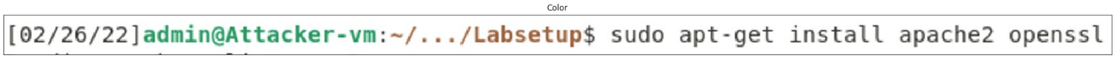
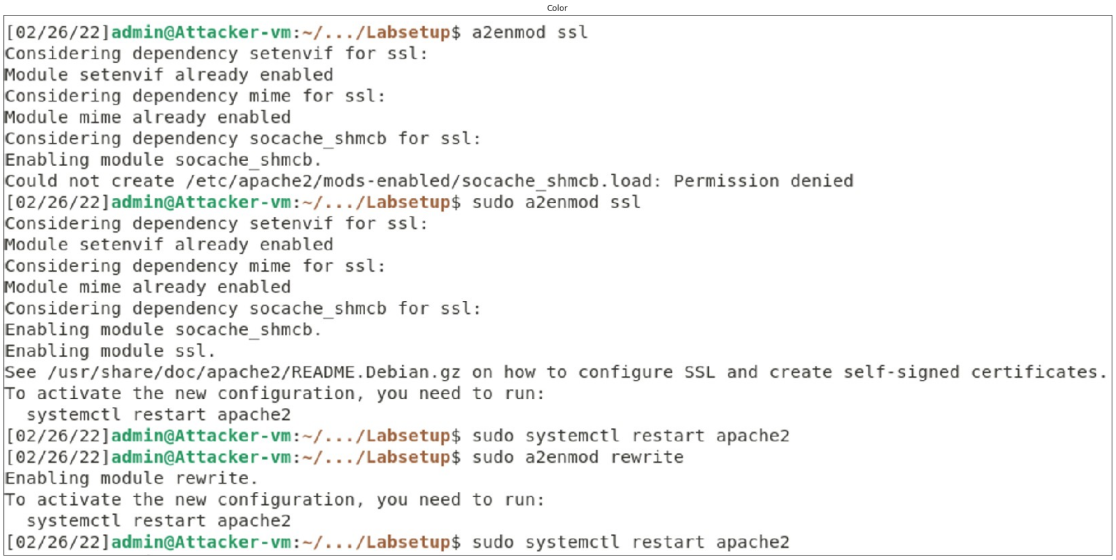
Set Up Process

Since my cloud VM has its openssl and apache2 packages corrupted, I had to reinstall it

```
import cv2
from matplotlib import pyplot as plt

# This is a bit of magic to make matplotlib figures appear inline in
the notebook
# rather than in a new window.
%matplotlib inline
plt.rcParams['figure.figsize'] = (100.0, 80.0) # set default size of
plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'


def show_img(img):
    img = cv2.imread(img,-1)
    plt.subplot(131),plt.imshow(img),
    plt.title('Color'),plt.xticks([]), plt.yticks([])
    plt.show()

show_img('Task 1/setup.png')
show_img('Task 1/setup_2.png')
```



```
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ sudo apt-get install apache2 openssl
```

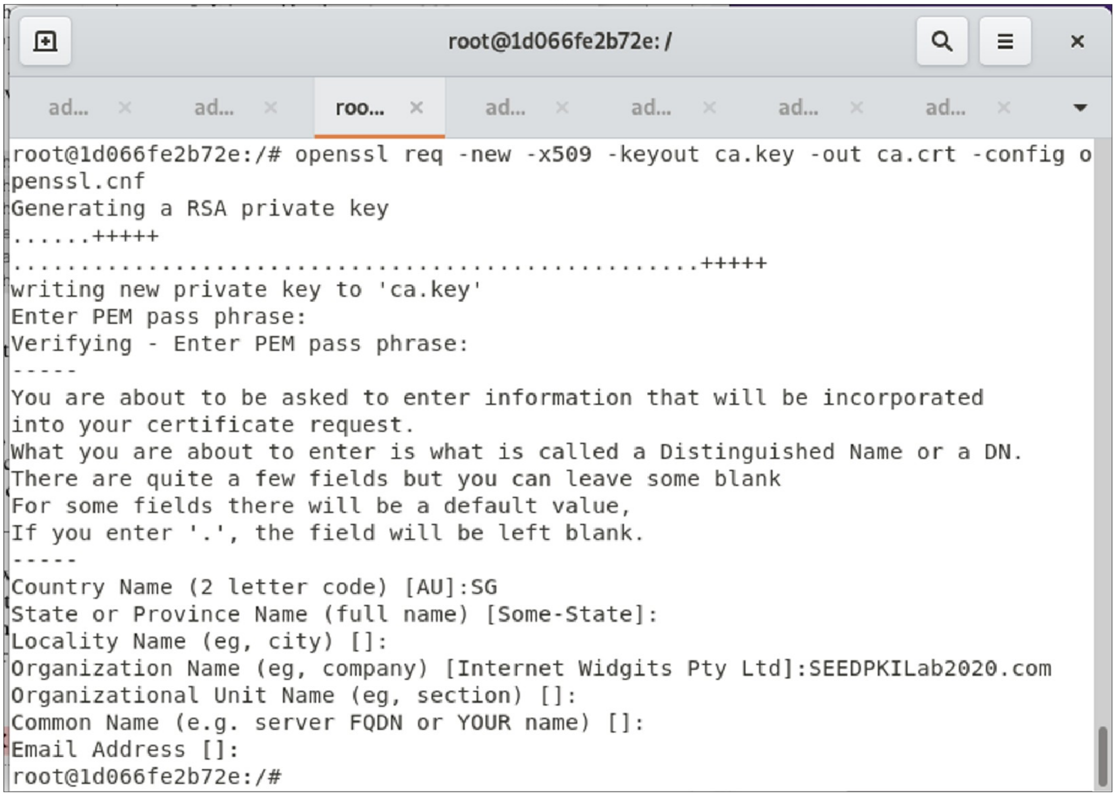


```
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Could not create /etc/apache2/mods-enabled/socache_shmcb.load: Permission denied
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ sudo systemctl restart apache2
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
[02/26/22]admin@Attacker-vm:~/.../Labsetup$ sudo systemctl restart apache2
```

