Public-Key Infrastructure (PKI) Lab

Maryam Fahmi (501096276)
Sidra Musheer (501122840)
Veezish Ahmad (501080184)

CPS633 Sec. 10 - Group 61
Computer Security
Toronto Metropolitan University

Task 1: Becoming our own CA

1. Initialization and task:

a. Docker setup

b. Copy config file to directory and edit the file (uncomment unique subject to allow us to create multiple certificates with same subject), create index.txt (empty) and serial (contains 1000 as str) files.

```
Step 4/7 : COPY ./bank32_apache_ssl.conf /etc/apache2/sites-available
 ---> ef16b32a7aae
Step 5/7 : COPY ./certs/bank32.crt ./certs/bank32.key  /certs/
 ---> d67dfb00e9ad
Step 6/7 : RUN  chmod 400 /certs/bank32.key      && chmod 644 $WWWDIR/i
ndex.html      && chmod 644 $WWWDIR/index_red.html      && a2ensite ban
k32_apache_ssl
 ---> Running in 9f739f1cf728
Enabling site bank32_apache_ssl.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container 9f739f1cf728
 ---> 96343dd9c078
Step 7/7 : CMD  tail -f /dev/null
 ---> Running in 8161fc3b1cd5
Removing intermediate container 8161fc3b1cd5
 ---> ad2cd6ab9bca

Successfully built ad2cd6ab9bca
Successfully tagged seed-image-www-pki:latest
[10/02/24]seed@VM:~/.../Labsetup$ docker-compose up
Creating network "net-10.9.0.0" with the default driver
Creating www-10.9.0.80 ... done
Attaching to www-10.9.0.80
```

c. Generating ca.key and ca.crt using given code (password = cupcake)

```
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 \
> -keyout ca.key -out ca.crt
Generating a RSA private key
.....................................................................................
.....................................................................................
............++++
.....................................................................................++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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]:CA
State or Province Name (full name) [Some-State]:Ontario
Locality Name (eg, city) []:Toronto
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cupcakes
Organizational Unit Name (eg, section) []:Cookies
Common Name (e.g. server FQDN or YOUR name) []:633 User
Email Address []:633user@gmail.com
```

2. Answering questions

a. What part of the certificate indicates this is a CA's certificate? When analyzing the output of ca.crt, there is a part that says CA: TRUE that proves that this is a CA Certificate.

(Screenshot below - highlighted yellow)

```
[10/02/24]seed@VM:~/.../Labsetup$ cp /usr/lib/ssl/openssl.cnf
cp: missing destination file operand after '/usr/lib/ssl/openssl.cnf'
Try 'cp --help' for more information.
[10/02/24]seed@VM:~/.../Labsetup$ cp /usr/lib/ssl/openssl.cnf .
[10/02/24]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  image_www  openssl.cnf  volumes
[10/02/24]seed@VM:~/.../Labsetup$ nano openssl.cnf
[10/02/24]seed@VM:~/.../Labsetup$ touch index.txt
[10/02/24]seed@VM:~/.../Labsetup$ echo 1000 > serial
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -x509 -newkey rsa:4096 -s
ha256 -days 3650 \
> -keyout ca.key -out ca.crt
Generating a RSA private key
```

b. What part of the certificate indicates this is a self-signed certificate? When analyzing the output of ca.crt, Subject Key Identifier and Authority Key Identifier are the same, meaning this is self-signed. (Screenshot below - highlighted green)

```
                8D:58:8T:16:C1:03:46:e0:ac:28:a0:De:13:23:9c:
                d7:f4:cc:33:9b:b8:ab:b5:d8:45:3d:4d:c9:d2:49:
                f6:9c:10:78:6e:97:d7:11:96:aa:4f:bd:a2:ed:7c:
                9f:ac:84:c5:41:18:8c:25:b4:3a:21:3b:b0:ed:67:
                7c:de:2e:d9:17:3f:46:41:d6:0a:45:dd:95:8c:bd:
                55:e1:f7
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            8A:0F:D4:A6:8F:46:95:8E:16:37:A2:B8:95:1F:B9:0C:C9:44:19:69
        X509v3 Authority Key Identifier:
            keyid:8A:0F:D4:A6:8F:46:95:8E:16:37:A2:B8:95:1F:B9:0C:C9:44:19:69

        X509v3 Basic Constraints: critical
            CA:TRUE
```

c. Public exponent *(e)*

i. Certificate file:

```
                55:e1:f7
            Exponent: 65537 (0x10001)
```

ii. Key file:

```
publicExponent: 65537 (0x10001)
```

d. Private exponent *(d)*

i. Key file:

```
privateExponent:
    4b:61:9e:1f:a0:38:32:d9:a7:c0:57:de:ea:71:e2:
    3a:11:4d:19:c8:04:9b:da:a9:0d:10:00:23:c9:19:
    f4:32:64:14:dd:4e:3c:cb:6b:e3:a4:ec:8d:0e:d2:
    28:3f:c8:14:3c:df:10:f0:aa:9f:88:8f:35:86:83:
    9f:8e:e1:97:6b:0b:9d:12:5c:7f:05:2f:75:7b:a4:
    03:2b:46:3a:cc:3f:be:01:cc:d2:2f:e2:fd:bf:0f:
    e2:51:90:1f:8d:38:fd:dd:6b:68:2f:f7:73:99:be:
    3f:91:25:b4:0b:c8:c6:7d:83:2b:b0:6a:1f:ce:2f:
    e1:1b:6b:a2:2e:d1:9b:57:4c:fa:85:f3:fa:7f:5b:
    0b:0d:1f:03:20:09:dc:9a:89:f9:01:c3:51:c6:46:
    52:58:c7:03:ce:f7:7f:ea:16:6c:a5:b7:d0:93:d7:
    91:ed:e9:3a:77:de:63:0e:17:71:2f:83:a2:d9:19:
    a2:26:a3:c6:94:41:87:19:a8:e0:12:c4:7c:d2:f3:
    22:b9:b9:a7:8e:28:e0:e6:19:6f:9f:6e:e4:cf:cd:
    ad:3a:a2:44:b2:7d:fb:54:ab:6d:0d:7c:31:b1:a6:
    fd:f0:c1:cd:72:e8:1a:48:ac:81:37:66:3a:fa:05:
    1b:41:36:44:41:e1:34:f9:2a:86:c5:07:f3:ab:75:
    c5:3e:8e:1e:be:c0:29:fe:b5:87:f5:fa:f1:36:94:
    84:3a:f9:36:d4:59:06:90:d4:31:47:61:4e:63:1e:
    18:ec:36:79:dc:b2:64:47:ba:56:d0:5d:20:d3:e4:
    86:31:3d:62:4b:68:8e:76:1d:b3:2f:85:0c:ad:25:
    b8:eb:6f:f8:00:85:4e:08:5c:99:30:ed:ef:09:04:
    a5:53:ea:c1:97:d5:e2:3b:1c:0b:9a:e7:73:e0:5d:
    58:0b:e0:a7:1f:a7:82:da:6e:a4:3e:c7:7c:79:e8:
    11:0c:b8:a5:da:d3:a8:ab:0c:27:52:38:3c:7f:0c:
    2c:05:e1:4d:98:1b:0a:69:ac:13:d7:4b:17:d3:0e:
    5f:66:1d:3c:6a:d3:a8:54:c7:41:7c:f9:4e:fc:8b:
    de:d0:54:d2:55:50:fc:e8:f3:42:2f:54:b4:86:14:
    50:13:f2:2a:a3:3c:c0:23:77:16:e3:68:0f:d0:a3:
    10:a1:64:26:7f:90:fd:95:4c:73:72:e5:0a:0d:cd:
    ad:73:50:85:be:6c:63:eb:0e:7f:74:ea:1c:8f:36:
    23:50:cb:d6:e4:3a:fb:39:74:9a:8c:b0:e8:c9:b0:
    3f:84:65:25:38:f9:2f:d8:31:d1:eb:13:41:61:59:
    03:b3:45:fd:31:04:38:d4:16:98:6e:e9:dd:1a:c0:
    bc:f1
```

