EAST WEST UNIVERSITY

SUBJECT: CYBERSECURITY, LAW, AND ETHICS

CSE487 || SEC3 || SUMMER-22

INSTRUCTOR

Rashedul Amin Tuhin

Senior Lecturer | Assistant Proctor

Department of Computer Science & Engineering

ResearchGate Google Scholar

SUBMITTED BY

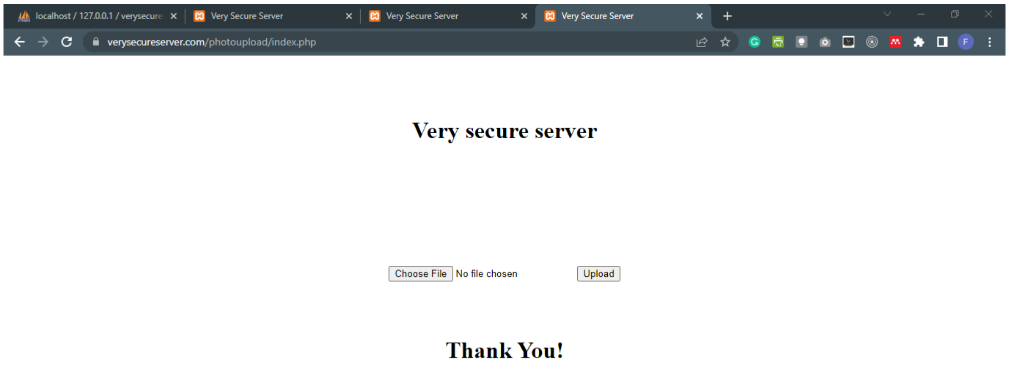
Fahadul Islam	**ID: 2018-2-60-102**
Sabbir Ahmed	**ID: 2018-2-60-100**
Tauhid Hossain Ayon	**ID: 2019-1-60-106**

Mini Project-1: Securing a networked system with Public Key Infrastructure

Implementing Transport Layer Security on HTTP for https:// connection

1. Photo App Demonstration:

We initially designed this home page for the Secure File Transfer System. The Choose File button in this case is used to select an image from the computer. The image is then uploaded to the MySQL database server by clicking the Upload button.



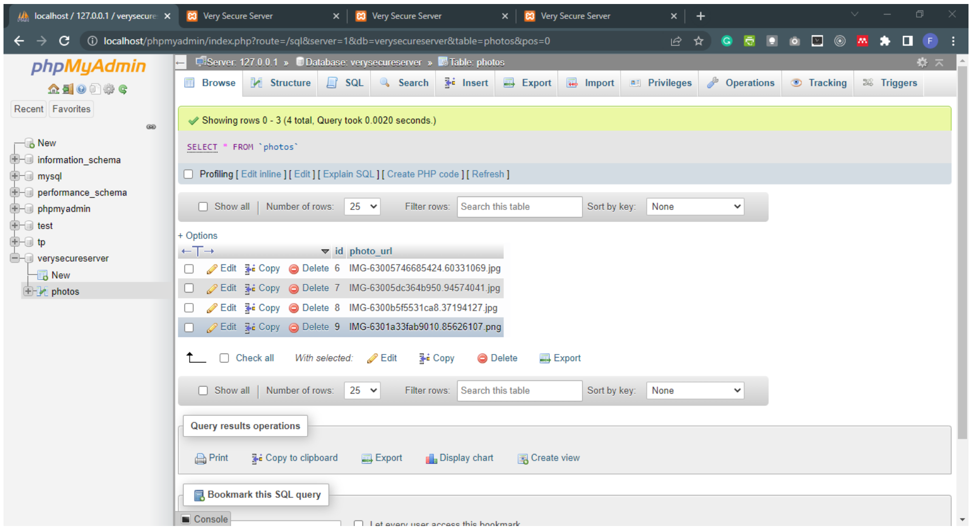
verysecureserver.com/photoupload/index.php

Very secure server

Choose File No file chosen Upload

Thank You!

Connected the website with the MySQL database server after creating the home page. and used MySQL to construct a "Photos" database that holds the photo URL.



localhost/phpmyadmin/index.php?route=/sql&server=1&db=verysecureserver&table=photos&pos=0

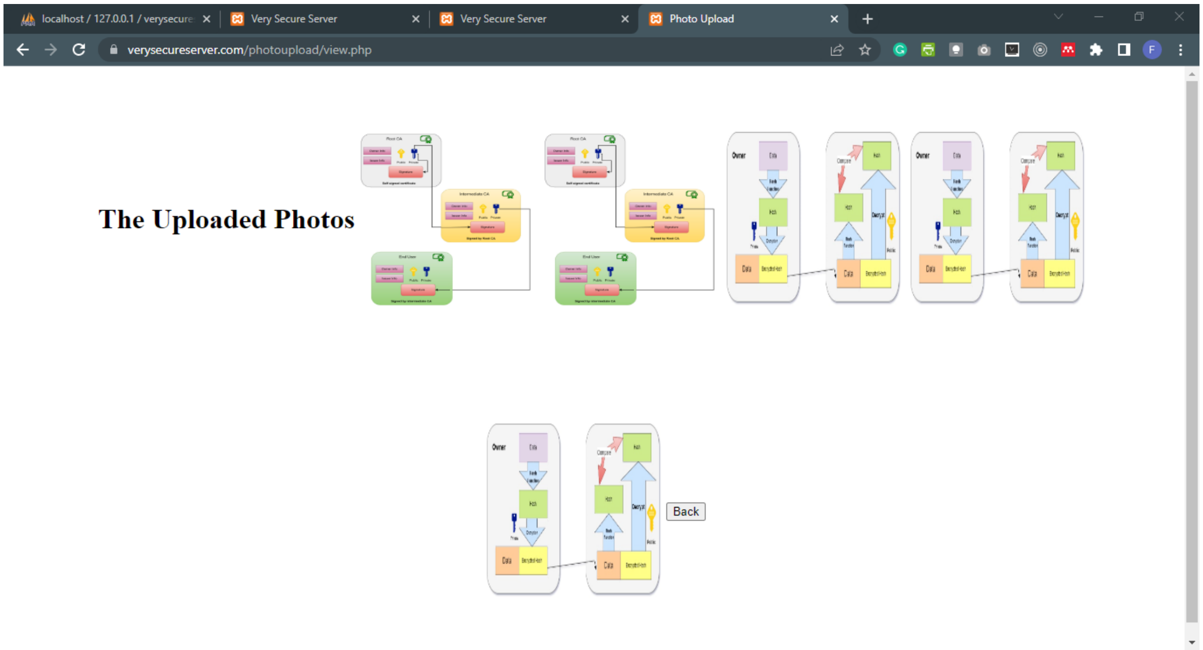
Server: 127.0.0.1 » Database: verysecureserver » Table: photos

Showing rows 0 - 3 (4 total, Query took 0.0020 seconds.)

```
SELECT * FROM `photos`
```

id	photo_url
6	IMG-63005746685424.60331069.jpg
7	IMG-63005dc364b950.94574041.jpg
8	IMG-6300b5f5531ca8.37194127.jpg
9	IMG-6301a33fab9010.85626107.png

After the uploading of the image, this system has a view page, which shows the images. The images have been extracted from the MySQL database. Also, this page has a back button that will redirect to the home page.


localhost / 127.0.0.1 / verysecure
Very Secure Server
Very Secure Server
Photo Upload
verysecureserver.com/photoupload/view.php
The Uploaded Photos
Back

2. SSL Certificate Implementation:

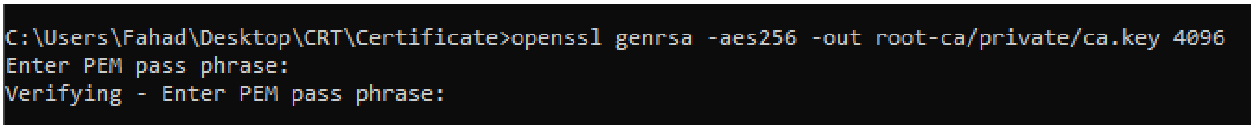
To secure the web page, we need to create a folder in the **"C:\xampp\apache"** path, in this case, the name of the folder is **"crt".** Then I used the OpenSSL command to create the certificate.

openssl req -new -newkey rsa:2048 -x509 -days 365 -nodes -subj "/O=Ord\CN=AcmeCA" -keyout server.key -out server.crt

At first, we need to create a few folders to store the files of the root, sub, and server certificates. Then we must create a root private key using the below command and the key will be stored in the

"root-ca/private/" folder.

