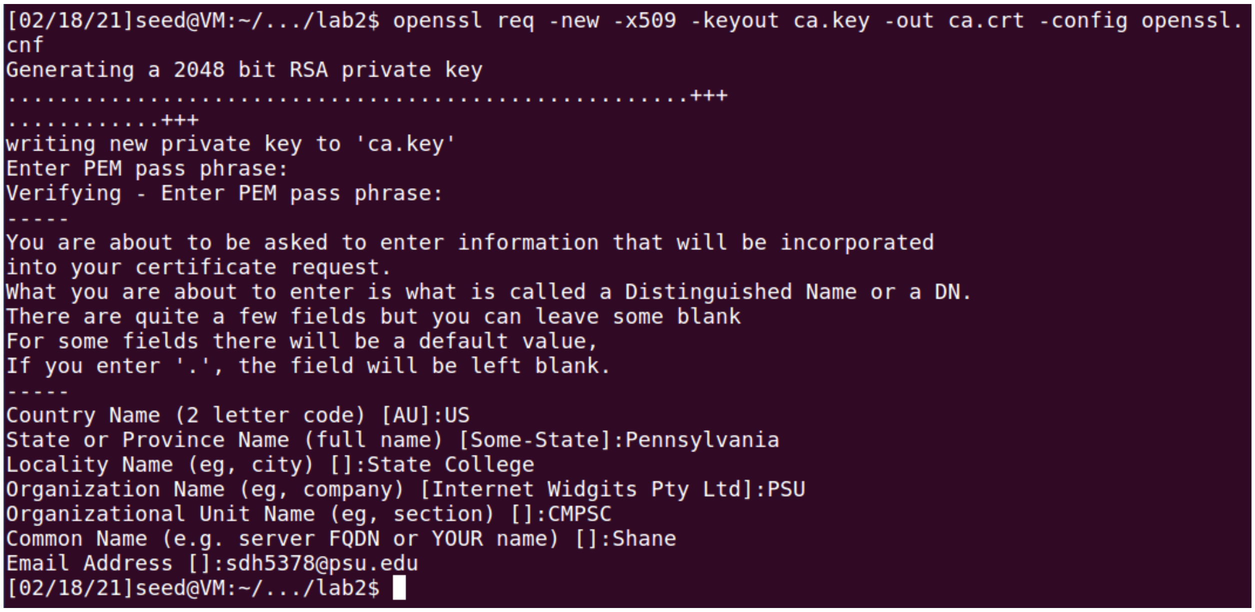
Shane Hagan

CMPSC 443 – Lab 2

02/26/21

Task 1

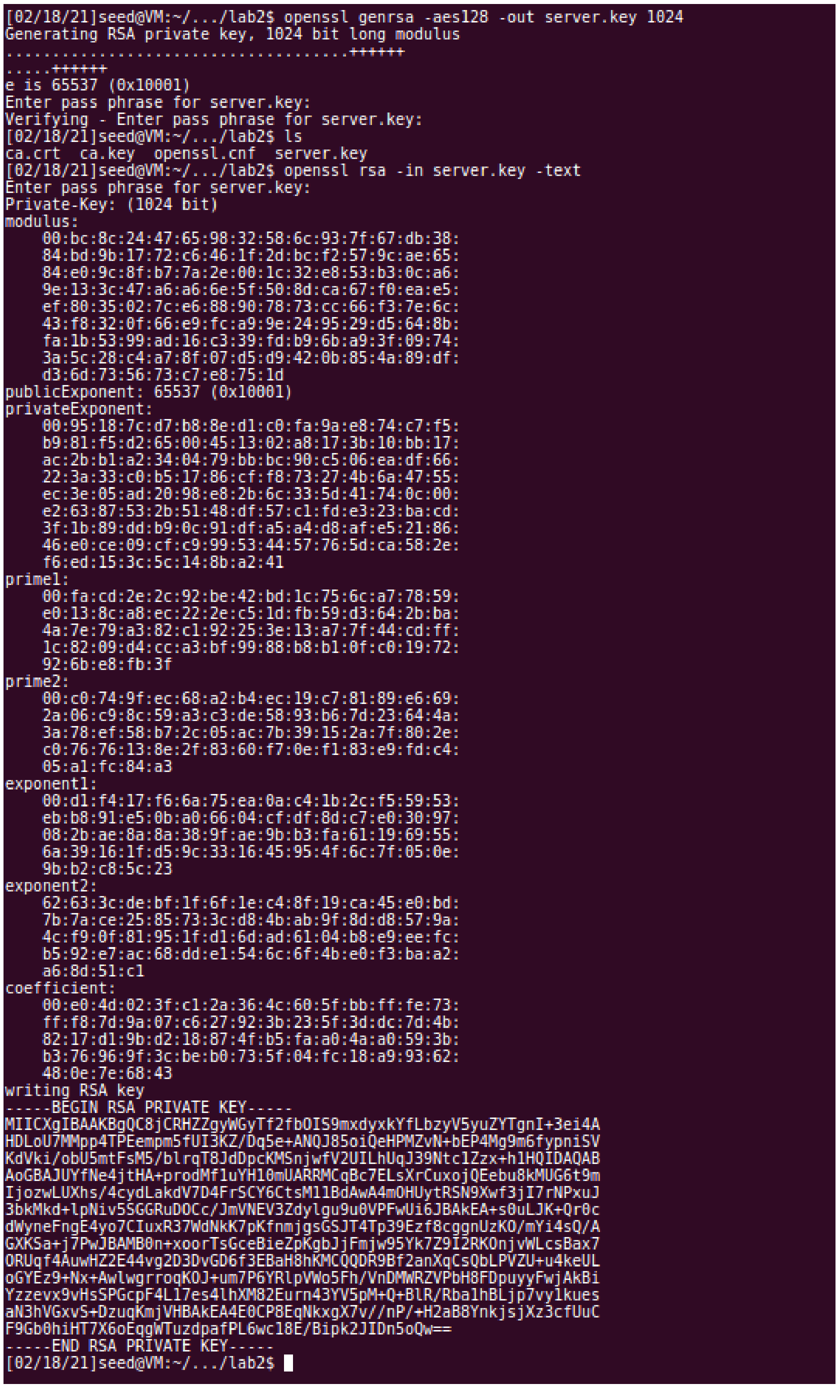


```
[02/18/21]seed@VM:~/.../lab2$ openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.
cnf
Generating a 2048 bit RSA private key
.....................................................+++
............+++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass 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]:US
State or Province Name (full name) [Some-State]:Pennsylvania
Locality Name (eg, city) []:State College
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PSU
Organizational Unit Name (eg, section) []:CMPSC
Common Name (e.g. server FQDN or YOUR name) []:Shane
Email Address []:sdh5378@psu.edu
[02/18/21]seed@VM:~/.../lab2$ 
```

Observation: public key certificate (ca.crt) and CA’s private key (ca.key) are generated using these commands

Task 2

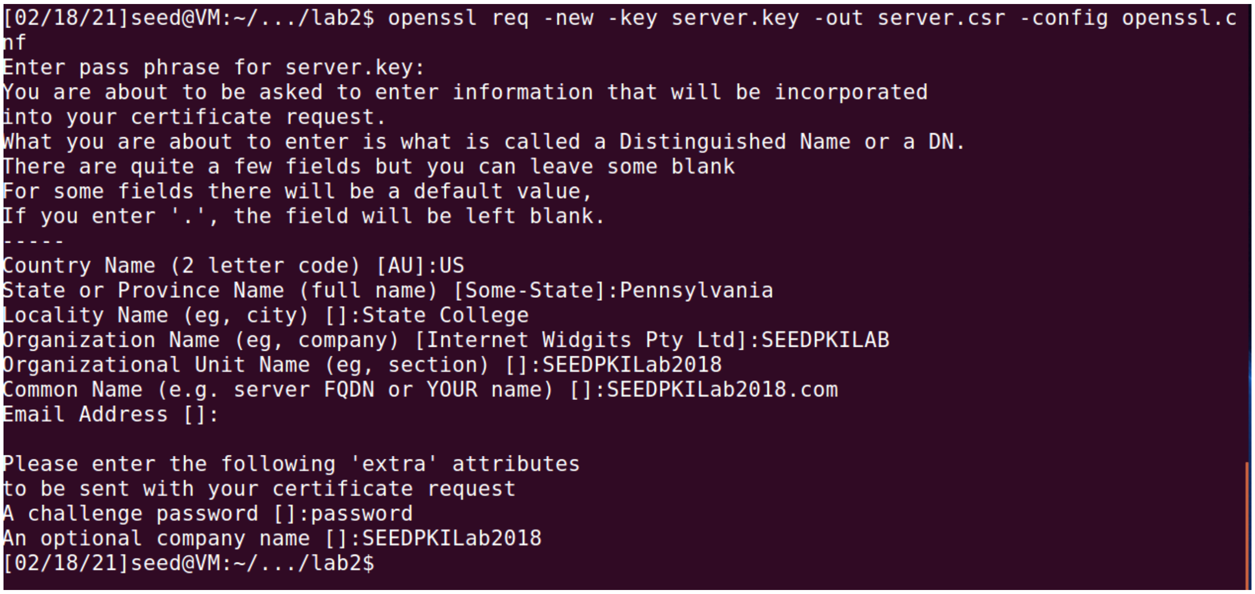
Step 1



```
[02/18/21]seed@VM:~/.../lab2$ openssl genrsa -aes128 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....................................++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
[02/18/21]seed@VM:~/.../lab2$ ls
ca.crt  ca.key  openssl.cnf  server.key
[02/18/21]seed@VM:~/.../lab2$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
    00:bc:8c:24:47:65:98:32:58:6c:93:7f:67:db:38:
    84:bd:9b:17:72:c6:46:1f:2d:bc:f2:57:9c:ae:65:
    84:e0:9c:8f:b7:7a:2e:00:1c:32:e8:53:b3:0c:a6:
    9e:13:3c:47:a6:a6:6e:5f:50:8d:ca:67:f0:ea:e5:
    ef:80:35:02:7c:e6:88:90:78:73:cc:66:f3:7e:6c:
    43:f8:32:0f:66:e9:fc:a9:9e:24:95:29:d5:64:8b:
    fa:1b:53:99:ad:16:c3:39:fd:b9:6b:a9:3f:09:74:
    3a:5c:28:c4:a7:8f:07:d5:d9:42:0b:85:4a:89:df:
    d3:6d:73:56:73:c7:e8:75:1d
publicExponent: 65537 (0x10001)
privateExponent:
    00:95:18:7c:d7:b8:8e:d1:c0:fa:9a:e8:74:c7:f5:
    b9:81:f5:d2:65:00:45:13:02:a8:17:3b:10:bb:17:
    ac:2b:b1:a2:34:04:79:bb:bc:90:c5:06:ea:df:66:
    22:3a:33:c0:b5:17:86:cf:f8:73:27:4b:6a:47:55:
    ec:3e:05:ad:20:98:e8:2b:6c:33:5d:41:74:0c:00:
    e2:63:87:53:2b:51:48:df:57:c1:fd:e3:23:ba:cd:
    3f:1b:89:dd:b9:0c:91:df:a5:a4:d8:af:e5:21:86:
    46:e0:ce:09:cf:c9:99:53:44:57:76:5d:ca:58:2e:
    f6:ed:15:3c:5c:14:8b:a2:41
prime1:
    00:fa:cd:2e:2c:92:be:42:bd:1c:75:6c:a7:78:59:
    e0:13:8c:a8:ec:22:2e:c5:1d:fb:59:d3:64:2b:ba:
    4a:7e:79:a3:82:c1:92:25:3e:13:a7:7f:44:cd:ff:
    1c:82:09:d4:cc:a3:bf:99:88:b8:b1:0f:c0:19:72:
    92:6b:e8:fb:3f
prime2:
    00:c0:74:9f:ec:68:a2:b4:ec:19:c7:81:89:e6:69:
    2a:06:c9:8c:59:a3:c3:de:58:93:b6:7d:23:64:4a:
    3a:78:ef:58:b7:2c:05:ac:7b:39:15:2a:7f:80:2e:
    c0:76:76:13:8e:2f:83:60:f7:0e:f1:83:e9:fd:c4:
    05:a1:fc:84:a3
exponent1:
    00:d1:f4:17:f6:6a:75:ea:0a:c4:1b:2c:f5:59:53:
    eb:b8:91:e5:0b:a0:66:04:cf:df:8d:c7:e0:30:97:
    08:2b:ae:8a:8a:38:9f:ae:9b:b3:fa:61:19:69:55:
    6a:39:16:1f:d5:9c:33:16:45:95:4f:6c:7f:05:0e:
    9b:b2:c8:5c:23
exponent2:
    62:63:3c:de:bf:1f:6f:1e:c4:8f:19:ca:45:e0:bd:
    7b:7a:ce:25:85:73:3c:d8:4b:ab:9f:8d:d8:57:9a:
    4c:f9:0f:81:95:1f:d1:6d:ad:61:04:b8:e9:ee:fc:
    b5:92:e7:ac:68:dd:e1:54:6c:6f:4b:e0:f3:ba:a2:
    a6:8d:51:c1
coefficient:
    00:e0:4d:02:3f:c1:2a:36:4c:60:5f:bb:ff:fe:73:
    ff:f8:7d:9a:07:c6:27:92:3b:23:5f:3d:dc:7d:4b:
    82:17:d1:9b:d2:18:87:4f:b5:fa:a0:4a:a0:59:3b:
    b3:76:96:9f:3c:be:b0:73:5f:04:fc:18:a9:93:62:
    48:0e:7e:68:43
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQC8jCRHZZgyWGyTf2fbOIS9mxdyxkYfLbzyV5yuZYTgnI+3ei4A
HDLoU7MMpp4TPEempm5fUI3KZ/Dq5e+ANQJ85oiQeHPMZvN+bEP4Mg9m6fypniSV
KdVki/obU5mtFsM5/blrqT8JdDpcKMSnjwfV2UILhUqJ39Ntc1Zzx+h1HQIDAQAB
AoGBAJUYfNe4jtHA+prodMf1uYH10mUARRMCqBc7ELsXrCuxojQEebu8kMUG6t9m
IjozwLUXhs/4cydLakdV7D4FrSCY6CtsM11BdAwA4mOHUytRSN9Xwf3jI7rNPxuJ
3bkMkd+lpNiv5SGGRuDOCc/JmVNEV3Zdylgu9u0VPFwUi6JBAkEA+s0uLJK+Qr0c
dWyneFngE4yo7CIuxR37WdNkK7pKfnmjgsGSJT4Tp39Ezf8cggnUzKO/mYi4sQ/A
GXKSa+j7PwJBAMB0n+xoorTsGceBieZpKgbJjFmjw95Yk7Z9I2RKOnjvWLcsBax7
ORUqf4AuwHZ2E44vg2D3DvGD6f3EBaH8hKMCQQDR9Bf2anXqCsQbLPVZU+u4keUL
oGYEz9+Nx+AwlwgrroqKOJ+um7P6YRlpVWo5Fh/VnDMWRZVPbH8FDpuyyFwjAkBi
Yzzevx9vHsSPGcpF4Ll7es4lhXM82Eurn43YV5pM+Q+BlR/Rba1hBLjp7vy1kues
aN3hVGxvS+DzuqKmjVHBAkEA4E0CP8EqNkxgX7v//nP/+H2aB8YnkjsjXz3cfUuC
F9Gb0hiHT7X6oEqgWTuzdpafPL6wc18E/Bipk2JIDn5oQw==
-----END RSA PRIVATE KEY-----
[02/18/21]seed@VM:~/.../lab2$ 
```

