EAST WEST UNIVERSITY

Department of CSE

Project Report

Course Name: Cyber Security, Law and Ethics

Course Code: CSE487

Project topic: Securing a networked system with Public Key Infrastructure Implementing Transport Layer Security on HTTP for https:// connection

1.0 Environment Setup:

1.1- Download Oracle VM VirtualBox From this website

- https://www.virtualbox.org/
- Install it in your pc

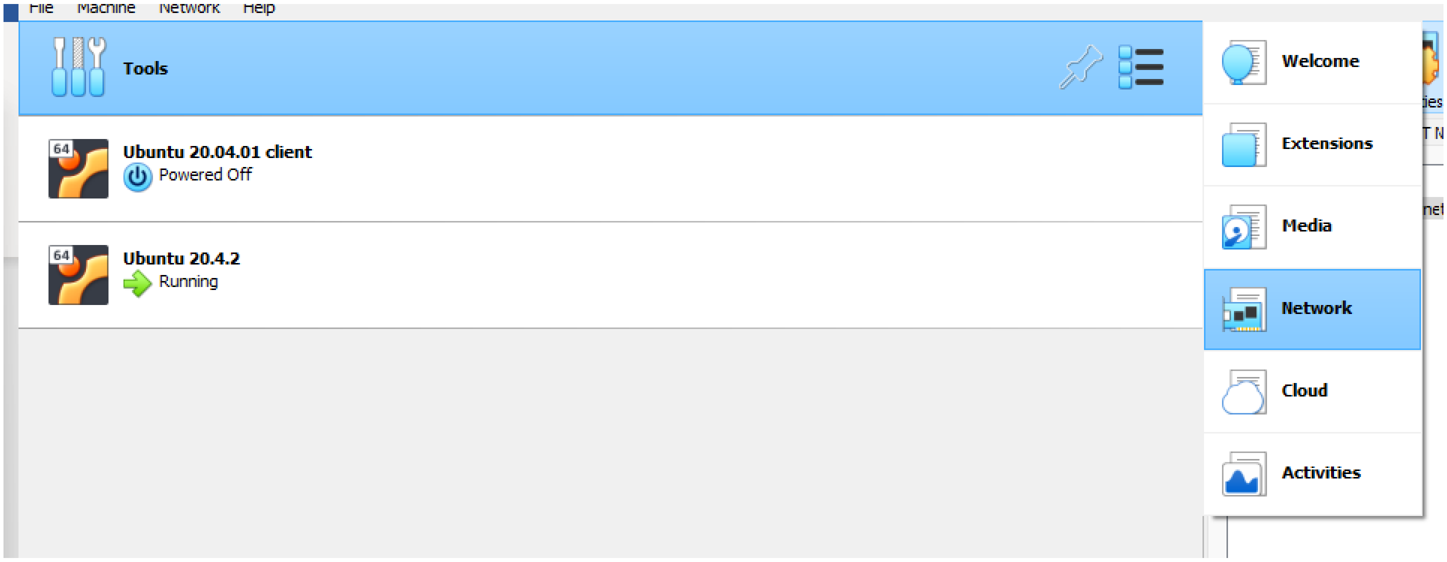
1.2- Download Ubuntu (Any Version) From this website

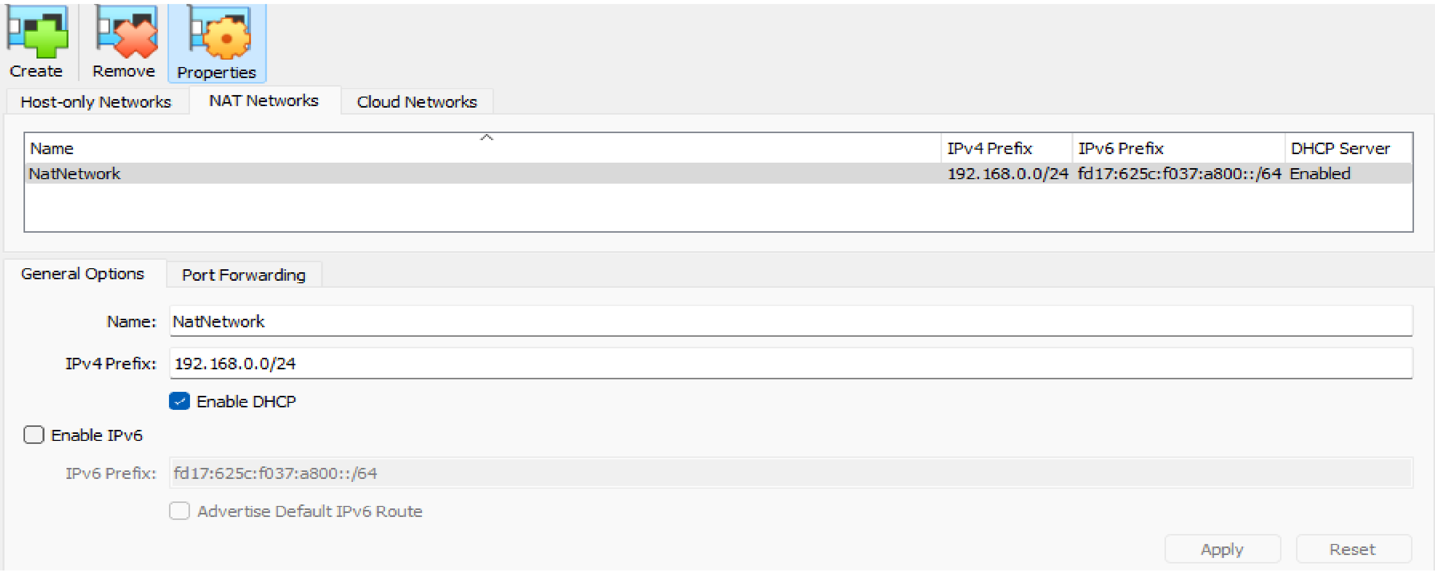
- https://ubuntu.com/download
- Install it in inside of the VirtualBox

For this project, we need two Ubuntu Operating System, First One we will configure that as a **Server** and the Second one, we will configure that as a **Client.**

2.0 Network Setup:

First, we go to the network option of the VirtualBox and configure the ip address of **NAT Network**.





Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
NatNetwork	192.168.0.0/24	fd17:625c:f037:a800::/64	Enabled

Name: NatNetwork

IPv4 Prefix: 192.168.0.0/24

☑ Enable DHCP

□ Enable IPv6

IPv6 Prefix: fd17:625c:f037:a800::/64

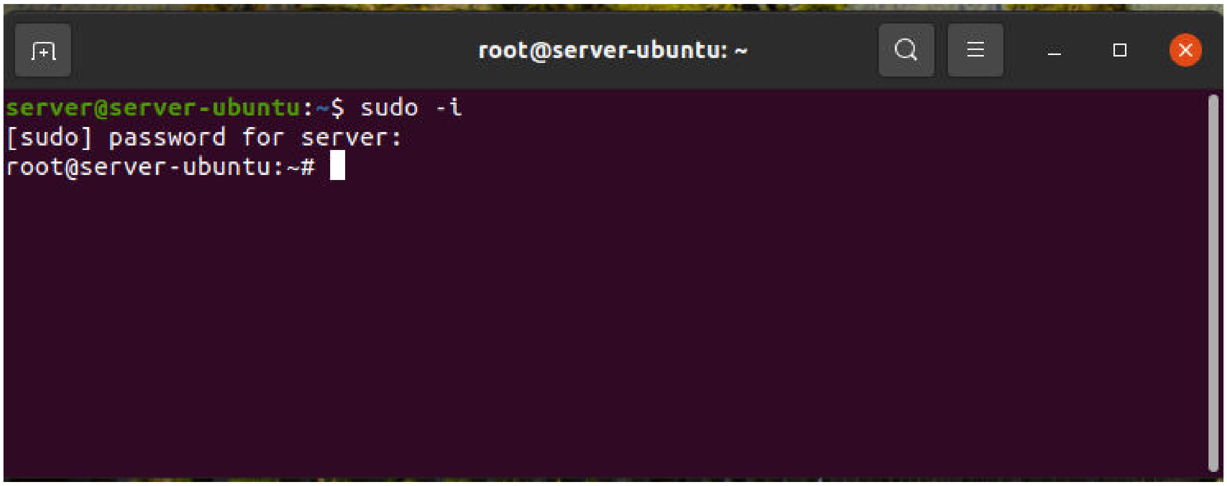
□ Advertise Default IPv6 Route

For this project, We will use 2 network adapter in both pc. In Adapter 1, set it as **NAT NETWORK** and In Adapter 2 set is as **NAT.** We will do the same on the both Server and Client PC.

3.0 Certificate Generation:

Step 01:

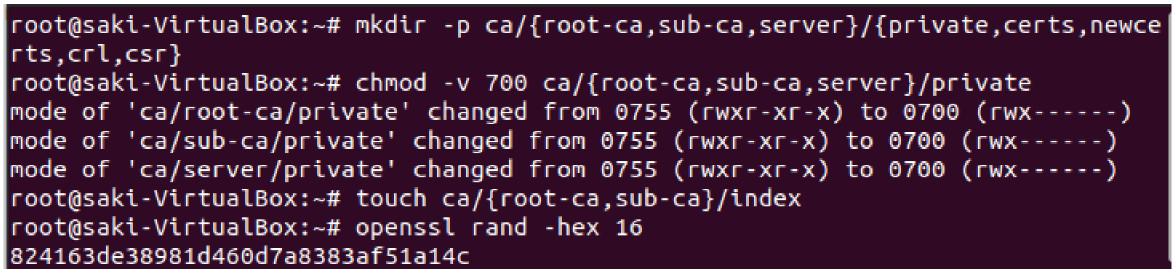
First of take root access in the terminal.



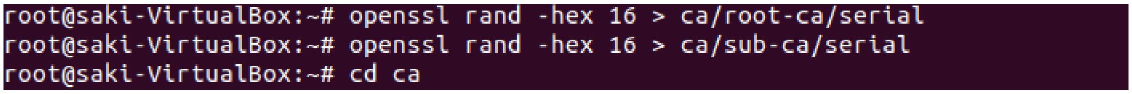
```
server@server-ubuntu:~$ sudo -i
[sudo] password for server:
root@server-ubuntu:~#
```

Now type the codes in the terminal and run them accordingly as picture:

Step 02:



```
root@saki-VirtualBox:~# mkdir -p ca/{root-ca,sub-ca,server}/{private,certs,newce
rts,crl,csr}
root@saki-VirtualBox:~# 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) to 0700 (rwx------)
mode of 'ca/server/private' changed from 0755 (rwxr-xr-x) to 0700 (rwx------)
root@saki-VirtualBox:~# touch ca/{root-ca,sub-ca}/index
root@saki-VirtualBox:~# openssl rand -hex 16
824163de38981d460d7a8383af51a14c
```



```
root@saki-VirtualBox:~# openssl rand -hex 16 > ca/root-ca/serial
root@saki-VirtualBox:~# openssl rand -hex 16 > ca/sub-ca/serial
root@saki-VirtualBox:~# cd ca
```

At first, we created three directories named rootCA, subCA, and server . Then we have created sub directories for the three previous directories that we created name private,certs,newcerts,crl,csr . This can be done using the command below :

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

Now we need to change the mode of the files, so that nobody can access those files except the user. This can be done by this command:

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

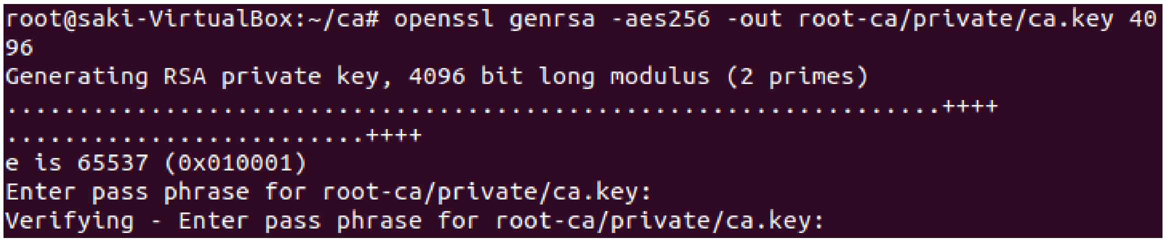
Then we need to create sub directories for RootCA and SubCA named “index”. This directory will be used to keep the index of the certificates . It is done by using this command below:

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

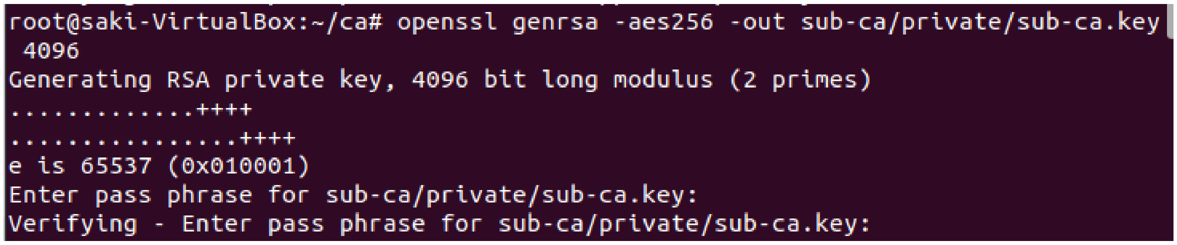
Then we generate for 16 bit random numbers for both rootCA and aubCA.

<u>Step 03:</u>

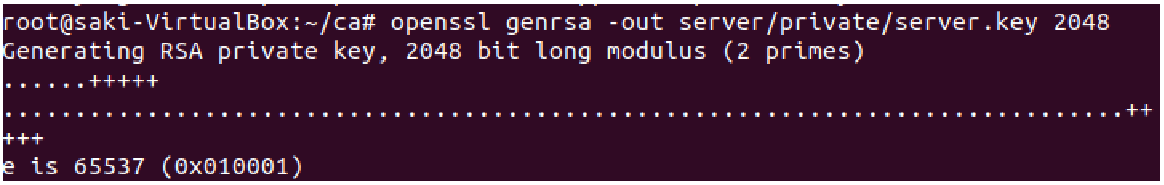
Now we will generate public key for RootC, SubCA and Server and this can be done by this command:



```
root@saki-VirtualBox:~/ca# openssl genrsa -aes256 -out root-ca/private/ca.key 40
96
Generating RSA private key, 4096 bit long modulus (2 primes)
.................................................................++++
.........................++++
e is 65537 (0x010001)
Enter pass phrase for root-ca/private/ca.key:
Verifying - Enter pass phrase for root-ca/private/ca.key:
```



```
root@saki-VirtualBox:~/ca# openssl genrsa -aes256 -out sub-ca/private/sub-ca.key
 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.............++++
................++++
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@saki-VirtualBox:~/ca# openssl genrsa -out server/private/server.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
......+++++
..............................................................................++
+++
e is 65537 (0x010001)
```

Why choose 4096 key lengths for RootCA and Sub CA ?