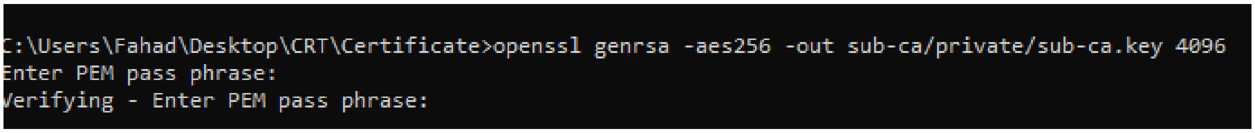
openssl genrsa -aes256 -out root-ca/private/ca.key 4096



```
C:\Users\Fahad\Desktop\CRT\Certificate>openssl genrsa -aes256 -out root-ca/private/ca.key 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Same for the sub certificate and the key will be stored in the “sub-ca/private/” folder.

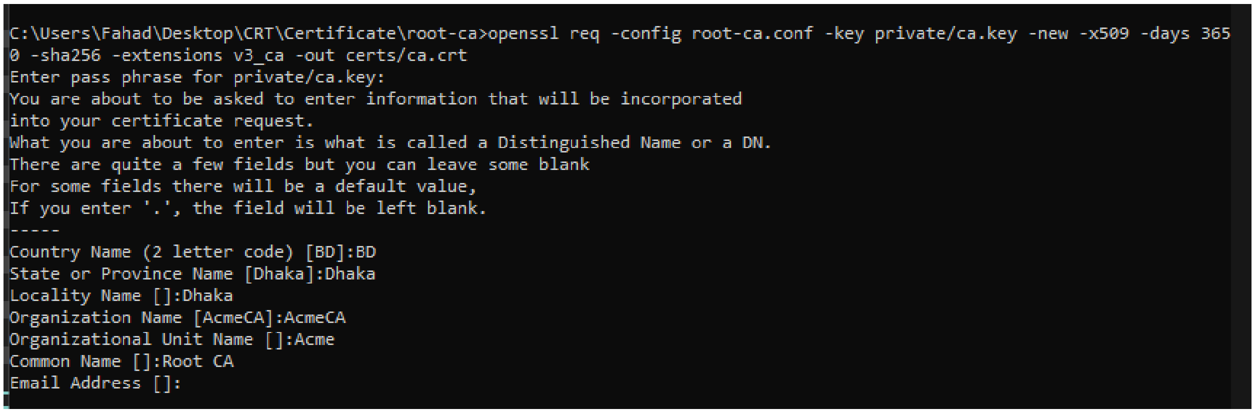
openssl genrsa -aes256 -out sub-ca/private/sub-ca.key 4096



```
C:\Users\Fahad\Desktop\CRT\Certificate>openssl genrsa -aes256 -out sub-ca/private/sub-ca.key 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Then we have to be in the root-ca folder to create the root certificate, and I will use the **“root-ca. conf”** file which is a configuration file that will store the requirements. The contact of the **“root-ca. conf**” file will be given below the command.

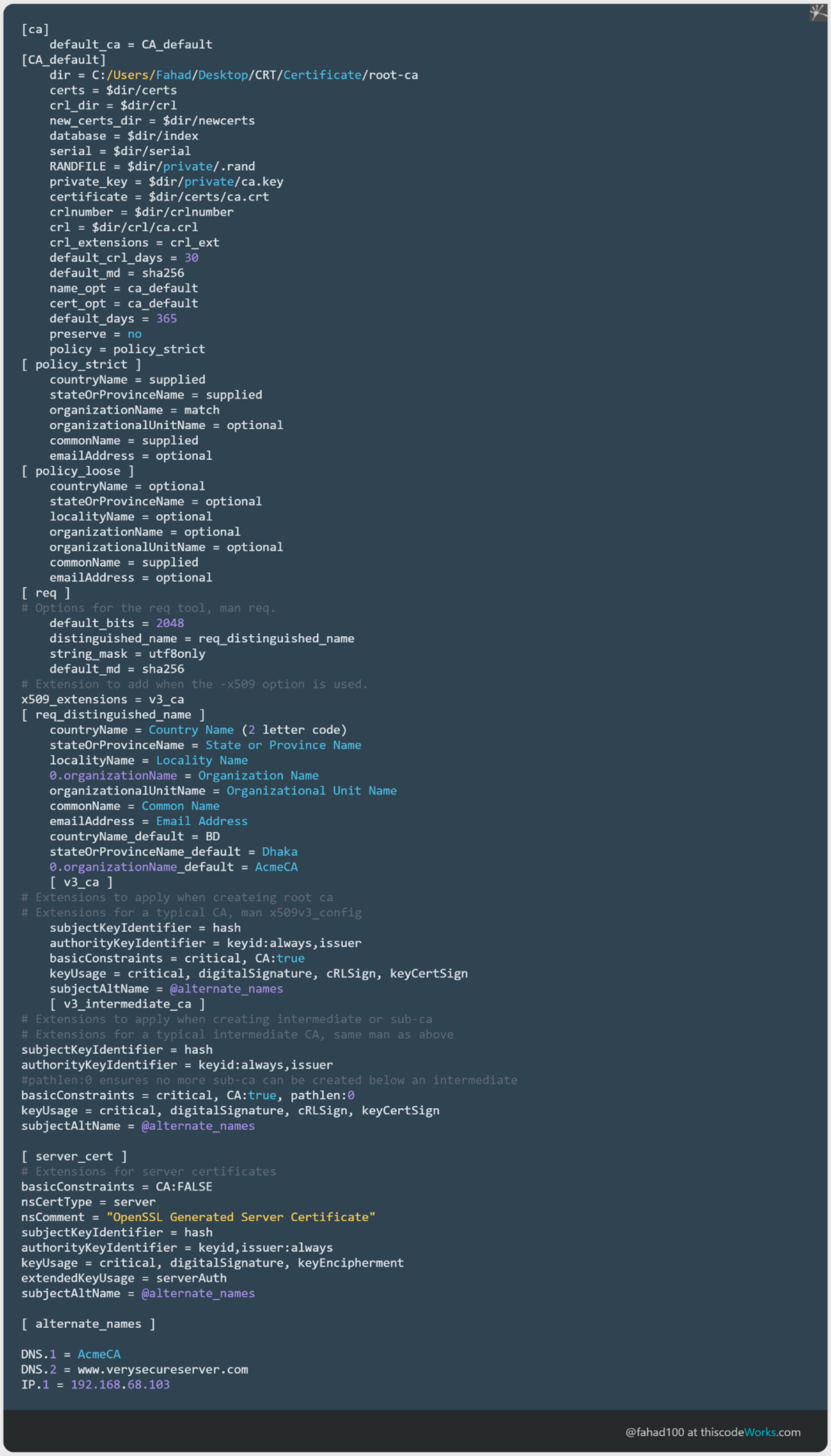
openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 3650 -sha256 -extensions v3_ca -out certs/ca.crt



```
C:\Users\Fahad\Desktop\CRT\Certificate\root-ca>openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 365
0 -sha256 -extensions v3_ca -out certs/ca.crt
Enter pass phrase for private/ca.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) [BD]:BD
State or Province Name [Dhaka]:Dhaka
Locality Name []:Dhaka
Organization Name [AcmeCA]:AcmeCA
Organizational Unit Name []:Acme
Common Name []:Root CA
Email Address []:
```

