Tampereen yliopisto
Tampere University

COMP.SEC.110-2021-2022-1 Cyber Security II

Public-Key Cryptography

Authors / 28.2.2022, Rev.0:

Rauli Virtanen, rauli.virtanen@tuni.fi

Henri Dyster, henri.dyster@tuni.fi

Table of Contents

1. Introduction

Public key cryptography is the foundation of today's secure communication, but it is subject to man-in-the-middle attacks when one side of communication sends its public key to the other side. The fundamental problem is that there is no easy way to verify the ownership of a public key, i.e., given a public key and its claimed owner information, how do we ensure that the public key is indeed owned by the claimed owner?

The Public Key Infrastructure (PKI) is a practical solution to this problem. In this lab, we learnt how PKI works, how PKI is used to protect the Web and how Man-in-the-middle attacks can be defeated by PKI. This lab also helped us to understand the root of the trust in the public-key infrastructure, and what problems will arise if the root trust is broken.

The lab covers the following topics:
• Public-key encryption, Public-Key Infrastructure (PKI),
• Certificate Authority (CA), X.509 certificate, and root CA,
• Apache, HTTP, and HTTPS,
• Man-in-the-middle attacks [1].

2. Task 1: Becoming a Certificate Authority (CA)

A Certificate Authority (CA) is a trusted entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. A number of commercial CAs are treated as root CAs. Users who want to get digital certificates issued by the commercial CAs need to pay those CAs. In this lab, we created digital certificates, but we didn't pay any commercial CA. We became a root CA ourselves, and then used this CA to issue certificate for others (e.g. servers). In this task, we made ourselves a root CA, and generated a certificate for this CA. Unlike other certificates, which are usually signed by another CA, the root CA's certificates are self-signed. Root CA's certificates are usually pre-loaded into most operating systems, web browsers, and other software that rely on PKI. Root CA's certificates are unconditionally trusted [1].

In this task, we created directories and files as defined in seed-instructions and in openssl.cnf. We made subdirectories and required files by using the following commands in ubuntu20.04-vm terminal (see picture 1). Note: In this task, all command shall be written in ubuntu20.04-vm terminal.

```
[02/27/22]seed@VM:~/Labsetup$ mkdir demoCA
[02/27/22]seed@VM:~/Labsetup$ cd demoCA/
[02/27/22]seed@VM:~/.../demoCA$ mkdir certs
[02/27/22]seed@VM:~/.../demoCA$ mkdir crl
[02/27/22]seed@VM:~/.../demoCA$ mkdir newcerts
[02/27/22]seed@VM:~/.../demoCA$ touch index.txt
[02/27/22]seed@VM:~/.../demoCA$ echo 1000 > serial
[02/27/22]seed@VM:~/.../demoCA$ cd ..
[02/27/22]seed@VM:~/Labsetup$ cp /usr/lib/ssl/openssl.cnf demoCA_openssl.cnf
[02/27/22]seed@VM:~/Labsetup$ nano demoCA_openssl.cnf
[02/27/22]seed@VM:~/Labsetup$
```

Picture 1, The commands given in the terminal.

In the penultimate command we copied openssl.cnf file and we renamed the file to demoCA_openssl.cnf as it was instructed in the seed instructions.

The last command opens demoCA_openssl.cnf file, where we removed # mark from the row highlighted in yellow (see picture 2).

```
  GNU nano 4.8                    demoCA_openssl.cnf
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = ./demoCA              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
unique_subject  = no                    # Set to 'no' to allow creation of
                                        # several certs with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.
```

Picture 2, demoCA_openssl.cnf file.

We uncommented the unique-subject line to allow creation of certifications with the same subject.

Certificate Authority (CA). Next, we generated a self-signed certificate for our CA. This means that this CA is totally trusted, and its certificate will serve as the root certificate [1]. We used the following command (marked in yellow) in the terminal to generate the self-signed certificate for the CA (see picture 3).

```
[02/28/22]seed@VM:~/Labsetup$ openssl req -x509 -newkey rsa:4096 -sha256 -days 3
650 \
> -keyout ca.key -out ca.crt \
> -subj "/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'
-----
[02/28/22]seed@VM:~/Labsetup$
```

Picture 3, Command in the terminal to generate self-signed certificate for the CA.

In the command presented above, we used *-subj* to set the subject information and we used *-passout pass:dees* to set the password to *dees*.