Task 1: Becoming a Certificate Authority (CA)

```
show_img('Task 1/gen_cert.png')
```

Color



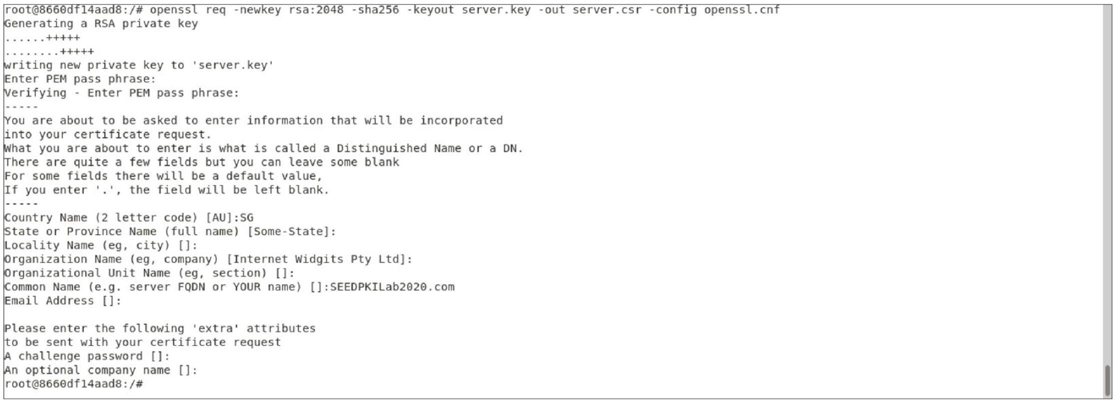
```
root@1d066fe2b72e:/# openssl req -new -x509 -keyout ca.key -out ca.crt -config o
penssl.cnf
Generating a RSA private key
......+++++
.....................................................+++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:SG
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SEEDPKILab2020.com
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
root@1d066fe2b72e:/#
```

Task 2: Creating a Certificate for SEEDPKILab2020.com

Step 2: Generate a Certificate Signing Request (CSR)

```
show_img('Task 2/gen_csr.png')
```

Color

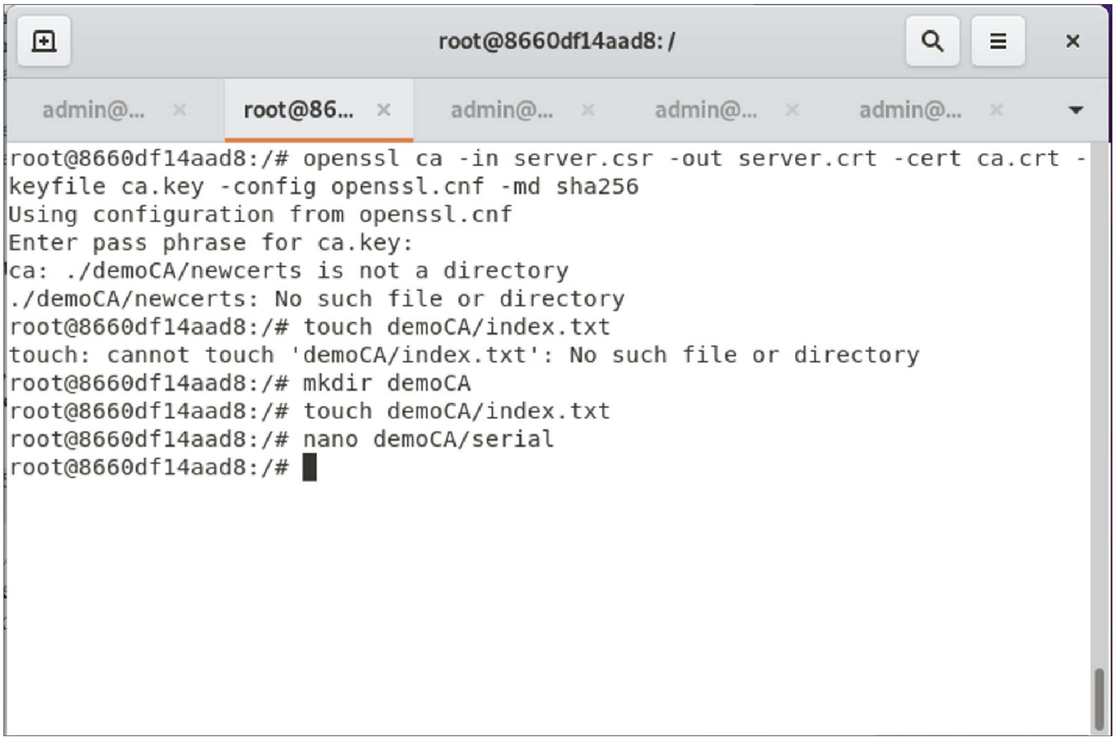


```
root@8660df14aad8:/# openssl req -newkey rsa:2048 -sha256 -keyout server.key -out server.csr -config openssl.cnf
Generating a RSA private key
......+++++
........+++++
writing new private key to 'server.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked 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]:SG
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2020.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@8660df14aad8:/#
```