Observation: a public and private RSA key pair are generated using openssl and it is stores within the “server.key” file

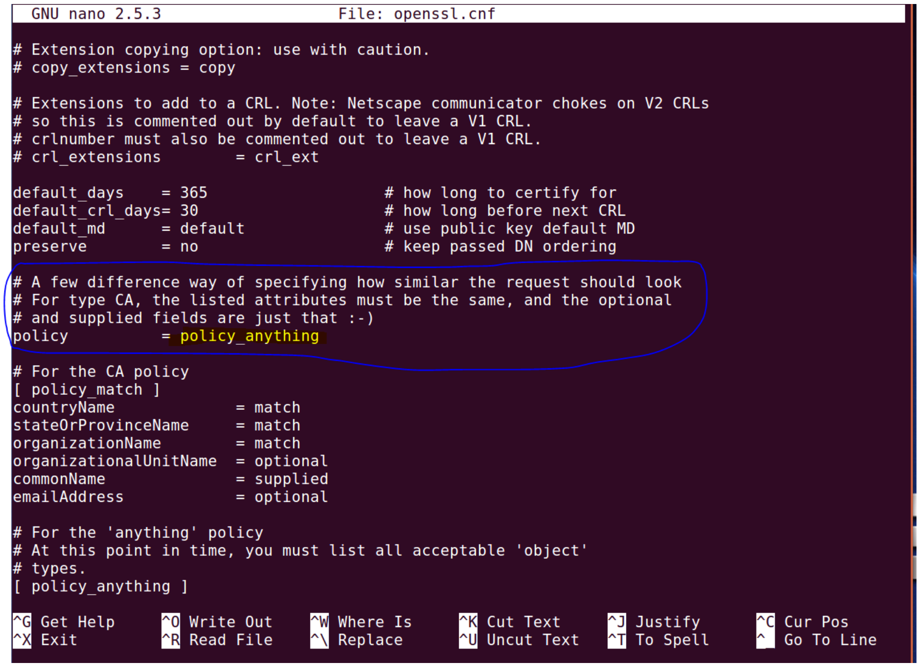
Step 2



```
[02/18/21]seed@VM:~/.../lab2$ openssl req -new -key server.key -out server.csr -config openssl.c
nf
Enter pass phrase for server.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]:US
State or Province Name (full name) [Some-State]:Pennsylvania
Locality Name (eg, city) []:State College
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SEEDPKILAB
Organizational Unit Name (eg, section) []:SEEDPKILab2018
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2018.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:SEEDPKILab2018
[02/18/21]seed@VM:~/.../lab2$
```

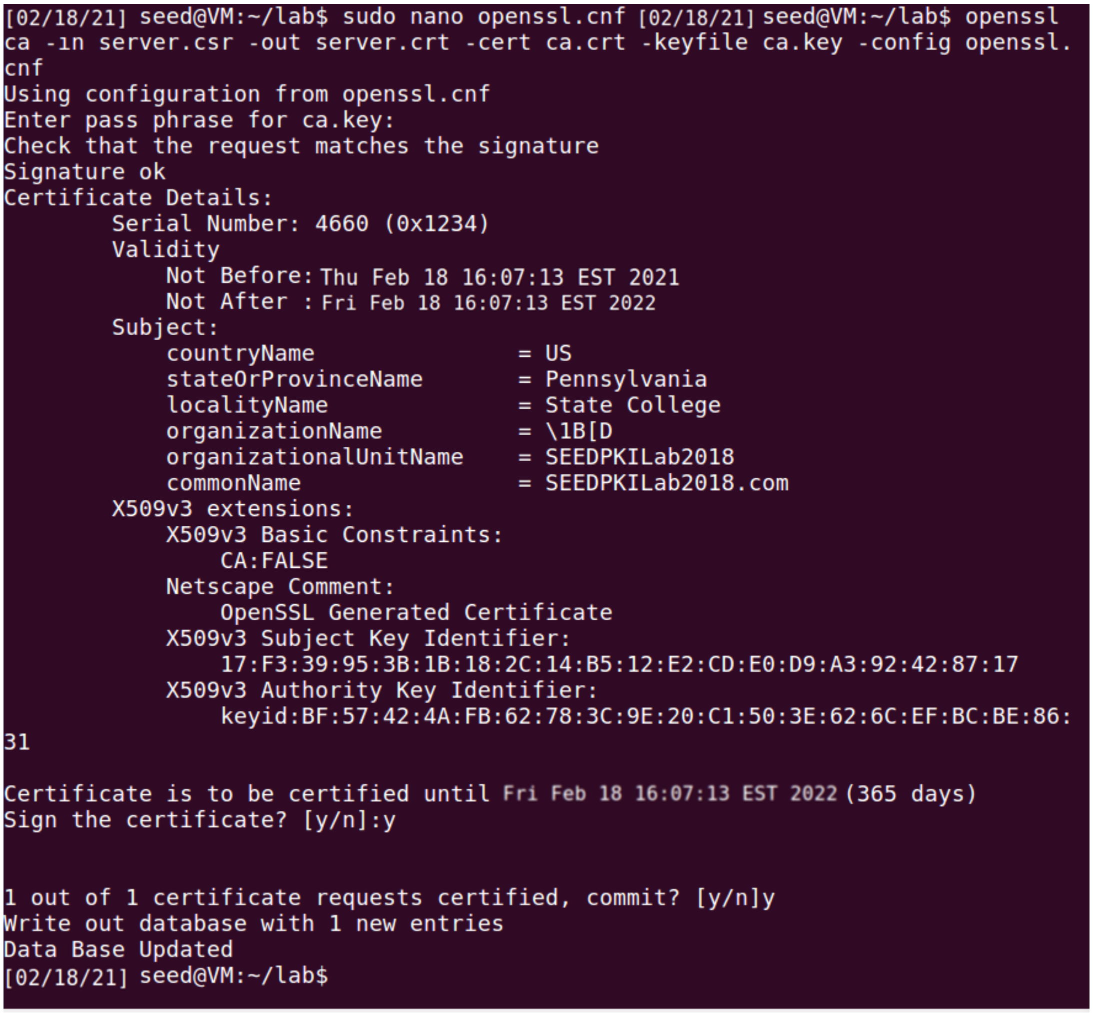
Observation: The certificate signing request is generated using server key which generates the certificate for the key

Step 3



```
  GNU nano 2.5.3                 File: openssl.cnf

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = default               # use public key default MD
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_anything

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]

^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line
```