Then we used the following commands (marked in yellow) to look at the decoded content of the X509 certificate and the RSA key (-text means decoding the content into plain text; -noout means not printing out the encoded version):

```
[02/28/22]seed@VM:~/Labsetup$ openssl x509 -in ca.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1f:1b:33:18:a7:91:62:88:f7:bd:4f:e6:fd:ba:c8:84:22:e4:8a:ef
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = www.modelCA.com, O = Model CA LTD., C = US
        Validity
            Not Before: Feb 28 14:53:53 2022 GMT
            Not After : Feb 26 14:53:53 2032 GMT
        Subject: CN = www.modelCA.com, O = Model CA LTD., C = US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:b7:99:fc:dc:52:1f:ad:d0:4c:96:b4:85:c4:95:
                    ec:de:f5:0c:2d:10:67:be:32:c4:79:e9:5d:45:36:
                    b4:1c:52:1c:c5:3d:8f:8d:34:f9:3e:5d:74:17:32:
                    c3:c9:7d:e5:95:a3:0a:b0:ce:eb:12:e3:ad:69:ea:
                    52:4d:0e:ea:9b:f9:05:fb:00:4d:08:c9:19:81:69:
                    05:bd:1c:97:9a:b1:a1:0b:fb:9b:fa:6a:ca:eb:56:
                    1d:d8:af:87:4d:0c:99:21:2d:ba:c2:8f:3a:cb:91:
                    ac:3d:2b:d7:fa:01:f0:cc:37:6b:9c:11:4f:ac:1d:
                    de:66:e0:94:e1:81:92:ab:df:bc:40:93:37:a7:de:
                    a9:44:1d:ae:b3:7a:ee:3c:14:8d:f1:f1:b5:08:b2:
                    70:5c:d8:d0:44:15:b3:d3:2b:02:8d:b5:c6:65:a1:
                    91:41:fe:3f:3b:c1:7f:72:22:7d:f9:bb:d0:dd:a9:
```

```
            c4:6c:47:9e:06:3f:9e:fc:e3:f9:aa:c7:17:82:99:
            b2:f1:77:82:41:85:41:6a:52:31:c9:78:41:e2:15:
            5b:c8:2b:14:c7:b6:3e:e4:61:60:dc:dc:30:9d:af:
            0d:41:ed:10:3a:51:87:d4:97:9b:25:04:9e:54:1d:
            48:38:52:8a:21:10:8d:92:ce:56:1d:07:69:5b:e1:
            ea:57:c2:34:19:46:b8:d8:da:e3:b9:f0:ea:45:1d:
            ca:ef:f2:1c:8b:9d:e5:f1:bb:90:f1:3a:3e:38:6b:
            ac:02:0e:ed:dc:21:da:0a:02:14:4c:91:ca:71:0b:
            59:35:b8:d6:62:a1:c8:ee:cf:bb:46:fa:74:d1:cf:
            18:e4:24:aa:6d:47:f0:f4:1a:2e:4b:45:fe:59:ec:
            9a:00:98:91:36:41:52:0f:b1:f7:ba:61:90:d4:4d:
            68:a6:48:7a:da:c4:86:1c:5a:86:f1:92:84:4a:71:
            c4:e8:c9:7c:83:1f:c5:29:d5:8e:5c:5a:6a:46:1a:
            72:24:f1:bd:10:20:1c:15:de:de:03:e1:29:e9:f4:
            b2:a0:6c:4b:7e:4a:b1:67:97:ae:15:cc:26:9b:90:
            0f:ab:61:2b:1f:7f:10:7a:63:c7:24:c4:2c:7c:a1:
            2f:10:50:bf:0a:9a:a7:dd:a1:3d:e2:93:a0:f1:94:
            79:20:f4:b3:52:f6:e1:80:c1:bf:1a:d8:cf:86:11:
            18:fa:e8:e8:89:5a:38:d9:67:f4:1b:4b:5d:2a:b9:
            d3:97:df:a5:36:77:15:dd:cc:13:72:94:d4:87:60:
            6c:91:8e:59:81:f3:ea:63:48:de:58:9f:59:9c:79:
            39:9f:f1:5d:c9:c3:68:92:52:8e:91:da:f9:ba:e2:
            fa:46:af
        Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            E9:1A:55:DE:22:A8:46:94:44:B9:97:1D:A0:CE:AD:12:7D:ED:AE:0B
        X509v3 Authority Key Identifier:
            keyid:E9:1A:55:DE:22:A8:46:94:44:B9:97:1D:A0:CE:AD:12:7D:ED:AE:0B

        X509v3 Basic Constraints: critical
            CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
    62:3c:91:9c:c0:87:ef:44:e0:36:b1:9e:1b:6d:60:ed:8b:b6:
    a7:05:c8:71:f4:11:9e:78:9e:05:5e:91:ab:22:18:c2:e4:02:
    84:85:f5:d2:5b:da:c1:c0:0b:af:22:b6:33:e0:7d:bb:07:78:
    39:4b:13:b2:9b:86:39:e8:e3:51:24:4c:cd:48:46:0d:a9:23:
    50:5c:c2:18:6a:4d:97:16:7b:6d:0e:2e:4c:d9:4d:44:34:2a:
    ae:6b:c8:34:72:fd:8a:68:0f:d3:a9:6d:bc:7d:04:d2:8d:ea:
    ba:79:c1:6f:a7:d3:90:96:7e:d5:3f:51:d1:fd:69:1f:2c:9e:
    07:5a:15:6b:25:64:7b:e3:92:80:ab:3a:51:ab:62:32:04:c9:
    73:e3:64:0b:f8:15:1a:b4:a8:45:c8:35:05:db:5f:53:1b:91:
    e0:3e:45:65:d7:d2:1b:b3:75:f2:0a:bf:07:8a:be:6f:7f:5a:
    fd:63:83:29:a8:9f:a0:ba:70:95:d3:28:5c:d8:7e:18:65:5a:
    a9:4b:c5:e6:d4:e5:8f:5f:80:29:a6:96:f4:82:6c:6d:20:79:
    c3:c8:5a:ac:6e:61:22:fc:8d:cf:3a:18:93:61:12:21:0d:ce:
```

```
        c5:d4:d8:74:a9:2a:ce:da:34:dd:d8:e8:87:bf:21:95:af:01:
        0a:90:82:de:f8:9d:c4:a0:ef:f8:5a:9e:c3:d6:d4:99:84:ca:
        7e:0f:b7:e9:2f:c9:0c:43:41:d4:5d:da:d1:b8:6a:e9:57:59:
        ed:dd:56:25:e3:b8:3e:cc:e4:3c:38:08:cf:1b:b8:22:61:72:
        ca:d9:c9:74:c1:5d:a4:1b:c6:8e:51:88:d5:c4:b7:ac:b9:2f:
        00:b7:83:54:07:c3:b3:c5:9c:bb:97:12:00:cd:80:87:34:be:
        2e:9c:6e:c7:72:ea:ab:96:70:64:f6:6c:cd:8b:f3:b9:95:91:
        14:f0:ed:0b:58:ff:79:c7:4f:78:dd:50:9a:da:39:63:68:de:
        d2:34:f9:8f:63:77:0b:de:2e:84:e6:84:d7:17:33:97:6e:7d:
        89:eb:02:37:8d:f7:03:1b:a2:09:40:a2:57:e4:e2:02:c3:a4:
        2a:eb:3c:67:71:71:11:36:69:27:80:79:5d:45:ea:81:c2:ff:
        03:11:7f:75:e7:ff:d4:88:f9:ad:1a:2a:76:d6:6b:3a:6c:5f:
        7c:26:e0:78:12:88:46:ba:8a:71:59:30:04:ad:80:30:69:e5:
        04:92:f2:02:95:24:cf:95:e0:f4:c2:d3:48:87:8b:40:17:93:
        a6:42:2a:7b:6b:e0:6b:4f:48:c1:03:17:cb:d6:d7:80:97:18:
        1c:0b:4d:6b:13:a5:d1:0f
[02/28/22]seed@VM:~/Labsetup$


[02/28/22]seed@VM:~/Labsetup$ openssl rsa -in ca.key -text -noout
Enter pass phrase for ca.key:
RSA Private-Key: (4096 bit, 2 primes)
modulus:
    00:b7:99:fc:dc:52:1f:ad:d0:4c:96:b4:85:c4:95:
    ec:de:f5:0c:2d:10:67:be:32:c4:79:e9:5d:45:36:
    b4:1c:52:1c:c5:3d:8f:8d:34:f9:3e:5d:74:17:32:
    c3:c9:7d:e5:95:a3:0a:b0:ce:eb:12:e3:ad:69:ea:
    52:4d:0e:ea:9b:f9:05:fb:00:4d:08:c9:19:81:69:
    05:bd:1c:97:9a:b1:a1:0b:fb:9b:fa:6a:ca:eb:56:
    1d:d8:af:87:4d:0c:99:21:2d:ba:c2:8f:3a:cb:91:
    ac:3d:2b:d7:fa:01:f0:cc:37:6b:9c:11:4f:ac:1d:
    de:66:e0:94:e1:81:92:ab:df:bc:40:93:37:a7:de:
    a9:44:1d:ae:b3:7a:ee:3c:14:8d:f1:f1:b5:08:b2:
    70:5c:d8:d0:44:15:b3:d3:2b:02:8d:b5:c6:65:a1:
    91:41:fe:3f:3b:c1:7f:72:22:7d:f9:bb:d0:dd:a9:
    c4:6c:47:9e:06:3f:9e:fc:e3:f9:aa:c7:17:82:99:
    b2:f1:77:82:41:85:41:6a:52:31:c9:78:41:e2:15:
    5b:c8:2b:14:c7:b6:3e:e4:61:60:dc:dc:30:9d:af:
    0d:41:ed:10:3a:51:87:d4:97:9b:25:04:9e:54:1d:
    48:38:52:8a:21:10:8d:92:ce:56:1d:07:69:5b:e1:
    ea:57:c2:34:19:46:b8:d8:da:e3:b9:f0:ea:45:1d:
    ca:ef:f2:1c:8b:9d:e5:f1:bb:90:f1:3a:3e:38:6b:
    ac:02:0e:ed:dc:21:da:0a:02:14:4c:91:ca:71:0b:
    59:35:b8:d6:62:a1:c8:ee:cf:bb:46:fa:74:d1:cf:
    18:e4:24:aa:6d:47:f0:f4:1a:2e:4b:45:fe:59:ec:
    9a:00:98:91:36:41:52:0f:b1:f7:ba:61:90:d4:4d:
    68:a6:48:7a:da:c4:86:1c:5a:86:f1:92:84:4a:71:
```

```
    c4:e8:c9:7c:83:1f:c5:29:d5:8e:5c:5a:6a:46:1a:
    72:24:f1:bd:10:20:1c:15:de:de:03:e1:29:e9:f4:
    b2:a0:6c:4b:7e:4a:b1:67:97:ae:15:cc:26:9b:90:
    0f:ab:61:2b:1f:7f:10:7a:63:c7:24:c4:2c:7c:a1:
    2f:10:50:bf:0a:9a:a7:dd:a1:3d:e2:93:a0:f1:94:
    79:20:f4:b3:52:f6:e1:80:c1:bf:1a:d8:cf:86:11:
    18:fa:e8:e8:89:5a:38:d9:67:f4:1b:4b:5d:2a:b9:
    d3:97:df:a5:36:77:15:dd:cc:13:72:94:d4:87:60:
    6c:91:8e:59:81:f3:ea:63:48:de:58:9f:59:9c:79:
    39:9f:f1:5d:c9:c3:68:92:52:8e:91:da:f9:ba:e2:
    fa:46:af
publicExponent: 65537 (0x10001)
privateExponent:
    32:bf:c2:b0:17:97:20:11:3a:8a:51:9d:c4:07:f8:
    f0:6b:e5:90:b3:3f:29:c8:98:9c:66:0f:dc:e8:d1:
    02:1f:04:4a:d4:c8:50:2c:bb:54:2c:ba:c1:0a:cc:
    29:6e:be:3e:0d:52:10:2b:31:b9:07:eb:d0:21:ef:
    db:d2:ea:b0:70:35:e1:fa:64:50:b0:5d:77:72:d3:
    1e:41:bf:fa:07:6d:4f:85:9d:c8:ab:2a:29:c0:a5:
    01:57:0b:5c:15:c6:61:f0:64:f6:10:67:1d:0b:c8:
    03:24:92:4f:4d:a8:04:9a:b1:08:f3:94:05:2c:40:
    64:49:61:7d:af:b6:1e:01:fa:e4:12:8d:48:be:78:
    25:16:db:4a:4b:09:7c:db:e5:9c:61:c6:3b:80:2f:
    24:ca:dd:4e:64:7d:4f:3a:4d:3b:64:6a:51:7f:46:
    4a:6a:7f:8a:01:e3:3b:66:23:52:1b:43:11:72:32:
    94:07:7b:bb:98:6c:56:d4:a1:17:3e:6e:d6:2f:ab:
    d1:a8:c9:fd:58:5b:0b:8b:6a:ad:98:ab:05:56:08:
    07:9d:38:e2:37:12:cd:47:c9:61:63:0f:a1:16:58:
    e0:c7:29:3c:25:7b:19:1d:4a:ca:01:ab:cd:ee:46:
    1f:23:59:80:e8:35:69:c5:d4:f1:b3:31:c1:2f:01:
    fa:05:dd:9f:45:b7:f4:25:dd:0c:6f:28:20:77:2d:
    d7:f9:2c:e3:16:0b:85:c5:03:84:65:ae:72:4d:4e:
    03:45:06:2a:3f:d3:2b:44:e3:c7:19:25:60:0f:d3:
    b2:95:1a:9f:2f:25:f1:72:d6:da:77:d2:aa:ef:e0:
    32:2a:73:2f:3c:75:9b:09:b1:64:e8:19:bc:1d:ae:
    42:86:5e:be:3f:28:f7:f1:4b:dc:c5:d8:91:2c:42:
    e8:43:a3:2a:24:bf:fa:79:b9:2a:ab:f6:d6:95:ac:
    42:f0:77:23:e9:a6:14:f6:bf:2f:bf:f4:09:e4:a9:
    77:62:b7:67:b1:fc:e9:9c:e3:36:fc:cb:ef:20:32:
    f1:85:b9:61:27:09:da:36:6a:ca:a7:73:5c:24:59:
    86:ef:88:91:26:50:5a:47:5a:9a:0b:8d:c0:16:58:
    50:7f:b5:e1:62:ac:6f:2a:1f:6a:8f:76:a2:09:b8:
    38:29:1b:0e:bf:96:15:1a:ba:50:f1:28:7f:59:f3:
    df:da:08:38:24:99:f6:38:d4:b7:8d:d9:a6:b7:67:
    b1:b1:6e:5b:42:ac:a7:c2:be:40:8e:a4:01:35:4a:
    3c:e9:51:22:b1:f5:fd:5f:26:37:ae:0a:20:48:39:
```

```
    58:bc:8a:7f:66:6c:a1:ee:32:37:41:53:19:56:4b:
    49:b1
prime1:
    00:e7:6a:92:e3:6f:66:72:94:91:99:c0:01:e8:77:
    94:1f:6c:00:3f:94:9a:3a:ed:f4:2d:68:52:e9:77:
    cf:b4:9f:a6:d9:9e:ae:17:1f:14:8c:15:54:6b:26:
    91:70:fa:48:a3:28:cb:58:8d:a9:57:7b:6c:71:5d:
    d4:d3:86:49:aa:34:b3:47:cf:c1:c6:6f:ea:a3:20:
    3a:d6:34:a8:01:f0:c0:f5:35:3d:dc:32:b9:ed:72:
    12:f9:fb:e4:db:34:12:c8:fc:d0:41:ad:8d:15:fb:
    e6:cb:58:89:ea:c5:6d:18:d0:e6:09:e7:6a:37:f1:
    a4:ab:4e:21:4b:5a:a4:26:5a:aa:7c:a7:e4:52:3a:
    d4:d4:8d:39:d1:03:3a:03:78:00:9f:4b:00:c1:fd:
    0e:aa:a8:ba:41:54:79:23:27:a0:01:e4:f5:9c:39:
    51:e3:7e:9a:35:e0:45:7e:14:8c:5b:bd:a3:13:29:
    bb:67:7f:d7:8d:18:8d:9c:e3:36:a3:00:d5:1e:29:
    00:07:71:35:bb:60:5f:ad:da:1c:6a:f8:eb:e6:d2:
    66:37:02:66:d8:f0:1e:13:55:1a:12:6f:95:9c:7f:
    4a:77:10:8a:3b:80:35:a7:c0:8d:41:52:2d:c8:50:
    e9:bf:b8:03:0a:11:ef:a8:3f:f3:79:89:60:a6:f8:
    7e:19
prime2:
    00:cb:1b:14:11:de:a0:20:79:4b:78:36:35:98:4d:
    23:73:50:af:0d:72:3a:bc:86:15:36:83:48:0d:d9:
    e9:ed:12:d9:a2:fe:d1:07:3f:d5:eb:63:19:ec:78:
    a5:42:34:40:a2:74:8c:16:5b:24:ff:32:81:aa:2b:
    20:1f:f5:03:78:95:b1:bb:50:61:ef:9d:12:84:e5:
    28:7d:7a:8c:8d:98:ea:4f:e3:d9:d6:97:d8:89:5e:
    5d:f7:0a:9d:8f:4c:5e:06:71:a9:84:7b:dd:13:d7:
    78:61:3b:86:ab:b1:73:a9:d7:f3:d8:b4:20:01:43:
    18:de:82:92:0c:a3:8b:f3:d3:c6:9a:a2:62:9e:f2:
    b8:02:b5:90:5f:3d:4d:7f:ca:61:da:07:c5:d3:a1:
    98:54:c7:a9:1a:b8:f5:49:6d:d3:37:9b:09:c0:a7:
    95:13:8d:0c:01:b7:a4:41:32:a0:3e:3a:fc:fc:b4:
    e1:3a:cb:71:e1:13:34:0d:fc:cf:34:13:bb:8e:42:
    21:b7:73:6c:7b:29:36:b7:5c:5e:2c:9d:c8:5a:fd:
    8f:89:10:85:86:0a:21:03:af:5c:94:b3:8c:60:f0:
    c3:f3:f0:bd:e7:bf:3f:6f:02:a6:3d:34:0a:d8:0f:
    4a:dd:0c:ed:14:43:48:a5:b2:b1:d5:29:08:8b:d7:
    f4:07
exponent1:
    5a:d4:76:c0:f4:09:96:f1:7f:50:84:8f:7b:29:ec:
    26:85:22:77:d2:20:d2:fe:70:b7:9d:d9:e5:2f:14:
    84:45:a1:9c:8a:ee:b3:be:8c:37:0b:6c:2d:fe:5e:
    59:a4:b2:fb:ef:58:18:f2:c9:43:bf:fa:e9:68:35:
    cf:ac:46:9f:9a:bd:bf:72:e9:10:b4:fe:b8:76:3f:
```

```
    01:b2:7b:3a:ff:bf:0c:bf:8f:ff:2c:9d:d3:77:d1:
    c0:f2:c0:79:d5:2f:86:59:cc:77:ea:e1:94:7e:61:
    b0:f0:98:79:60:72:18:aa:6d:8f:f7:97:b6:4b:8f:
    21:79:b2:11:bc:8a:ef:4e:e8:d9:b2:a0:28:32:55:
    34:f4:15:7c:57:32:df:07:ab:de:d9:f6:7b:2d:93:
    23:22:c4:b0:3d:b2:aa:3c:b4:4d:ea:0b:08:fd:6a:
    89:cb:38:45:eb:37:fb:6b:9f:47:a6:e0:29:f6:58:
    4b:20:5f:7f:4e:e2:ce:70:54:12:98:92:0d:9b:74:
    1a:77:b1:ea:50:c2:5d:1f:5d:c6:12:db:ac:a6:6a:
    5e:2f:00:25:27:e1:f6:1e:91:f4:28:7d:e5:86:04:
    b9:44:f1:ac:b6:fe:d8:4f:f8:61:6a:1a:f4:23:e6:
    0b:4c:f3:4c:71:8b:ca:53:07:82:6d:8c:55:1c:f5:
    89
exponent2:
    35:39:55:5b:11:71:f8:d0:90:5c:62:28:4d:4e:f9:
    99:40:6d:7f:22:8e:0d:d5:3d:3d:d9:cd:4a:03:ee:
    ef:37:5f:5c:fd:55:9d:86:b6:f2:46:38:06:e5:de:
    b8:a8:a4:bf:6d:b7:40:2c:86:57:71:ce:d4:df:14:
    3e:0b:a0:eb:3a:9f:26:ac:fb:dd:24:d6:33:89:31:
    c1:20:1a:31:08:ff:6e:0c:11:8b:d0:a2:e7:ec:98:
    c7:22:3b:03:d0:49:5b:7a:f1:a7:4c:26:35:0a:e3:
    fa:e6:f4:75:8d:bc:f7:a7:25:b9:86:0a:55:0a:56:
    3c:09:0f:68:cc:7a:e2:5c:7e:d1:0a:f8:b9:1a:75:
    ea:0f:d6:53:7e:e7:0e:33:e4:cc:93:f8:dd:6a:9d:
    56:aa:33:c0:52:f3:c4:58:49:5f:6a:fb:75:46:f5:
    23:f8:74:4b:06:ea:be:43:1c:06:f2:c2:55:1e:b9:
    5f:04:ed:44:c4:19:82:08:db:eb:93:b8:34:a1:53:
    e0:95:18:ba:75:a4:67:0c:87:89:3d:97:c1:70:6c:
    ea:c8:2f:95:5e:96:24:37:12:15:cd:af:5d:bf:fa:
    39:b4:be:af:3a:48:cf:38:6c:83:e1:5d:6e:1d:23:
    cb:25:96:03:17:56:b4:45:4d:a0:c4:6d:fe:e7:8a:
    85
coefficient:
    36:18:8a:03:e4:60:99:67:f6:ce:8d:3c:7b:37:dd:
    26:43:8d:46:b6:2b:0c:29:4f:db:e4:28:39:41:d3:
    b0:42:4d:1e:5d:23:61:1d:c6:a5:5c:cf:56:f5:ca:
    48:6b:96:5a:53:4c:11:ee:61:ea:9e:b1:61:85:af:
    da:af:5a:3f:0f:28:b5:c4:94:81:bd:70:da:d9:15:
    24:1f:a7:23:06:07:e6:e0:f2:4f:cc:8d:f9:67:8e:
    df:56:b4:d0:8d:0b:4f:e3:3e:94:a5:45:15:bf:2a:
    41:84:90:9d:b6:68:ad:ba:da:07:39:c1:85:a3:ed:
    ee:64:59:5e:0d:75:39:1a:94:dc:d2:8c:96:0e:af:
    1a:1f:c0:92:bf:be:6b:99:57:26:c1:8d:b7:89:94:
    8c:2a:85:75:22:ef:82:37:72:84:d2:b6:40:29:87:
    01:17:10:ec:50:23:35:ab:a5:4a:17:71:0b:17:66:
    aa:2b:6a:65:75:0a:ce:c0:34:3f:60:cc:7b:5a:82:
```

```
    05:93:31:94:ec:3e:df:28:99:ec:11:da:86:60:3d:
    47:99:3c:89:ee:52:c2:6f:16:a7:3b:04:0d:f6:fe:
    61:69:f4:6a:09:b3:e2:4e:70:ad:cc:95:ff:b0:f1:
    75:8f:2f:71:fc:c5:5d:f5:3e:fd:aa:99:e1:d6:52:
    1b
[02/28/22]seed@VM:~/Labsetup$
```

• What part of the certificate indicates this is a CA's certificate?

