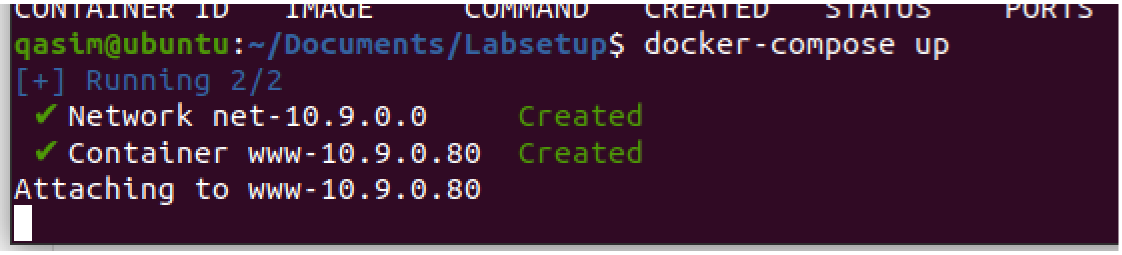
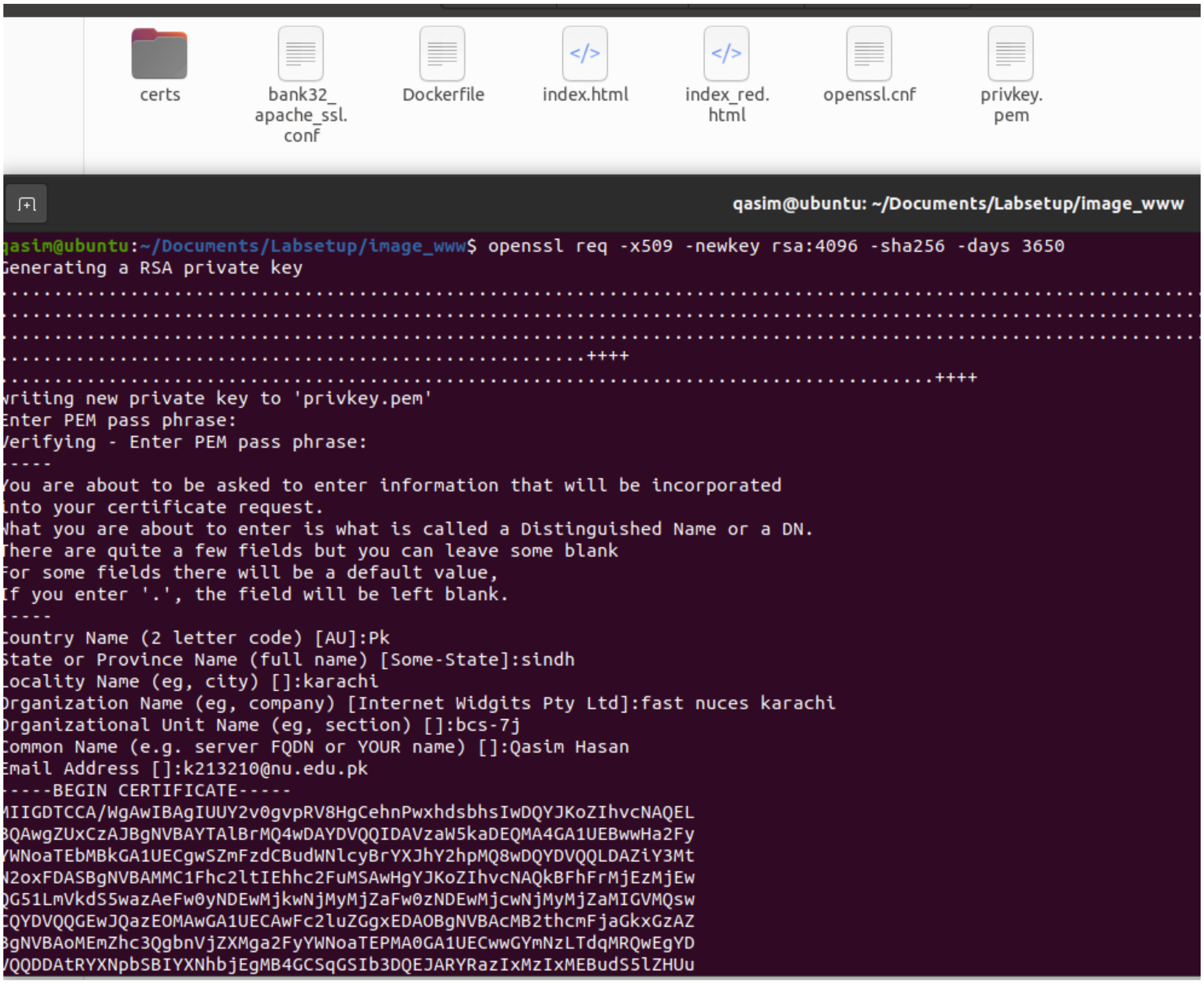
IS LAB 3

SETUP/START DOCKER CONTAINER:



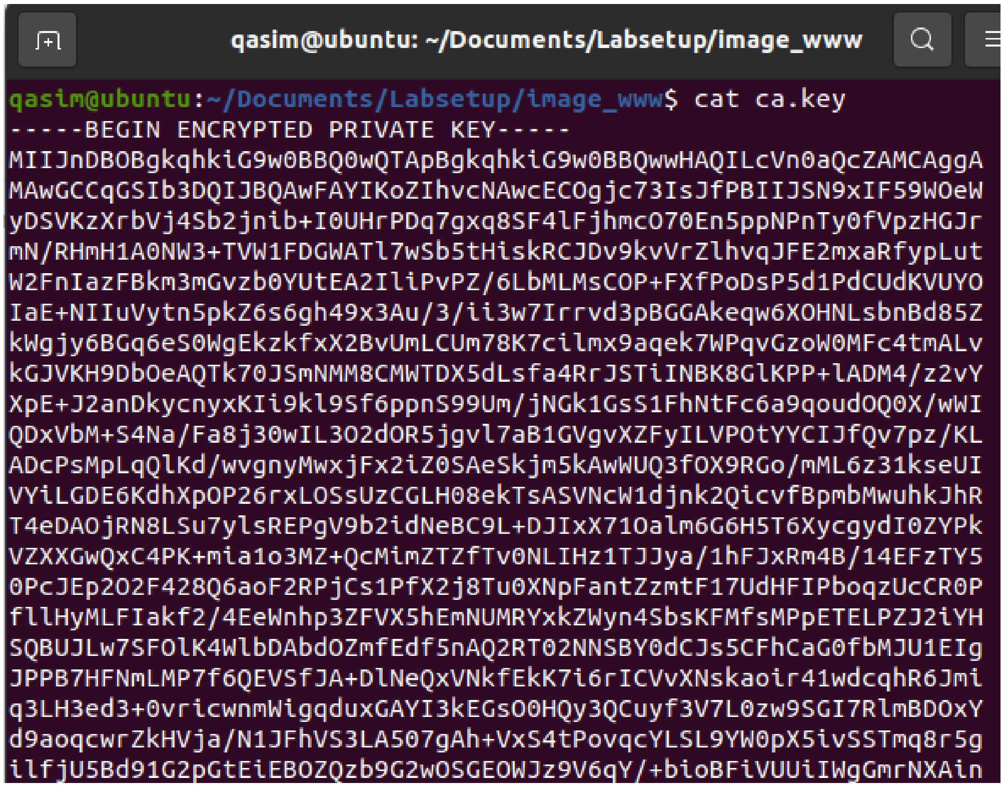
```
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
qasim@ubuntu:~/Documents/Labsetup$ docker-compose up
[+] Running 2/2
 ✔ Network net-10.9.0.0       Created
 ✔ Container www-10.9.0.80    Created
Attaching to www-10.9.0.80
```

TASK 1:

