Cybersecurity, Law, and Ethics

CSE 487 / ICE 453

Section: 03

Mini Project-1

"Securing a networked system with Public Key Infrastructure"

Submitted to

Dr. Md. Hassanul Ferdous
Associate Professor
Department of Computer Science & Engineering
East West University

Submitted by (Group - 11)

Sumit kumer Das	2021-3-50-002
Hrittik Chakraborty	2021-2-50-011
Debnil Malaker	2020-3-50-003
Junayed Karim	2021-1-50-011

Submission Date: 15th April 2025

Recorded Presentation

(Google Drive Link, student mail address) :

https://drive.google.com/file/d/1z3QS4yJOz7nmwkCIYUfIsY3omIx-JOcl/view?usp=sharing

Table of Contents

Problem Statement

We have to secure a networked system with Public Key Infrastructure by implementing Transport Layer Security on HTTP for the https:// connection.

Requirements :

- Configuration of Certification Authority AcmeCA with AcmeRootCA as the RootCA.
- Configuration of the Web Server with Apache2 on a Linux Host.
- DNS configuration for www.verysecureserver.com
- Firewall configuration to allow necessary ports (53, 80, 443) only
- CSR Configuration and Generation for the www.verysecureserver.com
- Transferring the CSR to AcmeCA
- Certification process (Verification and Certificate Generation from CSR)
- Transferring the certificate from AcmeCA to www.verysecureserver.com
- Installation of the signed SSL certificate in the server of www.verysecureserver.com
- Making the system trust Acme-RootCA
- Implementation of a simple file-uploading page in the server.
- Verifying the security of the connection by inspection (the padlock icon)

Necessary Elements :

- Oracle VM Virtualbox
- Linux Ubuntu 18.04
- Firefox version 59.0.2 (64-bit)
- XAMPP

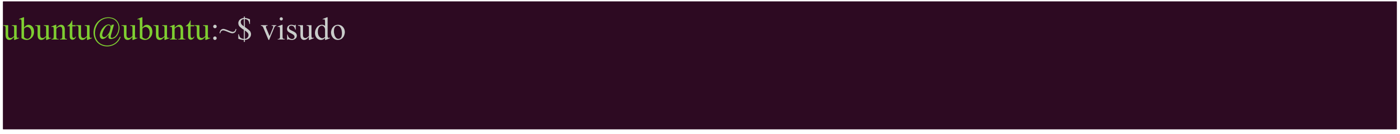
Create Virtual Machines In Windows 11

We need to create a virtual machine to work with our project.

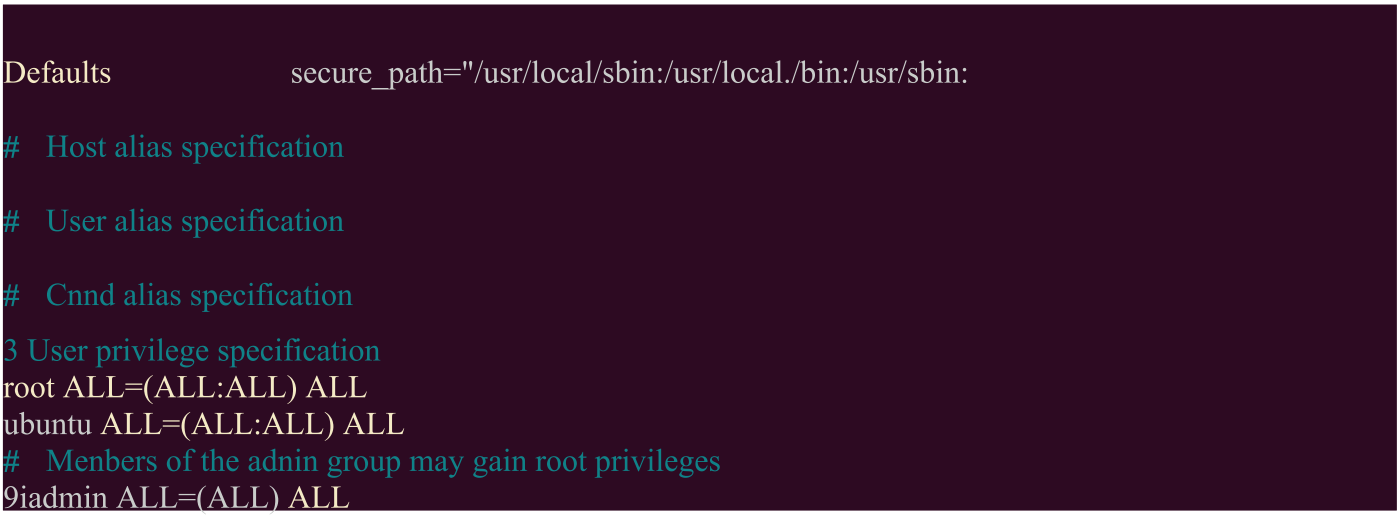
- Download linux ubuntu-18.04-desktop-amd64

  https://releases.ubuntu.com/18.04/
- Download and install VMware Workstation 16 Player
- Extract ubuntu-18.04-desktop-amd64 from zip file
- In the VM, click on new => Give the VM a name, Folder Directory and insert the necessary iso file
- Start the VM and give Username = ubuntu, Password=ubuntu, Hostname = ubuntu
- Open the terminal and go to root user Su -
  Password: ubuntu

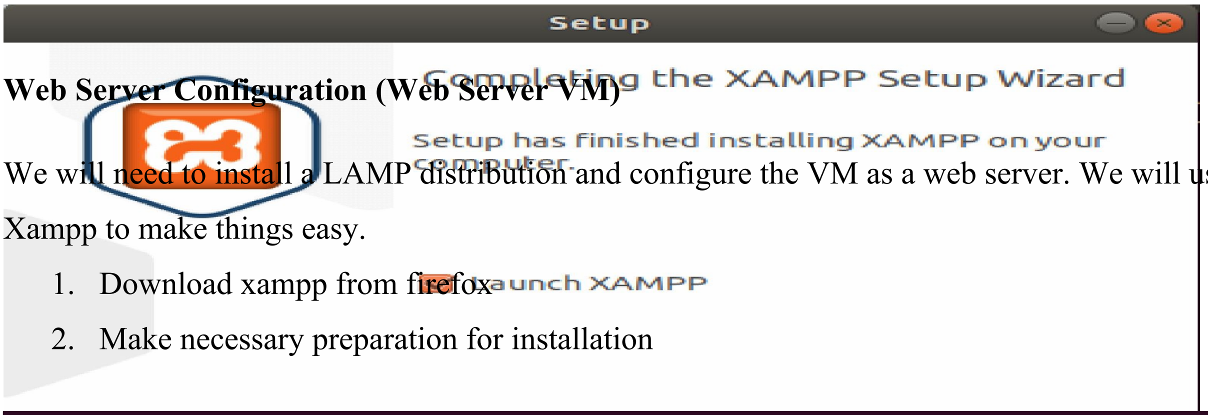
and check if sudo is in the sudoers file, and fix the situation if it is not then, add it there



```
ubuntu@ubuntu:~$ visudo
```



```
Defaults        secure_path="/usr/local/sbin:/usr/local./bin:/usr/sbin:

#   Host alias specification

#   User alias specification

#   Cnnd alias specification

3 User privilege specification
root ALL=(ALL:ALL) ALL
ubuntu ALL=(ALL:ALL) ALL
#   Menbers of the adnin group may gain root privileges
9iadmin ALL=(ALL) ALL

#   Allow members of group sudo to execute any command
9Gsudo ALL=(ALL:ALL) ALL

#   See sudoers(5) for more information on "Sinclude"' directives:
```



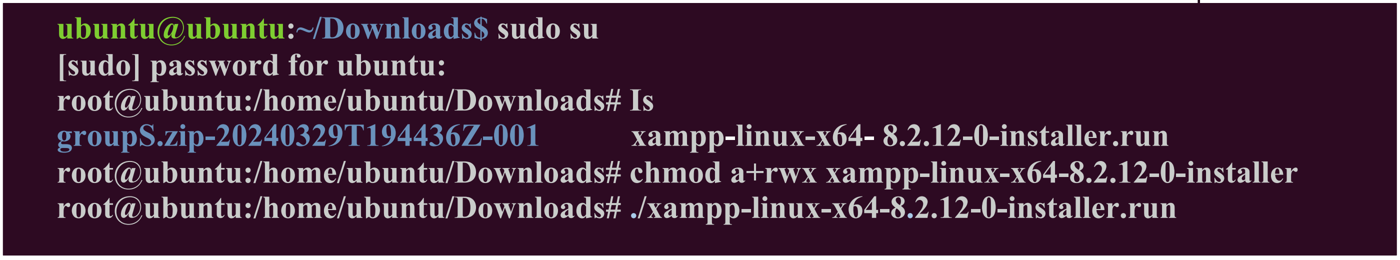
Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

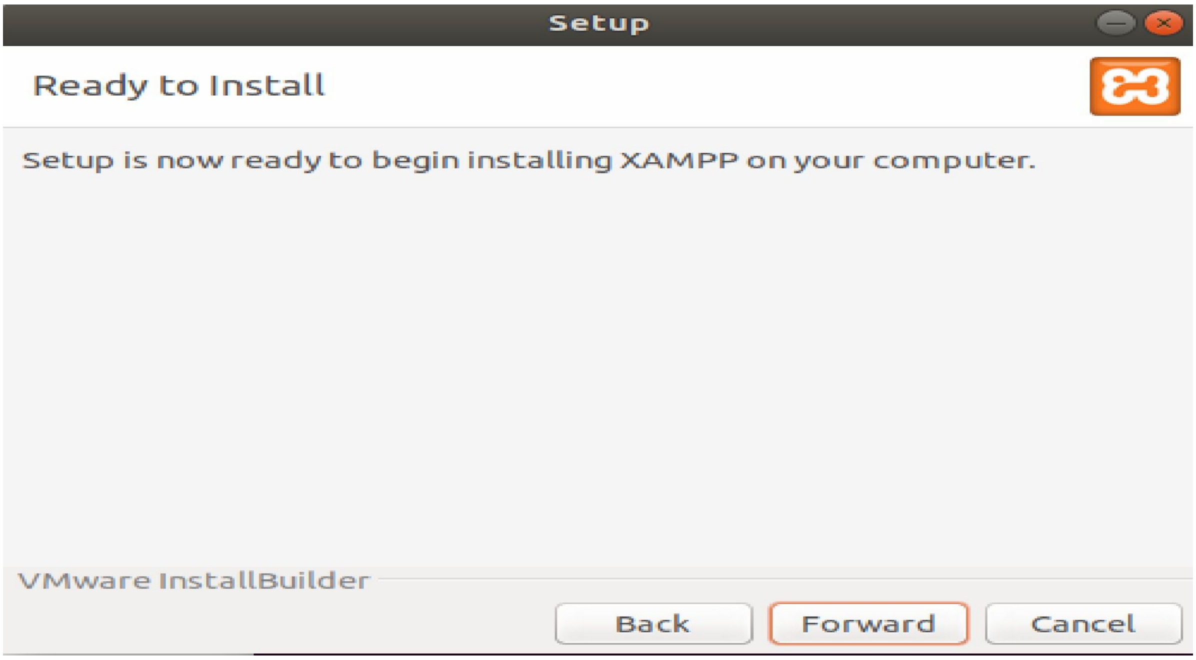
Web Server Configuration (Web Server VM)

We will need to install a LAMP distribution and configure the VM as a web server. We will use Xampp to make things easy.