Step 3: Generating Certificates

```
show_img('Task 1/setup_3.png')
```

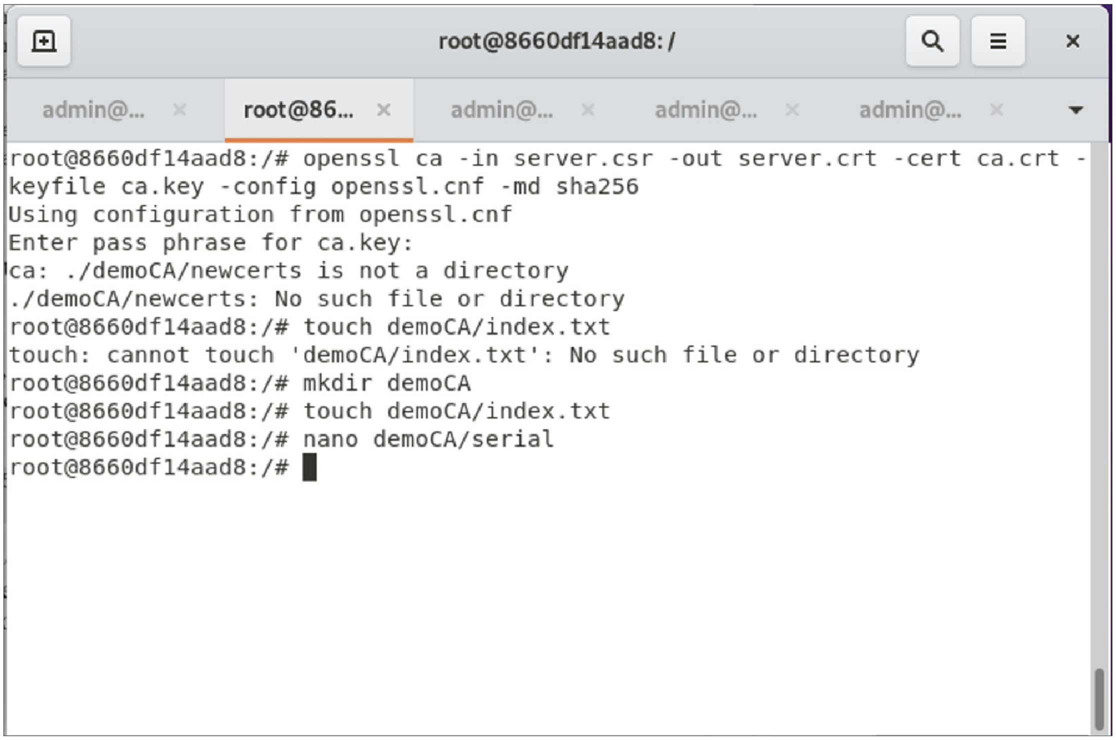
Color



```
root@8660df14aad8:/# openssl ca -in server.csr -out server.crt -cert ca.crt -
keyfile ca.key -config openssl.cnf -md sha256
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
ca: ./demoCA/newcerts is not a directory
./demoCA/newcerts: No such file or directory
root@8660df14aad8:/# touch demoCA/index.txt
touch: cannot touch 'demoCA/index.txt': No such file or directory
root@8660df14aad8:/# mkdir demoCA
root@8660df14aad8:/# touch demoCA/index.txt
root@8660df14aad8:/# nano demoCA/serial
root@8660df14aad8:/# █
```

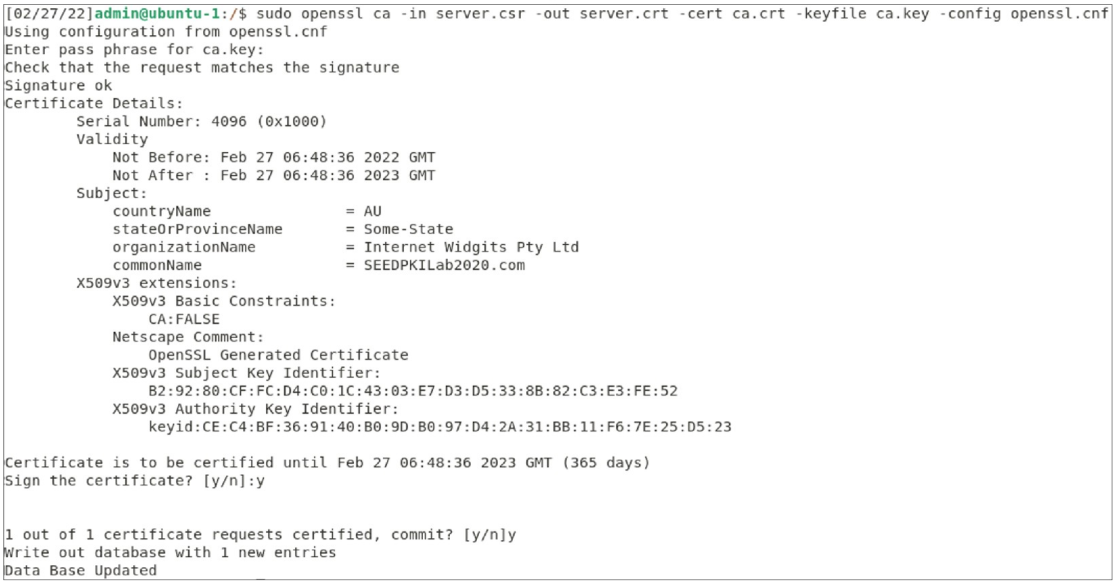
```
show_img('Task 2/convert_crt.png')
show_img('Task 2/convert_crt_2.png')
```

Color



```
root@8660df14aad8:/# openssl ca -in server.csr -out server.crt -cert ca.crt -
keyfile ca.key -config openssl.cnf -md sha256
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
ca: ./demoCA/newcerts is not a directory
./demoCA/newcerts: No such file or directory
root@8660df14aad8:/# touch demoCA/index.txt
touch: cannot touch 'demoCA/index.txt': No such file or directory
root@8660df14aad8:/# mkdir demoCA
root@8660df14aad8:/# touch demoCA/index.txt
root@8660df14aad8:/# nano demoCA/serial
root@8660df14aad8:/# █
```