“root-ca.conf” file

```
[ alternate_names ]


DNS.1 = AcmeCA

DNS.2 = www.verysecureserver.com

IP.1 = 192.168.68.103

[CA_default]

    dir = C:/Users/Fahad/Desktop/CRT/Certificate/root-ca
```

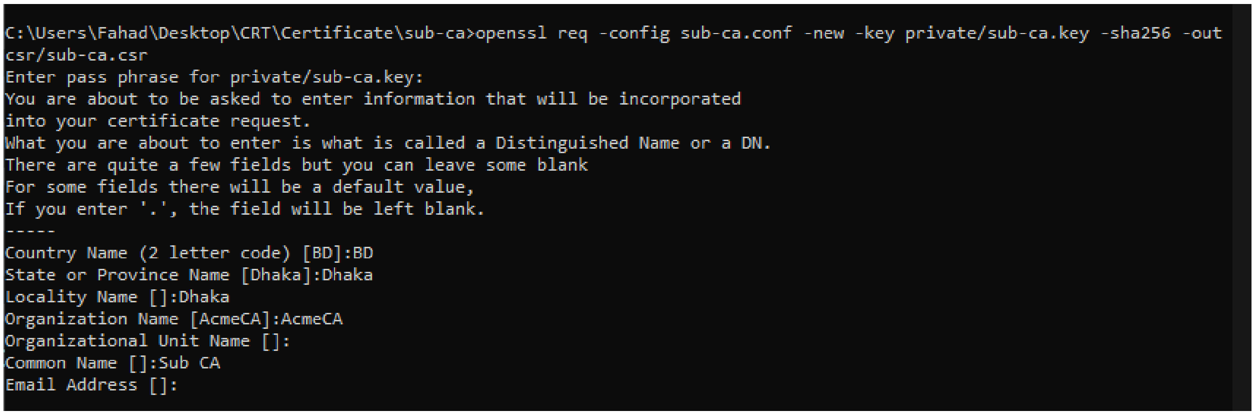


```
[ca]
    default_ca = CA_default
[CA_default]
    dir = C:/Users/Fahad/Desktop/CRT/Certificate/root-ca
    certs = $dir/certs
    crl_dir = $dir/crl
    new_certs_dir = $dir/newcerts
    database = $dir/index
    serial = $dir/serial
    RANDFILE = $dir/private/.rand
    private_key = $dir/private/ca.key
    certificate = $dir/certs/ca.crt
    crlnumber = $dir/crlnumber
    crl = $dir/crl/ca.crl
    crl_extensions = crl_ext
    default_crl_days = 30
    default_md = sha256
    name_opt = ca_default
    cert_opt = ca_default
    default_days = 365
    preserve = no
    policy = policy_strict
[ policy_strict ]
    countryName = supplied
    stateOrProvinceName = supplied
    organizationName = match
    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 = AcmeCA
    [ 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
    subjectAltName = @alternate_names
    [ 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
subjectAltName = @alternate_names

[ server_cert ]
# Extensions for server certificates
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alternate_names

[ alternate_names ]

DNS.1 = AcmeCA
DNS.2 = www.verysecureserver.com
IP.1 = 192.168.68.103
```

@fahad100 at thiscodeWorks.com

Now, for the Sub certificate, we have to be in the sub-ca folder to create sub csr file using the sub-ca.key which is created which will be signed by the root certificate later, and I will use the **“sub-ca.conf”** file which is a configuration file which will store the requirements. The contact of the **“sub-ca.conf”** file will be given below the command.

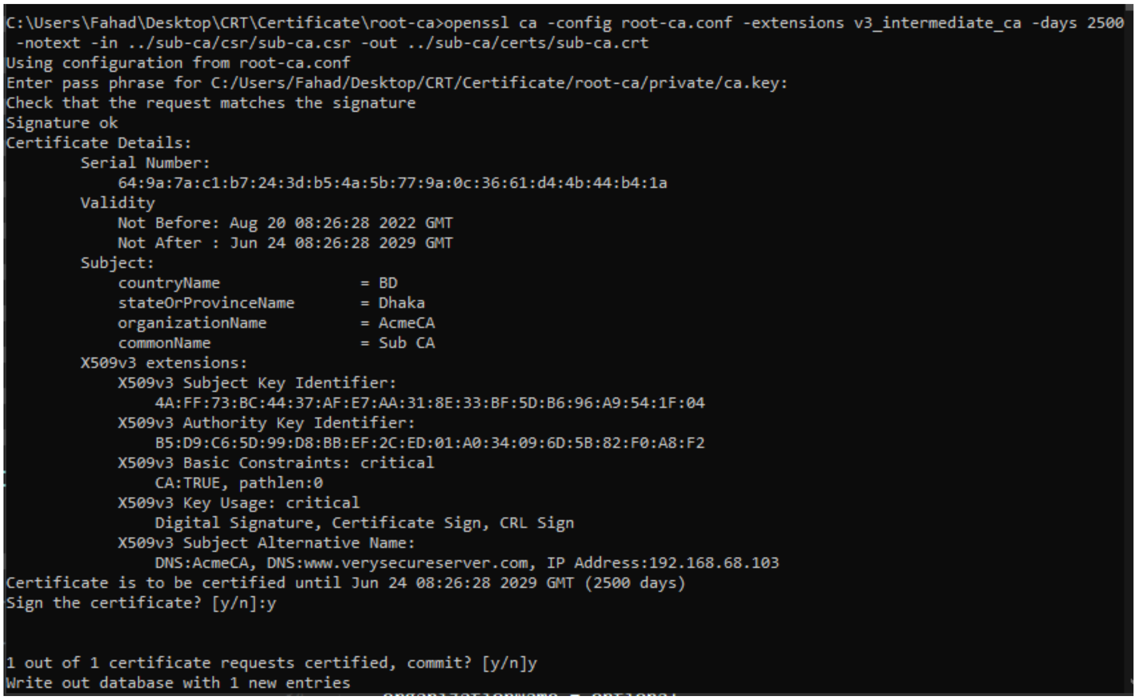
openssl req -config sub-ca.conf -new -key private/sub-ca.key -sha256 -out csr/sub-ca.csr



```
C:\Users\Fahad\Desktop\CRT\Certificate\sub-ca>openssl req -config sub-ca.conf -new -key private/sub-ca.key -sha256 -out
csr/sub-ca.csr
Enter pass phrase for private/sub-ca.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) [BD]:BD
State or Province Name [Dhaka]:Dhaka
Locality Name []:Dhaka
Organization Name [AcmeCA]:AcmeCA
Organizational Unit Name []:
Common Name []:Sub CA
Email Address []:
```

Now, to sign the **sub-ca** using the root-ca certificate, I must be in the root-ca folder and will execute the below command.