1. Download xampp from firefox
2. Make necessary preparation for installation



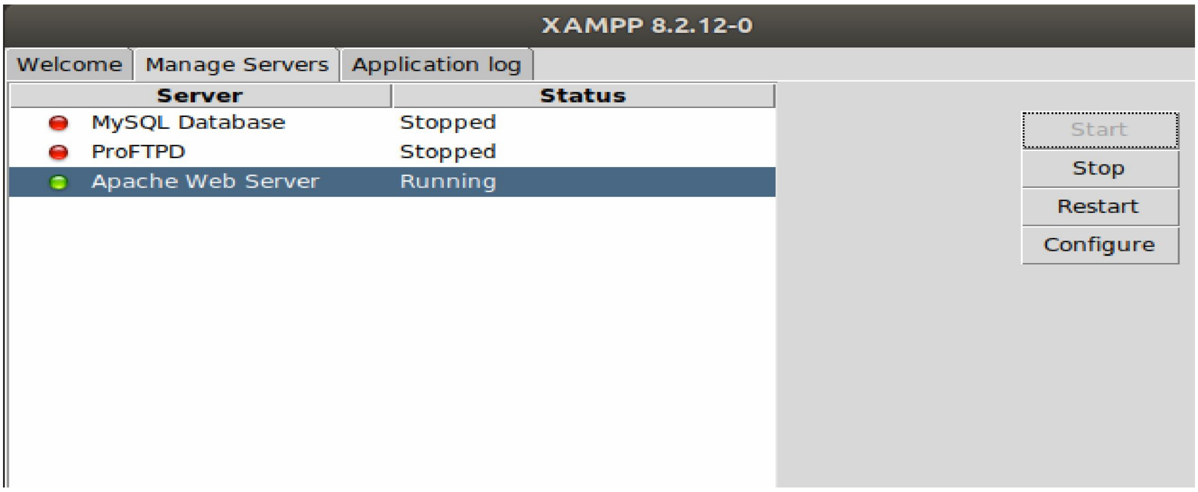
```
ubuntu@ubuntu:~/Downloads$ sudo su
[sudo] password for ubuntu:
root@ubuntu:/home/ubuntu/Downloads# Is
groupS.zip-20240329T194436Z-001         xampp-linux-x64- 8.2.12-0-installer.run
root@ubuntu:/home/ubuntu/Downloads# chmod a+rwx xampp-linux-x64-8.2.12-0-installer
root@ubuntu:/home/ubuntu/Downloads# ./xampp-linux-x64-8.2.12-0-installer.run
```



Ready to Install

Setup is now ready to begin installing XAMPP on your computer.

- ❖ Then start the server



Server	Status
🟥 MySQL Database	Stopped
🟥 ProFTPD	Stopped
🟩 Apache Web Server	Running

- ❖ This is to check whether the Xampp server is working or not

Creating CA, Sub-CA, and Generating SSL Certificates

i. Preparing environment su –

   The password: ubuntu

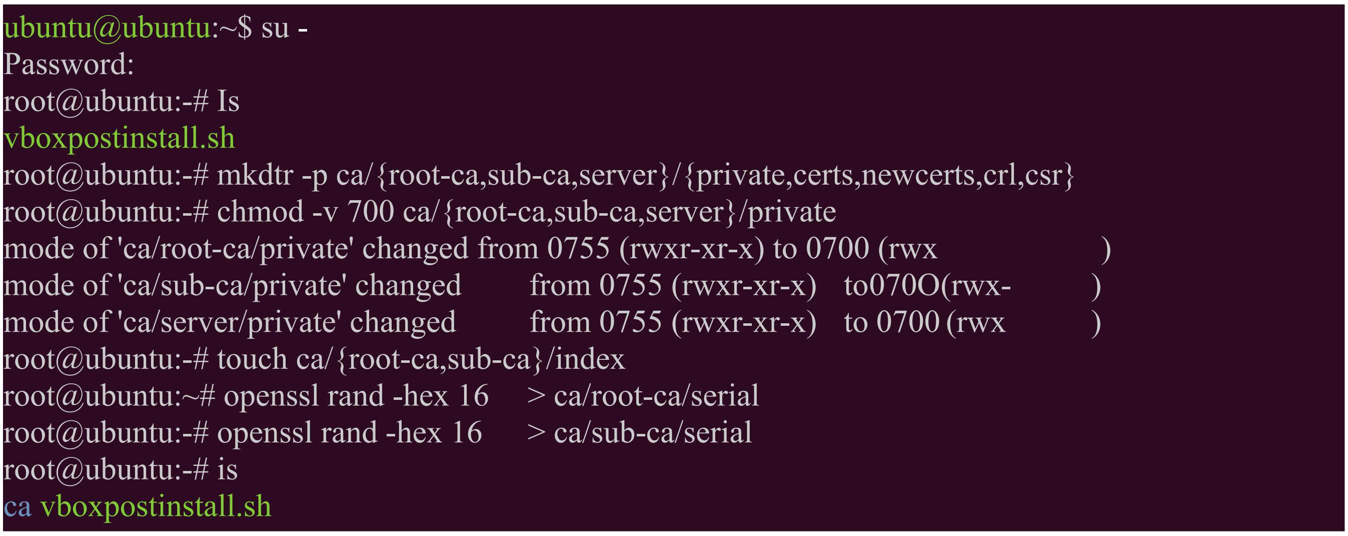
Then, I prepared all the directories

ii. Changing the root of the ca and sub-ca private folder
   —
   chmod -v 700 ca/{root-ca,sub-ca,server}/private

iii. Creating a file index in both root ca and sub ca
   —
   touch ca/{root-ca,sub-ca}/index

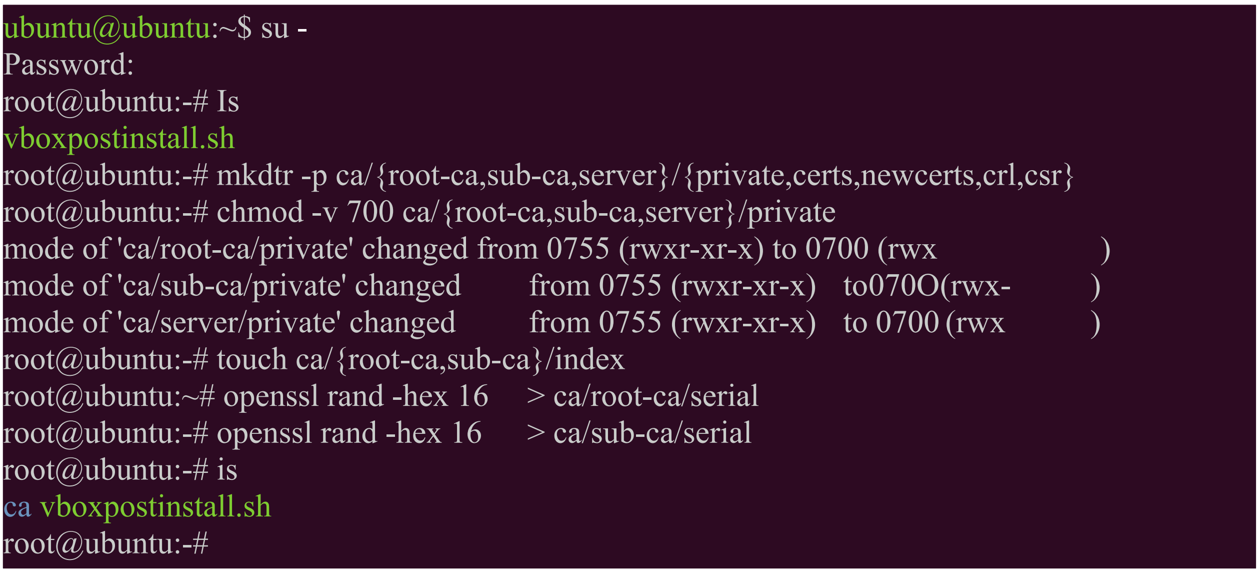
iv. Writing serial number of the root ca
   —
   openssl rand -hex 16 > ca/root-ca/serial

v. writing serial number of sub ca
   —
   openssl rand -hex 16 > ca/sub-ca/serial

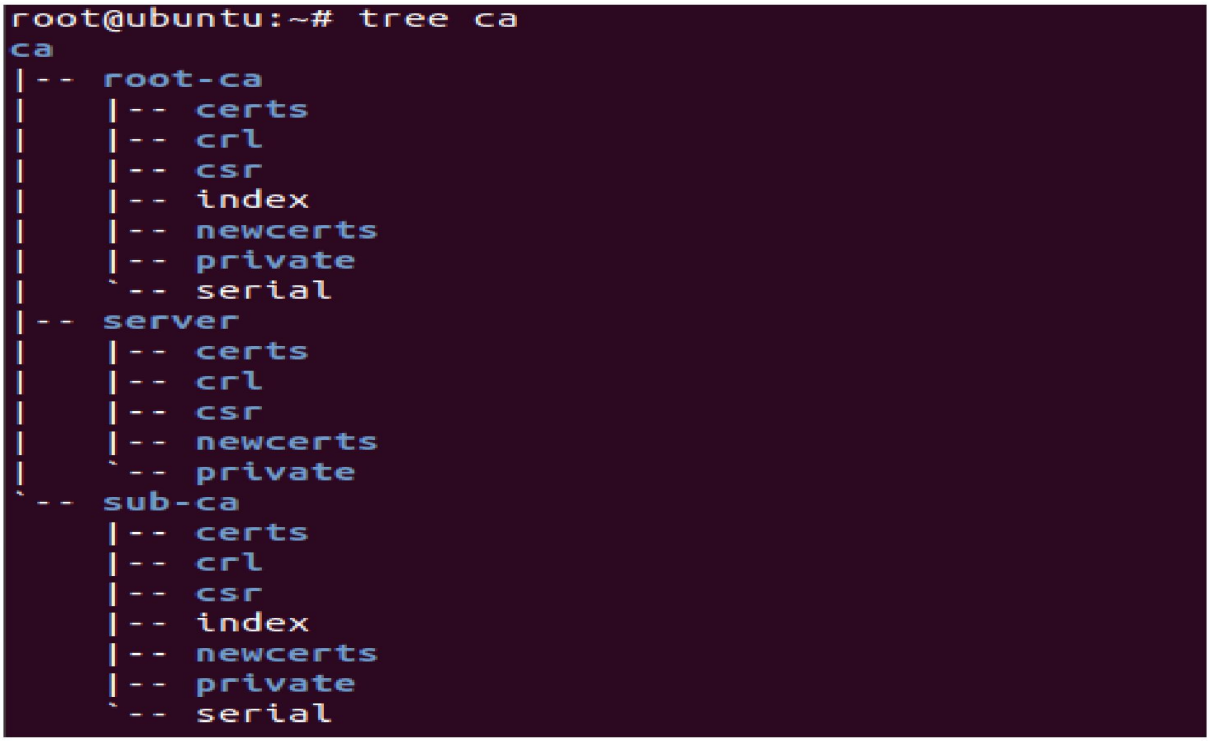


```
ubuntu@ubuntu:~$ su -
Password:
root@ubuntu:-# Is
vboxpostinstall.sh
root@ubuntu:-# mkdtr -p ca/{root-ca,sub-ca,server}/{private,certs,newcerts,crl,csr}
root@ubuntu:-# chmod -v 700 ca/{root-ca,sub-ca,server}/private
mode of 'ca/root-ca/private' changed from 0755 (rwxr-xr-x) to 0700 (rwx          )
mode of 'ca/sub-ca/private' changed     from 0755 (rwxr-xr-x)   to070O(rwx-       )
mode of 'ca/server/private' changed     from 0755 (rwxr-xr-x)   to 0700 (rwx      )
root@ubuntu:-# touch ca/{root-ca,sub-ca}/index
root@ubuntu:~# openssl rand -hex 16    > ca/root-ca/serial
root@ubuntu:-# openssl rand -hex 16    > ca/sub-ca/serial
root@ubuntu:-# is
ca vboxpostinstall.sh
```

root@ubuntu:-#



```
ubuntu@ubuntu:~$ su -
Password:
root@ubuntu:-# Is
vboxpostinstall.sh
root@ubuntu:-# mkdtr -p ca/{root-ca,sub-ca,server}/{private,certs,newcerts,crl,csr}
root@ubuntu:-# chmod -v 700 ca/{root-ca,sub-ca,server}/private
mode of 'ca/root-ca/private' changed from 0755 (rwxr-xr-x) to 0700 (rwx                )
mode of 'ca/sub-ca/private' changed       from 0755 (rwxr-xr-x)   to070O(rwx-          )
mode of 'ca/server/private' changed       from 0755 (rwxr-xr-x)   to 0700 (rwx         )
root@ubuntu:-# touch ca/{root-ca,sub-ca}/index
root@ubuntu:~# openssl rand -hex 16     > ca/root-ca/serial
root@ubuntu:-# openssl rand -hex 16     > ca/sub-ca/serial
root@ubuntu:-# is
ca vboxpostinstall.sh
root@ubuntu:-#
```



```
root@ubuntu:~# tree ca
ca
|-- root-ca
|   |-- certs
|   |-- crl
|   |-- csr
|   |-- index
|   |-- newcerts
|   |-- private
|   `-- serial
|-- server
|   |-- certs
|   |-- crl
|   |-- csr
|   |-- newcerts
|   `-- private
`-- sub-ca
    |-- certs
    |-- crl
    |-- csr
    |-- index
    |-- newcerts
    |-- private
    `-- serial
```