Color



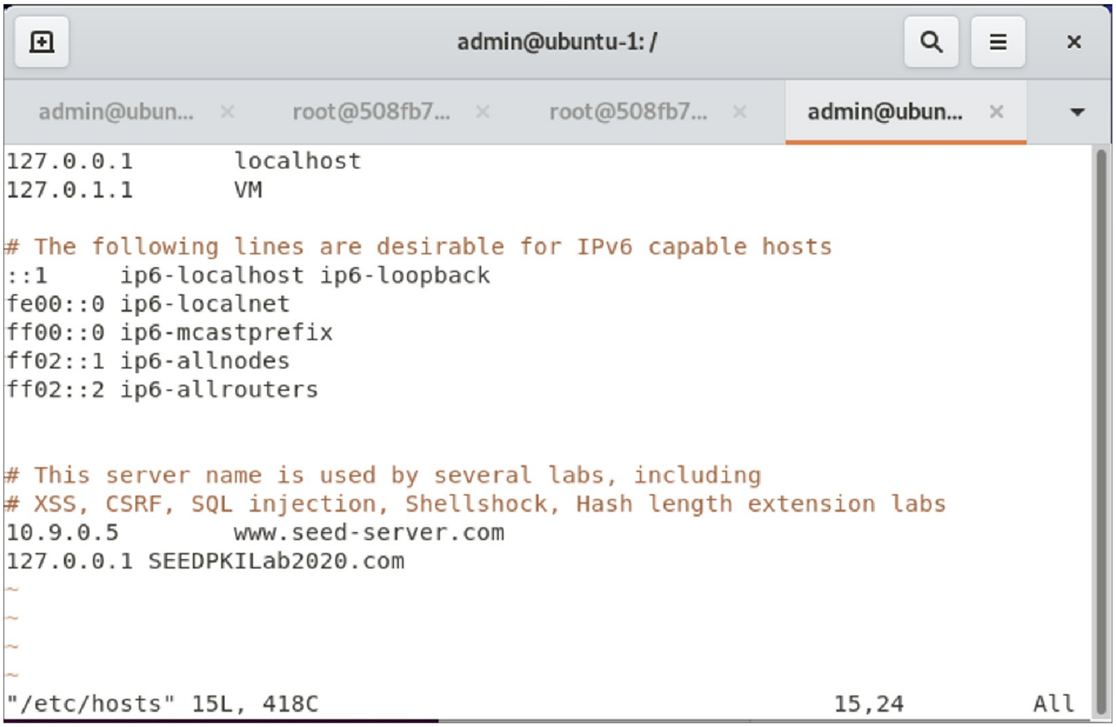
```
[02/27/22]admin@ubuntu-1:/$ sudo openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Feb 27 06:48:36 2022 GMT
            Not After : Feb 27 06:48:36 2023 GMT
        Subject:
            countryName               = AU
            stateOrProvinceName       = Some-State
            organizationName          = Internet Widgits Pty Ltd
            commonName                = SEEDPKILab2020.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                B2:92:80:CF:FC:D4:C0:1C:43:03:E7:D3:D5:33:8B:82:C3:E3:FE:52
            X509v3 Authority Key Identifier:
                keyid:CE:C4:BF:36:91:40:B0:9D:B0:97:D4:2A:31:BB:11:F6:7E:25:D5:23

Certificate is to be certified until Feb 27 06:48:36 2023 GMT (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
```

Task 3: Deploying Certificate in an HTTPS Web Server

Step 1: Configuring DNS

```
show_img('Task 3/config_dns.png')
```

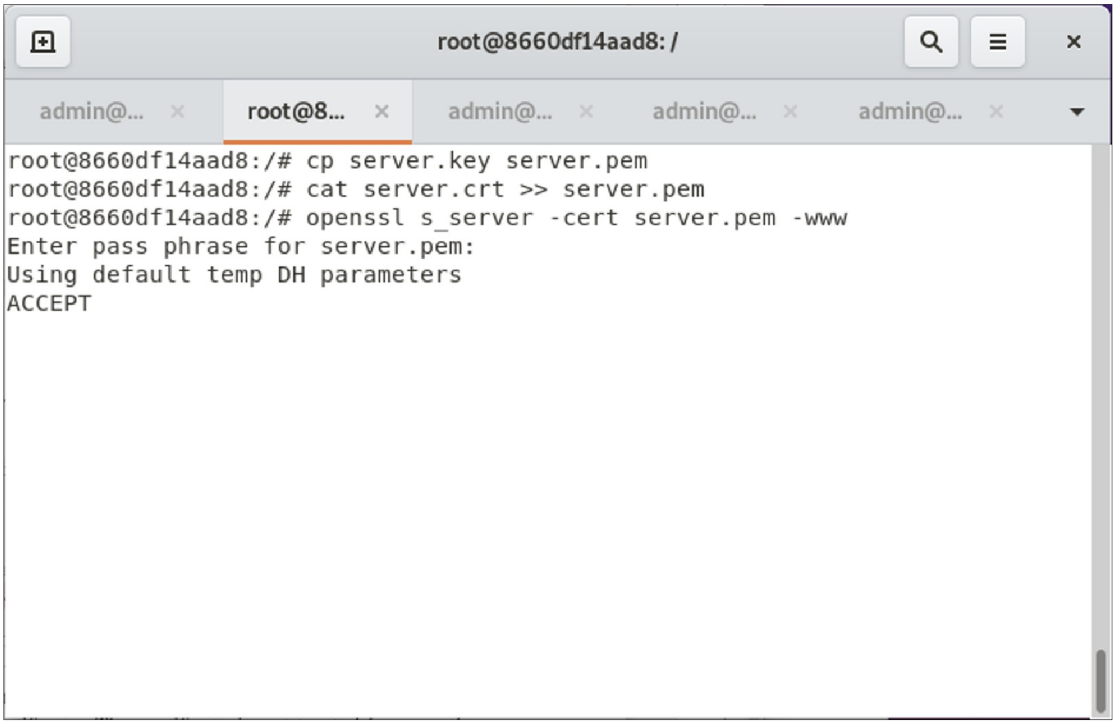
Color



```
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


# This server name is used by several labs, including
# XSS, CSRF, SQL injection, Shellshock, Hash length extension labs
10.9.0.5        www.seed-server.com
127.0.0.1 SEEDPKILab2020.com
~
~
~
~
"/etc/hosts" 15L, 418C                                    15,24         All
```