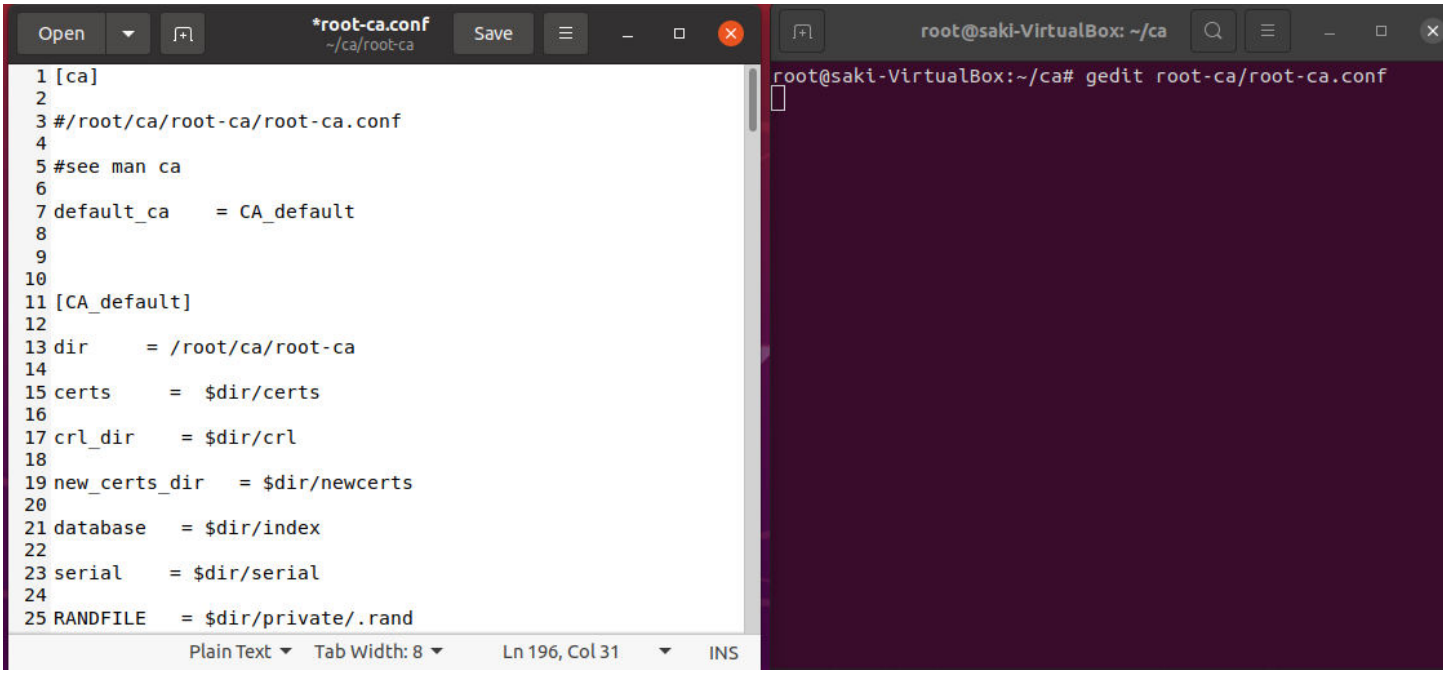
Private key of RootCA and SubCA has to be very strong. If the private key of RootCA or SubCA gets broken or leaked then all the certificated that were singed by these CA’s will be useless. That’s why we choose 2048 key length.

Why choose 2048 key length for server private key?

Privat key for server used very frequently. If the private key for server was too big like 4096 then the process of the server will slow.

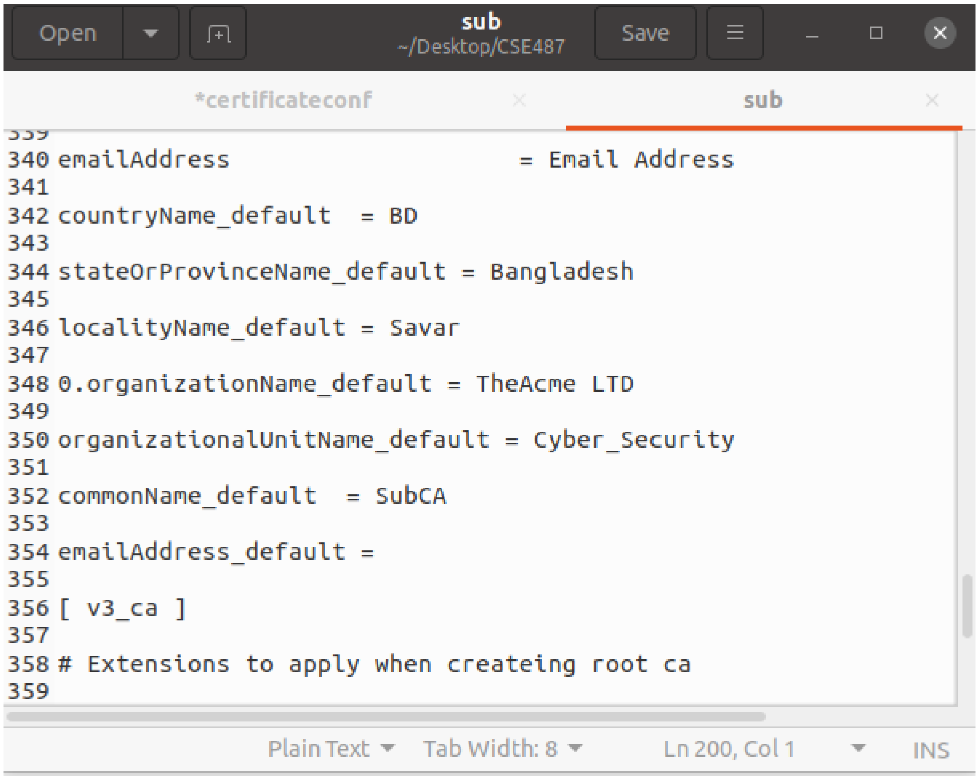
Step 04:

Now we will create the configuration file for both RootC Aans SubCA and edit the files.



```
[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
```

```
root@saki-VirtualBox:~/ca# gedit root-ca/root-ca.conf
```

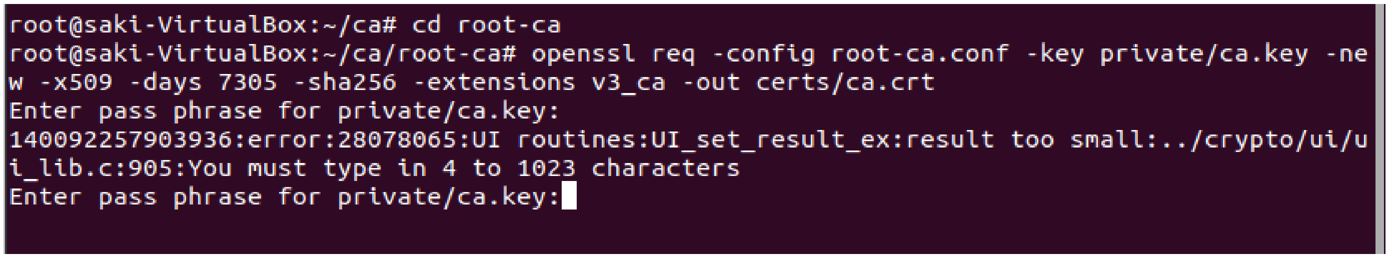


```
emailAddress                        = Email Address

countryName_default   = BD

stateOrProvinceName_default = Bangladesh

localityName_default = Savar

0.organizationName_default = TheAcme LTD

organizationalUnitName_default = Cyber_Security

commonName_default   = SubCA

emailAddress_default =

[ v3_ca ]

# Extensions to apply when createing root ca

```

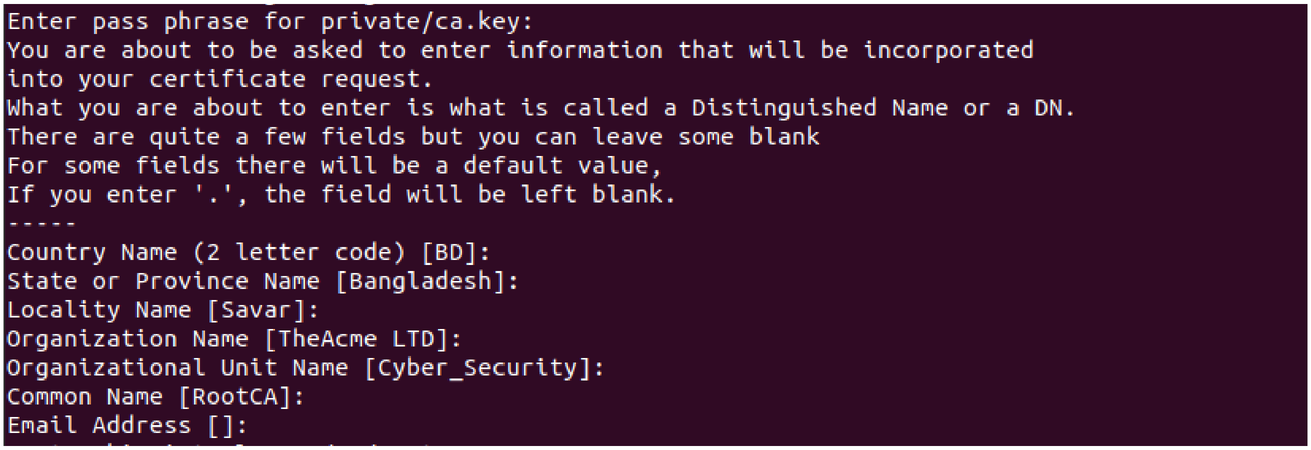
Full text given in **Appendix 1** & **Appendix 2**

Step 05:

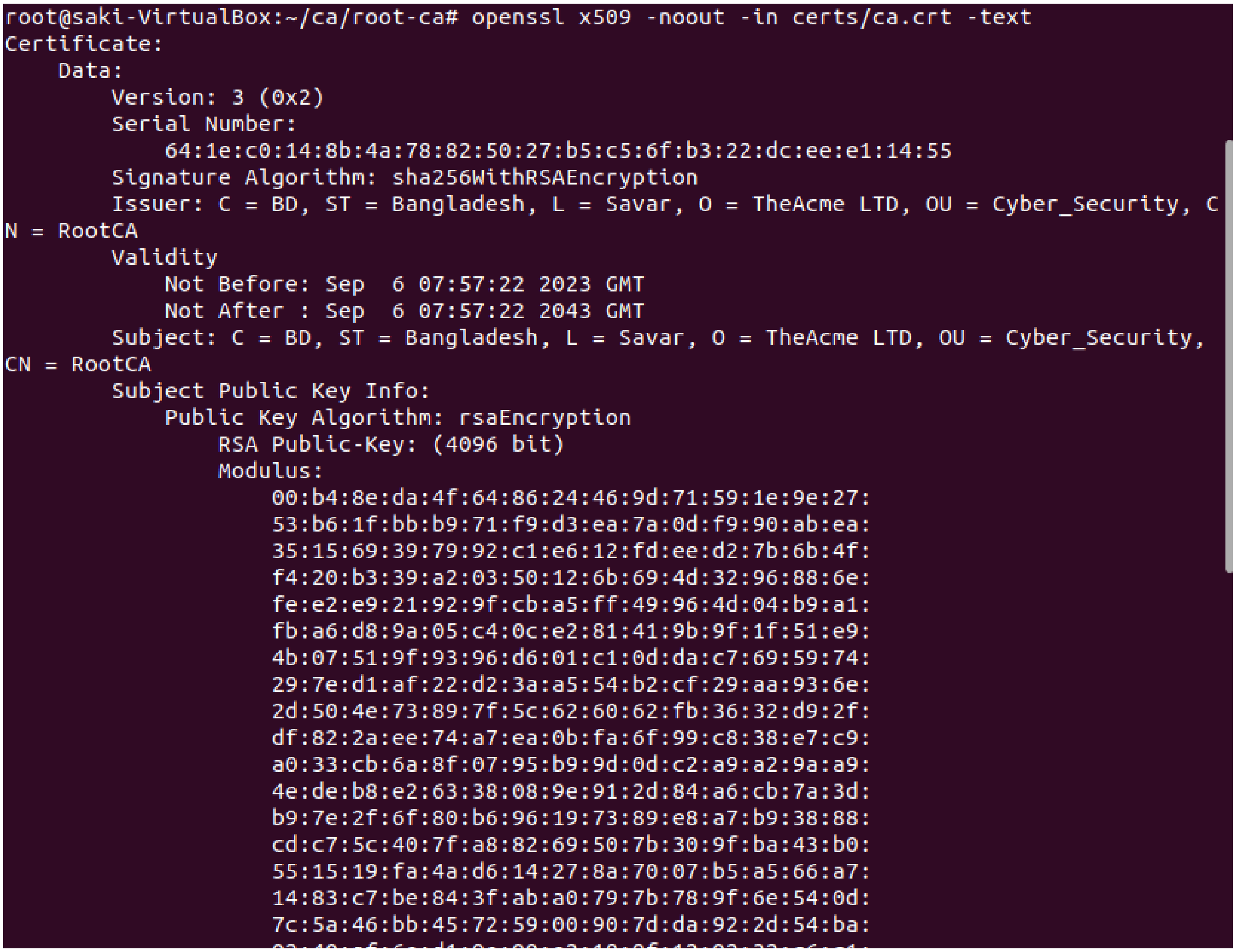
Now We will generate the RootCA Certificate. To do that, type **cd root-ca** and follow the rest given in the image below:



```
root@saki-VirtualBox:~/ca# cd root-ca
root@saki-VirtualBox:~/ca/root-ca# openssl req -config root-ca.conf -key private/ca.key -ne
w -x509 -days 7305 -sha256 -extensions v3_ca -out certs/ca.crt
Enter pass phrase for private/ca.key:
140092257903936:error:28078065:UI routines:UI_set_result_ex:result too small:../crypto/ui/u
i_lib.c:905:You must type in 4 to 1023 characters
Enter pass phrase for private/ca.key:
```

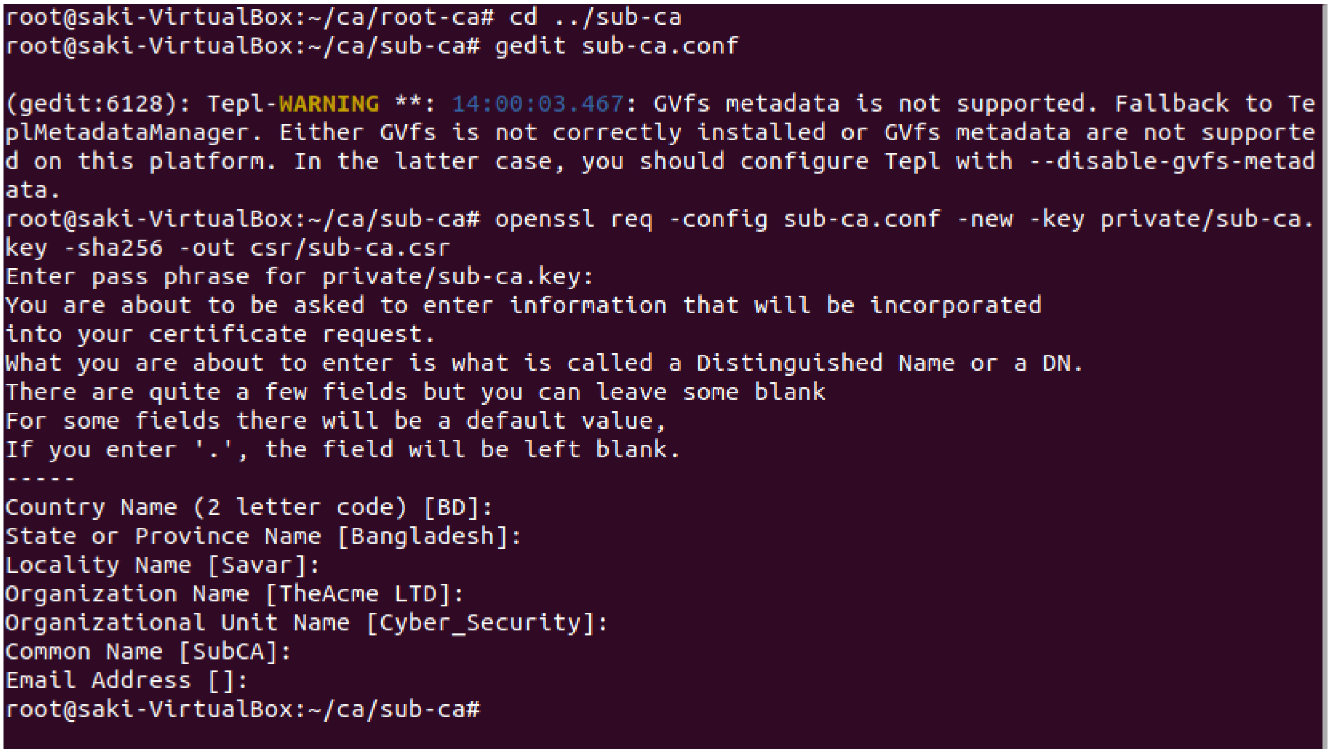


```
Enter pass phrase for private/ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [BD]:
State or Province Name [Bangladesh]:
Locality Name [Savar]:
Organization Name [TheAcme LTD]:
Organizational Unit Name [Cyber_Security]:
Common Name [RootCA]:
Email Address []:
```



