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

CSE487: Cybersecurity, Law, and Ethics

Submitted To-

Rashedul Amin Tuhin
Senior Lecturer
Department of Computer Science and Engineering
East West University

Submitted By-

Members	ID
Md.Asad Chowdhury Dipu	2019-1-60-093
Md. Mizanur Rahman Riad Khan	2019-1-60-094
Kazi Mostaq Hridoy	2019-1-60-098

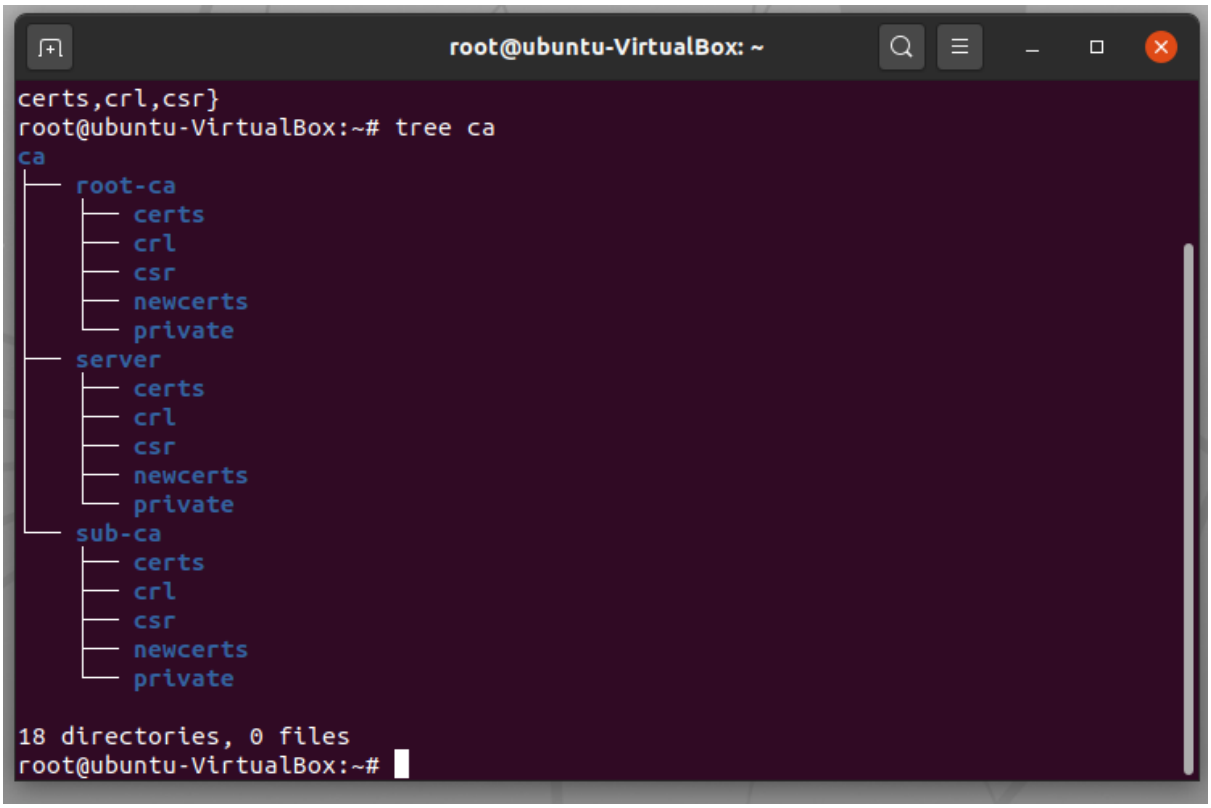
On linux terminal inside VirtualBox the following commands need to be given to generate Transport Layer Security over HTTP:

