Submitted To:

Rashedul Amin Tuhin

Senior Lecturer

Department of Computer Science and Engineering

East West University

Submitted By:

Name	ID
Lamyea Tasneem Maha	2019-1-60-055
Noor Fabi Shah Safa	2019-1-60-060
Rafid Mahmud Haque	2019-1-60-085

Submission date:

25 August 2022

Certificate Generation:

(configuration files for root and sub-ca certificates can be found on the last page of the report)

1. Make directories

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

2. Change mode to private

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

3. Create index file

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

4. Generate 16 bithexcode for rootca and subca

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

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

5. Generate private key for rootca

root@maha-virtualbox:~/ca# opensslgenrsa -aes256 -out root-ca/private/ca.key 4096

*Enter pass phrase

6. Generate private key for sub ca

root@maha-virtualbox:~/ca# opensslgenrsa -aes256 -out sub-ca/private/sub-ca.key 4096

*Enter pass phrase

7. Generate private key for server

root@maha-virtualbox:~/ca# opensslgenrsa -out server/private/server.key 2048

8. Create config file for rootca

root@maha-virtualbox:~/ca# vim root-ca/root-ca.conf

9. Create ca certificate using root ca config file and private key

root@maha-virtualbox:~/ca/root-ca# openssl req -config root-ca.conf -key private/ca.key -new -x509 -days 7500 -sha256 -extensions v3_ca -out certs/ca.crt

*Enter organization name, unit name, and common name.

Check if the certificate is created or not:

root@maha-virtualbox:~/ca/root-ca# openssl x509 -noout -in certs/ca.crt -text

*We will seeit is self signed, the issuer and the subject are the same.

10. Create configfile for sub-ca

root@maha-virtualbox:~/ca/sub-ca# vim sub-ca.conf

11. Generate signing request using sub ca config file and sub ca private key

root@maha-virtualbox:~/ca/sub-ca# openssl req -config sub-ca.conf -new -key private/sub-ca.key -sha256 -out csr/sub-ca.csr

12. Sign sub ca certificate using the root ca certificate

root@maha-virtualbox:~/ca/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

Check if the certificate is signed:

root@maha-virtualbox:~/ca/root-ca# cat index

root@maha-virtualbox:~/ca/root-ca# openssl x509 -nout -text -in ../sub-ca/certs/sub-ca.crt

*Here we will see signed by root ca and subject is sub ca

13. Create server certificate

root@maha-virtualbox:~/ca/server# openssl req -key private/server.key -new -sha256 -out csr/server.csr

Enter a common name. Example: www.mysecureserver.com

14. Sign server certificate using sub-ca

root@maha-virtualbox:~/ca/sub-ca# openssl ca -config sub-ca.conf -extensions server_cert -days 365 -notext -in ../server/csr/server.csr -out ../server/certs/server.crt

15. Map 127.0.0.2 to our website

root@maha-virtualbox:~/ca/server# echo "127.0.0.2 www.mysecureserver.com" >> /etc/hosts

16. Turn on 443 port of our server

root@maha-virtualbox:~/ca/server# openssls_server -accept 443 -www -key private/server.key -cert certs/server.crt -CAfile ../sub-ca/certs/sub-ca.crt

17. In another terminal check:

curl https://www.mysecureserver.com

*You will see that you can not connect to the website securely.

You have to update the ca-certificates folder:

root@maha-virtualbox:~# cp ca/root-ca/certs/ca.crt /usr/local/share/ca-certificates/

root@maha-virtualbox:~# update-ca-certificates -v

18. Now check with curl:

root@maha-virtualbox:~/ca/server# curl https://www.mysecureserver.com

DNS Setup:

1.sudo apt install bind9

2. Check if bind9 is installed

named -v

```
BIND 9.11.3-1ubuntu1.17-Ubuntu (Extended Support Version) <id:a375815>
```

3. Check status of the machine

hostnamectl status

You can see the static hostname for your machine.

```
   Static hostname: lamyea22-VirtualBox
         Icon name: computer-vm
           Chassis: vm
        Machine ID: a80f860aa7e848eca0a6db5828bdf718
           Boot ID: 6539b01741644ca28f20221519f00b80
    Virtualization: oracle
  Operating System: Ubuntu 18.04.5 LTS
            Kernel: Linux 5.4.0-58-generic
      Architecture: x86-64
```

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

sudo vim /etc/hosts

```
192.168.43.69   lamyea22-VirtualBox.mysecureserver.com lamyea22-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
```

*192.168.43.69 is the machine IP in your LAN where your server is going to be.

