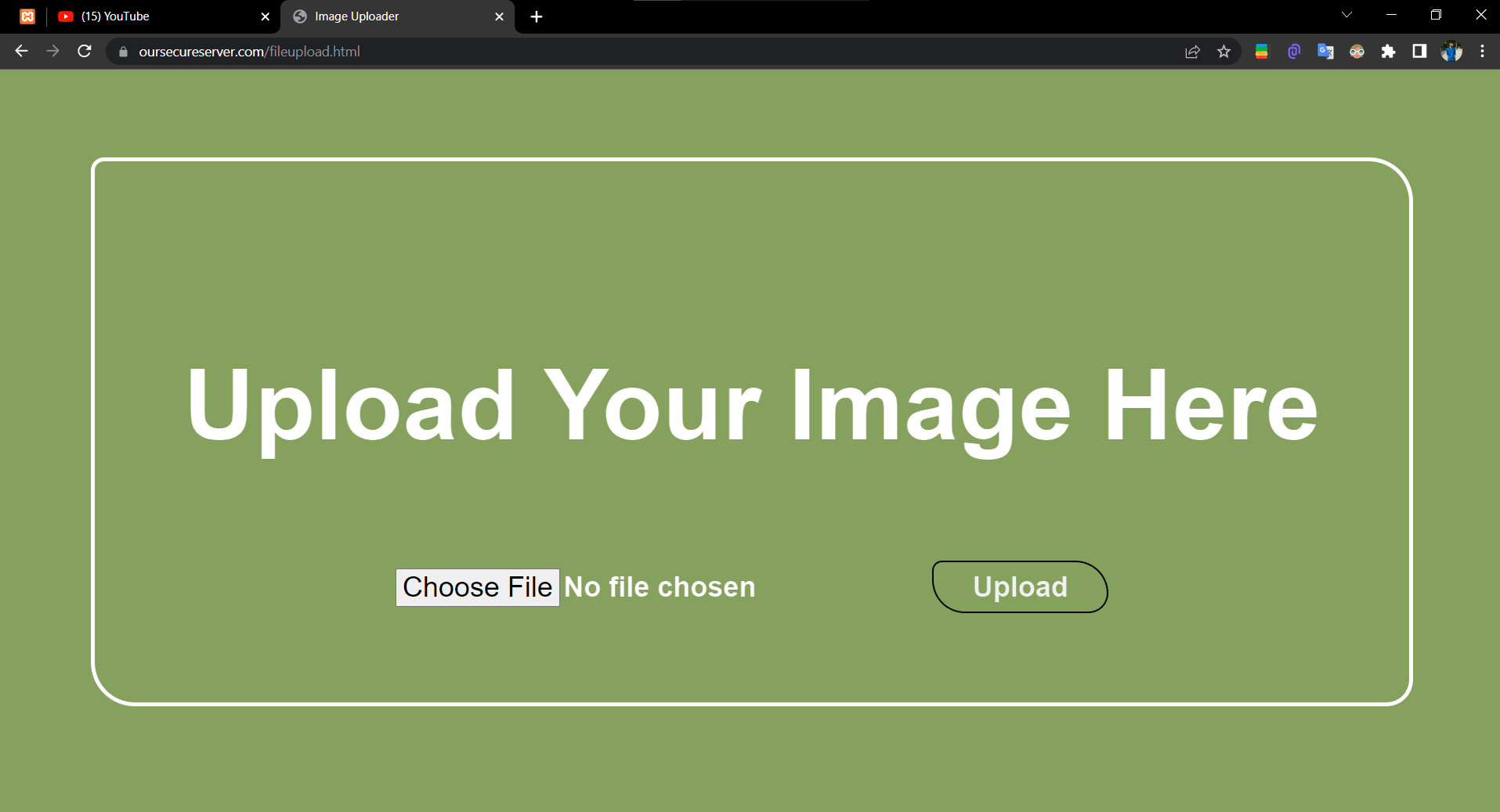


Figure 1: Created HTML website

For configuring of certification authority and implementing the transport layer security over http, we must follow some steps. All the steps are given below:

**Step-1: Configuring DNS:**

We configured DNS (Domain Name System) for our public IP. So, at first, we need to go to

Localdisk (C:)→Windows→System32→drivers→etc→hosts (open with notepad or any text editor)

And after going there we need to write the DNS configurations for our server and write the localhost Ip address and the server’s name [www.oursecureserver.com](http://www.oursecureserver.com)

hosts:

127.0.0.1 localhost

127.0.0.1 oursecureserver

127.0.0.1 [www.oursecureserver.com](http://www.oursecureserver.com)

