EAST WEST UNIVERSITY

Project Report

Course Title: **Cyber Security, Law, and Ethics**

Course Code: **CSE487**

Section: **2**

Group: 12

Semester: **Spring 2024**

Course Instructor: Dr. Md. Hasanul Ferdaus

Project name:

Securing a networked system with Public Key Infrastructure

Group Information

Name	ID
Fahima Afrin Nidha	2020-2-60-175
Md. Tanvir Islam Mahin	2020-2-60-075
Md. Abu Salha Akram Rayhan	2020-1-60-020
Sifat Khan Shishir	2020-2-60-120

Configuration of Certification Authority AcmeCA with AcmeRootCA as the RootCA:

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

Changing the root of ca and sub ca private folder

---

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

Creating file index in both root ca and sub ca

---

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

Generating hexadecimal random number of 16 charecter

---

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

moving to ca directory

---

cd ca

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

Public key for rootCA

---

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

Public key for subCA

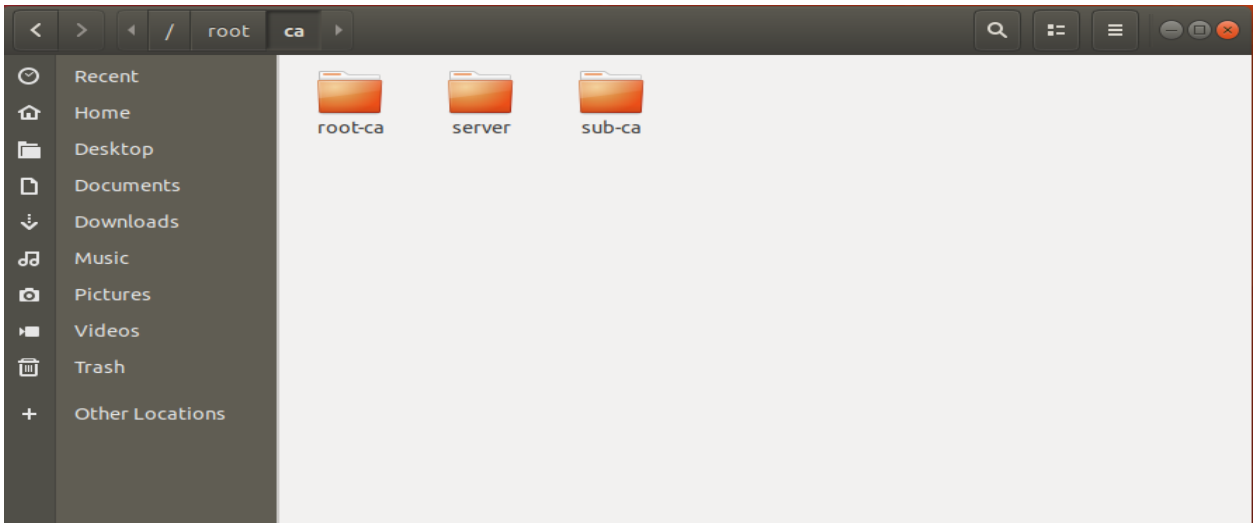
---

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

Public key for server

---

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



3. Generating certificates

Root-CA

Creating root ca.config

---

gedit root-ca/root-ca.conf

[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 = Aftabnagar

0.organizationName_default = EWU

organizationalUnitName_default = Cyber_Security

commonName_default = AcmeRootCA

emailAddress_default = cyber@securityroot_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

```
organizationalUnitName          = Organizational Unit Name

commonName                      = Common Name

emailAddress                    = Email Address

countryName_default  = BD

stateOrProvinceName_default = Dhaka

localityName_default = Aftabnagar

0.organizationName_default = EWU

organizationalUnitName_default = Cyber_Security

commonName_default = AcmeRootCA

emailAddress_default = cyber@securityroot_ca.com
```

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

cd ../sub-ca

Sub-CA

Creating sub-ca.config

---

gedit sub-ca.conf

[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 = Aftabnagar

0.organizationName_default = EWU

organizationalUnitName_default = Cyber_Security

commonName_default = AcmeCA

emailAddress_default = cyber@securitysub_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

```
countryName_default  = BD
stateOrProvinceName_default = Dhaka
localityName_default = Aftabnagar
0.organizationName_default = EWU
organizationalUnitName_default = Cyber_Security
commonName_default  = AcmeCA
emailAddress_default = cyber@securitysub_ca.com
```

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

Signing the request of sub ca by root ca

---

cd ..