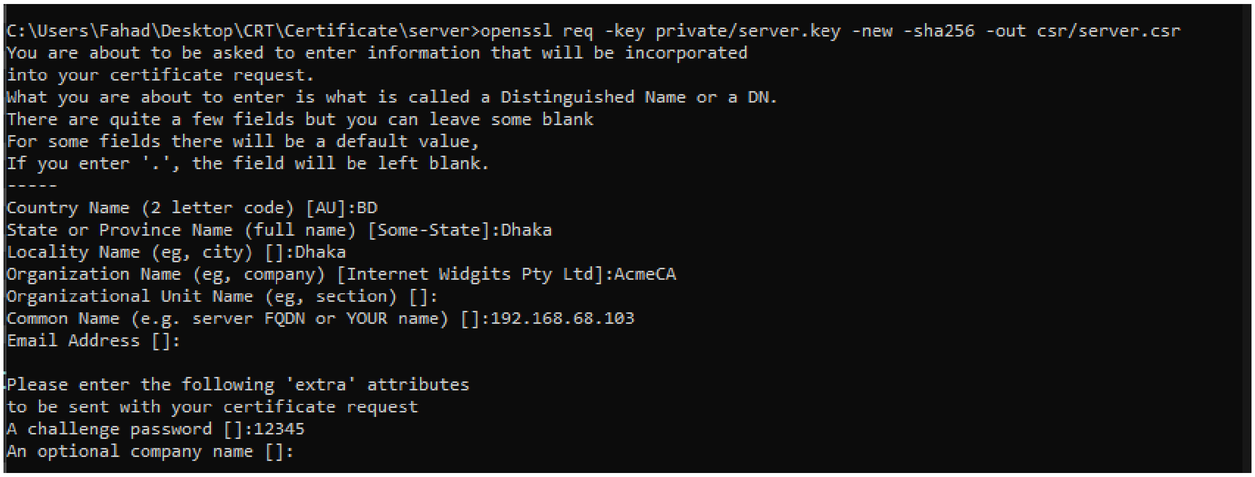
openssl ca -config root-ca.conf -extensions v3_intermediate_ca -days 2500 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt



```
C:\Users\Fahad\Desktop\CRT\Certificate\root-ca>openssl ca -config root-ca.conf -extensions v3_intermediate_ca -days 2500
 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt
Using configuration from root-ca.conf
Enter pass phrase for C:/Users/Fahad/Desktop/CRT/Certificate/root-ca/private/ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            64:9a:7a:c1:b7:24:3d:b5:4a:5b:77:9a:0c:36:61:d4:4b:44:b4:1a
        Validity
            Not Before: Aug 20 08:26:28 2022 GMT
            Not After : Jun 24 08:26:28 2029 GMT
        Subject:
            countryName               = BD
            stateOrProvinceName       = Dhaka
            organizationName          = AcmeCA
            commonName                = Sub CA
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                4A:FF:73:BC:44:37:AF:E7:AA:31:8E:33:BF:5D:B6:96:A9:54:1F:04
            X509v3 Authority Key Identifier:
                B5:D9:C6:5D:99:D8:BB:EF:2C:ED:01:A0:34:09:6D:5B:82:F0:A8:F2
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Subject Alternative Name:
                DNS:AcmeCA, DNS:www.verysecureserver.com, IP Address:192.168.68.103
Certificate is to be certified until Jun 24 08:26:28 2029 GMT (2500 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
```

Now, for the server certificate, let’s move to the server folder and create the **“server.csr”** file using the private **“server.key”** using this command.

```
openssl req -key private/server.key -new -sha256 -out csr/server.csr
```

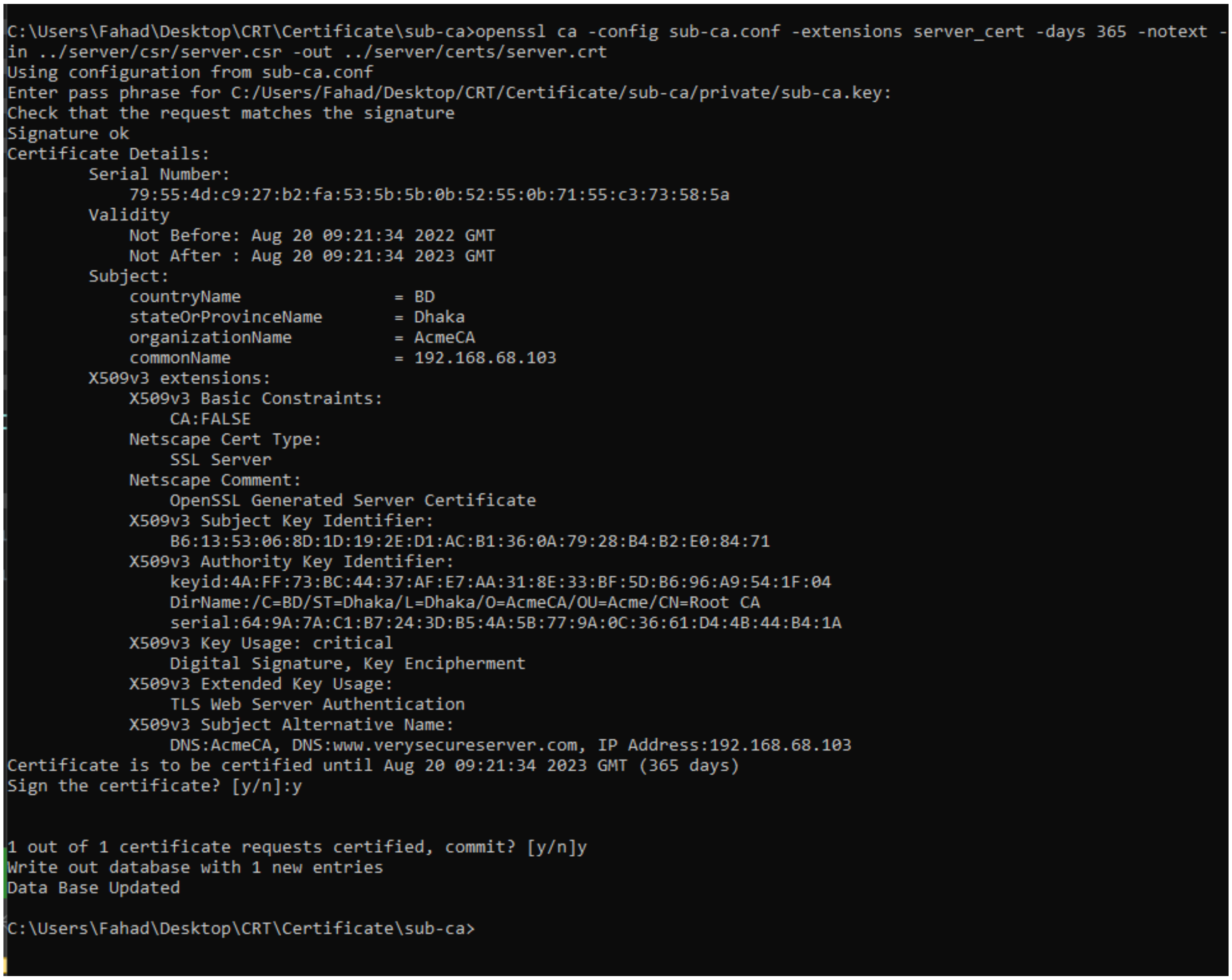


```
C:\Users\Fahad\Desktop\CRT\Certificate\server>openssl req -key private/server.key -new -sha256 -out csr/server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There 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]:BD
State or Province Name (full name) [Some-State]:Dhaka
Locality Name (eg, city) []:Dhaka
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AcmeCA
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.68.103
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:
```

