SSL/TLS

An Implementation using OpenSSL

Group Members:

Abhay Mishra - 202111001
Charu Chandra Joshi - 202111019
Falgun Soni - 202111028
Prakhar Shukla - 202111067
Sahil Sonkar - 202111075

Objective:

The aim of this implementation is to be able to estable a secure communication between the client and the server. We will use 3 Laptops, one laptop will act as the Server (Apache), 1 will act as a client and the other will be the certificate authority.

In the first step we focus on installing the server. Then we establish the http connection between the server and the client.

Both server and CA will generate RSA key pairs. Server will generate a CSR (signing request).

We then configure the certificate authority which will sign the certificate signing request. This is followed by the server installing the certificate.

At last, this will establish secure https communication.

Setup

The three actors involved have the following IP addresses:

1. **Client IP**: 192.168.158.37
2. **Server IP**: 192.168.158.229
3. **CA IP**: 192.168.158.145

Establishing network connection:

First of all, we need to make sure that all the devices are connected to the same network. We can ensure this by pinging devices.

Client pings Server

```
C:\Users\Charu Chandra Joshi>ping 192.168.158.229

Pinging 192.168.158.229 with 32 bytes of data:
Reply from 192.168.158.229: bytes=32 time=34ms TTL=128
Reply from 192.168.158.229: bytes=32 time=55ms TTL=128
Reply from 192.168.158.229: bytes=32 time=299ms TTL=128
Reply from 192.168.158.229: bytes=32 time=187ms TTL=128

Ping statistics for 192.168.158.229:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 34ms, Maximum = 299ms, Average = 143ms
```

CA pings Client

```
PS C:\Users\abhay> ping 192.168.158.37

Pinging 192.168.158.37 with 32 bytes of data:
Reply from 192.168.158.37: bytes=32 time=39ms TTL=128
Reply from 192.168.158.37: bytes=32 time=25ms TTL=128
Reply from 192.168.158.37: bytes=32 time=30ms TTL=128
Reply from 192.168.158.37: bytes=32 time=7ms TTL=128

Ping statistics for 192.168.158.37:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 39ms, Average = 25ms
```

Server pings CA

```
C:\Users\Lenovo>ping 192.168.158.145

Pinging 192.168.158.145 with 32 bytes of data:
Reply from 192.168.158.145: bytes=32 time=79ms TTL=128
Reply from 192.168.158.145: bytes=32 time=78ms TTL=128
Reply from 192.168.158.145: bytes=32 time=91ms TTL=128
Reply from 192.168.158.145: bytes=32 time=9ms TTL=128

Ping statistics for 192.168.158.145:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 91ms, Average = 64ms
```

Setting up Apache HTTP Server (On Windows)

Install using the following link: https://www.apachelounge.com/download/

Apache 2.4 binaries VS17

Info & Changelog

Apache 2.4.59-240404 Win64

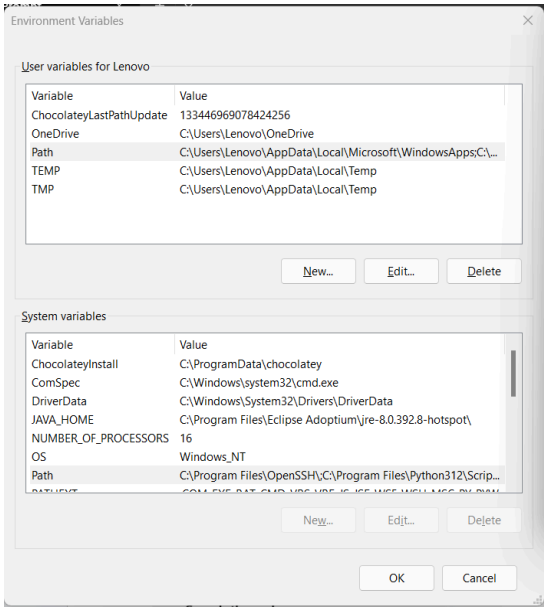
httpd-2.4.59-240404-win64-VS17.zip 04 Apr '24 11.431k
PGP Signature (Public PGP key), SHA1-SHA512 Checksums

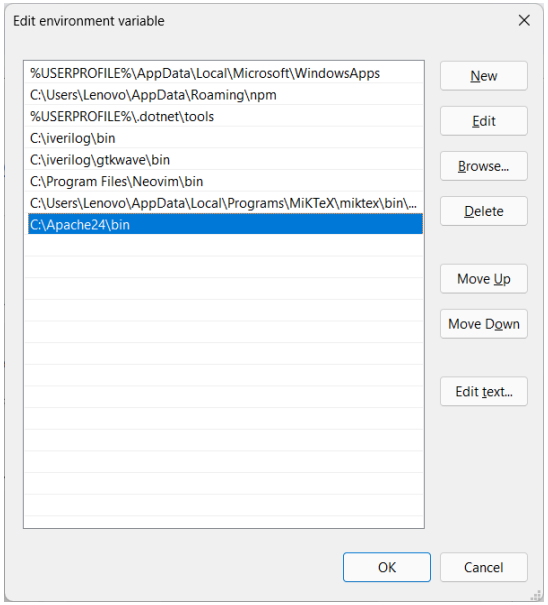
Apache 2.4.59-240404 Win32

httpd-2.4.59-240404-win32-vs17.zip 04 Apr '24 10.264k
PGP Signature (Public PGP key), SHA1-SHA512 Checksums