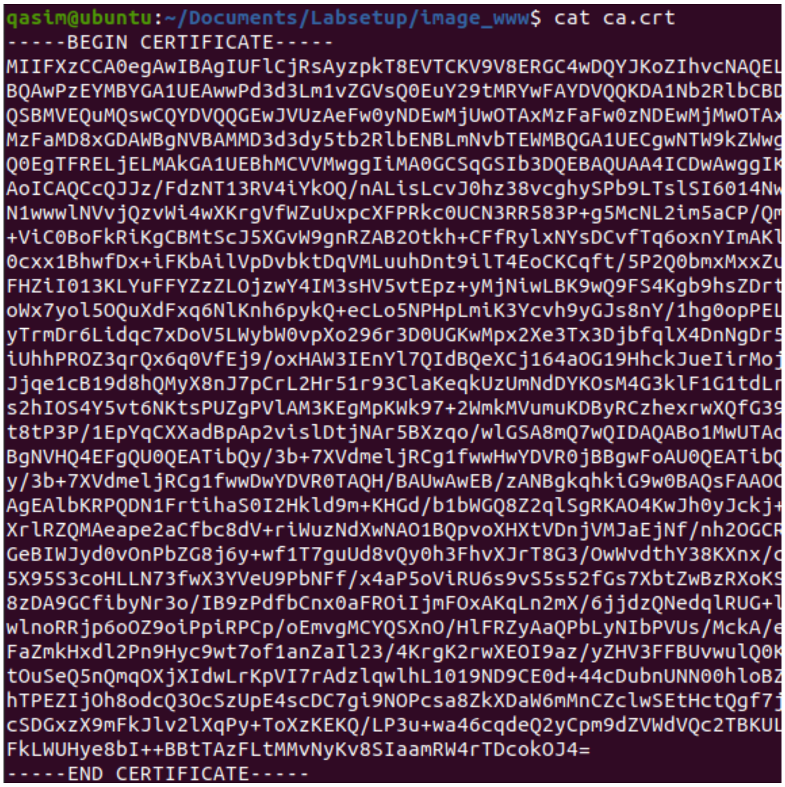


```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl req -x509 -newkey rsa:4096 -sha256 -days 3650
Generating a RSA private key
..........................................................................................................
..........................................................................................................
..........................................................................................................
.....................................................++++
.........................................................................................++++
writing new private key to 'privkey.pem'
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]:Pk
State or Province Name (full name) [Some-State]:sindh
Locality Name (eg, city) []:karachi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:fast nuces karachi
Organizational Unit Name (eg, section) []:bcs-7j
Common Name (e.g. server FQDN or YOUR name) []:Qasim Hasan
Email Address []:k213210@nu.edu.pk
-----BEGIN CERTIFICATE-----
MIIGDTCCA/WgAwIBAgIUUY2v0gvpRV8HgCehnPwxhdsbhsIwDQYJKoZIhvcNAQEL
BQAwgZUxCzAJBgNVBAYTAlBrMQ4wDAYDVQQIDAVzaW5kaDEQMA4GA1UEBwwHa2Fy
YWNoaTEbMBkGA1UECgwSZmFzdCBudWNlcyBrYXJhY2hpMQ8wDQYDVQQLDAZiY3Mt
N2oxFDASBgNVBAMMC1Fhc2ltIEhhc2FuMSAwHgYJKoZIhvcNAQkBFhFrMjEzMjEw
QG51LmVkdS5wazAeFw0yNDEwMjkwNjMyMjZaFw0zNDEwMjcwNjMyMjZaMIGVMQsw
CQYDVQQGEwJQazEOMAwGA1UECAwFc2luZGgxEDAOBgNVBAcMB2thcmFjaGkxGzAZ
BgNVBAoMEmZhc3QgbnVjZXMga2FyYWNoaTEPMA0GA1UECwwGYmNzLTdqMRQwEgYD
VQQDDAtRYXNpbSBIYXNhbjEgMB4GCSqGSIb3DQEJARYRazIxMzIxMEBudS5lZHUu
```

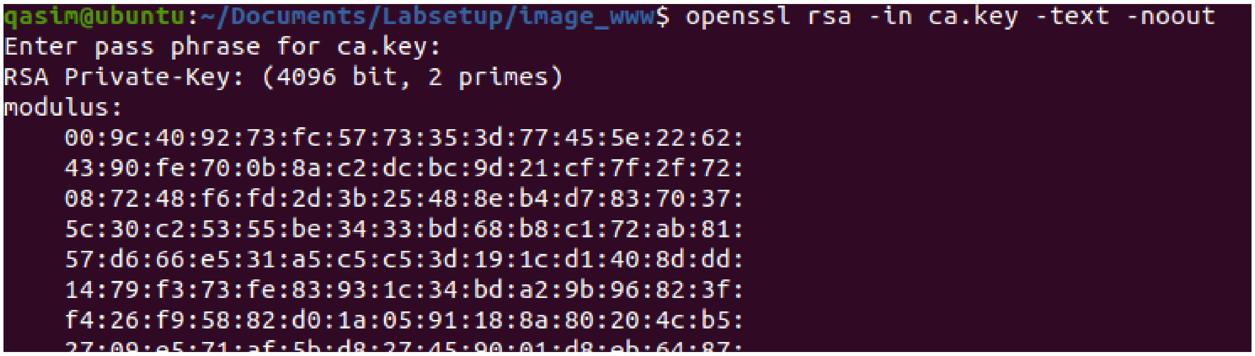
For our Certificate Authority, we create a self-signed certificate. This indicates that this CA is completely reliable, and the root certificate will be its certificate. Two files, ca.key and ca.crt, contain the command's output. The CA's private key is in the ca.key file, and the public-key certificate is in the ca.crt file.



```
qasim@ubuntu:~/Documents/Labsetup/image_www$ cat ca.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIJnDBOBgkqhkiG9w0BBQ0wQTApBgkqhkiG9w0BBQwwHAQILcVn0aQcZAMCAggA
MAwGCCqGSIb3DQIJBQAwFAYIKoZIhvcNAwcECOgjc73IsJfPBIIJSN9xIF59WOeW
yDSVKzXrbVj4Sb2jnib+I0UHrPDq7gxq8SF4lFjhmcO70En5ppNPnTy0fVpzHGJr
mN/RHmH1A0NW3+TVW1FDGWATl7wSb5tHiskRCJDv9kvVrZlhvqJFE2mxaRfypLut
W2FnIazFBkm3mGvzb0YUtEA2IliPvPZ/6LbMLMsCOP+FXfPoDsP5d1PdCUdKVUYO
IaE+NIIuVytn5pkZ6s6gh49x3Au/3/ii3w7Irrvd3pBGGAkeqw6XOHNLsbnBd85Z
kWgjy6BGq6eS0WgEkzkfxX2BvUmLCUm78K7cilmx9aqek7WPqvGzoW0MFc4tmALv
kGJVKH9DbOeAQTk70JSmNMM8CMWTDX5dLsfa4RrJSTiINBK8GlKPP+lADM4/z2vY
XpE+J2anDkycnyxKIi9kl9Sf6ppnS99Um/jNGk1GsS1FhNtFc6a9qoudOQ0X/wWI
QDxVbM+S4Na/Fa8j30wIL3O2dOR5jgvl7aB1GVgvXZFyILVPOtYYCIJfQv7pz/KL
ADcPsMpLqQlKd/wvgnyMwxjFx2iZ0SAeSkjm5kAwWUQ3fOX9RGo/mML6z31kseUI
VYiLGDE6KdhXpOP26rxLOSsUzCGLH08ekTsASVNcW1djnk2QicvfBpmbMwuhkJhR
T4eDAOjRN8LSu7ylsREPgV9b2idNeBC9L+DJIxX71Oalm6G6H5T6XycgydI0ZYPk
VZXXGwQxC4PK+mia1o3MZ+QcMimZTZfTv0NLIHz1TJJya/1hFJxRm4B/14EFzTY5
0PcJEp2O2F428Q6aoF2RPjCs1PfX2j8Tu0XNpFantZzmtF17UdHFIPboqzUcCR0P
fllHyMLFIakf2/4EeWnhp3ZFVX5hEmNUMRYxkZWyn4SbsKFMfsMPpETELPZJ2iYH
SQBUJLw7SFOlK4WlbDAbdOZmfEdf5nAQ2RT02NNSBY0dCJs5CFhCaG0fbMJU1EIg
JPPB7HFNmLMP7f6QEVSfJA+DlNeQxVNkfEkK7i6rICVvXNskaoir41wdcqhR6Jmi
q3LH3ed3+0vricwnmWigqduxGAYI3kEGsO0HQy3QCuyf3V7L0zw9SGI7RlmBDOxY
d9aoqcwrZkHVja/N1JFhVS3LA507gAh+VxS4tPovqcYLSL9YW0pX5ivSSTmq8r5g
ilfjU5Bd91G2pGtEiEBOZQzb9G2wOSGEOWJz9V6qY/+bioBFiVUUiIWgGmrNXAin
```