Then to sign the server certificate using the sub-ca certificate we have to move into the sub-ca folder and run the below command which will sign the server certificate

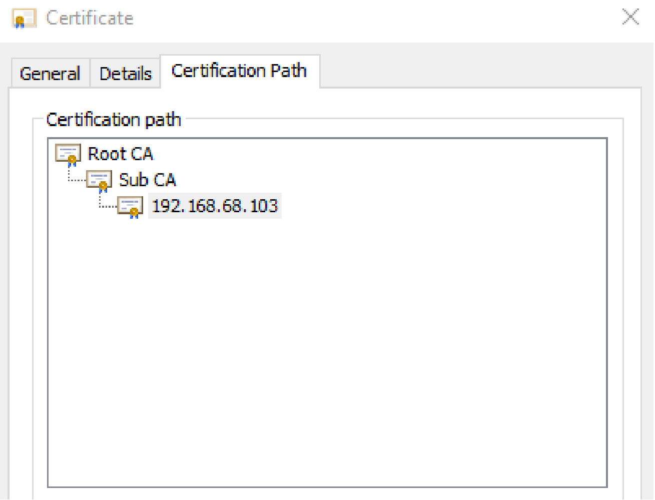
```
openssl ca -config sub-ca.conf -extensions server_cert -days 365 -notext -in ../server/csr/server.csr -out ../server/certs/server.crt
```

```
Signature ok
Certificate Details:
    Serial Number:
      79:55:4d:c9:27:b2:fa:53:5b:5b:0b:52:55:0b:71:55:c3:73:58:5a
    Validity
      Not Before: Aug 20 09:21:34 2022 GMT
      Not After : Aug 20 09:21:34 2023 GMT
    Subject:
      countryName               = BD
      stateOrProvinceName       = Dhaka
      organizationName          = AcmeCA
      commonName                = 192.168.68.103
```



```
C:\Users\Fahad\Desktop\CRT\Certificate\sub-ca>openssl ca -config sub-ca.conf -extensions server_cert -days 365 -notext -
in ../server/csr/server.csr -out ../server/certs/server.crt
Using configuration from sub-ca.conf
Enter pass phrase for C:/Users/Fahad/Desktop/CRT/Certificate/sub-ca/private/sub-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            79:55:4d:c9:27:b2:fa:53:5b:5b:0b:52:55:0b:71:55:c3:73:58:5a
        Validity
            Not Before: Aug 20 09:21:34 2022 GMT
            Not After : Aug 20 09:21:34 2023 GMT
        Subject:
            countryName               = BD
            stateOrProvinceName       = Dhaka
            organizationName          = AcmeCA
            commonName                = 192.168.68.103
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Server
            Netscape Comment:
                OpenSSL Generated Server Certificate
            X509v3 Subject Key Identifier:
                B6:13:53:06:8D:1D:19:2E:D1:AC:B1:36:0A:79:28:B4:B2:E0:84:71
            X509v3 Authority Key Identifier:
                keyid:4A:FF:73:BC:44:37:AF:E7:AA:31:8E:33:BF:5D:B6:96:A9:54:1F:04
                DirName:/C=BD/ST=Dhaka/L=Dhaka/O=AcmeCA/OU=Acme/CN=Root CA
                serial:64:9A:7A:C1:B7:24:3D:B5:4A:5B:77:9A:0C:36:61:D4:4B:44:B4:1A
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:AcmeCA, DNS:www.verysecureserver.com, IP Address:192.168.68.103
Certificate is to be certified until Aug 20 09:21:34 2023 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

C:\Users\Fahad\Desktop\CRT\Certificate\sub-ca>
```

Now, we need to install the **root-ca** certificate to the local machine then make the certificate as Trusted Root Certification Authorities and **sub-ca** certificate as Immediate Certificate authority, and finally the server certificate as the personal certificate. The process to install the certificate is given below.



Then I added the domain to the Windows host.