Step 2: Configuring the web server

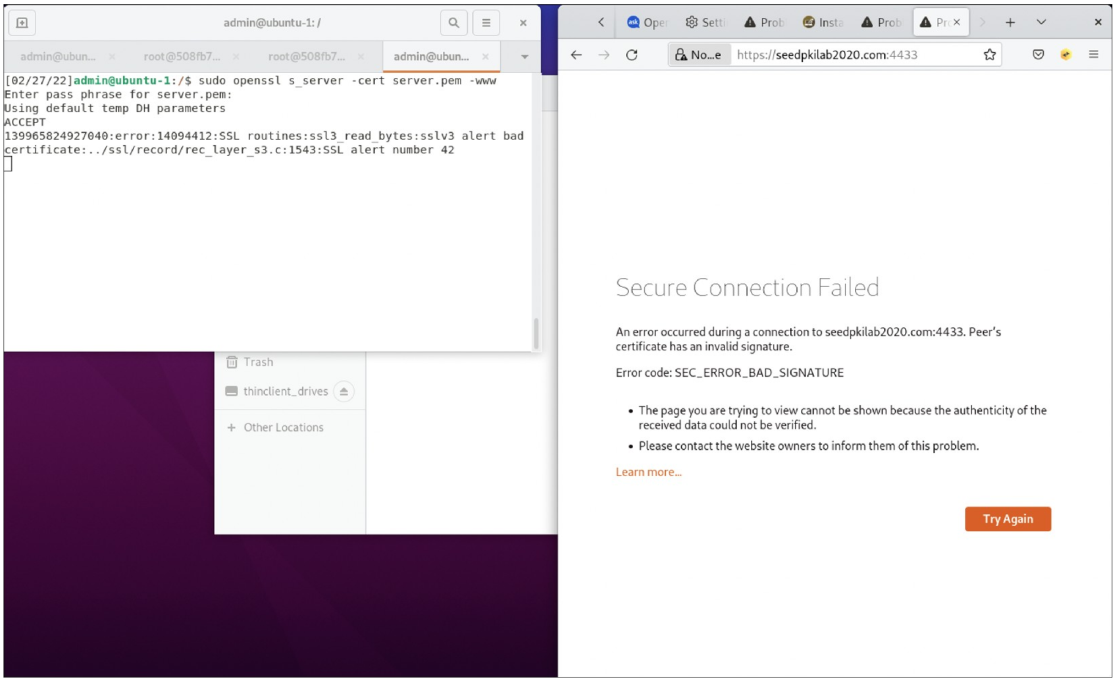
```
show_img('Task 3/config_server.png')
```



```
root@8660df14aad8:/# cp server.key server.pem
root@8660df14aad8:/# cat server.crt >> server.pem
root@8660df14aad8:/# openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

Step 3: Getting the browser to accept our CA certificate

```
show_img('Task 3/config_server_vm.png')
```



```
[02/27/22]admin@ubuntu-1:/$ sudo openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
139965824927040:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad
certificate:../ssl/record/rec_layer_s3.c:1543:SSL alert number 42
```

Secure Connection Failed

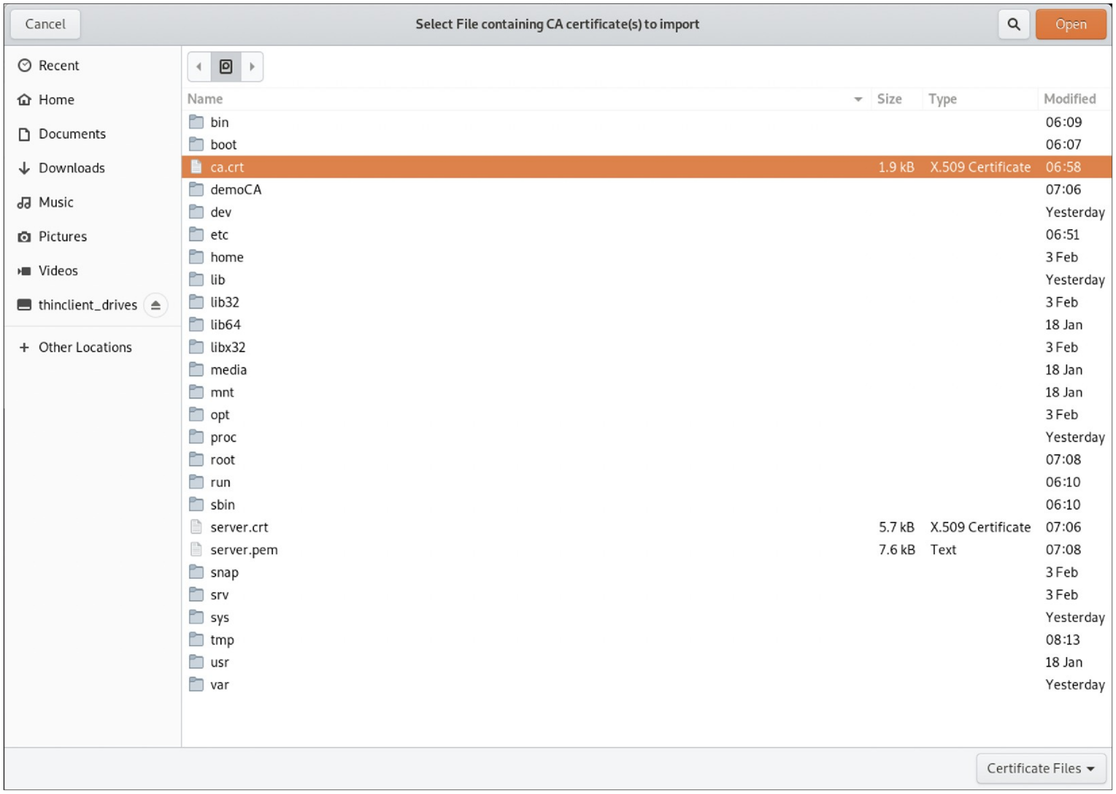
An error occurred during a connection to seedpkilab2020.com:4433. Peer's certificate has an invalid signature.

Error code: SEC_ERROR_BAD_SIGNATURE

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