Changing the policy in openssl.cnf

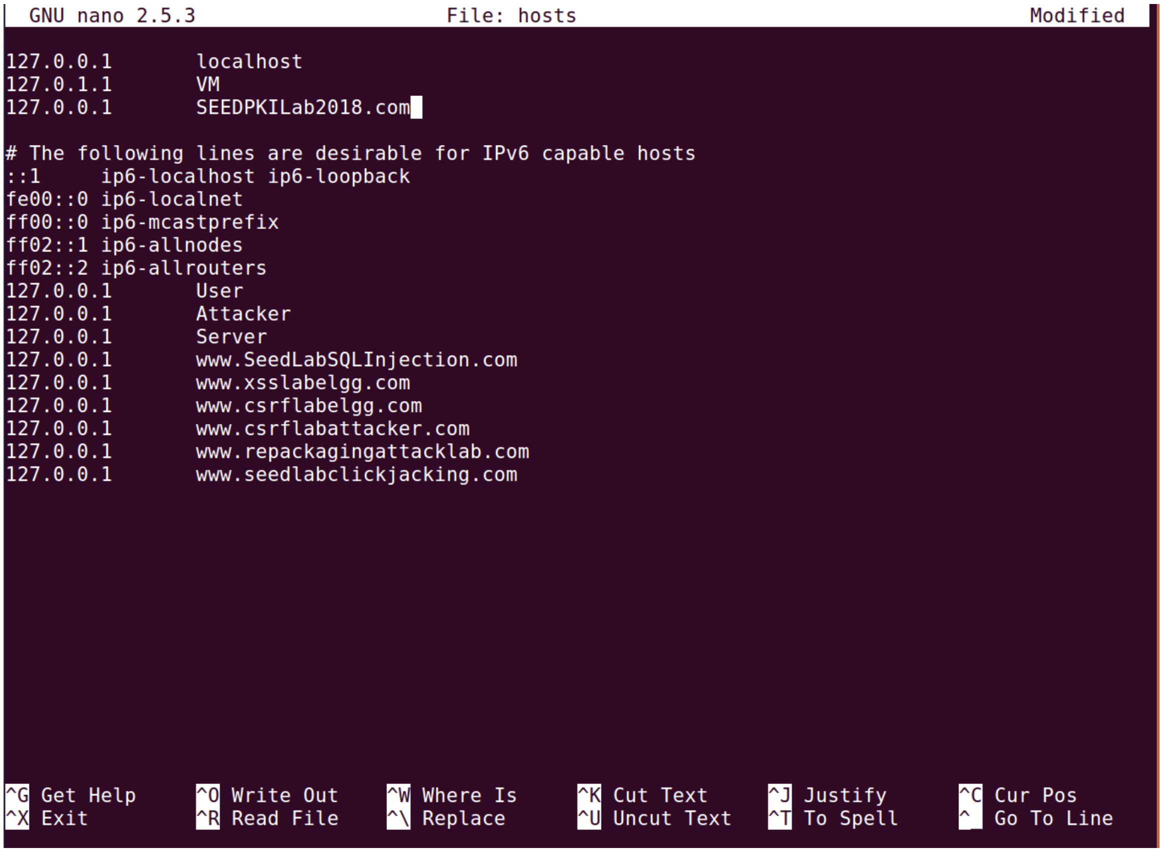


```
[02/18/21] seed@VM:~/lab$ sudo nano openssl.cnf [02/18/21] seed@VM:~/lab$ openssl
ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.
cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4660 (0x1234)
        Validity
            Not Before: Thu Feb 18 16:07:13 EST 2021
            Not After : Fri Feb 18 16:07:13 EST 2022
        Subject:
            countryName               = US
            stateOrProvinceName       = Pennsylvania
            localityName              = State College
            organizationName          = \1B[D
            organizationalUnitName    = SEEDPKILab2018
            commonName                = SEEDPKILab2018.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                17:F3:39:95:3B:1B:18:2C:14:B5:12:E2:CD:E0:D9:A3:92:42:87:17
            X509v3 Authority Key Identifier:
                keyid:BF:57:42:4A:FB:62:78:3C:9E:20:C1:50:3E:62:6C:EF:BC:BE:86:
31

Certificate is to be certified until Fri Feb 18 16:07:13 EST 2022 (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[02/18/21] seed@VM:~/lab$
```

Observation: CSR file is sent to the CA and it is signed using cs.crt and ca.key to generate certificate for our created site, SEEDPKILab2018.com

Task 3

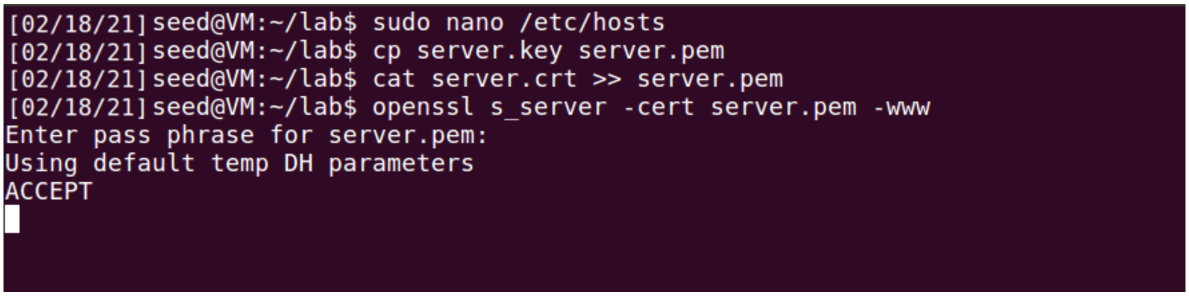
Step 1



```
 GNU nano 2.5.3                    File: hosts                              Modified

127.0.0.1       localhost
127.0.1.1       VM
127.0.0.1       SEEDPKILab2018.com

# 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       User
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
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com









^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line
```

Observation: we map SEEDPKILab2018.com to the local host so it can be used as a local domain name