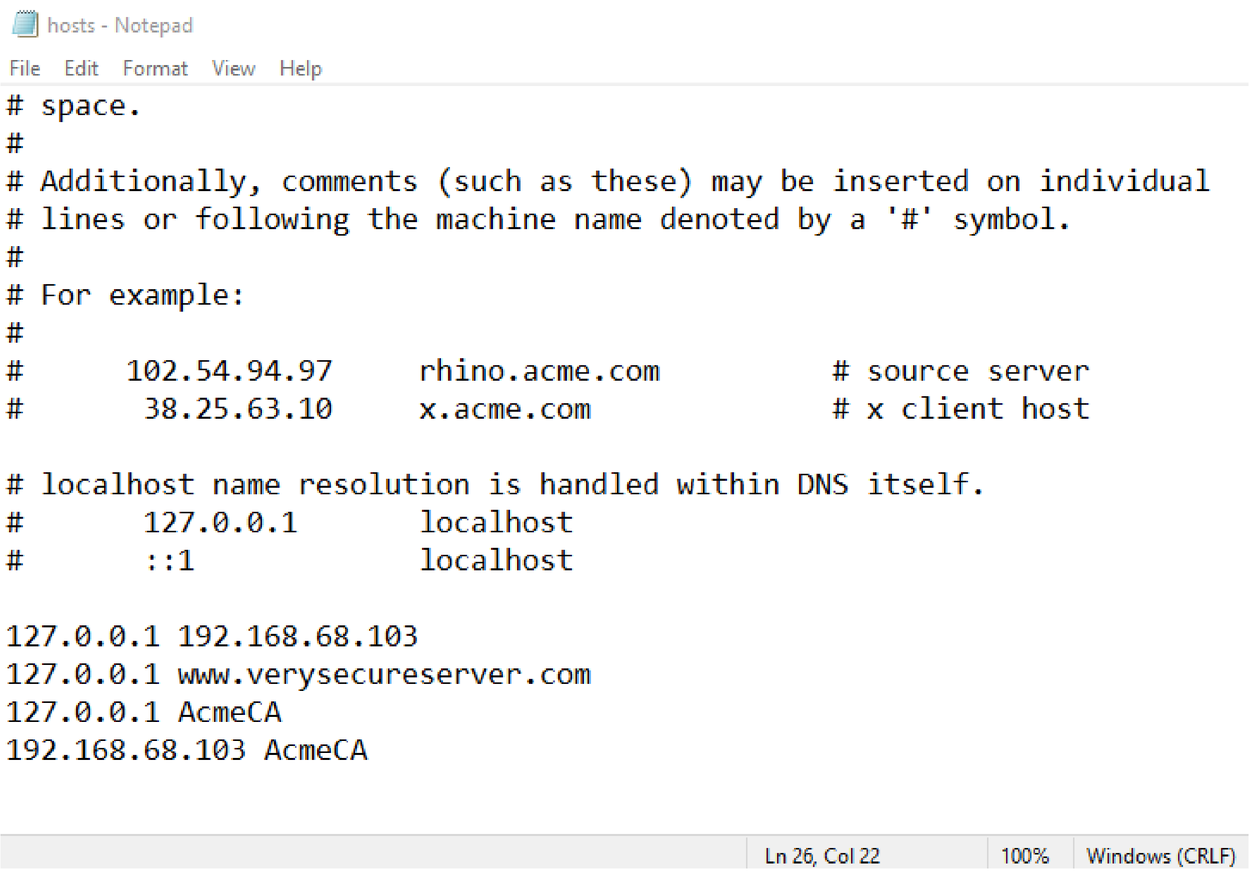
C:\Windows\System32\drivers\etc\hosts

127.0.0.1 192.168.68.103

127.0.0.1 www.verysecureserver.com

127.0.0.1 AcmeCA

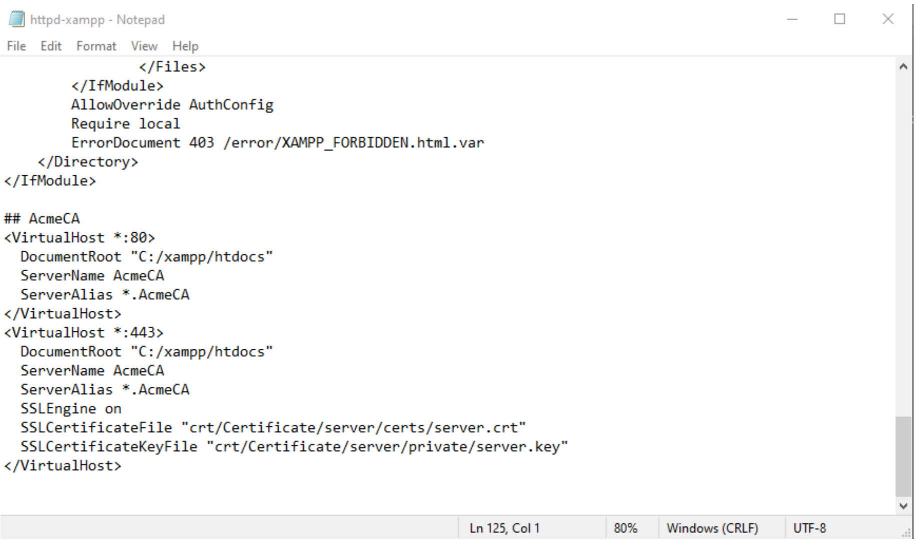
192.168.68.103 AcmeCA



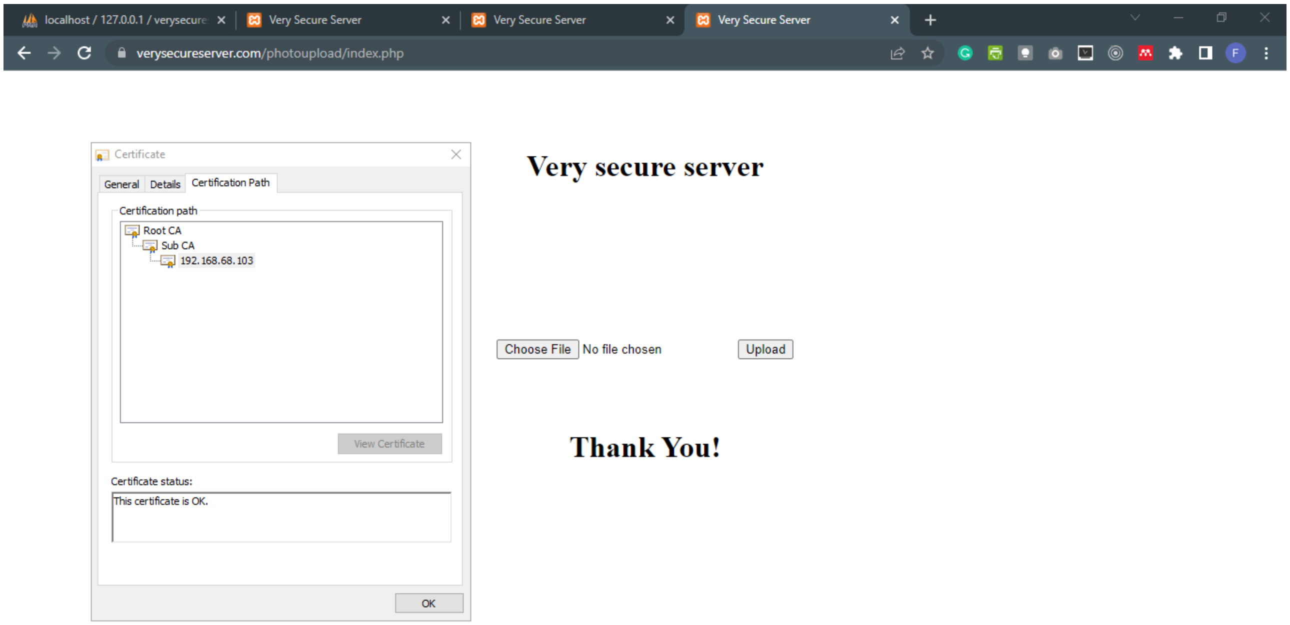
```
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

127.0.0.1 192.168.68.103
127.0.0.1 www.verysecureserver.com
127.0.0.1 AcmeCA
192.168.68.103 AcmeCA
```

Then the website needs to be added to the XAMPP conf. file. The file is available in C:\xampp\apache\conf\extra\httpd-xampp.conf which needs to be edited. Below code is added to this file for configuration. [*After adding this code and saving it the XAMMP Apache Module needs to be restarted.]*

```
## AcmeCA
<VirtualHost *:80>
  DocumentRoot "C:/xampp/htdocs"
  ServerName AcmeCA
  ServerAlias *.AcmeCA
</VirtualHost>
<VirtualHost *:443>
  DocumentRoot "C:/xampp/htdocs"
  ServerName AcmeCA
  ServerAlias *.AcmeCA
  SSLEngine on
  SSLCertificateFile "crt/Certificate/server/certs/server.crt"
  SSLCertificateKeyFile "crt/Certificate/server/private/server.key"
</VirtualHost>
```



```
                </Files>
        </IfModule>
        AllowOverride AuthConfig
        Require local
        ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
    </Directory>
</IfModule>

## AcmeCA
<VirtualHost *:80>
  DocumentRoot "C:/xampp/htdocs"
  ServerName AcmeCA
  ServerAlias *.AcmeCA
</VirtualHost>
<VirtualHost *:443>
  DocumentRoot "C:/xampp/htdocs"
  ServerName AcmeCA
  ServerAlias *.AcmeCA
  SSLEngine on
  SSLCertificateFile "crt/Certificate/server/certs/server.crt"
  SSLCertificateKeyFile "crt/Certificate/server/private/server.key"
</VirtualHost>
```