```
show_img('Task 3/import_ca.png')
show_img('Task 3/import_ca_2.png')
```

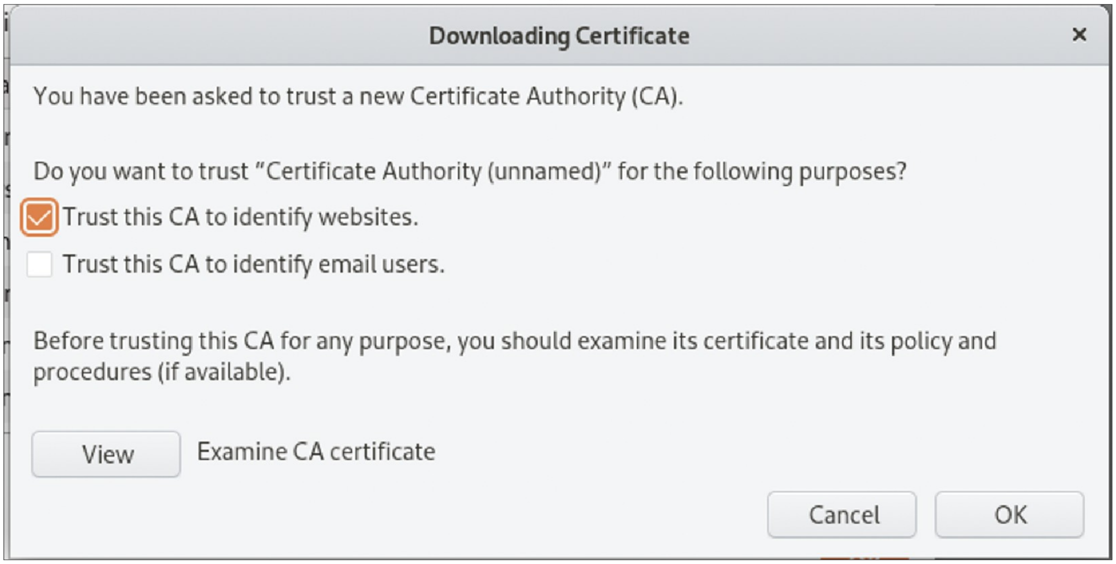
Color



Select File containing CA certificate(s) to import

Name	Size	Type	Modified
bin			06:09
boot			06:07
ca.crt	1.9 kB	X.509 Certificate	06:58
demoCA			07:06
dev			Yesterday
etc			06:51
home			3 Feb
lib			Yesterday
lib32			3 Feb
lib64			18 Jan
libx32			3 Feb
media			18 Jan
mnt			18 Jan
opt			3 Feb
proc			Yesterday
root			07:08
run			06:10
sbin			06:10
server.crt	5.7 kB	X.509 Certificate	07:06
server.pem	7.6 kB	Text	07:08
snap			3 Feb
srv			3 Feb
sys			Yesterday
tmp			08:13
usr			18 Jan
var			Yesterday

Color



Downloading Certificate

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

Do you want to trust "Certificate Authority (unnamed)" for the following purposes?

☑ Trust this CA to identify websites.

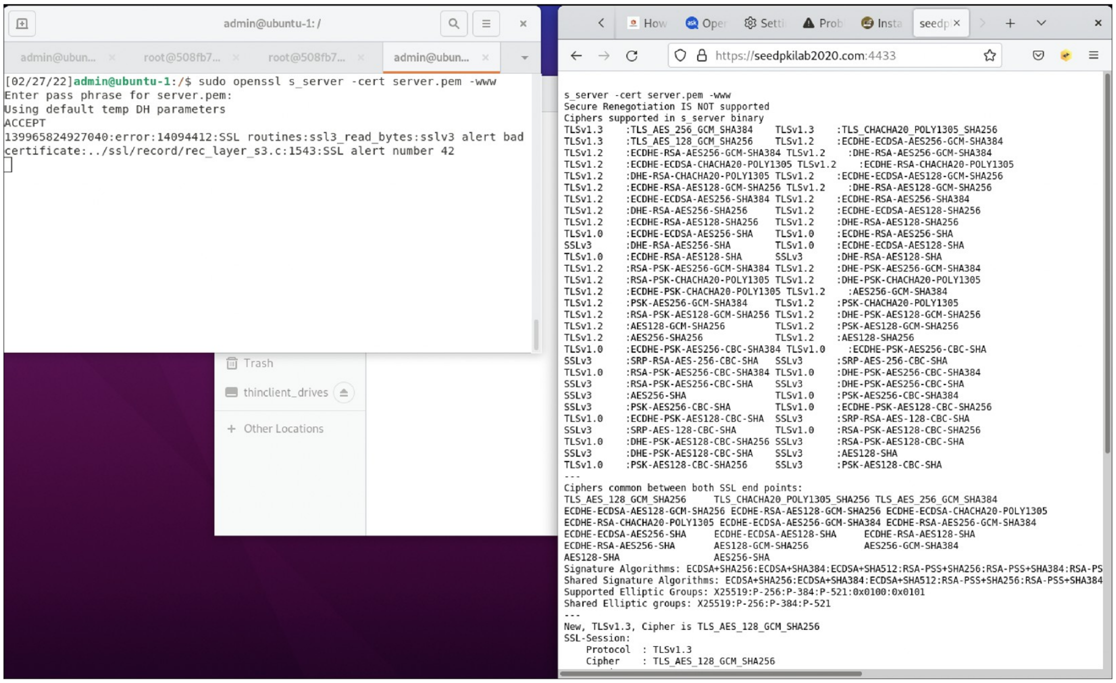
□ Trust this CA to identify email users.

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

View Examine CA certificate