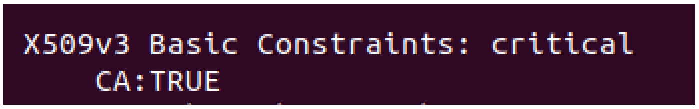
```
qasim@ubuntu:~/Documents/Labsetup/image_www$ cat ca.crt
-----BEGIN CERTIFICATE-----
MIIFXzCCA0egAwIBAgIUFlCjRsAyzpkT8EVTCKV9V8ERGC4wDQYJKoZIhvcNAQEL
BQAwPzEYMBYGA1UEAwwPd3d3Lm1vZGVsQ0EuY29tMRYwFAYDVQQKDA1Nb2RlbCBD
QSBMVEQuMQswCQYDVQQGEwJVUzAeFw0yNDEwMjUwOTAxMzFaFw0zNDEwMjMwOTAx
MzFaMD8xGDAWBgNVBAMMD3d3dy5tb2RlbENBLmNvbTEWMBQGA1UECgwNTW9kZWwg
Q0EgTFRELjELMAkGA1UEBhMCVVMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCcQJJz/FdzNT13RV4iYkOQ/nALisLcvJ0hz38vcghySPb9LTslSI6014Nw
N1wwwlNVvjQzvWi4wXKrgVfWZuUxpcXFPRkc0UCN3RR583P+g5McNL2im5aCP/Qm
+ViC0BoFkRiKgCBMtScJ5XGvW9gnRZAB2Otkh+CFfRylxNYsDCvfTq6oxnYImAKl
0cxx1BhwfDx+iFKbAilVpDvbktDqVMLuuhDnt9ilT4EoCKCqft/5P2Q0bmxMxxZu
FHZiI013KLYuFFYZzZLOjzwY4IM3sHV5vtEpz+yMjNiwLBK9wQ9FS4Kgb9hsZDrt
oWx7yol5OQuXdFxq6NlKnh6pykQ+ecLo5NPHpLmiK3Ycvh9yGJs8nY/1hg0opPEL
yTrmDr6Lidqc7xDoV5LWybW0vpXo296r3D0UGKwMpx2Xe3Tx3DjbfqlX4DnNgDr5
iUhhPROZ3qrQx6q0VfEj9/oxHAW3IEnYl7QIdBQeXCj164aOG19HhckJueIirMoj
Jjqe1cB19d8hQMyX8nJ7pCrL2Hr51r93ClaKeqkUzUmNdDYKOsM4G3klF1G1tdLr
s2hIOS4Y5vt6NKtsPUZgPVlAM3KEgMpKWk97+2WmkMVumuKDByRCzhexrwXQfG39
t8tP3P/1EpYqCXXadBpAp2vislDtjNAr5BXzqo/wlGSA8mQ7wQIDAQABo1MwUTAd
BgNVHQ4EFgQU0QEATibQy/3b+7XVdmeljRCg1fwwHwYDVR0jBBgwFoAU0QEATibQ
y/3b+7XVdmeljRCg1fwwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOC
AgEAlbKRPQDN1FrtihaS0I2Hkld9m+KHGd/b1bWGQ8Z2qlSgRKAO4KwJh0yJckj+
XrlRZQMAeape2aCfbc8dV+riWuzNdXwNAO1BQpvoXHXtVDnjVMJaEjNf/nh2OGCR
GeBIWJyd0vOnPbZG8j6y+wf1T7guUd8vQy0h3FhvXJrT8G3/OwWvdthY38KXnx/c
5X95S3coHLLN73fwX3YVeU9PbNFf/x4aP5oViRU6s9vS5s52fGs7XbtZwBzRXoKS
8zDA9GCfibyNr3o/IB9zPdfbCnx0aFROiIjmFOxAKqLn2mX/6jjdzQNedqlRUG+l
wlnoRRjp6oOZ9oiPpiRPCp/oEmvgMCYQSXnO/HlFRZyAaQPbLyNIbPVUs/MckA/e
FaZmkHxdl2Pn9Hyc9wt7of1anZaIl23/4KrgK2rwXEOI9az/yZHV3FFBUvwulQ0K
tOuSeQ5nQmqOXjXIdwLrKpVI7rAdzlqwlhL1019ND9CE0d+44cDubnUNN00hloBZ
hTPEZIjOh8odcQ3OcSzUpE4scDC7gi9NOPcsa8ZkXDaW6mMnCZclwSEtHctQgf7j
cSDGxzX9mFkJlv2lXqPy+ToXzKEKQ/LP3u+wa46cqdeQ2yCpm9dZVWdVQc2TBKUL
FkLWUHye8bI++BBtTAzFLtMMvNyKv8SIaamRW4rTDcokOJ4=
-----END CERTIFICATE-----
```



```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl rsa -in ca.key -text -noout
Enter pass phrase for ca.key:
RSA Private-Key: (4096 bit, 2 primes)
modulus:
    00:9c:40:92:73:fc:57:73:35:3d:77:45:5e:22:62:
    43:90:fe:70:0b:8a:c2:dc:bc:9d:21:cf:7f:2f:72:
    08:72:48:f6:fd:2d:3b:25:48:8e:b4:d7:83:70:37:
    5c:30:c2:53:55:be:34:33:bd:68:b8:c1:72:ab:81:
    57:d6:66:e5:31:a5:c5:c5:3d:19:1c:d1:40:8d:dd:
    14:79:f3:73:fe:83:93:1c:34:bd:a2:9b:96:82:3f:
    f4:26:f9:58:82:d0:1a:05:91:18:8a:80:20:4c:b5:
    27:09:e5:71:af:5b:d8:27:45:90:01:d8:eb:64:87:
```

OBSERVATIONS:

1. Indicating a CA Certificate:

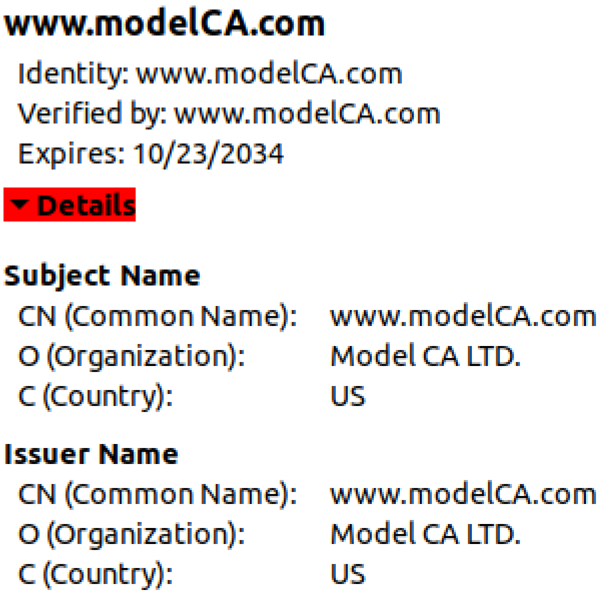


```
X509v3 Basic Constraints: critical
    CA:TRUE
```

The line CA:TRUE under X509v3 Basic Constraints verifies that this certificate is configured as a Certificate Authority (CA) certificate. This setting allows it to issue other certificates. The generated certificate is identified as a CA certificate by the issuer and the subject.

2. Indicating a Self-Signed Certificate:

- Check the Issuer and Subject fields in the certificate output. For a self-signed certificate, both fields should be identical. Authority key and subject key indicates that this is a self-signed certificate only.



www.modelCA.com

Identity: www.modelCA.com
Verified by: www.modelCA.com
Expires: 10/23/2034

▾ Details

Subject Name

CN (Common Name):	www.modelCA.com
O (Organization):	Model CA LTD.
C (Country):	US

Issuer Name

CN (Common Name):	www.modelCA.com
O (Organization):	Model CA LTD.
C (Country):	US

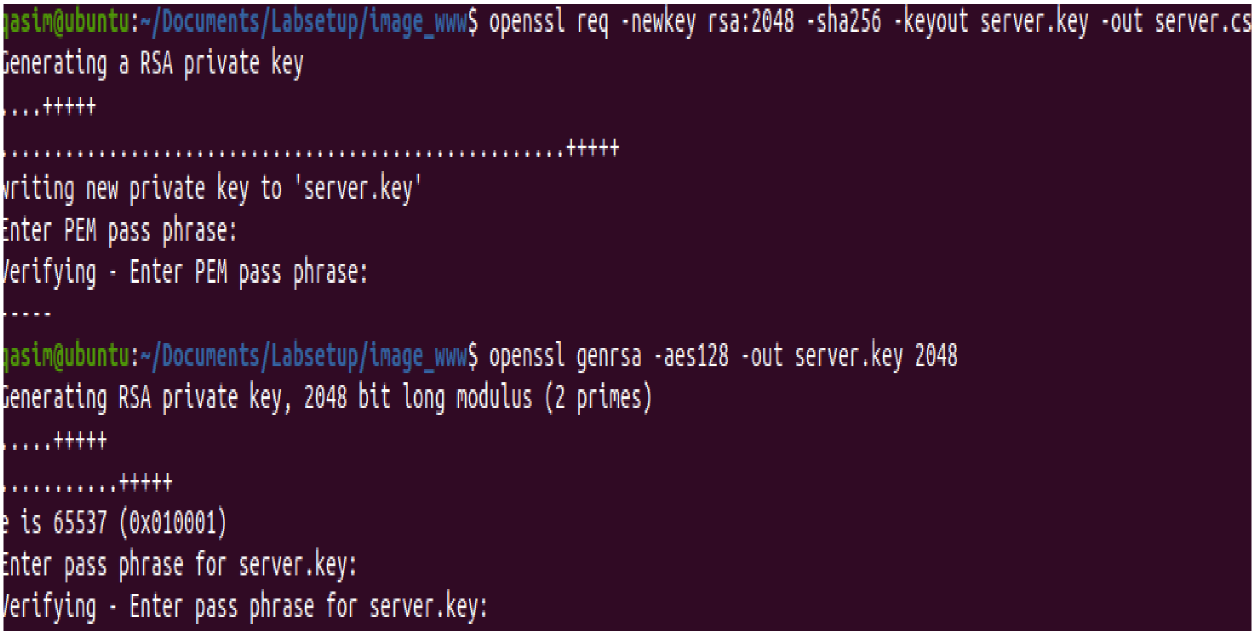
3. RSA Algorithm Values:

- In the RSA key output, find the following elements:
  - Modulus (n): The large integer value labeled as modulus.
  - Public Exponent (e): Found near publicExponent, with a value of 65537.
  - Private Exponent (d): Look for privateExponent.
  - Prime Factors (p and q): Found under prime1 (p) and prime2 (q).