e. Modulus (n)

i. Certificate file

```
Modulus:
    00:c6:75:c4:77:e5:03:4f:ea:63:a1:ab:ce:b3:a0:
    fb:8b:e8:bf:e3:ea:63:13:2c:13:89:6a:61:43:96:
    ca:b6:17:a2:b2:1c:e3:8b:8f:95:09:60:89:be:71:
    77:3b:cd:09:29:cb:31:d8:3d:3f:42:50:2e:8b:dc:
    19:9b:a2:19:23:20:5f:d9:25:90:25:0a:2d:02:7d:
    60:a5:b0:50:32:0c:04:6c:bd:0c:9e:4c:81:2f:1c:
    93:52:8b:4a:8f:6f:49:f6:aa:95:71:7a:48:94:83:
    72:e0:2a:2d:72:72:0e:e4:e2:4e:7a:b9:a5:27:3c:
    99:b7:dc:57:b7:f4:70:48:53:97:7d:f5:3a:3e:36:
    21:5d:93:71:4d:31:3d:5f:0f:2c:c2:fb:72:91:c8:
    33:2d:fd:bb:1f:aa:0f:45:42:33:a2:21:04:35:8e:
    3d:5f:66:44:10:42:91:23:35:04:eb:2a:04:c6:87:
    bb:bf:60:c8:38:c4:a1:a8:79:84:03:5e:96:7d:f9:
    31:0a:87:86:d3:8a:50:6e:90:2f:72:95:49:bf:cb:
    34:f6:88:17:ef:2e:e4:4d:a6:a9:06:75:1b:4b:ad:
    b1:9c:6d:75:de:02:22:6c:2c:dd:f2:d0:9d:a8:67:
    40:9b:65:0e:5c:7e:f6:00:9e:f1:d1:53:9b:2e:c6:
    a3:eb:b8:91:2b:94:84:8d:fc:48:79:aa:a2:f6:dd:
    84:ea:6d:6b:69:f9:a2:6e:f6:3c:b6:c9:20:98:8e:
    b8:10:2c:ef:ab:7c:23:97:12:7b:44:4e:b9:f9:4e:
    87:08:2f:1f:a6:b5:12:88:d6:97:f5:cc:c1:d9:9e:
    d9:9a:d6:3b:07:c9:2f:cc:60:6d:45:d3:14:9f:62:
    53:95:4f:ba:50:e6:0b:aa:fa:b0:59:48:1d:07:2f:
    6c:ef:eb:80:f6:df:e3:2f:03:3e:8f:9e:f5:e7:3c:
    ca:63:00:01:fb:02:a4:34:78:2e:8a:45:ab:bd:2e:
    68:3f:18:3b:83:29:12:9c:4d:6a:01:83:fd:20:e4:
    c6:bc:e1:e1:e4:72:bc:d1:9f:d7:7a:44:29:73:2b:
    94:77:5d:03:b9:c8:75:00:27:00:23:13:64:10:47:
    27:94:69:6a:d6:1f:a1:22:ba:2a:b8:4d:12:69:fd:
    8b:58:8f:16:c1:03:46:e0:ac:28:a0:be:13:23:9c:
    d7:f4:cc:33:9b:b8:ab:b5:d8:45:3d:4d:c9:d2:49:
    f6:9c:10:78:6e:97:d7:11:96:aa:4f:bd:a2:ed:7c:
    9f:ac:84:c5:41:18:8c:25:b4:3a:21:3b:b0:ed:67:
    7c:de:2e:d9:17:3f:46:41:d6:0a:45:dd:95:8c:bd:
    55:e1:f7
```

ii. Key file

```
modulus:
    00:c6:75:c4:77:e5:03:4f:ea:63:a1:ab:ce:b3:a0:
    fb:8b:e8:bf:e3:ea:63:13:2c:13:89:6a:61:43:96:
    ca:b6:17:a2:b2:1c:e3:8b:8f:95:09:60:89:be:71:
    77:3b:cd:09:29:cb:31:d8:3d:3f:42:50:2e:8b:dc:
    19:9b:a2:19:23:20:5f:d9:25:90:25:0a:2d:02:7d:
    60:a5:b0:50:32:0c:04:6c:bd:0c:9e:4c:81:2f:1c:
    93:52:8b:4a:8f:6f:49:f6:aa:95:71:7a:48:94:83:
    72:e0:2a:2d:72:72:0e:e4:e2:4e:7a:b9:a5:27:3c:
    99:b7:dc:57:b7:f4:70:48:53:97:7d:f5:3a:3e:36:
    21:5d:93:71:4d:31:3d:5f:0f:2c:c2:fb:72:91:c8:
    33:2d:fd:bb:1f:aa:0f:45:42:33:a2:21:04:35:8e:
    3d:5f:66:44:10:42:91:23:35:04:eb:2a:04:c6:87:
    bb:bf:60:c8:38:c4:a1:a8:79:84:03:5e:96:7d:f9:
    31:0a:87:86:d3:8a:50:6e:90:2f:72:95:49:bf:cb:
    34:f6:88:17:ef:2e:e4:4d:a6:a9:06:75:1b:4b:ad:
    b1:9c:6d:75:de:02:22:6c:2c:dd:f2:d0:9d:a8:67:
    40:9b:65:0e:5c:7e:f6:00:9e:f1:d1:53:9b:2e:c6:
    a3:eb:b8:91:2b:94:84:8d:fc:48:79:aa:a2:f6:dd:
    84:ea:6d:6b:69:f9:a2:6e:f6:3c:b6:c9:20:98:8e:
    b8:10:2c:ef:ab:7c:23:97:12:7b:44:4e:b9:f9:4e:
    87:08:2f:1f:a6:b5:12:88:d6:97:f5:cc:c1:d9:9e:
    d9:9a:d6:3b:07:c9:2f:cc:60:6d:45:d3:14:9f:62:
    53:95:4f:ba:50:e6:0b:aa:fa:b0:59:48:1d:07:2f:
    6c:ef:eb:80:f6:df:e3:2f:03:3e:8f:9e:f5:e7:3c:
    ca:63:00:01:fb:02:a4:34:78:2e:8a:45:ab:bd:2e:
    68:3f:18:3b:83:29:12:9c:4d:6a:01:83:fd:20:e4:
    c6:bc:e1:e1:e4:72:bc:d1:9f:d7:7a:44:29:73:2b:
    94:77:5d:03:b9:c8:75:00:27:00:23:13:64:10:47:
    27:94:69:6a:d6:1f:a1:22:ba:2a:b8:4d:12:69:fd:
    8b:58:8f:16:c1:03:46:e0:ac:28:a0:be:13:23:9c:
    d7:f4:cc:33:9b:b8:ab:b5:d8:45:3d:4d:c9:d2:49:
    f6:9c:10:78:6e:97:d7:11:96:aa:4f:bd:a2:ed:7c:
    9f:ac:84:c5:41:18:8c:25:b4:3a:21:3b:b0:ed:67:
    7c:de:2e:d9:17:3f:46:41:d6:0a:45:dd:95:8c:bd:
    55:e1:f7
```