1. Preparing the environment 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}**
   -

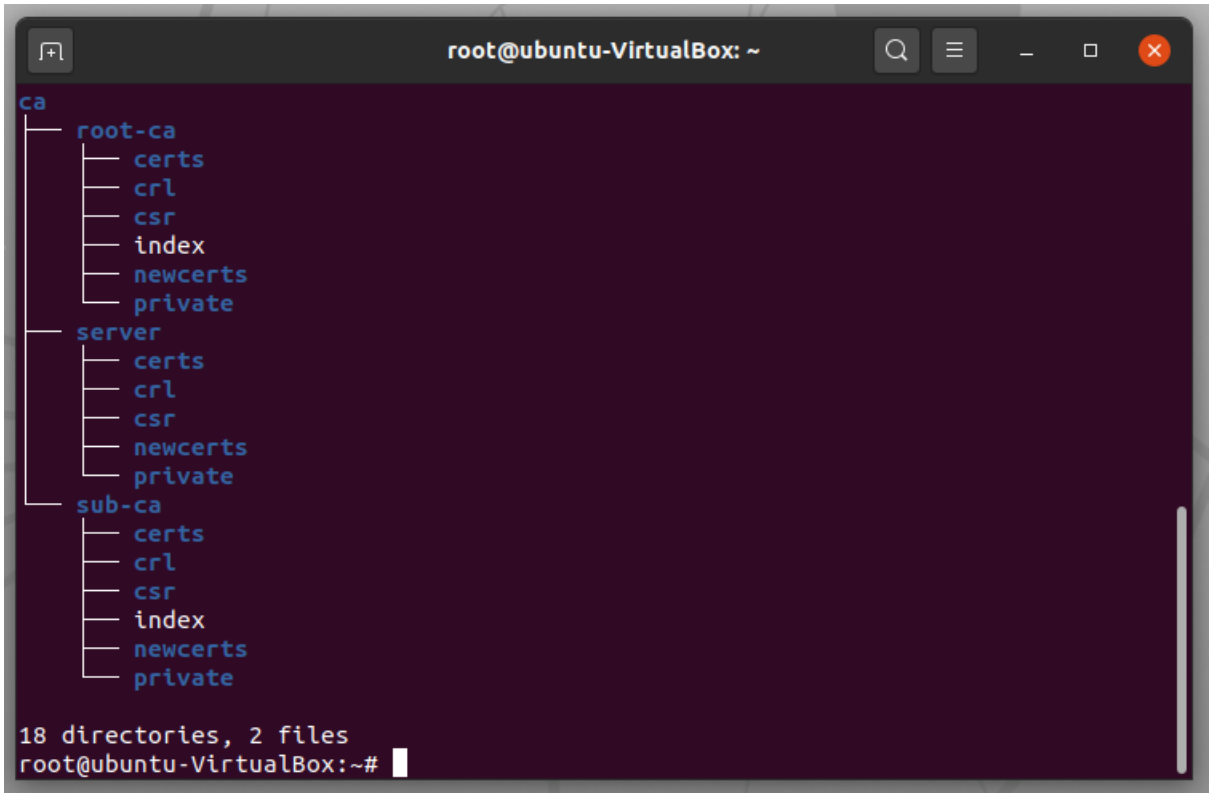
See if the folders are created successfully:
   - **tree ca**



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**

Seeing ca tree again
   - **tree ca**



Generating hexadecimal random number of 16 character

- **openssl rand -hex 16**

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**
- **tree ca**

```
root@ubuntu-VirtualBox: ~
root@ubuntu-VirtualBox:~# openssl rand -hex 16 > ca/sub-ca/serial
root@ubuntu-VirtualBox:~# tree ca
ca
├── root-ca
│   ├── certs
│   ├── crl
│   ├── csr
│   ├── index
│   ├── newcerts
│   ├── private
│   └── serial
├── server
│   ├── certs
│   ├── crl
│   ├── csr
│   ├── newcerts
│   └── private
└── sub-ca
    ├── certs
    ├── crl
    ├── csr
    ├── index
    ├── newcerts
    ├── private
    └── serial

18 directories, 4 files
root@ubuntu-VirtualBox:~#
```

moving to ca directory

- **cd ca**

2. Generating private key for root ca, sub ca, and server

a) **Public key for rootCA**

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

```
root@ubuntu-VirtualBox:~/ca# openssl genrsa -aes256 -out root-ca/private/ca.key
4096
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:
```

b) **Public key for subCA**

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

```
root@ubuntu-VirtualBox:~/ca# openssl genrsa -aes256 -out sub-ca/private/sub-ca.k
ey 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@ubuntu-VirtualBox:~/ca#
```

c) **Public key for server**

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

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

3. Generating certificates

a) **Root-CA**

 # Creating root ca.config

- **vim root-ca/root-ca.conf**

*******Code to be used-**

[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

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
localityName_default = Demra
0.organizationName_default = EWU
organizationalUnitName_default = Cyber_Security
commonName_default = AcmeRootCA
emailAddress_default = riad@acmesub_ca.com

[ 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
[ 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
```

[save and exit] (:wq then ctrl+c)

Moving inside root-ca

- **cd root-ca**

Generating root ca certificate

- openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 7305 -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**

Sub-CA

- **Creating sub-ca.config**
    - **vim sub-ca.conf**

*******Code to be used-**

```
[ca]

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

#see man ca

default_ca    = CA_default

[CA_default]

dir     = /root/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

stateOrProvinceName_default = Dhaka

localityName_default = Demra

0.organizationName_default = EWU

organizationalUnitName_default = Cyber_Security
```

```
commonName_default = AcmeRootCA

emailAddress_default = riad@acmeroot_ca.com


[ 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

[ 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

[save and exit] (:wq then ctrl+c)

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 3652 -notext -in ../sub-ca/csr/sub-ca.csr -out ../sub-ca/certs/sub-ca.crt**