For example, the certificate contains Issuer and Validity information (see yellow marked rows).

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1f:1b:33:18:a7:91:62:88:f7:bd:4f:e6:fd:ba:c8:84:22:e4:8a:ef
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = www.modelCA.com, O = Model CA LTD., C = US
        Validity
            Not Before: Feb 28 14:53:53 2022 GMT
            Not After : Feb 26 14:53:53 2032 GMT
        Subject: CN = www.modelCA.com, O = Model CA LTD., C = US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
X509v3 Basic Constraints: critical
                CA:TRUE
```

• What part of the certificate indicates this is a self-signed certificate?

For example, Issuer and Subject are same in the self-signed certificate (see yellow marked rows).

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1f:1b:33:18:a7:91:62:88:f7:bd:4f:e6:fd:ba:c8:84:22:e4:8a:ef
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = www.modelCA.com, O = Model CA LTD., C = US
        Validity
            Not Before: Feb 28 14:53:53 2022 GMT
            Not After : Feb 26 14:53:53 2032 GMT
        Subject: CN = www.modelCA.com, O = Model CA LTD., C = US
```

• In the RSA algorithm, we have a public exponent e, a private exponent d, a modulus n, and two secret numbers p and q, such that n = pq. Please identify the values for these elements in your certificate and key files.

See the following printouts of the terminal commands above: openssl x509 -in ca.crt -text -noout and openssl rsa -in ca.key -text -noout. The asked elements have been marked in grey.

3. Task 2: Generating a Certificate Request for Your Web Server

In this task, a company called student22.com (web server) wants to get a public key certificate from our CA. First, we generated a Certificate Signing Request (CSR), which basically includes the company's public key and identity information. The CSR will be sent to the CA, who will verify the identity information in the request, and then generate a certificate [1].

The command to generate a CSR was quite similar with the one we used in creating the self-signed certificate for the CA. The only difference is the -x509 option. Without the option, command generates a request. With the option, command generates a self-signed certificate [1]. The following command (marked in yellow) generate a CSR for www.student22.com (see picture 4). Note: In this task, all command shall be written in ubuntu20.04-vm terminal.

```
[02/28/22]seed@VM:~/Labsetup$ openssl req -newkey rsa:4096 -sha256 \
> -keyout server.key -out server.csr \
> -subj "/CN=www.student22.com/O=Student22 Inc./C=US" \
> -passout pass:dees
Generating a RSA private key
.............++++
.........................................++++
writing new private key to 'server.key'
-----
[02/28/22]seed@VM:~/Labsetup$
```

Picture 4, Command in the terminal to generate the CSR for www.student22.com.

The command will generate a pair of public/private key and then create a certificate signing request from the public key. We used the following commands (marked in yellow) to look at the decoded content of the CSR and private key file:

```
[02/28/22]seed@VM:~/Labsetup$ openssl req -in server.csr -text -noout
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: CN = www.student22.com, O = Student22 Inc., C = US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:b1:7f:f7:40:98:c8:dd:bc:fa:cf:7f:66:7d:57:
                    0d:2b:43:f2:53:51:b4:68:ff:52:6c:bf:aa:16:f7:
                    4e:8e:17:c9:6f:47:e2:ab:3e:1b:e0:8b:f8:fa:16:
                    f0:11:d7:7c:d7:15:1d:29:5c:ea:b2:f2:de:5e:55:
                    f2:2f:f9:8d:aa:d8:00:74:23:d0:0a:bd:bd:a6:6a:
```

```
            02:1f:54:e4:31:87:e9:9a:a6:27:3a:64:f4:88:93:
            e7:78:b9:92:ec:37:92:77:a8:d7:90:51:ee:0a:02:
            d7:c0:29:8a:0c:5c:27:5b:66:f4:52:1b:b0:81:b9:
            fb:de:22:f9:89:c7:42:a8:cb:cf:ce:91:5d:34:aa:
            aa:a1:6e:51:de:32:20:d5:8b:35:4e:4d:4b:ff:be:
            4d:18:c5:9e:e4:b0:7f:82:70:8e:f5:27:fc:e1:e0:
            b9:1f:56:5b:aa:b2:ab:b9:39:f2:12:55:f0:7a:67:
            8d:e9:8a:98:76:d4:b1:6c:e8:78:e4:18:24:77:c2:
            ec:ef:8b:b5:94:ed:82:c3:d5:e9:c4:43:d4:1a:0f:
            e2:35:12:bb:74:13:8b:30:8d:08:6c:34:55:9c:e2:
            4a:7f:be:c8:8a:8e:9b:19:d1:ae:45:39:22:64:09:
            fb:10:00:75:a6:f5:c5:c2:d7:ee:3d:54:8e:36:00:
            dc:bd:76:cc:e0:ee:92:7f:f3:91:bb:2c:71:1b:07:
            fa:ca:d2:af:28:56:f4:4a:10:65:5c:4a:36:77:9a:
            95:25:06:28:27:b0:e5:02:40:5f:bd:4e:f0:8d:41:
            e4:41:ea:46:b1:ed:9f:70:57:49:7f:ae:f3:51:a9:
            46:dc:fc:30:b0:b1:3e:ea:3c:dc:e5:2f:ee:b6:42:
            40:00:46:b1:a9:01:42:cd:76:f2:c3:ee:b3:86:06:
            1a:a9:bf:fd:1a:0b:cc:c3:31:db:0d:60:27:55:e8:
            6e:3b:90:c8:b0:dd:87:39:1f:b8:75:16:f0:5b:65:
            d3:51:7c:43:aa:35:de:71:64:2e:e5:e1:45:ee:41:
            96:66:89:6d:02:ac:de:09:a5:58:2f:af:20:a2:a0:
            57:c9:d5:44:eb:4a:33:56:ac:48:f8:40:9a:69:cd:
            e3:11:90:ed:1b:dc:98:50:64:c2:17:12:43:64:bf:
            41:90:95:df:69:70:c9:f8:13:a8:d7:f3:22:15:3e:
            76:00:a3:59:c1:3f:cb:69:79:a3:ad:c9:3c:68:2e:
            e7:04:ca:2d:41:26:e7:18:fb:a0:2a:c9:ac:ff:65:
            8b:ce:6d:d1:24:98:56:ba:6f:85:d2:a6:fd:9a:1b:
            a7:de:89:3c:09:b1:45:42:48:d7:35:bf:25:15:48:
            7c:8e:fd
        Exponent: 65537 (0x10001)
    Attributes:
        a0:00
Signature Algorithm: sha256WithRSAEncryption
    9e:6b:ae:49:77:45:ce:d7:ba:1b:a3:e7:f3:cd:f7:0a:5d:90:
    4d:0f:56:6f:8f:dd:9c:26:a2:57:2b:84:20:26:b7:8a:9c:fb:
    2d:0c:ba:76:5c:20:34:53:76:36:f2:ec:bb:4b:5d:d2:7e:05:
    c0:4e:17:6d:23:64:04:a5:13:41:7b:96:d1:de:16:0c:2f:95:
    89:4e:f8:db:c0:f7:83:82:d9:de:65:ce:62:3e:dd:cf:b9:2c:
    77:98:14:35:cd:fd:56:ee:7c:65:7d:9f:a1:43:70:01:0b:0a:
    ed:5a:8d:62:e2:cc:47:dc:3c:77:b2:f8:58:a7:c9:47:0c:dd:
    d1:98:01:9d:c8:64:6a:77:1e:cb:3f:a1:50:f3:36:39:ea:d2:
    82:18:8a:a6:4d:bd:8f:69:2d:92:02:a6:32:a3:f8:7a:79:e5:
    28:53:c4:f1:4a:5d:14:6a:f7:36:5f:09:1c:46:e0:8a:1c:c8:
    3f:7c:5d:89:2c:a6:10:96:d8:66:0c:18:a5:a2:21:c3:c8:bb:
    98:55:eb:23:5f:5b:48:18:55:28:c2:8a:0f:b4:0b:d5:2a:43:
```

```
        83:8a:f2:13:cd:3d:19:4d:8c:46:ec:78:17:3a:95:87:b2:e0:
        eb:9b:e2:10:ea:2c:52:ae:ba:81:97:76:60:a1:d3:ca:56:eb:
        3a:2d:1e:ba:5b:1b:1e:5c:e4:64:d2:eb:1e:47:bb:71:cb:55:
        62:40:26:55:13:5c:ef:ff:59:6d:e1:94:13:75:36:99:61:3c:
        e1:5a:ce:e9:a5:67:b9:bd:6f:7e:b0:eb:3f:b6:ea:47:14:8e:
        a2:55:31:d7:ed:89:ce:95:f2:f8:b2:12:2a:0b:d8:fc:73:d2:
        43:05:9d:06:07:aa:ea:c8:d8:0b:24:17:07:7d:60:02:82:26:
        1c:7c:2e:b3:fd:9d:22:a7:7c:ca:75:58:64:35:e2:23:ae:2e:
        97:9a:de:dc:97:e6:d3:d4:65:9c:e0:b9:09:a0:80:fb:18:7b:
        69:5f:d1:ba:3d:68:7e:a3:e4:0a:b9:e1:3f:df:61:19:45:cc:
        d2:3a:f6:e8:7d:03:3e:4e:29:d6:db:90:15:40:7a:bd:fe:bf:
        61:44:1e:f0:34:dd:8b:63:b5:26:b6:2a:2c:ad:86:9b:f5:57:
        66:55:3e:43:6b:6b:2d:9b:f1:b5:77:f8:a6:5e:d8:b1:5e:ca:
        48:e0:b4:c0:e5:15:95:cb:34:5c:c2:4d:32:56:11:44:0b:4a:
        f6:2a:08:d7:04:2a:1e:a2:cc:17:40:77:f2:c5:c3:11:9b:7c:
        2f:72:3d:c7:b5:e3:4e:2c:9b:4d:cb:2c:31:78:e3:cf:b6:73:
        d6:a7:74:92:37:5f:1f:e8
[02/28/22]seed@VM:~/Labsetup$


[02/28/22]seed@VM:~/Labsetup$ openssl rsa -in server.key -text -noout
Enter pass phrase for server.key:
RSA Private-Key: (4096 bit, 2 primes)
modulus:
    00:b1:7f:f7:40:98:c8:dd:bc:fa:cf:7f:66:7d:57:
    0d:2b:43:f2:53:51:b4:68:ff:52:6c:bf:aa:16:f7:
    4e:8e:17:c9:6f:47:e2:ab:3e:1b:e0:8b:f8:fa:16:
    f0:11:d7:7c:d7:15:1d:29:5c:ea:b2:f2:de:5e:55:
    f2:2f:f9:8d:aa:d8:00:74:23:d0:0a:bd:bd:a6:6a:
    02:1f:54:e4:31:87:e9:9a:a6:27:3a:64:f4:88:93:
    e7:78:b9:92:ec:37:92:77:a8:d7:90:51:ee:0a:02:
    d7:c0:29:8a:0c:5c:27:5b:66:f4:52:1b:b0:81:b9:
    fb:de:22:f9:89:c7:42:a8:cb:cf:ce:91:5d:34:aa:
    aa:a1:6e:51:de:32:20:d5:8b:35:4e:4d:4b:ff:be:
    4d:18:c5:9e:e4:b0:7f:82:70:8e:f5:27:fc:e1:e0:
    b9:1f:56:5b:aa:b2:ab:b9:39:f2:12:55:f0:7a:67:
    8d:e9:8a:98:76:d4:b1:6c:e8:78:e4:18:24:77:c2:
    ec:ef:8b:b5:94:ed:82:c3:d5:e9:c4:43:d4:1a:0f:
    e2:35:12:bb:74:13:8b:30:8d:08:6c:34:55:9c:e2:
    4a:7f:be:c8:8a:8e:9b:19:d1:ae:45:39:22:64:09:
    fb:10:00:75:a6:f5:c5:c2:d7:ee:3d:54:8e:36:00:
    dc:bd:76:cc:e0:ee:92:7f:f3:91:bb:2c:71:1b:07:
    fa:ca:d2:af:28:56:f4:4a:10:65:5c:4a:36:77:9a:
    95:25:06:28:27:b0:e5:02:40:5f:bd:4e:f0:8d:41:
    e4:41:ea:46:b1:ed:9f:70:57:49:7f:ae:f3:51:a9:
    46:dc:fc:30:b0:b1:3e:ea:3c:dc:e5:2f:ee:b6:42:
    40:00:46:b1:a9:01:42:cd:76:f2:c3:ee:b3:86:06:
```

```
    1a:a9:bf:fd:1a:0b:cc:c3:31:db:0d:60:27:55:e8:
    6e:3b:90:c8:b0:dd:87:39:1f:b8:75:16:f0:5b:65:
    d3:51:7c:43:aa:35:de:71:64:2e:e5:e1:45:ee:41:
    96:66:89:6d:02:ac:de:09:a5:58:2f:af:20:a2:a0:
    57:c9:d5:44:eb:4a:33:56:ac:48:f8:40:9a:69:cd:
    e3:11:90:ed:1b:dc:98:50:64:c2:17:12:43:64:bf:
    41:90:95:df:69:70:c9:f8:13:a8:d7:f3:22:15:3e:
    76:00:a3:59:c1:3f:cb:69:79:a3:ad:c9:3c:68:2e:
    e7:04:ca:2d:41:26:e7:18:fb:a0:2a:c9:ac:ff:65:
    8b:ce:6d:d1:24:98:56:ba:6f:85:d2:a6:fd:9a:1b:
    a7:de:89:3c:09:b1:45:42:48:d7:35:bf:25:15:48:
    7c:8e:fd
publicExponent: 65537 (0x10001)
privateExponent:
    19:b3:e0:07:d8:97:ed:93:f9:f5:a6:0b:1f:47:45:
    be:e8:1d:e4:f6:c5:db:16:6d:c7:15:91:f9:b5:62:
    08:ad:65:1b:da:1a:f6:95:0d:d0:5c:34:77:72:6e:
    dd:06:89:2b:54:01:5f:05:f7:6e:66:40:d1:04:3d:
    07:b9:69:4c:3d:ef:02:e2:54:b3:d9:79:e6:e9:1e:
    d9:8e:9f:64:94:bb:bd:0b:26:4f:0c:63:64:8b:96:
    4f:23:00:25:78:7e:9b:19:35:ed:ce:0f:5c:1f:c1:
    42:30:41:af:3c:80:ee:79:60:13:ab:b2:c9:76:37:
    04:99:a7:ec:cd:8f:d7:85:7b:d7:fb:b0:b5:79:a9:
    2f:ab:1b:eb:c2:f2:82:39:a0:44:70:68:fa:e5:81:
    a0:7e:ad:5d:8c:39:4a:98:25:fb:56:09:06:64:56:
    2a:35:85:1c:98:b6:c9:21:9c:ee:0e:c7:bd:d1:d4:
    f8:e6:7a:e1:6a:37:2f:00:12:d1:64:b5:93:2a:04:
    c9:43:3f:ce:0a:fe:8e:81:21:52:ac:47:c0:66:60:
    69:33:07:35:ed:c6:90:07:3c:1a:ff:11:0e:a0:ae:
    0d:05:81:60:9e:f9:d2:a9:4c:44:f5:db:07:05:72:
    f7:84:39:85:2d:bd:af:53:39:4a:64:3e:1b:8f:d0:
    66:d9:26:06:61:d2:04:6d:b3:50:52:46:ec:8e:99:
    34:d0:b1:4f:a1:9d:81:ce:59:f8:4e:d9:91:ce:48:
    eb:f7:ab:4d:20:86:05:de:e8:9b:c0:f1:10:b1:1f:
    e9:b5:7d:67:b3:b0:4a:af:ac:be:3a:0e:43:34:41:
    3c:cc:f9:a6:98:d9:a0:ec:e3:d8:0c:7b:9f:2f:92:
    51:78:fb:33:29:46:f9:20:37:a7:a9:d9:74:e4:50:
    ca:e7:09:d6:19:0f:ce:6c:ad:77:c8:70:97:60:02:
    67:e6:8a:52:b5:43:76:09:93:19:5f:86:c7:4c:32:
    0d:77:09:e4:19:9a:75:63:d7:1c:c2:c3:f9:64:2a:
    61:29:45:15:a6:ce:4e:a9:bf:5e:d9:6a:a2:71:43:
    e5:f9:f2:41:ab:bd:ee:07:52:38:90:3b:76:dd:77:
    3f:74:55:43:e0:aa:c3:61:52:1c:62:b2:7f:30:88:
    5b:89:64:70:e9:c4:f4:4b:ab:6b:34:1d:1a:a9:24:
    06:24:e4:1e:33:dc:38:0d:72:33:56:88:9e:f9:e6:
    69:d3:54:d2:bc:0a:00:48:ba:0f:3d:4c:c4:06:c1:
```

```
    08:36:6c:6d:be:57:35:85:77:ce:28:67:da:06:8f:
    41:58:76:da:30:6a:3a:98:46:12:86:3d:52:2c:01:56:79
prime1:
    00:e7:51:03:68:55:9b:d8:d9:81:1a:07:05:50:35:
    89:d2:58:07:f2:65:7a:e8:d8:45:70:4f:1f:f5:38:
    12:3b:3d:03:80:2e:c2:2d:8d:af:36:ba:6f:d9:5c:
    3b:a8:3a:17:8c:87:b3:16:05:78:40:13:30:8c:66:
    3b:c9:c2:dd:67:1e:c9:af:99:aa:98:24:e1:bb:7b:
    7d:86:cf:14:c2:c7:23:e1:2b:a0:7c:d4:ff:f8:e8:
    5a:46:37:7d:8c:27:a7:8c:f4:4e:b0:31:2c:72:a7:
    68:02:b0:36:35:30:90:f8:d6:13:3c:66:a6:c3:22:
    c9:41:68:22:a1:8f:0c:c9:ba:53:6c:15:1f:42:b7:
    cd:1a:3e:43:fe:38:cd:fa:ba:ec:13:19:b2:4d:e0:
    44:cd:49:e1:4c:2a:2f:28:05:06:03:dc:7f:89:70:
    da:b5:90:c2:6c:e9:7c:b3:5a:0c:4d:cf:11:ce:28:
    99:6b:36:9d:96:44:ac:11:1e:a6:f2:4e:fb:76:43:
    86:d3:91:95:68:b4:2b:ea:73:bd:78:8c:4b:4f:55:
    8c:98:6f:5b:36:ea:5a:30:ec:43:6e:36:b0:72:ca:
    33:2f:31:32:49:13:3b:ed:83:ba:3f:89:e4:a3:1b:
    73:e6:fa:eb:d8:c8:6a:c0:e2:99:e6:51:5f:7a:7c:
    6b:6b
prime2:
    00:c4:70:d1:63:b2:b6:e6:8e:32:b1:0d:5f:a4:f2:
    a4:99:3c:4d:cb:85:46:7f:c4:96:6c:a7:b7:40:82:
    2f:6c:ca:f6:fb:ce:f3:0f:3f:2e:5e:a6:65:dc:2f:
    e6:2c:46:d6:aa:e6:0a:30:d3:6f:52:7b:da:79:ee:
    93:1c:22:85:43:1b:0f:d0:16:7a:4d:05:3d:60:42:
    75:9a:b1:bf:c5:24:77:45:27:3a:b6:16:73:8d:64:
    d4:86:52:9c:26:92:d5:0f:ff:7f:06:eb:dc:b9:8b:
    b2:ac:4d:6d:04:dc:5e:1a:3a:9f:12:5f:a7:b6:fe:
    2d:74:39:b2:c1:13:28:01:f9:8b:e1:05:1a:86:21:
    d7:53:7f:b3:84:67:0a:91:b0:13:bb:b5:e8:49:3b:
    62:58:33:44:50:60:66:71:c0:d8:de:2a:01:1e:23:
    34:55:1c:24:30:1c:58:e5:a5:2d:54:1a:7f:6c:8d:
    03:6b:9e:1b:0c:7b:be:e8:eb:a8:84:ba:50:e9:f3:
    cd:ae:7c:9c:0f:59:d1:aa:51:6b:e5:f9:11:87:89:
    2b:1a:bd:ae:3c:78:80:59:aa:3f:50:c7:93:db:79:
    80:48:96:a9:8f:ed:03:7a:e7:9d:7f:30:be:4e:cf:
    36:09:e3:d1:17:a7:54:17:72:d5:75:73:d3:c4:16:
    31:37
exponent1:
    0d:c6:9e:5c:55:54:dd:ba:3e:5d:0d:73:fc:8d:e3:
    b2:5a:39:c6:ee:d2:3e:11:bb:38:f0:0d:68:2e:39:
    af:bb:77:7c:e7:cb:fa:a8:88:79:ca:ea:a4:58:40:
    d9:48:f4:64:13:24:d0:37:ed:2c:6b:b8:25:74:65:
    35:23:dc:c9:aa:64:ba:87:7d:48:68:51:e5:37:7a:
```