verysecureserver.com/photoupload/index.php

Certification path

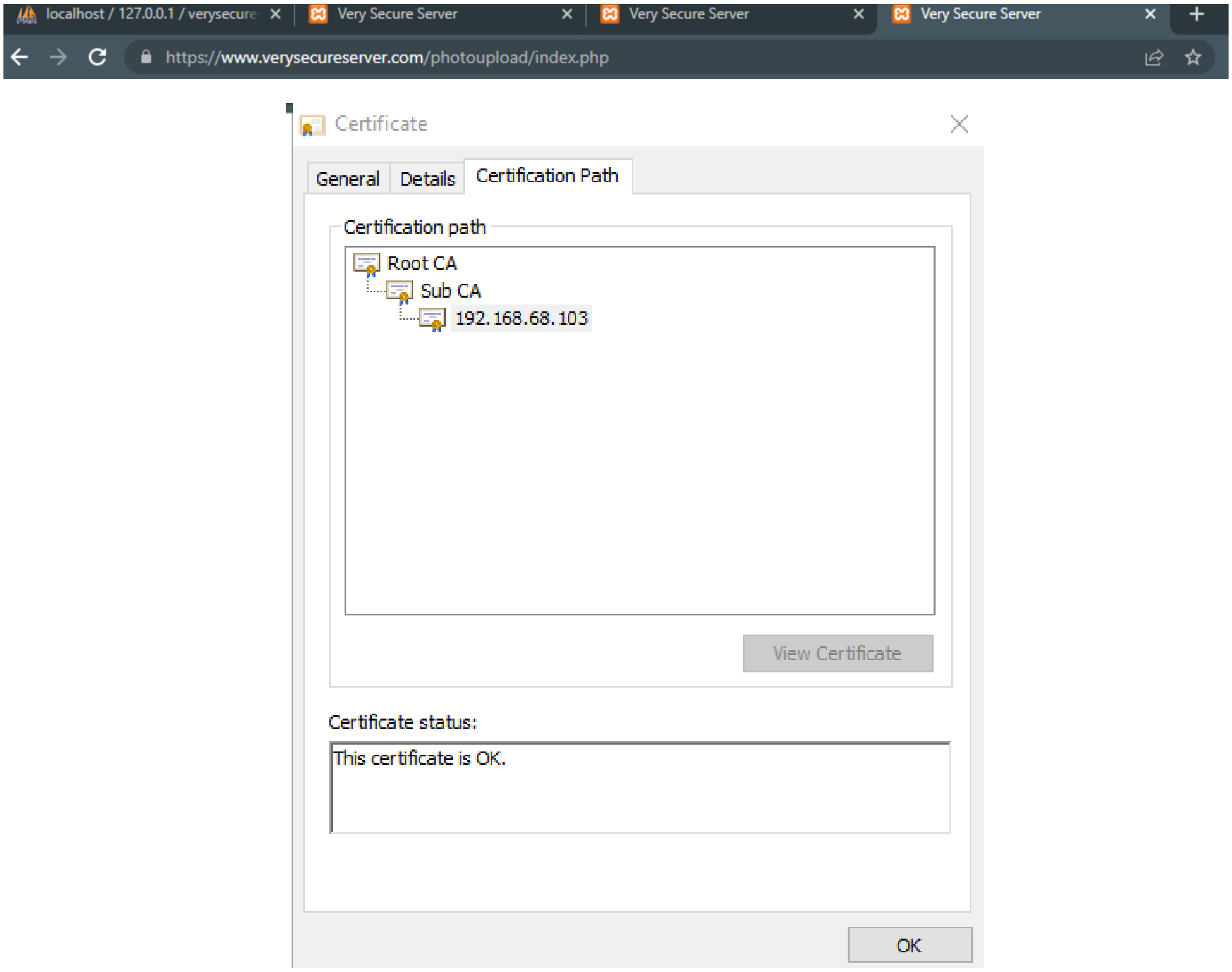
Root CA
Sub CA
192.168.68.103

Certificate status:

This certificate is OK.

Very secure server

Thank You!



https://www.verysecureserver.com/photoupload/index.php

Certification path

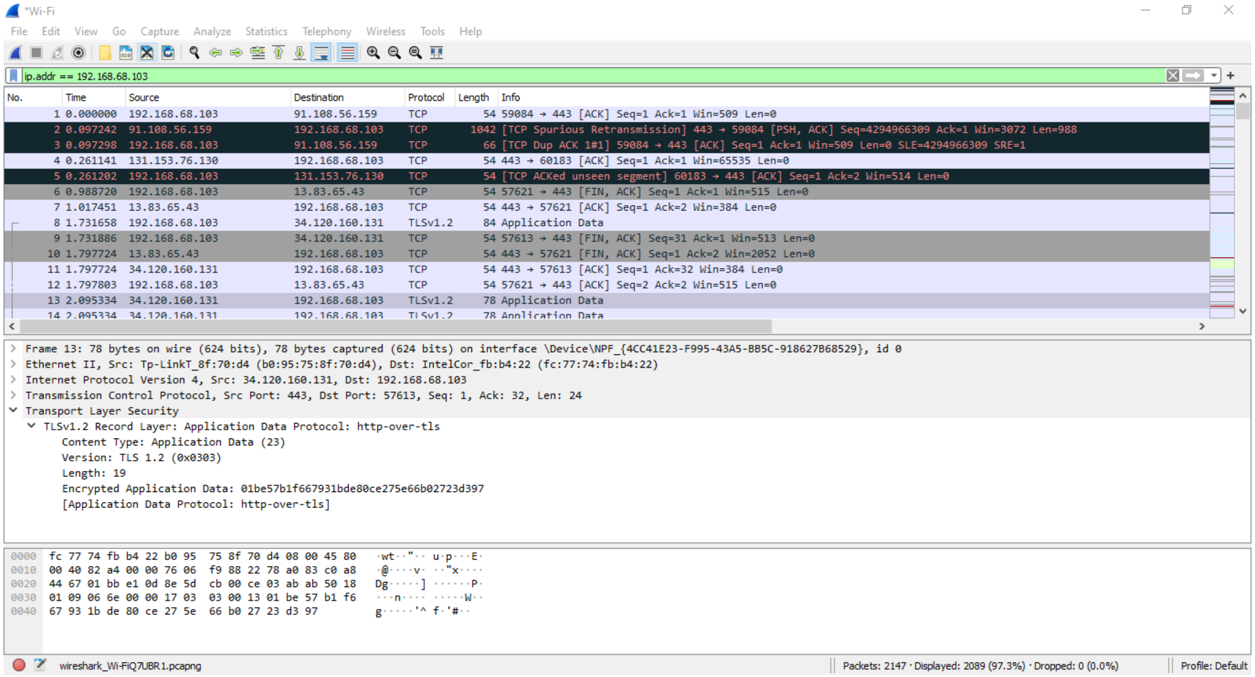
Root CA
Sub CA
192.168.68.103

Certificate status:

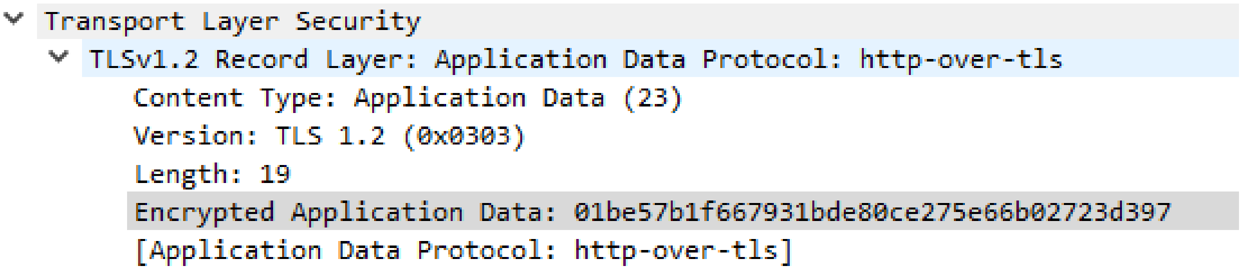
This certificate is OK.

