EAST WEST UNIVERSITY

Cybersecurity Project Report

Public-Key-Infrastructure-Implementation

Configuration of Certification Authority , Web server and DNS server. Certificate authority generates the certificate that we use on our website. It ensures that our website is secure. We host our websites on our web server and the dns server enables the dns name when we type example.com it takes us to example.com .

1. Preparing the environment
We create our certificate authority and servers on virtual machine. Using virtual machine on our computer, first install virtualbox and then install any linux operating system. Your computer ram is greater than 8 gb you can use any linux distribution but your ram 8 gb or less then 8gb i recommended use the "Xubuntu" it is light weight. You can find many tutorials on youtube how to install virtual box and virtual machine.

First of all you need to install OpenSSL software. Now open a terminal on your linux machine then type the command

sudo apt update

For updating your machine. After updating type the below command for install OpenSSL

sudo apt install openssl -y

See the folder structure on your linux machine type tree on the terminal:

tree

Now Creating directories for Root-ca , Sub-ca, server in these directories has some sub directory private, certs, newcerts, crl, csr. Private directory holds the private key , certs hold the certificate and the csr directory holds certificate signing requests. In the below command we create all the directories and sub directories at a time:

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

See if the folders are created successfully type the below command:

tree ca

ca your root directory all the directories and sub directories are under the ca directory.

Changing the permission root-ca and sub-ca directory private folder because the private folder holds the private key. We ensure other users can not show the private key.

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

Creating file index in both root ca and sub ca. This index file store backup our certificate.

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

Generating hexadecimal random numbers of 16 characters. This helps us to create a serial number that is used to generate certificates.

openssl rand -hex 16

writing serial number for root ca

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

writing serial number for sub ca

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

tree ca

Now we moving to our ca directory that we create

cd ca

2. After moving the ca directory we generated private keys for the root ca, sub ca and server. We use this private key for signing the certificate. When we generate private key it ask a pass phrase for secure the key when we use this key this pass phrase need.

Generate private key for rootCA using below command.

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

Generate private key for subCA using below command

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

Generate private key for server using below command

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

Now reviewing the change using tree command

tree

3. Now we Generate certificates for our root-ca. First we create a file ca.config in our root-ca directory. This is a configuration file in this file. When we create a certificate it takes some information about our certificate authority. We use this configuration file like a template for our certificate.

Now create root ca.config file using below command

gedit root-ca/root-ca.conf

gedit is a text editor in linux. You can install gedit using this command "**sudo apt install gedit**". In the above command create the root-ca,conf file and open it the gedit text editor. Now in this editor we pest the configure file template code. We can find this code in the link below.

----

https://github.com/maruf-shahriar/Public-Key-Infrastructure-Implementation/blob/main/Root-ca.config

Save and exit

Now we move our root-ca directory

cd root-ca

We create a certificate for our Root certificate authority. This is a self sign certificate because the Root certificate authority is the organisation. This organisation is the top authority; others can not verify this organisation so it signs itself. Like a country, the president is in top position so others can not verify the president. The President verifies by himself.

openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 1095 -sha256 -extensions v3_ca -out certs/ca.crt

Ensuring that the certificate has been created properly using the below command. It output the certificate with the text format.

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

Currently we are in root-ca directory now moving a step back and then go to sub-ca directory:

cd ../sub-ca

Sub-CA

Similarly in the roo-ca we create a configuration file for our sub-ca sub-ca.config.

gedit sub-ca.conf

Inserting the code from the link into the sub-ca.config

https://github.com/maruf-shahriar/Public-Key-Infrastructure-Implementation/blob/main/Sub-ca.config
Save and exit

Seeing the directory once again using tree command

tree

Our sub-ca is a sub organisation in our root ca. So we can request that our root CA sign our sub ca certificate. First we create a certificate signing request for the sub ca using sub-ca private key. This task is performed when we type this command on our terminal.