Generating private key for root CA, sub CA, and server Public key for rootCA
Public key for rootCA

—

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

Public key for subCA

—

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

Public key for server

—

openssl genrsa -out server/private/server.key 2048

```
  root@ubuntu:~# cd ca
root@ubuntu:~/ca# Ls
root-ca server sub-ca
root@ubuntu:~/ca# openssl genrsa -aes256 -out root-ca/private/ca.key 4096
Generating RSA private key, 4096 bit long moduius
e is 65537 (OxOlOOOl)
Enter pass phrase for root-ca/private/ca.key:
Verifying - Enter pass phrase for root-ca/private/ca.key:
root@ubuntu:~/ca# openssl genrsa -aes256 -out sub-ca/private/sub-ca.key 4096
Generating RSA private key, 4096 bit Long noduLus
e is 65537 (0X010001)
Enter pass phrase for sub-ca/private/sub-ca.key:
  Verifying - Enter pass phrase for sub-ca/private/sub-ca.key:
root@ubuntu:~/ca# openssL genrsa -out server/private/server.key 2048
Generating RSA private key, 2048 bit long modulus
e is 65537 (OX0100O1)
```

Verifying the changes via the directories:

```
  root@ubuntu:~# tree ca
ca

  |-- root-ca
          |" certs
I I" crL


  I csr
|       |--     index
|       |--     neucerts
|       |--     private
|       |       *-- ca.key
|               sertal

  |-- server
          |--     certs
I          I-   crL


I               csr
|          |--   neucerts
|          '--   private
|               '-- server.key

  '-- sub-ca
          |-- certs
          |-- crL
          |-- csr
          |-- index
```

```
        |-- neucerts
        |-- private
        |       '-- sub-ca.key
        '-- sertal
```

Create a file named root-ca.conf and paste the following code:

```
[ca]

#/root/ca/root-ca/root-ca.conf

#see man ca

default_ca = CA_default

[CA_default]

dir =/root/ca/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 ]

Country name = Country Name (2 letter code)

stateOrProvinceName = State or Province Name

localityName  = Locality Name

O.organizationName = Organization Name

organizationalUnitName = Organizational Unit Name

commonName = cyberproject

emailAddress = cybergroup@gmail.com

countryName_default = BD

stateOrProvinceName_default = Dhaka

0.organizationName_default = ATMS

[ v3_ca]
```

```
#  Extensions to apply when creating root ca

# Extensions for a typical CA, man x509v3_config

subjectKeyldentifier = hash

authorityKeyldentifier = 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

subjectKeyldentifier = hash

authorityKeyldentifier = keyid: always, issuer

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

basicConstraints = critical, CA: true, pathlen:0

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ server_cert ]

#  Extensions for server certificates

basicConstraints = CA:FALSE

nsCertType = server

nsComment = "OpenSSL Generated Server Certificate"

subjectKeyldentifier = hash
```

authorityKeyldentifier = keyid,issuer:always

keyUsage = critical, digitalSignature, keyEncipherment

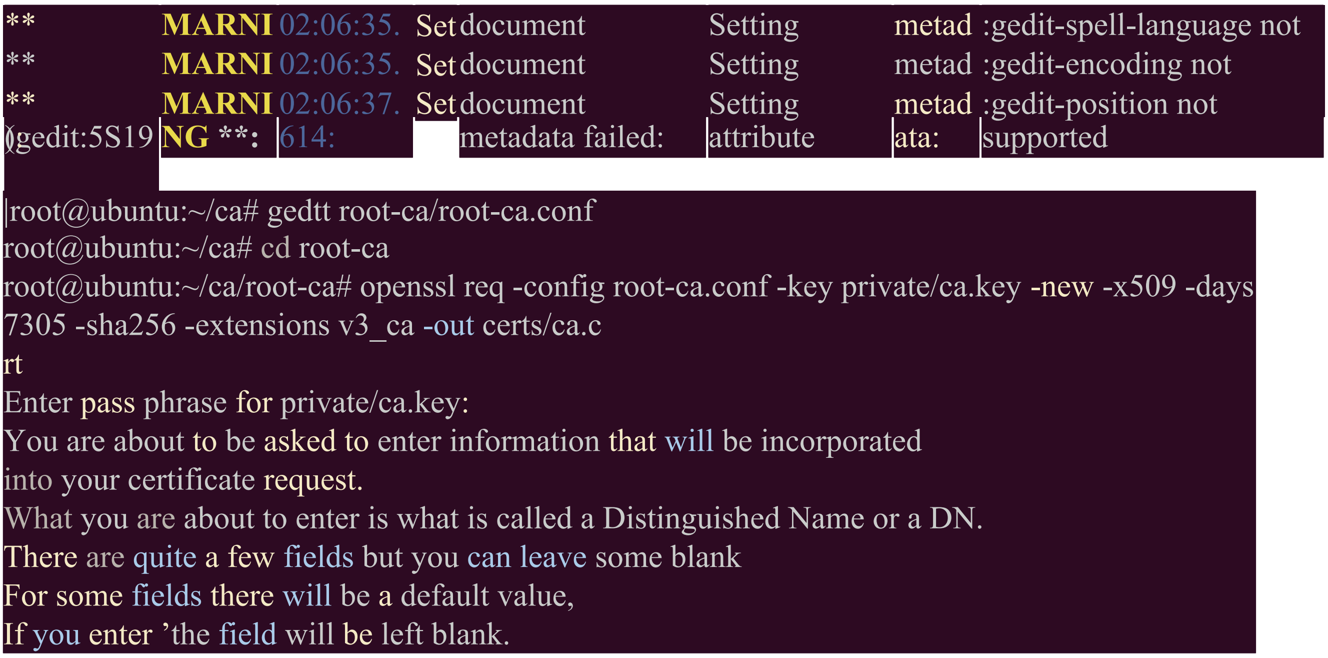
extendedKeyUsage = serverAuth

- Generating root CA certificate
- Ensuring that the certificate has been created properly

Moving inside root-ca
—
cd root-ca

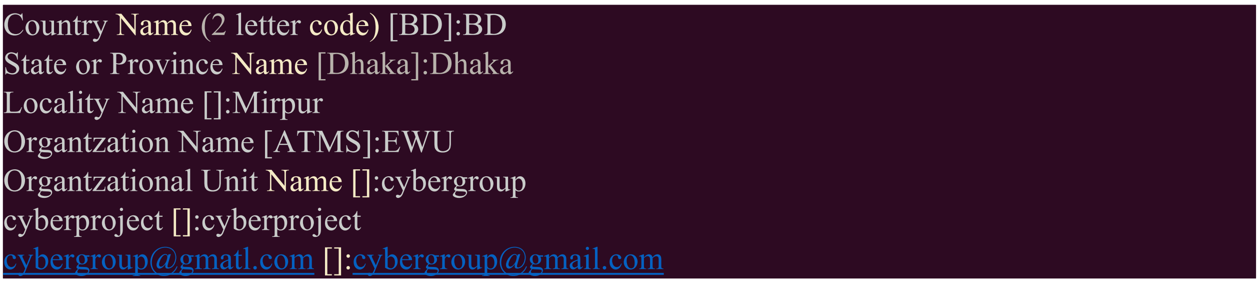
Generating root CA certificate
—
openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 7305 -sha256 -extensions v3_ca -out certs/ca.crt

Ensuring that the certificate has been created properly
—
openssl x509 -noout -in certs/ca.crt -text



```
**          MARNI 02:06:35.  Set document        Setting      metad :gedit-spell-language not
**          MARNI 02:06:35.  Set document        Setting      metad :gedit-encoding not
**          MARNI 02:06:37.  Set document        Setting      metad :gedit-position not
(gedit:5S19 NG **:  614:         metadata failed: attribute    ata:   supported

|root@ubuntu:~/ca# gedtt root-ca/root-ca.conf
root@ubuntu:~/ca# cd root-ca
root@ubuntu:~/ca/root-ca# openssl req -config root-ca.conf -key private/ca.key -new -x509 -days
7305 -sha256 -extensions v3_ca -out certs/ca.c
rt
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.
```

Root-Ca Certificate :



```
Country Name (2 letter code) [BD]:BD
State or Province Name [Dhaka]:Dhaka
Locality Name []:Mirpur
Organtzation Name [ATMS]:EWU
Organtzational Unit Name []:cybergroup
cyberproject []:cyberproject
cybergroup@gmatl.com []:cybergroup@gmail.com
```

```
root@ubuntu:~/ca/root-ca# openssi x509 -noout -in certs/ca.crt -text
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            ee:b2:e9:5b:85:3f:8f:46
    Stgnature Algorithm: sha256WtthRSAEncryptton
        Issuer: C = BD, ST = Dhaka, L = Mirpur, 0 = EWU, OU = cybergroup, CN = cyberproject,
        ematlAddress = cybergroup@gmail.com
        Validtty
        Not Before: Jan 09 29:10:17 2025 GMT
        Not After : Jan 09 20:10:17 2045 GMT
        Subject: C = BD, ST = Dhaka, L = Mirpur, 0 = EWU, OU = cybergroup, CN = cyberproject,
        emailAddress = cybergroup@gmail.com
        Subject Public Key Info:
            Publtc Key Algorithm: rsaEncryption
```