Now if we see directory Tree

→we can see a .pem file has been generated

```
root@ubuntu-VirtualBox:~/ca/root-ca# tree
.
├── certs
│   └── ca.crt
├── crl
├── csr
├── index
├── index.attr
├── index.old
├── newcerts
│   └── A45E78CB8551A04F1E2ADA54A7031893.pem
├── private
│   └── ca.key
├── root-ca.conf
├── serial
└── serial.old

5 directories, 9 files
root@ubuntu-VirtualBox:~/ca/root-ca#
```

We can see the signing

- **cat index**

  →Root ca signed sub ca

We can see the detail by

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

4. Configuring server

Moving to server

- **cd ../server**

Generating certificate signing request from server

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

```
root@ubuntu-VirtualBox:~/ca/server# openssl req -key private/server.key -new -sh
a256 -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]:Dhaka
Locality Name (eg, city) []:Demra
Organization Name (eg, company) [Internet Widgits Pty Ltd]:EWU
Organizational Unit Name (eg, section) []:Cyber_Security
Common Name (e.g. server FQDN or YOUR name) []:www.verysecureserver.com
Email Address []:server@riad.com

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

moving to sub ca to sign the server's certificate

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

Sub ca signing certificate request of server

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

moving to certs folder to see certificate of server

- **cd ../server/certs/**

We can see the directory by using the command:

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

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**

```
root@ubuntu-VirtualBox:~/ca/server/certs# ls
chained.crt  server.crt
root@ubuntu-VirtualBox:~/ca/server/certs#
```

moving back to server directory

- **cd ..**

echo "127.0.0.2 www.verysecureserver.com" >> /etc/hosts

ping www.verysecureserver.com

```
root@ubuntu-VirtualBox:~/ca/server# echo "127.0.0.2 www.verysecureserver.com" >>
 /etc/hosts
root@ubuntu-VirtualBox:~/ca/server# ping www.verysecureserver.com
PING www.verysecureserver.com (127.0.0.2) 56(84) bytes of data.
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=1 ttl=64 time=0.024
 ms
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=2 ttl=64 time=0.054
 ms
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=3 ttl=64 time=0.024
 ms
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=4 ttl=64 time=0.030
 ms
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=5 ttl=64 time=0.028
 ms
64 bytes from www.verysecureserver.com (127.0.0.2): icmp_seq=6 ttl=64 time=0.028
 ms
^C
--- www.verysecureserver.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5122ms
rtt min/avg/max/mdev = 0.024/0.031/0.054/0.010 ms
root@ubuntu-VirtualBox:~/ca/server#
```

Turning on the ssl port

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

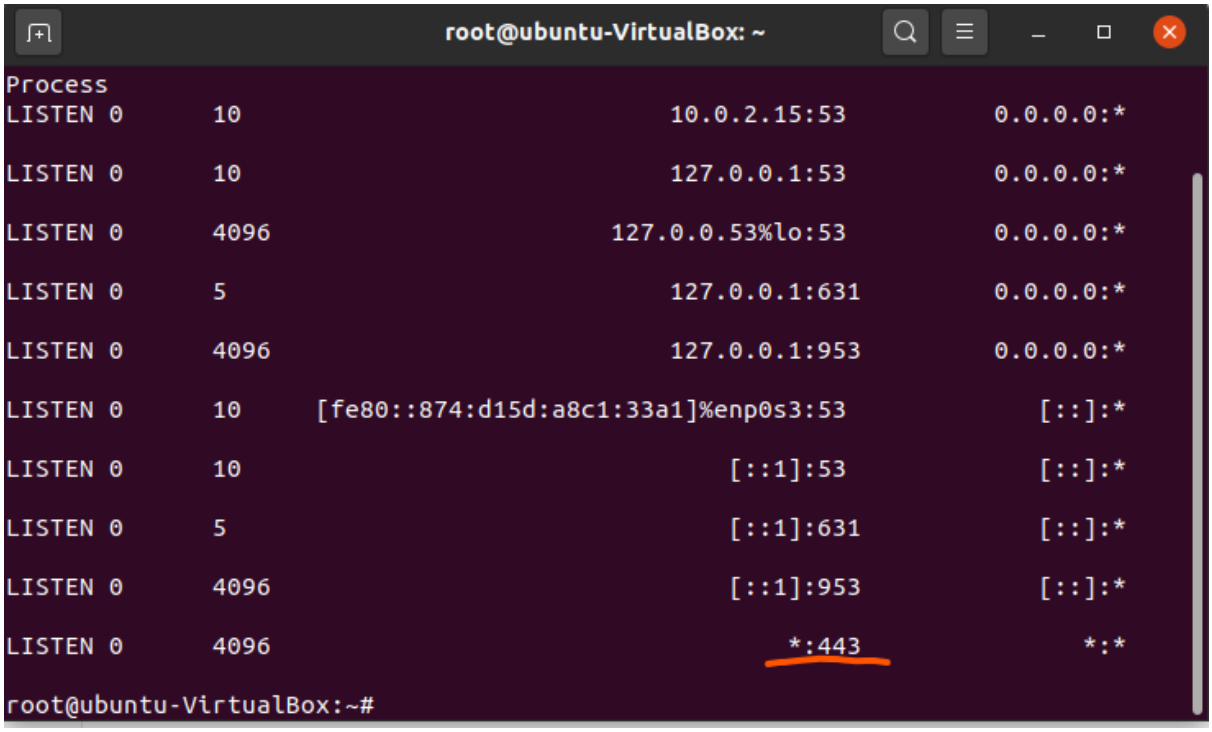
→Opening new terminal

Again to root

- **sudo -i**

See the port number used by different Ip addresses

- **ss -ntl**



- **sudo apt update**

to download or transfer files/data from or to a server using FTP, HTTP, HTTPS, SCP, SFTP, SMB, and other supported protocols, installing curl:

- **sudo apt install curl**

copying the certificate to ca certificate folder

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

Updating ca certificate folder

- **update-ca-certificates -v**

Exit_new_terminal

Now we have to install xampp and follow the following proceedure-
At first Download and install xampp-> https://www.apachefriends.org/download.html
----
[In download folder, edit file name xampp.run then open terminal here]

$ sudo -s
 # sudo chmod a+rwx xampp.run
 # ./xampp.run

[N.B: If you have apache already, remove it]

$ systemctl status apache2
$ sudo apt-get purge apache2 apache2-utils apache2.2-bin apache2-common
$ sudo apt-get autoremove
$ systemctl status apache2

TO START XAMPP

$ sudo -i
 # cd /opt/lampp
 # chmod a+rwx manager-linux-x64.run
 # ./manager-linux-x64.run

Next go to this location from your linux host
----
other Location/Computer/opt/lampp/etc/extra

[open terminal here]
$ sudo su
 # chmod 777 httpd-ssl.conf

line 106
----
change server.crt location with your server.crt file location

{106 SSLCertificateFile "/home/riad/certificate/server.crt"}

line 116
----
change server.key location with your server.key file location
{116 SSLCertificateKeyFile "/home/riad/certificate/server.key"}

line 136
----
change full line with your location
{136 SSLCACertificatePath "/home/riad/certificate"}

Now we have to remove all file from htdocs

[open new terminal]