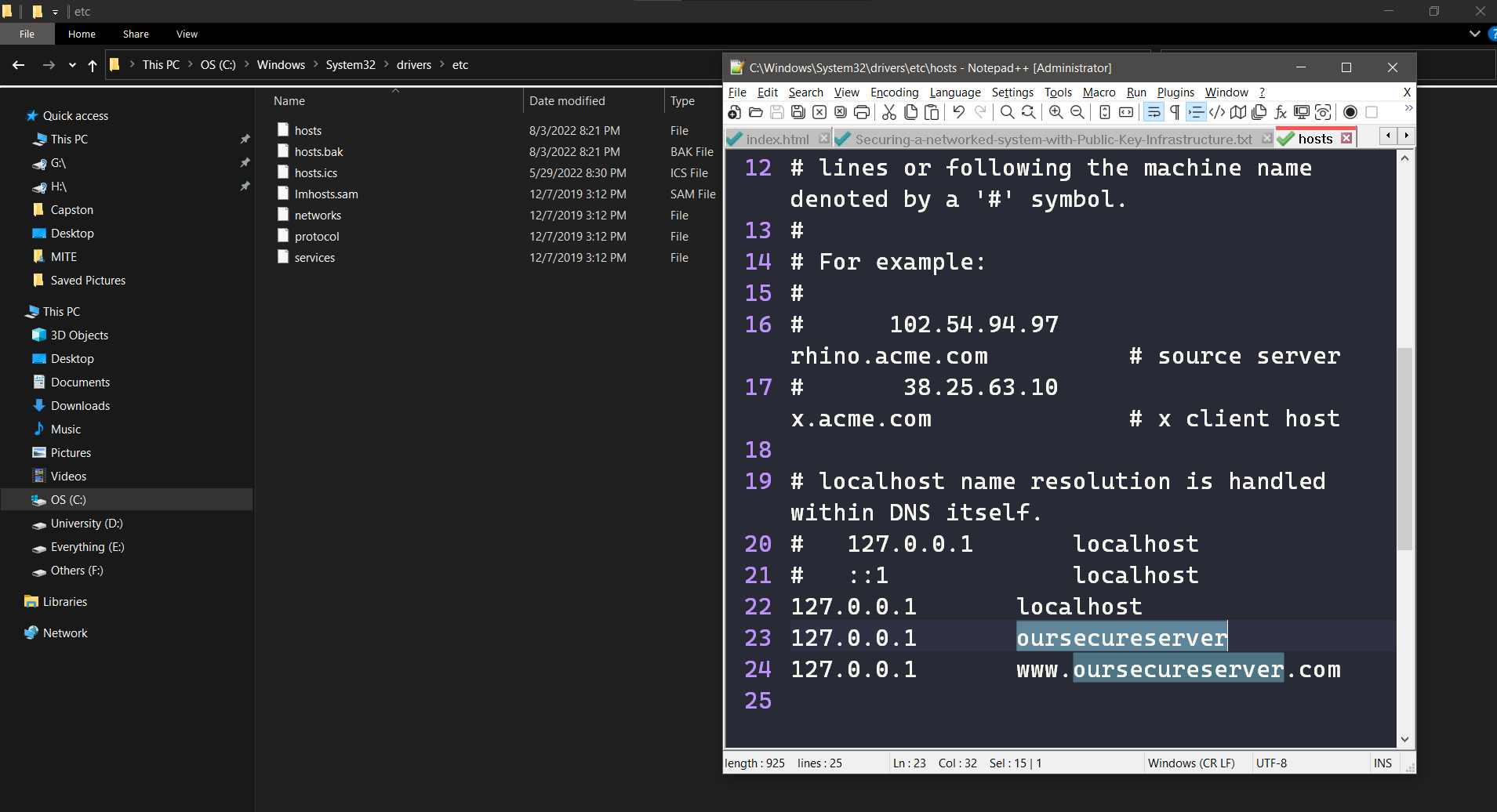


Figure 2: DNS for our public IP

**Step-2: Configuring of another DNS file in the xampp:**

Now, we need to configure another DNS file in the xampp for that we need to go to the document root for that: xampp→ apache→ conf→

**httpd.conf**:

DocumentRoot "C:/oursecureserver"