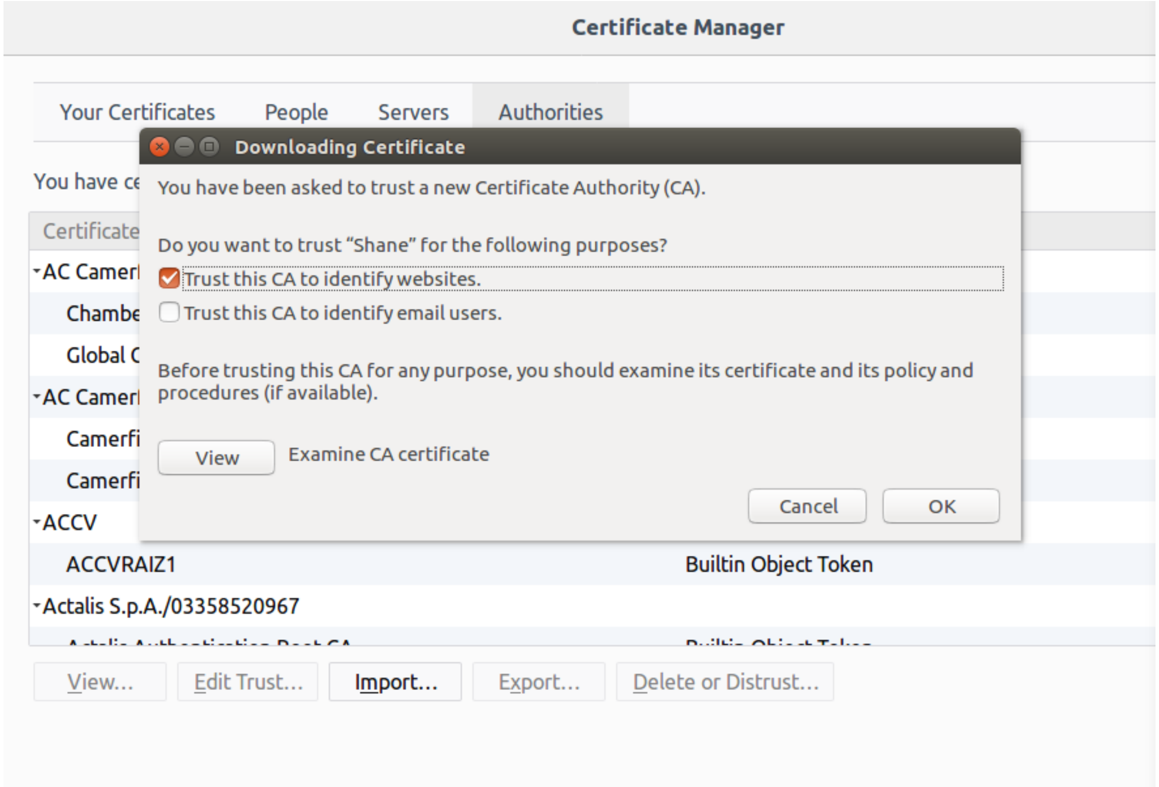
Step 2



```
[02/18/21]seed@VM:~/lab$ sudo nano /etc/hosts
[02/18/21]seed@VM:~/lab$ cp server.key server.pem
[02/18/21]seed@VM:~/lab$ cat server.crt >> server.pem
[02/18/21]seed@VM:~/lab$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

Observations: After we map the site to the local host we can launch it using the server.pem file, this joins with server.key and server.crt.

Step 3



You have been asked to trust a new Certificate Authority (CA).

Do you want to trust “Shane” for the following purposes?

☑ Trust this CA to identify websites.

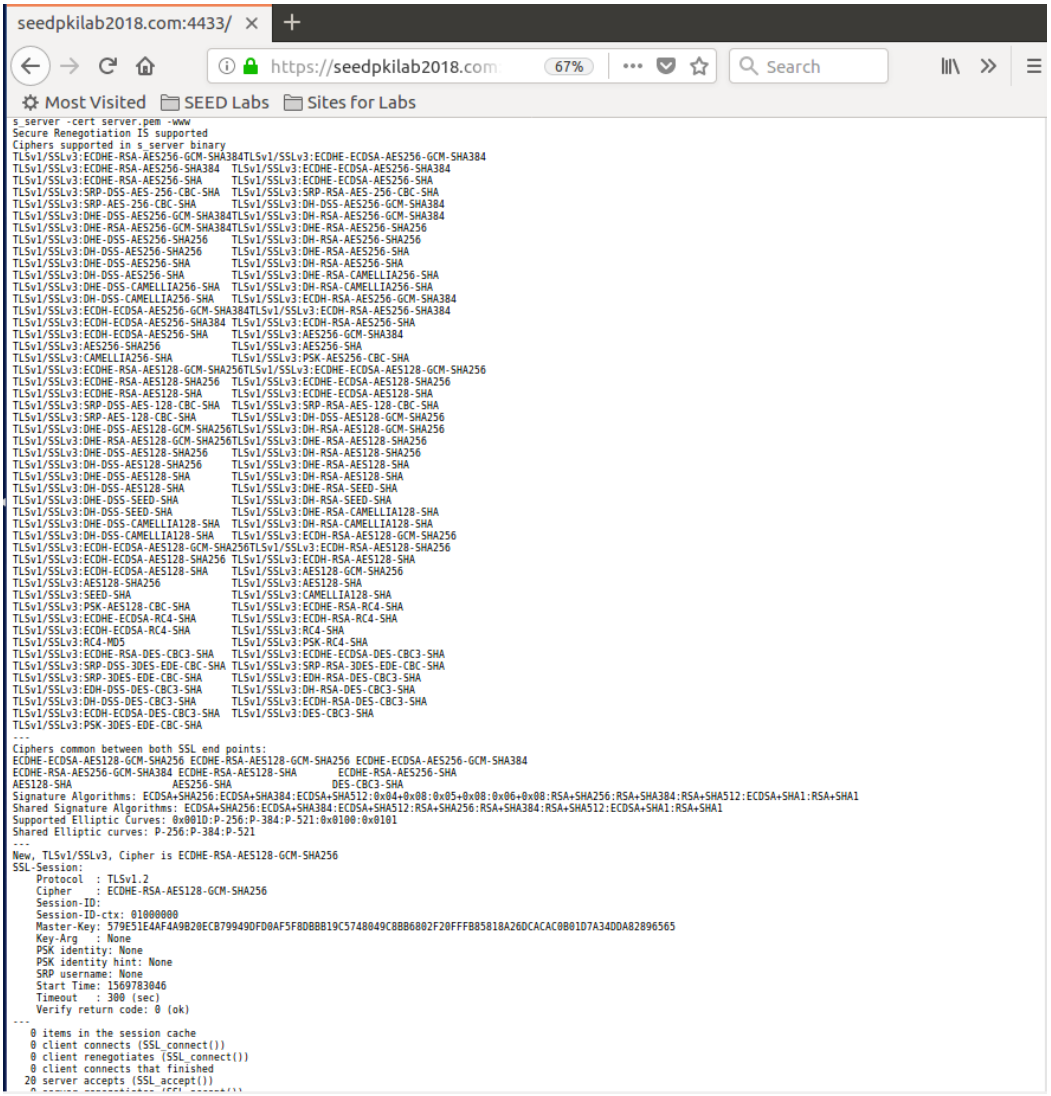
☐ Trust this CA to identify email users.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

Examine CA certificate

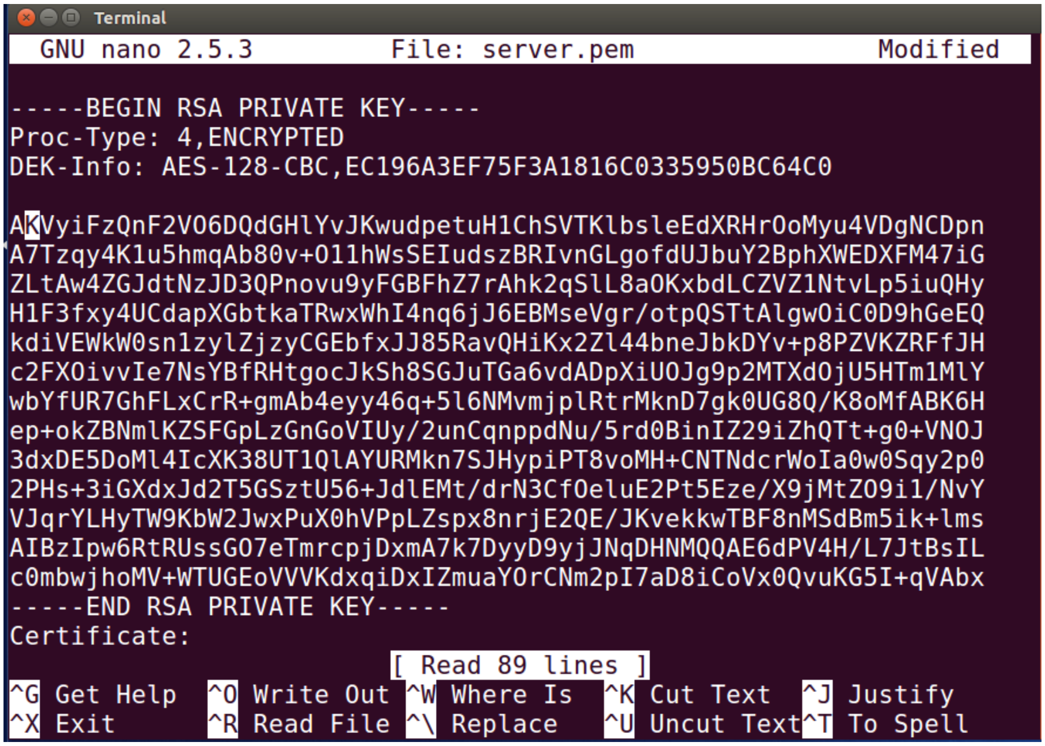
Observation: The site is at first untrusted, we must download certificate and allow it to trust the site in order to access it, since it is not designed or hosted by a trusted company.

Step 4

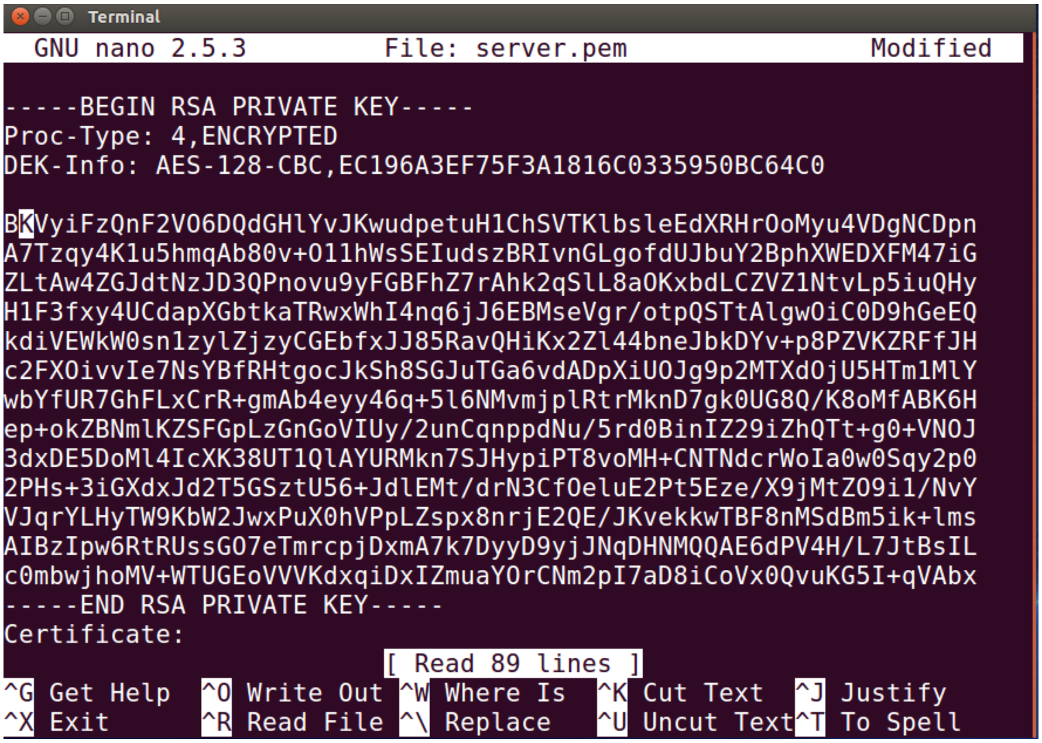


```
s_server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384  TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA     TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA
TLSv1/SSLv3:SRP-DSS-AES-256-CBC-SHA  TLSv1/SSLv3:SRP-RSA-AES-256-CBC-SHA
TLSv1/SSLv3:SRP-AES-256-CBC-SHA      TLSv1/SSLv3:DH-DSS-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-DSS-AES256-GCM-SHA384TLSv1/SSLv3:DH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:DHE-RSA-AES256-SHA256
TLSv1/SSLv3:DHE-DSS-AES256-SHA256    TLSv1/SSLv3:DH-RSA-AES256-SHA256
TLSv1/SSLv3:DH-DSS-AES256-SHA256     TLSv1/SSLv3:DHE-RSA-AES256-SHA
TLSv1/SSLv3:DHE-DSS-AES256-SHA       TLSv1/SSLv3:DH-RSA-AES256-SHA
TLSv1/SSLv3:DH-DSS-AES256-SHA        TLSv1/SSLv3:DHE-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA256-SHA  TLSv1/SSLv3:DH-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA256-SHA   TLSv1/SSLv3:ECDH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDH-RSA-AES256-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA384 TLSv1/SSLv3:ECDH-RSA-AES256-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA    TLSv1/SSLv3:AES256-GCM-SHA384
TLSv1/SSLv3:AES256-SHA256            TLSv1/SSLv3:AES256-SHA
TLSv1/SSLv3:CAMELLIA256-SHA          TLSv1/SSLv3:PSK-AES256-CBC-SHA
TLSv1/SSLv3:ECDHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA256  TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA     TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA
TLSv1/SSLv3:SRP-DSS-AES-128-CBC-SHA  TLSv1/SSLv3:SRP-RSA-AES-128-CBC-SHA
TLSv1/SSLv3:SRP-AES-128-CBC-SHA      TLSv1/SSLv3:DH-DSS-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-GCM-SHA256TLSv1/SSLv3:DH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:DHE-RSA-AES128-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-SHA256    TLSv1/SSLv3:DH-RSA-AES128-SHA256
TLSv1/SSLv3:DH-DSS-AES128-SHA256     TLSv1/SSLv3:DHE-RSA-AES128-SHA
TLSv1/SSLv3:DHE-DSS-AES128-SHA       TLSv1/SSLv3:DH-RSA-AES128-SHA
TLSv1/SSLv3:DH-DSS-AES128-SHA        TLSv1/SSLv3:DHE-RSA-SEED-SHA
TLSv1/SSLv3:DHE-DSS-SEED-SHA         TLSv1/SSLv3:DH-RSA-SEED-SHA
TLSv1/SSLv3:DH-DSS-SEED-SHA          TLSv1/SSLv3:DHE-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA128-SHA  TLSv1/SSLv3:DH-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA128-SHA   TLSv1/SSLv3:ECDH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDH-RSA-AES128-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA256 TLSv1/SSLv3:ECDH-RSA-AES128-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA    TLSv1/SSLv3:AES128-GCM-SHA256
TLSv1/SSLv3:AES128-SHA256            TLSv1/SSLv3:AES128-SHA
TLSv1/SSLv3:SEED-SHA                 TLSv1/SSLv3:CAMELLIA128-SHA
TLSv1/SSLv3:PSK-AES128-CBC-SHA       TLSv1/SSLv3:ECDHE-RSA-RC4-SHA
TLSv1/SSLv3:ECDHE-ECDSA-RC4-SHA      TLSv1/SSLv3:ECDH-RSA-RC4-SHA
TLSv1/SSLv3:ECDH-ECDSA-RC4-SHA       TLSv1/SSLv3:RC4-SHA
TLSv1/SSLv3:RC4-MD5                  TLSv1/SSLv3:PSK-RC4-SHA
TLSv1/SSLv3:ECDHE-RSA-DES-CBC3-SHA   TLSv1/SSLv3:ECDHE-ECDSA-DES-CBC3-SHA
TLSv1/SSLv3:SRP-DSS-3DES-EDE-CBC-SHA TLSv1/SSLv3:SRP-RSA-3DES-EDE-CBC-SHA
TLSv1/SSLv3:SRP-3DES-EDE-CBC-SHA     TLSv1/SSLv3:EDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:EDH-DSS-DES-CBC3-SHA     TLSv1/SSLv3:DH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:DH-DSS-DES-CBC3-SHA      TLSv1/SSLv3:ECDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:ECDH-ECDSA-DES-CBC3-SHA  TLSv1/SSLv3:DES-CBC3-SHA
TLSv1/SSLv3:PSK-3DES-EDE-CBC-SHA
---
Ciphers common between both SSL end points:
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA       ECDHE-RSA-AES256-SHA
AES128-SHA                 AES256-SHA                 DES-CBC3-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:0x04+0x08:0x05+0x08:0x06+0x08:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Supported Elliptic Curves: 0x001D:P-256:P-384:P-521:0x0100:0x0101
Shared Elliptic curves: P-256:P-384:P-521
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID: 
    Session-ID-ctx: 01000000
    Master-Key: 579E51E4AF4A9B20ECB79949DFD0AF5F8DBBB19C5748049C8BB6802F20FFFB85818A26DCACAC0B01D7A34DDA82896565
    Key-Arg   : None
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1569783046
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
   0 items in the session cache
   0 client connects (SSL_connect())
   0 client renegotiates (SSL_connect())
   0 client connects that finished
  20 server accepts (SSL_accept())
```

Observation: This is how the site first appears, after being accepted through certificates. It is trusted and certified

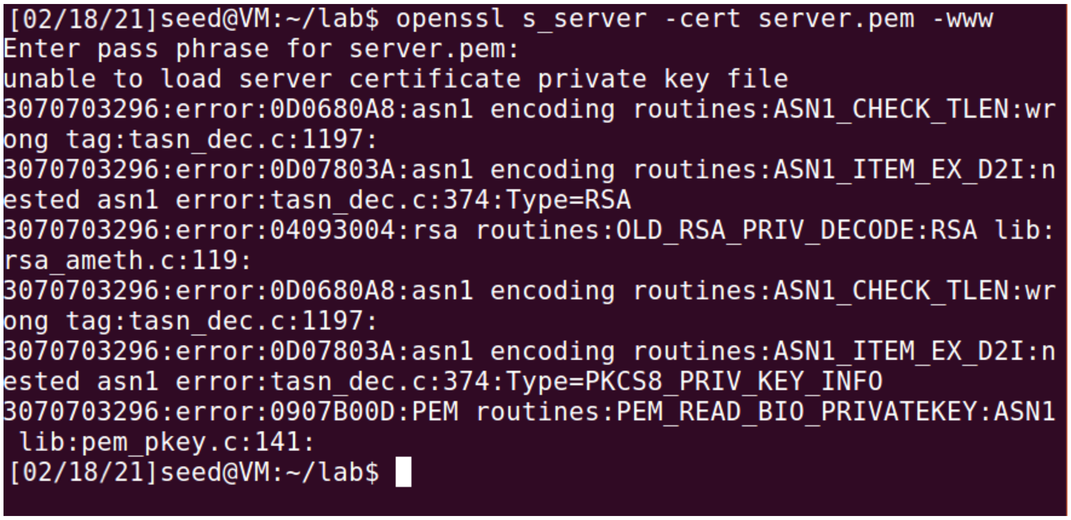