87:4f:c7:dd:bd:bc:5e:49:99:da:35:59:35:3a:a9:
3d:ab:4a:57:d1:78:c2:05:3e:b6:71:87:43:20:c3:
29:fd:d2:8c:a2:62:cf:de:f7:f0:4d:f0:ba:88:54:
d2:07:60:a0:9c:83:57:74:38:49:a4:8c:63:0d:cd:
32:35:e9:18:b7:cb:54:96:53:3e:e3:f5:e1:7b:6c:
bd:89:1b:d8:cc:5e:58:94:cc:01:e8:dc:d3:93:9c:
02:da:27:2a:13:51:b0:e0:b9:00:dd:47:23:2e:f9:
cc:0f:c9:9c:cc:64:2e:f4:d3:5e:74:e3:b6:ec:58:
98:3b:00:59:e6:f9:02:92:78:03:38:df:b7:4e:d1:
fb:b6:0e:09:9c:35:58:da:89:3d:84:16:48:0f:d0:
b2:8b:cc:81:db:9a:63:0f:c8:44:c6:72:8e:60:31:
2b:dd:5f:b4:5e:63:35:2f:89:98:d4:d9:c9:d2:a5:
17
exponent2:
3e:26:af:a9:0a:7f:21:bc:f5:be:dc:cb:59:05:c8:
1c:0f:5f:51:b8:0e:11:18:a2:bb:27:e9:2b:c8:c4:
b6:78:14:e2:a0:9c:78:43:76:29:4e:1e:46:27:05:
7d:16:c0:8e:7b:8f:d7:d8:dd:8c:cc:50:d8:69:2a:
34:29:9b:de:1a:6e:cb:58:81:43:63:18:03:1f:a3:
d3:d8:cf:22:d1:a1:c6:c8:5d:a4:04:c0:74:88:e5:
50:c6:4c:4f:4b:ff:8b:3b:3b:f6:75:a3:75:35:04:
c7:cd:0b:64:11:50:d8:9d:99:7c:97:b5:19:89:9b:
d9:fc:f3:58:70:f6:24:81:ce:c6:d7:f5:87:60:a7:
9e:ec:75:60:bb:b5:35:58:ef:35:61:9d:67:a6:19:
3c:6c:a3:fd:86:32:83:65:0f:78:62:f3:cd:94:37:
d7:81:ba:d5:59:f2:aa:5d:b3:54:bb:b9:7b:1f:b6:
68:53:32:9d:a5:7d:46:76:60:ef:f5:7e:ea:36:4f:
ea:17:d4:cc:6a:8f:92:82:86:4f:d1:98:c1:38:1d:
e2:2d:07:36:6c:ad:b7:30:cd:2d:82:a7:7e:36:17:
e7:5f:7b:b0:64:bd:c8:22:05:cb:9b:21:6b:0f:d6:
3a:14:d1:2f:d9:aa:1a:6d:23:aa:30:86:5c:b0:e4:
cb
coefficient:
00:8c:9d:12:3d:7c:7f:18:8c:8d:fe:b7:f3:27:50:
06:ad:0f:40:2c:c1:dd:75:a2:5c:3e:ea:b6:ba:01:
30:3d:b8:12:05:91:9d:68:ec:da:b9:0d:74:1f:9f:
83:10:c6:03:5f:a1:a4:e4:d7:9b:db:27:1a:9c:84:
69:18:63:02:54:71:7a:84:2f:84:9d:83:18:3b:d0:
6d:7c:af:76:64:3d:b8:cc:b1:17:ec:cb:e6:11:22:
d5:8f:2d:be:6f:d0:b9:f0:c0:1f:93:65:e3:9a:4c:
43:0b:05:6f:48:6e:01:c4:08:06:ad:9f:2a:58:55:
a1:ef:b1:22:88:a0:71:13:51:90:95:03:d9:14:2d:
8f:26:44:3e:4c:77:fb:a3:df:6d:c1:ae:4f:5e:64:
f5:55:b8:68:ff:c6:17:90:02:a2:f9:62:f1:eb:aa:
1c:57:93:1f:e7:19:27:36:19:03:c5:3e:e7:74:c8:
9c:05:f1:dd:21:d9:80:46:e3:be:43:af:bb:9c:be:

```
    47:f3:6a:9d:65:ad:f4:00:32:68:45:67:35:da:15:
    5d:72:15:ac:e4:e3:65:5c:ee:37:96:cd:a0:87:93:
    6f:ca:e3:0d:16:62:4e:33:76:e6:ba:02:33:d5:d4:
    01:ab:60:54:d9:0a:56:22:24:3d:e7:a7:7c:49:ba:
    bf:0b
[02/28/22]seed@VM:~/Labsetup$
```

Many websites have different URLs. For example, example.com and example.net are all pointing to the same web server. Due to the SEED Labs – PKI Lab 5 hostname matching policy enforced by browsers, the common name in a certificate must match with the server's hostname, or browsers will refuse to communicate with the server [1].