3. Wireshark security:

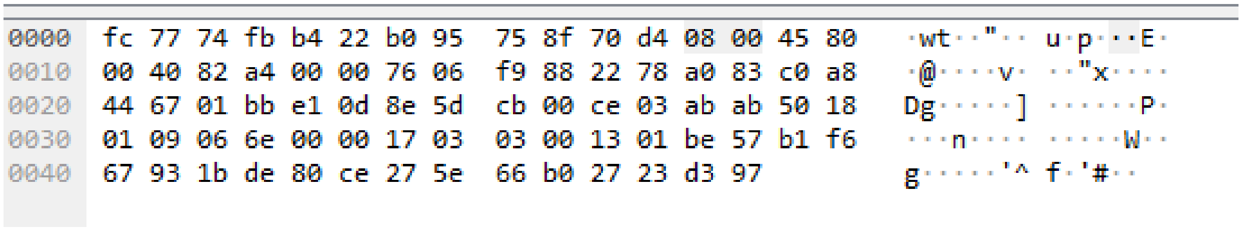
Let's, **investigate** the **Wireshark security** while uploading the photo which is given below.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 192.168.68.103

No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.68.103  91.108.56.159   TCP        54 59084 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=0
   2 0.097242  91.108.56.159   192.168.68.103  TCP      1042 [TCP Spurious Retransmission] 443 → 59084 [PSH, ACK] Seq=4294966309 Ack=1 Win=3072 Len=988
   3 0.097298  192.168.68.103  91.108.56.159   TCP        66 [TCP Dup ACK 1#1] 59084 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=0 SLE=4294966309 SRE=1
   4 0.261141  131.153.76.130  192.168.68.103  TCP        54 443 → 60183 [ACK] Seq=1 Ack=1 Win=65535 Len=0
   5 0.261202  192.168.68.103  131.153.76.130  TCP        54 [TCP ACKed unseen segment] 60183 → 443 [ACK] Seq=1 Ack=2 Win=514 Len=0
   6 0.988720  192.168.68.103  13.83.65.43     TCP        54 57621 → 443 [FIN, ACK] Seq=1 Ack=1 Win=515 Len=0
   7 1.017451  13.83.65.43     192.168.68.103  TCP        54 443 → 57621 [ACK] Seq=1 Ack=2 Win=384 Len=0
   8 1.731658  192.168.68.103  34.120.160.131  TLSv1.2    84 Application Data
   9 1.731886  192.168.68.103  34.120.160.131  TCP        54 57613 → 443 [FIN, ACK] Seq=31 Ack=1 Win=513 Len=0
  10 1.797724  13.83.65.43     192.168.68.103  TCP        54 443 → 57621 [FIN, ACK] Seq=1 Ack=2 Win=2052 Len=0
  11 1.797724  34.120.160.131  192.168.68.103  TCP        54 443 → 57613 [ACK] Seq=1 Ack=32 Win=384 Len=0
  12 1.797803  192.168.68.103  13.83.65.43     TCP        54 57621 → 443 [ACK] Seq=2 Ack=2 Win=515 Len=0
  13 2.095334  34.120.160.131  192.168.68.103  TLSv1.2    78 Application Data
  14 2.095334  34.120.160.131  192.168.68.103  TLSv1.2    78 Application Data

> Frame 13: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{4CC41E23-F995-43A5-BB5C-918627B68529}, id 0
> Ethernet II, Src: Tp-LinkT_8f:70:d4 (b0:95:75:8f:70:d4), Dst: IntelCor_fb:b4:22 (fc:77:74:fb:b4:22)
> Internet Protocol Version 4, Src: 34.120.160.131, Dst: 192.168.68.103
> Transmission Control Protocol, Src Port: 443, Dst Port: 57613, Seq: 1, Ack: 32, Len: 24
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 19
      Encrypted Application Data: 01be57b1f667931bde80ce275e66b02723d397
      [Application Data Protocol: http-over-tls]

0000  fc 77 74 fb b4 22 b0 95  75 8f 70 d4 08 00 45 80   ·wt··"·· u·p···E·
0010  00 40 82 a4 00 00 76 06  f9 88 22 78 a0 83 c0 a8   ·@····v· ··"x····
0020  44 67 01 bb e1 0d 8e 5d  cb 00 ce 03 ab ab 50 18   Dg·····] ······P·
0030  01 09 06 6e 00 00 17 03  03 00 13 01 be 57 b1 f6   ···n···· ·····W··
0040  67 93 1b de 80 ce 27 5e  66 b0 27 23 d3 97         g·····'^ f·'#··

wireshark_Wi-FiQ7UBR1.pcapng    Packets: 2147 · Displayed: 2089 (97.3%) · Dropped: 0 (0.0%)    Profile: Default
```



```
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 19
      Encrypted Application Data: 01be57b1f667931bde80ce275e66b02723d397
      [Application Data Protocol: http-over-tls]
```



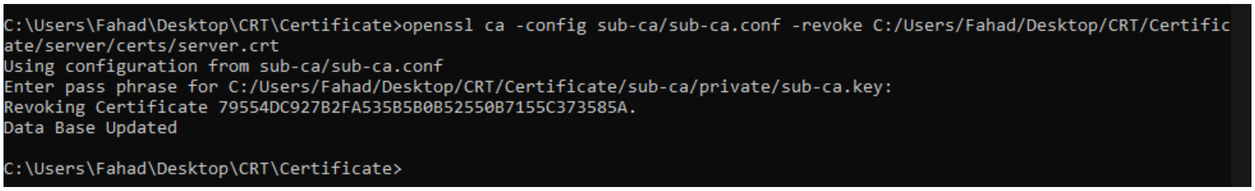
```
0000  fc 77 74 fb b4 22 b0 95  75 8f 70 d4 08 00 45 80   ·wt··"·· u·p···E·
0010  00 40 82 a4 00 00 76 06  f9 88 22 78 a0 83 c0 a8   ·@····v· ··"x····
0020  44 67 01 bb e1 0d 8e 5d  cb 00 ce 03 ab ab 50 18   Dg·····] ······P·
0030  01 09 06 6e 00 00 17 03  03 00 13 01 be 57 b1 f6   ···n···· ·····W··
0040  67 93 1b de 80 ce 27 5e  66 b0 27 23 d3 97         g·····'^ f·'#··
```

4. Revoke the certificate using OpenSSL.

To revoke the certificate, at first, we need to run the below command. Here, using the sub-ca certificate we can revoke the server certificate.

```
openssl ca -config sub-ca/sub-ca.conf -revoke
C:/Users/Fahad/Desktop/CRT/Certificate/server/certs/server.crt
```

```
Revoking Certificate 79554DC927B2FA535B5B0B52550B7155C373585A.

Data Base Updated
```



```
C:\Users\Fahad\Desktop\CRT\Certificate>openssl ca -config sub-ca/sub-ca.conf -revoke C:/Users/Fahad/Desktop/CRT/Certific
ate/server/certs/server.crt
Using configuration from sub-ca/sub-ca.conf
Enter pass phrase for C:/Users/Fahad/Desktop/CRT/Certificate/sub-ca/private/sub-ca.key:
Revoking Certificate 79554DC927B2FA535B5B0B52550B7155C373585A.
Data Base Updated

C:\Users\Fahad\Desktop\CRT\Certificate>
```

Which will generate a Revoking Certificate number like below. Then we can verify this from the sub-ca database which is stored in the index file. Now to Generate a Certificate Revocation List (CRL) we must run this command.

```
openssl ca -config sub-ca/sub-ca.conf -gencrl -out
C:/Users/Fahad/Desktop/CRT/Certificate/sub-ca/crl/sub-ca.crl
```

Now, to check the Revoked Certificate List in CRL we can use this command.

```
openssl crl -in C:/Users/Fahad/Desktop/CRT/Certificate/sub-ca/crl/sub-ca.crl -text
-noout.
```

---------------------Thank You-----------------------------