```
Terminal
  GNU nano 2.5.3            File: server.pem                       Modified

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,EC196A3EF75F3A1816C0335950BC64C0

AKVyiFzQnF2VO6DQdGHlYvJKwudpetuH1ChSVTKlbsleEdXRHrOoMyu4VDgNCDpn
A7Tzqy4K1u5hmqAb80v+O11hWsSEIudszBRIvnGLgofdUJbuY2BphXWEDXFM47iG
ZLtAw4ZGJdtNzJD3QPnovu9yFGBFhZ7rAhk2qSlL8aOKxbdLCZVZ1NtvLp5iuQHy
H1F3fxy4UCdapXGbtkaTRwxWhI4nq6jJ6EBMseVgr/otpQSTtAlgwOiC0D9hGeEQ
kdiVEWkW0sn1zylZjzyCGEbfxJJ85RavQHiKx2Zl44bneJbkDYv+p8PZVKZRFfJH
c2FXOivvIe7NsYBfRHtgocJkSh8SGJuTGa6vdADpXiUOJg9p2MTXdOjU5HTm1MlY
wbYfUR7GhFLxCrR+gmAb4eyy46q+5l6NMvmjplRtrMknD7gk0UG8Q/K8oMfABK6H
ep+okZBNmlKZSFGpLzGnGoVIUy/2unCqnppdNu/5rd0BinIZ29iZhQTt+g0+VNOJ
3dxDE5DoMl4IcXK38UT1QlAYURMkn7SJHypiPT8voMH+CNTNdcrWoIa0w0Sqy2p0
2PHs+3iGXdxJd2T5GSztU56+JdlEMt/drN3CfOeluE2Pt5Eze/X9jMtZO9i1/NvY
VJqrYLHyTW9KbW2JwxPuX0hVPpLZspx8nrjE2QE/JKvekkwTBF8nMSdBm5ik+lms
AIBzIpw6RtRUssGO7eTmrcpjDxmA7k7DyyD9yjJNqDHNMQQAE6dPV4H/L7JtBsIL
c0mbwjhoMV+WTUGEoVVVKdxqiDxIZmuaYOrCNm2pI7aD8iCoVx0QvuKG5I+qVAbx
-----END RSA PRIVATE KEY-----
Certificate:
                              [ Read 89 lines ]
^G Get Help    ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify
^X Exit        ^R Read File ^\ Replace    ^U Uncut Text^T To Spell
```



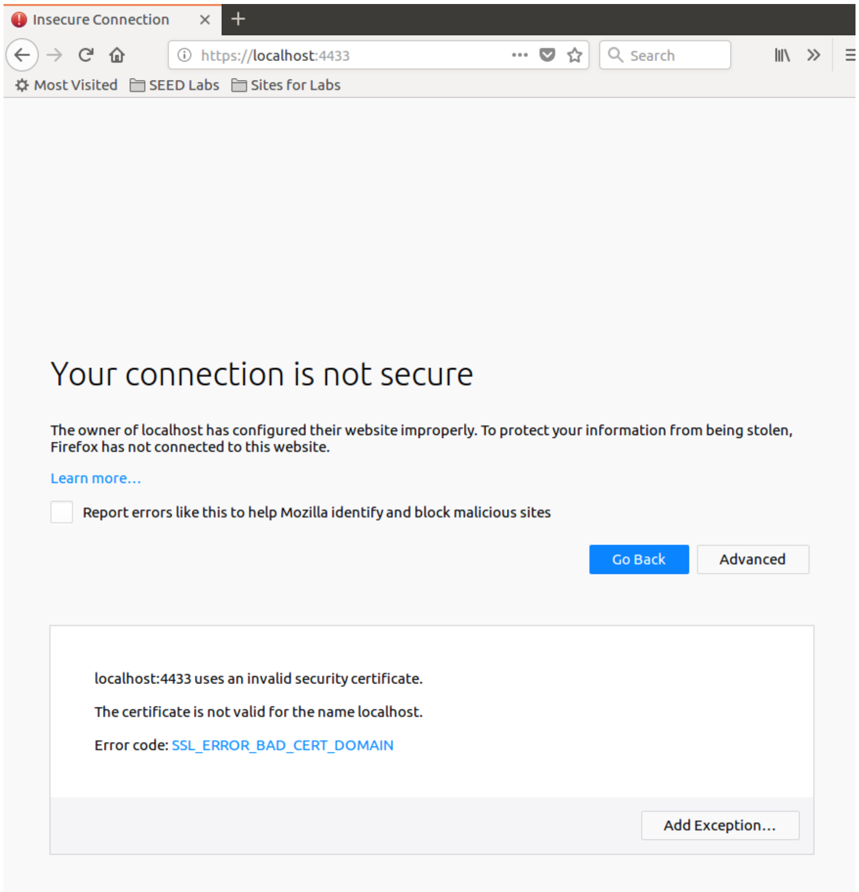
```
Terminal
  GNU nano 2.5.3            File: server.pem                       Modified

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,EC196A3EF75F3A1816C0335950BC64C0

BKVyiFzQnF2VO6DQdGHlYvJKwudpetuH1ChSVTKlbsleEdXRHrOoMyu4VDgNCDpn
A7Tzqy4K1u5hmqAb80v+O11hWsSEIudszBRIvnGLgofdUJbuY2BphXWEDXFM47iG
ZLtAw4ZGJdtNzJD3QPnovu9yFGBFhZ7rAhk2qSlL8aOKxbdLCZVZ1NtvLp5iuQHy
H1F3fxy4UCdapXGbtkaTRwxWhI4nq6jJ6EBMseVgr/otpQSTtAlgwOiC0D9hGeEQ
kdiVEWkW0sn1zylZjzyCGEbfxJJ85RavQHiKx2Zl44bneJbkDYv+p8PZVKZRFfJH
c2FXOivvIe7NsYBfRHtgocJkSh8SGJuTGa6vdADpXiUOJg9p2MTXdOjU5HTm1MlY
wbYfUR7GhFLxCrR+gmAb4eyy46q+5l6NMvmjplRtrMknD7gk0UG8Q/K8oMfABK6H
ep+okZBNmlKZSFGpLzGnGoVIUy/2unCqnppdNu/5rd0BinIZ29iZhQTt+g0+VNOJ
3dxDE5DoMl4IcXK38UT1QlAYURMkn7SJHypiPT8voMH+CNTNdcrWoIa0w0Sqy2p0
2PHs+3iGXdxJd2T5GSztU56+JdlEMt/drN3CfOeluE2Pt5Eze/X9jMtZO9i1/NvY
VJqrYLHyTW9KbW2JwxPuX0hVPpLZspx8nrjE2QE/JKvekkwTBF8nMSdBm5ik+lms
AIBzIpw6RtRUssGO7eTmrcpjDxmA7k7DyyD9yjJNqDHNMQQAE6dPV4H/L7JtBsIL
c0mbwjhoMV+WTUGEoVVVKdxqiDxIZmuaYOrCNm2pI7aD8iCoVx0QvuKG5I+qVAbx
-----END RSA PRIVATE KEY-----
Certificate:
                              [ Read 89 lines ]
^G Get Help    ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify
^X Exit        ^R Read File ^\ Replace    ^U Uncut Text^T To Spell
```

Observation: Changing the first bit of “A” to a “B”