To allow a certificate to have multiple names, the X.509 specification defines extensions to be attached to a certificate. This extension is called Subject Alternative Name (SAN). Using the SAN extension, it's possible to specify several hostnames in the subjectAlt-Name field of a certificate [1].

To generate a certificate signing request with such a field, we can put all the necessary information in a configuration file or at the command line. In this task, we used the command-line approach [1]. The following command (marked in yellow) generate a CSR for www.student22.com and add two alternative names to the certificate signing request (see picture 5).

```
[02/28/22]seed@VM:~/Labsetup$ openssl req -newkey rsa:4096 -sha256 \
> -keyout server.key -out server.csr \
> -subj "/CN=www.student22.com/O=Student22 Inc./C=US" \
> -passout pass:dees \
> -addext "subjectAltName = DNS:www.student22.com, DNS:www.student22A.com, DNS:w
ww.student22B.com"
Generating a RSA private key
.....................................................................................
.....................++++
.........................................++++
writing new private key to 'server.key'
-----
[02/28/22]seed@VM:~/Labsetup$
```

Picture 5, Command in the terminal to generate the CSR for www.student22.com and add two alternative names to the certificate signing request.

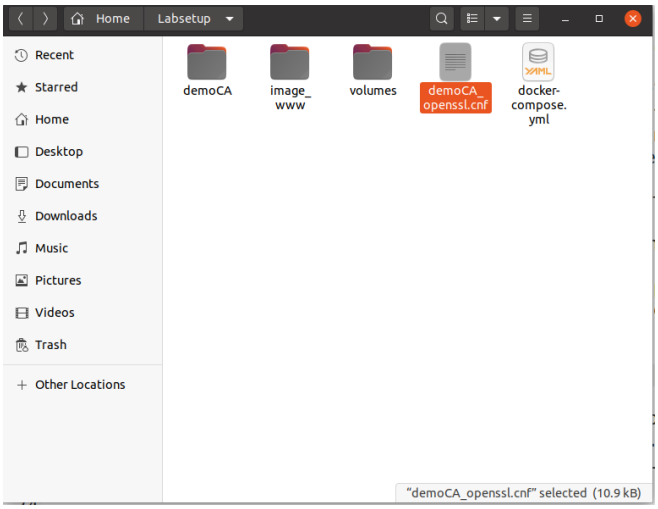
4. Task 3: Generating a Certificate for your server

The CSR file needs to have the CA's signature to form a certificate. In the real world, the CSR files are usually sent to a trusted CA for their signature. In this lab, we used our own trusted CA to generate certificates. The following command (marked in yellow) turns the certificate signing request (server.csr) into an X509 certificate (server.crt), using the CA's ca.crt and ca.key [1]. Note: In this task, all command shall be written in ubuntu20.04-vm terminal.

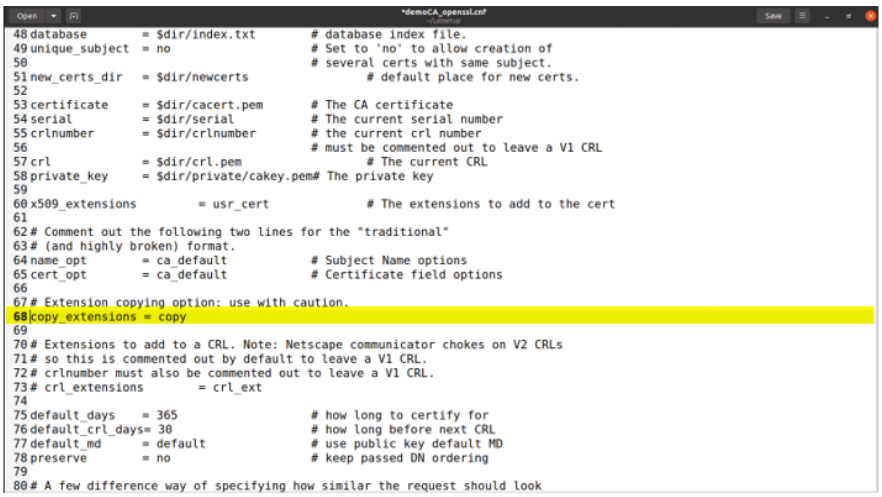
```
[02/28/22]seed@VM:~/Labsetup$ openssl ca -config demoCA_openssl.cnf -policy policy_anything \
> -md sha256 -days 3650 \
> -in server.csr -out server.crt -batch \
> -cert ca.crt -keyfile ca.key
Using configuration from demoCA_openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Feb 28 15:37:40 2022 GMT
            Not After : Feb 26 15:37:40 2032 GMT
        Subject:
            countryName               = US
            organizationName          = Student22 Inc.
            commonName                = www.student22.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                0D:C6:B3:73:88:EC:71:3D:19:1F:95:BF:B3:BA:58:7F:9E:9A:B4:47
            X509v3 Authority Key Identifier:
                keyid:E9:1A:55:DE:22:A8:46:94:44:B9:97:1D:A0:CE:AD:12:7D:ED:AE:0B

            X509v3 Subject Alternative Name:
                DNS:www.student22.com, DNS:www.student22A.com, DNS:www.student22B.com
Certificate is to be certified until Feb 26 15:37:40 2032 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
[02/28/22]seed@VM:~/Labsetup$
```

In the previous command, myCA openssl.cnf is the configuration file we copied from /usr/lib/ssl/openssl.cnf (see Task 1). We used the policy anything policy defined in the configuration file. This is not the default policy; the default policy has more restriction, requiring some of the subject information in the request to match those in the CA's certificate. The policy used in the command, as indicated by its name, does not enforce any matching rule [1].

For security reasons, the default setting in openssl.cnf does not allow the "openssl ca" command to copy the extension field from the request to the final certificate. To enable that, we removed # mark (uncomment) in our copy of the configuration file (see pictures 6 and 7) [1].



Picture 6, Location of demoCA_openssl.cnf file (our copy of the configuration file).



Picture 7, # mark removed from the row highlighted in yellow.

After signing the certificate, we used the following command to print out the decoded content of the certificate.

```
[02/28/22]seed@VM:~/Labsetup$ openssl x509 -in server.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4097 (0x1001)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = www.modelCA.com, O = Model CA LTD., C = US
        Validity
            Not Before: Feb 28 15:37:40 2022 GMT
            Not After : Feb 26 15:37:40 2032 GMT
        Subject: C = US, O = Student22 Inc., CN = www.student22.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:cf:a1:83:d6:c7:ab:ea:a9:df:7c:aa:2e:c7:24:
                    80:12:bf:2c:cb:7b:d6:98:5c:70:4e:4c:94:78:a2:
                    e4:a8:8c:b1:ec:f3:01:47:9d:d2:dc:ab:0d:1a:4d:
                    1f:de:27:35:f7:ee:72:4a:70:53:f9:12:51:43:59:
                    cc:94:ac:4e:ff:41:e0:60:71:e0:d6:4e:df:93:cc:
                    04:de:39:64:32:ea:9d:6c:6f:eb:13:ef:24:98:a0:
                    df:29:4b:6a:3c:72:98:5d:eb:fd:67:ed:4b:35:e1:
                    b4:9a:10:c9:d7:ee:7c:bb:05:fa:e1:33:5e:b7:3b:
                    0a:be:0e:d6:7d:59:4e:40:5b:a5:a3:2a:9f:ca:da:
                    b3:47:33:98:a1:e7:cf:00:30:18:3f:52:aa:28:ee:
                    45:5f:98:18:02:68:4e:3d:37:66:2c:48:1e:d3:a9:
                    34:d5:31:3d:77:06:f3:b3:da:5b:e3:00:3b:7b:7e:
                    81:6f:41:46:34:4c:df:d2:7f:c3:fd:25:e5:32:e9:
                    a9:6c:58:16:26:28:fd:80:d2:9e:12:18:be:fd:e6:
                    b0:d6:b1:b4:6b:c3:8c:7b:6f:cf:85:82:6f:2a:ff:
                    e3:db:18:79:68:5e:bc:56:5d:d6:39:b5:7e:94:24:
                    a8:0c:14:a7:10:c6:f1:2b:bb:0d:93:8d:37:ca:9a:
                    e8:ed:b1:f4:85:96:3b:85:15:d8:fa:f1:83:5e:f1:
                    b5:ed:a1:7d:de:be:86:83:76:be:53:29:bb:dd:86:
                    c2:2f:c1:cd:f6:a7:8d:ca:ff:29:4a:1a:a8:b1:53:
                    d2:a4:a3:a0:2d:23:03:fa:31:e1:d4:cc:21:bc:8b:
                    4d:73:9d:0e:d8:48:03:29:6a:76:26:e0:2e:1e:fe:
                    5a:52:4f:5f:c4:b7:72:01:88:cd:00:26:17:ff:78:
                    f7:49:60:b5:87:f8:fd:7f:81:9a:6a:66:b7:5d:e3:
                    f5:46:4c:8e:e3:a1:88:ff:a2:21:ae:7f:17:24:d2:
                    82:2d:a7:8e:db:6e:de:db:3e:4b:cf:a1:54:ea:ad:
                    03:00:ad:bb:78:56:9b:ce:f8:17:11:ca:28:47:36:
                    ff:11:47:ee:21:ac:fb:24:72:bb:1b:15:d2:95:6e:
```

```
                69:2f:c4:1c:43:2b:2f:a8:7a:2a:a5:6b:62:95:96:
                20:12:cb:72:1c:ec:b9:1e:ee:bf:8f:0b:06:53:a9:
                17:c4:d1:a1:52:79:38:ee:ba:56:90:c5:73:23:16:
                4b:84:fc:ca:ee:ad:e9:f5:9c:a4:82:ba:68:fb:db:
                17:76:76:65:0f:32:43:88:63:20:09:ac:49:e8:07:
                c5:0d:74:17:b6:fa:64:59:bb:ef:13:17:90:9c:68:
                53:86:75
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                0D:C6:B3:73:88:EC:71:3D:19:1F:95:BF:B3:BA:58:7F:9E:9A:B4:47
            X509v3 Authority Key Identifier:
                keyid:E9:1A:55:DE:22:A8:46:94:44:B9:97:1D:A0:CE:AD:12:7D:ED:AE:0B

            X509v3 Subject Alternative Name:
                DNS:www.student22.com, DNS:www.student22A.com, DNS:www.stu-
dent22B.com
    Signature Algorithm: sha256WithRSAEncryption
        0b:b9:ee:43:19:3c:aa:14:82:fd:3c:fa:da:44:fe:5a:9f:77:
        20:d6:4a:8b:49:9b:78:14:02:db:c3:4c:cb:11:0c:84:86:64:
        db:a4:cd:a7:fa:f9:e9:6f:46:51:8c:e3:78:55:d1:38:a1:08:
        63:1c:69:e9:f6:fb:47:74:64:96:bd:cc:85:e2:4c:cc:d7:4a:
        96:b1:b9:98:06:47:7a:3b:47:a2:4d:c3:85:69:4d:2f:b1:53:
        4a:8d:93:19:95:73:c8:a6:a2:7f:9a:51:aa:85:60:db:99:db:
        67:c1:0b:17:96:44:b2:fe:7d:77:d6:23:7c:5d:23:9c:57:72:
        33:cb:4b:eb:87:e9:90:7b:50:49:3a:fb:bf:26:09:47:9e:6d:
        0a:00:52:57:b2:8b:8e:1f:33:34:2c:27:76:52:65:0c:72:08:
        9b:d6:b6:b2:d2:0b:51:c7:ff:88:28:19:b3:ea:b4:2b:22:93:
        c2:d5:35:74:ab:07:56:b8:cd:09:29:41:c4:4b:88:c6:13:58:
        5c:cd:a4:19:a0:3a:83:d8:81:07:9a:96:3d:81:aa:d9:03:de:
        8b:82:db:6f:13:7d:85:ca:06:cc:6b:c2:a6:e9:35:dd:df:3a:
        68:63:0d:27:a8:0c:b1:f6:fc:0d:30:95:df:c2:80:ee:f4:4b:
        63:cc:19:a3:ab:8c:ea:7b:01:28:62:e3:73:be:28:c6:c3:4e:
        96:50:a7:d9:a5:63:13:0d:b5:99:97:d0:c5:6a:4e:37:2c:7a:
        d9:7d:69:d5:75:3a:00:37:38:41:4d:7f:da:46:7b:90:c9:f0:
        35:dc:19:07:c5:f3:39:63:58:48:51:f7:2d:e8:13:6d:47:17:
        b0:08:49:65:b5:ee:e1:ef:63:1d:c8:c7:29:c2:ce:d7:bb:c7:
        d8:a0:1f:6c:4c:53:cb:d7:0e:a9:d3:6a:99:51:69:a8:2d:36:
        be:04:00:77:41:d5:a8:db:ea:35:f3:0b:55:83:8f:ee:dd:2b:
        3c:c8:7d:4f:70:53:69:47:50:84:6f:2c:41:54:99:48:e6:5f:
        3d:60:78:99:1a:4a:ed:c8:00:85:f7:88:85:96:03:4b:46:48:
        e5:ba:f5:03:10:90:57:d7:cc:05:0e:f8:ae:7a:b0:f8:a5:44:
```

```
        55:33:f2:db:88:b8:69:5f:fb:71:28:5c:ee:fe:43:ea:47:50:
        0a:28:63:c1:72:00:1e:94:ed:c1:dd:d9:a1:7c:3f:2e:f4:94:
        84:09:c5:46:dc:26:ad:08:38:7f:88:aa:e0:48:63:6a:63:2c:
        34:c6:ba:30:e7:5b:d6:43:82:1b:ae:1c:44:69:48:c0:63:5e:
        d8:2d:0e:ae:9b:63:91:c8
[02/28/22]seed@VM:~/Labsetup$
```