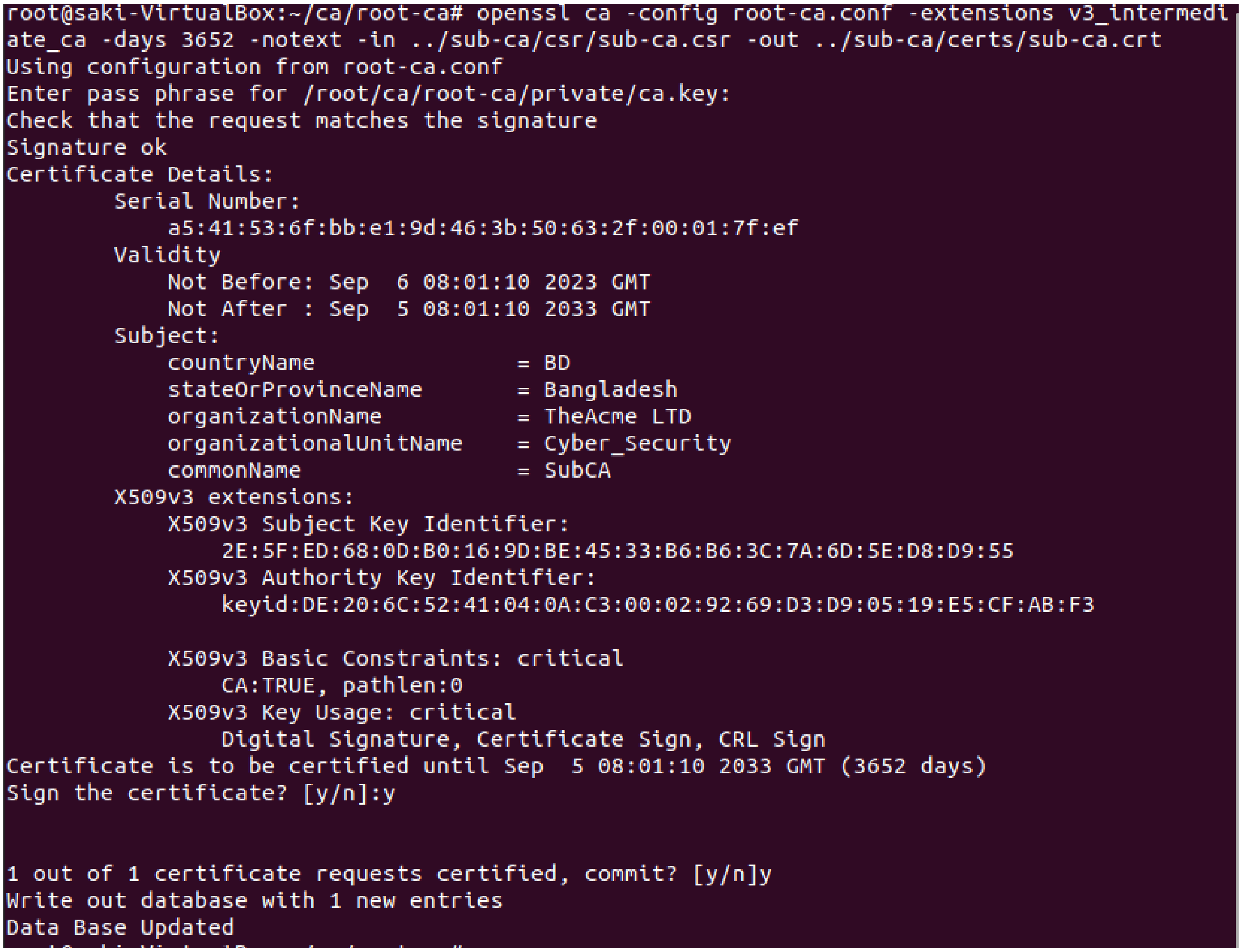
```
root@saki-VirtualBox:~/ca/root-ca# openssl x509 -noout -in certs/ca.crt -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            64:1e:c0:14:8b:4a:78:82:50:27:b5:c5:6f:b3:22:dc:ee:e1:14:55
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = BD, ST = Bangladesh, L = Savar, O = TheAcme LTD, OU = Cyber_Security, C
N = RootCA
        Validity
            Not Before: Sep  6 07:57:22 2023 GMT
            Not After : Sep  6 07:57:22 2043 GMT
        Subject: C = BD, ST = Bangladesh, L = Savar, O = TheAcme LTD, OU = Cyber_Security,
CN = RootCA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:b4:8e:da:4f:64:86:24:46:9d:71:59:1e:9e:27:
                    53:b6:1f:bb:b9:71:f9:d3:ea:7a:0d:f9:90:ab:ea:
                    35:15:69:39:79:92:c1:e6:12:fd:ee:d2:7b:6b:4f:
                    f4:20:b3:39:a2:03:50:12:6b:69:4d:32:96:88:6e:
                    fe:e2:e9:21:92:9f:cb:a5:ff:49:96:4d:04:b9:a1:
                    fb:a6:d8:9a:05:c4:0c:e2:81:41:9b:9f:1f:51:e9:
                    4b:07:51:9f:93:96:d6:01:c1:0d:da:c7:69:59:74:
                    29:7e:d1:af:22:d2:3a:a5:54:b2:cf:29:aa:93:6e:
                    2d:50:4e:73:89:7f:5c:62:60:62:fb:36:32:d9:2f:
                    df:82:2a:ee:74:a7:ea:0b:fa:6f:99:c8:38:e7:c9:
                    a0:33:cb:6a:8f:07:95:b9:9d:0d:c2:a9:a2:9a:a9:
                    4e:de:b8:e2:63:38:08:9e:91:2d:84:a6:cb:7a:3d:
                    b9:7e:2f:6f:80:b6:96:19:73:89:e8:a7:b9:38:88:
                    cd:c7:5c:40:7f:a8:82:69:50:7b:30:9f:ba:43:b0:
                    55:15:19:fa:4a:d6:14:27:8a:70:07:b5:a5:66:a7:
                    14:83:c7:be:84:3f:ab:a0:79:7b:78:9f:6e:54:0d:
                    7c:5a:46:bb:45:72:59:00:90:7d:da:92:2d:54:ba:
```

The same procedure we will follow to generate the SubCA certificate. Type **cd ../sub-ca** and follow the instruction given in the image below:



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

(gedit:6128): Tepl-WARNING **: 14:00:03.467: GVfs metadata is not supported. Fallback to Te
plMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supporte
d on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metad
ata.
root@saki-VirtualBox:~/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 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) [BD]:
State or Province Name [Bangladesh]:
Locality Name [Savar]:
Organization Name [TheAcme LTD]:
Organizational Unit Name [Cyber_Security]:
Common Name [SubCA]:
Email Address []:
root@saki-VirtualBox:~/ca/sub-ca#
```

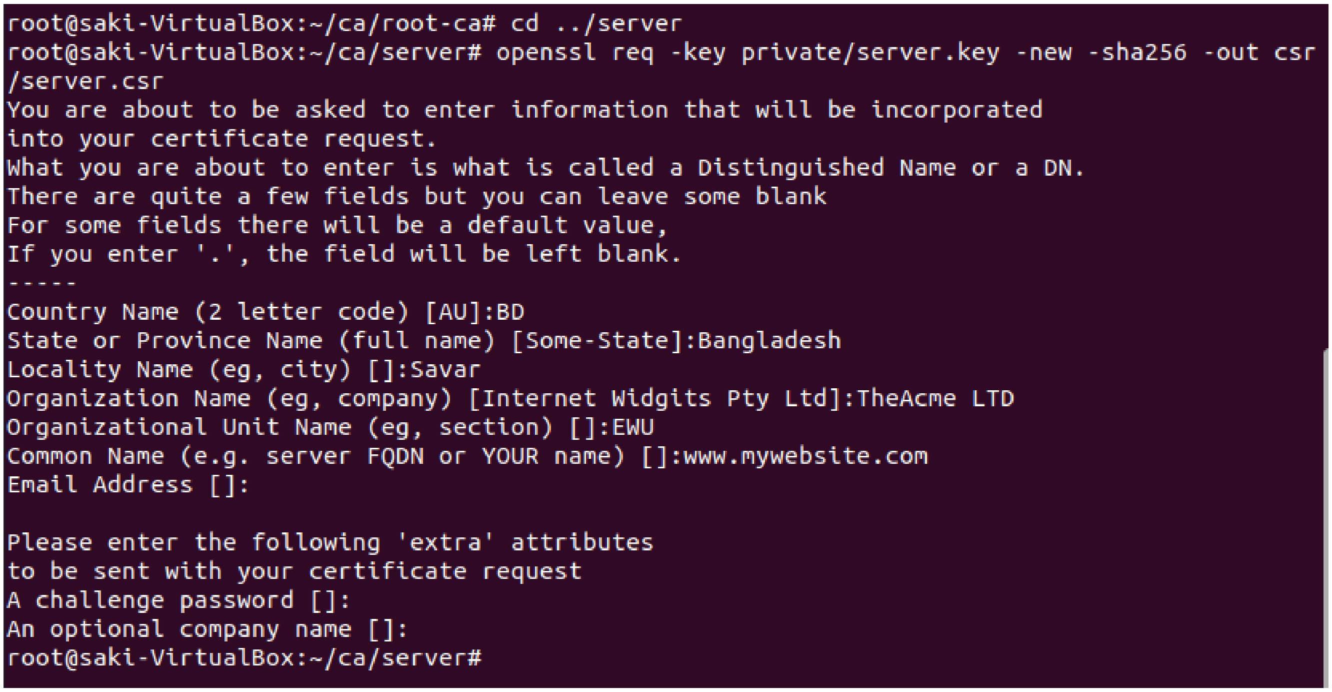
Now the RootCA sign the SubCA and then now anyone request for certification, SubCA will sign them.



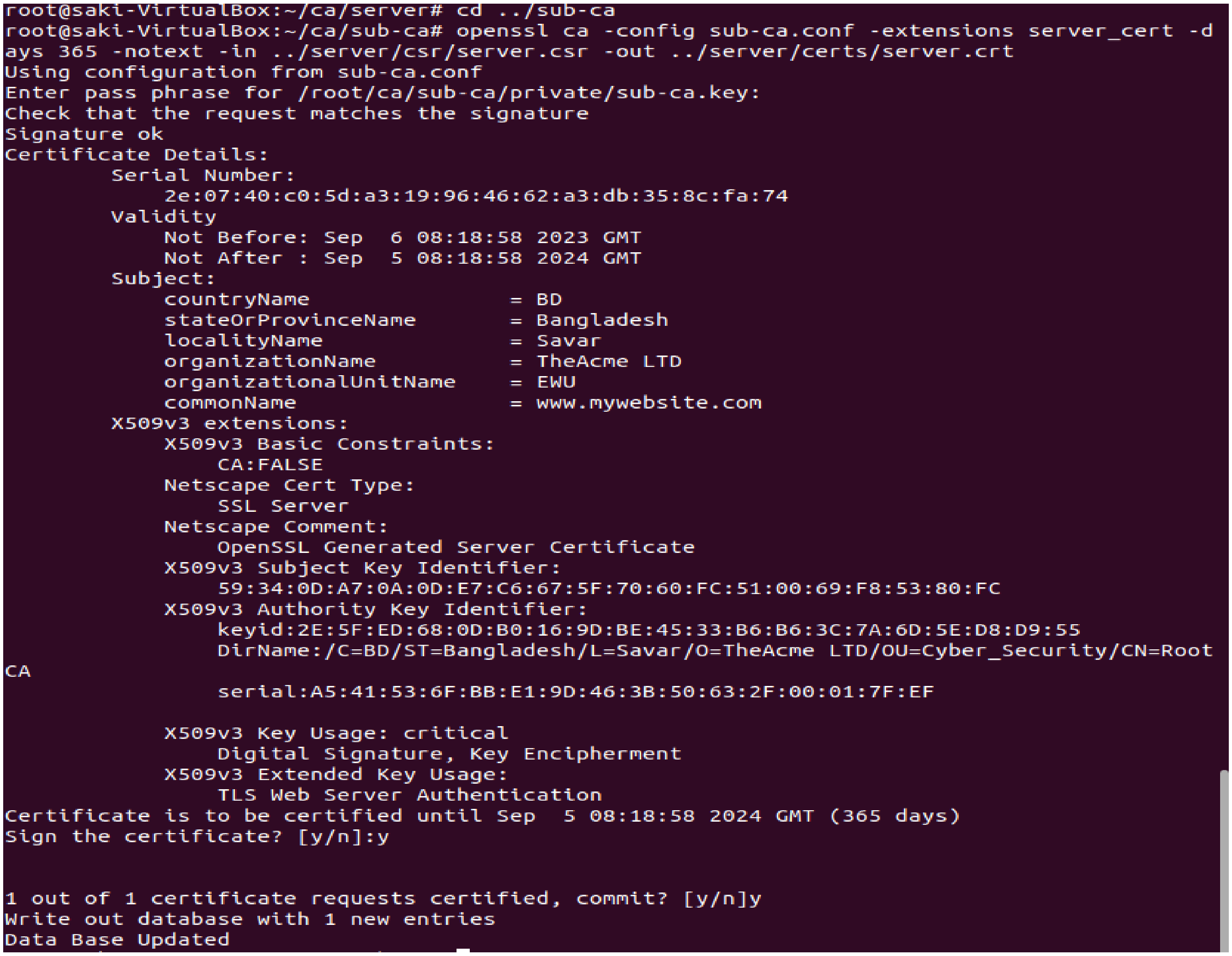
```
root@saki-VirtualBox:~/ca/root-ca# openssl ca -config root-ca.conf -extensions v3_intermedi
ate_ca -days 3652 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt
Using configuration from root-ca.conf
Enter pass phrase for /root/ca/root-ca/private/ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            a5:41:53:6f:bb:e1:9d:46:3b:50:63:2f:00:01:7f:ef
        Validity
            Not Before: Sep  6 08:01:10 2023 GMT
            Not After : Sep  5 08:01:10 2033 GMT
        Subject:
            countryName               = BD
            stateOrProvinceName       = Bangladesh
            organizationName          = TheAcme LTD
            organizationalUnitName    = Cyber_Security
            commonName                = SubCA
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                2E:5F:ED:68:0D:B0:16:9D:BE:45:33:B6:B6:3C:7A:6D:5E:D8:D9:55
            X509v3 Authority Key Identifier:
                keyid:DE:20:6C:52:41:04:0A:C3:00:02:92:69:D3:D9:05:19:E5:CF:AB:F3

            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until Sep  5 08:01:10 2033 GMT (3652 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
Data Base Updated
```

Step 06:

Now for the server certification we will move to the server folder, generate certificate, which will be signed by the SubCA. The detailed procedure will given below:



```
root@saki-VirtualBox:~/ca/root-ca# cd ../server
root@saki-VirtualBox:~/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 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]:BD
State or Province Name (full name) [Some-State]:Bangladesh
Locality Name (eg, city) []:Savar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TheAcme LTD
Organizational Unit Name (eg, section) []:EWU
Common Name (e.g. server FQDN or YOUR name) []:www.mywebsite.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@saki-VirtualBox:~/ca/server#
```



