**East West University**

**Group No. 306**

**Group Members**

Md. Maimoon Hossain Shomoy

ID: 2019-1-60-105

Md Tabib Khan

ID: 2018-2-60-019

Submitted To:

Rashedul Amin Tuhin

Senior Lecturer

Department of Computer Science and Engineering

Submission date: 25/08/22

**Configuration of Certification Authority and Implementation of Transport Layer Security over HTTP**

**1. Preparing the environment and generating private key and certificates**

Step 1: install tree and curl in the linux machine.

Step 2: make the directories using:

**mkdir {root-ca,sub-ca,server}**

Step 3: Create private, certs, newecerts, crl, csr in the above directories:

**mkdir {root-ca,sub-ca,server}/{private,certs,newcerts,crl,csr}**

Step 4: check if all the directories are created or not using:

**tree ca**

Step 5: Creating file index in both root ca and sub ca:

**touch ca/{root-ca,sub-ca}/index**

Step 6: Check if indexes are created for root-ca and sub-ca:

**tree ca**

Step 7: Create private keys for ca, sub-ca and server using:

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

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

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

Step 8: check if the key were generated or not using:

**tree ca**

Step 9: Moving to root-ca folder:

**cd root-ca**

Step 10: Generate root certificate:

**openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 7200 -sha256 -extensions v3\_ca -out certs/ca.crt**

Step 11: Check the ca.crt:

**openssl x509 -noout -in certs/ca.crt -text**

Step 12: Moving to sub-ca folder:

**cd ../sub-ca/**

Step 13:Certificate signing request from RootCA to SubCA:

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

**cd ../root-ca/**

**openssl ca -config root-ca.conf -extensions v3\_intermediate\_ca -days 3650 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt -rand\_serial**

Step 14: check the certificate in .pem in root-ca folder:

**Tree**

Step 15: Check sub-ca .crt:

**openssl x509 -noout -text -in ../sub-ca/certs/sub-ca.crt**

Step 16: Moving to server folder:

**cd ../server**

Step 17: Server signing request:

**openssl req -config server.conf -key private/server.key -new -sha256 -out csr/server.csr**

Step 18: Generating server certificate:

**openssl req -config server.conf -key private/server.key -new -sha256 -out csr/server.csr**

**cd ../sub-ca**

**openssl ca -config sub-ca.conf -extensions server\_cert -days 365 -notext -in ../server/csr/server.csr -out ../server/certs/server.crt -rand\_serial**

Step 19: Moving out:

**cd ..**

Step 20:Creating chained certificate:

**cat ./server/certs/server.crt ./sub-ca/certs/sub-ca.crt > chained.crt**

**2. DNS Configuration**

Step 21: install apache2:

**Sudo apt install apache2**

Step 22: apache 2 commands:

**service apache2 stop**

**service apache2 start**

**service apache2 status**

**service apache2 restart**

**service apache2 reload**

Step 23: Changing the default file to main:

**sudo a2dissite 000-default**

**systemctl reload apache2**

**sudo a2ensite main**

**systemctl reload apache2**

Step 24: intall bind9 and dnsutils:

**sudo apt install bind9**

**sudo apt install dnsutils**

Step 25: Restart bind9 service:

**sudo systemctl restart bind9.service**

Step 25: Editing resolv.conf file:

nameserver 127.0.0.1

options edns0 trust-ad

search localdomain

Step 26: Editing named.conf file:

//

//named.conf

// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS

// server as a caching only name server (as a localhost DNS resolver only).

// See /usr/share/doc/bind\*/sample/ for example named configuration files.