Subject Public Key Info: Public Key Algorithm: rsaEncryption

```
                Public-Key: (4096 bit)
                Modulus:
                    00:c7:c6:58:21:77:b1:a5:0e:96:86:90:25:07:2e:
                    1f:4c:99:47:9c:96:8d:01:ae:77:24:b8:73:97:df:
                    46:89:7b:1b:c4:28:fe:60:42:d8:4d:5f:2d:89:a0:
                    be:9a:0e:6e:21:29:11:c2:ac:88:8a:48:c2:15:52:
                    7b:6e:b1:69:78:c2:f9:8f:f1:56:e2:6b:f7:ca:e9:
                    84:cc:c8:31:f7:b1:62:49:df:dc:4c:39:fa:87:17:
                    15:8a:1e:30:2f:45:9f:70:39:5b:00:3c:a4:60:52:
                    fb:fd:1e:c9:7b:bc:82:58:66:45:19:fe:61:ba:01:
                    b9:91:2c:d2:c1:54:aa:7a:28:d8:6b:93:50:96:72:
                    ed:28:e5:94:a9:a0:2c:9c:29:69:8b:d4:c2:e4:73:
                    f1:02:05:a4:e8:ab:d4:5e:96:85:91:4e:a7:fe:0f:
                    3f:7c:31:40:72:00:be:83:81:76:3a:9c:81:d8:a7:
                    70:db:e5:b2:82:97:12:b7:8a:34:f4:c0:e6:e2:1b:
                    c0:25:d9:4d:bb:44:ad:27:95:8c:ab:2e:9f:2c:bb:
                    46:ce:09:47:6b:12:c0:ea:30:d5:5c:f7:81:ae:93:
                    76:38:73:99:b5:a1:5f:3d:75:26:b4:52:84:c5:ea:
                    58:f5:fb:aa:82:98:06:f1:48:d5:2f:1d:7d:20:0e:
                    25:9d:6e:d1:0b:5a:56:51:ec:9e:33:86:a5:06:96:
                    cc:fb:5d:cd:f8:e2:ea:39:8d:b4:f4:55:9a:94:06:
                    df:9d:26:dc:6c:f4:4b:e1:37:a4:cf:51:58:09:11:
                    41:90:8e:12:fc:e7:0a:20:d5:e9:b2:7a:9c:79:11:
                    e2:f7:9f:46:20:56:2a:92:11:5f:a8:85:af:92:43:
                    75:5f:a5:e4:0f:f9:a3:ba:03:a2:cd:9a:30:bc:21:
                    d7:c8:ef:bc:bc:ab:29:19:9e:57:43:25:98:ba:99:
                    14:e3:0d:a7:8a:19:98:cd:65:2e:4d:09:3a:ca:db:
                    27:6e:8a:69:5a:fd:05:f5:59:00:02:c4:d6:4c:7d:
                    64:68:a3:c8:b3:55:ee:2a:63:1c:68:f8:92:76:23:
                    0f:69:e1:db:d8:59:88:c8:39:49:d3:8a:a3:7a:a1:
                    6b:b0:ec:97:75:9d:58:2a:c0:aa:5c:d4:b5:16:0c:
                    17:66:21:8f:ec:34:9e:a0:a1:0a:d9:90:e0:8f:f5:
                    62:2c:a8:8f:25:da:52:d4:a7:38:b3:a2:8c:e4:28:
                    a7:a2:0f:d5:df:90:d8:ee:c9:ab:76:86:ff:ff:82:
                    d8:51:08:42:9b:ce:e0:d0:98:91:68:27:91:06:c5:
                    f2:dc:bc:44:0c:3d:15:c3:f4:e1:57:45:03:b4:67:
                    6b:e9:e1
Exponent: 65537 (0x10001)
```

```
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            9F:52:19:2D:CC:72:66:C0:59:24:55:D0:24:7C:97:C0:BA:6A:73:D
        X509v3 Authority Key Identifier:
            keyid:9F:52:19:2D:CC:72:66:C0:59:24:55:D0:24:7C:97:C0:BA:6A


        X509v3 Basic Constraints: critical
            CA:TRUE
        X509v3 Key Usage: critical
            Digital Signature, Certificate Sign, CRL Sign
Signature Algorithm: sha256WithRSAEncryption

    50:7f:98:96:72:f6:8e:31:c4:f9:67:0b:c6:71:4a:1c:e6:5b:
    6c:a3:16:15:87:64:dc:ad:9b:e9:6e:15:da:60:37:8d:a4:88:
    0f:c2:8d:f6:03:12:d4:36:06:54:e4:dd:ab:ff:b6:8d:a4:0d:
    1f:bb:bc:91:c8:02:23:63:3b:df:4a:70:35:26:75:97:b9:4e:
    63:1b:ac:c7:e8:e7:b9:64:7e:93:0f:e3:70:8a:cb:56:06:7f:
    7b:fa:6e:38:01:2c:95:b8:6a:00:05:81:12:fe:35:c7:fd:46:
    68:62:b2:56:05:87:25:56:0c:a2:01:bc:a3:a5:2c:f3:75:42:
    50:8d:68:5c:4d:c1:16:3c:63:fc:aa:e5:e6:6a:18:f4:7a:77:
    4b:94:78:92:89:a7:55:d0:16:ce:ad:a3:86:8f:ff:69:11:50:
    dd:f5:53:08:14:9c:e1:8c:1b:6f:50:ef:3b:f0:d5:16:59:71:
    ce:e3:82:cc:6c:42:bb:2b:8e:20:71:42:fc:c4:c8:51:a3:34:
    ff:84:4c:e1:6c:07:24:a9:4b:88:78:0c:4f:ce:5a:47:80:5d:
    7e:1c:ee:62:82:1f:49:db:3b:1b:16:a5:13:87:86:ab:50:6d:
    e3:87:44:71:f7:31:cb:90:ff:6c:32:44:dd:54:60:f4:8a:a4:
    fe:ff:ef:d5:21:9c:30:66:c2:86:bf:1f:0d:17:24:5d:29:af:
    b4:84:40:2e:7d:72:d6:69:70:65:fb:ae:f3:8f:0a:42:80:b1:
    e9:00:71:4c:d8:12:a6:c3:73:48:27:7a:89:2c:c7:3e:47:50:
    72:cd:43:49:78:39:f9:ae:50:c7:93:9a:8f:08:23:5b:0f:ae:
    6b:ac:9e:51:ab:72:16:23:e3:72:05:75:7e:a1:cb:98:e9:80:
    2f:ea:7c:f5:61:6d:40:de:da:f3:48:23:d1:0e:e0:26:e3:e1:
    64:70:6d:b7:71:76:10:d0:4f:e9:d3:a6:78:f5:0f:37:12:a9:
    1f:89:6b:9c:b7:da:b3:f7:47:4e:ed:ad:89:21:a3:99:17:a4:
    aa:a4:fb:ec:35:c8:58:a4:89:62:37:7d:c8:2d:50:4e:8d:56:
    13:d8:1c:30:bf:79:ae:67:1c:49:e6:cc:82:72:c9:90:e1:6c:
    ac:c5:dd:04:4e:6d:67:54:01:23:d5:c7:c7:9d:2e:43:2e:30:
    44:2f:09:ab:48:5d:d3:f3:ae:0c:51:8b:7f:1c:be:5b:84:ae:
    9c:a2:f2:ef:27:c2:0e:3e:90:ad:74:a8:76:e4:7d:02:d3:50:
    8c:14:43:94:72:c5:2c:74:47:49:e4:c5:16:c4:1d:6c:0d:5b:
    28:fd:af:57:58:7d:b8:7a
```

Moving a step back and then to sub-ca

—

```
cd ../sub-ca
```

Sub-CA

Creating sub-ca.config

—

```
gedit sub-ca.conf
```

Inserting the code into sub-ca.config file

—

```
[ca]

#/root/ca/sub-ca/sub-ca.conf

#see man ca

default_ca = CA_default

[CA_default]

dir =/root/ca/sub-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/sub-ca.key

certificate = $dir/certs/sub-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_loose

[ 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

O.organizationName = Organization Name

organizationalUnitName = Organizational Unit Name

commonName = cyberproject

emailAddress = cybergroup@gmail.com

countryName_default = BD

stateOrProvinceName_default = Dhaka

0.organizationName_default = ATMS

[ v3_ca]

#   Extensions to apply when createing root ca

#   Extensions for a typical CA, man x509v3_config

subjectKeyldentifier = hash

authorityKeyldentifier = 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

subjectKeyldentifier = hash

authorityKeyldentifier = keyid:always,issuer

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

```
basicConstraints = critical, CA:true, pathlen:0

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ server_cert ]

#  Extensions for server certificates

basicConstraints = CA:FALSE
nsCertType = server

nsComment = "OpenSSL Generated Server Certificate"

subjectKeyldentifier = hash

uthorityKeyldentifier = keyid, issuer:always

keyUsage = critical, digitalSignature, keyEncipherment

extendedKeyUsage = serverAuth
```

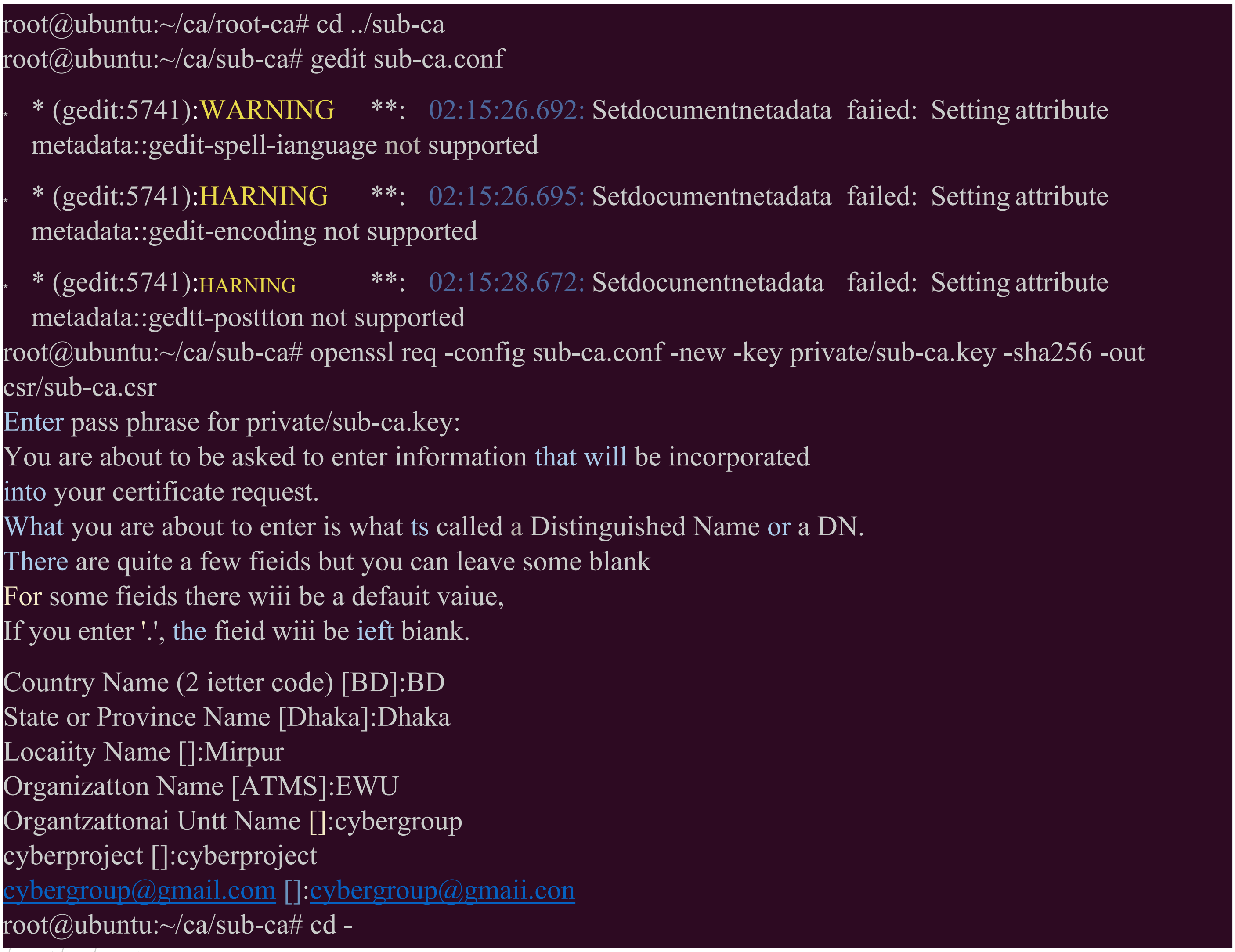
Saving and exiting

Requesting for sub ca certificate signing request.

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

Moving to the previous folder