f. n = pq (the prime factors of modulus n), p and q will be the prime factors in the key file (Screenshots below for p and q):

```
prime1:
    00:f4:1a:ae:ff:34:01:01:cc:55:60:ed:49:9d:0a:
    b5:70:e3:6b:a4:f8:84:b2:c4:04:3a:2e:62:b0:c8:
    b9:1f:b8:cd:6b:ac:20:a5:1b:3d:a6:b6:78:d1:0c:
    3e:0d:7b:f5:e6:da:d1:03:d5:9a:f4:37:c8:f7:d3:
    14:4e:c3:dd:13:61:bb:a8:f3:fd:59:12:65:cf:1d:
    17:ec:07:5a:6f:7a:a5:15:70:01:b3:08:36:de:79:
    2c:c9:3f:9b:f5:33:2d:db:94:69:db:4b:f0:9f:10:
    34:3d:b3:03:b0:b6:1d:5e:d5:0e:34:f4:0b:9a:60:
    29:3e:56:06:96:0a:39:ca:30:90:8b:04:e5:12:91:
    17:4d:11:77:e5:87:de:48:d4:15:43:59:79:70:71:
    07:1c:05:b5:7f:c7:e2:ec:5c:cf:ca:04:93:70:51:
    64:f3:01:d3:c3:4e:70:8c:4c:e9:d5:f6:04:dd:03:
    ee:74:be:0e:40:42:5c:e5:4f:ad:60:bb:3d:47:6d:
    67:39:8b:51:7b:96:9b:12:18:c0:e4:b8:83:90:20:
    f8:03:44:e1:b9:81:5c:d2:c2:f9:74:5a:65:30:57:
    9d:3f:02:c5:74:26:76:49:ed:e8:5e:2a:a1:b0:cd:
    be:c9:57:d0:7f:9b:2c:91:0d:17:05:7d:ea:ea:d2:
    69:9f
prime2:
    00:d0:21:a6:95:a2:67:5f:01:7a:37:45:7c:87:f6:
    df:2b:ec:f9:68:bf:f9:a6:62:b8:2d:87:a8:fd:fc:
    37:18:92:f5:c4:67:49:3c:f0:f9:3f:be:a0:bd:98:
    cc:26:45:86:53:0c:2a:a5:ce:5d:0d:fe:43:1a:92:
    95:33:eb:57:41:68:44:b7:52:c3:87:9a:56:0d:4e:
    d6:bb:2f:30:3b:8f:ec:05:60:31:d4:8b:1a:4d:22:
    72:8b:c5:bb:21:7e:aa:b7:e7:35:00:3a:b0:0b:0d:
    12:80:2f:73:2f:a5:ec:15:81:52:34:9d:02:72:e6:
    5f:a6:12:69:a0:82:31:92:ec:4b:ad:cb:4f:d7:b2:
    fe:76:07:0e:ba:d8:b6:2a:84:c1:22:a3:7c:bd:96:
    66:59:29:d7:22:b9:93:79:ad:61:b8:24:ac:c3:d2:
    bd:17:62:24:ce:d4:e0:a3:df:87:d4:5d:ec:28:c7:
    29:8e:45:28:e8:50:0f:41:81:2b:3b:ef:94:47:74:
    0a:e4:db:98:aa:41:94:56:3c:ff:30:21:23:d9:39:
    76:5c:40:c4:b8:63:59:2b:db:33:ad:71:e6:6b:e2:
    46:d1:f4:72:58:1d:cf:e6:04:95:1f:d2:e8:30:61:
    8c:c0:f6:c4:36:b0:f4:f4:34:25:4b:99:6d:2e:b0:
    d8:a9
```

Task 2: Intermediate CA Setup

Steps:

1. Generating Private Key and CSR for the Website (bank32.com):
   Command: openssl req -newkey rsa:2048 -sha256 \
   -keyout server.key -out server.csr \
   -subj "/CN=www.bank32.com/O=Bank32 Inc./C=US" \
   -passout pass:dees

2. Check that its properly generated:
   openssl req -in server.csr -text -noout
   openssl rsa -in server.key -text -noout

3. Add the alternative website names:
   Command: openssl req -newkey rsa:2048 -sha256 \
   -keyout server.key -out server.csr \
   -subj "/CN=www.bank32.com/O=Bank32 Inc./C=US" \
   -passout pass:dees -addext "subjectAltName = DNS:www.bank32.com, \
   DNS:www.bank32A.com, \
   DNS:www.bank32B.com"

Output:

server.key (private key) and server.crt (signed certificate) are generated.

Lab. 3

Screenshot:

```
8423a04990d8   seed-image-www-pki   "/bin/sh -c 'tail -f…"   6 seconds ago   Up 4 seconds            www-10.9.0.80
[10/02/24]seed@VM:~/.../Labsetup$ echo "alias dcbuild='docker-compose build'" >> ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ echo "alias dcup='docker-compose up -d'" >> ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ echo "alias dcdown='docker-compose down'" >> ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ echo "alias dockps='docker ps --format \"{{.ID}} {{.Names}}\"'" >> ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ echo "alias docksh='docker exec -it'" >> ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ source ~/.bashrc
[10/02/24]seed@VM:~/.../Labsetup$ dockps
8423a04990d8 www-10.9.0.80
[10/02/24]seed@VM:~/.../Labsetup$ openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.......................+++++
..........+++++
e is 65537 (0x010001)
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -new -key server.key -out 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]:OM
State or Province Name (full name) [Some-State]:Muscat
Locality Name (eg, city) []:Azaiba
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Zara Stone
Organizational Unit Name (eg, section) []:Marble
Common Name (e.g. server FQDN or YOUR name) []:zarastoneintl.com
Email Address []:zarastoneintl@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:sidza
An optional company name []:google
[10/02/24]seed@VM:~/.../Labsetup$ openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
Signature ok
subject=C = OM, ST = Muscat, L = Azaiba, O = "Zara Stone ", OU = Marble, CN = neintl.com, emailAddress = zarastoneintl@
gmail.com
Getting Private key
[10/02/24]seed@VM:~/.../Labsetup$
```

```
10.9.0.80       www.seedlab-shellshock.com
10.9.0.80       www.bank32.com
10.9.0.80       neintl.com

[10/02/24]seed@VM:~/.../Labsetup$ openssl req -newkey rsa:2048 -sha256 \
> -keyout server.key -out sever.csr \
> -subj "/CN=www.bank32.com/O=Bank32 Inc./C=US" \
> -passout pass:dees
Generating a RSA private key
.....................................+++++
.................................+++++
writing new private key to 'server.key'
-----
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -in server.csr -text -noout
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = OM, ST = Muscat, L = Azaiba, O = "Zara Stone ", OU = Marble, CN = neintl.com, emailAddress = zaras
toneintl@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:ce:2b:fa:d2:bb:fd:80:ae:63:e1:16:e3:0e:e2:
                    4e:72:26:37:ed:28:89:fc:0b:4f:d2:5b:00:9b:70:
                    50:b1:2b:08:f1:77:36:34:3f:43:c0:b2:44:4f:4f:
                    f6:d0:05:05:30:68:3c:98:1c:5c:c6:46:e7:b0:ad:
                    35:da:57:d5:aa:00:22:dc:ef:e8:f3:60:5b:ba:57:
                    80:59:3b:bb:6b:7a:ad:39:fa:a3:29:57:00:33:31:
                    17:2e:86:59:18:2e:65:97:ad:0d:94:23:c3:17:36:
                    6f:82:71:56:c6:41:73:74:5d:87:29:a9:65:13:93:
                    54:83:f1:d7:4a:2e:d8:98:9d:ff:30:b3:e2:2b:9c:
                    fc:50:31:af:96:d3:bf:f8:02:70:ed:ea:ab:a6:6d:
                    2b:6f:0e:7a:02:d6:dd:8c:b6:11:e4:1d:68:22:ee:
                    b6:fb:b6:a8:e4:97:ac:0a:f2:cb:e6:16:0d:27:22:
                    c1:38:a1:df:36:77:db:a5:a8:5c:db:49:c2:46:2f:
                    32:73:2b:2e:f1:bd:36:21:bb:ae:cb:f3:08:45:8f:
                    58:cb:61:b6:43:e3:a1:6d:f8:2e:66:10:d1:9a:29:
                    a9:98:38:17:b2:66:16:81:b9:c9:9f:b6:60:26:3f:
                    ba:4c:88:93:a9:5b:2b:35:b1:f9:0e:91:a5:9d:bd:
                    1f:fb
```