```
$ sudo -i
# cd /opt/lampp/htdocs
# ls
# rm -r dashboard img webalizer
# rm applications.html  bitnami.css  favicon.ico  index.php
```

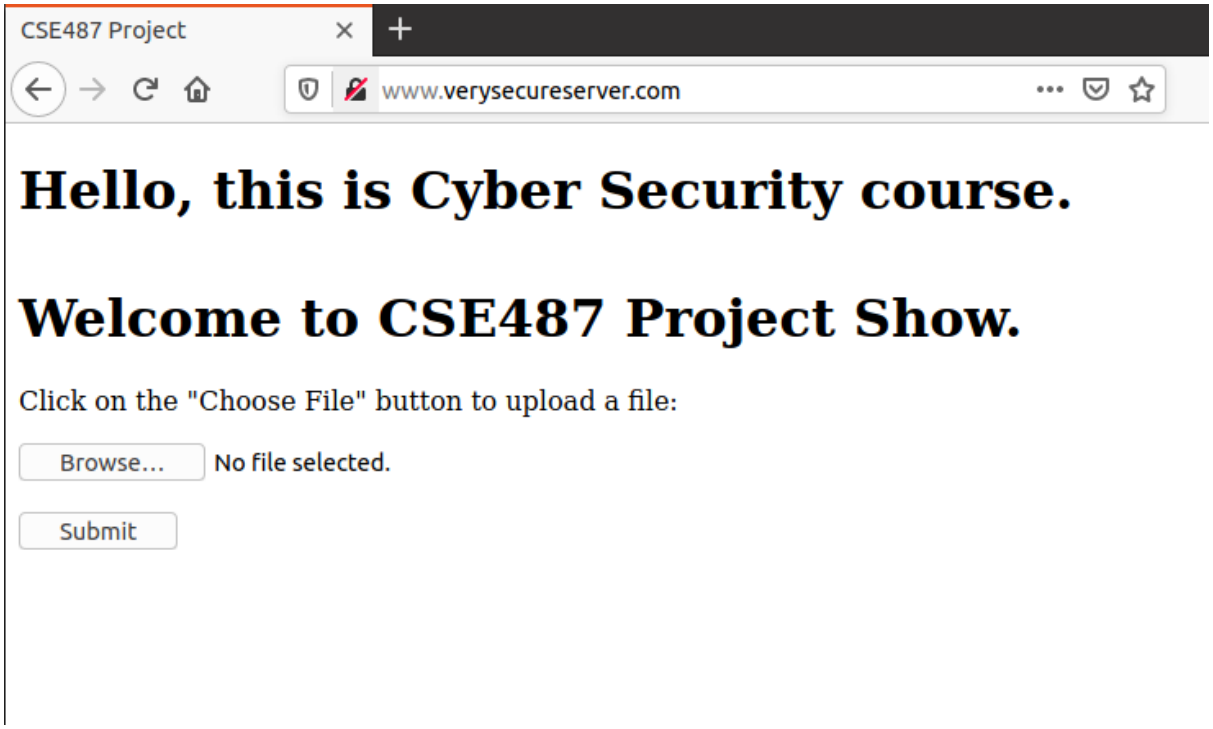
[Now make a html file and write some html code]