```
show_img('Task 3/browser_w_ca.png')
```

Color



```
[02/27/22]admin@ubuntu-1:/$ sudo openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
139965824927040:error:14094412:SSL routines:ssl3_read_bytes:sslv3 alert bad
certificate:../ssl/record/rec_layer_s3.c:1543:SSL alert number 42
```

```
s_server -cert server.pem -www
Secure Renegotiation IS NOT supported
Ciphers supported in s_server binary
TLSv1.3    :TLS_AES_256_GCM_SHA384    TLSv1.3    :TLS_CHACHA20_POLY1305_SHA256
TLSv1.3    :TLS_AES_128_GCM_SHA256    TLSv1.2    :ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2    :DHE-RSA-AES256-GCM-SHA384
TLSv1.2    :ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-RSA-CHACHA20-POLY1305
TLSv1.2    :DHE-RSA-CHACHA20-POLY1305 TLSv1.2    :ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2    :DHE-RSA-AES128-GCM-SHA256
TLSv1.2    :ECDHE-ECDSA-AES256-SHA384 TLSv1.2    :ECDHE-RSA-AES256-SHA384
TLSv1.2    :DHE-RSA-AES256-SHA256     TLSv1.2    :ECDHE-ECDSA-AES128-SHA256
TLSv1.2    :ECDHE-RSA-AES128-SHA256   TLSv1.2    :DHE-RSA-AES128-SHA256
TLSv1.0    :ECDHE-ECDSA-AES256-SHA    TLSv1.0    :ECDHE-RSA-AES256-SHA
SSLv3      :DHE-RSA-AES256-SHA        TLSv1.0    :ECDHE-ECDSA-AES128-SHA
TLSv1.0    :ECDHE-RSA-AES128-SHA      SSLv3      :DHE-RSA-AES128-SHA
TLSv1.2    :RSA-PSK-AES256-GCM-SHA384 TLSv1.2    :DHE-PSK-AES256-GCM-SHA384
TLSv1.2    :RSA-PSK-CHACHA20-POLY1305 TLSv1.2    :DHE-PSK-CHACHA20-POLY1305
TLSv1.2    :ECDHE-PSK-CHACHA20-POLY1305 TLSv1.2    :AES256-GCM-SHA384
TLSv1.2    :PSK-AES256-GCM-SHA384     TLSv1.2    :PSK-CHACHA20-POLY1305
TLSv1.2    :RSA-PSK-AES128-GCM-SHA256 TLSv1.2    :DHE-PSK-AES128-GCM-SHA256
TLSv1.2    :AES128-GCM-SHA256         TLSv1.2    :PSK-AES128-GCM-SHA256
TLSv1.2    :AES256-SHA256             TLSv1.2    :AES128-SHA256
TLSv1.0    :ECDHE-PSK-AES256-CBC-SHA384 TLSv1.0    :ECDHE-PSK-AES256-CBC-SHA
SSLv3      :SRP-RSA-AES-256-CBC-SHA   SSLv3      :SRP-AES-256-CBC-SHA
TLSv1.0    :RSA-PSK-AES256-CBC-SHA384 TLSv1.0    :DHE-PSK-AES256-CBC-SHA384
SSLv3      :RSA-PSK-AES256-CBC-SHA    SSLv3      :DHE-PSK-AES256-CBC-SHA
SSLv3      :AES256-SHA                TLSv1.0    :PSK-AES256-CBC-SHA384
SSLv3      :PSK-AES256-CBC-SHA        TLSv1.0    :ECDHE-PSK-AES128-CBC-SHA256
TLSv1.0    :ECDHE-PSK-AES128-CBC-SHA  SSLv3      :SRP-RSA-AES-128-CBC-SHA
SSLv3      :SRP-AES-128-CBC-SHA       TLSv1.0    :RSA-PSK-AES128-CBC-SHA256
TLSv1.0    :DHE-PSK-AES128-CBC-SHA256 SSLv3      :RSA-PSK-AES128-CBC-SHA
SSLv3      :DHE-PSK-AES128-CBC-SHA    SSLv3      :AES128-SHA
TLSv1.0    :PSK-AES128-CBC-SHA256     SSLv3      :PSK-AES128-CBC-SHA
---
Ciphers common between both SSL end points:
TLS_AES_128_GCM_SHA256     TLS_CHACHA20_POLY1305_SHA256 TLS_AES_256_GCM_SHA384
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA     ECDHE-ECDSA-AES128-SHA     ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES256-SHA       AES128-GCM-SHA256          AES256-GCM-SHA384
AES128-SHA                 AES256-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PS
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384
Supported Elliptic Groups: X25519:P-256:P-384:P-521:0x0100:0x0101
Shared Elliptic groups: X25519:P-256:P-384:P-521
---
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_128_GCM_SHA256
```

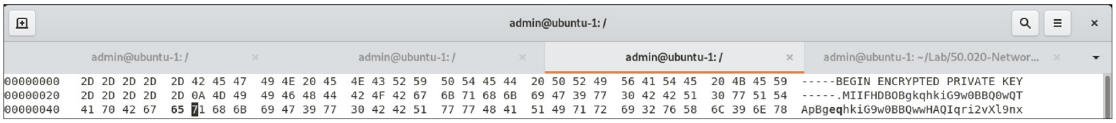
Step 4: Testing our HTTPS website

Using hexdit to change the bytes from 6B to 65: `hexedit server.pem`

The edit is made at a crucial place, and thus corrupting the file. This results in the inability to load the server certificate private key file.

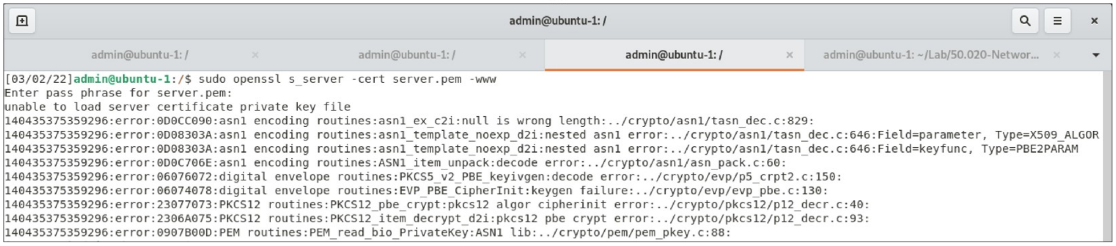
```
show_img('Task 3/mod_byte.png')
show_img('Task 3/corrupted_out.png')
```

Color