<Directory "C:/oursecureserver">

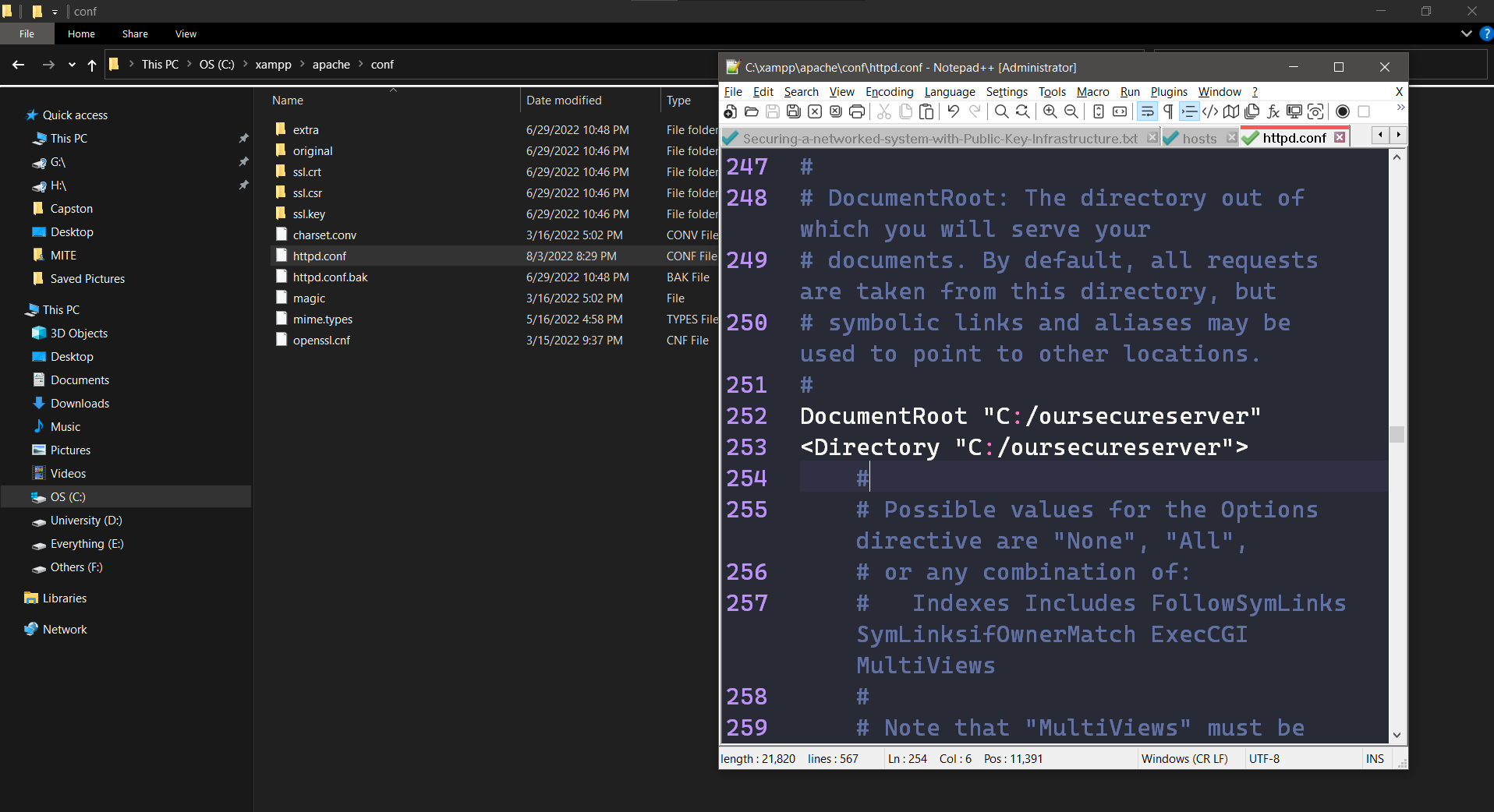


Figure 3: xampp dns setup

**Step-3: For openssl environment path configuration:**

Now, our website is up and running but it is not secured because there is no SSL certificate. In this part, we will create a signed certificate for our server. We used OpenSSL tool to generate private key and certificate request for our server. We used our DNS as the name for signing the certificate. Then, we used OpenSSL to generate a signed certificate for our corresponding DNS. The certificate request CSR (Certificate Signing Request) must be created with a key length of 2048 as the hash algorithm. It checks if DNS is owned by us or not. After successful check, it gives us the signed certificate.

First, we need to install OPENSSL in our Windows. For that, we need to run cmd as administrator. Then we used the following command, **choco install openssl**

And for openssl environment path we need to go to:

**set OPENSSL\_CONF=C:\xampp\apache\conf\openssl.cnf**

Now we need to go to C: by using cd command in the terminal after that we will go to

**cd xampp → cd apache→ cd bin → openssl.exe**

Then openssl will be on.

To generate a pair of private key and public Certificate Signing Request (CSR) for a web server, “server”, use the following command:

**req -newkey rsa:2048 -nodes -keyout server.key -out server.csr**

This creates two files. The file myserver.key contains a private key. The private key is used as input in the command to generate a Certificate Signing Request (CSR). We will now be asked to enter details to be entered into the CSR. What we are about to enter is what is called a Distinguished Name or a DN. **Common name:** [www.oursecureserver.com](http://www.oursecureserver.com)

**Password for creating certificates: anik123**

Country Name (2 letter code) [AU]: BD

State or Province Name (full name) [Some-State]: Dhaka

Locality Name (eg, city) []: Dhaka

Organization Name (eg, company) []: EWU

Organizational Unit Name (eg, section) []: CSE

Common Name (server name) []: www.oursecureserver.com

E-mail Address []: shahadatanik777@gmail.com

Please enter the following ‘extra’ attributes to be sent with your certificate request.

A challenge password []: (anik123)

Our CSR will now have been created.

To verify the certificate:

**~ x509 -signkey server.key -in server.csr -req -days 365 -out server.crt**

Now for creating the subroot CA certificate at first, we need to control c and again start the openssl otherwise we can get error because we are going to create another certificate for the subroot. **Common Name: AcmeCA** and it will ask for a **PEM password** which is **anik123**

**~ req -newkey rsa:2048 -keyout subrootCA.key -out subrootCA.csr**

Country Name (2 letter code) [AU]: BD

State or Province Name (full name) [Some-State]: Dhaka

Locality Name (eg, city) []: Dhaka

Organization Name (eg, company) []: EWU

Organizational Unit Name (eg, section) []: CSE

Common Name (server name) []: AcmeCA

E-mail Address []: shahadatanik777@gmail.com

Please enter the following ‘extra’ attributes to be sent with your certificate request.

A challenge password []: (anik123)

**~ x509 -signkey subrootCA.key -in subrootCA.csr -req days 365 -out subrootCA.crt**

Now for creating the root CA certificate at first, we need to control c and again start the openssl otherwise we can get error because we are going to create another certificate for the root. **Common Name: Acme-RootCA** and it will ask for a **PEM password which is anik123**

Country Name (2 letter code) [AU]: BD

State or Province Name (full name) [Some-State]: Dhaka

Locality Name (eg, city) []: Dhaka

Organization Name (eg, company) []: EWU

Organizational Unit Name (eg, section) []: CSE

Common Name (server name) []: Acme-RootCA

E-mail Address []: shahadatanik777@gmail.com

Please enter the following ‘extra’ attributes to be sent with your certificate request.

A challenge password []: (anik123)

Now we must open two **.ext** file because as we created the three certificates and now it’s the time for signing those certificates so the files are:

We need to go: **xampp→ apache→ bin**

**domain.ext:**

authorityKeyIdentifier = keyid,issuer

basicConstraints = CA: FALSE

subjectAltName = @alt\_names

[alt\_names]

DNS.1 =www.oursecureserver.com

DNS.2 =127.0.0.1

**root.ext:**

authorityKeyIdentifier = keyid,issuer

basicConstraints = CA: TRUE

subjectAltName = @alt\_names

[alt\_names]

DNS.1 =www.oursecureserver.com

DNS.2 =127.0.0.1

Now, Signing subrootCA certificate with rootCA certificate:

**~ x509 -req -CA rootCA.crt -CAkey rootCA.key -in subrootCA.csr -out subrootCA.crt -days 365 -CAcreateserial -extfile root.ext**

For checking the subrootCa certificate:

**~ x509 -text -noout -in subrootCA.crt**

**~ x509 -in subrootCA.crt -outform der -out subrootCA.der**

Now, Exporting the subrootCA key file in subrootCA pfx file:

**~ pkcs12 -inkey subrootCA.key -in subrootCA.crt -export -out subrootCA.pfx**

And it will ask for the **PEM pass which is anik123**

Signing server certificate with subrootCA certificate:

**~ x509 -req -CA subrootCA.crt -CAkey subrootCA.key -in server.csr -out server.crt -days 365 -CAcreateserial -extfile domain.ext**

**~ x509 -in server.crt -outform der -out server.der**

Exporting the server key file in the server .pfx file:

**~ pkcs12 -inkey server.key -in server.crt -export -out server.pfx**

Replacing the RSA encryption from the server and subrootCA key for setting the validity:

**~ rsa -in server.key -out server.key**

**~ rsa -in subrootCA.key -out subrootCA.key**

**Step-4: Installing Certificate:**

We opened **C drive → xampp → apache → bin** then we will see our certificates then we need to install it first the rootCA.cert in local **machine→trusted** root certificate **authorities→finish** then our installation will be completed in this way we need to install server.crt,subrootCA.pfx and the subrootCA.cert will be already installed by installing the rootCA.cert. After that now we need to copy the server.crt, server.csr and the server.key file and paste it to the **apache→conf** file into the ssl.crt, ssl.csr, ssl.key.

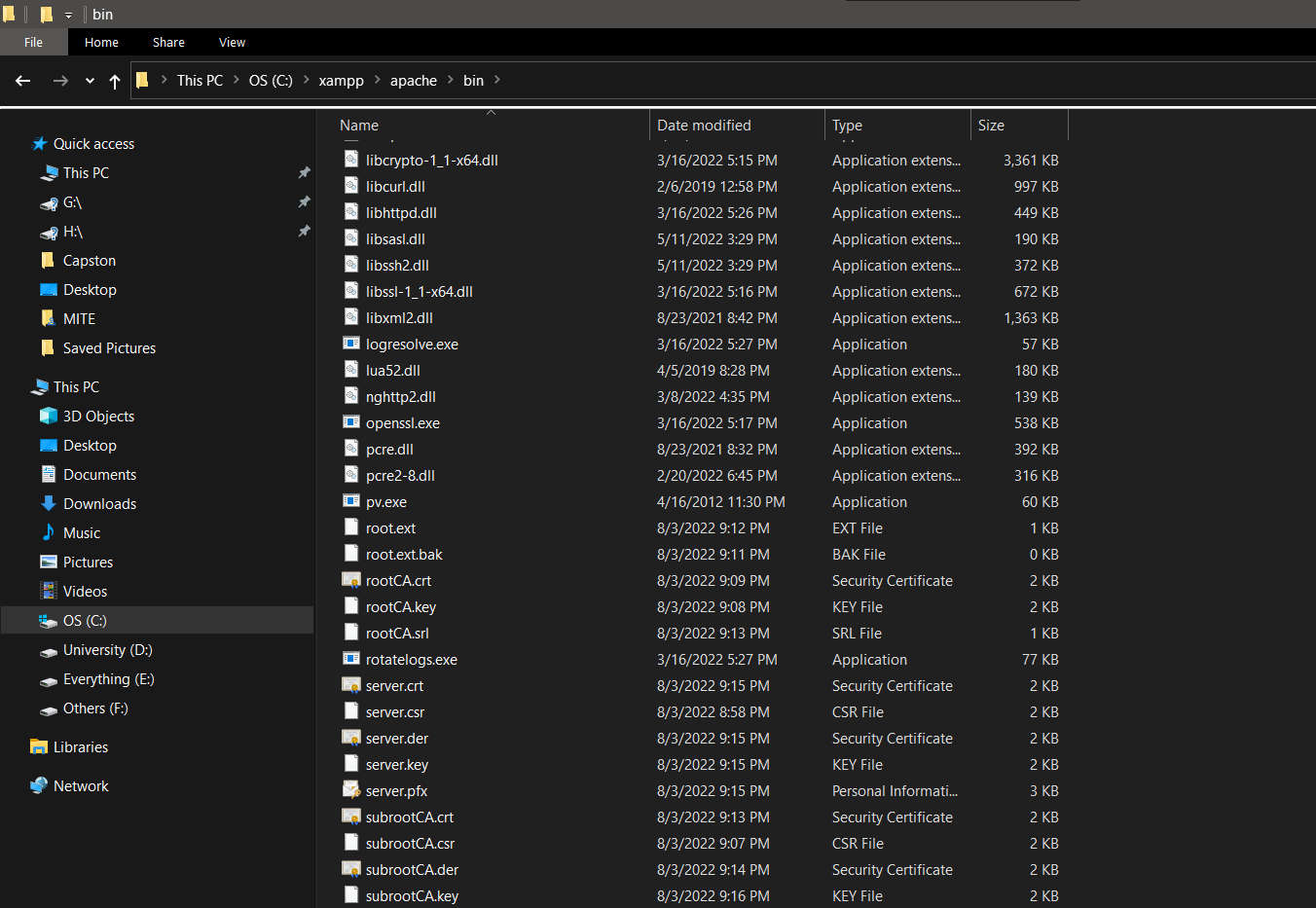


Figure 4: Certificate Install

For pasting the files, we need to go: **apache → conf**

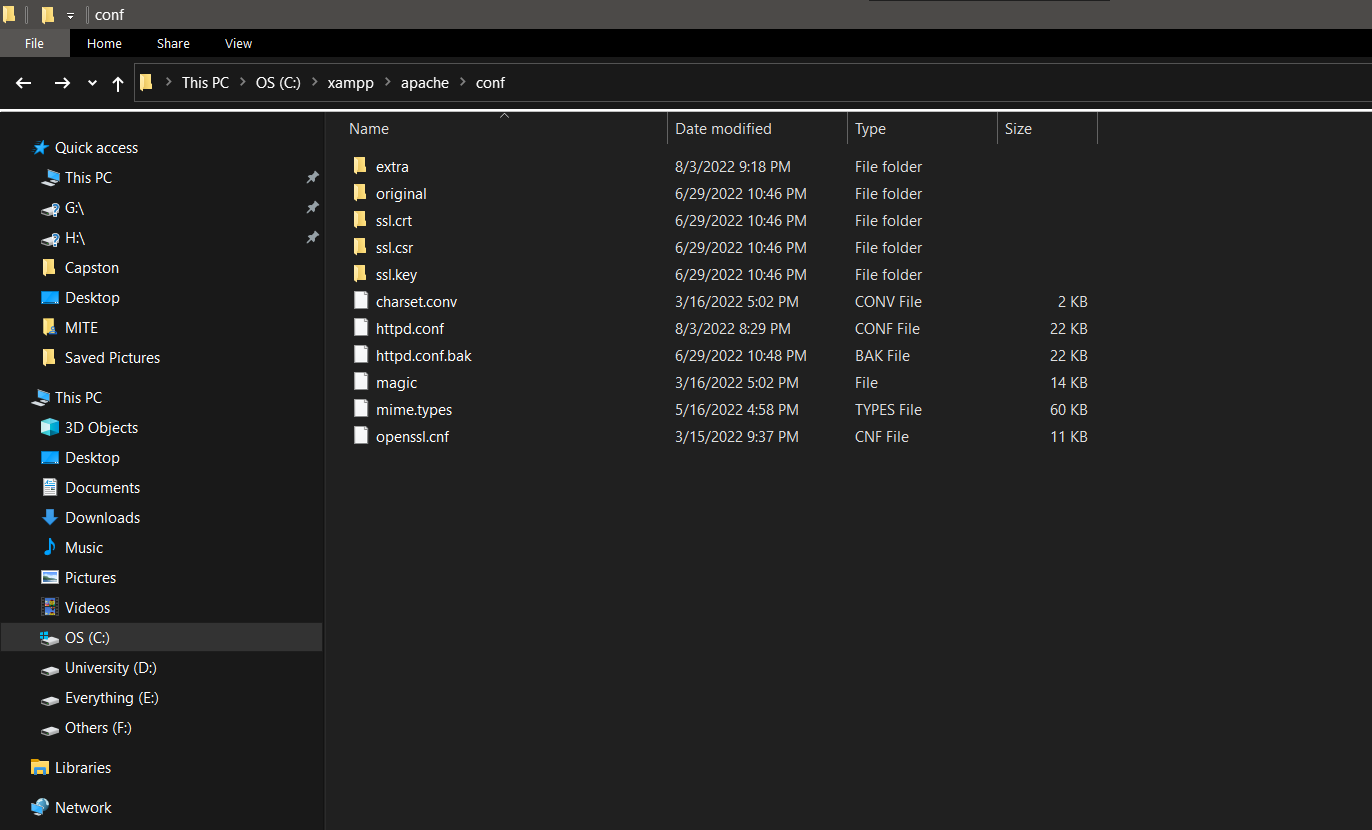


Figure 5: Pasting The Files

**Step5: Creating certificate:**

For that we need to go: **xampp → apache → conf → extra → httpd-vhosts.conf**

**Configuring httpd-vhosts:**

<VirtualHost \*:443>

DocumentRoot "C:/oursecureserver/"

ServerName oursecureserver

ServerAlias www.oursecureserver.com

SSLEngine on

SSLCertificateFile "conf/ssl.crt/server.crt"

SSLCertificateKeyFile "conf/ssl.key/server.key"