```
# touch index.html
# gedit index.html
```

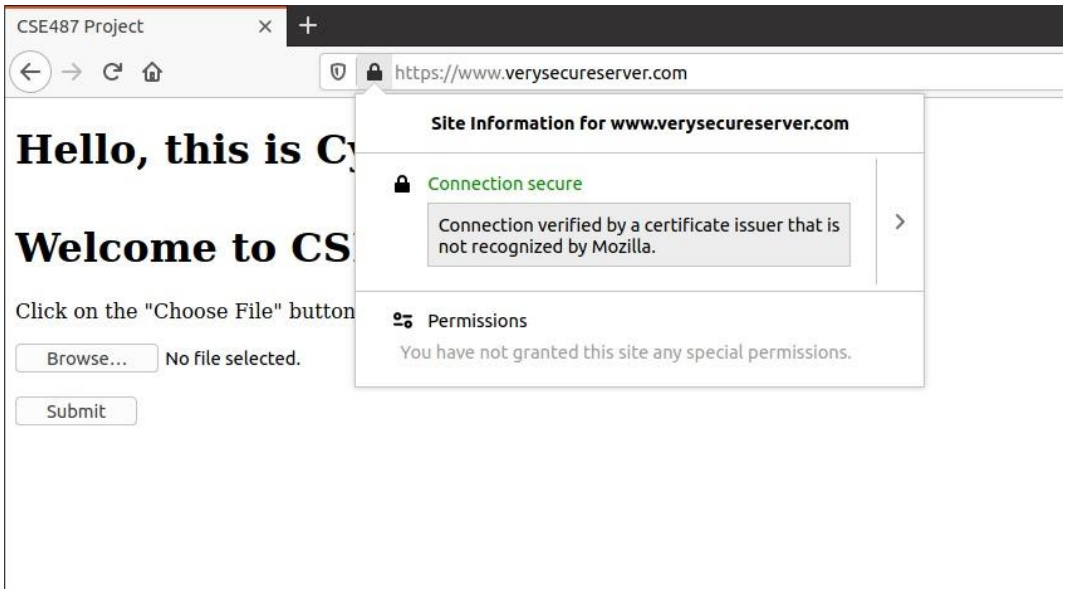
save and exit.

Before installation certificate from localhost

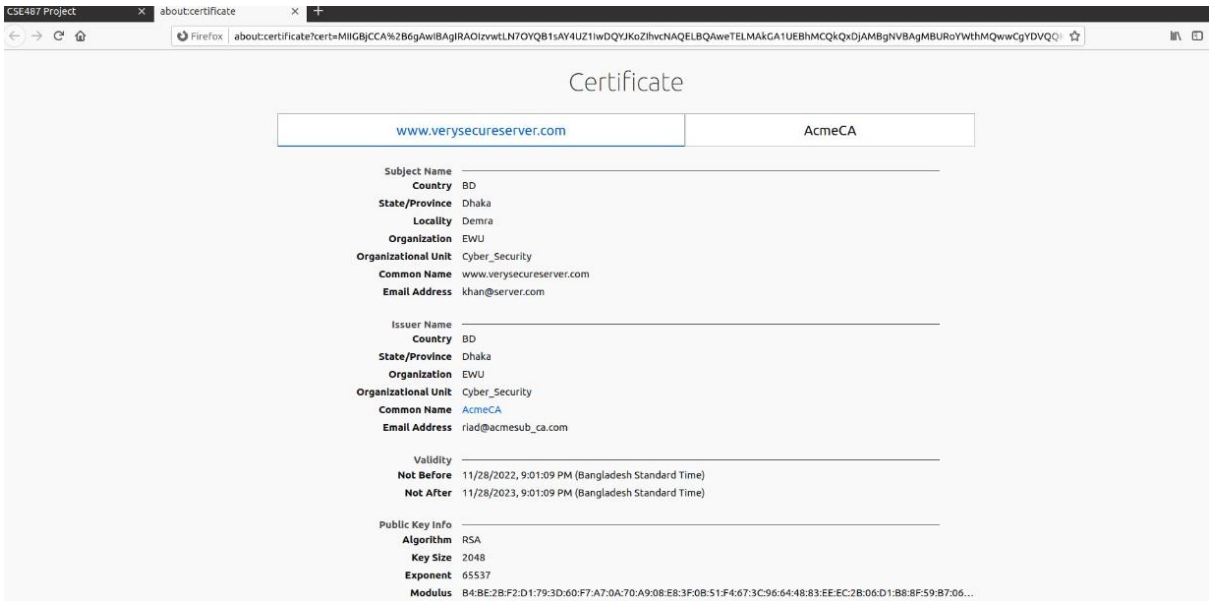
www.verysecureserver.com



After install certificate

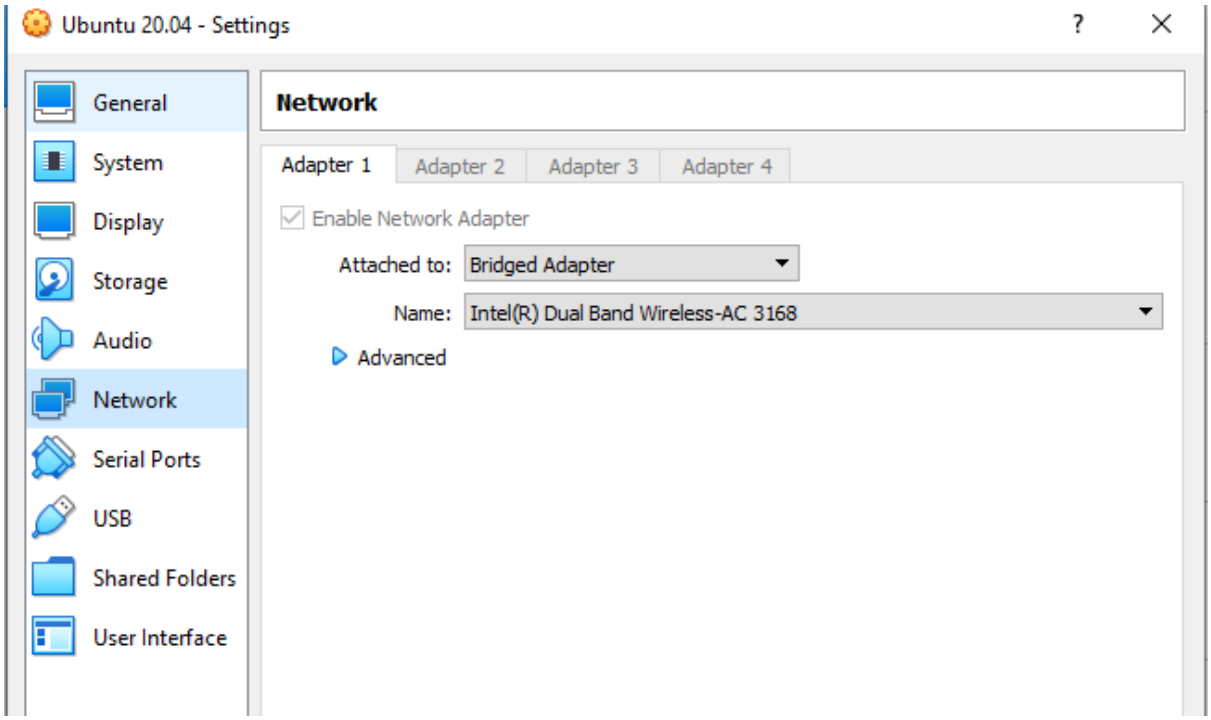


Certificate



DNS Configuration-

##CHANGE VIRTUAL BOX NETWORK TO BRIDGE ADAPTER



1. **At first, check your ip address.**

   Command: ip addr

   Here, you get your ip and the default ip.

2. **Check if bind9 is installed**

   Command: named -v