```
cd
```



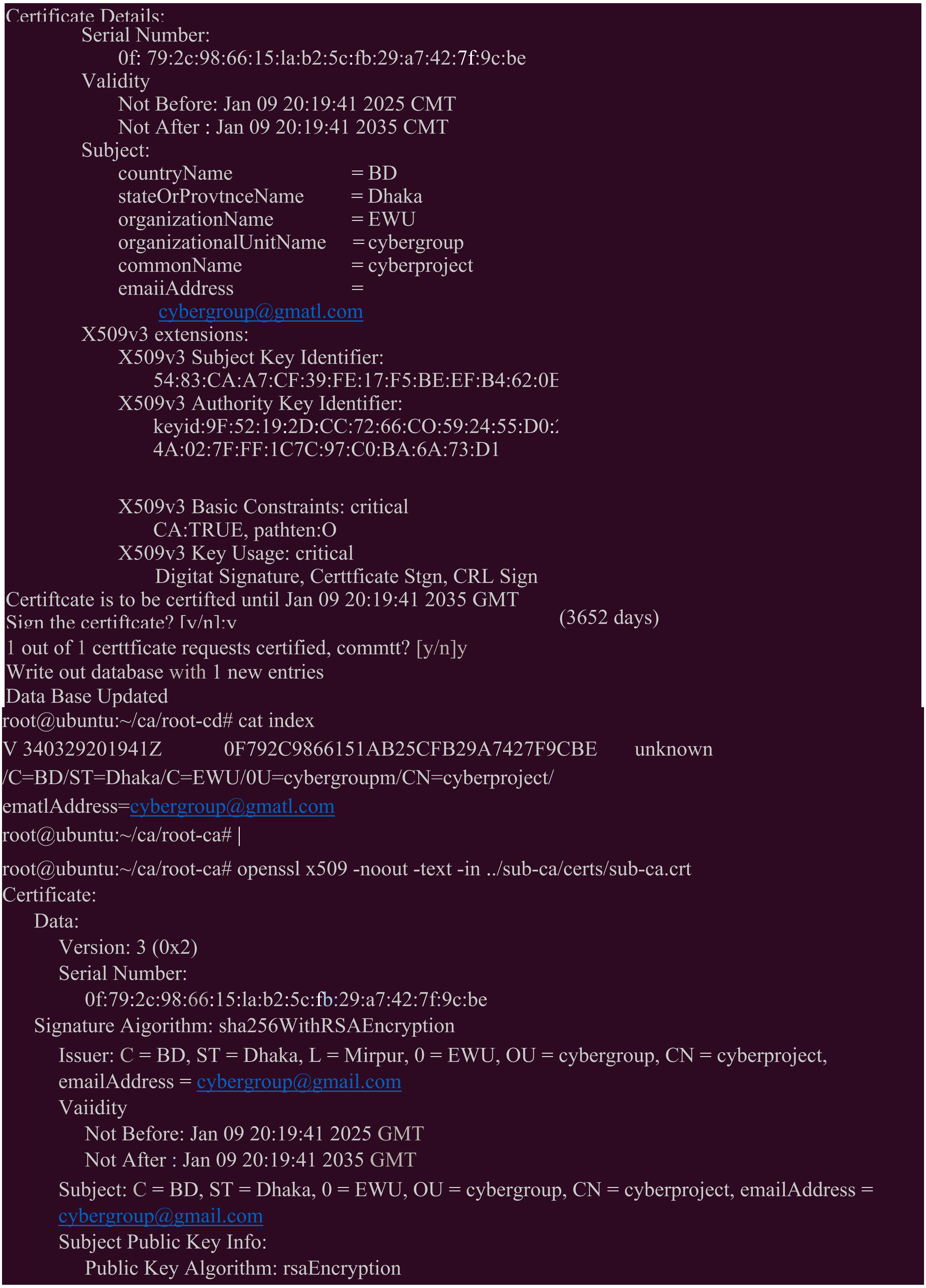
```
root@ubuntu:~/ca/root-ca# cd ../sub-ca
root@ubuntu:~/ca/sub-ca# gedit sub-ca.conf

** (gedit:5741):WARNING **: 02:15:26.692: Setdocumentnetadata faiied: Setting attribute
   metadata::gedit-spell-ianguage not supported

** (gedit:5741):HARNING **: 02:15:26.695: Setdocumentnetadata failed: Setting attribute
   metadata::gedit-encoding not supported

** (gedit:5741):HARNING **: 02:15:28.672: Setdocunentnetadata failed: Setting attribute
   metadata::gedtt-posttton not supported
root@ubuntu:~/ca/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 ts called a Distinguished Name or a DN.
There are quite a few fieids but you can leave some blank
For some fieids there wiii be a defauit vaiue,
If you enter '.', the fieid wiii be ieft biank.

Country Name (2 ietter code) [BD]:BD
State or Province Name [Dhaka]:Dhaka
Locaiity Name []:Mirpur
Organizatton Name [ATMS]:EWU
Organtzattonai Untt Name []:cybergroup
cyberproject []:cyberproject
cybergroup@gmail.com []:cybergroup@gmaii.con
root@ubuntu:~/ca/sub-ca# cd -
/root/ca/root-ca
root@ubuntu:~/ca/root-ca#
```

- Generating sub-ca certificate
- Ensuring that the certificate has been created properly

Signing the request of sub ca by root ca
—

```
openssl ca -config root-ca.conf -extensions v3_intermediate_ca -days
3652 -notext -in ../sub-ca/csr/sub-ca.csr
-out ../sub-ca/certs/sub-ca.crt
```

```
root@ubuntu:~/ca/root-ca# openssi ca -conftg root-ca.conf -extensions v3_tnternedtate_ca -days 3652 -
notext -in ../sub-ca/csr/sub-ca.csr -out
../sub-ca/certs/sub-ca.crt
Using conftguratton from root-ca.conf
Enter pass phrase for /root/ca/root-ca/private/ca.key:
Can't open /root/ca/root-ca/index.attr for readtng, No such fiie or directory
140464032383424:error:020O1002:system itbrary:fopen:No such fiie or
dtrectory:../crypto/bio/bss_ftie.c:74:fopen('/root/ca/root-ca/tndex.attr',
'r')
14O464032383424:error:2006DO80:BIO routines:BIO_new_ftie:no such fiie:../crypto/bto/bss_ftie.c:81:
Check that the request matches the signature
Signature ok
```



```
Certificate Details:
          Serial Number:
               0f: 79:2c:98:66:15:la:b2:5c:fb:29:a7:42:7f:9c:be
          Validity
               Not Before: Jan 09 20:19:41 2025 CMT
               Not After : Jan 09 20:19:41 2035 CMT
          Subject:
               countryName               = BD
               stateOrProvtnceName       = Dhaka
               organizationName          = EWU
               organizationalUnitName    = cybergroup
               commonName                = cyberproject
               emaiiAddress              =
                    cybergroup@gmatl.com
          X509v3 extensions:
               X509v3 Subject Key Identifier:
                    54:83:CA:A7:CF:39:FE:17:F5:BE:EF:B4:62:0E
               X509v3 Authority Key Identifier:
                    keyid:9F:52:19:2D:CC:72:66:CO:59:24:55:D0::
                    4A:02:7F:FF:1C7C:97:C0:BA:6A:73:D1


               X509v3 Basic Constraints: critical
                    CA:TRUE, pathten:O
               X509v3 Key Usage: critical
                    Digitat Signature, Certtficate Stgn, CRL Sign
Certiftcate is to be certifted until Jan 09 20:19:41 2035 GMT
Sign the certiftcate? [y/n]:y                                        (3652 days)
1 out of 1 certtficate requests certified, commtt? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@ubuntu:~/ca/root-cd# cat index
V 340329201941Z        0F792C9866151AB25CFB29A7427F9CBE      unknown
/C=BD/ST=Dhaka/C=EWU/0U=cybergroupm/CN=cyberproject/
ematlAddress=cybergroup@gmatl.com
root@ubuntu:~/ca/root-ca# |
root@ubuntu:~/ca/root-ca# openssl x509 -noout -text -in ../sub-ca/certs/sub-ca.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            0f:79:2c:98:66:15:la:b2:5c:fb:29:a7:42:7f:9c:be
    Signature Aigorithm: sha256WithRSAEncryption
        Issuer: C = BD, ST = Dhaka, L = Mirpur, 0 = EWU, OU = cybergroup, CN = cyberproject,
        emailAddress = cybergroup@gmail.com
        Vaiidity
            Not Before: Jan 09 20:19:41 2025 GMT
            Not After : Jan 09 20:19:41 2035 GMT
        Subject: C = BD, ST = Dhaka, 0 = EWU, OU = cybergroup, CN = cyberproject, emailAddress =
        cybergroup@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
```

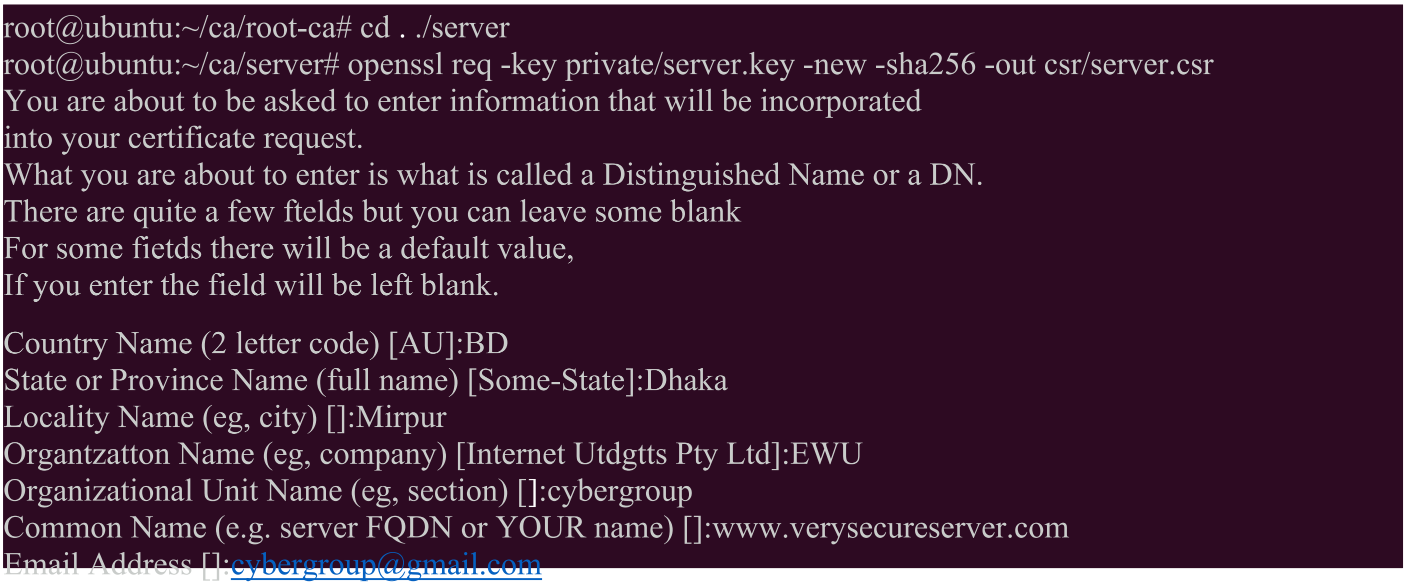
Sub-ca Certificate:

```
Public-Key: (4096 bit)
            Modulus:
                00:e2:84:1e:38:52:bc:5c:e0:50:24:bd:b5:a0:7e:
                28:e3:e8:95:9b:a0:33:ab:bf:14:15:07:37:8e:5b:
                a9:12:52:61:34:cf:8f:45:89:85:a1:30:0f:ce:36:
                47:6c:55:a1:5f:f6:e3:24:8b:c9:c9:c8:68:d6:c2:
                7b:cc:0e:d3:b3:66:09:54:24:fa:10:e0:b9:83:b7:
                be:d9:88:76:df:a0:89:25:74:d3:7c:be:1d:09:7a:
                a6:f1:d4:93:83:25:94:a5:16:0a:84:0d:f6:7a:36:
                1d:f4:af:4d:b2:b1:86:cd:05:37:be:bd:bd:d6:36:
                90:c4:af:cb:47:bd:90:54:83:6e:8f:4e:21:1e:52:
                43:43:81:3f:1a:44:48:66:71:43:de:2f:53:c6:44:
                e0:34:24:1a:32:5d:a6:67:77:f1:aa:3c:b8:79:8d:
                ea:25:4a:a2:95:0e:0e:67:66:4c:66:ac:32:bf:28:
                ce:07:8f:4d:a8:21:8b:ef:86:a7:45:81:ae:80:1d:
                6e:f2:c9:bb:50:3c:9c:91:29:81:c9:96:10:91:89:
                05:e3:a6:83:d0:c3:26:5d:42:4c:62:57:6e:b8:db:
                20:47:a6:a3:e2:56:5a:f7:27:c3:42:ee:43:9e:12:
                15:cf:55:8f:15:8a:92:73:42:3e:90:3e:70:02:02:
                ae:0b:e0:ce:2d:cf:a1:6f:88:38:14:ac:76:b1:0d:
                c8:f4:1f:95:c4:31:be:16:86:0e:8f:bd:b5:3b:e0:
                9f:34:1e:7b:cb:4f:10:f9:3a:76:2f:cc:38:87:7a:
                3d:f5:86:23:4f:39:3a:47:ca:a7:36:e6:50:e3:9b:
                d1:ac:4d:a4:a6:31:91:f6:86:db:13:73:95:3f:ee:
                d0:35:25:4a:85:55:60:83:c4:6b:78:78:96:ab:ce:
                36:9b:08:ee:2d:12:5c:7e:80:b4:57:c8:97:0a:33:
                ec:3c:08:29:91:42:9e:cb:13:aa:43:4a:b8:01:d0:
                e7:69:06:97:9b:67:62:df:30:80:a5:21:78:eb:47:
                4f:be:53:a6:d4:fc:9e:16:db:a8:4a:93:c0:57:2d:
                cf:37:2c:9d:62:83:38:41:89:d9:19:90:3a:c3:b5:
                42:be:85:e2:84:93:de:0f:87:e4:9a:b8:60:8c:2a:
                79:fe:c9:43:82:01:41:1d:0f:6e:19:f9:fd:36:2a:
                2b:df:29:91:fe:80:0e:67:f7:b0:97:06:0c:16:40:
                2a:29:9b:ea:fd:1c:63:78:7a:6c:71:c5:48:09:09:
                52:fd:fd:b1:ee:2f:ac:93:d0:f3:33:1d:74:2d:b1:
                7f:e1:d9:70:ab:e5:0b:10:cf:87:9d:fc:03:38:59:
                dc:7e:eb
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                54:83:CA:A7:CF:39:FE:17:F5:BE:EF:B4:62:0E:7A:4A:02:7F:FF:1C
            X509v3 Authority Key Identifier:
                keyid:9F:52:19:2D:CC:72:66:C0:59:24:55:D0:24:7C:97:C0:BA:6A