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

---

Proof of certificate and pem file creation

```
root@naf-VirtualBox:~/ca/sub-ca# ls
certs  csr    index.attr  newcerts  serial      sub-ca.conf
crl    index  index.old   private   serial.old
root@naf-VirtualBox:~/ca/sub-ca# cd certs
root@naf-VirtualBox:~/ca/sub-ca/certs# ls
sub-ca.crt
root@naf-VirtualBox:~/ca/sub-ca/certs# cd ..
root@naf-VirtualBox:~/ca/sub-ca# cd newcerts
root@naf-VirtualBox:~/ca/sub-ca/newcerts# ls
8149874F6A19B1E4811F63FE9BF4DE71.pem
```

- Transferring the certificate from CA to www.verysecureserver.com:
- Installation of the signed SSL certificate in the server of www.cybersecurity.com:
- Making the system trust RootCA:

after installation

cd /opt/lampp/etc/extra

```
104 #   require an ECC certificate which can also be configured in
105 #   parallel.
106 SSLCertificateFile "/home/naf/certificate/server.crt"
107 #SSLCertificateFile "/opt/lampp/etc/server-dsa.crt"
108 #SSLCertificateFile "/opt/lampp/etc/server-ecc.crt"
109
110 #   Server Private Key:
111 #   If the key is not combined with the certificate, use this
112 #   directive to point at the key file.  Keep in mind that if
113 #   you've both a RSA and a DSA private key you can configure
114 #   both in parallel (to also allow the use of DSA ciphers, etc.)
115 #   ECC keys, when in use, can also be configured in parallel
116 SSLCertificateKeyFile "/home/naf/certificate/server.key"
117 #SSLCertificateKeyFile "/opt/lampp/etc/server-dsa.key"
118 #SSLCertificateKeyFile "/opt/lampp/etc/server-ecc.key"
119
120 #   Server Certificate Chain:
121 #   Point SSLCertificateChainFile at a file containing the
122 #   concatenation of PEM encoded CA certificates which form the
123 #   certificate chain for the server certificate. Alternatively
124 #   the referenced file can be the same as SSLCertificateFile
125 #   when the CA certificates are directly appended to the server
126 #   certificate for convenience.
127 #SSLCertificateChainFile "/opt/lampp/etc/server-ca.crt"
128
129 #   Certificate Authority (CA):
130 #   Set the CA certificate verification path where to find CA
131 #   certificates for client authentication or alternatively one
132 #   huge file containing all of them (file must be PEM encoded)
133 #   Note: Inside SSLCACertificatePath you need hash symlinks
134 #         to point to the certificate files. Use the provided
135 #         Makefile to update the hash symlinks after changes.
136 SSLCACertificatePath "/home/naf/certificate"
137 #SSLCACertificateFile "/opt/lampp/etc/ssl.crt/ca-bundle.crt"
```