```
root@hridoy-VirtualBox:~# named -v
BIND 9.16.1-Ubuntu (Stable Release) <id:d497c32>
```

3. **Check status of the machine**

```
root@hridoy-VirtualBox:~# hostnamectl status
   Static hostname: hridoy-VirtualBox
         Icon name: computer-vm
           Chassis: vm
        Machine ID: e9d7932b07cb4a68b01a5ec201695119
           Boot ID: 00c6e0a918b14ec6bbdce3b2d4bba692
    Virtualization: oracle
  Operating System: Ubuntu 20.04 LTS
            Kernel: Linux 5.15.0-56-generic
      Architecture: x86-64
```

4. **Use the hostname and the domain name to edit the hosts file:**

```
192.168.31.44   hridoy-virtualbox.verysecureserver.com hridoy-virtualbox

# 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
127.0.0.2 www.verysecureserver.com
```

5. **Verify hostname, dns domain name, and fully qualified domain name respectively:**

```
root@hridoy-VirtualBox:~# hostname
hridoy-VirtualBox
root@hridoy-VirtualBox:~# dnsdomainname
verysecureserver.com
root@hridoy-VirtualBox:~# hostname --fqdn
hridoy-virtualbox.verysecureserver.com
root@hridoy-VirtualBox:~#
```

6. **Configure named.conf.options**

A – make a copy of original sudo cp named.conf.optionsnamed.conf.options.orig

B – Edit named.conf.options file:

```
//============================================================================
// 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{ 192.168.31.44;};
allow-transfer {none;};
forwarders {

 192.168.31.1;
 };
```

***192.168.31.44 is the machine IP where you are going to configure your server.**

***192.168.31.1 is the default gateway for the LAN you created.**

7. **Make forward lookup zone and reverse lookup zone**

A- make a copy of

named.conf.local sudo cp named.conf.local named.conf.local.orig

B – edit named.conf.local

sudo gedit named.conf.local Here, create a forward lookup zone and a reverse lookup zone

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

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "verysecureserver.com" IN{
        type master;

        file "/etc/bind/db.verysecureserver.com";  //this is domain name
};

//reverse lookup zone
zone "31.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/db.31.168.192";   //this is reverse ip
};
```

C-check configuration:

named -checkconf

8. Make records for forward and reverse lookup zone database

A – copy db.local to db.mysecureserver.com (which you mentioned in named.conf.local)

sudo cp db.local db.mysecureserver.com

Edit db.mysecureserver.com:

After editing:

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.verysecureserver.com. root.verysecureserver.com. (
                              2                 ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.verysecureserver.com.
ns1     IN      A       192.168.31.44
www     IN      A       192.168.31.44
ftp     IN      A       192.168.31.44
@       IN      MX      10      mail
mail    in      A       192.168.31.44
@       IN      AAAA    ::1
```

B- copy db.127 to db.31.168.192 file(which you mentioned in named.conf.local in reverse lookup zone)

named-checkzone verysecureserver.com db.verysecureserver.com

```
root@hridoy-VirtualBox:/etc/bind# named-checkzone verysecureserver.com db.verysecureserver.com
zone verysecureserver.com/IN: loaded serial 2
OK
root@hridoy-VirtualBox:/etc/bind#
```

sudo cp db.127 db.31.168.192

Edit db.31.168.192

sudo gedit db.31.168.192

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

```
root@hridoy-VirtualBox:/etc/bind# named-checkzone verysecureserver.com db.verysecureserver.com
zone verysecureserver.com/IN: loaded serial 2
OK
root@hridoy-VirtualBox:/etc/bind# cp db.127 db.31.168.192
root@hridoy-VirtualBox:/etc/bind# gedit db.31.168.192

(gedit:3922): Tepl-WARNING **: 09:13:05.773: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly ins
talled or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
root@hridoy-VirtualBox:/etc/bind# named-checkzone 31.168.192.in-addr.arpa db.31.168.192
zone 31.168.192.in-addr.arpa/IN: loaded serial 1
OK
root@hridoy-VirtualBox:/etc/bind# named-checkconf
root@hridoy-VirtualBox:/etc/bind# named-checkzone verysecureserver.com db.verysecureserver.com
zone verysecureserver.com/IN: loaded serial 2
OK
root@hridoy-VirtualBox:/etc/bind# named-checkzone 31.168.192.in-addr.arpa db.31.168.192
zone 31.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

9. **Restart bind9 and check status**

sudo service bind9 restart

sudo service bind9 status

```
root@hridoy-VirtualBox:/etc/bind# service bind9 restart
root@hridoy-VirtualBox:/etc/bind# service bind9 status
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-12-15 09:16:06 EST; 8s ago
       Docs: man:named(8)
   Main PID: 3965 (named)
      Tasks: 8 (limit: 2288)
     Memory: 16.1M
     CGroup: /system.slice/named.service
             └─3965 /usr/sbin/named -f -u bind

Dec 15 09:16:09 hridoy-VirtualBox named[3965]: no valid RRSIG resolving './NS/IN': 192.36.148.17#53
Dec 15 09:16:09 hridoy-VirtualBox named[3965]: validating ./NS: no valid signature found
Dec 15 09:16:09 hridoy-VirtualBox named[3965]: no valid RRSIG resolving './NS/IN': 202.12.27.33#53
Dec 15 09:16:10 hridoy-VirtualBox named[3965]: validating ./NS: no valid signature found
Dec 15 09:16:10 hridoy-VirtualBox named[3965]: no valid RRSIG resolving './NS/IN': 199.7.83.42#53
Dec 15 09:16:10 hridoy-VirtualBox named[3965]: validating ./NS: no valid signature found
Dec 15 09:16:10 hridoy-VirtualBox named[3965]: no valid RRSIG resolving './NS/IN': 198.41.0.4#53
Dec 15 09:16:12 hridoy-VirtualBox named[3965]: validating ./NS: no valid signature found
Dec 15 09:16:12 hridoy-VirtualBox named[3965]: no valid RRSIG resolving './NS/IN': 198.97.190.53#53
Dec 15 09:16:12 hridoy-VirtualBox named[3965]: resolver priming query complete
root@hridoy-VirtualBox:/etc/bind#
```

10. A- delete resolv.conf

sudo rm /etc/resolv.conf

B- link resolv.conf

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

C- edit resolv.conf

sudo gedit /etc/resolv.conf

```
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.31.44
search verysecureserver.com
```