</VirtualHost>

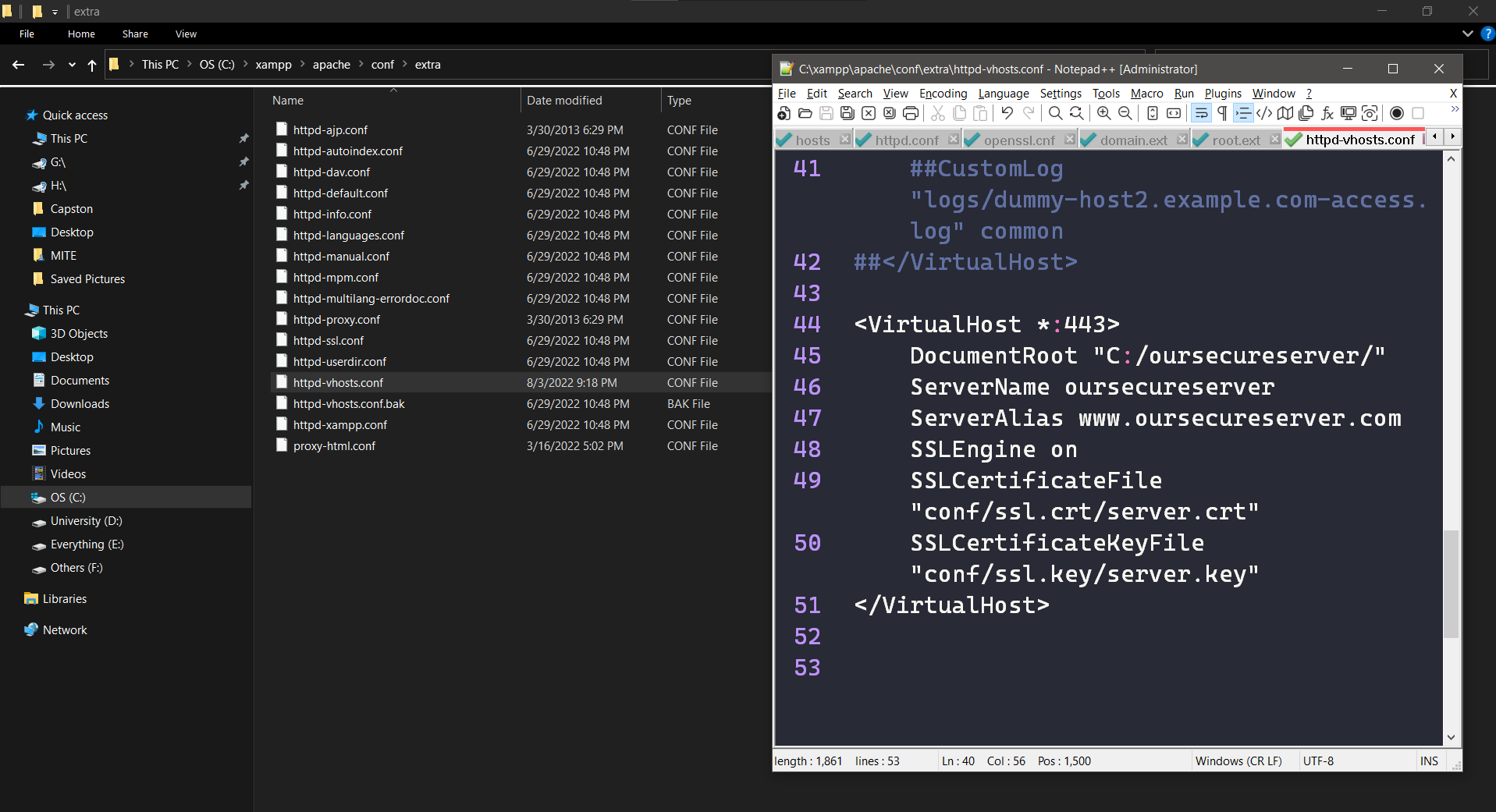


Figure 6: Configuring httpd-vhosts

**Firewall configuration to allow necessary ports (53, 80, 443) only necessary screenshots are given:**

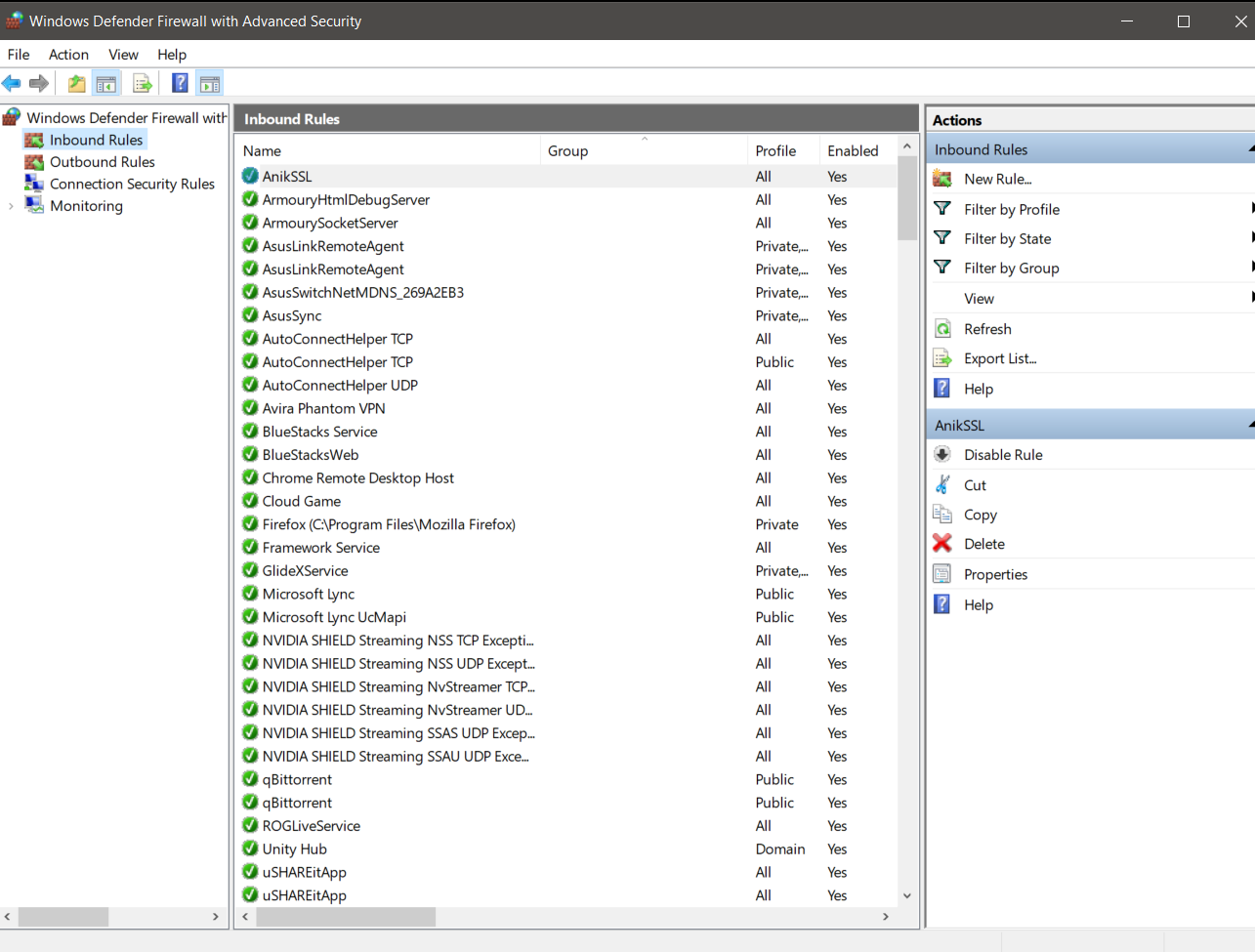