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

Now moving to our root-ca directory because now our root-ca sign our sub-ca certificate

cd -

In the use of below command root-ca sign our sub-ca

openssl ca -config root-ca.conf -extensions v3_intermediate_ca -days 1095 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt

to confirm insert “y”

See directory

tree

→Root ca signed sub ca
Seeing detail type openssl command below and type sub-ca certificate path:

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

4. Now we configure our server for the specific domain. First we go to our server folder.

cd ../server

Same to the sub ca firstly the server generates a certificate signing request using its private key. Then this certificate signifies the certificate authority.When generating the certificate signing request it asks for some information we provide this information and must provide the common name section with your server name like example.com .

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

Now we go to our sub-ca to sign our server certificate. We go sub-ca because our root CA permits the sub ca to sign any server certificate.

cd ../sub-ca

Now sub-ca signs the certificate using its configuration file.

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

Moving to server certs folder to see certificate of server

cd ../server/certs/

See the directory by using command:

ls

→ We can see that the server.crt file has been generated

Now, concating root-ca.crt ,sub-ca.crt and server.crt file and naming the new file chained.crt. We cocating these files because when we use this server certificate the web browser finds who signed this certificate.

cat /home/maruf/ca/server/certs/server.crt /home/maruf/ca/sub-ca/certs/sub-ca.crt /home/maruf/ca/root-ca/certs/ca.crt > chained.crt

You change the file path where you save your file . My files are save /home/maruf/ca. You replace this with your path.

Seeing the change

ls

Now copy all the certificates in a folder. We use this folder later when we configure our server. We use graphical interface for copying like when we copy and paste in our windows machine. Otherwise we also use terminal for copying using below commands:

```
cp /home/maruf/ca/root-ca/certs/ca.crt /home/maruf/certificate/
cp /home/maruf/ca/sub-ca/certs/sub-ca.crt /home/maruf/certificate/
cp /home/maruf/ca/server/certs/chained.crt /home/maruf/certificate/
cp /home/maruf/ca/server/certs/server.crt /home/maruf/certificate/
cp /home/maruf/ca/server/private/server.key /home/maruf/certificate/
```

Now we configure the web server to host our websites. We use xampp software for configuring the web server. So we download the xampp on the official website for the linux version. After downloading the software it store download folder. But by default it execution permission is denied so first we change the permission. Write the below command for changing the permission:

```
sudo chmod 777 xampp-linux-x64-8.0.30-0-installer.run
```

Now install this file using this command “ **sudo ./xampp-linux-x64-8.0.30-0-installer.run** ”. After installing xampp we go to the xampp folder using this command “ **cd /opt/lampp/ ” .** In this folder we type “ **sudo ./xampp-manager-linux.run** ” to run the software. After opening the software we run the apache server.

Next go to this location

```
sudo cd /opt/lampp/etc/extra
```

In this folder you find a file httpd-ssl.conf we change the file permission first.

```
chmod 777 httpd-ssl.conf
```

Now we edit the file

```
gedit httpd-ssl.conf
```

In this file search the virtualhost section and replace the virtualhost section with the below code.

```
<VirtualHost _default_:80>
ServerName www.verysecureserv.com:80
ServerAdmin you@example.com
Redirect permanent / https://www.verysecureserv.com
</VirtualHost>
```

```
<VirtualHost _default_:443>

#   General setup for the virtual host
DocumentRoot "/opt/lampp/htdocs/verysecureserver"
ServerName www.verysecureserv.com:443
ServerAdmin you@example.com
ErrorLog "/opt/lampp/logs/error_log"
TransferLog "/opt/lampp/logs/access_log"

#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   Server Certificate:

SSLCertificateFile "/home/maruf/certificate/Vserver.crt"

#   Server Private Key:

SSLCertificateKeyFile "/home/maruf/certificate/Vserver.key"

#   Server Certificate Chain:

SSLCertificateChainFile "/home/maruf/certificate/Vchained.crt"

#   Certificate Authority (CA):

SSLCACertificatePath "/home/maruf/certificate"

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "/opt/lampp/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-5]" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

CustomLog "/opt/lampp/logs/ssl_request_log" \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
```

This code is for hosting our websites. We use the code multiple times for our multiple websites only. We change the ServerName with our specific domain name and change the DocumentRoot file path in this section we give our website html code path. Also we change the SSLCertificateFile path where we store the server.crt file, SSLCertificateKeyFile in this section give the server.key file path, SSLCertificateChainFile in this section give the chained.crt file path those files we generate when we configure the server certificate.
save and exit.