sudo gedit httpd-ssl.conf

/home/naf/certificate/server.crt

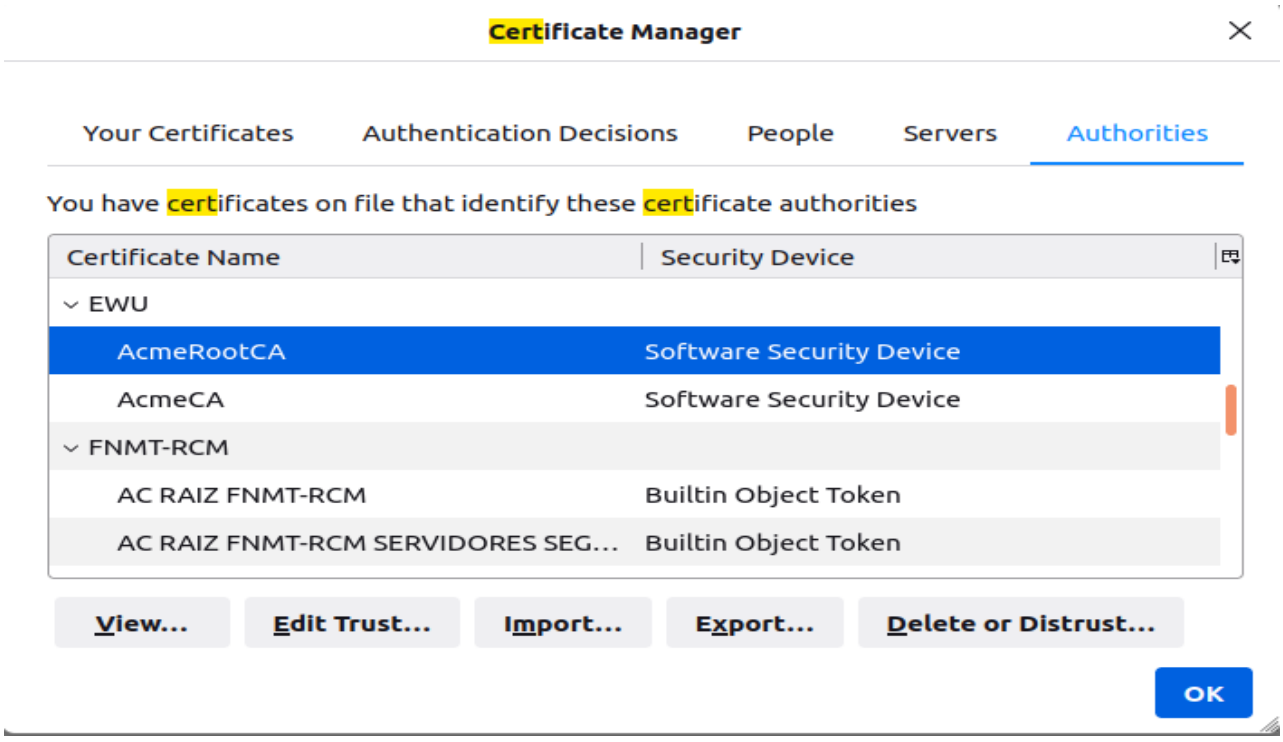
cd /etc

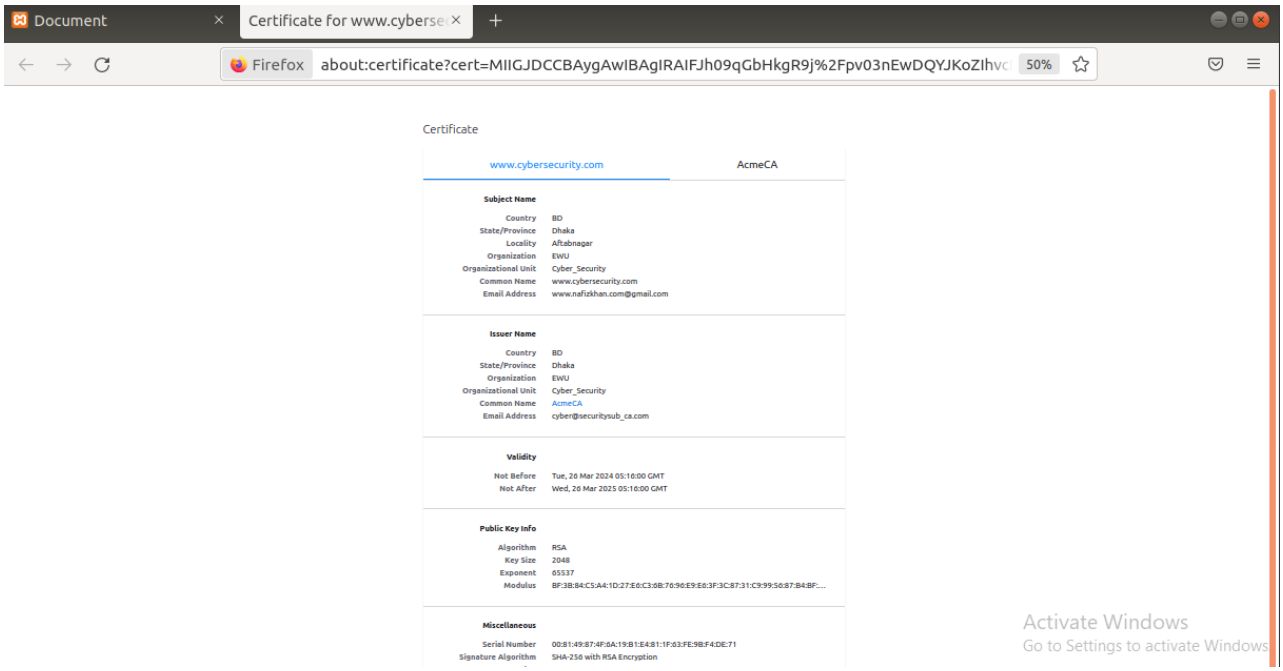
sudo gedit hosts

add website name

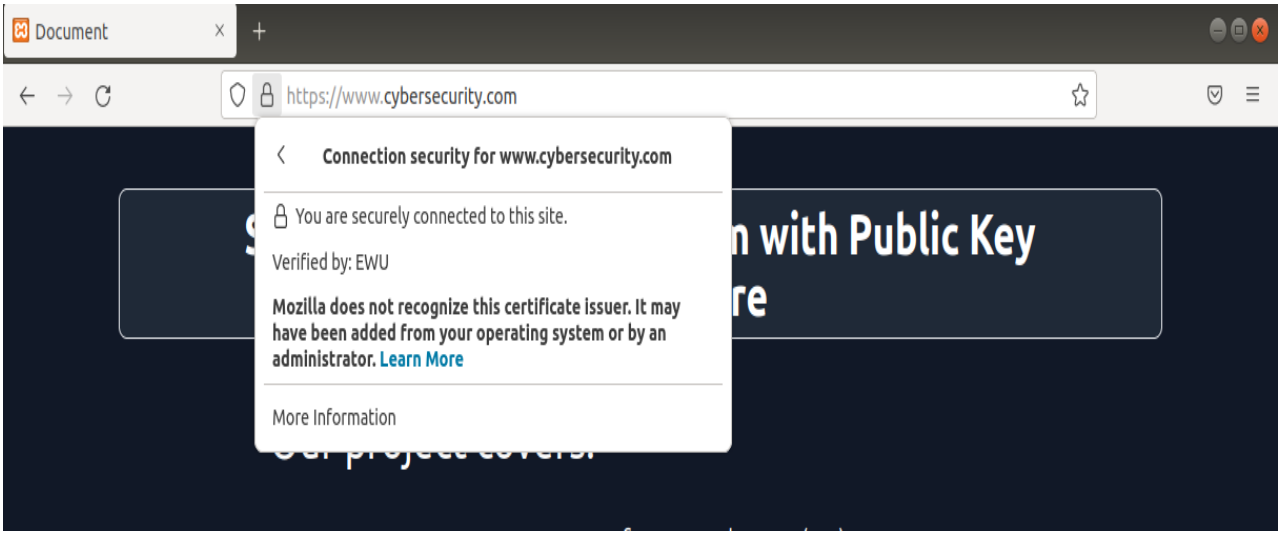