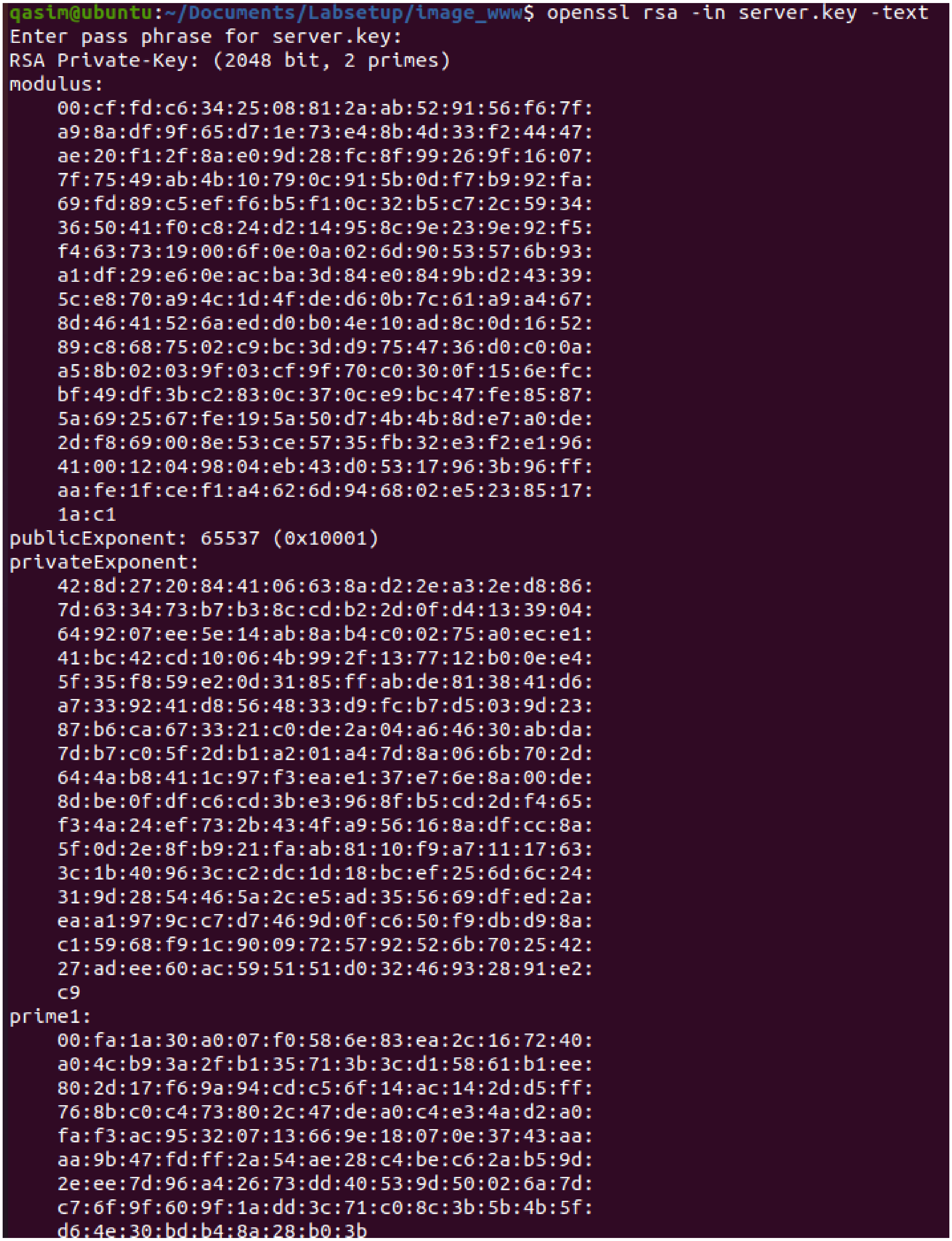
TASK 2:

Step 1: Generate public/private key pair.

- We can run the following command to generate an RSA key pair (both private and public keys). We also provide a password(admin) to encrypt the private key. The keys will be stored in the file *server.key*



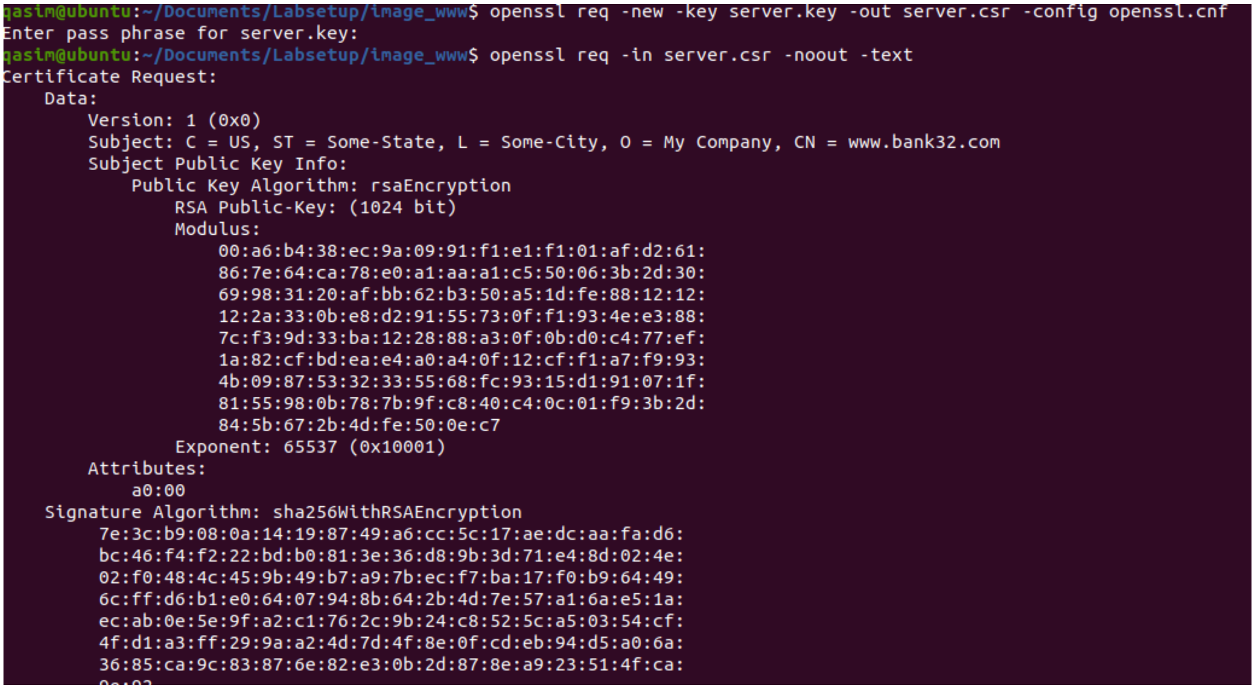
```
asim@ubuntu:~/Documents/Labsetup/image_www$ openssl req -newkey rsa:2048 -sha256 -keyout server.key -out server.cs
enerating a RSA private key
...+++++
...................................................+++++
riting new private key to 'server.key'
nter PEM pass phrase:
erifying - Enter PEM pass phrase:
----
asim@ubuntu:~/Documents/Labsetup/image_www$ openssl genrsa -aes128 -out server.key 2048
enerating RSA private key, 2048 bit long modulus (2 primes)
....+++++
..........+++++
 is 65537 (0x010001)
nter pass phrase for server.key:
erifying - Enter pass phrase for server.key:
```



```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
RSA Private-Key: (2048 bit, 2 primes)
modulus:
    00:cf:fd:c6:34:25:08:81:2a:ab:52:91:56:f6:7f:
    a9:8a:df:9f:65:d7:1e:73:e4:8b:4d:33:f2:44:47:
    ae:20:f1:2f:8a:e0:9d:28:fc:8f:99:26:9f:16:07:
    7f:75:49:ab:4b:10:79:0c:91:5b:0d:f7:b9:92:fa:
    69:fd:89:c5:ef:f6:b5:f1:0c:32:b5:c7:2c:59:34:
    36:50:41:f0:c8:24:d2:14:95:8c:9e:23:9e:92:f5:
    f4:63:73:19:00:6f:0e:0a:02:6d:90:53:57:6b:93:
    a1:df:29:e6:0e:ac:ba:3d:84:e0:84:9b:d2:43:39:
    5c:e8:70:a9:4c:1d:4f:de:d6:0b:7c:61:a9:a4:67:
    8d:46:41:52:6a:ed:d0:b0:4e:10:ad:8c:0d:16:52:
    89:c8:68:75:02:c9:bc:3d:d9:75:47:36:d0:c0:0a:
    a5:8b:02:03:9f:03:cf:9f:70:c0:30:0f:15:6e:fc:
    bf:49:df:3b:c2:83:0c:37:0c:e9:bc:47:fe:85:87:
    5a:69:25:67:fe:19:5a:50:d7:4b:4b:8d:e7:a0:de:
    2d:f8:69:00:8e:53:ce:57:35:fb:32:e3:f2:e1:96:
    41:00:12:04:98:04:eb:43:d0:53:17:96:3b:96:ff:
    aa:fe:1f:ce:f1:a4:62:6d:94:68:02:e5:23:85:17:
    1a:c1
publicExponent: 65537 (0x10001)
privateExponent:
    42:8d:27:20:84:41:06:63:8a:d2:2e:a3:2e:d8:86:
    7d:63:34:73:b7:b3:8c:cd:b2:2d:0f:d4:13:39:04:
    64:92:07:ee:5e:14:ab:8a:b4:c0:02:75:a0:ec:e1:
    41:bc:42:cd:10:06:4b:99:2f:13:77:12:b0:0e:e4:
    5f:35:f8:59:e2:0d:31:85:ff:ab:de:81:38:41:d6:
    a7:33:92:41:d8:56:48:33:d9:fc:b7:d5:03:9d:23:
    87:b6:ca:67:33:21:c0:de:2a:04:a6:46:30:ab:da:
    7d:b7:c0:5f:2d:b1:a2:01:a4:7d:8a:06:6b:70:2d:
    64:4a:b8:41:1c:97:f3:ea:e1:37:e7:6e:8a:00:de:
    8d:be:0f:df:c6:cd:3b:e3:96:8f:b5:cd:2d:f4:65:
    f3:4a:24:ef:73:2b:43:4f:a9:56:16:8a:df:cc:8a:
    5f:0d:2e:8f:b9:21:fa:ab:81:10:f9:a7:11:17:63:
    3c:1b:40:96:3c:c2:dc:1d:18:bc:ef:25:6d:6c:24:
    31:9d:28:54:46:5a:2c:e5:ad:35:56:69:df:ed:2a:
    ea:a1:97:9c:c7:d7:46:9d:0f:c6:50:f9:db:d9:8a:
    c1:59:68:f9:1c:90:09:72:57:92:52:6b:70:25:42:
    27:ad:ee:60:ac:59:51:51:d0:32:46:93:28:91:e2:
    c9
prime1:
    00:fa:1a:30:a0:07:f0:58:6e:83:ea:2c:16:72:40:
    a0:4c:b9:3a:2f:b1:35:71:3b:3c:d1:58:61:b1:ee:
    80:2d:17:f6:9a:94:cd:c5:6f:14:ac:14:2d:d5:ff:
    76:8b:c0:c4:73:80:2c:47:de:a0:c4:e3:4a:d2:a0:
    fa:f3:ac:95:32:07:13:66:9e:18:07:0e:37:43:aa:
    aa:9b:47:fd:ff:2a:54:ae:28:c4:be:c6:2a:b5:9d:
    2e:ee:7d:96:a4:26:73:dd:40:53:9d:50:02:6a:7d:
    c7:6f:9f:60:9f:1a:dd:3c:71:c0:8c:3b:5b:4b:5f:
    d6:4e:30:bd:b4:8a:28:b0:3b
```