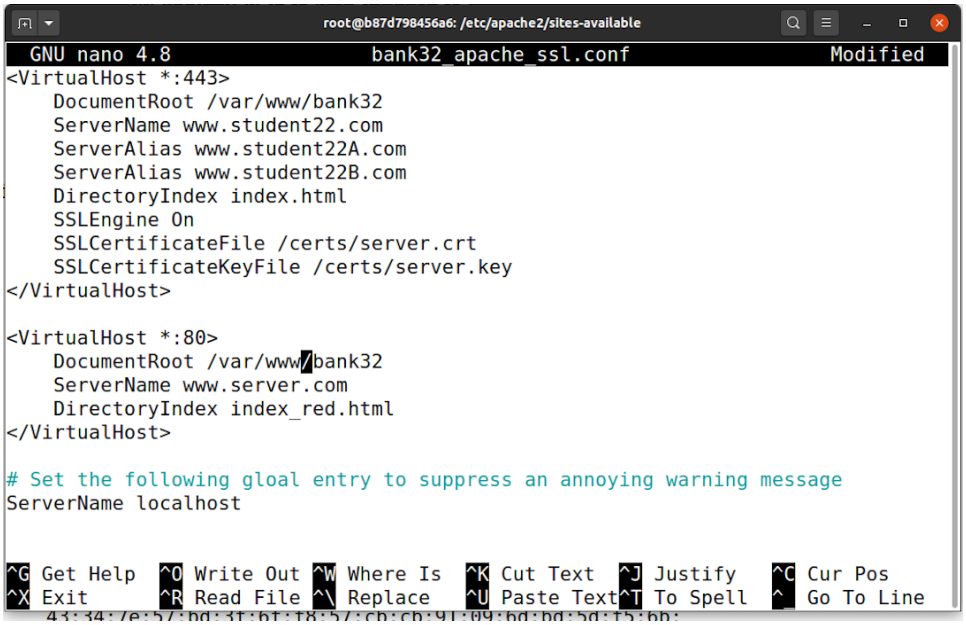
In the previous printout alternative names have been marked in green.

5. Task 4: Deploying Certificate in an Apache-Based HTTPS Website

In this task we were supposed to operate inside the container and start the Apache –server to be connected from the virtual machine side.

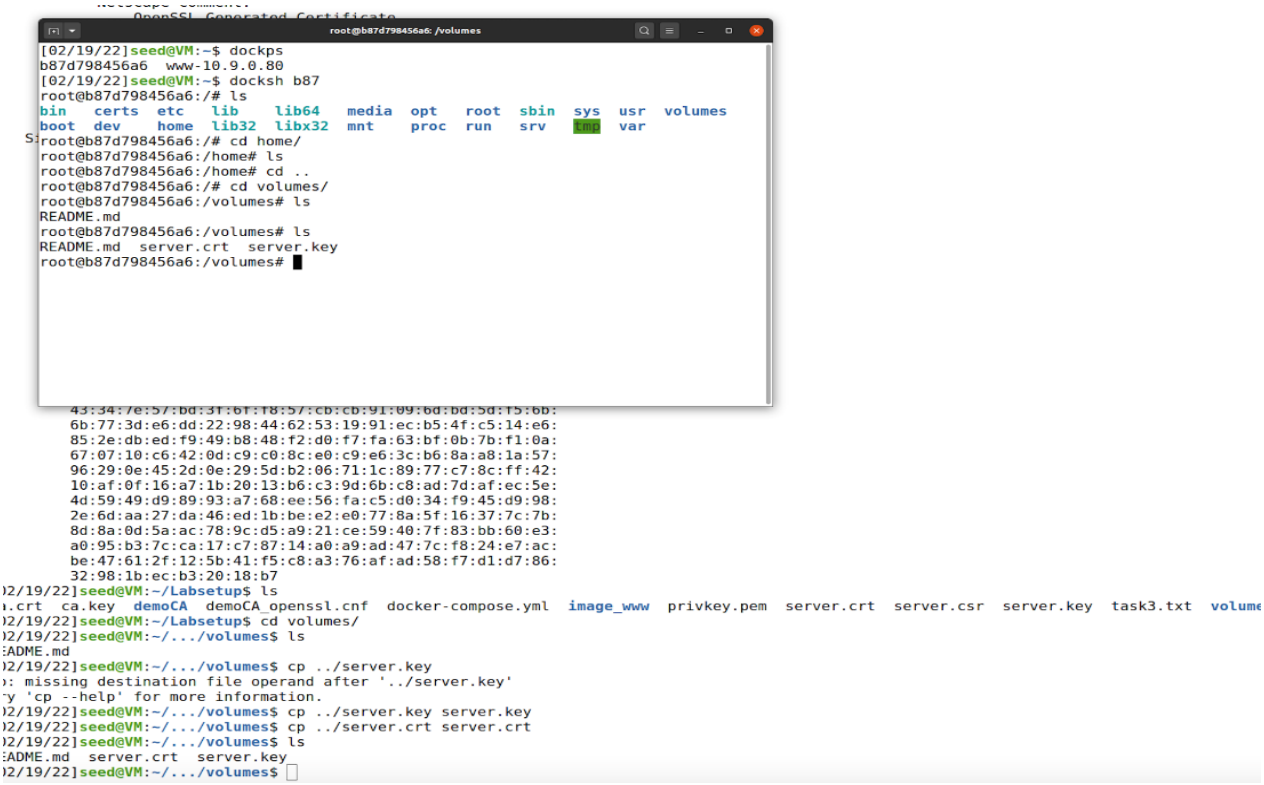
Firstly, we needed to start the docker inside the “Labsetup” -folder. This can be done with the command dcup. After starting the docker –service we can see the available containers by listing them using the command dockps. From the list shown, it is possible to connect the docker using command docksh + it’s id. (Or two first characters of it).

To be able to run the server as https://www.student22.com, some changes in the bank32_apache_ssl.conf -file was to be made. It’s important to notice, as we speak now of https –server, that the right port to be edited is 443. There was no need to change the name of the file, nor the folder where the server files are located. Also index.html as index –file could be left intact.



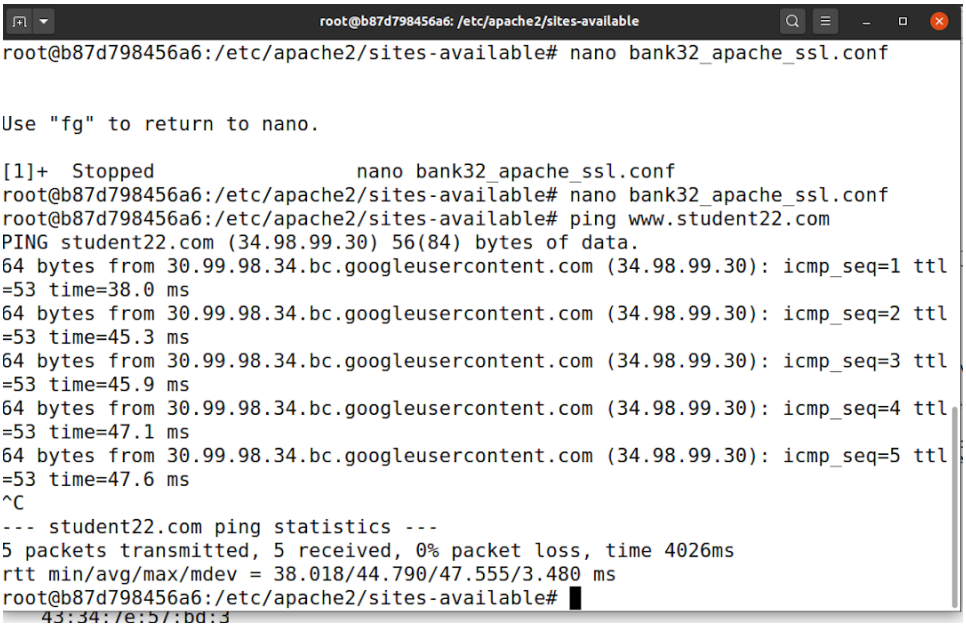
Picture 8, Edited bank32_apache_ssl.conf -file in the container’s /etc/apache2/sites-available directory.

After setting up the right address for our server, we needed to copy server.crt and server.key -files to container, so that our server could use them. Location for those was /certs/ as it was set up in the above mentioned .conf -file. We were able to make the copies easily, since a directory “volumes” was pre-mounted as shared folder between VM and the container.



Picture 9, Copying crt- and key- files to be used by server.

When trying to reach the student22.com using ping from the VM (to the Apache –server in container) every packet was delivered but on the other hand student22a was unreachable.



```
^C
--- student22.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4026ms
rtt min/avg/max/mdev = 38.018/44.790/47.555/3.480 ms
root@b87d798456a6:/etc/apache2/sites-available# ping www.student22A.com
ping: www.student22A.com: Name or service not known
root@b87d798456a6:/etc/apache2/sites-available#
```

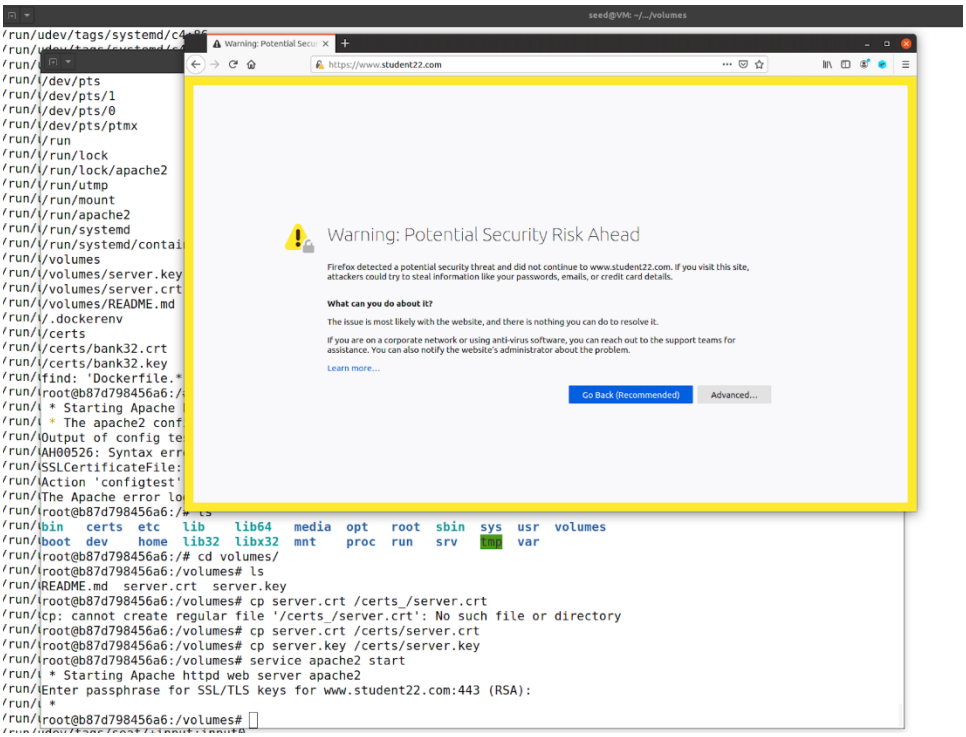
Picture 9-10, Ping for student22.com and student22a.com

To fix that problem we needed to edit etc/hosts -file to map alternative domains under the comment #For PKI Lab.

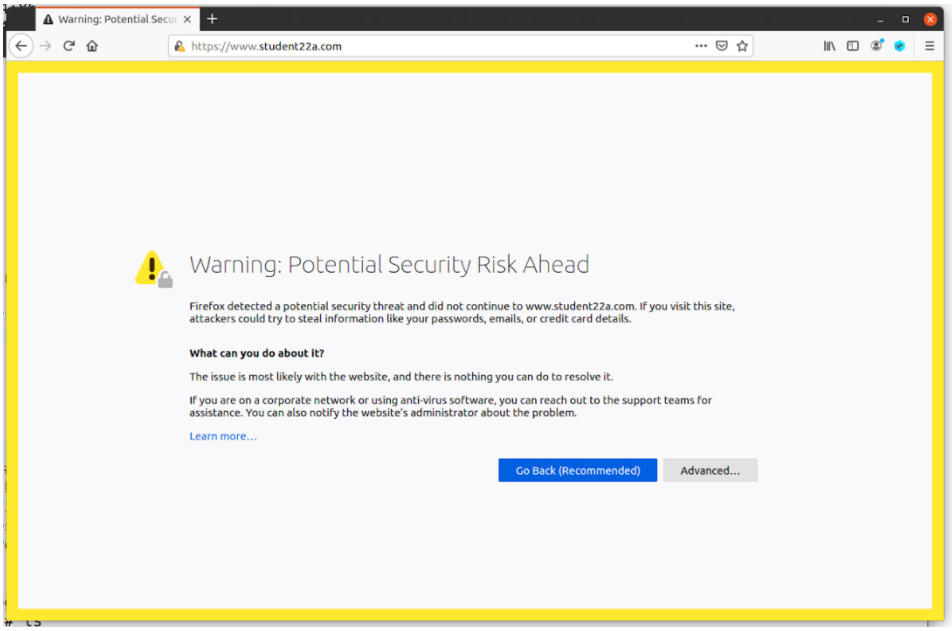
```
127.0.0.1       localhost
127.0.1.1       VM

# 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

# For DNS Rebinding Lab
#192.168.60.80   www.seedIoT32.com

# For SQL Injection Lab
#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.seed-server.com
#10.9.0.5        www.example32.com
#10.9.0.105      www.attacker32.com

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

#For PKI Lab
10.9.0.80 www.student22.com
10.9.0.80 www.student22A.com
10.9.0.80 www.student22B.com
```

Picture 11, Editing etc/hosts -file

After restarting the server, it was reachable from browser on VM as both, https://www.student22.com and https://student22a.com. However, neither of those were a trusted site and FireFox gave a warning about potential security risk.