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

hostname

dnsdomainname

hostname --fqdn

```
root@lamyea22-VirtualBox:/etc/bind# hostname
lamyea22-VirtualBox
root@lamyea22-VirtualBox:/etc/bind# dnsdomainname
mysecureserver.com
root@lamyea22-VirtualBox:/etc/bind# hostname --fqdn
lamyea22-VirtualBox.mysecureserver.com
root@lamyea22-VirtualBox:/etc/bind#
```

8. Configure named.conf.options

A – make a copy of original

sudo cp named.conf.optionsnamed.conf.options.orig

B – Edit named.conf.options file:

nano named.conf.options

```
GNU nano 2.9.3                    named.conf.options

        // forwarders {
        //      0.0.0.0;
        // };

        //============================================================$
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-ke$
        //============================================================$
        dnssec-validation auto;

        #auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        recursion yes;
        listen-on {192.168.43.69;};
        allow-transfer {none;};

        forwarders {
        192.168.43.1;
        };
};
```

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

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

9. 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";

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

//reverse lookup zone
zone "43.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/db.43.168.192";
};
```

C-check configuration:

named -checkconf

10. 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:

Open ▾ db.mysecureserver.com /etc/bind Save

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.mysecureserver.com. root.mysecureserver.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.mysecureserver.com.
ns1     IN      A       192.168.43.69
www     IN      A       192.168.43.69
ftp     IN      A       192.168.43.69
@       IN      MX      10      mail
mail    IN      A       192.168.43.69
@       IN      AAAA    ::1
```

B- copy db.127 to db.43.168.192 file(which you mentioned in named.conf.local in reverse lookup zone)

sudo cp db.127 db.43.168.192

Edit db.43.168.192

sudo gedit db.43.168.192

```
Open ▾                db.43.168.192                Save
                      /etc/bind
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.mysecureserver.com. root.mysecureserver.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.mysecureserver.com.
69      IN      PTR     ns1.mysecureserver.com.
69      IN      PTR     www.mysecureserver.com.
69      IN      PTR     ftp.mysecureserver.com.
69      IN      PTR     mail.mysecureserver.com.
```

12.Restart bind9 and check status

sudo service bind9 restart

sudo service bind9 status

```
root@lamyea22-VirtualBox:~# sudo service bind9 status
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: e
   Active: active (running) since Thu 2022-08-25 21:27:55 +06; 24min ago
     Docs: man:named(8)
 Main PID: 855 (named)
    Tasks: 5 (limit: 2033)
   CGroup: /system.slice/bind9.service
           └─855 /usr/sbin/named -f -u bind

আগ 25 21:28:00 lamyea22-VirtualBox named[855]: automatic empty zone: EMPTY.AS
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: none:103: 'max-cache-size 90%'
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: configuring command channel fr
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: configuring command channel fr
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: reloading configuration succee
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: scheduled loading new zones
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: managed-keys-zone: Unable to f
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: resolver priming query complet
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: any newly configured zones are
আগ 25 21:28:00 lamyea22-VirtualBox named[855]: running
lines 1-19/19 (END)
```

13.