//

options {

listen-on port 53 { 127.0.0.1;};

// listen-on-v6 port 53 { ::1; };

forwarders { 8.8.8.8; 8.8.4.4; };

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

allow-query { localhost; 192.168.0.0/24, 127.0.0.1 };

recursion yes;

dnssec-enable yes;

dnssec-validation yes;

dnssec-lookaside auto;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

};

logging {

channel default\_debug {

file "data/named.run";

severity dynamic;

};

};

zone "." IN {

type hint;

file "named.ca";

};

include "/etc/named.rfc1912.zones";

include "/etc/named.root.key";

Step 28: Checking google.com server

**dig google.com**

**nslookup google.com**

Step 29: Enabing and starting named

systemctl enable named

systemctl start named

Step 30: Editing verysecureserver.com.zone file:

; Authoritative data for maimoontabib.com zone

;

$TTL 1D

@ IN SOA verysecureserver.com root.verysecureserver.com. (

2022041301 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

$ORIGIN verysecureserver.com.

verysecureserver.com. IN NS verysecureserver.com.

@ IN A 127.0.0.1

Step 30: Editing named.conf.local file:

zone "maimoontabib.com" IN {

type master;

file "/etc/bind/maimoontabib.com.zone";

};

Step 31: Enabling and restarting named:

systemctl enable named

systemctl start named

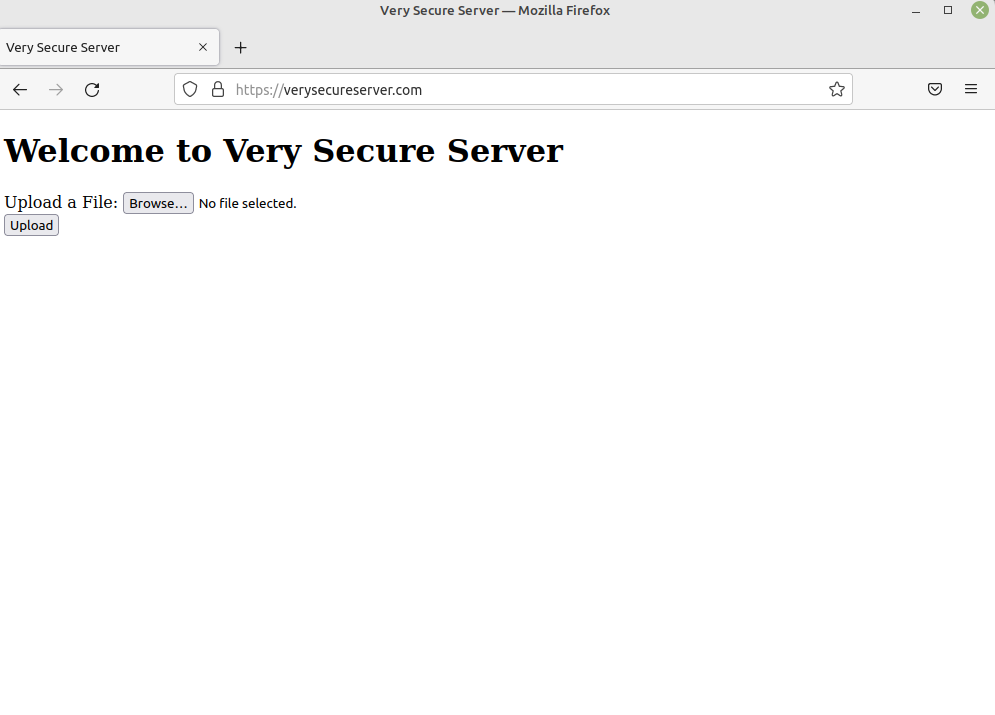
systemctl restart named

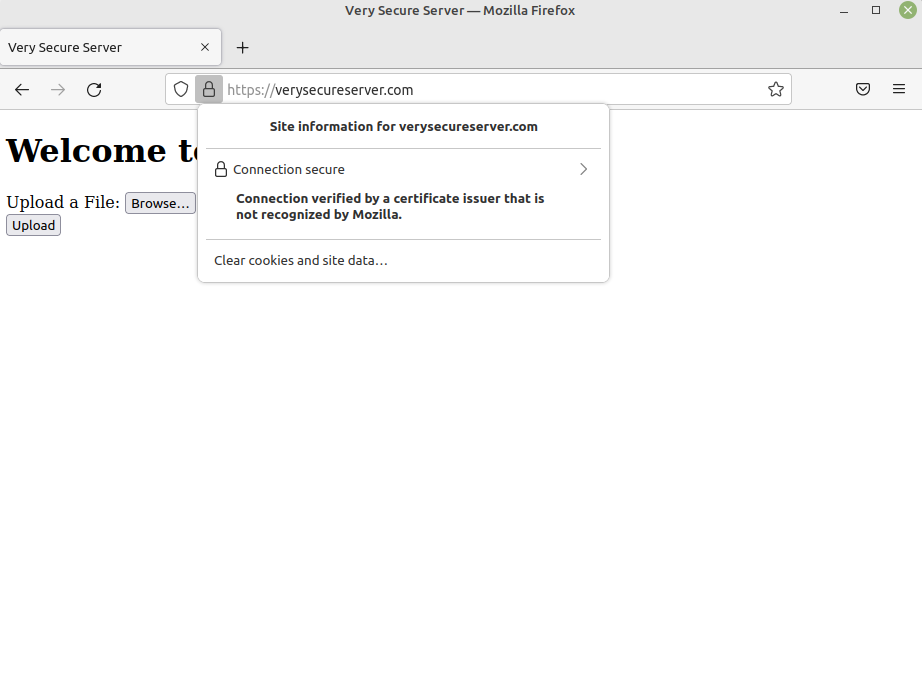
Step 32: Checking the server:

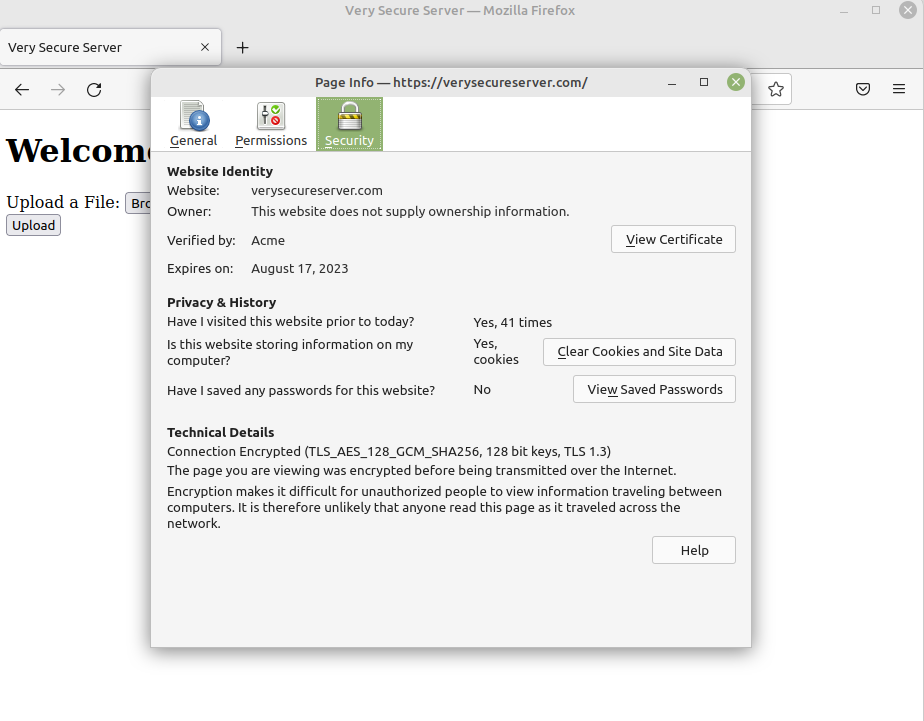
dig verysecureserver.com

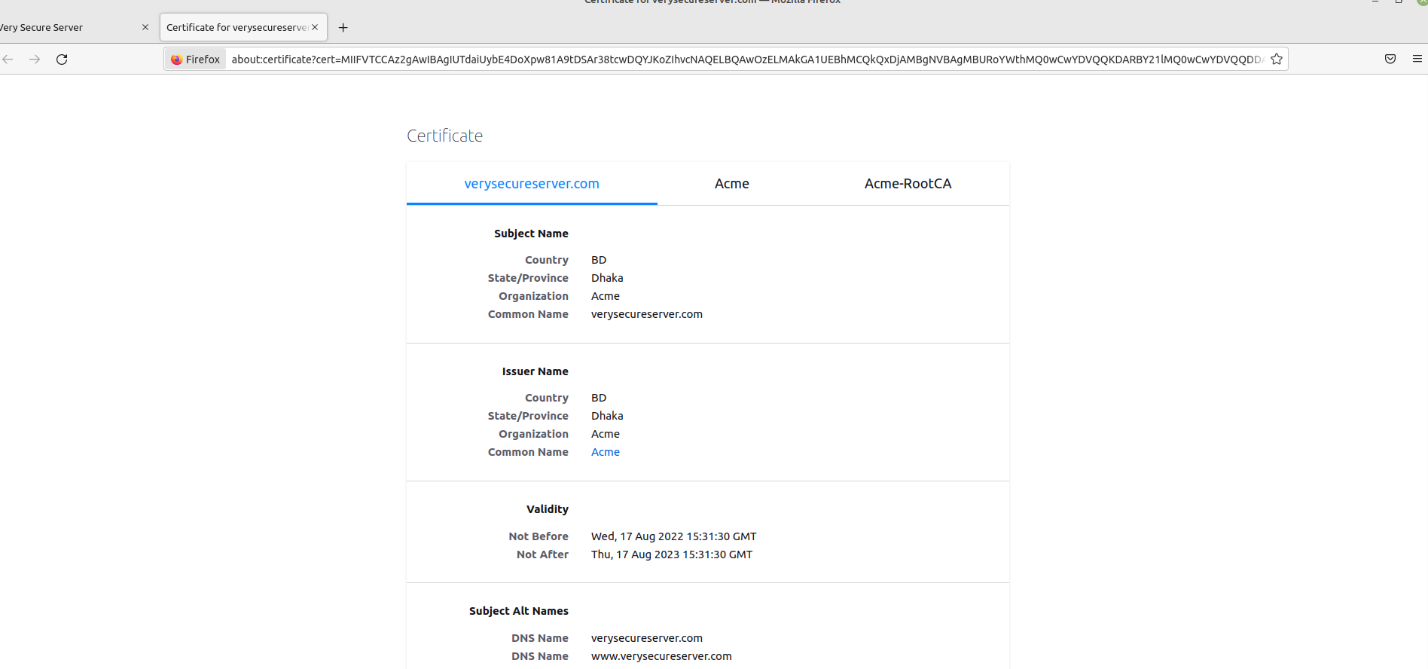
nslookup verysecureserver.com

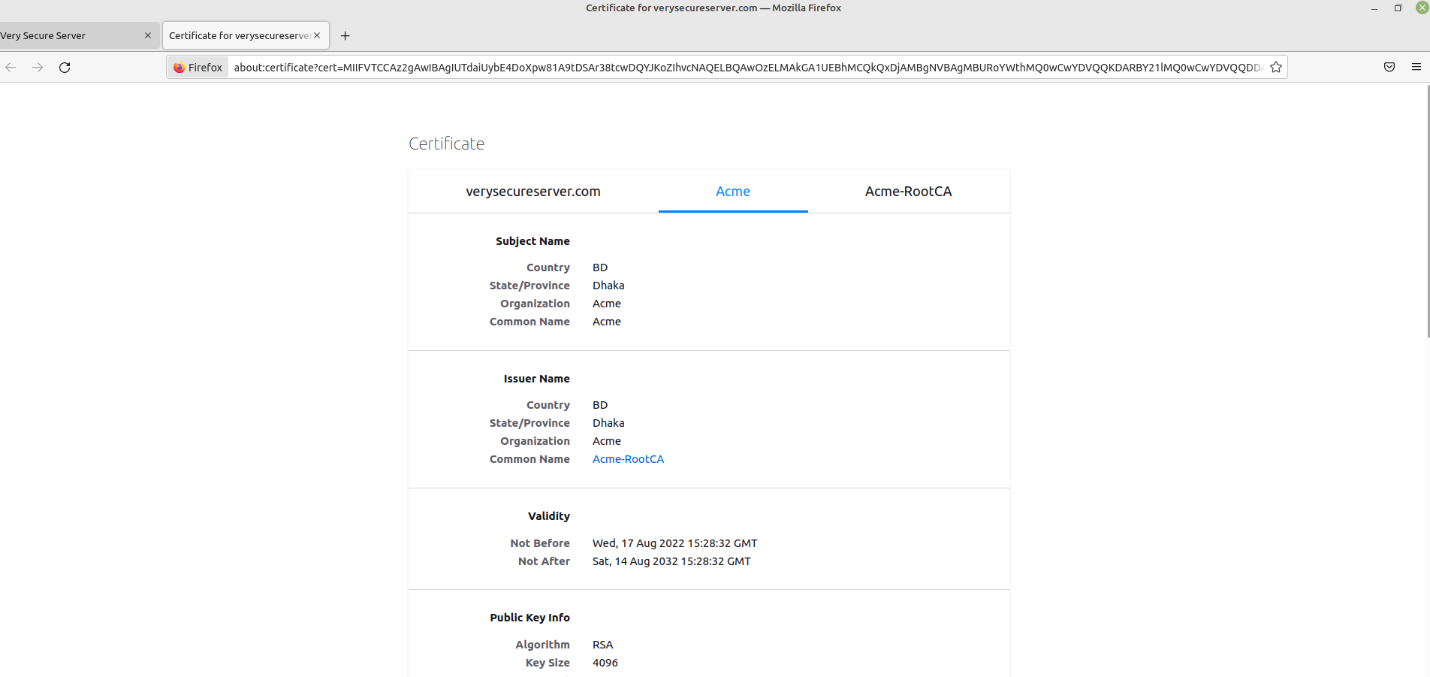
**Screenshots:**

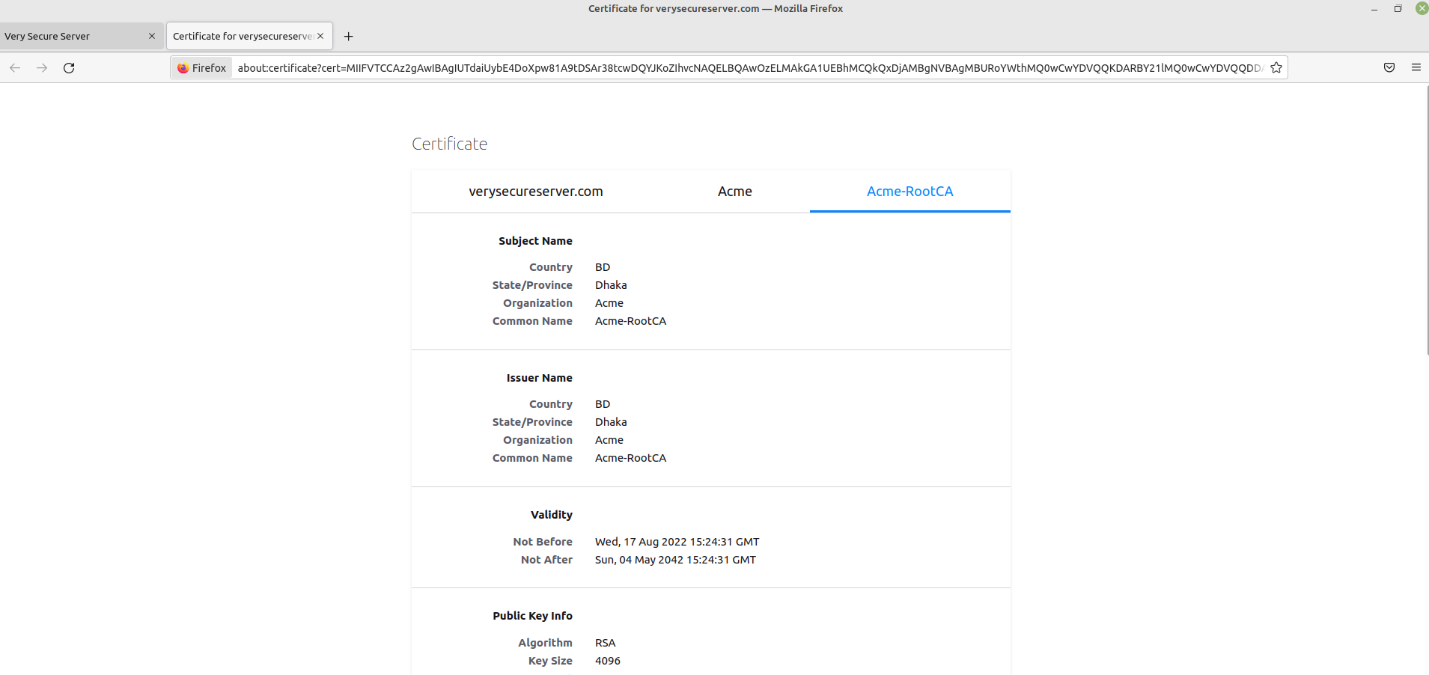












**3. Revoking Certificate**

Step 33: Moving to sub-ca folder:

**cd sub-ca**

Step 34: Revoking server.crt:

**openssl ca -config sub-ca.conf -revoke ../server/certs/server.crt**

Step 34: Adding CRL to server:

**cd sub-ca**

**nano crlnumber**

Step 35: Implementing a Certificate Revocation List:

**openssl ca -config sub-ca.conf -gencrl -out crl/rev.crl**

**Codes:**

**root-ca.conf code**

[ca]

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

#see man ca