```
root@saki-VirtualBox:~/ca/server# cd ../sub-ca
root@saki-VirtualBox:~/ca/sub-ca# openssl ca -config sub-ca.conf -extensions server_cert -d
ays 365 -notext -in ../server/csr/server.csr -out ../server/certs/server.crt
Using configuration from sub-ca.conf
Enter pass phrase for /root/ca/sub-ca/private/sub-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            2e:07:40:c0:5d:a3:19:96:46:62:a3:db:35:8c:fa:74
        Validity
            Not Before: Sep  6 08:18:58 2023 GMT
            Not After : Sep  5 08:18:58 2024 GMT
        Subject:
            countryName               = BD
            stateOrProvinceName       = Bangladesh
            localityName              = Savar
            organizationName          = TheAcme LTD
            organizationalUnitName    = EWU
            commonName                = www.mywebsite.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Server
            Netscape Comment:
                OpenSSL Generated Server Certificate
            X509v3 Subject Key Identifier:
                59:34:0D:A7:0A:0D:E7:C6:67:5F:70:60:FC:51:00:69:F8:53:80:FC
            X509v3 Authority Key Identifier:
                keyid:2E:5F:ED:68:0D:B0:16:9D:BE:45:33:B6:B6:3C:7A:6D:5E:D8:D9:55
                DirName:/C=BD/ST=Bangladesh/L=Savar/O=TheAcme LTD/OU=Cyber_Security/CN=Root
CA
                serial:A5:41:53:6F:BB:E1:9D:46:3B:50:63:2F:00:01:7F:EF

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Sep  5 08:18:58 2024 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
Data Base Updated
```

Step 07:

Now we move to Server Certificate folder and merge the certificate by tyoing the following command:

cd ../server/certs/

cat server.crt ../../sub-ca/certs/sub-ca.crt ../../root-ca/certs/ca.crt > mywebsite.crt

Step 08:

At this stage we turn on SSL port which is 443.

openssl s_server -accept 443 -www -key private/server.key -cert certs/server.crt -CAfile ../sub-ca/certs/sub-ca.crt

Step 09:

Now we update our certificates by typing the following commands in the terminal.

cp ca/root-ca/certs/ca.crt /usr/local/share/ca-certificates/

update-ca-certificates -v

Step 10:

Copying Certificates:

cp /root/ca/root-ca/certs/ca.crt /home/[username]/[folder_name]

cp /root/ca/sub-ca/certs/sub-ca.crt /home/[username] /[folder_name]/

cp /root/ca/server/certs/verysecureserver.crt /[username] /server/[folder_name]/

cp /root/ca/server/certs/server.crt /home/[username] /[folder_name]/

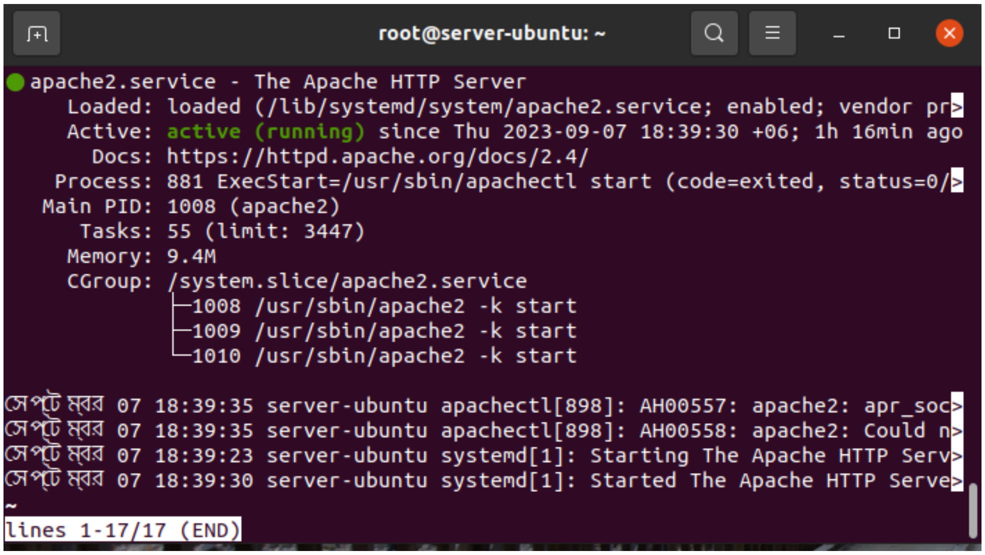
cp /root/ca/server/private/server.key /home/[username] /[folder_name]/

Step 11:

Open a new terminal and type:

sudo apt install apache2

systemctl status apache2

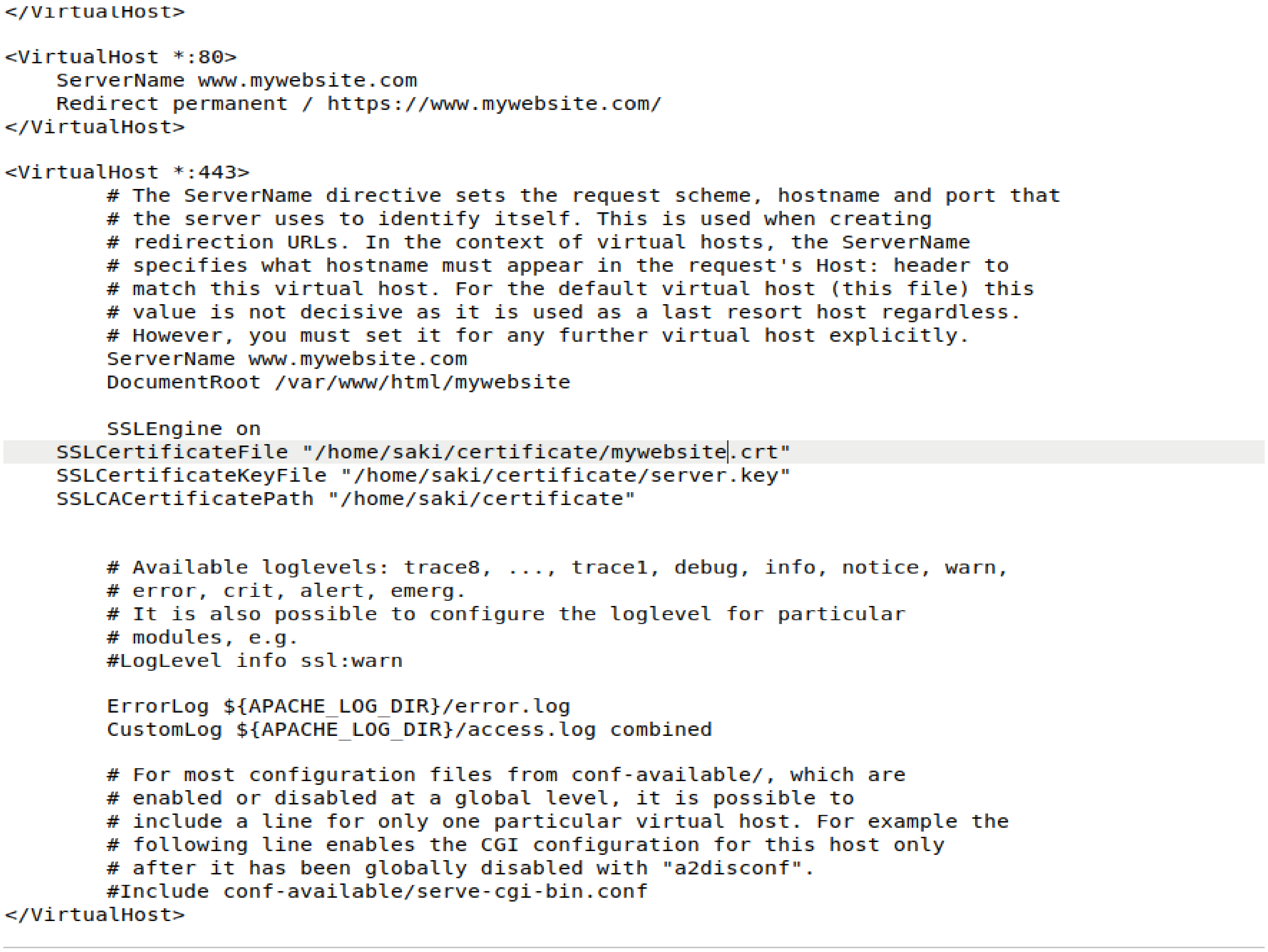


```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pr>
     Active: active (running) since Thu 2023-09-07 18:39:30 +06; 1h 16min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 881 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/>
   Main PID: 1008 (apache2)
      Tasks: 55 (limit: 3447)
     Memory: 9.4M
     CGroup: /system.slice/apache2.service
             ├─1008 /usr/sbin/apache2 -k start
             ├─1009 /usr/sbin/apache2 -k start
             └─1010 /usr/sbin/apache2 -k start

সেপ্টেম্বর 07 18:39:35 server-ubuntu apachectl[898]: AH00557: apache2: apr_soc>
সেপ্টেম্বর 07 18:39:35 server-ubuntu apachectl[898]: AH00558: apache2: Could n>
সেপ্টেম্বর 07 18:39:23 server-ubuntu systemd[1]: Starting The Apache HTTP Serv>
সেপ্টেম্বর 07 18:39:30 server-ubuntu systemd[1]: Started The Apache HTTP Serve>
~
lines 1-17/17 (END)
```

Check the status of apache2, if apache2 is not showing active, then type the following command:

sudo start apache2. This will start your apache server.

Now go to this location: **/etc/apache2/sites-enabled** and paste the following code in 000-default.conf file.



```
</VirtualHost>

<VirtualHost *:80>
    ServerName www.mywebsite.com
    Redirect permanent / https://www.mywebsite.com/
</VirtualHost>

<VirtualHost *:443>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.mywebsite.com
        DocumentRoot /var/www/html/mywebsite

        SSLEngine on
    SSLCertificateFile "/home/saki/certificate/mywebsite.crt"
    SSLCertificateKeyFile "/home/saki/certificate/server.key"
    SSLCACertificatePath "/home/saki/certificate"


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

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

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Step 12:

Now go to this location: **var/www/html** and create a folder named **mywebsite** and make a html file for your website inside of this folder.

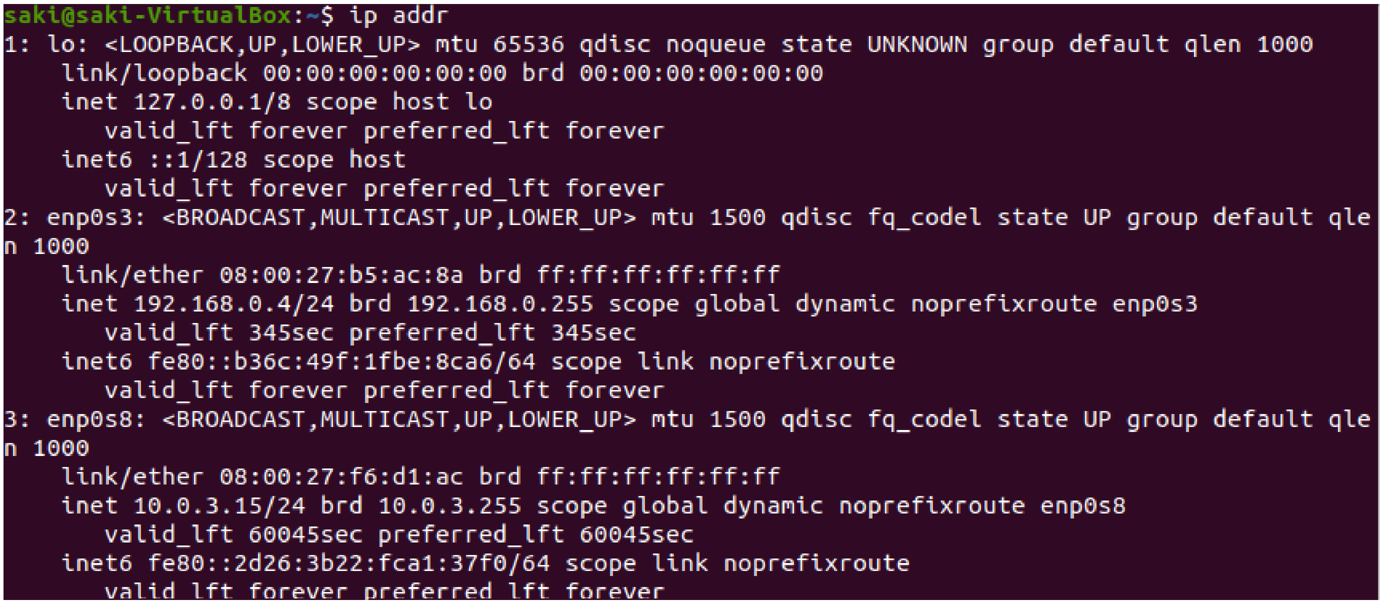
4.0 DNS SETUP (Server)

First install the following commands:

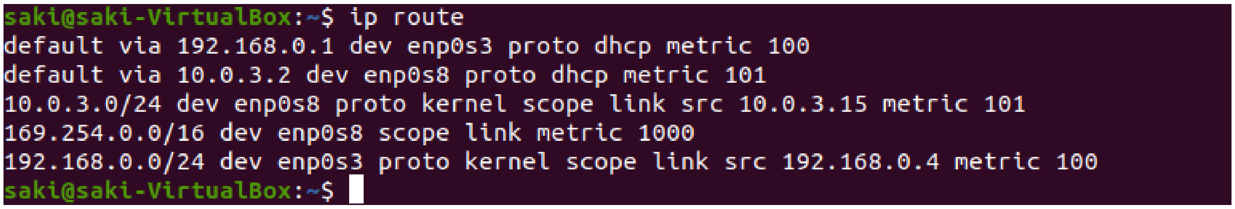
sudo apt install net-tools

sudo apt install bind9

sudo apt-get install bind9 bind9utils bind9-doc



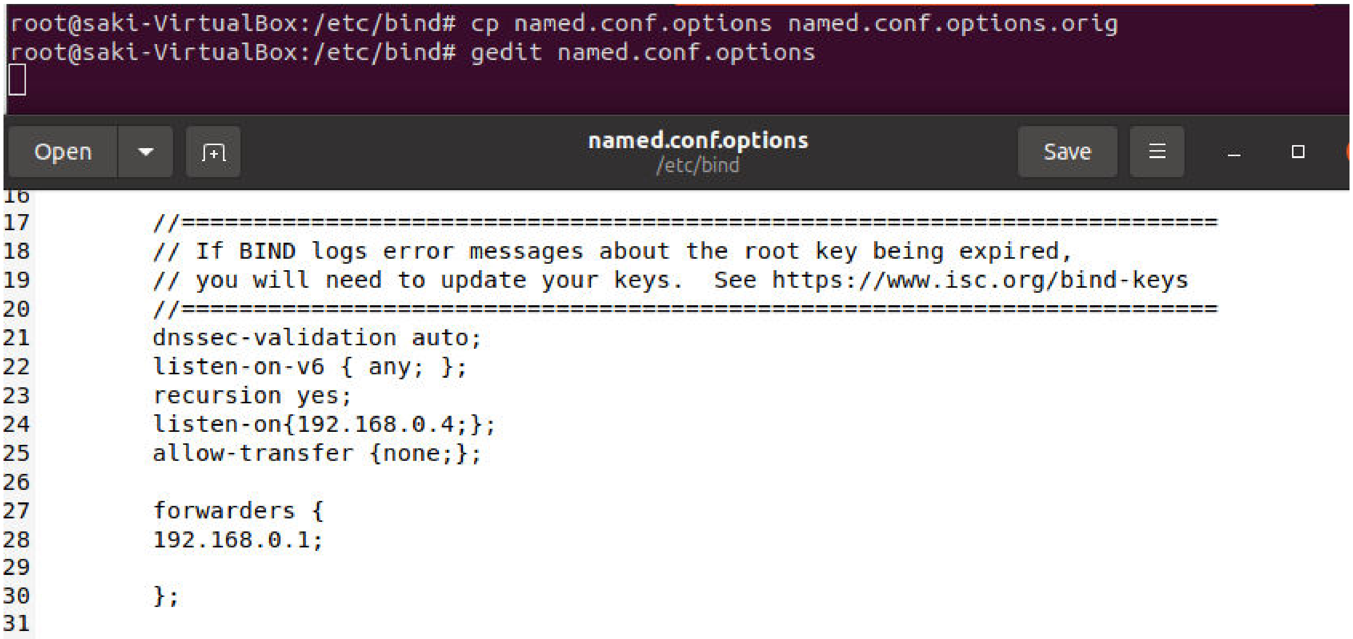
```
saki@saki-VirtualBox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qle
n 1000
    link/ether 08:00:27:b5:ac:8a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
       valid_lft 345sec preferred_lft 345sec
    inet6 fe80::b36c:49f:1fbe:8ca6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qle
n 1000
    link/ether 08:00:27:f6:d1:ac brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.15/24 brd 10.0.3.255 scope global dynamic noprefixroute enp0s8
       valid_lft 60045sec preferred_lft 60045sec
    inet6 fe80::2d26:3b22:fca1:37f0/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```