```
[02/18/21]seed@VM:~/lab$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
unable to load server certificate private key file
3070703296:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wr
ong tag:tasn_dec.c:1197:
3070703296:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:n
ested asn1 error:tasn_dec.c:374:Type=RSA
3070703296:error:04093004:rsa routines:OLD_RSA_PRIV_DECODE:RSA lib:
rsa_ameth.c:119:
3070703296:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wr
ong tag:tasn_dec.c:1197:
3070703296:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:n
ested asn1 error:tasn_dec.c:374:Type=PKCS8_PRIV_KEY_INFO
3070703296:error:0907B00D:PEM routines:PEM_READ_BIO_PRIVATEKEY:ASN1
 lib:pem_pkey.c:141:
[02/18/21]seed@VM:~/lab$ 
```

Observation: Changing the byte makes the terminal unable to load the server certificate key file.



Your connection is not secure

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

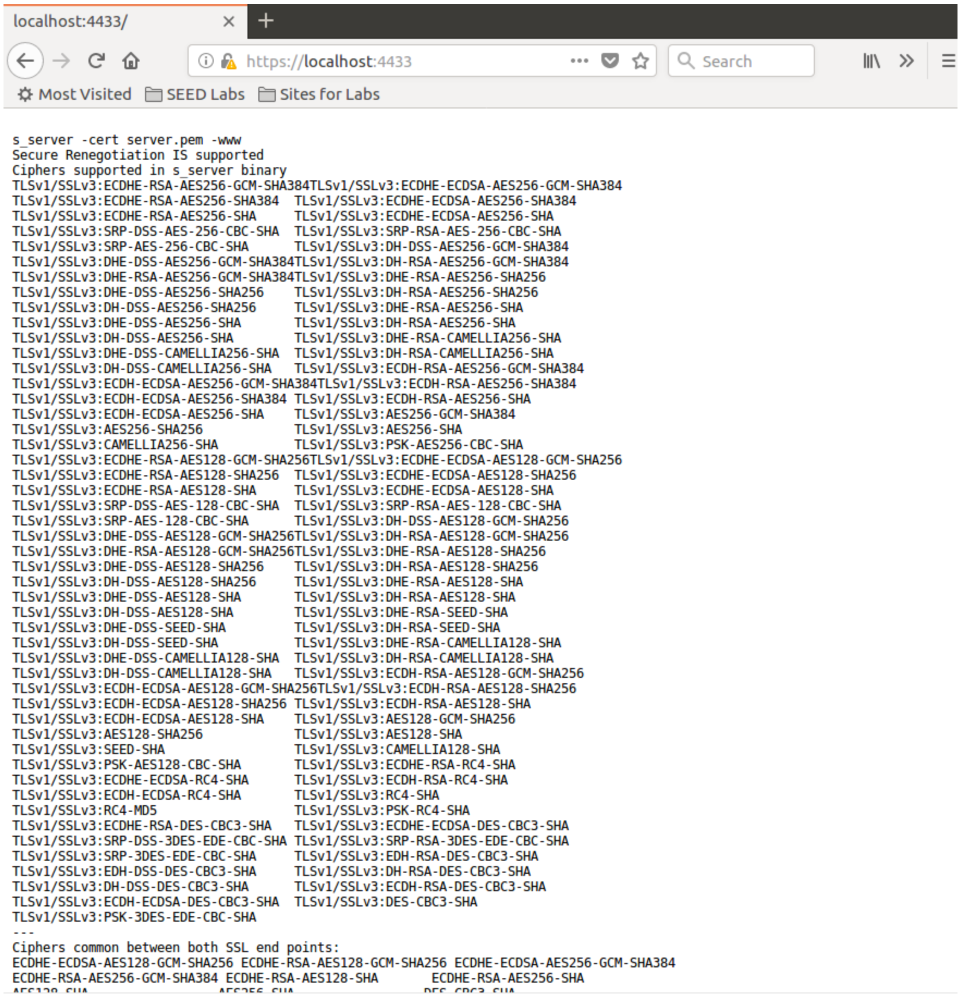
Learn more…

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

localhost:4433 uses an invalid security certificate.

The certificate is not valid for the name localhost.

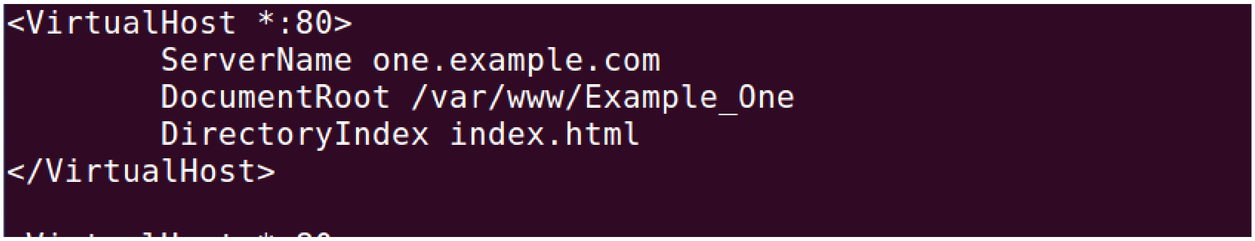
Error code: SSL_ERROR_BAD_CERT_DOMAIN