Step 2: Generate a Certificate Signing Request (CSR)

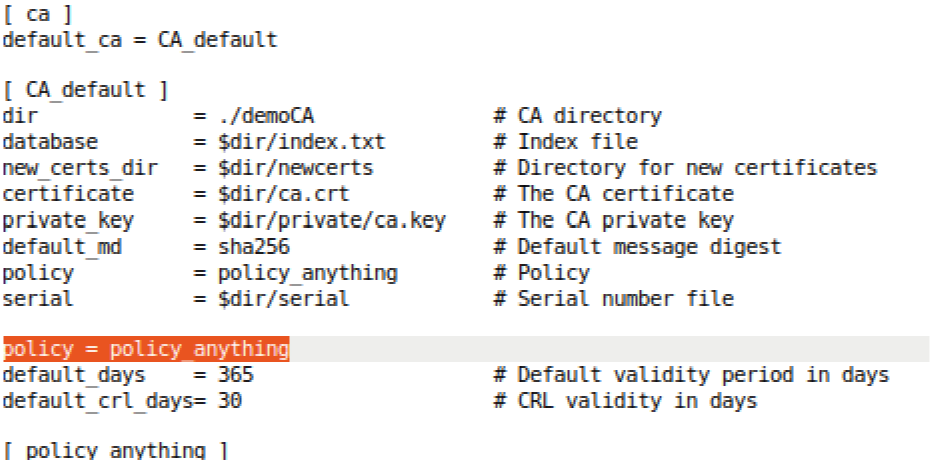
Since we now have the key file, we create a CSR, which essentially contains the public key of the business. The CA will create a certificate for the key after receiving the CSR. Once the server has been mapped to the local host, we can see that it can be easily started and launched using server.pem. We only used server.key and server.crt to confirm that we could accomplish this. But since the reliable business hasn't signed the certificate, the browser does not yet have faith in the URL of the server.



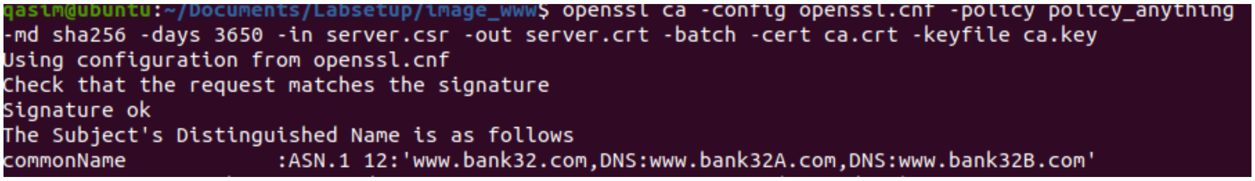
```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl req -new -key server.key -out server.csr -config openssl.cnf
Enter pass phrase for server.key:
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl req -in server.csr -noout -text
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = US, ST = Some-State, L = Some-City, O = My Company, CN = www.bank32.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:a6:b4:38:ec:9a:09:91:f1:e1:f1:01:af:d2:61:
                    86:7e:64:ca:78:e0:a1:aa:a1:c5:50:06:3b:2d:30:
                    69:98:31:20:af:bb:62:b3:50:a5:1d:fe:88:12:12:
                    12:2a:33:0b:e8:d2:91:55:73:0f:f1:93:4e:e3:88:
                    7c:f3:9d:33:ba:12:28:88:a3:0f:0b:d0:c4:77:ef:
                    1a:82:cf:bd:ea:e4:a0:a4:0f:12:cf:f1:a7:f9:93:
                    4b:09:87:53:32:33:55:68:fc:93:15:d1:91:07:1f:
                    81:55:98:0b:78:7b:9f:c8:40:c4:0c:01:f9:3b:2d:
                    84:5b:67:2b:4d:fe:50:0e:c7
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: sha256WithRSAEncryption
         7e:3c:b9:08:0a:14:19:87:49:a6:cc:5c:17:ae:dc:aa:fa:d6:
         bc:46:f4:f2:22:bd:b0:81:3e:36:d8:9b:3d:71:e4:8d:02:4e:
         02:f0:48:4c:45:9b:49:b7:a9:7b:ec:f7:ba:17:f0:b9:64:49:
         6c:ff:d6:b1:e0:64:07:94:8b:64:2b:4d:7e:57:a1:6a:e5:1a:
         ec:ab:0e:5e:9f:a2:c1:76:2c:9b:24:c8:52:5c:a5:03:54:cf:
         4f:d1:a3:ff:29:9a:a2:4d:7d:4f:8e:0f:cd:eb:94:d5:a0:6a:
         36:85:ca:9c:83:87:6e:82:e3:0b:2d:87:8e:a9:23:51:4f:ca:
```

TASK 3:

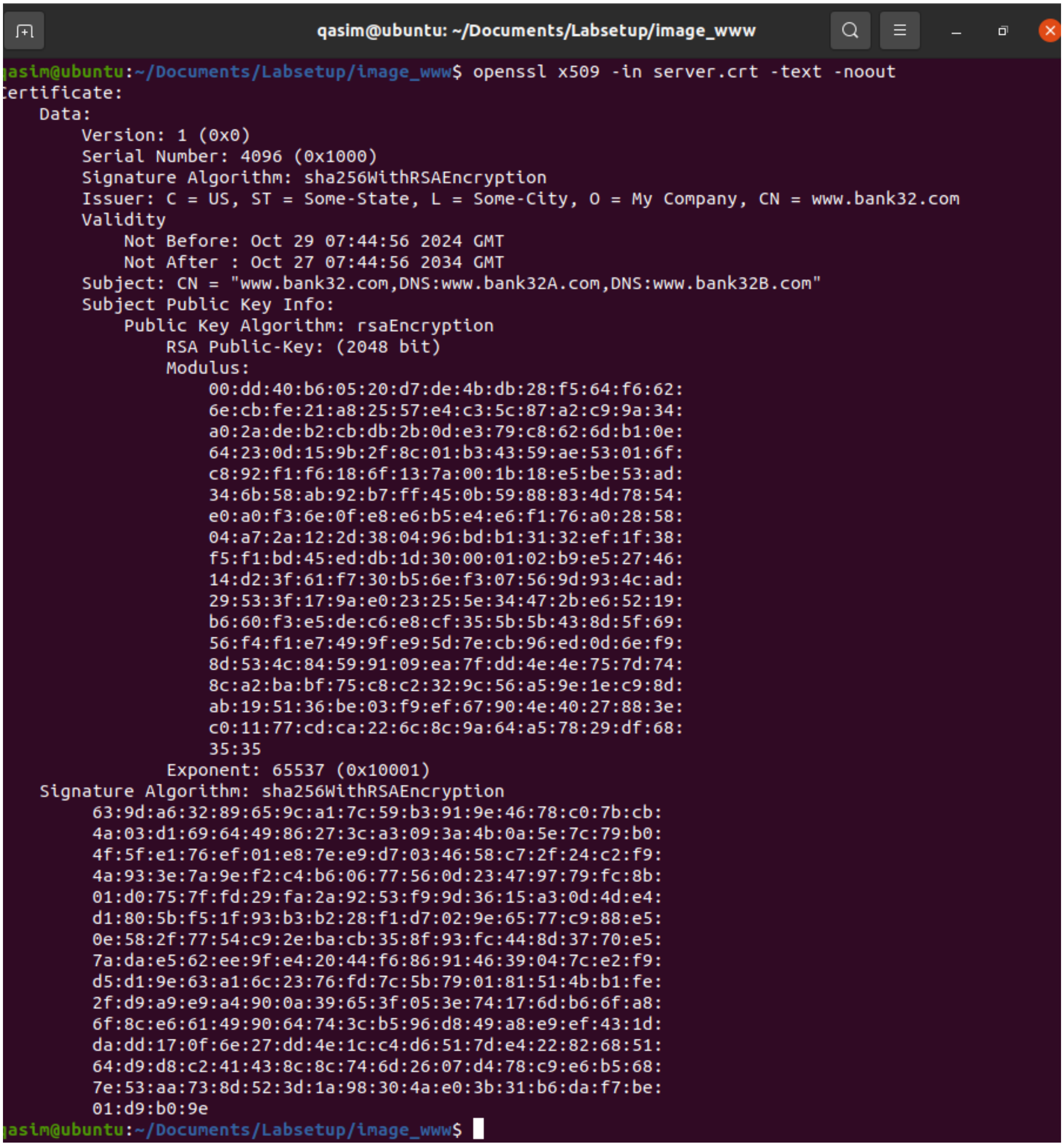
We will first remove the comments from a few lines so that we can copy our final certificate's extension.



```
[ ca ]
default_ca = CA_default

[ CA_default ]
dir             = ./demoCA              # CA directory
database        = $dir/index.txt        # Index file
new_certs_dir   = $dir/newcerts         # Directory for new certificates
certificate     = $dir/ca.crt           # The CA certificate
private_key     = $dir/private/ca.key   # The CA private key
default_md      = sha256                # Default message digest
policy          = policy_anything       # Policy
serial          = $dir/serial           # Serial number file

policy = policy_anything
default_days    = 365                   # Default validity period in days
default_crl_days= 30                    # CRL validity in days

[ policy_anything ]
```



```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl ca -config 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 openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'www.bank32.com,DNS:www.bank32A.com,DNS:www.bank32B.com'
```

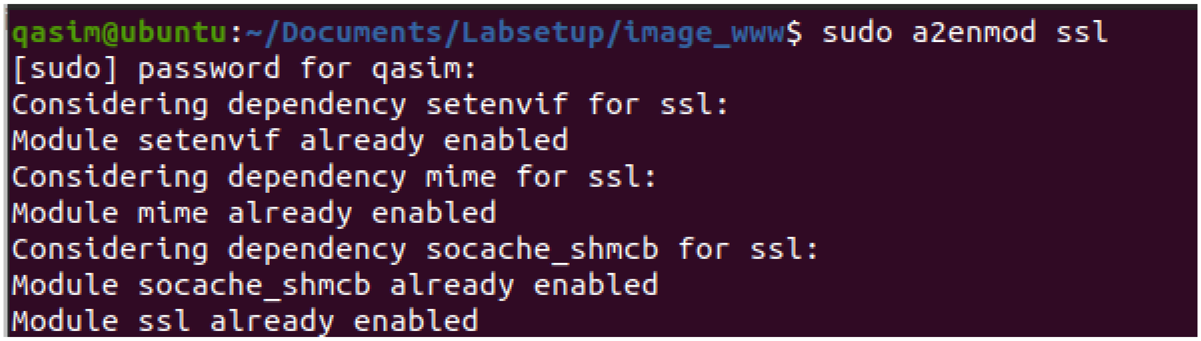


```
asim@ubuntu:~/Documents/Labsetup/image_www$ openssl x509 -in server.crt -text -noout
ertificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 4096 (0x1000)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, ST = Some-State, L = Some-City, O = My Company, CN = www.bank32.com
        Validity
            Not Before: Oct 29 07:44:56 2024 GMT
            Not After : Oct 27 07:44:56 2034 GMT
        Subject: CN = "www.bank32.com,DNS:www.bank32A.com,DNS:www.bank32B.com"
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:dd:40:b6:05:20:d7:de:4b:db:28:f5:64:f6:62:
                    6e:cb:fe:21:a8:25:57:e4:c3:5c:87:a2:c9:9a:34:
                    a0:2a:de:b2:cb:db:2b:0d:e3:79:c8:62:6d:b1:0e:
                    64:23:0d:15:9b:2f:8c:01:b3:43:59:ae:53:01:6f:
                    c8:92:f1:f6:18:6f:13:7a:00:1b:18:e5:be:53:ad:
                    34:6b:58:ab:92:b7:ff:45:0b:59:88:83:4d:78:54:
                    e0:a0:f3:6e:0f:e8:e6:b5:e4:e6:f1:76:a0:28:58:
                    04:a7:2a:12:2d:38:04:96:bd:b1:31:32:ef:1f:38:
                    f5:f1:bd:45:ed:db:1d:30:00:01:02:b9:e5:27:46:
                    14:d2:3f:61:f7:30:b5:6e:f3:07:56:9d:93:4c:ad:
                    29:53:3f:17:9a:e0:23:25:5e:34:47:2b:e6:52:19:
                    b6:60:f3:e5:de:c6:e8:cf:35:5b:5b:43:8d:5f:69:
                    56:f4:f1:e7:49:9f:e9:5d:7e:cb:96:ed:0d:6e:f9:
                    8d:53:4c:84:59:91:09:ea:7f:dd:4e:4e:75:7d:74:
                    8c:a2:ba:bf:75:c8:c2:32:9c:56:a5:9e:1e:c9:8d:
                    ab:19:51:36:be:03:f9:ef:67:90:4e:40:27:88:3e:
                    c0:11:77:cd:ca:22:6c:8c:9a:64:a5:78:29:df:68:
                    35:35
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
         63:9d:a6:32:89:65:9c:a1:7c:59:b3:91:9e:46:78:c0:7b:cb:
         4a:03:d1:69:64:49:86:27:3c:a3:09:3a:4b:0a:5e:7c:79:b0:
         4f:5f:e1:76:ef:01:e8:7e:e9:d7:03:46:58:c7:2f:24:c2:f9:
         4a:93:3e:7a:9e:f2:c4:b6:06:77:56:0d:23:47:97:79:fc:8b:
         01:d0:75:7f:fd:29:fa:2a:92:53:f9:9d:36:15:a3:0d:4d:e4:
         d1:80:5b:f5:1f:93:b3:b2:28:f1:d7:02:9e:65:77:c9:88:e5:
         0e:58:2f:77:54:c9:2e:ba:cb:35:8f:93:fc:44:8d:37:70:e5:
         7a:da:e5:62:ee:9f:e4:20:44:f6:86:91:46:39:04:7c:e2:f9:
         d5:d1:9e:63:a1:6c:23:76:fd:7c:5b:79:01:81:51:4b:b1:fe:
         2f:d9:a9:e9:a4:90:0a:39:65:3f:05:3e:74:17:6d:b6:6f:a8:
         6f:8c:e6:61:49:90:64:74:3c:b5:96:d8:49:a8:e9:ef:43:1d:
         da:dd:17:0f:6e:27:dd:4e:1c:c4:d6:51:7d:e4:22:82:68:51:
         64:d9:d8:c2:41:43:8c:8c:74:6d:26:07:d4:78:c9:e6:b5:68:
         7e:53:aa:73:8d:52:3d:1a:98:30:4a:e0:3b:31:b6:da:f7:be:
         01:d9:b0:9e
asim@ubuntu:~/Documents/Labsetup/image_www$
```