www.cybersecurity.com

Importing the certificates into mozilla firefox



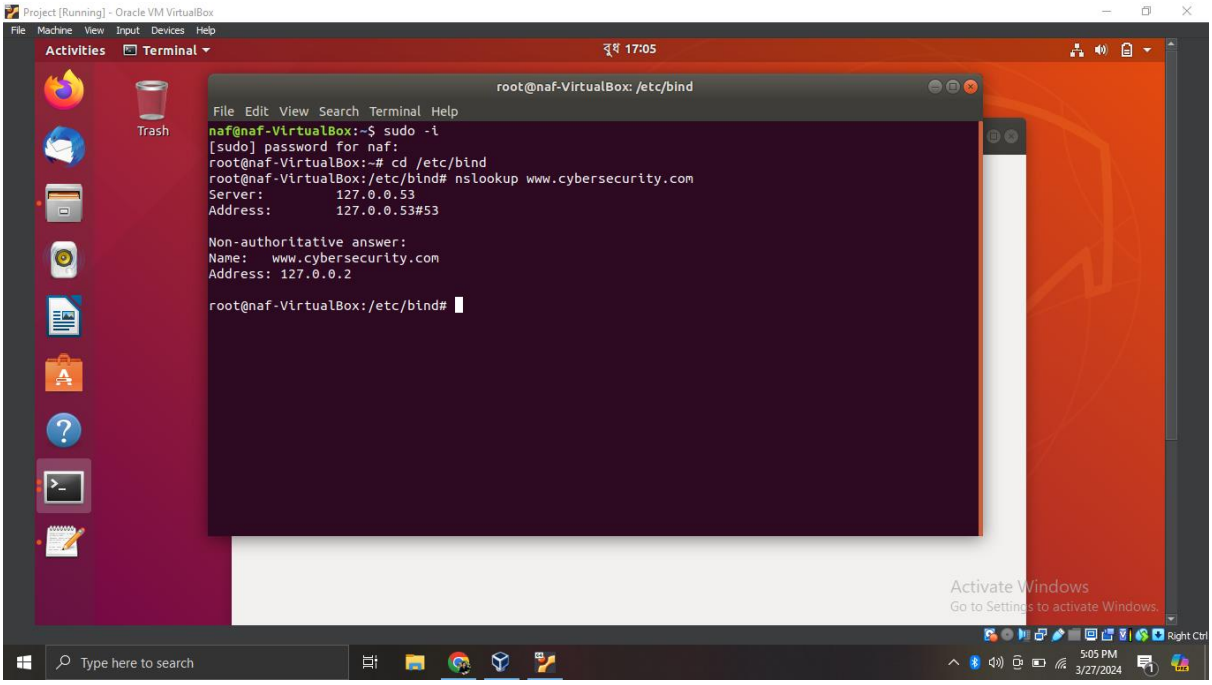


Running the website:

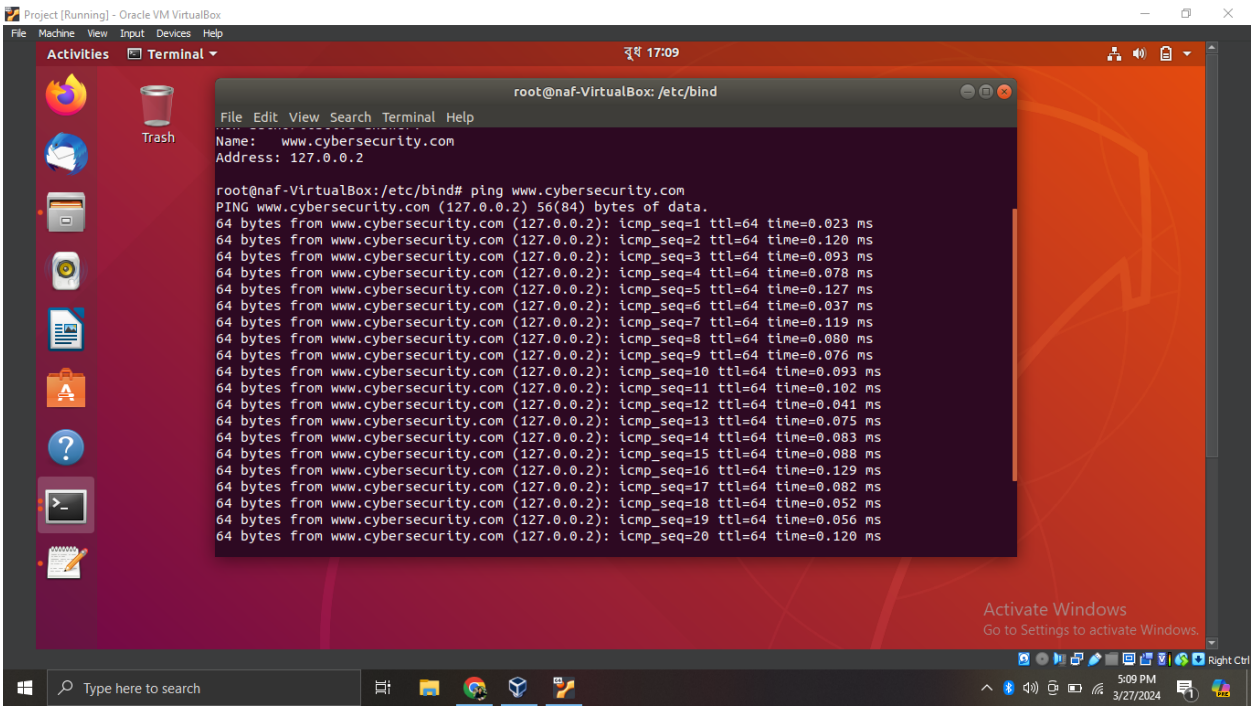


Configured DNS Server:

nslookup www.cybersecurity.com



ping www.cybersecurity.com



Conclusion:

In this project the PKI was implemented using OpenSSL and the DNS server was also configured. PKI helps the users to safely browse the internet and exchange data in encrypted format.