Lab. 3

```
                    fc:50:31:af:96:d3:bf:f8:02:70:ed:ea:ab:a6:6d:
                    2b:6f:0e:7a:02:d6:dd:8c:b6:11:e4:1d:68:22:ee:
                    b6:fb:b6:a8:e4:97:ac:0a:f2:cb:e6:16:0d:27:22:
                    c1:38:a1:df:36:77:db:a5:a8:5c:db:49:c2:46:2f:
                    32:73:2b:2e:f1:bd:36:21:bb:ae:cb:f3:08:45:8f:
                    58:cb:61:b6:43:e3:a1:6d:f8:2e:66:10:d1:9a:29:
                    a9:98:38:17:b2:66:16:81:b9:c9:9f:b6:60:26:3f:
                    ba:4c:88:93:a9:5b:2b:35:b1:f9:0e:91:a5:9d:bd:
                    1f:fb
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :sidza
            unstructuredName         :google
    Signature Algorithm: sha256WithRSAEncryption
         67:f2:76:9e:43:7c:b0:06:75:d5:4c:d9:35:ba:c9:c1:7d:df:
         12:36:e1:bf:da:1b:40:85:8f:2c:2b:03:ed:49:37:d3:be:fb:
         1f:ff:c4:cf:05:d1:15:9e:f1:cb:41:01:b0:b6:ee:14:4e:28:
         25:81:8f:20:06:a1:98:2f:2a:97:7e:02:ba:f8:cc:f1:76:ed:
         f1:58:d6:88:dd:d4:76:8a:01:69:2b:9e:3d:f3:c3:ff:08:29:
         ba:1c:11:bd:fb:ed:62:dd:83:1e:fe:a8:bd:05:d9:53:42:99:
         60:78:c7:fc:8c:a1:18:2d:5e:76:0e:44:d9:33:0f:10:56:83:
         20:7d:fe:5d:36:6d:f3:20:8d:af:d0:72:1c:8d:ad:2d:42:3b:
         73:3e:42:48:b4:16:67:74:f7:68:f1:0d:7a:12:5d:d2:02:f7:
         e3:78:d5:fb:34:29:5e:6a:4a:83:15:08:0c:f8:37:9a:ae:32:
         66:95:0e:10:e6:ab:19:b3:2b:08:97:dd:9c:de:41:80:f3:8f:
         c9:fb:96:85:7d:71:97:15:9e:81:cd:f9:ca:f5:80:b0:28:6b:
         3d:e5:a3:d4:8b:2a:be:72:76:9e:49:d0:da:b3:fc:63:70:af:
         17:51:7e:ed:b4:34:33:5e:b4:4b:59:d5:8a:1e:db:bf:92:91:
         55:36:98:cf
[10/02/24]seed@VM:~/.../Labsetup$ openssl rsa -in server.key -text -noout
Enter pass phrase for server.key:
```

```
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -in server.csr -text -noout
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = OM, ST = Muscat, L = Azaiba, O = "Zara Stone ", OU = Marble, CN = neintl.com, emailAddress = zaras
toneintl@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:ce:2b:fa:d2:bb:fd:80:ae:63:e1:16:e3:0e:e2:
                    4e:72:26:37:ed:28:89:fc:0b:4f:d2:5b:00:9b:70:
                    50:b1:2b:08:f1:77:36:34:3f:43:c0:b2:44:4f:4f:
                    f6:d0:05:05:30:68:3c:98:1c:5c:c6:46:e7:b0:ad:
                    35:da:57:d5:aa:00:22:dc:ef:e8:f3:60:5b:ba:57:
                    80:59:3b:bb:6b:7a:ad:39:fa:a3:29:57:00:33:31:
                    17:2e:86:59:18:2e:65:97:ad:0d:94:23:c3:17:36:
                    6f:82:71:56:c6:41:73:74:5d:87:29:a9:65:13:93:
                    54:83:f1:d7:4a:2e:d8:98:9d:ff:30:b3:e2:2b:9c:
                    fc:50:31:af:96:d3:bf:f8:02:70:ed:ea:ab:a6:6d:
                    2b:6f:0e:7a:02:d6:dd:8c:b6:11:e4:1d:68:22:ee:
                    b6:fb:b6:a8:e4:97:ac:0a:f2:cb:e6:16:0d:27:22:
                    c1:38:a1:df:36:77:db:a5:a8:5c:db:49:c2:46:2f:
                    32:73:2b:2e:f1:bd:36:21:bb:ae:cb:f3:08:45:8f:
                    58:cb:61:b6:43:e3:a1:6d:f8:2e:66:10:d1:9a:29:
                    a9:98:38:17:b2:66:16:81:b9:c9:9f:b6:60:26:3f:
                    ba:4c:88:93:a9:5b:2b:35:b1:f9:0e:91:a5:9d:bd:
                    1f:fb
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :sidza
            unstructuredName         :google
    Signature Algorithm: sha256WithRSAEncryption
         67:f2:76:9e:43:7c:b0:06:75:d5:4c:d9:35:ba:c9:c1:7d:df:
         12:36:e1:bf:da:1b:40:85:8f:2c:2b:03:ed:49:37:d3:be:fb:
         1f:ff:c4:cf:05:d1:15:9e:f1:cb:41:01:b0:b6:ee:14:4e:28:
         25:81:8f:20:06:a1:98:2f:2a:97:7e:02:ba:f8:cc:f1:76:ed:
         f1:58:d6:88:dd:d4:76:8a:01:69:2b:9e:3d:f3:c3:ff:08:29:
         ba:1c:11:bd:fb:ed:62:dd:83:1e:fe:a8:bd:05:d9:53:42:99:
         60:78:c7:fc:8c:a1:18:2d:5e:76:0e:44:d9:33:0f:10:56:83:
         20:7d:fe:5d:36:6d:f3:20:8d:af:d0:72:1c:8d:ad:2d:42:3b:
```

```
            c9:1b:96:85:7d:71:97:15:9e:81:c0:19:ca:15:80:b0:28:6b:
            3d:e5:a3:d4:8b:2a:be:72:76:9e:49:d0:da:b3:fc:63:70:af:
            17:51:7e:ed:b4:34:33:5e:b4:4b:59:d5:8a:1e:db:bf:92:91:
            55:36:98:cf
[10/02/24]seed@VM:~/.../Labsetup$ openssl rsa -in server.key -text -noout
Enter pass phrase for server.key:
RSA Private-Key: (2048 bit, 2 primes)
modulus:
    00:c5:b5:f5:3b:22:27:5b:ac:0d:d1:f3:05:60:20:
    20:f8:5d:21:21:ec:ac:9b:17:7e:4f:59:1f:7b:b0:
    0a:f4:ed:82:9c:09:77:df:01:cc:8a:56:5a:a1:af:
    d7:22:ac:b1:06:16:c8:7a:83:8c:12:43:77:65:15:
    21:c3:f8:71:5c:32:fd:b9:47:63:2d:02:fa:31:3b:
    b8:23:e0:22:1c:ce:38:0b:34:29:f2:f0:a9:d7:0e:
    6b:79:fc:b7:16:76:eb:a4:ec:03:02:e6:0d:36:1d:
    75:c1:25:a6:46:7e:30:65:71:9d:24:5d:70:c1:4f:
    03:55:da:e1:d0:3f:1c:58:41:d3:ca:20:51:55:59:
    95:28:74:1d:d2:0b:18:f2:9b:f8:be:af:cf:e3:8a:
    91:f9:b5:1a:23:77:23:4b:1c:df:1e:26:bc:17:24:
    67:05:88:f7:2d:83:06:35:dd:59:3e:ec:92:93:71:
    ff:50:22:c6:84:c5:49:27:3d:a6:30:df:ae:65:4f:
    4d:48:30:79:9b:0a:79:a2:eb:6d:87:77:95:62:f8:
    b7:56:41:e2:d6:0c:25:7a:d3:25:df:40:61:85:0a:
    43:bd:b0:a6:62:1c:21:99:7f:31:9e:4c:c4:6a:b8:
    a1:2a:e7:6f:08:08:67:0d:f6:c9:03:51:4c:ea:0b:
    62:fb
publicExponent: 65537 (0x10001)
privateExponent:
    00:9f:49:ff:3e:da:40:7d:82:3c:4c:37:90:d0:26:
    72:89:c7:76:87:3f:88:bd:17:8b:83:60:59:96:2c:
    7a:57:64:f8:02:c1:3a:1b:a8:f6:63:4a:39:90:e4:
    fb:de:8a:e5:c0:f3:20:28:5c:cd:c0:75:2c:bf:7a:
    ec:0e:58:9e:f4:5a:7c:c3:06:b9:e7:ac:eb:68:39:
    26:1d:79:59:e2:7d:5e:f7:df:da:20:4a:37:7b:f5:
    b9:2f:ba:24:c4:6b:a1:64:e9:65:6f:b3:6b:57:7b:
    1a:c1:37:95:24:5c:4c:9f:9e:8e:ad:9a:be:da:6f:
    cf:4f:1f:9f:29:49:7f:8b:b6:6a:1f:30:c7:5d:74:
    df:46:97:69:f3:9f:20:fe:1e:c9:b6:01:9e:1f:b1:
    45:72:0f:16:1b:05:9a:91:07:1c:2a:69:bc:6b:b4:
    6e:7f:16:68:9b:26:99:71:64:1c:40:5d:64:ec:ab:
    65:d9:15:80:34:e7:84:f3:27:e2:4c:1b:5e:27:16:
```

```
    d1:2d:e7:6f:06:06:b7:0d:f6:c9:05:51:4c:ea:0b:
    62:fb
publicExponent: 65537 (0x10001)
privateExponent:
    00:9f:49:ff:3e:da:40:7d:82:3c:4c:37:90:d0:26:
    72:89:c7:76:87:3f:88:bd:17:8b:83:60:59:96:2c:
    7a:57:64:f8:02:c1:3a:1b:a8:f6:63:4a:39:90:e4:
    fb:de:8a:e5:c0:f3:20:28:5c:cd:c0:75:2c:bf:7a:
    ec:0e:58:9e:f4:5a:7c:c3:06:b9:e7:ac:eb:68:39:
    26:1d:79:59:e2:7d:5e:f7:df:da:20:4a:37:7b:f5:
    b9:2f:ba:24:c4:6b:a1:64:e9:65:6f:b3:6b:57:7b:
    1a:c1:37:95:24:5c:4c:9f:9e:8e:ad:9a:be:da:6f:
    cf:4f:1f:9f:29:49:7f:8b:b6:6a:1f:30:c7:5d:74:
    df:46:97:69:f3:9f:20:fe:1e:c9:b6:01:9e:1f:b1:
    45:72:0f:16:1b:05:9a:91:07:1c:2a:69:bc:6b:b4:
    6e:7f:16:68:9b:26:99:71:64:1c:40:5d:64:ec:ab:
    65:d9:15:80:34:e7:84:f3:27:e2:4c:1b:5e:27:16:
    78:de:cc:b7:64:27:94:a7:58:4d:7b:80:dd:3d:18:
    10:ad:c3:a4:4b:bf:b4:ab:61:aa:6f:85:4c:0a:08:
    49:ab:73:82:39:09:79:b1:81:43:93:3e:a7:58:48:
    af:8c:fb:d1:f3:de:88:0a:1f:d5:a4:84:1c:b2:df:
    2f:c9
prime1:
    00:f0:4d:f2:6b:3c:3f:fb:e4:ac:64:e6:b5:8c:6a:
    bd:56:10:85:1b:fe:e5:07:f2:28:98:1b:ae:46:0f:
    62:aa:b9:8f:99:a8:23:80:f0:9f:b1:8a:0d:4c:fc:
    fa:41:cd:b9:53:48:1c:17:15:0f:b2:c6:97:7b:da:
    3c:32:38:84:fc:00:14:36:94:d1:86:75:18:3c:e3:
    c1:64:c5:2f:69:67:b5:ea:84:32:a2:6e:d2:3f:d2:
    8c:30:50:48:35:75:ea:1e:bf:e1:25:63:3c:73:d1:
    49:e9:11:3b:f8:ef:f5:bb:da:80:c5:17:6c:92:c0:
    45:84:1e:0b:d1:09:6d:08:0d
prime2:
    00:d2:9f:d0:c2:2c:d7:6c:9d:19:48:c2:df:ba:e2:
    90:94:fe:8b:f5:f9:a9:99:f9:68:8a:b0:69:c3:d4:
    5b:39:44:93:b0:27:a4:06:dd:2f:2a:fe:6f:21:b5:
    c1:37:b3:7a:bc:42:e4:99:02:b2:1a:5e:61:7f:08:
    e6:ae:ce:30:7e:c4:7d:2f:e0:86:a7:01:3d:3b:b0:
    ad:6a:b3:c6:1a:ef:4b:2b:64:e5:7d:f8:98:74:0a:
    27:a8:ab:f8:0d:93:5d:28:7a:8c:95:46:16:a1:bf:
    9a:39:ff:a0:8e:78:f6:60:75:c2:cd:91:3f:87:e8:
```

```
prime2:
    00:d2:9f:d0:c2:2c:d7:6c:9d:19:48:c2:df:ba:e2:
    90:94:fe:8b:f5:f9:a9:99:f9:68:8a:b0:69:c3:d4:
    5b:39:44:93:b0:27:a4:06:dd:2f:2a:fe:6f:21:b5:
    c1:37:b3:7a:bc:42:e4:99:02:b2:1a:5e:61:7f:08:
    e6:ae:ce:30:7e:c4:7d:2f:e0:86:a7:01:3d:3b:b0:
    ad:6a:b3:c6:1a:ef:4b:2b:64:e5:7d:f8:98:74:0a:
    27:a8:ab:f8:0d:93:5d:28:7a:8c:95:46:16:a1:bf:
    9a:39:ff:a0:8e:78:f6:60:75:c2:cd:91:3f:87:e8:
    b5:f2:ee:8a:f0:e1:00:8d:27
exponent1:
    35:5c:89:bd:68:56:a8:ff:e1:8c:52:72:f4:28:6b:
    bc:e5:d4:39:20:44:09:9c:ab:89:03:74:92:98:2e:
    07:cd:46:e7:0f:20:3b:2c:b3:b9:7f:f7:6e:26:2b:
    08:5b:bf:90:8f:cd:b5:0e:77:3a:f2:c0:86:bf:32:
    68:d0:86:2f:53:71:29:a1:a8:59:5c:3d:32:a0:3c:
    bc:bc:d0:c7:6e:41:46:3f:6e:e3:05:0a:e9:23:3f:
    00:27:83:b1:63:6a:c6:c3:43:22:c3:43:94:50:60:
    42:42:12:78:38:08:f0:5b:18:07:2c:29:6e:f9:05:
    36:ce:59:3f:65:01:72:2d
exponent2:
    00:b2:99:39:d7:ad:56:6e:8f:38:25:6e:b9:70:2b:
    36:16:61:52:25:1d:b3:27:11:4a:08:70:56:fe:eb:
    4e:ff:5c:9b:2f:40:2d:e2:74:23:ce:fd:39:17:08:
    9b:f2:b3:8a:5b:a1:0d:5e:81:de:ac:65:63:9e:e8:
    d9:53:59:1b:de:22:97:54:c5:ba:e8:5a:30:a6:30:
    c6:9c:38:1c:c0:32:31:37:76:b2:f3:02:9d:a3:a8:
    08:c1:0a:31:a8:b1:50:c9:46:41:77:42:9b:66:44:
    a6:6e:8f:c8:81:87:28:c3:da:97:78:e0:b5:52:75:
    8d:f3:95:3a:d3:ce:b7:3c:f9
coefficient:
    00:dc:7d:c0:22:61:84:dd:e4:2f:13:74:1b:3a:16:
    56:fe:22:45:50:f6:f9:7f:55:30:b8:20:6f:6f:a7:
    7a:1c:d6:3e:a2:8f:f7:d6:a5:f8:3c:b4:d0:31:93:
    d4:c1:33:a5:88:c2:ab:a6:76:5b:bb:d5:a9:62:07:
    c0:ed:67:ca:67:f9:c8:46:72:38:fa:3f:ec:ce:1f:
    36:2c:d9:ab:00:9a:7e:47:c0:6d:2a:71:54:55:f9:
    b1:b7:7c:4a:1b:e8:c7:b3:34:fa:72:5b:07:d4:d5:
    bb:f1:75:3c:5e:89:f2:e6:fa:fc:78:2e:c3:f9:d9:
    2d:0b:81:47:34:88:18:0c:35
[10/02/24]seed@VM:~/.../Labsetup$ openssl ca -config myCA_openssl.cf -policy policy_anything \
```