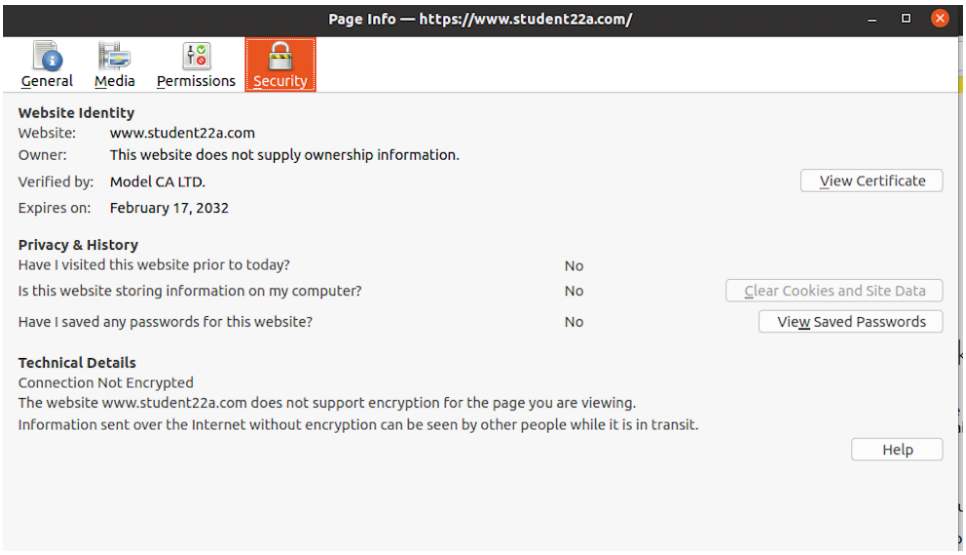


Picture 12, Student22.com: reachable, but untrusted

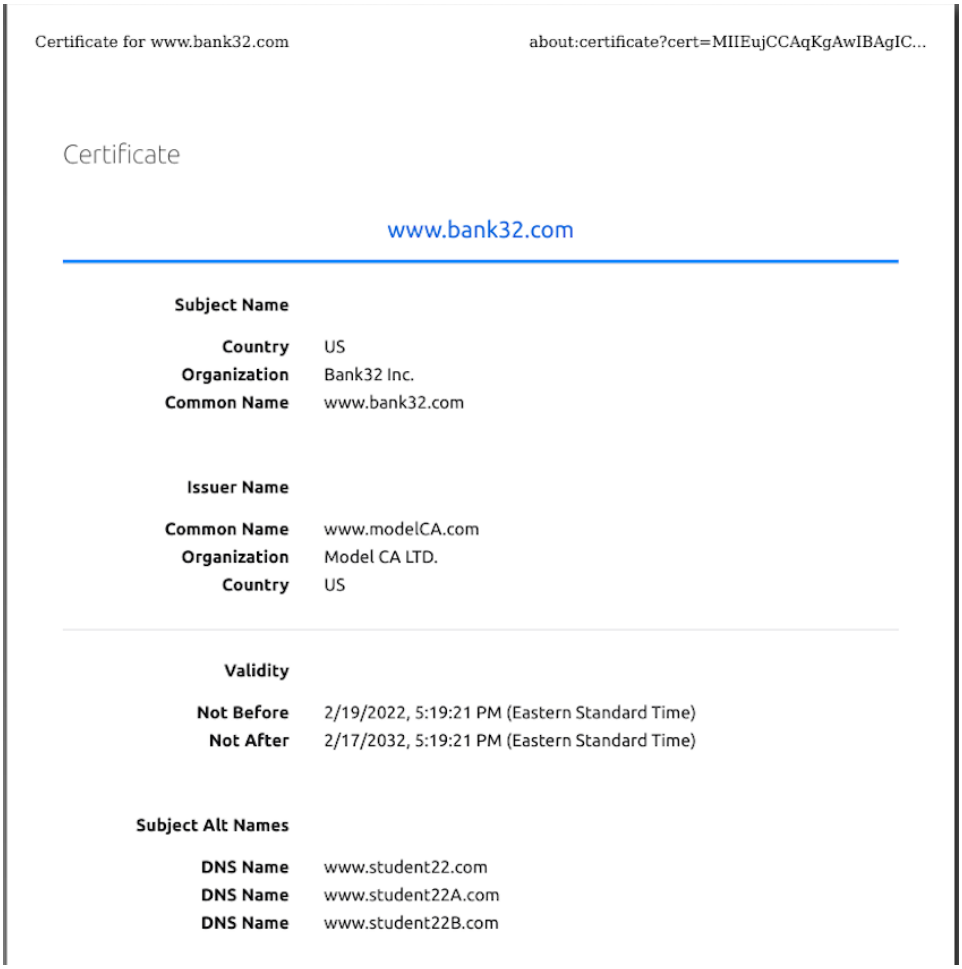


Picture 13, Student22A reachable too, but not secure either. Can't access. (Without warnings)

Checking the certificate, it tells that student22a.com provides no ownership.

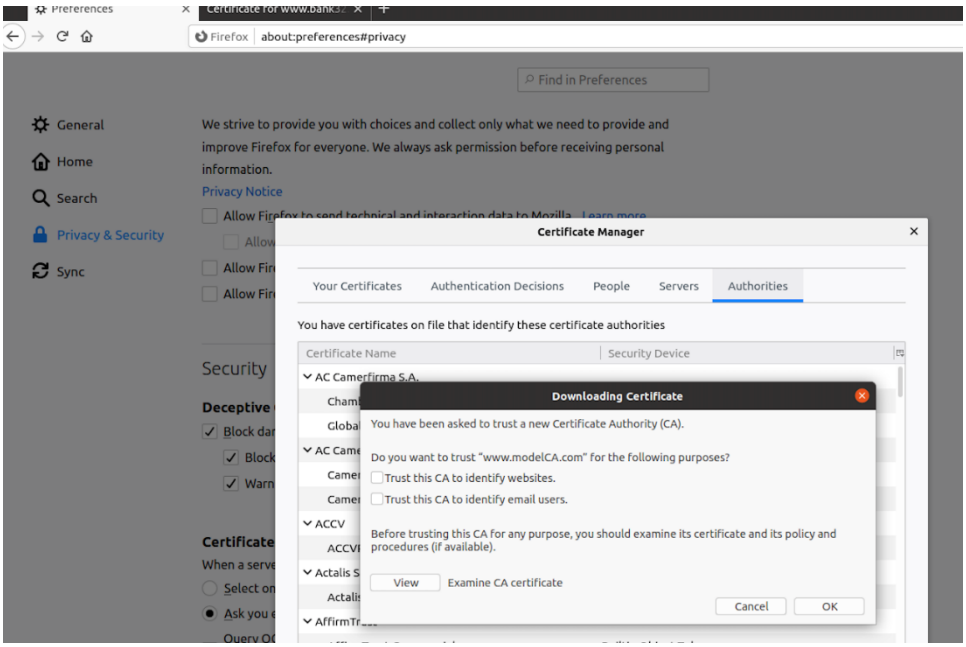


Picture 14, Student22a.com website identity



Picture 15, Certificate shows, that alternative names are correct, and issuer is www.modelCA.com as wanted.

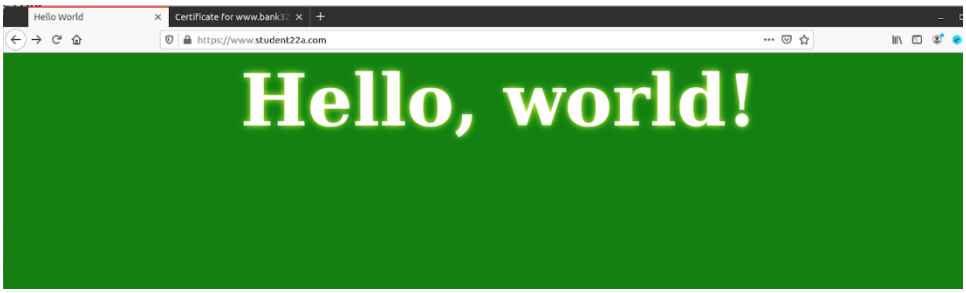
To be able to connect the server as trusted we had to add ModelCA as trusted CA manually from preferences page, about:preferences#privacy.



Picture 16, Adding ModelCA to trusted CA

Picture 17, ModelCA now trusted

Now site(s) were accessible without warnings.



Picture 18, Inside https://www.student22a.com running inside a container and as trusted, secure page.

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

The goal of this task was to set up a malicious website, using a well-known URL which would impersonate to be the legitimate one, thus enabling MitM –attack. This begins by editing the servers much like we did in Task 4. As we choose Twitter.com to be the site, our server wants to impersonate, following changes were made to the bank32_apache_ssl.conf -file:

```
<VirtualHost *:443>
    DocumentRoot /var/www/bank32
    ServerName www.twitter.com
    ServerAlias www.student22A.com
    ServerAlias www.student22B.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /certs/server.crt
    SSLCertificateKeyFile /certs/server.key
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /var/www/bank32
    ServerName www.server.com
    DirectoryIndex index_red.html
</VirtualHost>
```

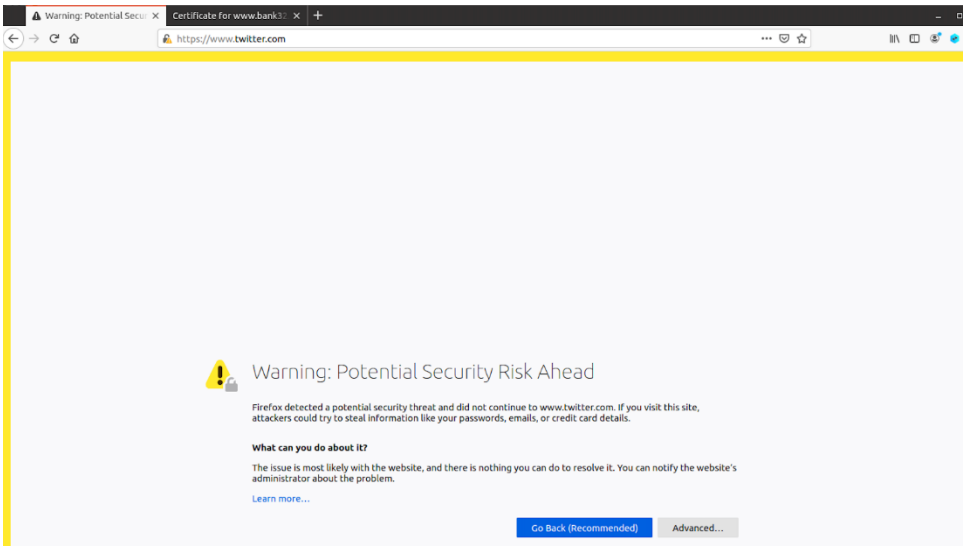
Picture 19, Edititng bank32_apache_ssl.conf -file in order to launch MitM -attack

Next we needed to take kind of a shortcut and voluntarily edit the victims, VM's, etc/hosts file in order to map a DNS name www.twitter.com to the IP –address 10.9.0.80, which is the address of our server running on the container. This could also be done by launching a DNS cache poisoning attack, but since it was not the point of this exercise, we used this shortcut instead.[2, p. 107]

```
# 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.seed-server.com
#10.9.0.5        www.example32.com
#10.9.0.105      www.attacker32.com

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

#For PKI Lab
10.9.0.80 www.twitter.com
10.9.0.80 www.student22A.com
10.9.0.80 www.student22B.com
```

Picture 20, Edited /etc/hosts -file

After the changes were made and the server restarted, we tried to visit this site, https://www.twitter.com. Since the domain twitter.com was not signed as trusted domain, FireFox will give an alert of a security risk. This happens even though www.modelCA.com was set as a trusted CA.



Picture 21, Twitter.com (https) is reachable, but not trusted, since the certificate has been signed for student22?.com

7. Task 6: Launching a Man-In-The-Middle Attack with a Compromised CA

In this last task out objective was to complete the MitM –attack in a such way, that browsing the malicious site would seem safe to the user. In order to do so, we need to get domain twitter.com to be signed by the trusted CA, www.modelCA.com. Luckily, as it happens, modelCA has been careless and leaked out it's private key.