            X509v3 Basic Constraints: critical
                CA: TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
```

Signature Algorithm:
sha256WithRSAEncryption

```
b6:97:c8:f1:8b:e4:68:d0:98:32:d3:d8:8c:8d:6d:bf:01:22:
9e:23:c3:fc:b5:81:76:d5:7d:17:a3:db:97:4c:95:54:36:f9:
08:c1:39:1b:a3:aa:44:16:db:52:b0:90:e3:52:0c:e8:7d:d3:
1f:89:44:68:86:5f:a7:a0:6a:e0:2a:15:41:12:13:4a:ac:e2:
08:fb:98:fc:bb:ad:b5:c7:0c:a2:5c:cd:da:ae:da:42:c7:41:
3b:1b:2f:90:24:a0:c5:1d:ac:2f:91:f0:b1:b6:b8:db:85:af:
d2:77:10:b0:de:a2:df:07:b7:b9:62:7e:6b:be:01:97:6f:98:
5c:3f:58:7d:a3:3f:7f:ea:55:f8:cb:46:a1:c0:12:3f:84:77:
c8:7c:84:bc:fc:1c:ae:a5:44:31:07:d7:07:b7:a5:9e:64:e8:
9d:28:3a:32:13:0b:0b:c7:ff:28:9d:22:81:93:dc:e2:e0:07:
96:eb:d0:74:3d:1a:9e:38:b6:4d:4b:ff:d5:11:55:18:3e:77:
30:4e:a5:d7:87:ad:41:e6:44:96:98:ca:c9:4d:58:8e:c4:97:
1f:4f:e7:23:05:d2:6e:4b:12:b1:9e:be:b7:f9:1a:61:a7:3f:
8a:7f:53:9b:5b:f3:5e:4e:95:0d:45:26:4d:a0:76:43:0a:49:
a5:c3:46:7a:32:85:01:c4:6d:6d:a7:2c:7a:b5:be:8a:3b:20:
c0:14:e9:46:e2:d3:8a:70:32:8e:e2:f3:71:3d:72:56:89:ac:
6c:57:9d:c4:c6:a8:52:c9:8e:71:a3:aa:12:b7:c2:e1:44:0d:
23:ad:90:89:fb:9f:03:a8:b3:fa:98:ba:ac:71:9c:e0:4b:cc:
3c:b3:77:d5:b5:fc:da:58:91:e0:f3:86:11:b0:8a:e8:a6:e2:
62:93:3c:8b:ca:36:18:8c:05:23:21:da:b9:14:20:3e:dd:b0:
a6:cd:3d:dd:34:b2:e7:c2:d9:dd:46:fd:94:5b:d6:e9:3c:4f:
0a:82:9b:9f:1d:d2:29:05:14:f2:88:95:c5:5b:e6:46:95:eb:
67:8d:91:ad:98:96:05:56:ff:da:ef:72:40:1d:4d:c8:5d:92:
d6:68:57:18:d7:56:c9:1c:ef:c8:9d:ec:ba:5a:cf:03:04:e4:
ec:0c:f0:4d:c7:10:34:3f:bc:df:68:58:cf:27:55:1f:6b:83:
0d:88:75:3d:a2:56:94:66:8e:19:b5:4f:61:08:f9:07:4a:71:
18:64:3c:52:db:2f:75:68:00:bc:00:1a:02:44:ae:df:66:eb:
10:df:5a:dd:57:24:a5:e8:13:2f:d2:bc:99:91:9c:8c:00:d2:
3b:4b:34:71:85:b5:5c:14
```

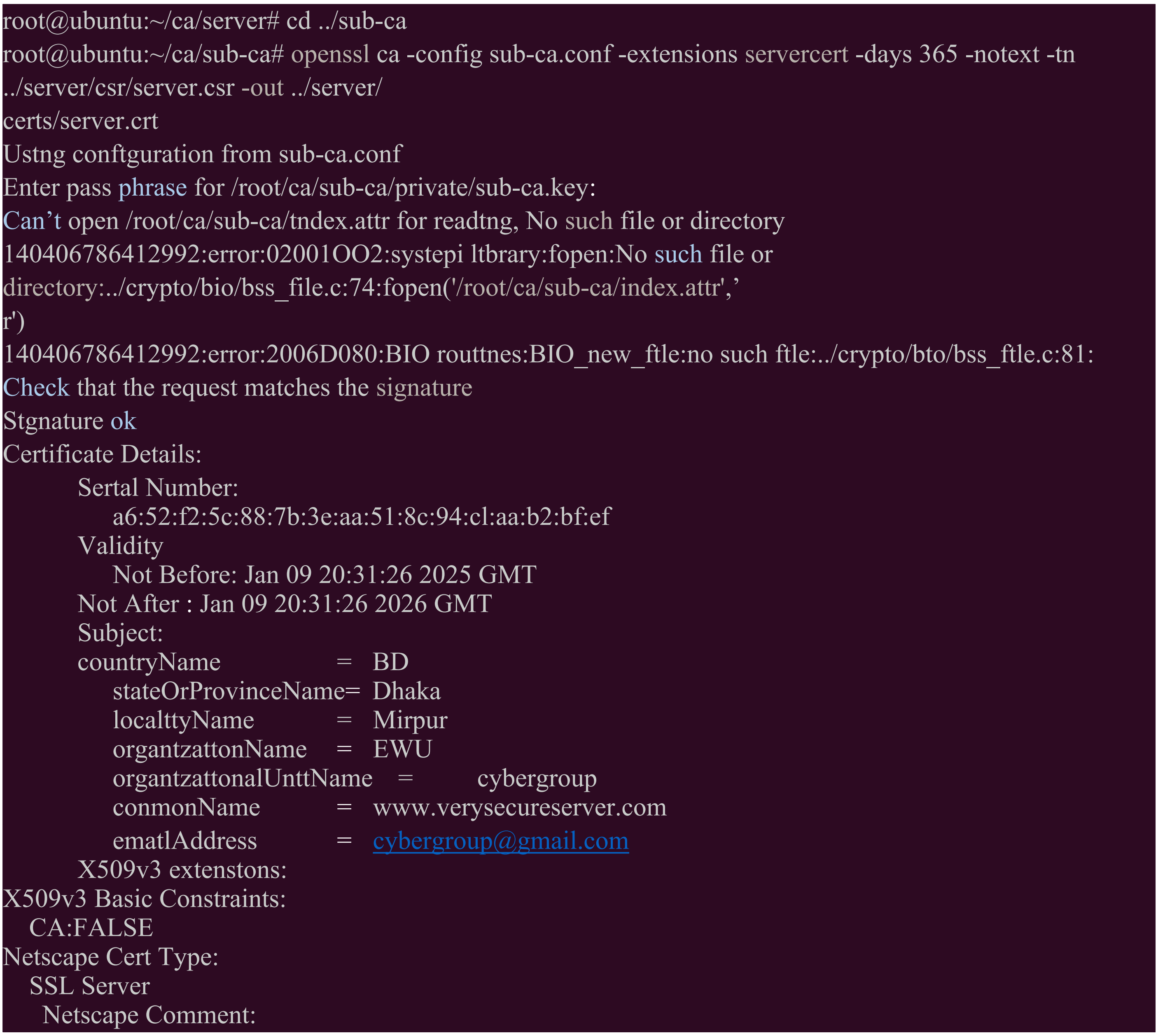
- Moving to server
- Generating certificate signing request from server
- openssl req -key private/server.key -new -sha256 -out csr/server.csr



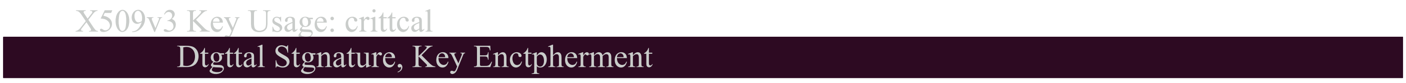
```
root@ubuntu:~/ca/root-ca# cd . ./server
root@ubuntu:~/ca/server# openssl req -key private/server.key -new -sha256 -out csr/server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few ftelds but you can leave some blank
For some fietds 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) []:Mirpur
Organtzatton Name (eg, company) [Internet Utdgtts Pty Ltd]:EWU
Organizational Unit Name (eg, section) []:cybergroup
Common Name (e.g. server FQDN or YOUR name) []:www.verysecureserver.com
Email Address []:cybergroup@gmail.com
```

Sub ca signing certificate request of server

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



```
root@ubuntu:~/ca/server# cd ../sub-ca
root@ubuntu:~/ca/sub-ca# openssl ca -config sub-ca.conf -extensions servercert -days 365 -notext -tn
../server/csr/server.csr -out ../server/
certs/server.crt
Ustng conftguration from sub-ca.conf
Enter pass phrase for /root/ca/sub-ca/private/sub-ca.key:
Can’t open /root/ca/sub-ca/tndex.attr for readtng, No such file or directory
140406786412992:error:02001OO2:systepi ltbrary:fopen:No such file or
directory:../crypto/bio/bss_file.c:74:fopen('/root/ca/sub-ca/index.attr',’
r')
140406786412992:error:2006D080:BIO routtnes:BIO_new_ftle:no such ftle:../crypto/bto/bss_ftle.c:81:
Check that the request matches the signature
Stgnature ok
Certificate Details:
         Sertal Number:
             a6:52:f2:5c:88:7b:3e:aa:51:8c:94:cl:aa:b2:bf:ef
         Validity
             Not Before: Jan 09 20:31:26 2025 GMT
         Not After : Jan 09 20:31:26 2026 GMT
         Subject:
         countryName              =  BD
             stateOrProvinceName=  Dhaka
             localttyName             =  Mirpur
             organtzattonName    =  EWU
             organtzattonalUnttName    =          cybergroup
             conmonName           =  www.verysecureserver.com
             ematlAddress           =  cybergroup@gmail.com
         X509v3 extenstons:
X509v3 Basic Constraints:
   CA:FALSE
Netscape Cert Type:
   SSL Server
     Netscape Comment:
      OpenSSL Generated Server Certificate
      X509v3 Subject Key Identifter:
        C8:58:24:5C:3D:F3:C5:50:DF:F7:F8:82:32:lD:63:0C:32:72:06:8C
  X509v3 Authority Key Identtfter:
      keyid:54:83:CA:A7:CF:39:FE:17:F5:BE:EF:B4:62:0E:7A:4A:02:7F:FF:lC
DtrName:/C=BD/ST=Dhaka/L=Mirpur/0=EWU/0U=cybergroup/CN=cyberproject/ematlAddress=cybergrou
p@gmatl.com
sertal:0F:79:2C:98:66:15:lA:B2:5C:FB:29:A7:42:7F:9C:BE
```

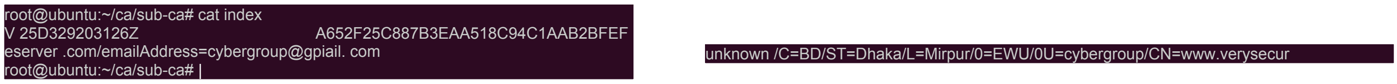