```
saki@saki-VirtualBox:~$ ip route
default via 192.168.0.1 dev enp0s3 proto dhcp metric 100
default via 10.0.3.2 dev enp0s8 proto dhcp metric 101
10.0.3.0/24 dev enp0s8 proto kernel scope link src 10.0.3.15 metric 101
169.254.0.0/16 dev enp0s8 scope link metric 1000
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.4 metric 100
saki@saki-VirtualBox:~$
```

Here we can see, our ip address is 192.168.0.4 and our default route is 192.168.0.1, using this ip and route, we will configure our DNS server.

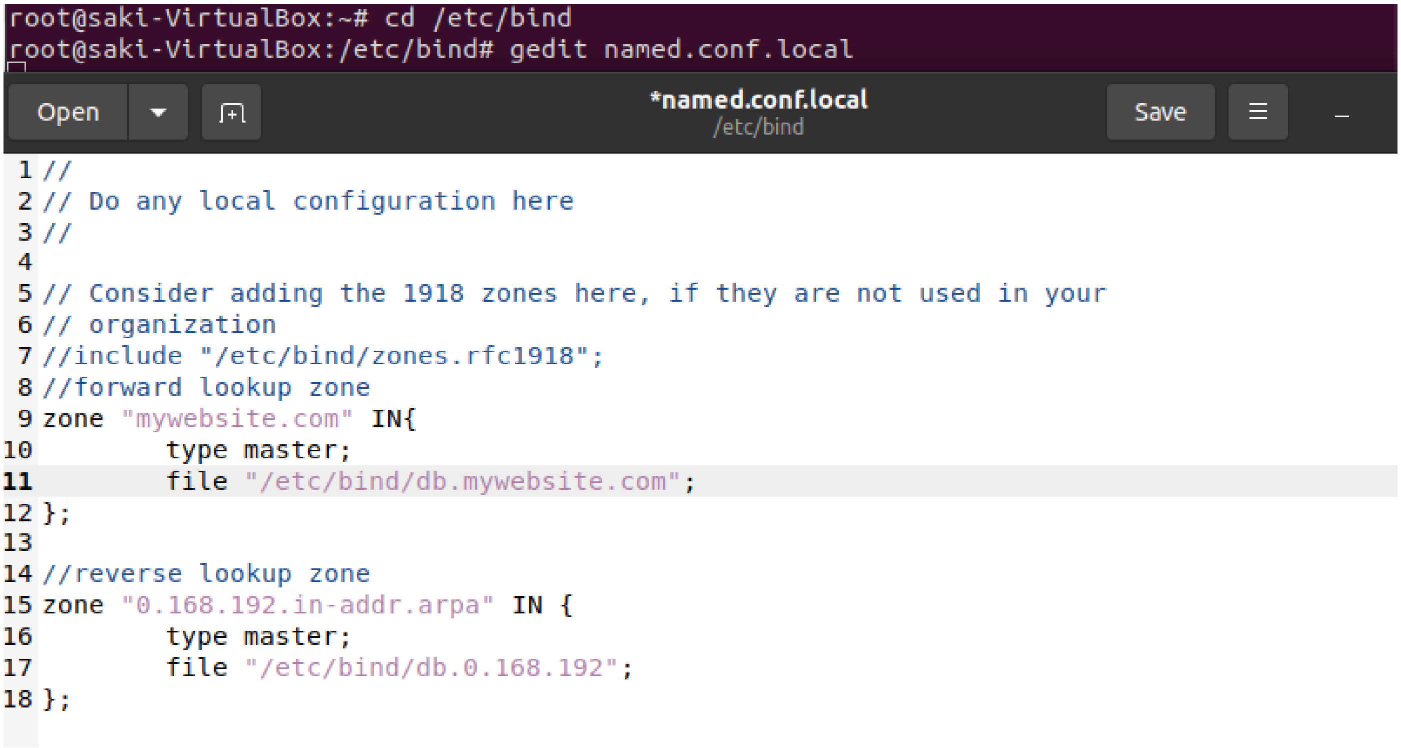
Next go to this location: **sudo /etc/bind** and edit the **named.conf.options** file.



```
root@saki-VirtualBox:/etc/bind# cp named.conf.options named.conf.options.orig
root@saki-VirtualBox:/etc/bind# gedit named.conf.options
```

```
        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;
        listen-on-v6 { any; };
        recursion yes;
        listen-on{192.168.0.4;};
        allow-transfer {none;};

        forwarders {
        192.168.0.1;

        };

```

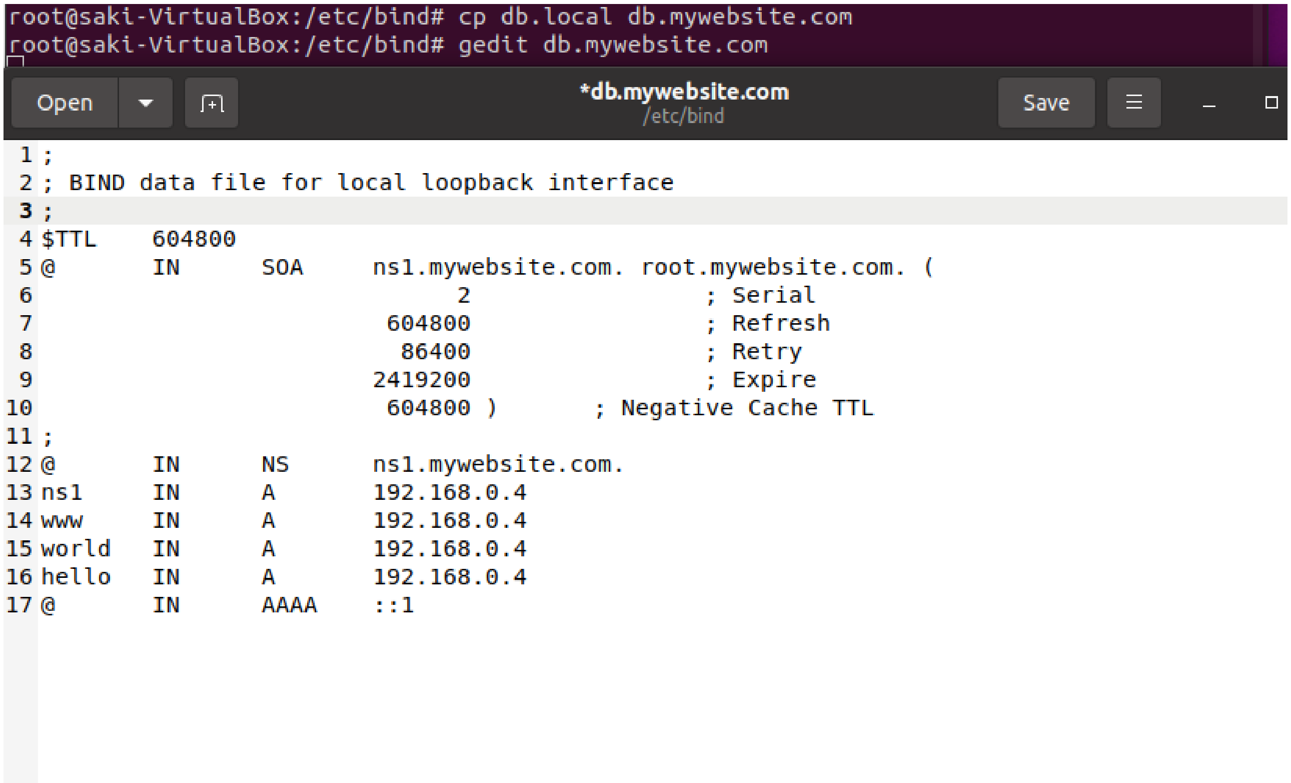
Then edit the **named.conf.local** file.



```
root@saki-VirtualBox:~# cd /etc/bind
root@saki-VirtualBox:/etc/bind# gedit named.conf.local
```

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
//forward lookup zone
zone "mywebsite.com" IN{
        type master;
        file "/etc/bind/db.mywebsite.com";
};

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

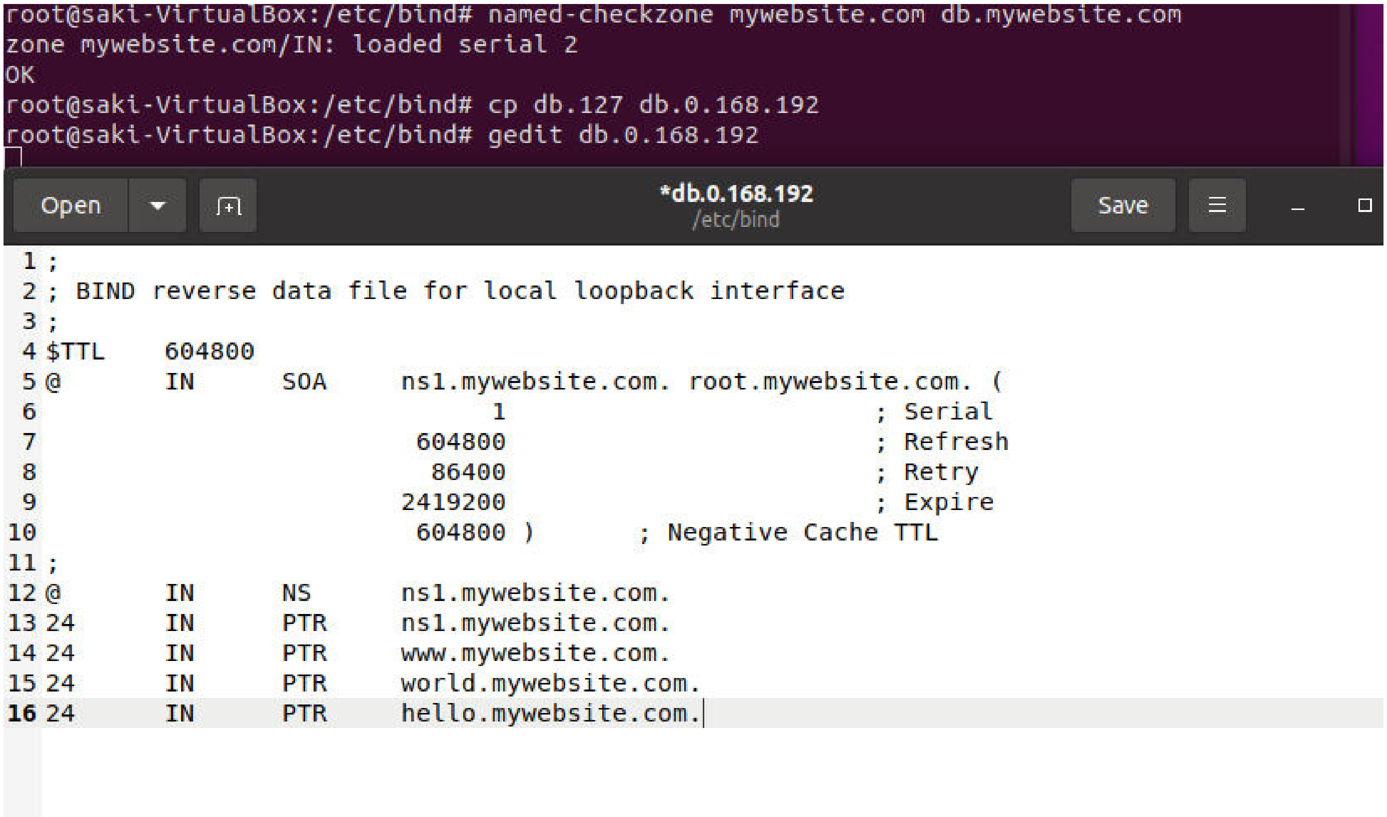
Edit db.mywebsite.com file-



```
root@saki-VirtualBox:/etc/bind# cp db.local db.mywebsite.com
root@saki-VirtualBox:/etc/bind# gedit db.mywebsite.com
```

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.mywebsite.com. root.mywebsite.com. (
                              2                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.mywebsite.com.
ns1     IN      A       192.168.0.4
www     IN      A       192.168.0.4
world   IN      A       192.168.0.4
hello   IN      A       192.168.0.4
@       IN      AAAA    ::1
```

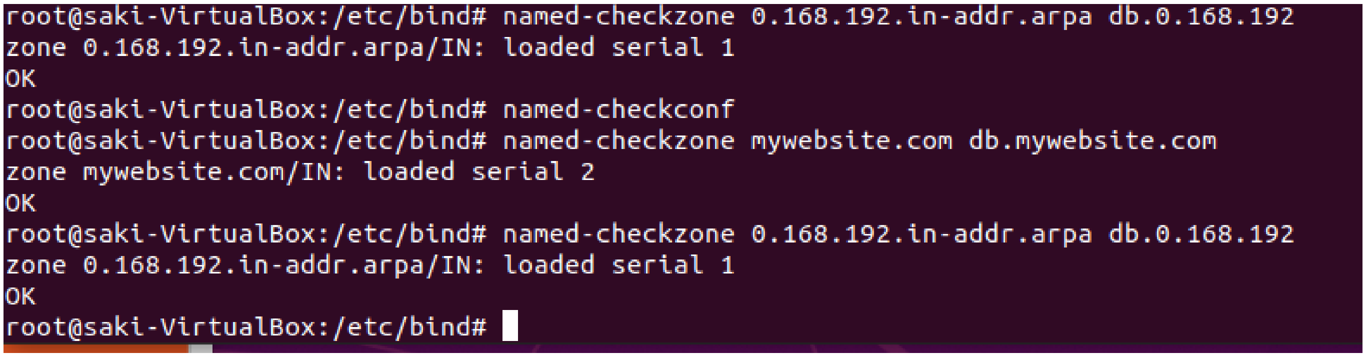
Edit db.0.168.192 file-



```
root@saki-VirtualBox:/etc/bind# named-checkzone mywebsite.com db.mywebsite.com
zone mywebsite.com/IN: loaded serial 2
OK
root@saki-VirtualBox:/etc/bind# cp db.127 db.0.168.192
root@saki-VirtualBox:/etc/bind# gedit db.0.168.192
```