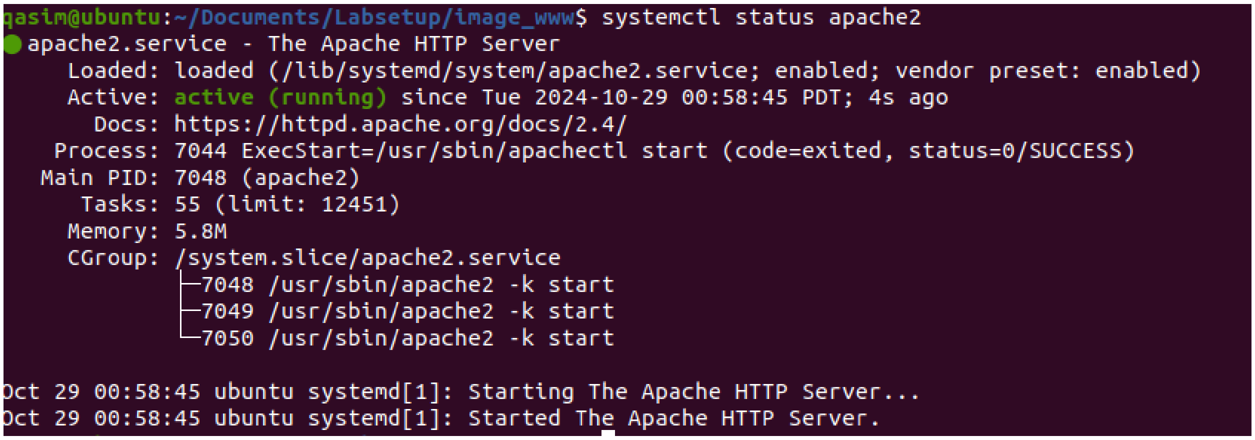
TASK 4:

We start by updating our present installation with the studo apt-get update command. The Apache package will then be installed using the subsequent command. Use sudo to install Apache 2.

Once this is enabled, use the following command to see the SSLs that are mod-available.

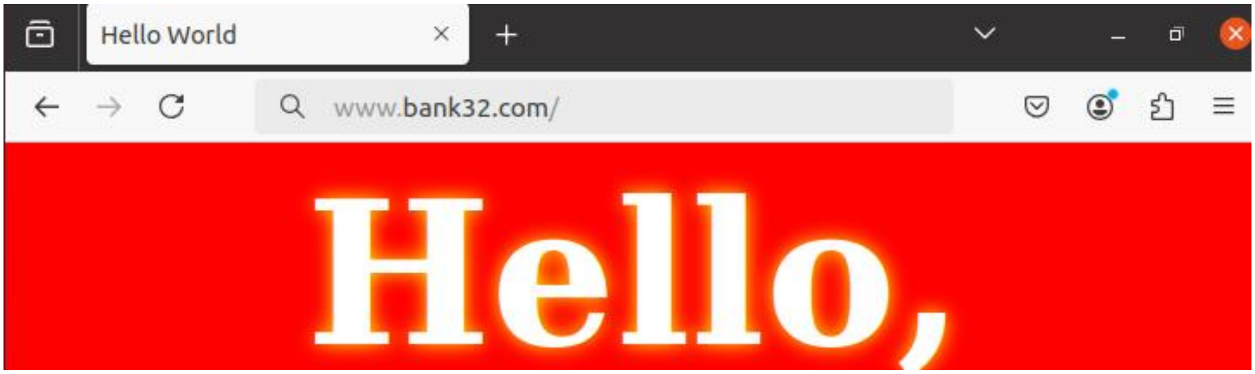


```
qasim@ubuntu:~/Documents/Labsetup/image_www$ sudo a2enmod ssl
[sudo] password for qasim:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```



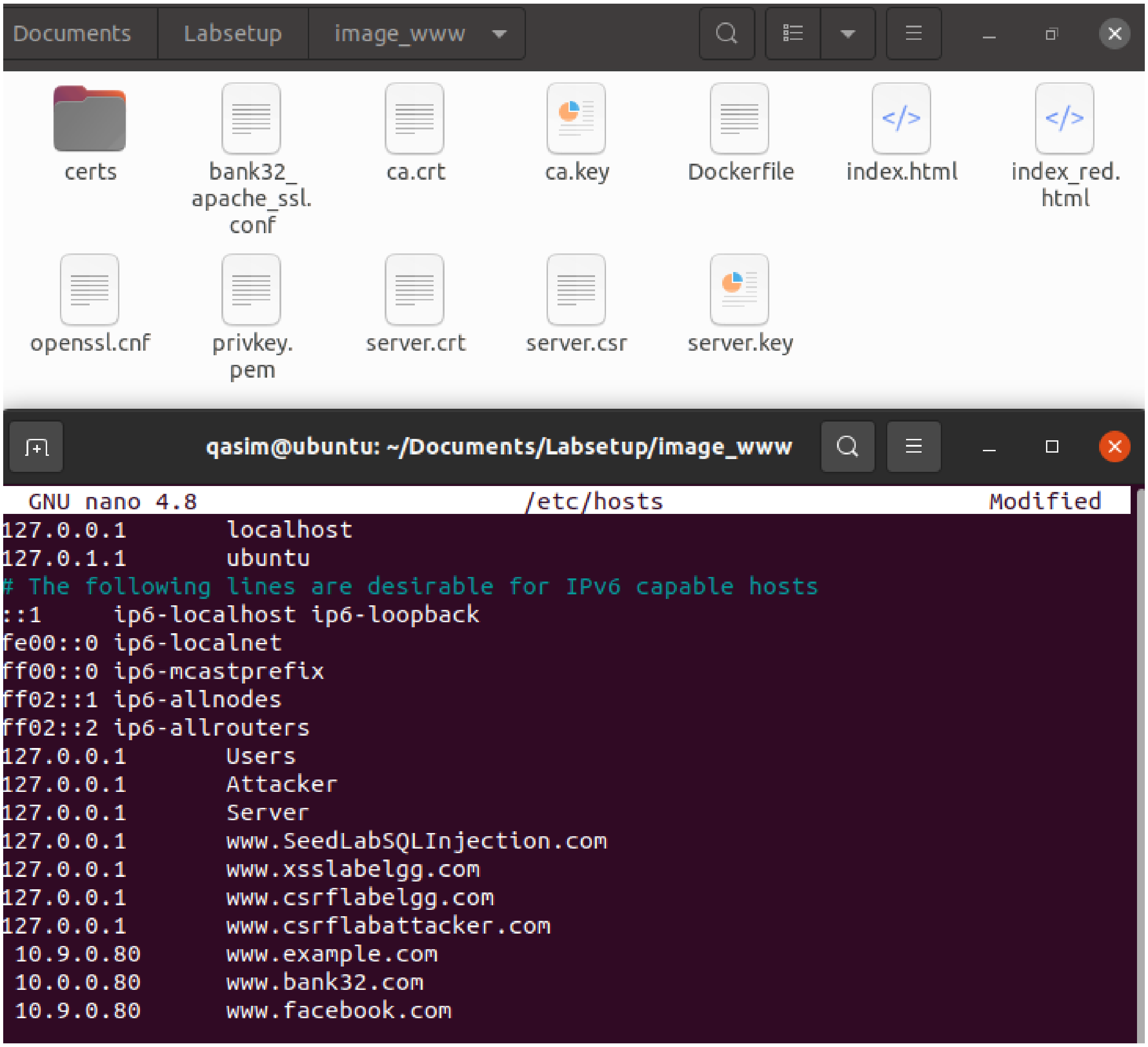
```
qasim@ubuntu:~/Documents/Labsetup/image_www$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-10-29 00:58:45 PDT; 4s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 7044 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 7048 (apache2)
      Tasks: 55 (limit: 12451)
     Memory: 5.8M
     CGroup: /system.slice/apache2.service
             ├─7048 /usr/sbin/apache2 -k start
             ├─7049 /usr/sbin/apache2 -k start
             └─7050 /usr/sbin/apache2 -k start

