EAST WEST UNIVERSITY

Course title: Cyber security, law and Ethics

Course Code: CSE-487

Fall 2022

Section: 03

Project on:

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

Submitted By:

Name	ID
Sharmin Akter	2019-2-60-055
Md. Abu Rayhan	2019-2-60-061
Md. Helal Khan	2019-2-60-063

Submitted To:

Rashedul Amin Tuhin

Senior Lecturer

Department of Computer Science and Engineering

East West University.

Date of Submission:

15 December 2022

DNS server configuration

Update Ubuntu

- sudo apt-get update

Install bind9

- sudo apt install bind9

  To continue write y and press enter

To go bind folder

- cd /etc/bind

To see all the file, write

- ls

To configure IP address

- sudo nano/etc/hosts

We have to insert ip address and domain name in this file. Then ctrl+s for save and ctrl+x for executing.

```
  GNU nano 6.2                          /etc/hosts
127.0.0.1       localhost
127.0.1.1       ubuntu-LTS.secureserver.com ubuntu-LTS
192.168.56.102  ubuntu-LTS.secureserver.com ubuntu-LTS

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

To verify file content

- cat /etc/hosts

To check host name

- hostname

To see dns domain name

- Dnsdomainname

We have to config the“named.conf.options” file

- sudo nano named.conf.options

We have to insert ip address for dns and forward ip in this file. Then ctrl+s for save and ctrl+x for executing

```
    dnssec-validation auto;

    listen-on-v6 { any; };
    recursion yes;
    listen-on {192.168.56.102;};
    allow-transfer {none;};

    forwarders {
    192.168.56.105;
    };
```

We have to config the “named.conf.local” file

- sudo nano named.conf.local

We have to insert forward lookup zone and reverse lookup zone in this file. Then ctrl+s for save and ctrl+x for executing

```
// Consider adding the 1918 zones here, if
// organization
//include "/etc/bind/zones.rfc1918";
//forward lookup zone
zone "secureserver.com" IN {
       type master;
       file "/etc/bind/db.forward.com";
};

//reverse lookup zone
zone "56.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/db.reverse.com";
};
```

To check the configurations

- Named-checkconf

If this command return nothing it means our configuration is okay.

Next we have to configure the “db.forward.com” file

- sudo nano db.forward.com

We have to insert domain name and ip address in this file. Then ctrl+s for save and ctrl+x for executing.

```
$TTL    604800
@       IN      SOA     ns1.secureserver.com. root.secureserver.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.secureserver.com.
ns1     IN      A       192.168.56.102
www     IN      A       192.168.56.102
@       IN      AAAA    ::1
```

Next, we have to configure the “db.reverse.com” file

- sudo nano db.reverse.com

We have to insert domain name and ip address in this file. Then ctrl+s for save and ctrl+x for executing.

```
$TTL    604800
@       IN      SOA     ns1.secureserver.com. root.secureserver.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.secureserver.com.
102     IN      PTR     ns1.secureserver.com.
102     IN      PTR     www.secureserver.com.
```

To permanent the edit name server and ‘‘resolv.conf’’ file, we have to remove the ‘‘resolv.conf’’ file

- sudo rm /etc/resolv.conf

Then we will link a resolv.conf which is under systemd and result folder

- sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf

Now we have to edit "resolv.conf" file under etc

- sudo nano /etc/resolv.conf

If nameserver is not here then we have to insert domain name and ip address in this file. Then ctrl+s for save and ctrl+x for executing.

```
# See man:systemd-resolved.ser
# operation for /etc/resolv.co

nameserver 192.168.56.102
search secureserver.com
```

Start bind

- sudo nano service bind9 restart

To check resolve domain

- nslookup www.secureserver.com

We get result like this

```
Server:         192.168.56.102
Address:        192.168.56.102#53

Name:   www.secureserver.com
Address: 192.168.56.102
```

For generate certificates

Moving to the root using

- sudo -i

See tree of files inside the root

- tree

Creating directory

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

Changing the root of ca and sub ca private folder

- chmod -v 700 ca/{root-ca,sub-ca,server}/private

Creating file index in both root ca and sub ca

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

Generating hexadecimal random number of 16 character

writing serial number of root ca

- openssl rand -hex 16 > ca/root-ca/serial

writing serial number of sub ca

- openssl rand -hex 16 > ca/sub-ca/serial

moving to ca directory

- cd ca

1. Generating private key for root ca, sub ca and server
    - 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

Reviewing the change

- tree

Creating root ca.config

- sudo nano root-ca/root-ca.conf

Inserting the code of root ca in this file. Press ctrl+s for save and ctrl+x for exit. Code of root ca is given in the end.

Moving inside root-ca

- cd root-ca

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

Ensuring that the certificate has been created properly

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

Moving a step back and then to sub-ca

- cd ../sub-ca

Creating sub-ca.config

- sudo nano sub-ca.conf

Inserting the code of sub ca in this file. Press ctrl+s for save and ctrl+x for exit. Code of root ca is given in the end.

Requesting for sub ca certificate signing request

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

Moving to the previous folder

- cd –

Signing the request of sub ca by 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
- to confirm insert y

See directory

- Tree
- →.pem file has been generated

See the list of signing

- cat index
- →Root ca signed sub ca

Seeing detail

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

Configuring server moving to server

- cd ../server
- sudo nano server.conf

Generating certificate signing request from server

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

Sub ca signing certificate request of server

- openssl ca -config sub-ca.conf -extensions server_cert -days 3650 - notext -in ../server/csr/server.csr -extensions req_ext -extfile ../server/server.conf -out ../server/certs/server.crt

Seeing detail

- cat index
- cd ../server/certs

Now, concating sub-ca.crt and server.crt and naming the new file chained.crt

- cat server.crt ../../sub-ca/certs/sub-ca.crt > chained.crt
- cat server.crt ../../root/certs/ca.crt > chain.crt
- openssl x509 -noout -text -in server.crt

Apache web server configuration

- Sudo apt-get install apache2
- To continue enter y

Moving to apache default configuration file

- cd /etc/apache2/sites-available

Configure the ‘‘default-ssl.conf’’ file

- sudo nano default-ssl.conf

We have to insert the location of certificate in this file. Then ctrl+s for save and ctrl+x for executing.

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost
                DocumentRoot /var/www/html
                servername secureserver.com
                ServerAlias www.secureserver.com


                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                #   server certificate
                SSLCertificateFile      /home/helal/ca/server/certs/server.crt
                SSLCertificateKeyFile /home/helal/ca/server/private/server.key
                #   Server Certificate Chain:
                SSLCertificateChainFile /home/helal/ca/server/certs/chain.crt
```