Then restart the apache server.
And run the websites in your web browser.

Now on the browser we import the root-ca certificate because our browser doesn't know our root-ca. So when we visit our website it is not secure. Certificate import setting given below.

Settings → privacy and security → view certificate → authorities → import → select the file
→open → select purpose → {view: to see the certificate} → OK

DNS Configuration

Dns configuration we use bind9 software.
Install dns configuration software write the below code on your terminal:

```
sudo apt install bind9
```

Next go to this location for configure the dns server

```
cd /etc/bind/
ls
sudo gedit ./named.conf.options
```

Now add the bottom code on this file:

```
listen-on-v6 { any; };
        recursion yes;
        listen-on {192.168.0.149;};
        allow-transfer {none;};
        forwarders {
        192.168.0.1;
        };
```

Save the configuration and exit.

Now configure the named.conf.local file

sudo gedit ./named.conf.local

Add this text in the file

```
//forward lookup zone
zone "ewubdca.com" IN {
        type master;
        file "/etc/bind/db.ewubdca.com";
};

//reverse lookup zone
zone "0.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/db.0.168.192";
}
```

Save and exit.

Replace the **ewubdca.com** with your domain name and replace the **reverse ip 0.168.192** with your reverse ip.

Now copy the db.127 file in your domain name file and reverse ip name

```
sudo cp db.127 db.ewubdca.com
sudo cp db.127 db.0.168.
```

Now open one by one file

sudo gedit ./db.ewubdca.com

Write the below text on this file

```
;
; BIND data file for local loopback interface
;
$TTL  604800
@       IN       SOA    ns1.ewubdca.com. root.ewubdca.com. (
                              2              ; Serial
                         604800                   ; Refresh
                          86400                   ; Retry
```

```
                                2419200                 ; Expire
                                 604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.ewubdca.com.
ns1     IN      A       192.168.0.149
www     IN      A       192.168.0.149
@       IN      AAAA    ::1
```

Replace the **ewubdca.com** with your domain name

Save and exit.

sudo gedit ./db.0.168.192

Paste the below code on your file.

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.ewubdca.com. root.ewubdca.com. (
                                     1          ; Serial
                                 604800                 ; Refresh
                                  86400                 ; Retry
                                2419200                 ; Expire
                                 604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.ewubdca.com.

149     IN      PTR     www.ewubdca.com.
149     IN      PTR     apply.ewubdca.com.
```

Replace the **domain name** and the **ip** with your domain and ip address.

Now restart the dns server

sudo service bind9 restart

Check the dns server status

sudo service bind9 status

Now add the dns server ip address in your client pc dns field. Then browse the website with your client pc using your domain name.

Configure the linux built in firewall ufw using the below link
----
https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-18-04

Now install “snort” for Intrusion Detection System (IDS)

```
sudo apt update
sudo apt install snort
```

Modify the **snort.conf** file using the built in alert or you made the custom alert for any incident.

To run Snort as a service

Create a file named snort.service in the **/etc/systemd/system/ directory.**

In this file write the below code.

```
[Unit]
Description=Snort IDS

[Service]
Type=simple
ExecStart=/usr/sbin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i
<YOUR_INTERFACE>

[Install]
WantedBy=multi-user.target
```

After modifying the service file

```
sudo systemctl daemon-reload
```

Enable and Start the Snort Service

```
sudo systemctl enable snort
sudo systemctl start snort
```

Restart the Snort service
----

sudo systemctl restart snort

Check the Snort status
----
systemctl status snort

Monitor Snort Logs
----
sudo gedit /var/log/snort/alert
Syn flood attack command
----
sudo hping3 --flood -S -p 80 192.168.0.149