Then configure your DNS from other os

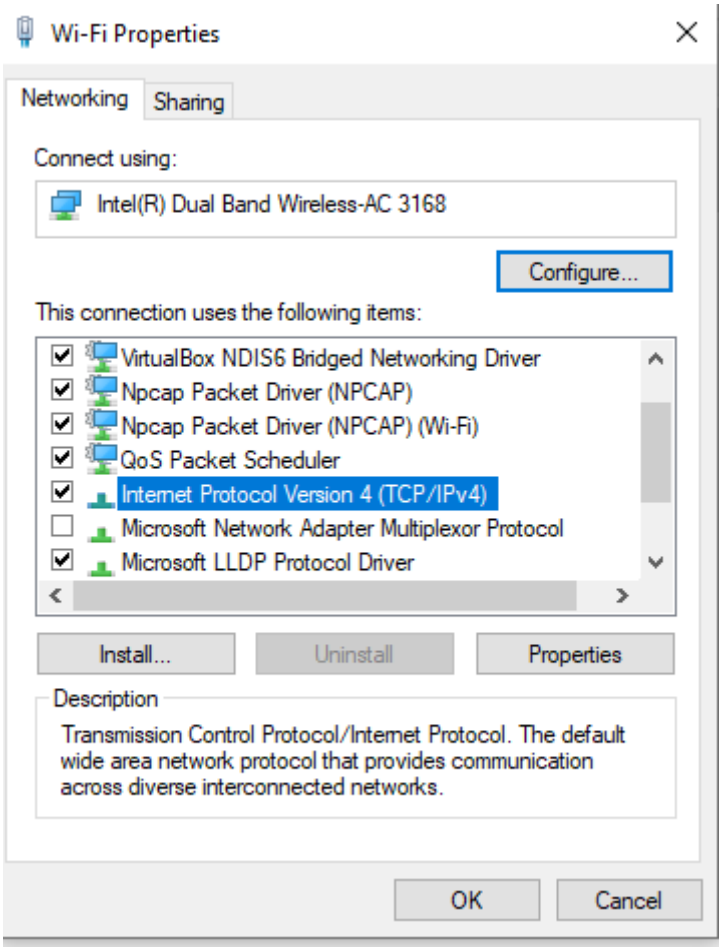
example if windows host....

->Control Panel\Network and Internet\Network and Sharing Center

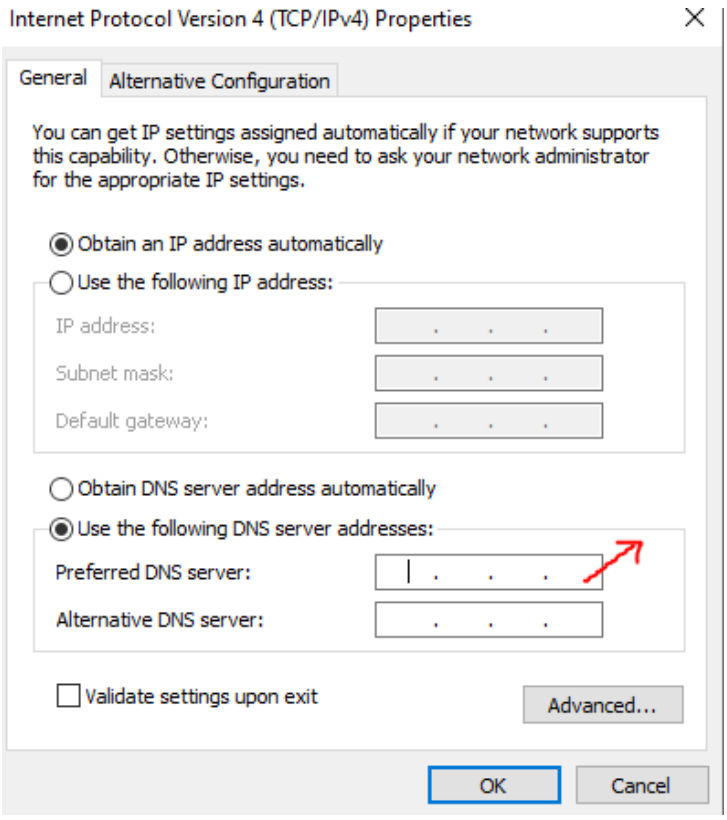
->Change adapter setting

->Select your internet connection and right click and go to properties

->Double click TCP/IPv4



>Change DNS with nameserver ip (192.168.31.44)



Firewall Configuration:

1.Install ufw package

sudo apt install ufw

2. Set default rules for ufw firewall

ufw default allow outgoing

ufw default deny incoming

3. Enable ssh

ufw allow ssh

4. Enable ufw

ufw enable

5. Allow port 80 (http), 443(https), and 53(DNS)

ufw allow 80

ufw allow 443

ufw allow 53