default\_ca = CA\_default

[CA\_default]

dir = Path directory

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 = 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 = maimoontabib.com

DNS.2 = www.maimoontabib.com

[crl\_dist\_points]

URI.0 = http://localhost:8086/rev.crl

**server-ca.conf code**

[ca]

#C:/openssl/root-ca/root-ca.conf

#see man ca

default\_ca = CA\_default

[CA\_default]

dir = Path directory

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 = 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 = maimoontabib.com

DNS.2 = www.maimoontabib.com

[crl\_dist\_points]

URI.0 = http://localhost:8086/rev.crl

**server-csr.conf code**

[req]

distinguished\_name = req\_distinguished\_name

req\_extensions = v3\_req

prompt = no

[req\_distinguished\_name]

C = BD

ST = Dhaka

L = Mymensingh

O = Acme

OU =

CN = maimoontabib.com

[v3\_req]

keyUsage = nonRepudiation, digitalSignature, keyEncipherment

extendedKeyUsage = serverAuth

subjectAltName = @alt\_names

[alt\_names]

DNS.1 = maimoontabib.com

DNS.2 = www.maimoontabib.com

**sub-ca.conf code**

[ca]

#C:/openssl/root-ca/root-ca.conf

#see man ca

default\_ca = CA\_default

[CA\_default]

dir = Path directory

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

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 = 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 = maimoontabib.com

DNS.2 = www.maimoontabib.com

[crl\_dist\_points]

URI.0 = http://localhost:8086/rev.crl