First, in the container, we will create a sign request for the domain name twitter.com. To do that we repeated preliminary procedures as we did in Task 1. It is to be noted that due to a stupid error of storing final signed certificate in a wrong directory this step was repeated several times. However, sign request was created as following:

```
root@b87d798456a6:/home# openssl req -newkey rsa:2048 -sha256   -keyout server.key   -out server.csr  -subj "/CN=ww
w.twitter.com/O=Twitter Inc./C=US" -passout pass:dees
Generating a RSA private key
..+++++
.....................+++++
writing new private key to 'server.key'
-----
```

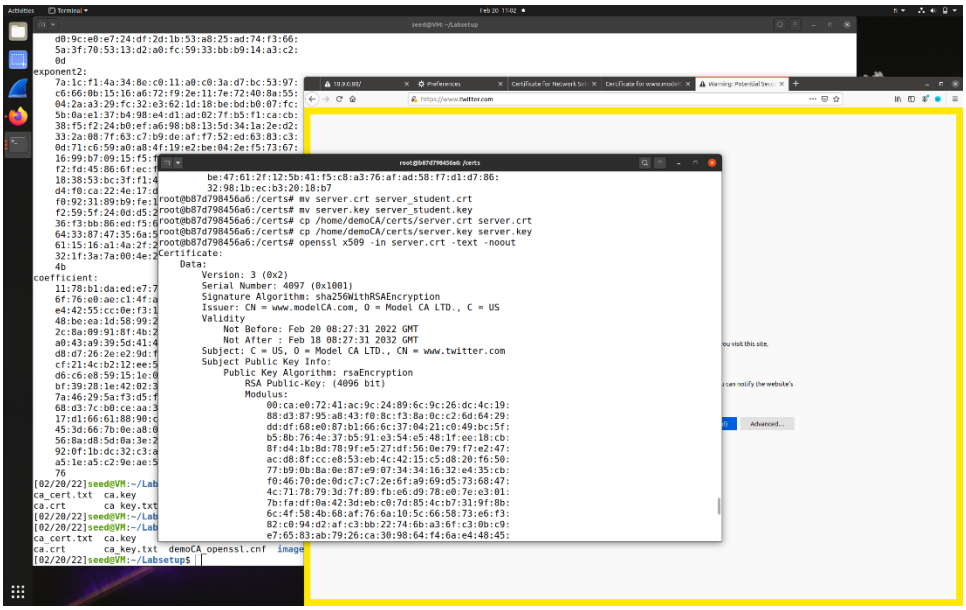
Picture 22, Sign request

Then we moved the stolen key and CA:s crt to volumes –directory, so we can use it to sign our request as if we were the rightful owner of the private key of modelCA. When the legitimate CA:s files were at our disposal, we were able to sign the certificate for Twitter.com.

```
root@b87d798456a6:/home# openssl ca -config myopenssl.cnf -policy policy_anything  -md sha256 -days 3650 -in server
.csr -out server.crt -batch  -cert ../volumes/ca.crt -keyfile ../volumes/ca.key
Using configuration from myopenssl.cnf
Enter pass phrase for ../volumes/ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Feb 20 08:27:31 2022 GMT
            Not After : Feb 18 08:27:31 2032 GMT
        Subject:
            countryName               = US
            organizationName          = Model CA LTD.
            commonName                = www.twitter.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                09:C0:EE:F8:FD:46:34:80:11:D5:10:5B:B7:73:AE:42:34:4D:F4:6E
            X509v3 Authority Key Identifier:
                keyid:F6:5E:AA:C8:53:27:AB:44:32:F6:B1:BE:AF:F2:64:78:0E:4D:8F:2E

Certificate is to be certified until Feb 18 08:27:31 2032 GMT (3650 days)
```

Picture 23, Signed Twitter -signature

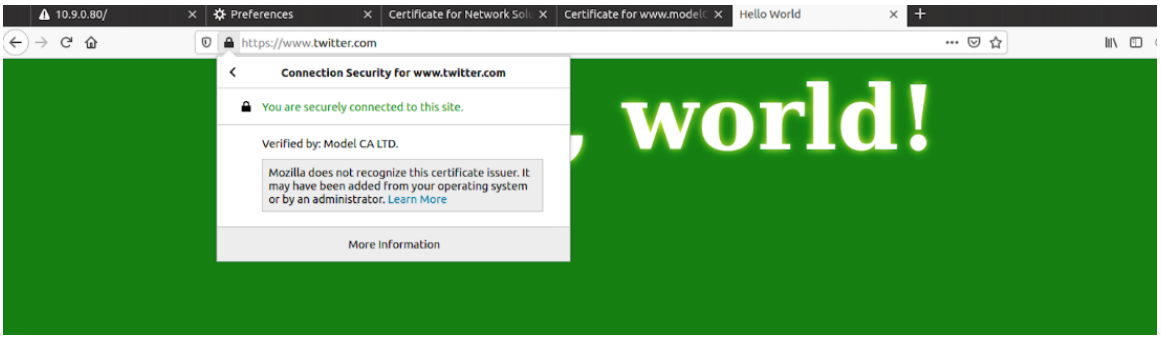
After moving these files to the /certs directory in the container to replace the old ones as location defined in bank32_apache_ssl.conf -file in the container's /etc/apache2/ sites-available directory we were able to enter the https://www.twitter.com as "secure" site.



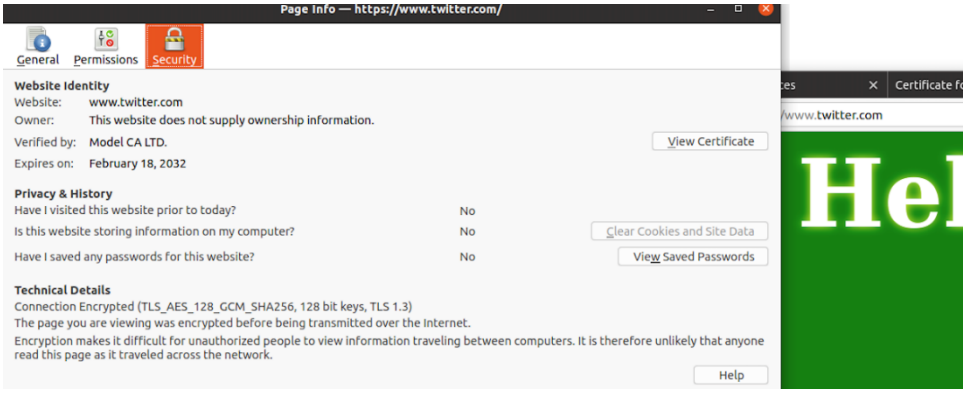
Picture 24, Copied signed server CA to right folder /certs

Picture 25, Entering the site as "secure" without any warnings.

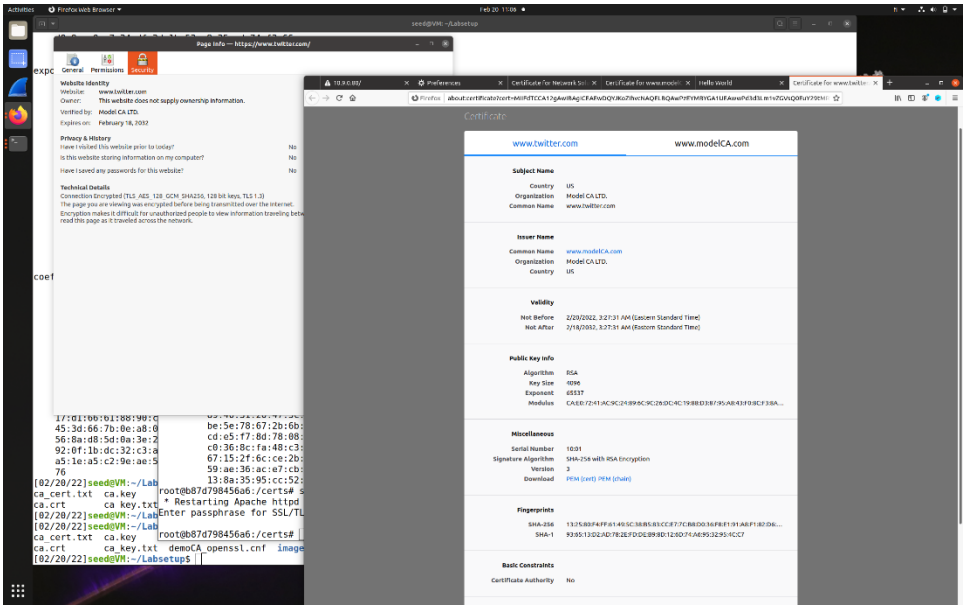
Twitter.com is now signed by the Model CA:s –certificate, which is configured as trusted CA in Task 4.



Picture 25, Verified by Model CA

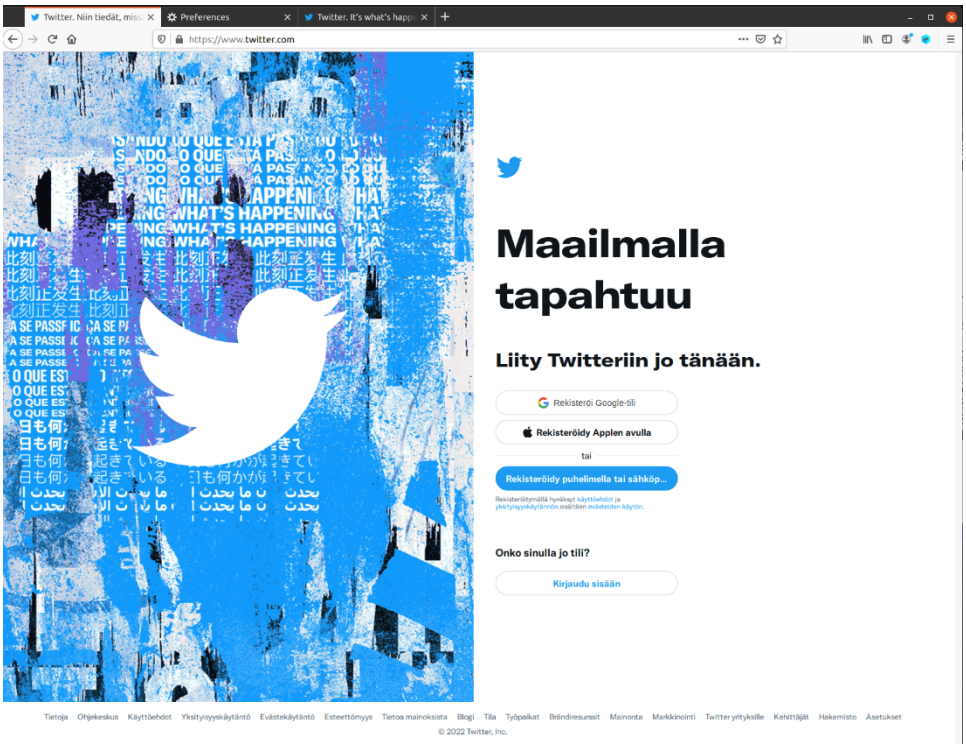


Picture 26, Malicious site is signed



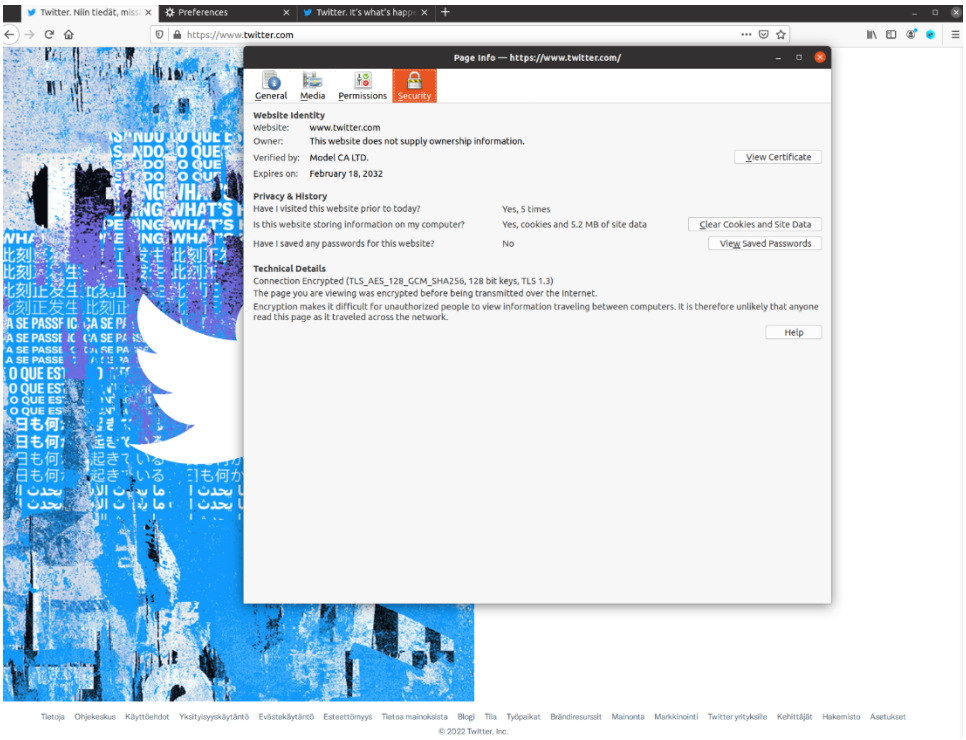
Picture 27, Twitter certificate

To finalize our MitM –attack to be more believeable we borrowed the source code wrom the real Twitter landing page.



Picture 28, Enhanced landing page.

And as we can see, the certificate issuer is still the Model CA LTD.



Picture 29, Show “Twitter” CA is ModelCa

8. Conclusion

This exercise was an interesting hands-on project and helped us to understand several aspects of Public Key infrastructure as well as introduction to containers and setting up a basic server. It was rewarding finally to see, when the self set up malicious server was running as https.

We found it easier to understand the theory behind the subject, when completing the tasks in practice via trial and error.

9. List of reference

[1] https://moodle.tuni.fi/pluginfile.php/2262824/mod_resource/content/1/Crypto_PKI.pdf

[2] Internet Security, Wenliang, Du (2019)