Ubuntu Software

Configure the ‘‘000-default.conf’’ file

- sudo nano 000-default.conf

We have to insert domain name in this file. Then ctrl+s for save and ctrl+x for executing.

```
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        ServerName secureserver.com
        serverAlias www.secureserver.com

        # Available loglevels: trace8, ..., trace1, debug, info, noti
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particula
        # modules, e.g.
        #LogLevel info ssl:warn

        Redirect permanent "/" "https://secureserver.com/"

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

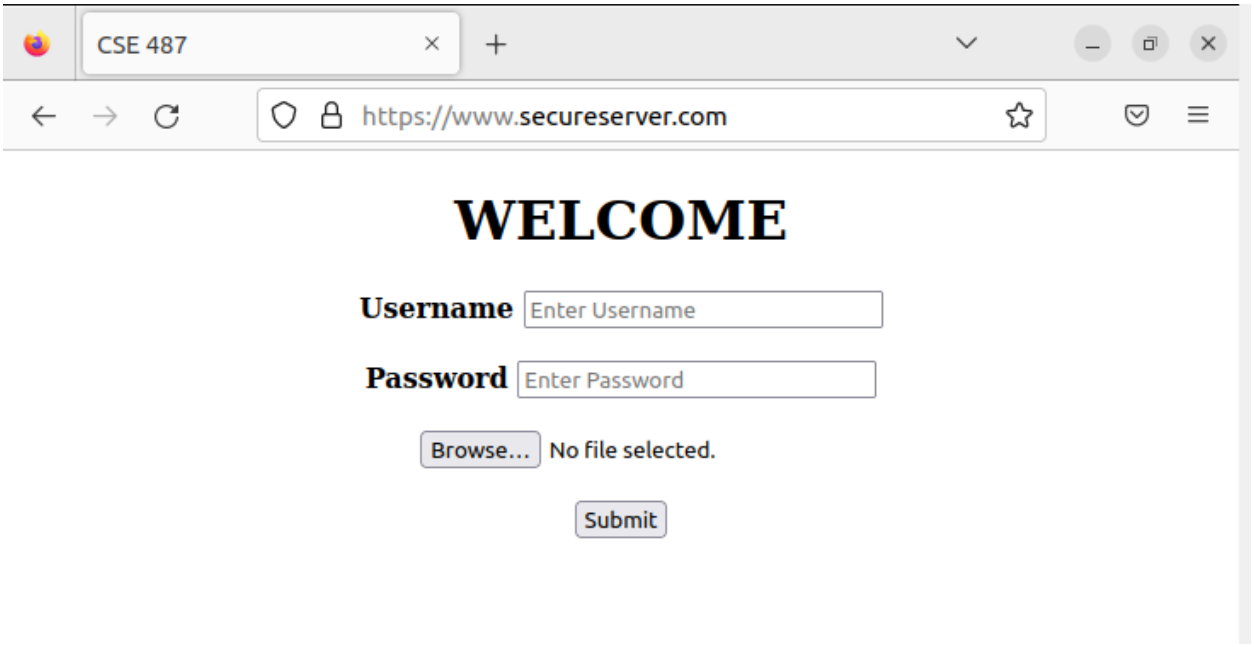
To edit index.html

- cd /var/www/html
- sudo nano index.html

To start the apache

- sudo systemctl restart apache2

Now on the browser Settings → privacy and security → view certificate → authorities → import → select the file → {ca.cert and sub-ca.crt}→open → select purpose → {view: to see the certificate} → OK



Our website is secure now.

Codes of ROOT CA & SUB-CA ROOT_CA:

```
[ca]
 #/root/ca/root-ca/root-ca.conf
#see man ca
default_ca = CA_default
[CA_default] dir = /home/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 ]

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 = NNS
commonName_default = RootCA

[ v3_ca ]
# Extensions to apply when creating 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

[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

SUB_CA:

[ca]
#/home/ca/sub-ca
#see man ca default_ca = CA_default

[CA_default] dir = /home/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
0.organizationName = Organization Name
organizationalUnitName = Organizational Unit Name
commonName = Common Name
emailAddress = Email Address
countryName_default = BD s
tateOrProvinceName_default = Dhaka
0.organizationName_default = NNS
commonName_default = SubCA

[ v3_ca ]
# Extensions to apply when createing root ca
# Extensions for a typical CA, man x509v3_config

subjectKeyIdentifier = hash authority
KeyIdentifier = 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

[ server_cert ]

# Extensions for server certificates
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
```

subjectKeyIdentifier = hash authority
KeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth

Server

[req]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
prompt = no

[req_distinguished_name]
countryName = BD
stateOrProvinceName = Dhaka
organizationName = NNS
commonName = www.secureserver.com

[req_ext]

subjectAltName = @alt_names
[alt_names]
DNS.1 = forward.com
DNS.2 = www.secureserver.com