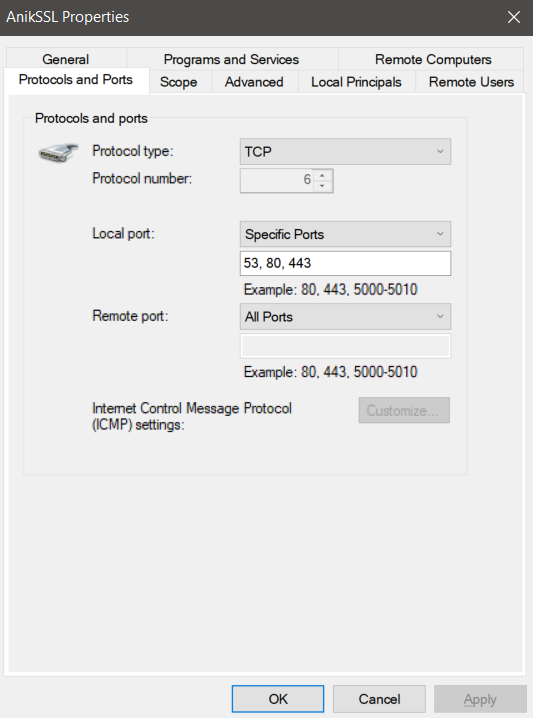
**** 

Figure 7: Firewall configuration

Finally, we will open our xampp and turn on Apache and go to our website which is running with SSL Certificate. We have also showed our certification path.