Oct 29 00:58:45 ubuntu systemd[1]: Starting The Apache HTTP Server...
Oct 29 00:58:45 ubuntu systemd[1]: Started The Apache HTTP Server.
```

Following the execution of the following command, we can access the following webpage to validate our conclusions. We discovered that the website is now insecure, therefore we downloaded the modelCA.crt file from the supplied zip file. This will now allow us to connect securely.



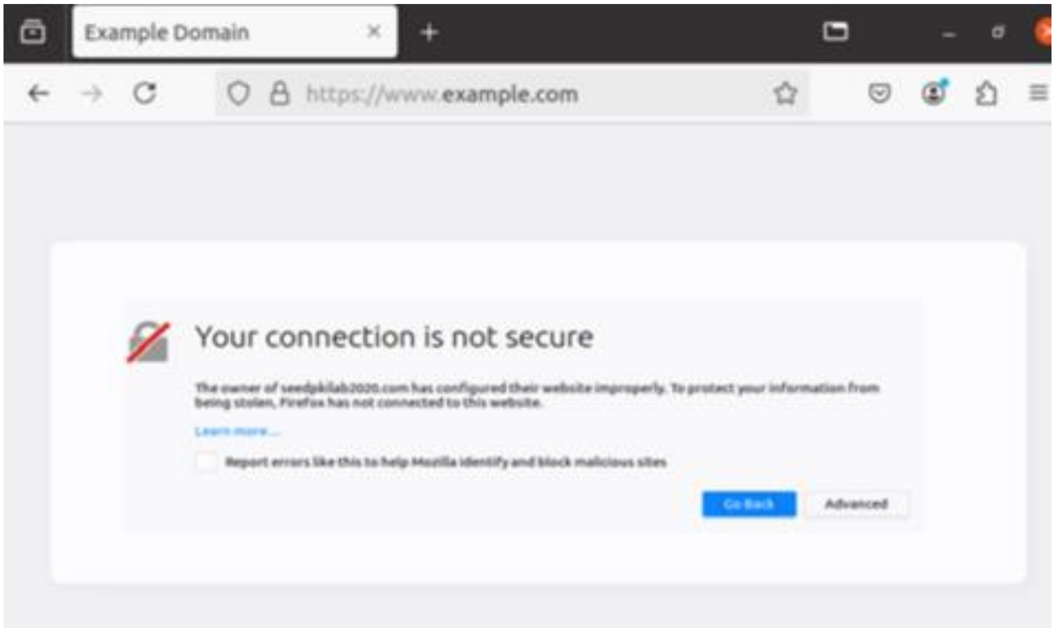
TASK 5:

Go to the file named hosts in the /etc directory The /etc/hosts file opens up. Adding the following entry in /etc/hosts file.



```
  GNU nano 4.8                        /etc/hosts                        Modified
127.0.0.1       localhost
127.0.1.1       ubuntu
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       Users
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
 10.9.0.80      www.example.com
 10.0.0.80      www.bank32.com
 10.9.0.80      www.facebook.com
```

After the website has been run, we can see that the previously indicated result is obtained. Secure connection is no longer formed since the domain www.example.com was not considered when we initially registered and created the certificates.



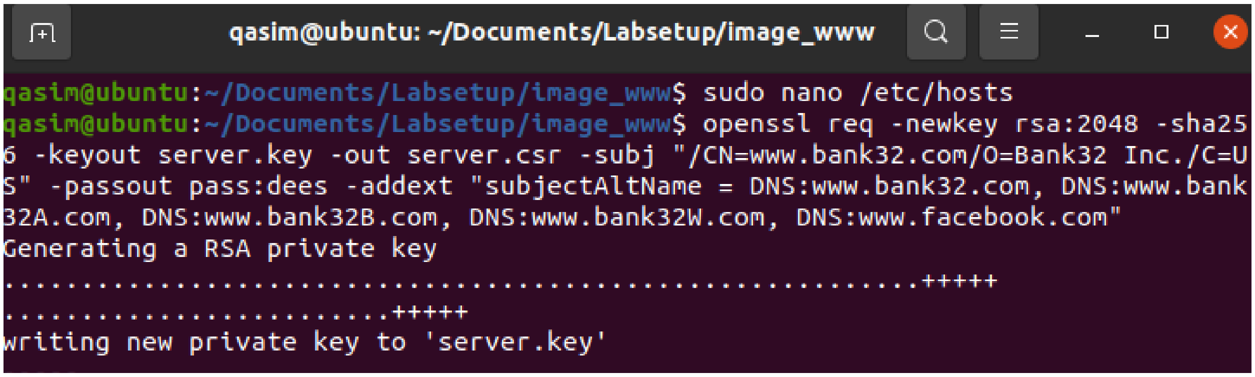
Your connection is not secure

The owner of seedpkilab2020.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

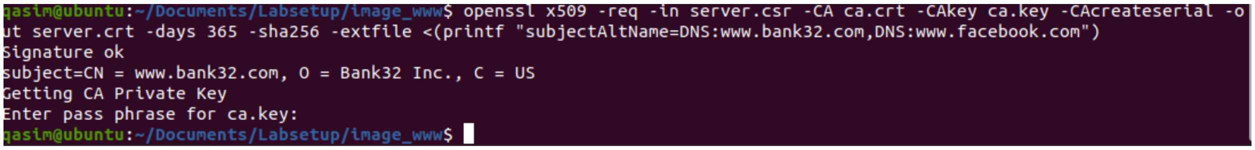
□ Report errors like this to help Mozilla identify and block malicious sites

TASK 6 : (Same setup as Task 5)

We will try a DNS cache poisoning attack, which involves rerouting the DNS to a different server, using Facebook.com as an example.

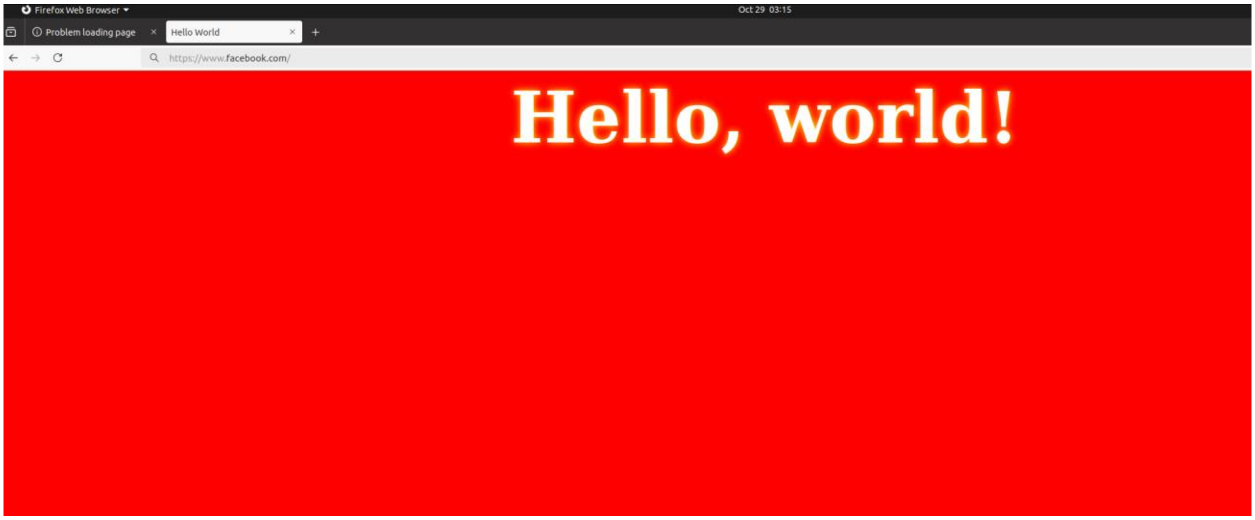


```
qasim@ubuntu:~/Documents/Labsetup/image_www$ sudo nano /etc/hosts
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl req -newkey rsa:2048 -sha25
6 -keyout server.key -out server.csr -subj "/CN=www.bank32.com/O=Bank32 Inc./C=U
S" -passout pass:dees -addext "subjectAltName = DNS:www.bank32.com, DNS:www.bank
32A.com, DNS:www.bank32B.com, DNS:www.bank32W.com, DNS:www.facebook.com"
Generating a RSA private key
...........................................................+++++
.........................+++++
writing new private key to 'server.key'
```



```
qasim@ubuntu:~/Documents/Labsetup/image_www$ openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -o
ut server.crt -days 365 -sha256 -extfile <(printf "subjectAltName=DNS:www.bank32.com,DNS:www.facebook.com")
Signature ok
subject=CN = www.bank32.com, O = Bank32 Inc., C = US
Getting CA Private Key
Enter pass phrase for ca.key:
qasim@ubuntu:~/Documents/Labsetup/image_www$ 
```

After executing the following command, we generated a ca.crt file for the user, which will now cause them to be diverted to a different server whenever they visit www.facebook.com.


Firefox Web Browser
Oct 29 03:15
Problem loading page
Hello World
https://www.facebook.com/

Hello, world!