```
    X509v3 Key Usage: crittcal
        Dtgttal Stgnature, Key Enctpherment
```

X509v3 Extended Key Usage:
TLS Web Server Authentication

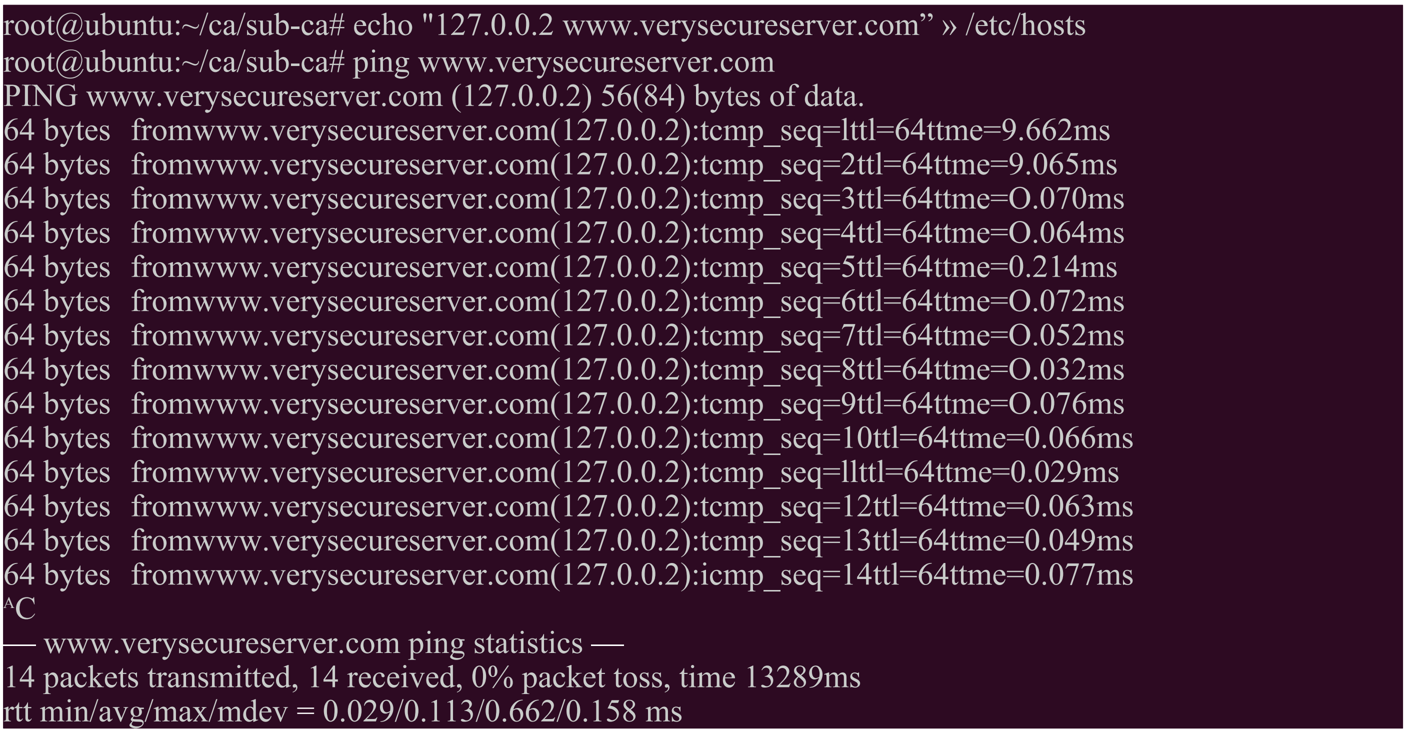
To see details

cat index



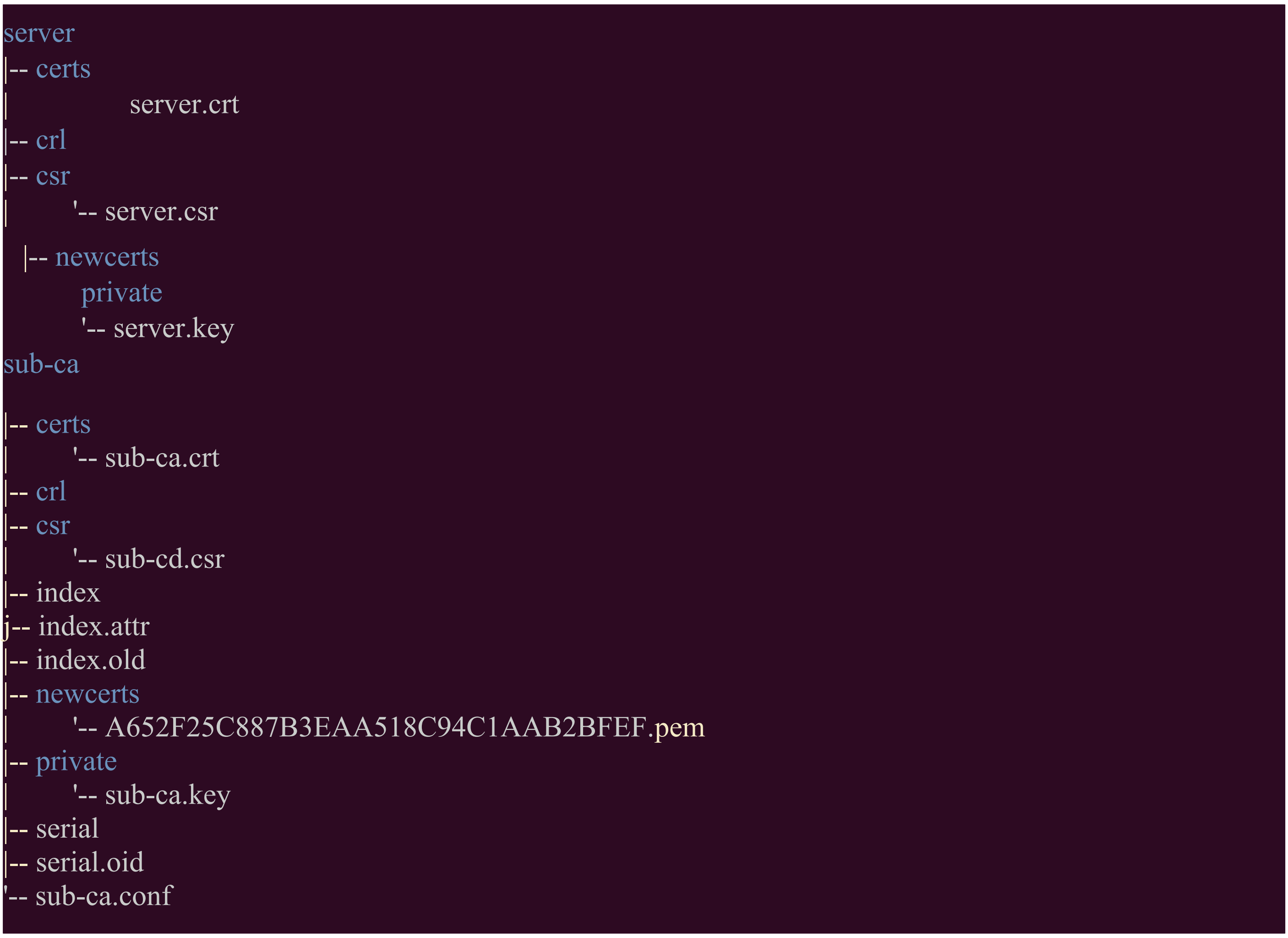
```
root@ubuntu:~/ca/sub-ca# cat index
V 25D329203126Z                    A652F25C887B3EAA518C94C1AAB2BFEF        unknown /C=BD/ST=Dhaka/L=Mirpur/0=EWU/0U=cybergroup/CN=www.verysecur
eserver .com/emailAddress=cybergroup@gpiail. com
root@ubuntu:~/ca/sub-ca# |
```

Verifying via the ping command



```
root@ubuntu:~/ca/sub-ca# echo "127.0.0.2 www.verysecureserver.com” » /etc/hosts
root@ubuntu:~/ca/sub-ca# ping www.verysecureserver.com
PING www.verysecureserver.com (127.0.0.2) 56(84) bytes of data.
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=lttl=64ttme=9.662ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=2ttl=64ttme=9.065ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=3ttl=64ttme=O.070ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=4ttl=64ttme=O.064ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=5ttl=64ttme=0.214ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=6ttl=64ttme=O.072ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=7ttl=64ttme=O.052ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=8ttl=64ttme=O.032ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=9ttl=64ttme=O.076ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=10ttl=64ttme=0.066ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=llttl=64ttme=0.029ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=12ttl=64ttme=0.063ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):tcmp_seq=13ttl=64ttme=0.049ms
64 bytes   fromwww.verysecureserver.com(127.0.0.2):icmp_seq=14ttl=64ttme=0.077ms
^C
— www.verysecureserver.com ping statistics —
14 packets transmitted, 14 received, 0% packet toss, time 13289ms
rtt min/avg/max/mdev = 0.029/0.113/0.662/0.158 ms
```

```
root-ca
 |-- certs
 |          ca.crt
 |-- crl
 |-- csr
 |-- index
 |-- index.attr
 |-- index.old
 |-- newcerts
 |     '-- 0F792C9866151AB25CFB29A7427F9CBE pern
 |-- private
 |     '-- ca.key
 |-- root-ca.conf
 |-- serial
 '-- serial.old
```



```
server
|-- certs
|               server.crt
|-- crl
|-- csr
|        '-- server.csr
  |-- newcerts
         private
         '-- server.key
sub-ca

|-- certs
|        '-- sub-ca.crt
|-- crl
|-- csr
|        '-- sub-cd.csr
|-- index
j-- index.attr
|-- index.old
|-- newcerts
|        '-- A652F25C887B3EAA518C94C1AAB2BFEF.pem
|-- private
|        '-- sub-ca.key
|-- serial
|-- serial.oid
'-- sub-ca.conf
```

Copying all certificates and pem file to certificate folder
And verifying via tree command

```
root@ubuntu: /# cp /root/ca/root-ca/newcerts/0F792C9866151AB25CFB29A7427F9CBE.pem
/home/group/certificate
root@ubuntu: /# cp /root/ca/sub-ca/newcerts/A652F25C887B3EAA518C94ClAAB2BFEF.pem
/home/group/certificate
root@ubuntu:/# cp /root/ca/root-ca/certs/ca.crt /home/group/certiftcate
root@ubuntu:/# cp /root/ca/sub-ca/certs/sub-ca.crt /home/group/certificate/
root@ubuntu:/# cp /root/ca/server/certs/server.crt /home/group/certificate/
root@ubuntu:/# cp /root/ca/server/private/server.key /home/group/certificate/
root@ubuntu:/# is
bin   cdrom     etc             tnitrd.img        tib   tost+found      mnt       proc run   snap    swapfite
        B5FI  var
boot  dev  home  initrd.img.old  lib64  media      opt   root  sbin  srv   sys        usr    vmtinuz
root@ubuntu:/# tree home
home
|-- group
|        *-- certificate

|        |--     A652F25C887B3EAA518C94C1AAB2BFEF.pem
            |  -  ca.crt
|        |--     server.crt
|        |--     server.key
           '-- sub-ca.crt
```

Editing the httpd-ssl.conf file

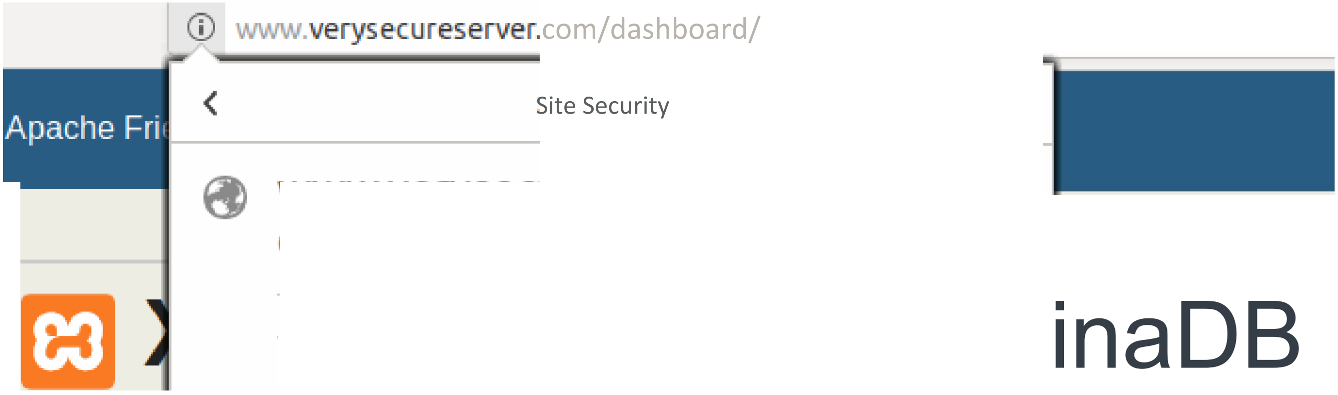
```
root@ubuntu:/# cd /opt/lanpp/etc/extra
root@ubuntu:/opt/lampp/etc/extra# chmod 777 httpd-ssl.conf
root@ubuntu:/opt/lampp/etc/extra# gedit httpdssl.conf
```