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

To Check all the files working fine or not



```
root@saki-VirtualBox:/etc/bind# named-checkzone 0.168.192.in-addr.arpa db.0.168.192
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
OK
root@saki-VirtualBox:/etc/bind# named-checkconf
root@saki-VirtualBox:/etc/bind# named-checkzone mywebsite.com db.mywebsite.com
zone mywebsite.com/IN: loaded serial 2
OK
root@saki-VirtualBox:/etc/bind# named-checkzone 0.168.192.in-addr.arpa db.0.168.192
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
OK
root@saki-VirtualBox:/etc/bind# 
```

Now restart the bind9 service by typing command: **service bind9 restart**

Edit the resolve.conf file:

sudo systemctl status resolvconf.service

sudo apt update

sudo apt install resolvconf

sudo systemctl status resolvconf.service

(if resolveconf isn't running, enable then start it)

sudo systemctl enable resolvconf.service

sudo systemctl start resolvconf.service

(check resolveconf status)

```
sudo systemctl status resolvconf.service
```

(edit the head file)

```
sudo nano /etc/resolvconf/resolv.conf.d/head
```

(enter your nameservers below the comments)

```
nameserver 192.168.0.4
nameserver 192.168.0.1
search localdomain
```

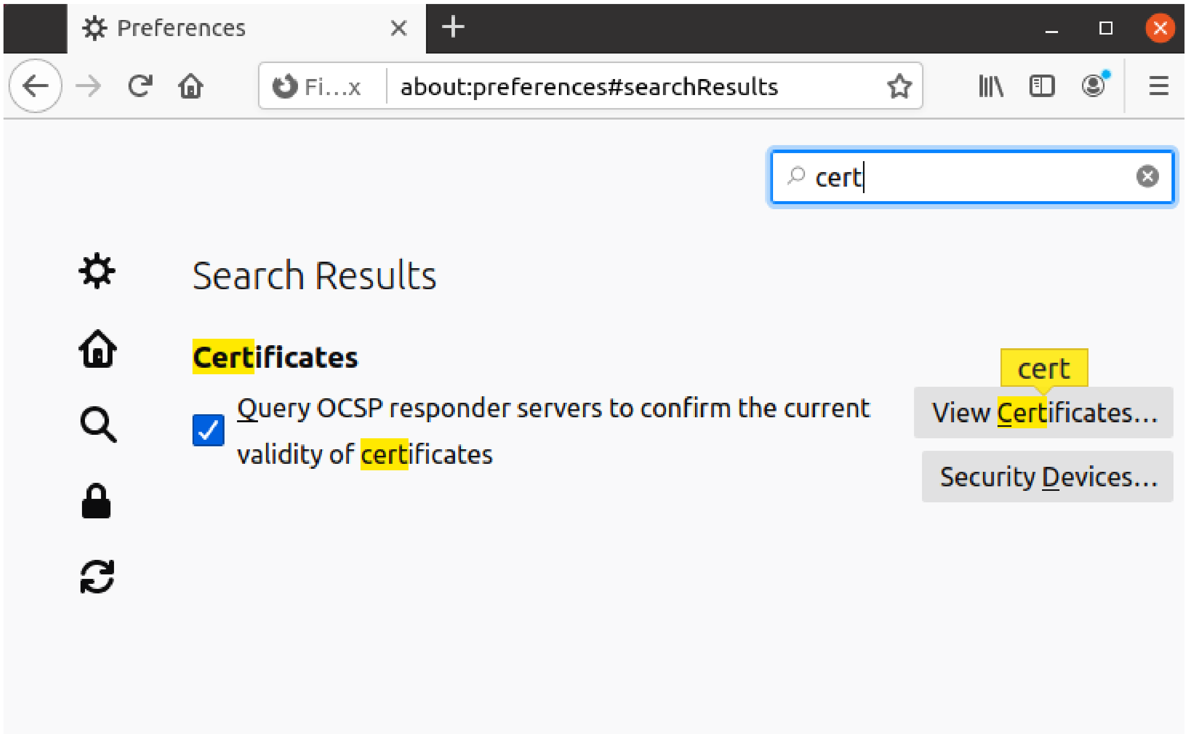
(update resolve.conf file)

```
sudo resolvconf --enable-updates
sudo resolvconf -u
```

(check if changes we successful)

```
sudo nano /etc/resolv.conf
```

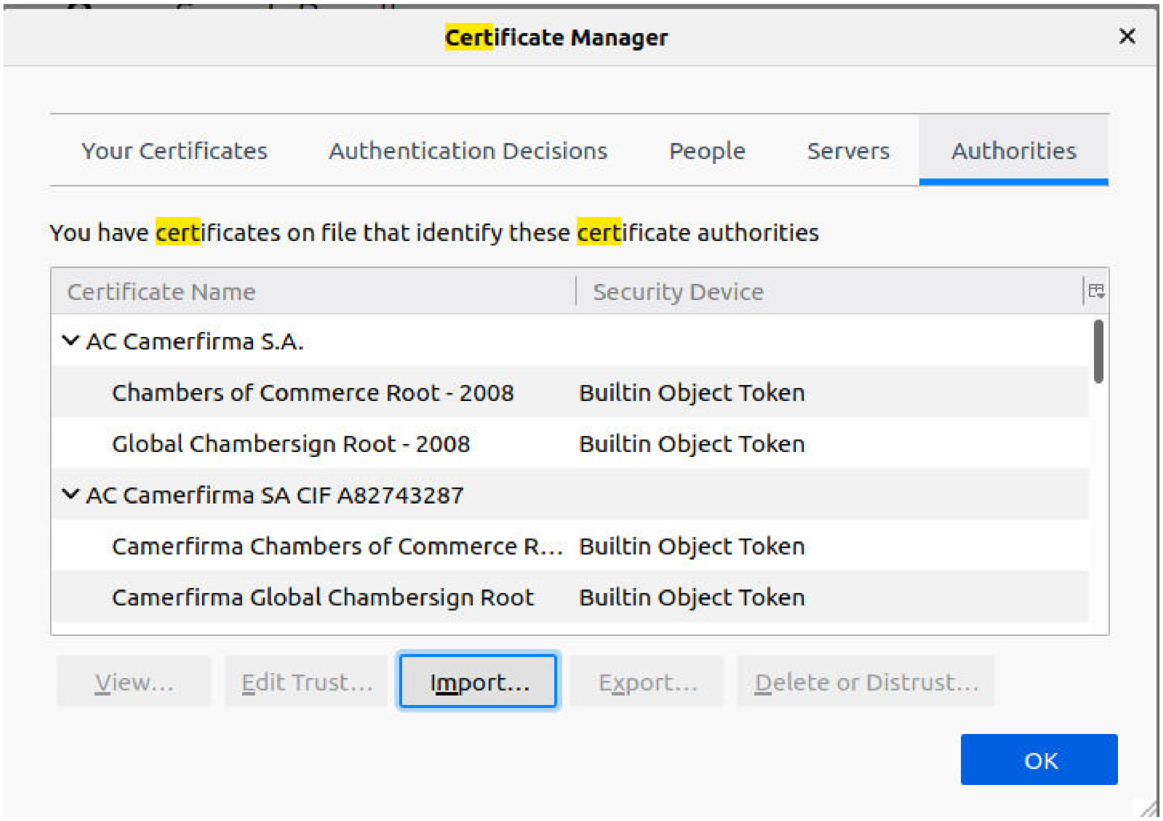
Now command nslookup www.mywebsite.com **and this will show you that the reply is coming from your ip. Now Paste root certificate in the browser and search for your website, this will show your website with padlock icon.**



Search Results

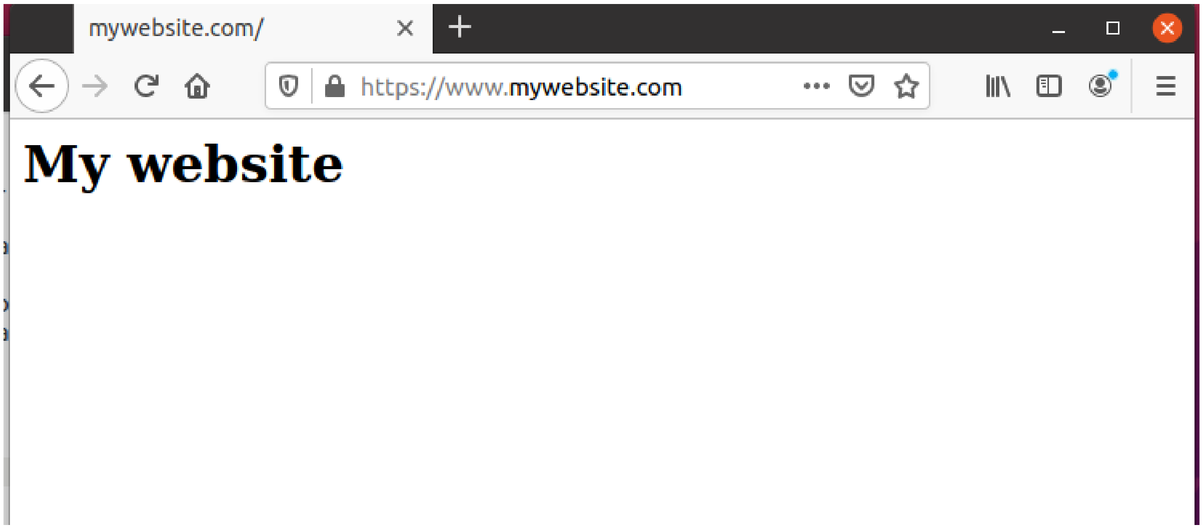
Certificates

☑ Query OCSP responder servers to confirm the current validity of certificates

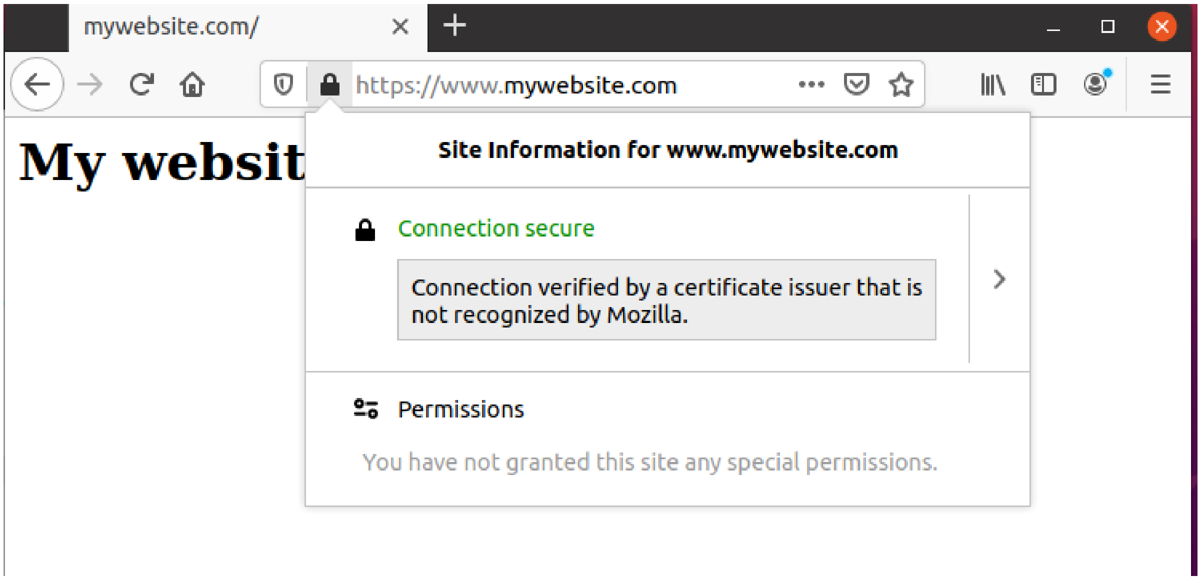


You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
AC Camerfirma SA CIF A82743287	
Camerfirma Chambers of Commerce R…	Builtin Object Token
Camerfirma Global Chambersign Root	Builtin Object Token



My website



mywebsite.com/

https://www.mywebsite.com

My websit

Site Information for www.mywebsite.com

Connection secure

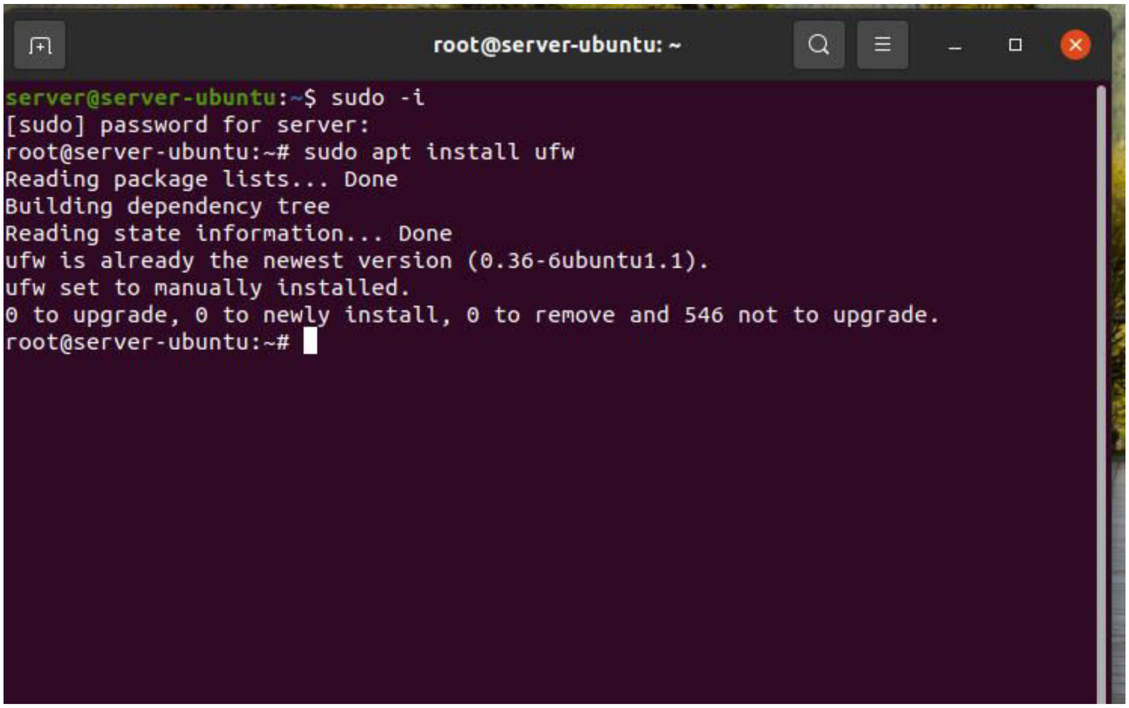
Connection verified by a certificate issuer that is not recognized by Mozilla.

Permissions

You have not granted this site any special permissions.