Task 3: Signing Certificate for a Website

Objective:
Sign the website's certificate using the intermediate CA.

Steps:
1. Signing the Website's CSR with the Root CA.
 Command: `openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -out server.crt -days 365 -sha256`

2. Uncommenting 'copy' in openssl.cnf to allow openssl ca command to access the website through alternative names.

Output:
- Generated `server.key` (private key) and `server.crt` (signed certificate).

Lab. 3

Screenshot:

```
                                # (optional, default: no)
ess_cert_id_alg         = sha1  # algorithm to compute certificate
                                # identifier (optional, default: sha1)
[10/02/24]seed@VM:~/.../Labsetup$ sudo nano /usr/lib/ssl/openssl.cnf
[10/02/24]seed@VM:~/.../Labsetup$ sudo nano /usr/lib/ssl/openssl.cnf
[10/02/24]seed@VM:~/.../Labsetup$ openssl req -config myCA_openssl.cnf -new -newkey rsa:2048 -nodes -keyout test.key -o
ut test.csr
Generating a RSA private key
.+++++
...........+++++
writing new private key to 'test.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) [AU]:OM
State or Province Name (full name) [Some-State]:Muscat
Locality Name (eg, city) []:Azaiba
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Zara Stone
Organizational Unit Name (eg, section) []:Marble
Common Name (e.g. server FQDN or YOUR name) []:zarastoneintl.com
Email Address []:zarastoneintl@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:sidza
An optional company name []:google
```

```
[10/03/24]seed@VM:~/.../Labsetup$ openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 -keyout ca.key -out ca.crt -su\
bj "/CN=www.modelCA.com/O=Model CA LTD./C=US" -passout pass:dees
Generating a RSA private key
.....................................................................................................................
.....................................................+++
.....................................................+++
writing new private key to 'ca.key'
-----
[10/03/24]seed@VM:~/.../Labsetup$ ls -l ca.key ca.crt
-rw-rw-r-- 1 seed docker 1923 Oct  3 00:16 ca.crt
-rw------- 1 seed docker 3414 Oct  3 00:16 ca.key
[10/03/24]seed@VM:~/.../Labsetup$ openssl ca -config myCA_openssl.cnf -policy policy_anything -md sha256 -days 3650 -i\
n server.csr -out server.crt -batch -cert ca.crt -keyfile ca.key
Using configuration from myCA_openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Oct  3 04:20:54 2024 GMT
            Not After : Oct  1 04:20:54 2034 GMT
        Subject:
            countryName               = OM
            stateOrProvinceName       = Muscat
            localityName              = Azaiba
            organizationName          = Zara Stone
            organizationalUnitName    = Marble
            commonName                = neintl.com
            emailAddress              = zarastoneintl@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                2C:DB:1C:CC:A2:B1:78:25:E7:B6:75:50:47:EB:14:E0:06:7E:B9:2F
```

Task 4: Deploying Certificate in Apache-Based HTTPS Website

Objective:
Deploy the signed certificate on an Apache server to secure the website with HTTPS.

Steps:
1. Modify the Apache VirtualHost file to configure SSL settings:
   - SSLCertificateFile: Path to the signed certificate (e.g., `server.crt`).
   - SSLCertificateKeyFile: Path to the private key (e.g., `server.key`).

2. Restart Apache.
   Command: ` service apache2 restart`

3. Enabling ssl module and bank32 site (a2enmod…, a2ensite…)

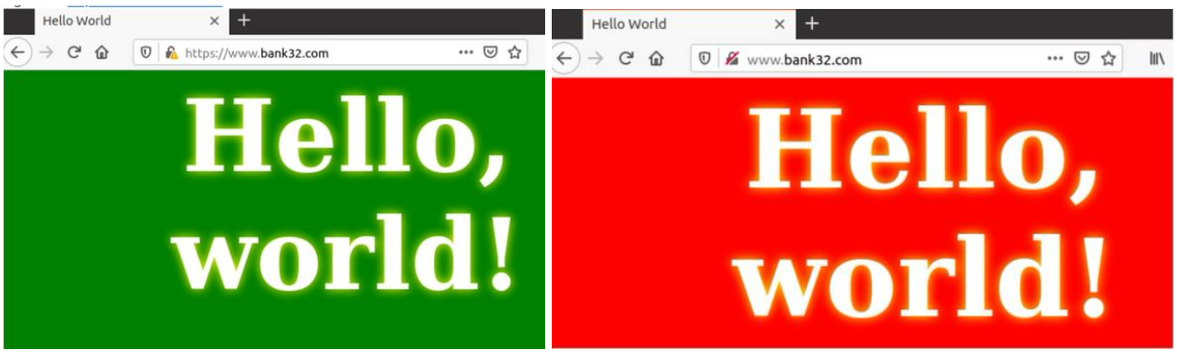
4. Access the site using HTTPS.

5. Website is initially inaccessible. In order to fix the "Potential Security Risk…" warning showing up, we had to add our docker hosts to /etc/hosts to set up the DNS. This allowed the website to resolve the www.bank32.com domain to the server's IP address.
   '10.9.0.80 www.bank32.com www.bank32A.com www.bank32B.com'

Output:
- Apache server successfully serves the site over HTTPS.

Screenshot:

```
[10/03/24]seed@VM:~/.../Labsetup$ sudo nano /etc/apache2/sites-availabl
e/bank32_apache_ssl.conf
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.^C
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.^C
[10/03/24]seed@VM:~/.../Labsetup$ ^Cdo nano /etc/apache2/sites-availabl
e/bank32_apache_ssl.conf
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.conf

Site bank32_apache_ssl already enabled
[10/03/24]seed@VM:~/.../Labsetup$ sudo apachectl configtest
Syntax OK
[10/03/24]seed@VM:~/.../Labsetup$ sudo mkdir -p /var/www/bank32
[10/03/24]seed@VM:~/.../Labsetup$ sudo nano /var/www/bank32/index.html
[10/03/24]seed@VM:~/.../Labsetup$ sudo ls -l /etc/ssl/certs/server.crt
-rw-r--r-- 1 root root 1988 Oct  3 01:36 /etc/ssl/certs/server.crt
[10/03/24]seed@VM:~/.../Labsetup$ sudo ls -l /etc/ssl/private/server.ke
y
-rw------- 1 root root 3272 Oct  3 01:36 /etc/ssl/private/server.key
```



Task 5: Launching a Man-In-The-Middle Attack

Objective:
Simulate a Man-In-The-Middle (MITM) attack by impersonating a target website (e.g., `example.com`).