```
s_server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384  TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA     TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA
TLSv1/SSLv3:SRP-DSS-AES-256-CBC-SHA  TLSv1/SSLv3:SRP-RSA-AES-256-CBC-SHA
TLSv1/SSLv3:SRP-AES-256-CBC-SHA      TLSv1/SSLv3:DH-DSS-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-DSS-AES256-GCM-SHA384TLSv1/SSLv3:DH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:DHE-RSA-AES256-SHA256
TLSv1/SSLv3:DHE-DSS-AES256-SHA256    TLSv1/SSLv3:DH-RSA-AES256-SHA256
TLSv1/SSLv3:DH-DSS-AES256-SHA256     TLSv1/SSLv3:DHE-RSA-AES256-SHA
TLSv1/SSLv3:DHE-DSS-AES256-SHA       TLSv1/SSLv3:DH-RSA-AES256-SHA
TLSv1/SSLv3:DH-DSS-AES256-SHA        TLSv1/SSLv3:DHE-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA256-SHA  TLSv1/SSLv3:DH-RSA-CAMELLIA256-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA256-SHA   TLSv1/SSLv3:ECDH-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDH-RSA-AES256-SHA384
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA384 TLSv1/SSLv3:ECDH-RSA-AES256-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA    TLSv1/SSLv3:AES256-GCM-SHA384
TLSv1/SSLv3:AES256-SHA256            TLSv1/SSLv3:AES256-SHA
TLSv1/SSLv3:CAMELLIA256-SHA          TLSv1/SSLv3:PSK-AES256-CBC-SHA
TLSv1/SSLv3:ECDHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA256  TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA256
TLSv1/SSLv3:ECDHE-RSA-AES128-SHA     TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA
TLSv1/SSLv3:SRP-DSS-AES-128-CBC-SHA  TLSv1/SSLv3:SRP-RSA-AES-128-CBC-SHA
TLSv1/SSLv3:SRP-AES-128-CBC-SHA      TLSv1/SSLv3:DH-DSS-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-GCM-SHA256TLSv1/SSLv3:DH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:DHE-RSA-AES128-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-SHA256    TLSv1/SSLv3:DH-RSA-AES128-SHA256
TLSv1/SSLv3:DH-DSS-AES128-SHA256     TLSv1/SSLv3:DHE-RSA-AES128-SHA
TLSv1/SSLv3:DHE-DSS-AES128-SHA       TLSv1/SSLv3:DH-RSA-AES128-SHA
TLSv1/SSLv3:DH-DSS-AES128-SHA        TLSv1/SSLv3:DHE-RSA-SEED-SHA
TLSv1/SSLv3:DHE-DSS-SEED-SHA         TLSv1/SSLv3:DH-RSA-SEED-SHA
TLSv1/SSLv3:DH-DSS-SEED-SHA          TLSv1/SSLv3:DHE-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA128-SHA  TLSv1/SSLv3:DH-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DH-DSS-CAMELLIA128-SHA   TLSv1/SSLv3:ECDH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDH-RSA-AES128-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA256 TLSv1/SSLv3:ECDH-RSA-AES128-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA    TLSv1/SSLv3:AES128-GCM-SHA256
TLSv1/SSLv3:AES128-SHA256            TLSv1/SSLv3:AES128-SHA
TLSv1/SSLv3:SEED-SHA                 TLSv1/SSLv3:CAMELLIA128-SHA
TLSv1/SSLv3:PSK-AES128-CBC-SHA       TLSv1/SSLv3:ECDHE-RSA-RC4-SHA
TLSv1/SSLv3:ECDHE-ECDSA-RC4-SHA      TLSv1/SSLv3:ECDH-RSA-RC4-SHA
TLSv1/SSLv3:ECDH-ECDSA-RC4-SHA       TLSv1/SSLv3:RC4-SHA
TLSv1/SSLv3:RC4-MD5                  TLSv1/SSLv3:PSK-RC4-SHA
TLSv1/SSLv3:ECDHE-RSA-DES-CBC3-SHA   TLSv1/SSLv3:ECDHE-ECDSA-DES-CBC3-SHA
TLSv1/SSLv3:SRP-DSS-3DES-EDE-CBC-SHA TLSv1/SSLv3:SRP-RSA-3DES-EDE-CBC-SHA
TLSv1/SSLv3:SRP-3DES-EDE-CBC-SHA     TLSv1/SSLv3:EDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:EDH-DSS-DES-CBC3-SHA     TLSv1/SSLv3:DH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:DH-DSS-DES-CBC3-SHA      TLSv1/SSLv3:ECDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:ECDH-ECDSA-DES-CBC3-SHA  TLSv1/SSLv3:DES-CBC3-SHA
TLSv1/SSLv3:PSK-3DES-EDE-CBC-SHA
---
Ciphers common between both SSL end points:
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA        ECDHE-RSA-AES256-SHA
AES128-SHA                  AES256-SHA                  DES-CBC3-SHA
```

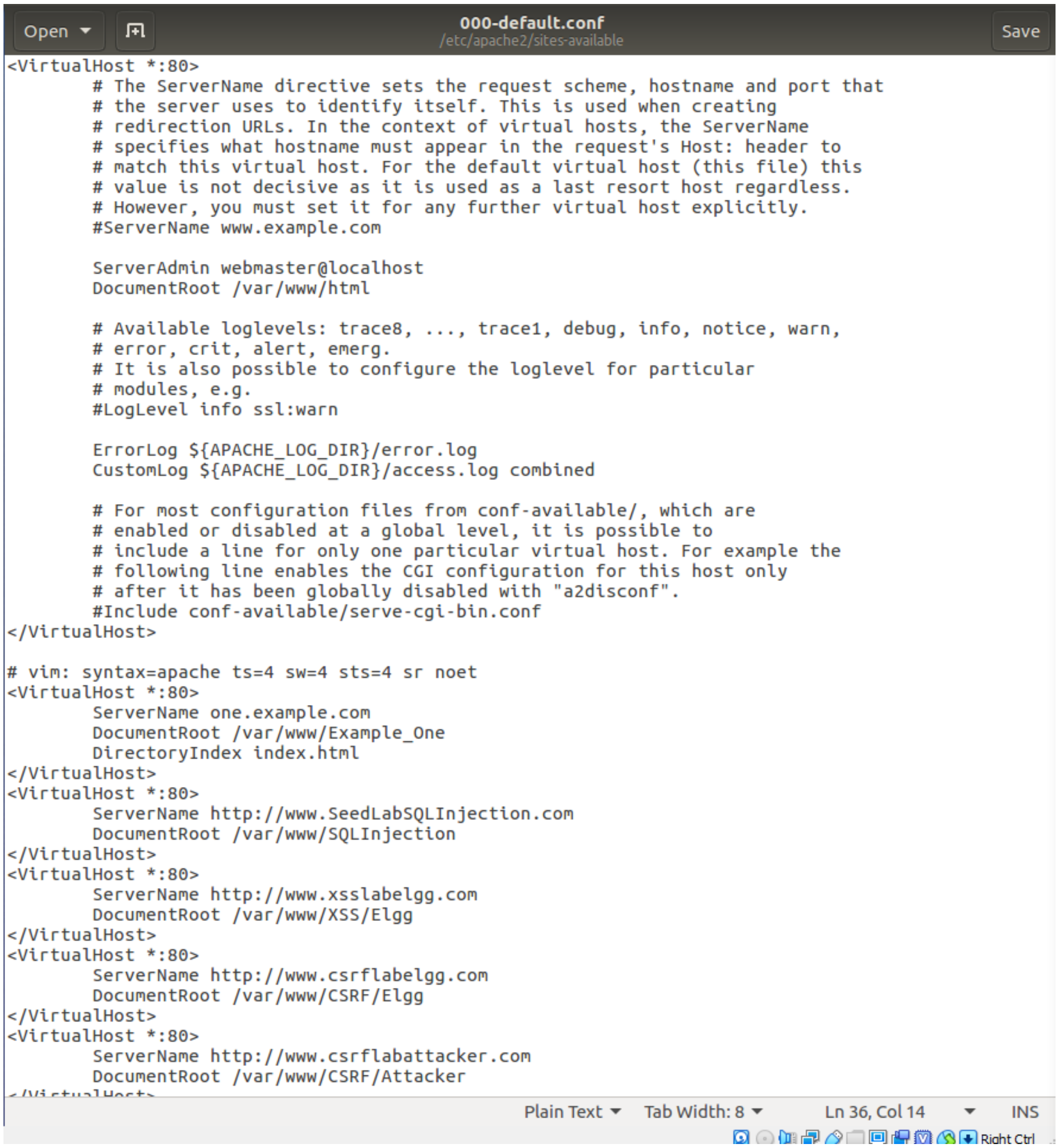
Observation: changing from the url to https://localhost:4433 brought us through the same process, getting to the site and having to go through the certificate approval process again. As shown, once the process is repeated and we approve the certificate, we can access the same exact site as if we typed in the SEEDPKILab2018.com

Task 4



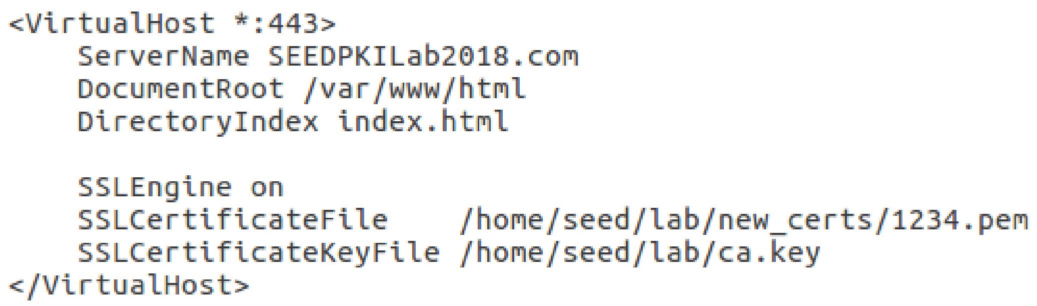
```
<VirtualHost *:80>
        ServerName one.example.com
        DocumentRoot /var/www/Example_One
        DirectoryIndex index.html
</VirtualHost>
```

Observation: just doing the first example give, getting a feel for the format and how to do so.



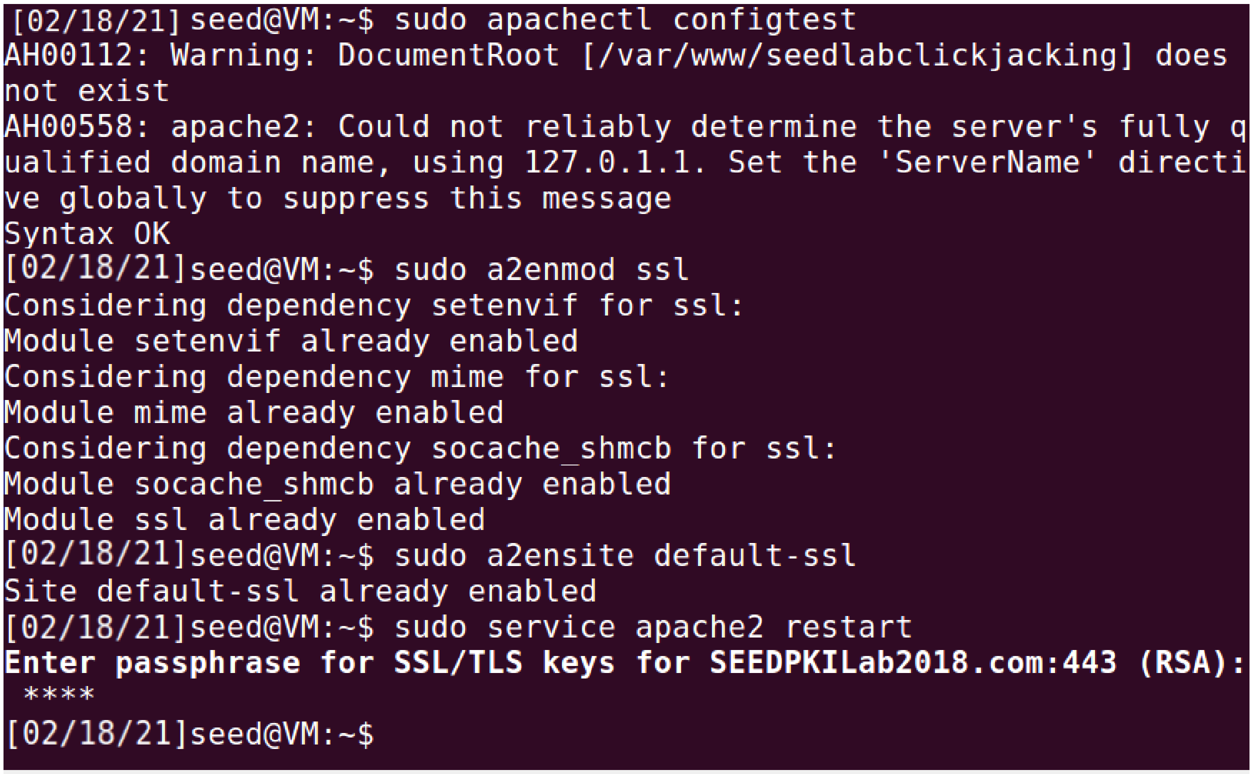
```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<VirtualHost *:80>
        ServerName one.example.com
        DocumentRoot /var/www/Example_One
        DirectoryIndex index.html
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.SeedLabSQLInjection.com
        DocumentRoot /var/www/SQLInjection
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.xsslabelgg.com
        DocumentRoot /var/www/XSS/Elgg
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabelgg.com
        DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabattacker.com
        DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
```

Observation: once again just doing the second part of the pdf example to see how it is done.



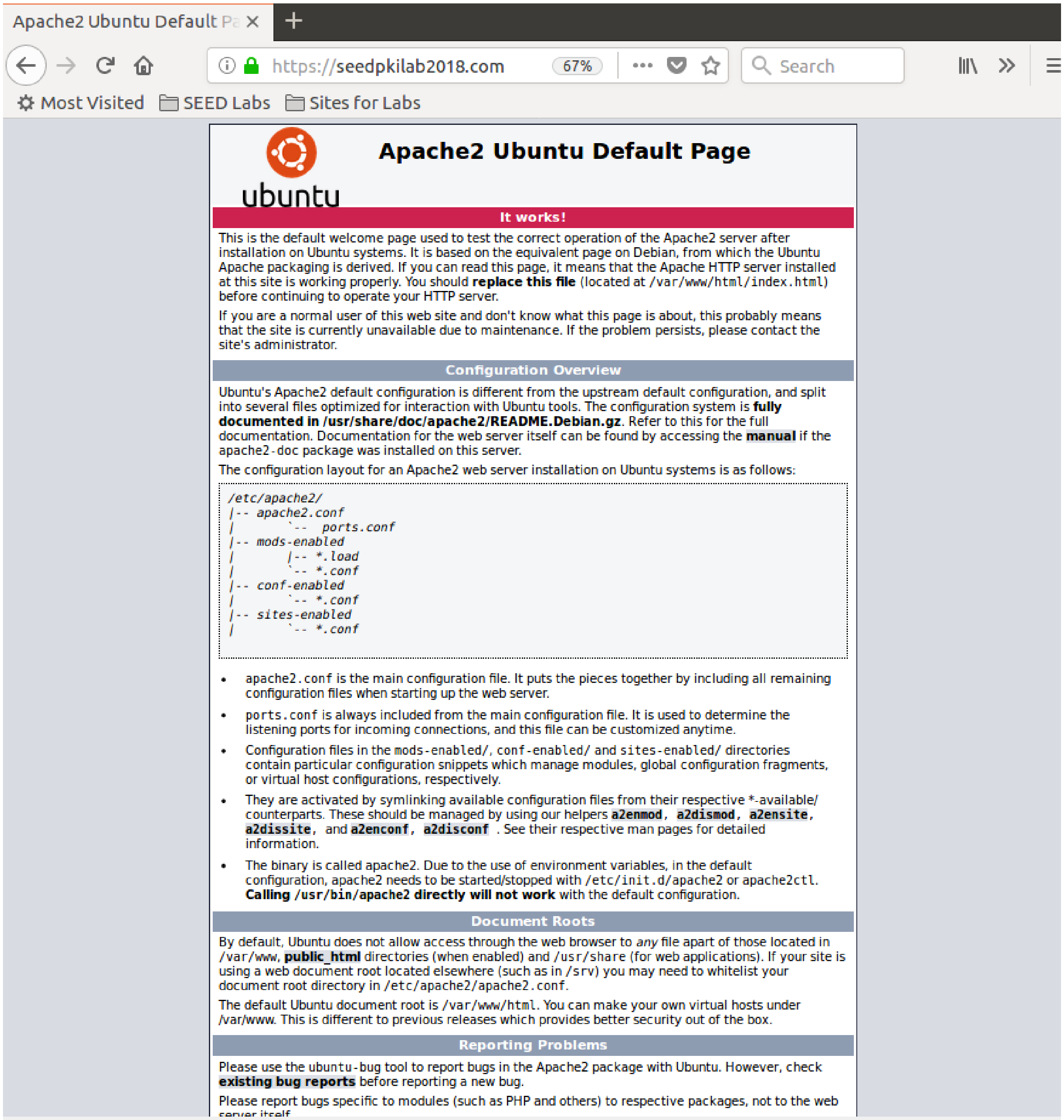
```
<VirtualHost *:443>
    ServerName SEEDPKILab2018.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html

    SSLEngine on
    SSLCertificateFile    /home/seed/lab/new_certs/1234.pem
    SSLCertificateKeyFile /home/seed/lab/ca.key
</VirtualHost>
```

Observation: adding the SEEDPKILab2018 site, using our created keys and the roots and data from the examples. Creating our own HTTPS Server.