```
#       Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#       require an ECC certificate which can also be configured in
#       parallel. Below is line 106
SSLCertif icateFile "/home/group/certificate/server.crt"
#SSLCertificateFile "/opt/lampp/etc/server-dsa.crt"
#SSLCertificateFile "/opt/lampp/etc/server-ecc.crt"

#       Server Private Key:
#       If the key is not combined with the certificate, use this
#       directive to point at the key file. Keep in mind that if
#       you've both a RSA and a DSA private key you can configure
#       both in parallel (to also allow the use of DSA ciphers, etc. )
```

```
#       Server Private Key:
#       If the key is not combined with the certificate, use this
#       directive to point at the key file. Keep in mind that if
#       you've both a RSA and a DSA private key you can configure
#       both in parallel (to also allow the use of DSA ciphers, etc.)
#       ECC keys, when in use, can also be configured in parallel below is line 116
|SSLCertif icateKeyFile "/home/group/certificate/server.key"
#SSLCertificateKeyFile "/opt/lampp/etc/server-dsa.key"
#SSLCertificateKeyFile "/opt/lampp/etc/server-ecc.key"
```

```
#SSLCertificateChainFile "/opt/lampp/etc/server-ca.crt"

#       Certificate Authority (CA):
#       Set the CA certificate verification path where to find CA
#       certificates for client authentication or alternatively one
#       huge file containing all of them (file must be PEM encoded)
#       Note: Inside SSLCACertificatePath, you need hash symlinks
#       Point to the certificate files. Use the provided
#       Makefile to update the hash symlinks after changes.Below is line 136
SSLCACertificatePath "/home/group/certificate"
#SSLCACertificateFile "/opt/lampp/etc/ssl.crt/ca-bundle.crt"

#       Certificate Revocation Lists (CRL):
#       Set the CA revocation path where to find CA CRLs for client
#       authentication or alternatively one huge file containing all
# of them (file must be PEM encoded).
```

Primarily, www.verysecureserver.com is not secure
Before inserting all the certificates



www.verysecureserver.corn
Connection is Not Secure

Your connection to this site is not private. Information you submit could be viewed by others (like passwords, messages, credit cards, etc.)

Certificate Manager x

#Importing all necessary certificates

P cert <3

Search Results

Certificates

When a server requests your personal certificate

Select one automatically

- Ask you every time

Query OCSP responder servers to confirm the current validity of certificates

cert

View Certificates...

Security Devices...

Certificate Manager

Your Certificates People Servers Authorities Others

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
Visa eCommerce Root	Builtin Object Token
-WISeKey	
OISTE WISeKey Global Root GA CA	Builtin Object Token
OISTE WISeKey Global Root GB CA	Builtin Object Token
’XRamp Security Services Inc	
XRamp Global CA Root	Builtin Object Token

View...

Edit Trust... Import... Export... Delete or Distrust...

Your Certificates

People Servers Authorities Others

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device	E*
Entrust Root Certification Authority	Builtin Object Token	
" Entrust.net		
Entrust.net Premium 2048 Secure Server CA	Builtin Object Token	1
cyberproject	Software Security Device	
” FNMT-RCM		
A(" DA 17 rMMT.DCM	Oj.i’dlrin Dhiart: Tnkan	

Certificate Manager

Your Certificates People Servers Authorities Others

You have certificates on file that identify

Certificate Name	ExpiresOn	E-MailAddress	E5
-EWU			
www.verysecureserver.con	January 09, 2035	cybergroup@gmail.com	

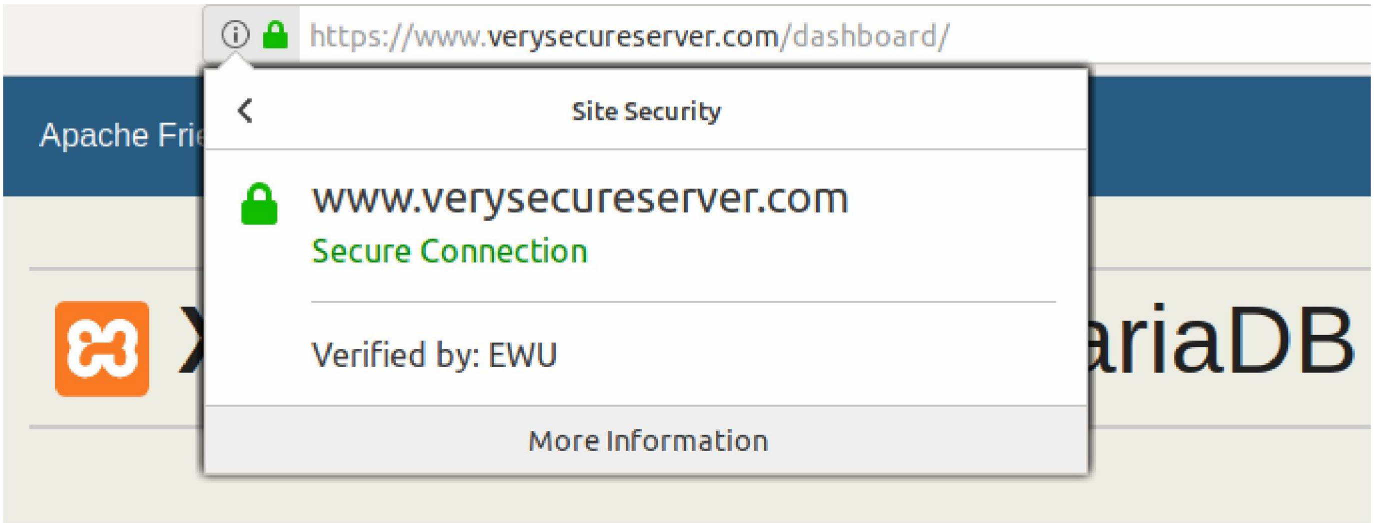
these people

View

...

#After importing all the necessary certificates

Import ... Export ... Delete ..



https://www.verysecureserver.com/dashboard/

Site Security

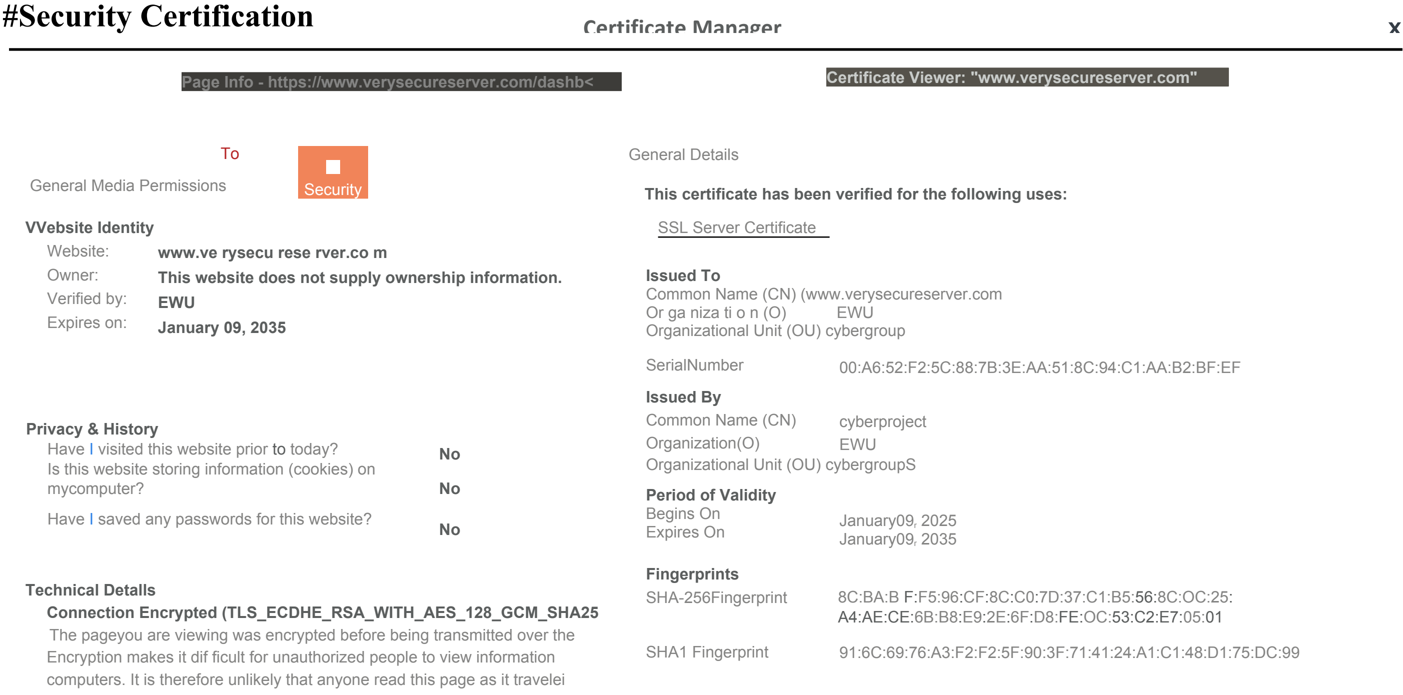
www.verysecureserver.com

Secure Connection

Verified by: EWU

More Information

#Security Certification



Certificate Manager x

Page Info - https://www.verysecureserver.com/dashb<

VVebsite Identity

Website:	**www.verysecureserver.com**
Owner:	**This website does not supply ownership information.**
Verified by:	**EWU**
Expires on:	**January 09, 2035**

Privacy & History

Have I visited this website prior to today?	**No**
Is this website storing information (cookies) on mycomputer?	**No**
Have I saved any passwords for this website?	**No**

Technical Detalls

Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA25

The pageyou are viewing was encrypted before being transmitted over the Encryption makes it dif ficult for unauthorized people to view information computers. It is therefore unlikely that anyone read this page as it travelei

Certificate Viewer: "www.verysecureserver.com"

General Details

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	(www.verysecureserver.com
Organization (O)	EWU
Organizational Unit (OU)	cybergroup
SerialNumber	00:A6:52:F2:5C:88:7B:3E:AA:51:8C:94:C1:AA:B2:BF:EF

Issued By

Common Name (CN)	cyberproject
Organization(O)	EWU
Organizational Unit (OU)	cybergroupS

Period of Validity

Begins On	January09, 2025
Expires On	January09, 2035

Fingerprints

SHA-256Fingerprint	8C:BA:B F:F5:96:CF:8C:C0:7D:37:C1:B5:56:8C:OC:25: A4:AE:CE:6B:B8:E9:2E:6F:D8:FE:OC:53:C2:E7:05:01
SHA1 Fingerprint	91:6C:69:76:A3:F2:F2:5F:90:3F:71:41:24:A1:C1:48:D1:75:DC:99

Conclusion:

The protocols TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are used to create encrypted and authorized connections between computers connected to a network. Our task involved using Public Key Infrastructure to implement Transport Layer Security (TLS) on HTTP for https:// connections to secure a networked system in this case (https://www.verysecureserver.com). At last, a secure website with a certificate from a reliable issuer has been achieved. We have utilized RSA for our public key.
The SHA-256 hash value is displayed in the certificate.
Lastly, it is demonstrated that a secured website has been created.