Steps:
1. Modify the Apache configuration (ssl conf file) to impersonate `www.example.com`.
   - Use `ServerName www.example.com` in the Apache config.

2. Modify the victim's `/etc/hosts` file to redirect requests to your malicious web server.
   - Add entry: `10.9.0.80 www.example.com`

3. Restart Apache and visit the impersonated site.

Output:
- Browser should display a warning that the certificate is invalid, indicating a potential MITM attack.

Screenshot:

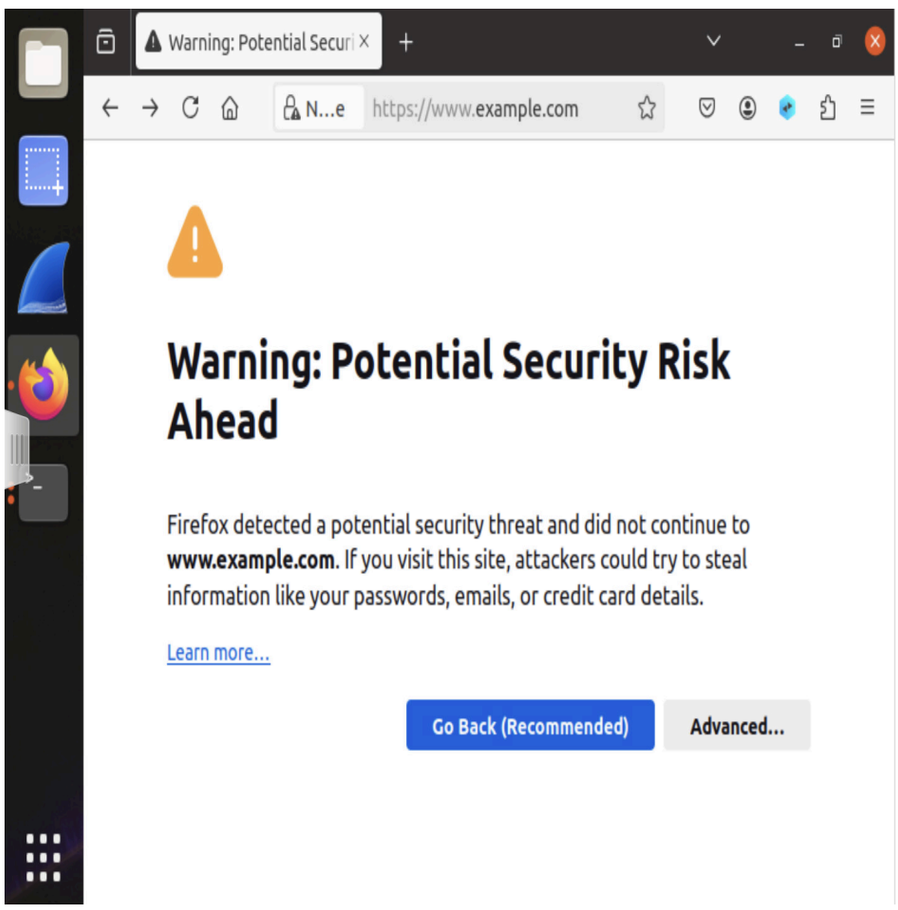
```
    /etc/apache2/sites-available/bank32_apache_ssl.conf
<VirtualHost *:443>
        DocumentRoot /var/www/bank32
        ServerName www.example.com
        ServerAlias www.exampleA.com
        ServerAlias www.exampleB.com
        DirectoryIndex index.html
        SSLEngine On
        SSLCertificateFile /etc/ssl/certs/server.crt
        SSLCertificateKeyFile /etc/ssl/private/server.key
</VirtualHost>
```

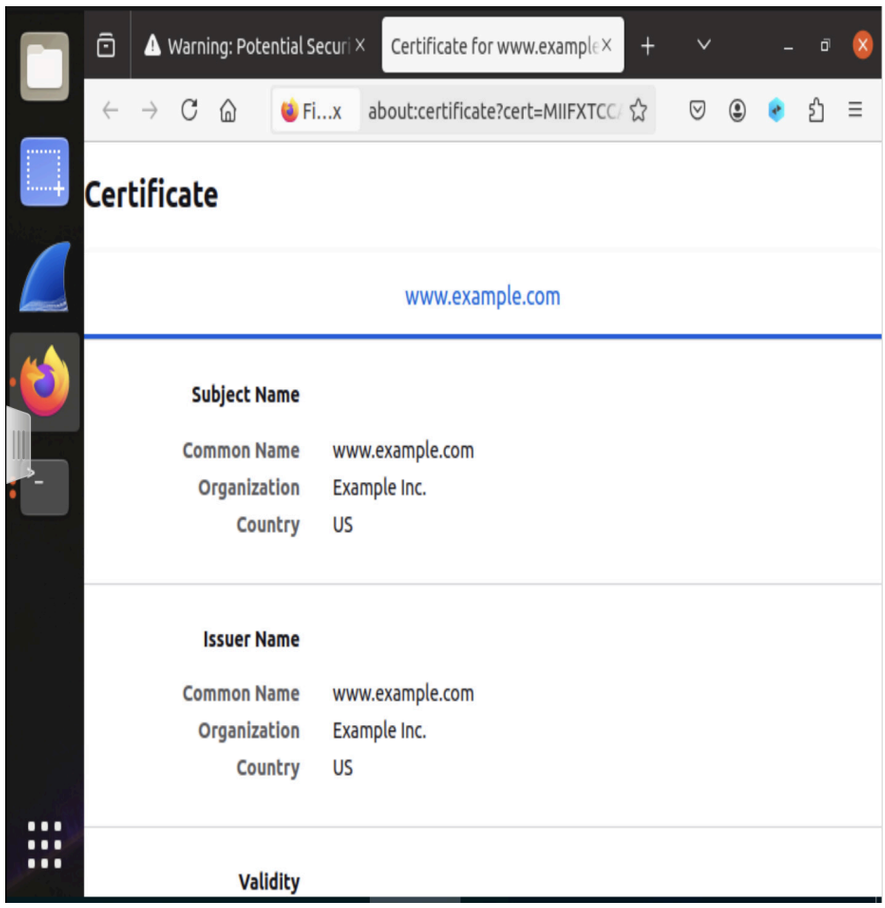
```
[10/03/24]seed@VM:~/.../Labsetup$ sudo nano /etc/apache2/sites-availabl
e/bank32_apache_ssl.conf
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.^C
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.^C
[10/03/24]seed@VM:~/.../Labsetup$ ^Cdo nano /etc/apache2/sites-availabl
e/bank32_apache_ssl.conf
[10/03/24]seed@VM:~/.../Labsetup$ sudo a2ensite bank32_apache_ssl.conf

Site bank32_apache_ssl already enabled
[10/03/24]seed@VM:~/.../Labsetup$ sudo apachectl configtest
Syntax OK
[10/03/24]seed@VM:~/.../Labsetup$ sudo mkdir -p /var/www/bank32
[10/03/24]seed@VM:~/.../Labsetup$ sudo nano /var/www/bank32/index.html
[10/03/24]seed@VM:~/.../Labsetup$ sudo ls -l /etc/ssl/certs/server.crt
-rw-r--r-- 1 root root 1988 Oct  3 01:36 /etc/ssl/certs/server.crt
[10/03/24]seed@VM:~/.../Labsetup$ sudo ls -l /etc/ssl/private/server.ke
y
-rw------- 1 root root 3272 Oct  3 01:36 /etc/ssl/private/server.key
```

```
Oct 03 02:30:17 VM systemd[1]: Starting The Apache HTTP Server...
Oct 03 02:30:17 VM systemd[1]: Started The Apache HTTP Server.

[10/03/24]seed@VM:~/.../demoCA$ sudo openssl req -newkey rsa:4096 -nod\
es -keyout /etc/ssl/private/server.key -x509 -days 365 -out /etc/ssl/ce
rts/server.crt -subj "/CN=www.example.com/O=Example Inc./C=US"
Generating a RSA private key
.....................................................................
.....................................................................
..............++++
.++++
writing new private key to '/etc/ssl/private/server.key'
-----
[10/03/24]seed@VM:~/.../demoCA$ sudo a2ensite bank32_apache_ssl
Site bank32_apache_ssl already enabled
[10/03/24]seed@VM:~/.../demoCA$ sudo systemctl restart apache2
[10/03/24]seed@VM:~/.../demoCA$ sudo nanao /etc/hosts
sudo: nanao: command not found
[10/03/24]seed@VM:~/.../demoCA$ sudo nano /etc/hosts
[10/03/24]seed@VM:~/.../demoCA$
```