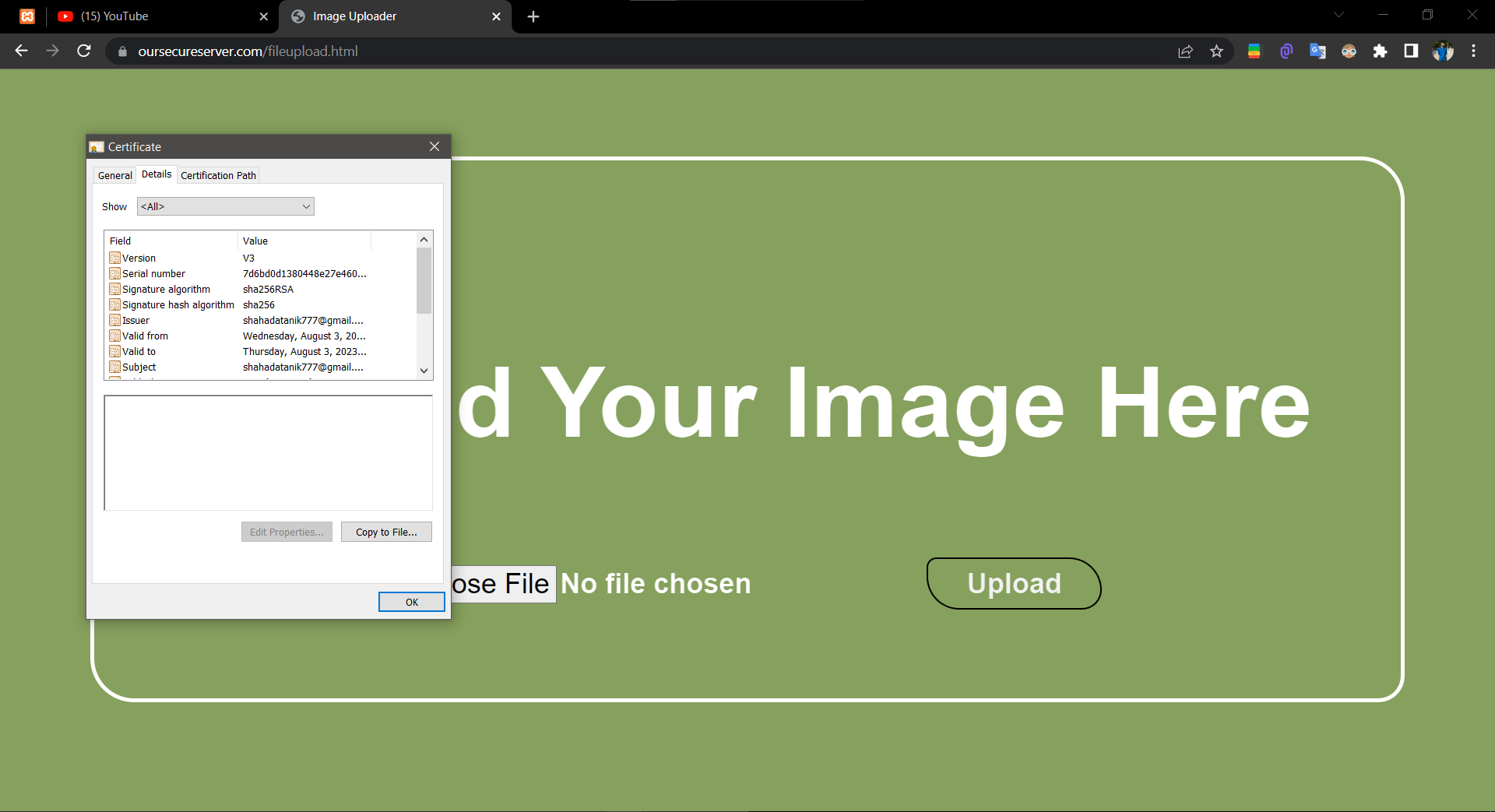


Figure 8: Certificate Info

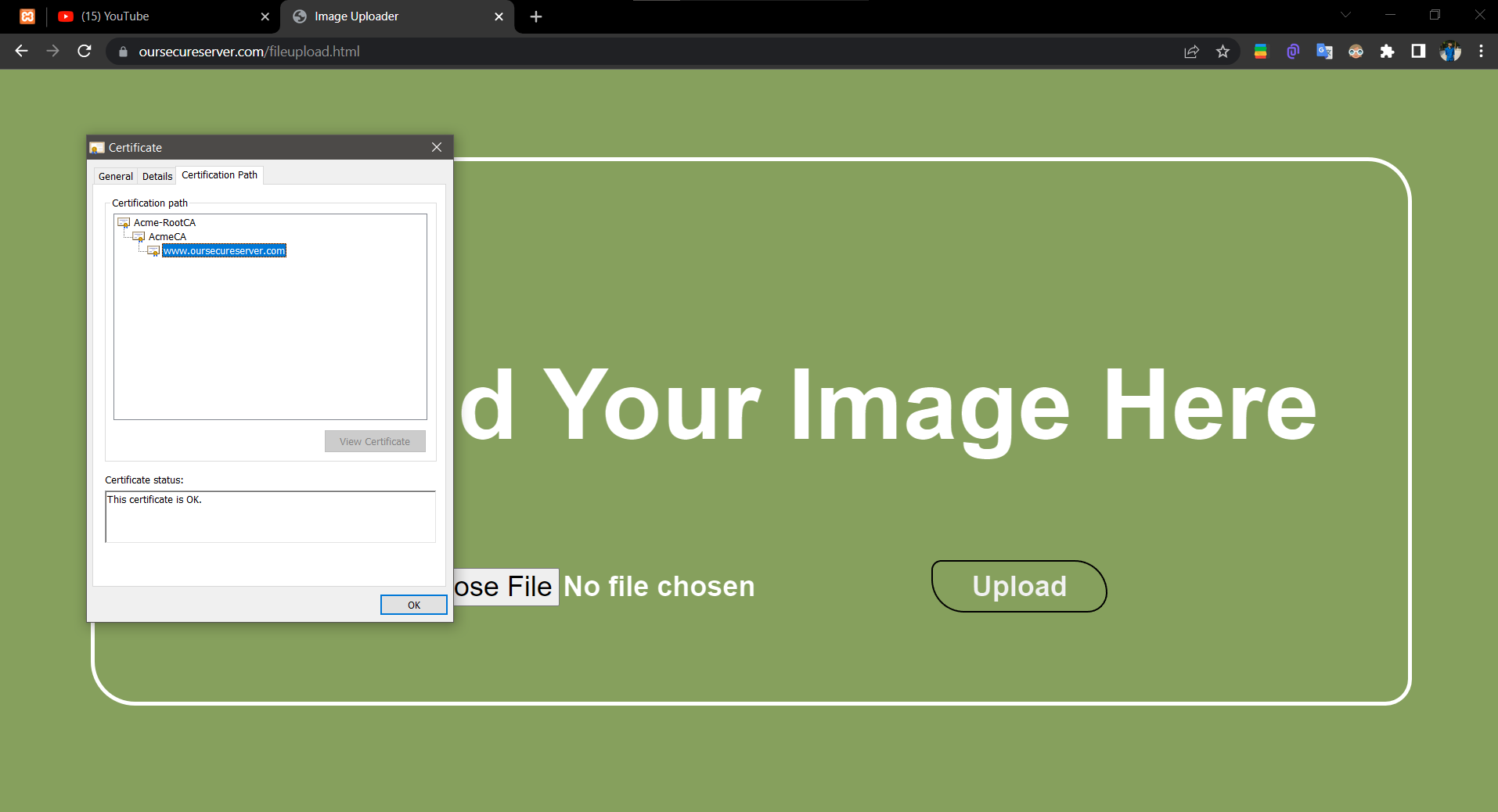
****

Figure 9: Certificate Path

**Step 6: Revocation of certificate:**

We must open three files in the bin folder which is **index.txt, serial.txt, crlnumber.txt** where the serial number will be stored.

Open **openssl.exe** to revoke the certificate issued to acmesecureserver.com from:

**AcmeCA** → **ca -config subrootCA.conf -revoke server.crt**

To generate revocation crl file:

**ca -config subrootCA.conf -gencrl -out rev.crl**

To see the revocation file in the form of text:

**crl -in rev.crl -noout -text**

**subrootCA.conf:**

[ca]

default\_ca = CA\_default

[CA\_default]

dir =C:/xampp/apache/bin

certs = $dir

crl\_dir = $dir

new\_certs\_dir = $dir

database = $dir/index.txt

serial = $dir/serial.txt

RANDFILE = $dir/private/. rand

private\_key = $dir/subrootCA.key

certificate = $dir/subrootCA.crt

crlnumber = $dir/crlnumber.txt

crl = $dir/crl/ca.crl

default\_crl\_days = 2

default\_md = sha256

name\_opt = ca\_default

cert\_opt = ca\_default

default\_days = 365

preserve = no

policy = policy\_loose

[ policy\_strict ]

countryName = supplied

stateOrProvinceName = supplied

organizationName = supplied

organizationalUnitName = optional

commonName = supplied

emailAddress = optional

[ policy\_loose ]

countryName = optional

stateOrProvinceName = optional

localityName = optional

organizationName = optional

organizationalUnitName = optional

commonName = supplied

emailAddress = optional

[ req ]

# Options for the req tool, man req.

default\_bits = 2048

distinguished\_name = req\_distinguished\_name

string\_mask = utf8only