To be sure that a download is intact and has not been tampered with, use PGP, see **PGP Signature**

Extract the zip file and set up a User Path in Environment Variables. **(C:/Apache24/bin)**





The server has been set up. You can run the command “httpd” on the command prompt to start the server.

```
C:\Users\Lenovo>httpd
```

The client has successfully connected to the server using http **(not secure)**.

Now that client can connect to the server, we need to make this connection secure.

Setting up OpenSSL

For windows, OpenSSL comes bundled up with the Apache Web Server binaries. By setting up the environment variable as done above, we have successfully installed OpenSSL.

Generating Certificate Signing Request (CSR)

On the server, we will generate the RSA private key and then the signing request. This can be achieved using the following command:

```
openssl req -new -newkey rsa:2048 -nodes -keyout mykey.pem -out myreq.pem
```

This command first generates an RSA private key pair of 2048 bits. The private key is saved in the mykey.pem file. It also requests a new CSR which is stored in myreq.pem file.

```
PS D:\SUNNY\Study\College\Sem6\CS304 Cryptography and Network Security\Cert> openssl req -new -newkey rsa:2048 -nodes -keyout mykey.pem -out myreq.pem
....+...+..+.........+......+..+......+......+...+.+.....+.......+..+.+................+....+...+++++++++++++++++++++++++++++++++++++++*...+......+......+
...++++++++++++++++++++++++++++++++++++++++*.+.+..+.+...........+....+...........+......+....+...+..+.+.....+.........+......+.......+........+...+....+...+.
...........+...........+........+.++++++
.....++++++++++++++++++++++++++++++++++++++++*.+.........+.+...+......+.........++++++++++++++++++++++++++++++++++++++++*.+.........+......+.+......+........+
.+.....+.++++++
-----
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]:IN
State or Province Name (full name) [Some-State]:DD
Locality Name (eg, city) []:Diu
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IIITVICD
Organizational Unit Name (eg, section) []:A
Common Name (e.g. server FQDN or YOUR name) []:CSE
Email Address []:falgunsoni.2022@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Falgun
An optional company name []:Falgun
PS D:\SUNNY\Study\College\Sem6\CS304 Cryptography and Network Security\Cert>
```

The myreq.pem is the CSR file and it will be sent to the CA.

Certificate Authority Setup

Create a minimal openssl CA configuration file and save it as *ca.conf*:

```
[ ca ]
default_ca = ca_default
[ ca_default ]
dir = ./ca
certs = $dir
new_certs_dir = $dir/ca.db.certs
database = $dir/ca.db.index
serial = $dir/ca.db.serial
RANDFILE = $dir/ca.db.rand
certificate = $dir/ca.crt
private_key = $dir/ca.key
default_days = 365
default_crl_days = 30
default_md = sha-256
preserve = no
policy = generic_policy
[ generic_policy ]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = optional
emailAddress = optionalSSS
```

Create the CA database directory and some other necessary directories and files (it will hold information about all the certificates you issue):

```
mkdir ca
cd ca
mkdir ca.db.certs
touch ca.db.index
echo "1234" > ca.db.serial
```

Generate a 2048-bit RSA private key for the CA:

```
openssl genrsa -des3 -out ca/ca.key 2048
```

```
C:\Users\abhay\OneDrive\Desktop\ssl>openssl genrsa -des3 -out ca/ca.key 2048
Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:
```

Create a self-signed X509 certificate for the CA (the CSR will be signed with it):

```
openssl req -new -x509 -days 10000 -key ca/ca.key -out ca/ca.crt
```

```
C:\Users\abhay\OneDrive\Desktop\ssl>openssl req -new -x509 -days 10000 -key ca/ca.key -out ca/ca.crt
Enter pass phrase for ca/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) [AU]:IN
State or Province Name (full name) [Some-State]:DIU
Locality Name (eg, city) []:Kevdi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IIITVICD
Organizational Unit Name (eg, section) []:CSE
Common Name (e.g. server FQDN or YOUR name) []:abhay
Email Address []:202111001@diu.iiitvadodara.ac.in

C:\Users\abhay\OneDrive\Desktop\ssl>
```

Sign CSR: (myreq.pem)