5.0 Firewall Setup (Server Machine)

To install firewall on servr pc, first take root access in terminal then type “**sudo apt install ufw**”. This will install firewall in your machine.



root@server-ubuntu: ~

```
server@server-ubuntu:~$ sudo -i
[sudo] password for server:
root@server-ubuntu:~# sudo apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1.1).
ufw set to manually installed.
0 to upgrade, 0 to newly install, 0 to remove and 546 not to upgrade.
root@server-ubuntu:~#
```

Then set some rules for firewall:

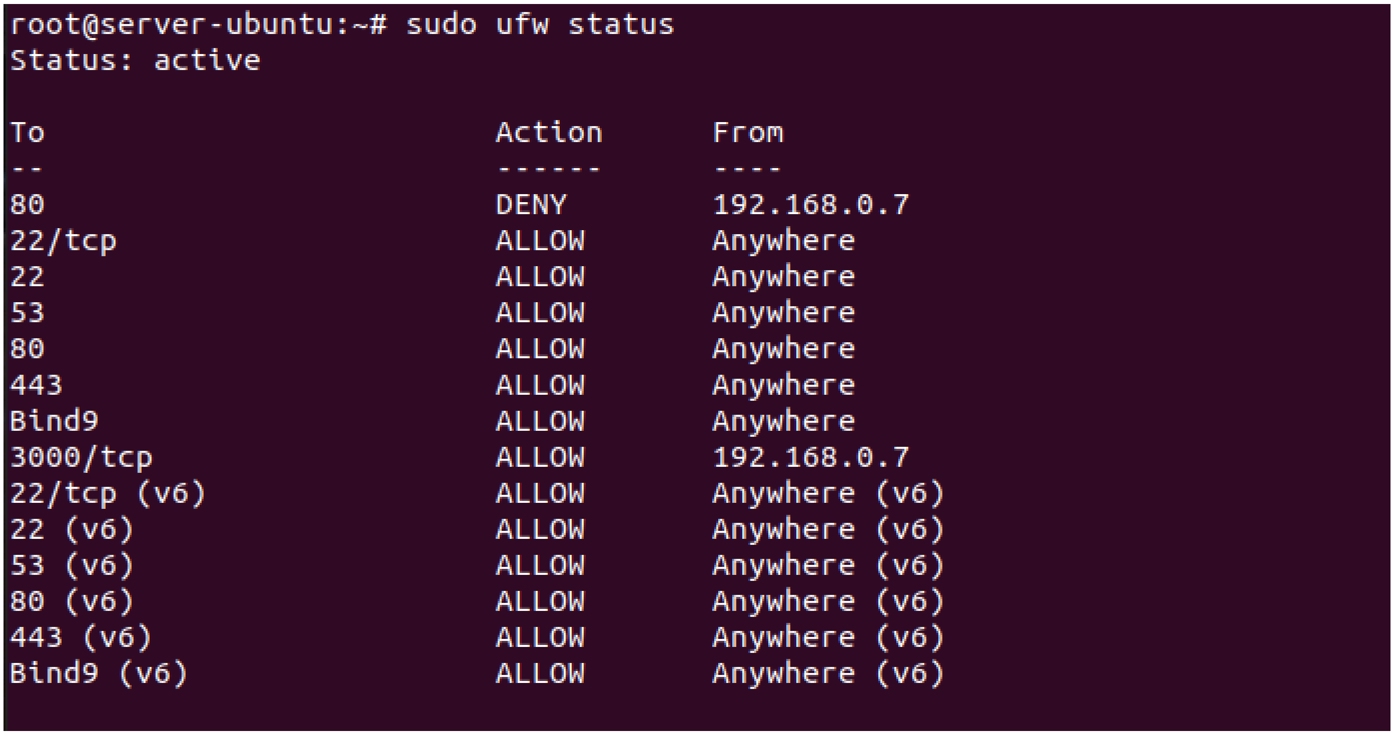
- ufw default allow outgoing
- ufw default deny incoming
- ufw allow ssh
- ufw enable

- sudo ufw allow 22
- sudo ufw allow 53
- sudo ufw allow 80 (for http)
- sudo ufw allow 443(for https)

You can check the rules that you set for your firewall:

Type: **sudo ufw status**. This will show you all the rules that you have set for your firewall.

If you want delete some rules just type in terminal: **sudo ufw delete 1** [Here I want to delete the 1st rule which is denying 192.168.0.7 ip to port 80].



```
root@server-ubuntu:~# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80                         DENY        192.168.0.7
22/tcp                     ALLOW       Anywhere
22                         ALLOW       Anywhere
53                         ALLOW       Anywhere
80                         ALLOW       Anywhere
443                        ALLOW       Anywhere
Bind9                      ALLOW       Anywhere
3000/tcp                   ALLOW       192.168.0.7
22/tcp (v6)                ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
53 (v6)                    ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
Bind9 (v6)                 ALLOW       Anywhere (v6)
```

6.0 Wireshark and Snort installation (Server)

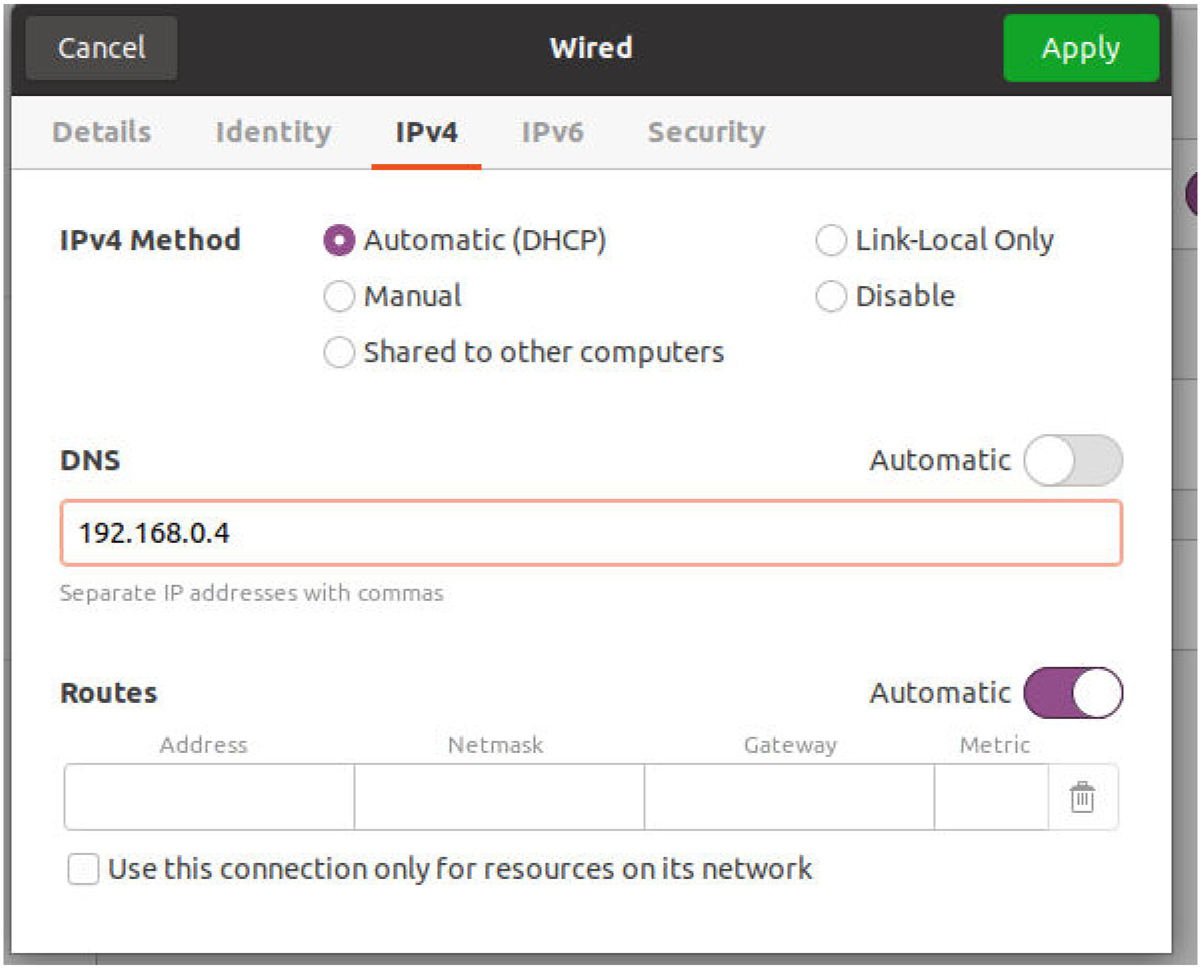
To install wireshark, first take root access in the terminal and type **sudo apt install wireshark**

and to install snort command: **sudo apt-get install snort.**

After installing the snort, set rules for the possible SYN Flood Attack. To do this, go to this location: **/etc/snort/rules** and edit file named local.rules for identifying SYN Flood Attack. Again Now open a new terminal and take the root access and type the following to run the snort.
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3
[As we use enp0s3 network for our DNS Configuration and configure and set snort rules on enp0s3 network]

7.0 Server PC Configuration:

Our Server machine's ip is 192.168.0.4 which we will as our DNS address in our client machine. To do this edit the network file of client pc and set server's ip as Client pc DNS address.



IPv4 Method
● Automatic (DHCP) ○ Link-Local Only
○ Manual ○ Disable
○ Shared to other computers

DNS Automatic

192.168.0.4

Separate IP addresses with commas

Routes Automatic

Address	Netmask	Gateway	Metric

□ Use this connection only for resources on its network

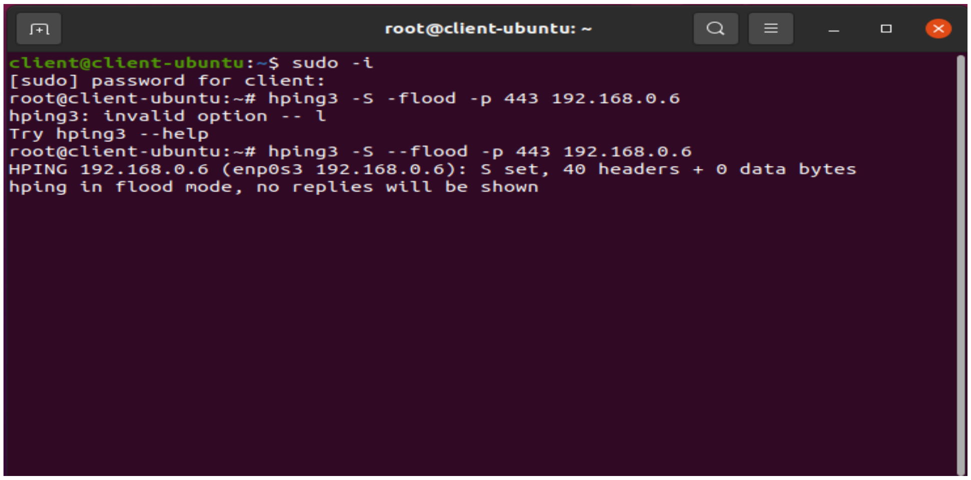
Now Open a terminal and type the following commands:

sudo apt install wireshark

sudo apt install hping3

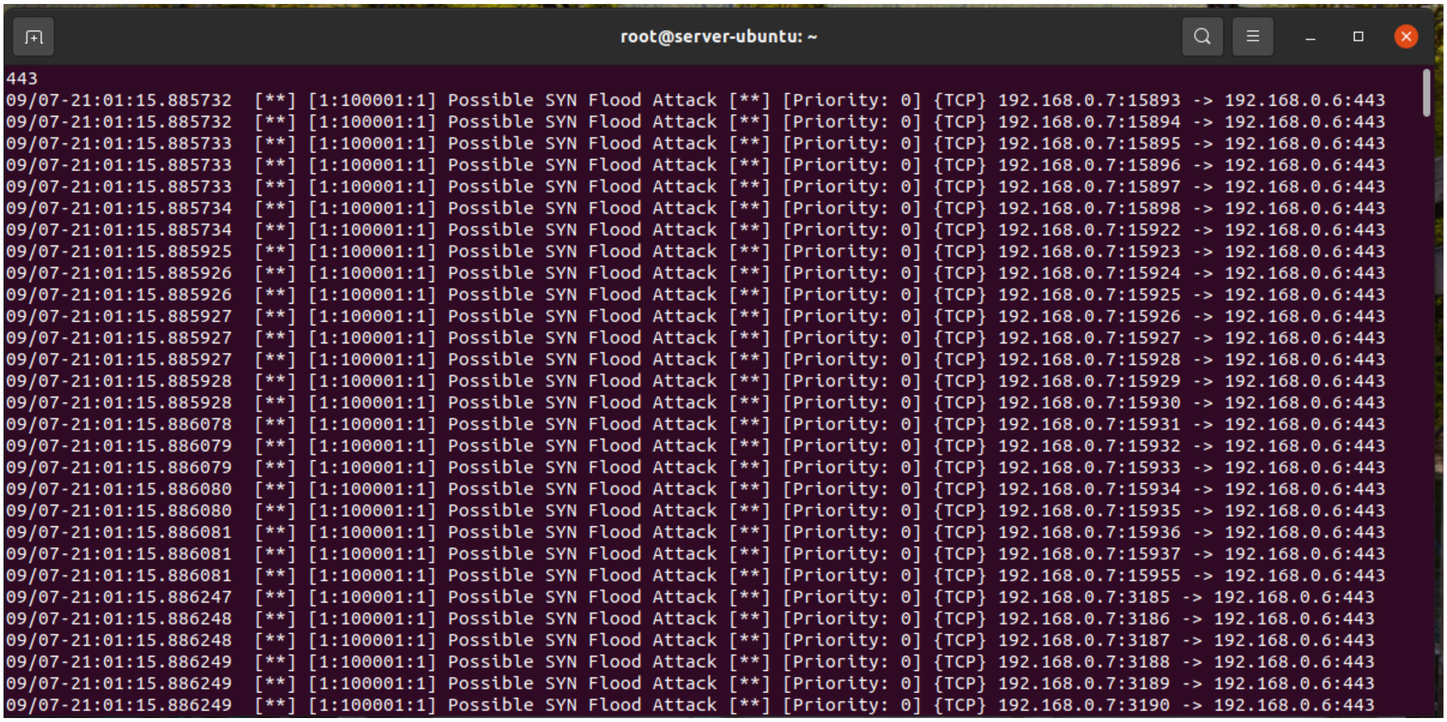
sudo apt install ssh

Performing attack from Client PC:



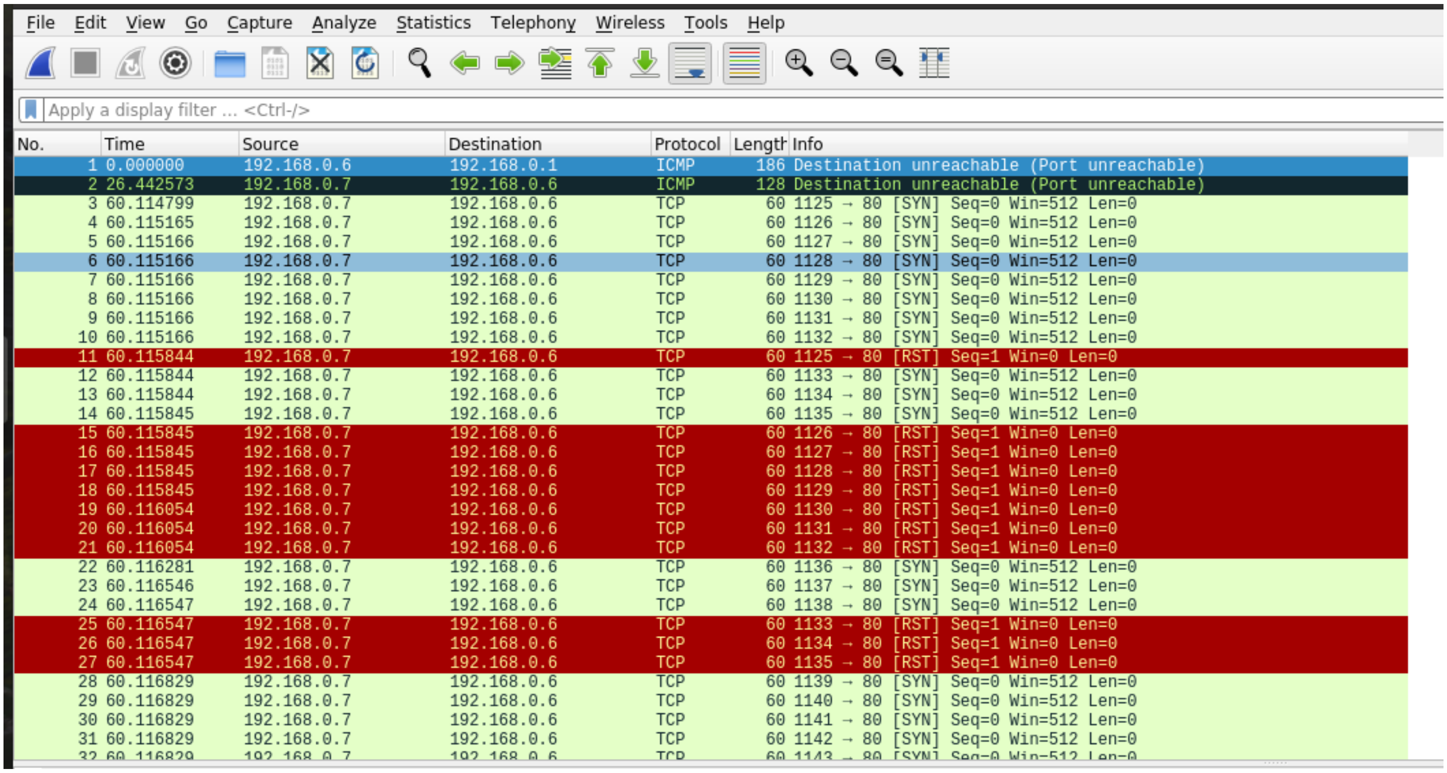
```
client@client-ubuntu:~$ sudo -i
[sudo] password for client:
root@client-ubuntu:~# hping3 -S -flood -p 443 192.168.0.6
hping3: invalid option -- l
Try hping3 --help
root@client-ubuntu:~# hping3 -S --flood -p 443 192.168.0.6
HPING 192.168.0.6 (enp0s3 192.168.0.6): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Attack from Client PC.