Lab. 3

```
 GNU nano 4.8                    /etc/hosts
10.9.0.5        www.SeedLabSQLInjection.com

# For XSS Lab
10.9.0.5        www.xsslabelgg.com
10.9.0.5        www.example32a.com
10.9.0.5        www.example32b.com
10.9.0.5        www.example32c.com
10.9.0.5        www.example60.com
10.9.0.5        www.example70.com

# For CSRF Lab
10.9.0.5        www.csrflabelgg.com
10.9.0.5        www.csrflab-defense.com
10.9.0.105      www.csrflab-attacker.com

# For Shellshock Lab
10.9.0.80       www.seedlab-shellshock.com
10.9.0.80       www.bank32.com
10.9.0.80       neintl.com

127.0.0.1       www.bank32.com
127.0.0.1       www.example.com

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell
```





Task 6: MITM Attack with Compromised CA

Objective:

Demonstrate a successful MITM attack using a compromised CA, where the browser does not raise any security warnings.

Steps:

1. Use the compromised CA to sign a fake certificate for `www.example.com`.

```
openssl req -newkey rsa:2048 -sha256  \
 -keyout fake.key   -out fake.csr  \
 -subj "/CN=www.example.com/O=example Inc./C=US" \
 -passout pass:dees
```

2. Generate certificate for server (using CA root):

```
openssl ca -config openssl.cnf -policy policy_anything \
  -md sha256 -days 3650 \
  -in fake.csr -out fake.crt -batch \
  -cert ca.crt -keyfile ca.key
```

2. Modify Apache config to use this fake certificate for the `example.com` server:

```
        <VirtualHost *:443>
    DocumentRoot /var/www/bank32
    ServerName www.example.com
    DirectoryIndex index.html
```

SSLEngine On
SSLCertificateFile /certs/fake.crt
SSLCertificateKeyFile /certs/fake.key
</VirtualHost>

3. Import the compromised CA certificate into the browser's trusted certificate store.

4. Visit the impersonated site again.

Output:

- The browser should now accept the fake certificate without raising any warnings.

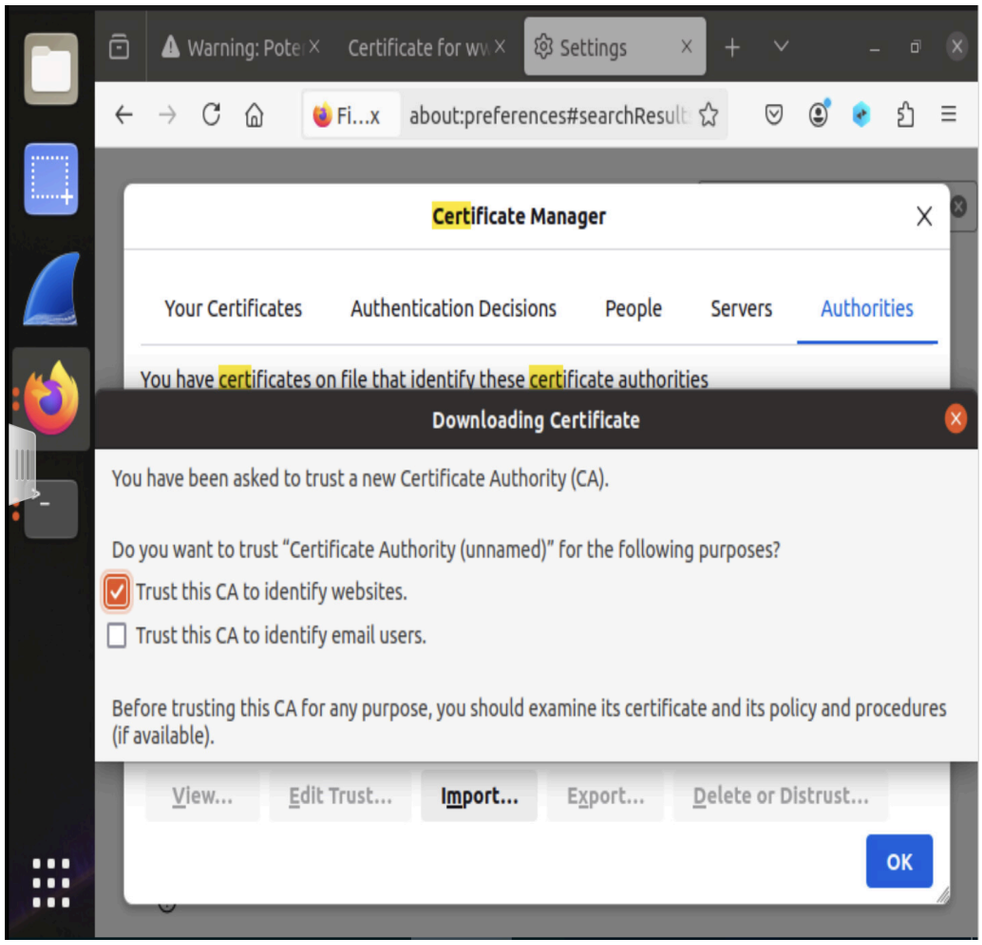
Screenshot:

```
<VirtualHost *:443>
    DocumentRoot /var/www/bank32
    ServerName www.example.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /certs/fake.crt
    SSLCertificateKeyFile /certs/fake.key
</VirtualHost>
```

```
[10/07/24]seed@VM:~/.../certs$ openssl req -newkey rsa:2048 -sha256  \
>      -keyout fake.key   -out fake.csr  \
>      -subj "/CN=www.example.com/O=example Inc./C=US" \
>      -passout pass:dees
Generating a RSA private key
.........................+++++
.............+++++
writing new private key to 'fake.key'
```

```
[10/07/24]seed@VM:~/.../certs$ openssl ca -config openssl.cnf -policy policy_anything -md sha256 -days 3650 -in fake.csr -out fake.crt -batch
 -cert ca.crt -keyfile ca.key
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4099 (0x1003)
        Validity
            Not Before: Oct  8 02:02:30 2024 GMT
            Not After : Oct  6 02:02:30 2034 GMT
        Subject:
            countryName               = US
            organizationName          = example Inc.
            commonName                = www.example.com
```

Lab. 3


Warning: Pote
Certificate for ww
Settings
Fi...x
about:preferences#searchResult
Certificate Manager
Your Certificates
Authentication Decisions
People
Servers
Authorities
You have certificates on file that identify these certificate authorities
Downloading Certificate
You have been asked to trust a new Certificate Authority (CA).
Do you want to trust "Certificate Authority (unnamed)" for the following purposes?
Trust this CA to identify websites.
Trust this CA to identify email users.
Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).
View...
Edit Trust...
Import...
Export...
Delete or Distrust...
OK


https://www.example.com
Hello,
world!