A- delete resolv.conf

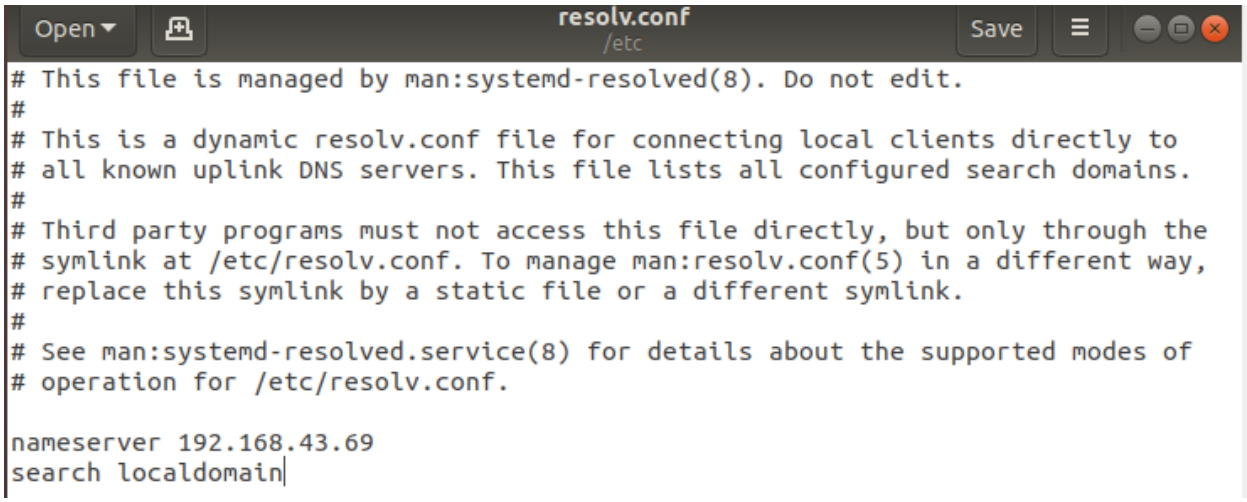
sudo rm /etc/resolv.conf

B- link resolv.conf

sudo ln –sf /run/systemd/resolve/resolv.conf /etc/resolv.conf

C- edit resolv.conf

sudo gedit /etc/resolv.conf



D- check if it can resolve now using nslookup

nslookup www.mysecureserver.com

```
root@lamyea22-VirtualBox:~# nslookup www.mysecureserver.com
Server:         192.168.43.69
Address:        192.168.43.69#53

Name:   www.mysecureserver.com
Address: 192.168.43.69
```

E- ping

ping www.mysecureserver.com

```
root@lamyea22-VirtualBox:~# ping www.mysecureserver.com
PING www.mysecureserver.com (192.168.43.69) 56(84) bytes of data.
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
1 ttl=64 time=0.061 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
2 ttl=64 time=0.048 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
3 ttl=64 time=0.056 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
4 ttl=64 time=0.087 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
5 ttl=64 time=0.056 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
6 ttl=64 time=0.150 ms
```

14.In 18.0.4 we need to editnsswitch.conf file to be able to ping: (Skip this step for 20.0.4 version)

sudo gedit /etc/nsswitch.conf

Before:

```
Open ▾                    nsswitch.conf                    Save
                          /etc
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat systemd
group:          compat systemd
shadow:         compat
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

After:

```
Open ▾                    *nsswitch.conf                    Save
                              /etc
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat systemd
group:          compat systemd
shadow:         compat
gshadow:        files

hosts:          files dns mdns4_minimal [NOTFOUND=return] myhostname
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Now try to ping with the ping command.

```
root@lamyea22-VirtualBox:~# ping www.mysecureserver.com
PING www.mysecureserver.com (192.168.43.69) 56(84) bytes of data.
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
1 ttl=64 time=0.061 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
2 ttl=64 time=0.048 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
3 ttl=64 time=0.056 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
4 ttl=64 time=0.087 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
5 ttl=64 time=0.056 ms
64 bytes from lamyea22-VirtualBox.mysecureserver.com (192.168.43.69): icmp_seq=
6 ttl=64 time=0.150 ms
```

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

*Ufw will now be active.

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

ufw allow 80

ufw allow 443

ufw allow 53

IDS Configuration:

1. **Installing snort:**
   #sudo apt-get install snort
   Give interface and IP:
   Usually interface is: enp0s3
   Give your host's ip: 192.168.something.something/24

2. **Go to snort directory**
   # cd /etc/snort
3. **Make a copy of snort.conf file**
   # cp snort.conftest_snort.conf
   We'll work with this test_snort.conf file

4. **Then open the test_snort.conf file**
   # sudogedittest_snort.conf
   Then go to line 51 and under "ipvar HOME_NET any" write
   ip var HOME_NET your host ip as follows:

```
49 # /etc/snort/snort.debian.conf configuration file
50 #
51 ipvar HOME_NET any
52 ipvar HOME_NET 192.168.56.0/24
53
```

   Then save it and close it.

5. **Now we'll create our own rule for TCP connection**
   Go to rules dir:
   # cd /etc/snort/rules
   Open local rules file
   # sudo nano local.rules
   Here we'll write the rule as follows:
   alert tcp any any -> $HOME_NET 80 (flags:S; msg: " DoS attack happening";
   flow:stateless; detection_filter: track by_dst,count 70. Seconds 10; sid: 100001;rev:1;)
   Save it and close

6. **Validate the conf file**
   # sudo snort –T –I enp0s3 –c /etc/snort/test_snort.conf

7. **Start snort:**
   # sudo snort –A console –q –I enp0s3 –c /etc/snort/ test_snort.conf

DoS attack configure using Hping3:

1. **Install hping3**
   #sudo apt install hping3 –y

2. **DoS attack command**
   Replace 192.168.56.10 with your own host ip
   # sudo hping3 192.168.56.10 -q -n -d 120 -S -p 80 --flood --rand-source

Certificate Revocation:

openssl ca –keyfile ca.key –cert ca.crt –revoke server.crt

openssl ocsp –Cafile ca.crt –issuer ca.crt –cert server.crt –url http://www.mysecureserver.com:8080 –resp_text –noverify

Configuration file for root certificate:

```
[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
0.organizationName_default = MSR IT

[ 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
```

Configuration file for sub-ca certificate:

```
[ca]
#/root/ca/root-ca/root-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
0.organizationName_default = MSR IT

[ 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
```