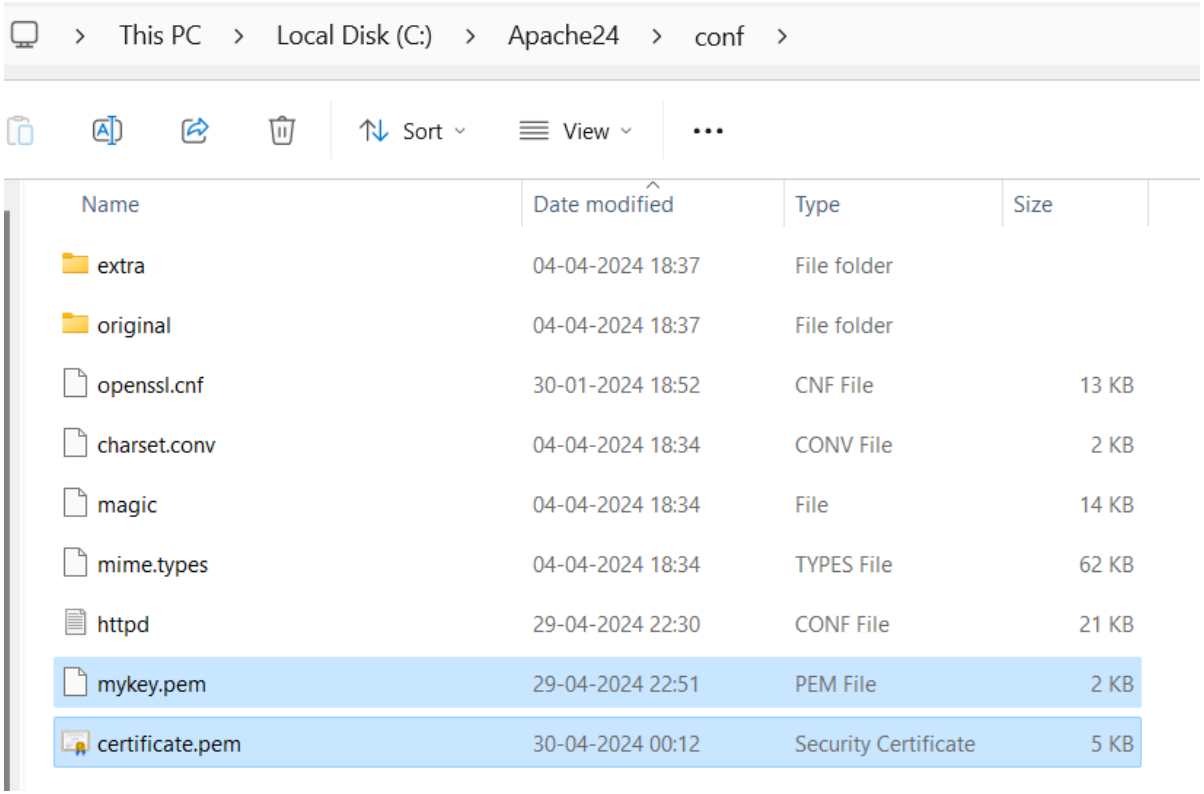
```
openssl ca -config ca.conf -out certificate.pem.crt -infiles myreq.pem
```

```
C:\Users\abhay\OneDrive\Desktop\ssl>openssl ca -config ca.conf -out certificate.pem.crt -infiles myreq.pem
Using configuration from ca.conf
Enter pass phrase for ./ca/ca.key:

Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IN'
stateOrProvinceName   :ASN.1 12:'DD'
localityName          :ASN.1 12:'Diu'
organizationName      :ASN.1 12:'IIITVICD'
organizationalUnitName:ASN.1 12:'A'
commonName            :ASN.1 12:'CSE'
emailAddress          :IA5STRING:'falgunsoni.2022@gmail.com'
Certificate is to be certified until Apr 29 18:40:05 2025 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
Database updated

C:\Users\abhay\OneDrive\Desktop\ssl>
```

Now the certificate.pem.crt file, i.e., the signed certificate file will be sent back to the server for installation.

Installing the Certificate on Server.

Move the private key file (mykey.pem) and the signed certificate file (certificate.pem.crt) to the conf folder inside Apache root (Apache24) folder.



Change the Apache24/conf/httpd.conf file:

Uncomment the following lines:

Line 527:

```
Include conf/extra/httpd-ssl.conf
```

Line 176:

```
LoadModule ssl_module modules/mod_ssl.so
```

Change the Apache24/conf/extra/httpd-ssl.conf file:

Change this line(144) from :

```
SSLCertificateFile "${SRVROOT}/conf/server.crt"
```

To:

```
SSLCertificateFile "${SRVROOT}/conf/certificate.pem.crt"
```

Change this line(154) from :

```
SSLCertificateKeyFile "${SRVROOT}/conf/server.pem"
```

To:

```
SSLCertificateKeyFile "${SRVROOT}/conf/mykey.pem"
```

With this the signed certificate has been successfully installed in the Apache HTTP Server. We can now access this server using https (secure).

HTTPS Connection