```
[02/18/21] seed@VM:~$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does
not exist
AH00558: apache2: Could not reliably determine the server's fully q
ualified domain name, using 127.0.1.1. Set the 'ServerName' directi
ve globally to suppress this message
Syntax OK
[02/18/21]seed@VM:~$ 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
[02/18/21]seed@VM:~$ sudo a2ensite default-ssl
Site default-ssl already enabled
[02/18/21]seed@VM:~$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2018.com:443 (RSA):
 ****
[02/18/21]seed@VM:~$
```

Observation: After running the given commands in the pdf, appears it has worked, and we need to enter our password from earlier that we set up for the site / certificate.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **`a2enmod`**, **`a2dismod`**, **`a2ensite`**, **`a2dissite`**, and **`a2enconf`**, **`a2disconf`** . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

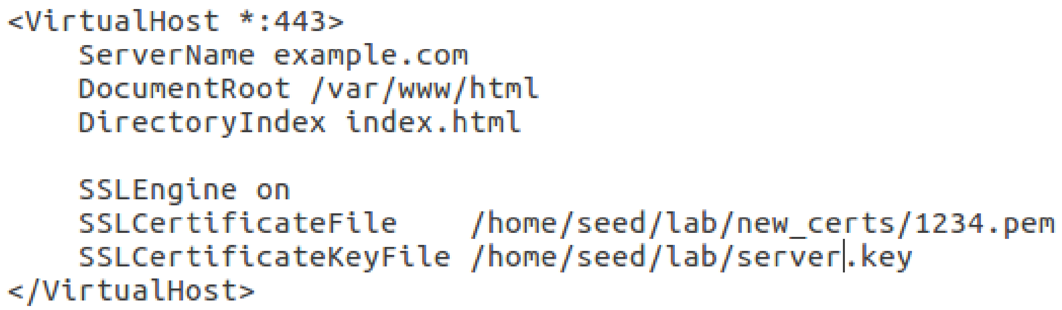
Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself

Observation: It appears the site worked, setting up was successful and the site brings us to the default HTML site set up within Ubuntu and in the apache folder. Setting up the site appears to be successful.

Task 5

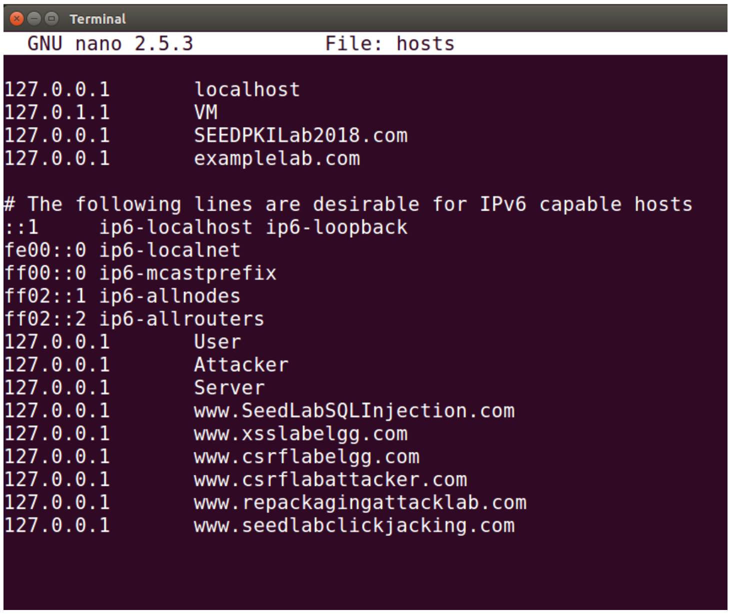
Step 1



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

    SSLEngine on
    SSLCertificateFile    /home/seed/lab/new_certs/1234.pem
    SSLCertificateKeyFile /home/seed/lab/server.key
</VirtualHost>
```

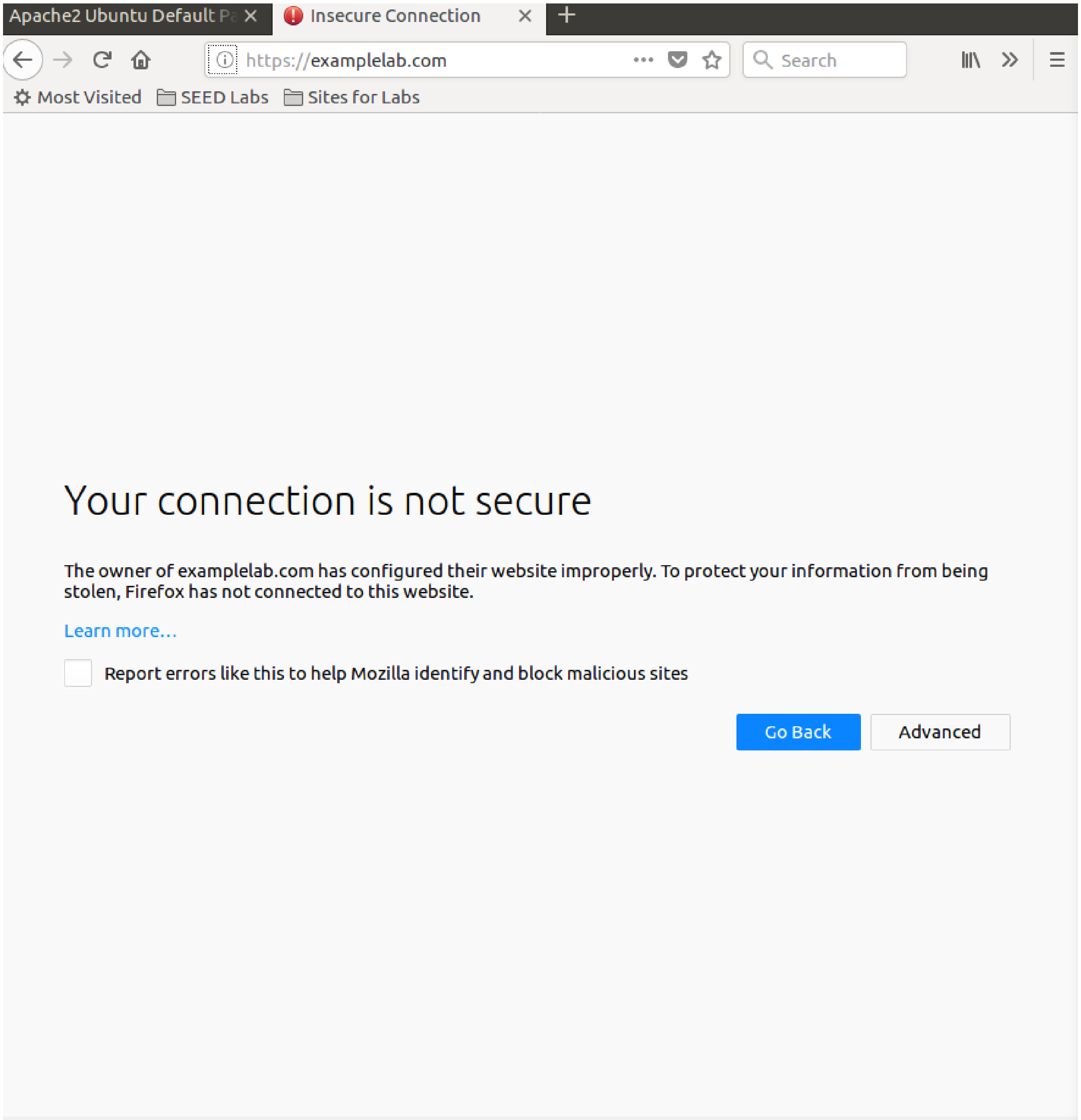
Observation: setting up the malicious site. Similar procedure to our previous step with our good website, but this time we change the server name (!!! HAD TO CHANGE SERVERNAME TO "EXAMPLELAB.COM" BECAUSE EXAMPLE.COM TOOK ME TO A REAL SITE !!!)

Step 2



```
  GNU nano 2.5.3            File: hosts

127.0.0.1       localhost
127.0.1.1       VM
127.0.0.1       SEEDPKILab2018.com
127.0.0.1       examplelab.com

# 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       User
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
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com
```

Observation: adding the examplelab.com (malicious site) to the hosts file, changing it to our host machine and becoming the man in the middle.



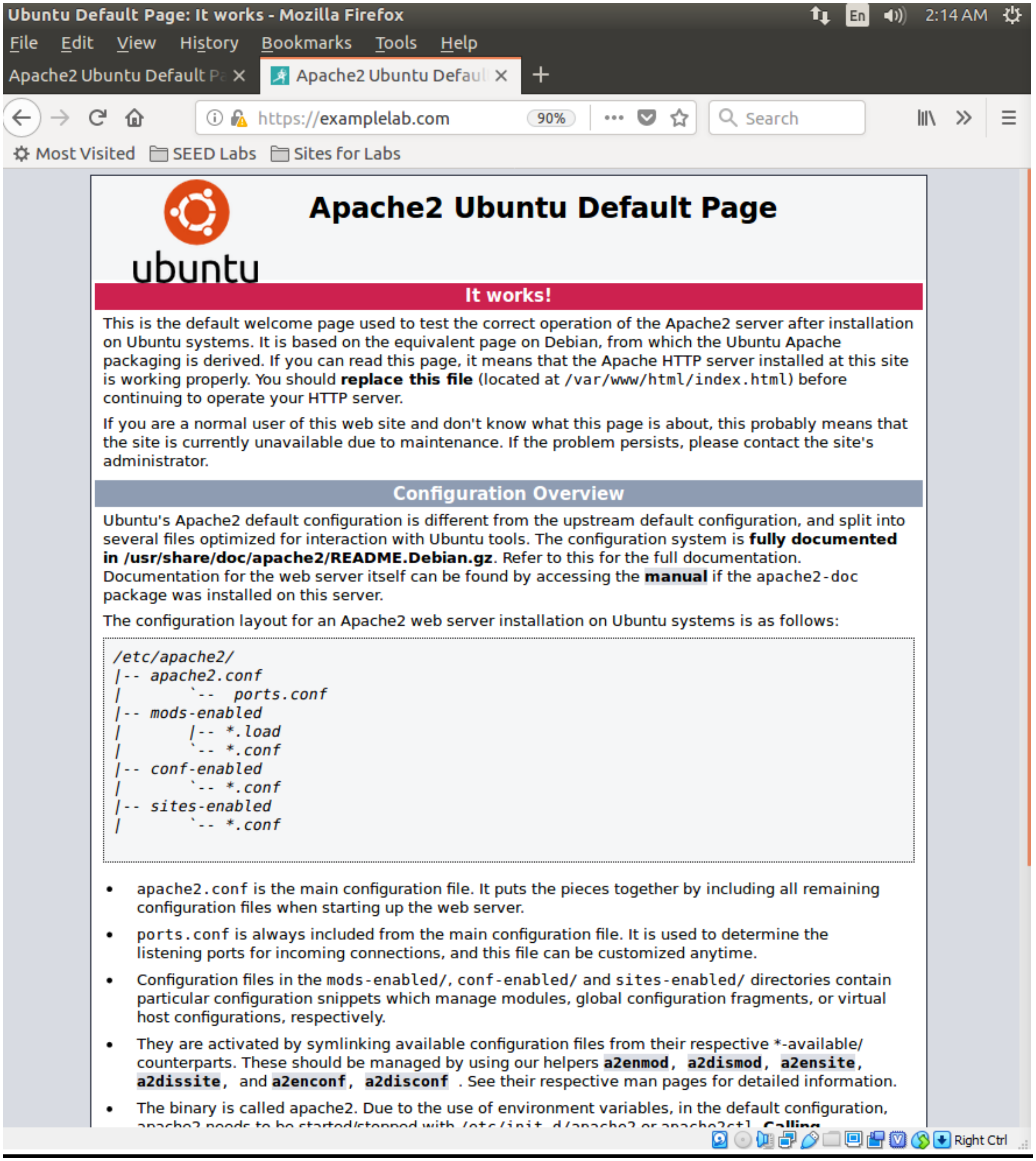
Your connection is not secure

The owner of examplelab.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

Observation: it seems the browser has caught the malicious site, and alerts the user that the connection is not secure, it could be a bad site. It explicitly says “to protect your information from being stolen…” meaning Firefox was able to pick up the activity that was going on.

Task 6


Ubuntu Default Page: It works - Mozilla Firefox
En
2:14 AM
File Edit View History Bookmarks Tools Help
Apache2 Ubuntu Default Pa
Apache2 Ubuntu Defaul
https://examplelab.com
90%
Search
Most Visited
SEED Labs
Sites for Labs
ubuntu
Right Ctrl

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **a2enmod**, **a2dismod**, **a2ensite**, **a2dissite**, and **a2enconf**, **a2disconf** . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. **Calling**

Observation: The attack was a success. As seen by the url, we are on the "fake site", however it brought us to the "SEEDPKILab2018.com" homepage, so the user would not be able to tell the difference. The browser was also unable to alert the user, so they would think they logged onto the correct site, maybe put in some personal information to be stolen by the man in the middle.