default\_md = sha256

# Extension to add when the -x509 option is used.

x509\_extensions = v3\_ca

[ req\_distinguished\_name ]

countryName = Country Name (2 letter code)

stateOrProvinceName = State or Province Name

localityName = Locality Name

0.organizationName = Organization Name

organizationalUnitName = Organizational Unit Name

commonName = Common Name

emailAddress = Email Address

countryName\_default = BD

stateOrProvinceName\_default = Dhaka

0.organizationName\_default = Acme

[ v3\_ca ]

# Extensions to apply when createing root ca

# Extensions for a typical CA, man x509v3\_config

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid:always,issuer

basicConstraints = critical, CA:true

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ v3\_intermediate\_ca ]

# Extensions to apply when creating intermediate or sub-ca

# Extensions for a typical intermediate CA, same man as above

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid:always,issuer

#pathlen:0 ensures no more sub-ca can be created below an intermediate

basicConstraints = critical, CA:true, pathlen:0

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

crlDistributionPoints = @crl\_dist\_points

[ server\_cert ]

# Extensions for server certificates

basicConstraints = CA:FALSE

nsComment = "OpenSSL Generated Server Certificate"

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer:always

keyUsage = nonRepudiation, digitalSignature, keyEncipherment

extendedKeyUsage = serverAuth

subjectAltName = @alt\_names

[alt\_names]

DNS.1 = www.oursecureserver.com

DNS.2 = 127.0.0.1

By this the revocation will be done and we will see the time and date of the revocation in the terminal.