```
443
09/07-21:01:15.885732  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15893 -> 192.168.0.6:443
09/07-21:01:15.885732  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15894 -> 192.168.0.6:443
09/07-21:01:15.885733  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15895 -> 192.168.0.6:443
09/07-21:01:15.885733  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15896 -> 192.168.0.6:443
09/07-21:01:15.885733  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15897 -> 192.168.0.6:443
09/07-21:01:15.885734  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15898 -> 192.168.0.6:443
09/07-21:01:15.885734  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15922 -> 192.168.0.6:443
09/07-21:01:15.885925  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15923 -> 192.168.0.6:443
09/07-21:01:15.885926  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15924 -> 192.168.0.6:443
09/07-21:01:15.885926  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15925 -> 192.168.0.6:443
09/07-21:01:15.885927  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15926 -> 192.168.0.6:443
09/07-21:01:15.885927  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15927 -> 192.168.0.6:443
09/07-21:01:15.885927  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15928 -> 192.168.0.6:443
09/07-21:01:15.885928  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15929 -> 192.168.0.6:443
09/07-21:01:15.885928  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15930 -> 192.168.0.6:443
09/07-21:01:15.886078  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15931 -> 192.168.0.6:443
09/07-21:01:15.886079  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15932 -> 192.168.0.6:443
09/07-21:01:15.886079  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15933 -> 192.168.0.6:443
09/07-21:01:15.886080  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15934 -> 192.168.0.6:443
09/07-21:01:15.886080  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15935 -> 192.168.0.6:443
09/07-21:01:15.886081  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15936 -> 192.168.0.6:443
09/07-21:01:15.886081  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15937 -> 192.168.0.6:443
09/07-21:01:15.886081  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:15955 -> 192.168.0.6:443
09/07-21:01:15.886247  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3185 -> 192.168.0.6:443
09/07-21:01:15.886248  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3186 -> 192.168.0.6:443
09/07-21:01:15.886248  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3187 -> 192.168.0.6:443
09/07-21:01:15.886249  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3188 -> 192.168.0.6:443
09/07-21:01:15.886249  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3189 -> 192.168.0.6:443
09/07-21:01:15.886249  [**] [1:100001:1] Possible SYN Flood Attack [**] [Priority: 0] {TCP} 192.168.0.7:3190 -> 192.168.0.6:443
```

Attack Detection from Server PC. Now if we want block the attacker IP, we need to see the snort log file to see the attacker ip. To do this go to this location of your server pc: **/var/log/snort** from here you can see the log file. Now open that log file with Wireshark and you can see the Attacker ip.

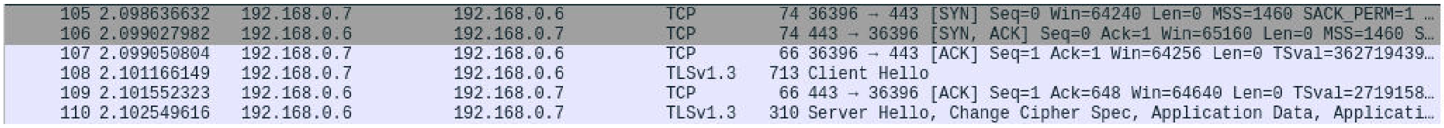


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.6	192.168.0.1	ICMP	186	Destination unreachable (Port unreachable)
2	26.442573	192.168.0.7	192.168.0.6	ICMP	128	Destination unreachable (Port unreachable)
3	60.114799	192.168.0.7	192.168.0.6	TCP	60	1125 → 80 [SYN] Seq=0 Win=512 Len=0
4	60.115165	192.168.0.7	192.168.0.6	TCP	60	1126 → 80 [SYN] Seq=0 Win=512 Len=0
5	60.115166	192.168.0.7	192.168.0.6	TCP	60	1127 → 80 [SYN] Seq=0 Win=512 Len=0
6	60.115166	192.168.0.7	192.168.0.6	TCP	60	1128 → 80 [SYN] Seq=0 Win=512 Len=0
7	60.115166	192.168.0.7	192.168.0.6	TCP	60	1129 → 80 [SYN] Seq=0 Win=512 Len=0
8	60.115166	192.168.0.7	192.168.0.6	TCP	60	1130 → 80 [SYN] Seq=0 Win=512 Len=0
9	60.115166	192.168.0.7	192.168.0.6	TCP	60	1131 → 80 [SYN] Seq=0 Win=512 Len=0
10	60.115166	192.168.0.7	192.168.0.6	TCP	60	1132 → 80 [SYN] Seq=0 Win=512 Len=0
11	60.115844	192.168.0.7	192.168.0.6	TCP	60	1125 → 80 [RST] Seq=1 Win=0 Len=0
12	60.115844	192.168.0.7	192.168.0.6	TCP	60	1133 → 80 [SYN] Seq=0 Win=512 Len=0
13	60.115844	192.168.0.7	192.168.0.6	TCP	60	1134 → 80 [SYN] Seq=0 Win=512 Len=0
14	60.115845	192.168.0.7	192.168.0.6	TCP	60	1135 → 80 [SYN] Seq=0 Win=512 Len=0
15	60.115845	192.168.0.7	192.168.0.6	TCP	60	1126 → 80 [RST] Seq=1 Win=0 Len=0
16	60.115845	192.168.0.7	192.168.0.6	TCP	60	1127 → 80 [RST] Seq=1 Win=0 Len=0
17	60.115845	192.168.0.7	192.168.0.6	TCP	60	1128 → 80 [RST] Seq=1 Win=0 Len=0
18	60.115845	192.168.0.7	192.168.0.6	TCP	60	1129 → 80 [RST] Seq=1 Win=0 Len=0
19	60.116054	192.168.0.7	192.168.0.6	TCP	60	1130 → 80 [RST] Seq=1 Win=0 Len=0
20	60.116054	192.168.0.7	192.168.0.6	TCP	60	1131 → 80 [RST] Seq=1 Win=0 Len=0
21	60.116054	192.168.0.7	192.168.0.6	TCP	60	1132 → 80 [RST] Seq=1 Win=0 Len=0
22	60.116281	192.168.0.7	192.168.0.6	TCP	60	1136 → 80 [SYN] Seq=0 Win=512 Len=0
23	60.116546	192.168.0.7	192.168.0.6	TCP	60	1137 → 80 [SYN] Seq=0 Win=512 Len=0
24	60.116547	192.168.0.7	192.168.0.6	TCP	60	1138 → 80 [SYN] Seq=0 Win=512 Len=0
25	60.116547	192.168.0.7	192.168.0.6	TCP	60	1133 → 80 [RST] Seq=1 Win=0 Len=0
26	60.116547	192.168.0.7	192.168.0.6	TCP	60	1134 → 80 [RST] Seq=1 Win=0 Len=0
27	60.116547	192.168.0.7	192.168.0.6	TCP	60	1135 → 80 [RST] Seq=1 Win=0 Len=0
28	60.116829	192.168.0.7	192.168.0.6	TCP	60	1139 → 80 [SYN] Seq=0 Win=512 Len=0
29	60.116829	192.168.0.7	192.168.0.6	TCP	60	1140 → 80 [SYN] Seq=0 Win=512 Len=0
30	60.116829	192.168.0.7	192.168.0.6	TCP	60	1141 → 80 [SYN] Seq=0 Win=512 Len=0
31	60.116829	192.168.0.7	192.168.0.6	TCP	60	1142 → 80 [SYN] Seq=0 Win=512 Len=0
32	60.116829	192.168.0.7	192.168.0.6	TCP	60	1143 → 80 [SYN] Seq=0 Win=512 Len=0

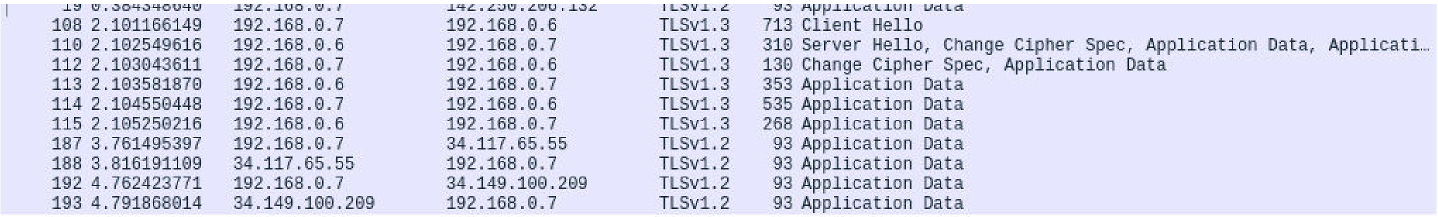
Here In source column, We can see the attacker ip, now using firewall block the attacker ip from server. To to do that type:
sudo ufw insert 1 deny from 192.168.0.7 to any port 80 and save it.

8.0 TCP & TLS Handshake:

From the client pc open wireshark and starting capture packet from enp0s3 adapter. Now hit the website from Client pc bowser.



```
105 2.098636632   192.168.0.7     192.168.0.6     TCP       74 36396 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 …
106 2.099027982   192.168.0.6     192.168.0.7     TCP       74 443 → 36396 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 S…
107 2.099050804   192.168.0.7     192.168.0.6     TCP       66 36396 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=362719439…
108 2.101166149   192.168.0.7     192.168.0.6     TLSv1.3  713 Client Hello
109 2.101552323   192.168.0.6     192.168.0.7     TCP       66 443 → 36396 [ACK] Seq=1 Ack=648 Win=64640 Len=0 TSval=2719158…
110 2.102549616   192.168.0.6     192.168.0.7     TLSv1.3  310 Server Hello, Change Cipher Spec, Application Data, Applicati…
```



```
 19 0.384348640   192.168.0.7     142.250.200.132 TLSv1.2   93 Application Data
108 2.101166149   192.168.0.7     192.168.0.6     TLSv1.3  713 Client Hello
110 2.102549616   192.168.0.6     192.168.0.7     TLSv1.3  310 Server Hello, Change Cipher Spec, Application Data, Applicati…
112 2.103043611   192.168.0.7     192.168.0.6     TLSv1.3  130 Change Cipher Spec, Application Data
113 2.103581870   192.168.0.6     192.168.0.7     TLSv1.3  353 Application Data
114 2.104550448   192.168.0.7     192.168.0.6     TLSv1.3  535 Application Data
115 2.105250216   192.168.0.6     192.168.0.7     TLSv1.3  268 Application Data
187 3.761495397   192.168.0.7     34.117.65.55    TLSv1.2   93 Application Data
188 3.816191109   34.117.65.55    192.168.0.7     TLSv1.2   93 Application Data
192 4.762423771   192.168.0.7     34.149.100.209  TLSv1.2   93 Application Data
193 4.791868014   34.149.100.209  192.168.0.7     TLSv1.2   93 Application Data
```

9.0 OPENSSH SERVER:

install on both pc.

sudo apt install openssh-server

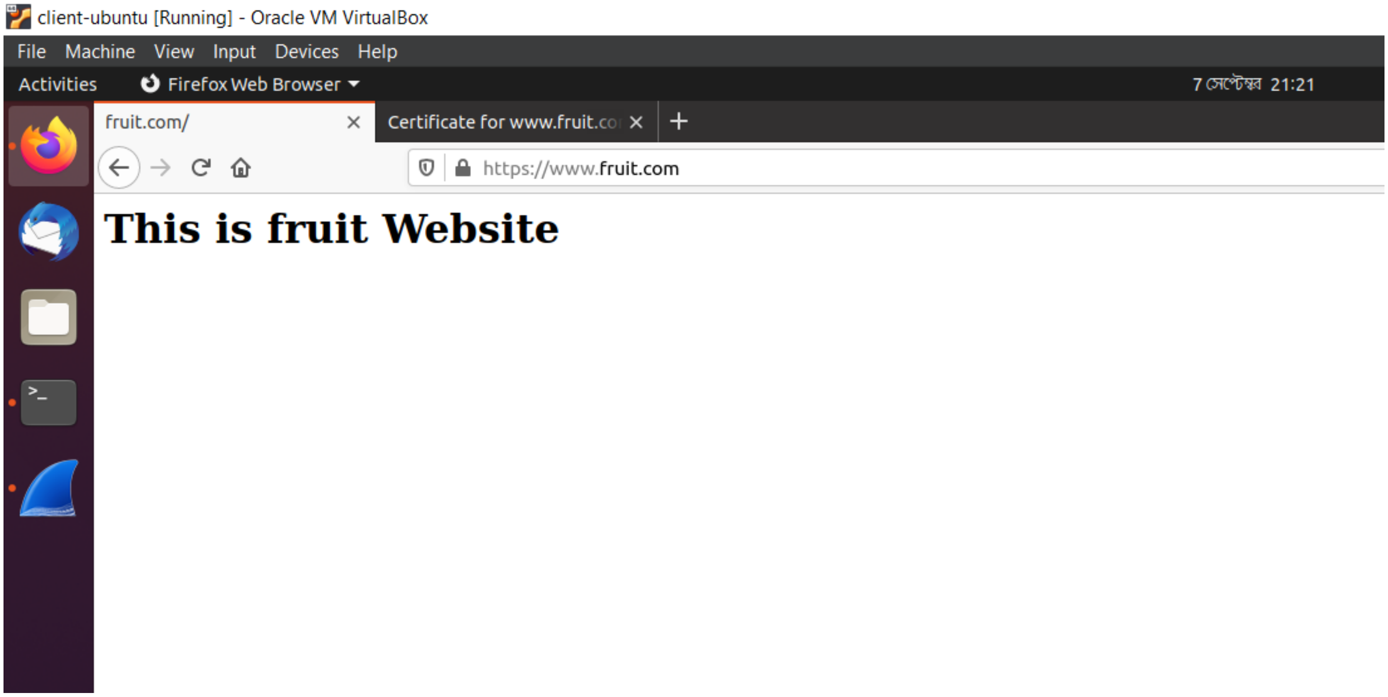
now from client pc type:

ssh server@192.168.0.7

If Firewall configured correctly in server, then using this command you can access the server pc from client pc.

10.0 Client PC VIEW:

Copy the Root Certificate file and paste the file in Client PC. Now Install the certificate in Client pc browser and after paste the certificate, the padlock icon should appear in the Client pc too.



client-ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
7 সেপ্টেম্বর 21:21
fruit.com/
Certificate for www.fruit.co

https://www.fruit.com

This is fruit Website

View From Client Website