```
00000000   2D 2D 2D 2D  2D 42 45 47  49 4E 20 45  4E 43 52 59  50 54 45 44  20 50 52 49  56 41 54 45  20 4B 45 59  -----BEGIN ENCRYPTED PRIVATE KEY
00000020   2D 2D 2D 2D  2D 0A 4D 49  49 46 48 44  42 4F 42 67  6B 71 68 6B  69 47 39 77  30 42 42 51  30 77 51 54  -----.MIIFHDBOBgkqhkiG9w0BBQ0wQT
00000040   41 70 42 67  65 71 68 6B  69 47 39 77  30 42 42 51  77 77 48 41  51 49 71 72  69 32 76 58  6C 39 6E 78  ApBgeqhkiG9w0BBQwwHAQIqri2vXl9nx
```

Color



```
[03/02/22]admin@ubuntu-1:/$ sudo openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
unable to load server certificate private key file
140435375359296:error:0D0CC090:asn1 encoding routines:asn1_ex_c2i:null is wrong length:../crypto/asn1/tasn_dec.c:829:
140435375359296:error:0D08303A:asn1 encoding routines:asn1_template_noexp_d2i:nested asn1 error:../crypto/asn1/tasn_dec.c:646:Field=parameter, Type=X509_ALGOR
140435375359296:error:0D08303A:asn1 encoding routines:asn1_template_noexp_d2i:nested asn1 error:../crypto/asn1/tasn_dec.c:646:Field=keyfunc, Type=PBE2PARAM
140435375359296:error:0D0C706E:asn1 encoding routines:ASN1_item_unpack:decode error:../crypto/asn1/asn_pack.c:60:
140435375359296:error:06076072:digital envelope routines:PKCS5_v2_PBE_keyivgen:decode error:../crypto/evp/p5_crpt2.c:150:
140435375359296:error:06074078:digital envelope routines:EVP_PBE_CipherInit:keygen failure:../crypto/evp/evp_pbe.c:130:
140435375359296:error:23077073:PKCS12 routines:PKCS12_pbe_crypt:pkcs12 algor cipherinit error:../crypto/pkcs12/p12_decr.c:40:
140435375359296:error:2306A075:PKCS12 routines:PKCS12_item_decrypt_d2i:pkcs12 pbe crypt error:../crypto/pkcs12/p12_decr.c:93:
140435375359296:error:0907B00D:PEM routines:PEM_read_bio_PrivateKey:ASN1 lib:../crypto/pem/pem_pkey.c:88:
```

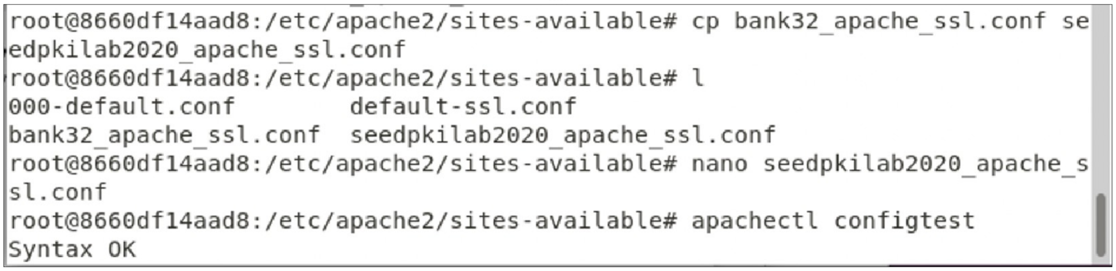
Task 4: Deploying Certificate in an Apache-Based HTTPS Website

ServerName is changed to SEEDPKILab2020.com for 443 and 80. DocumentRoot is changed to /var/www/html, which is the default Apache Ubuntu default page, since the

SSLCertificateFile server.crt and SSLCertificateKeyFile server.key is added to the server.pem, we can use it in the config below.

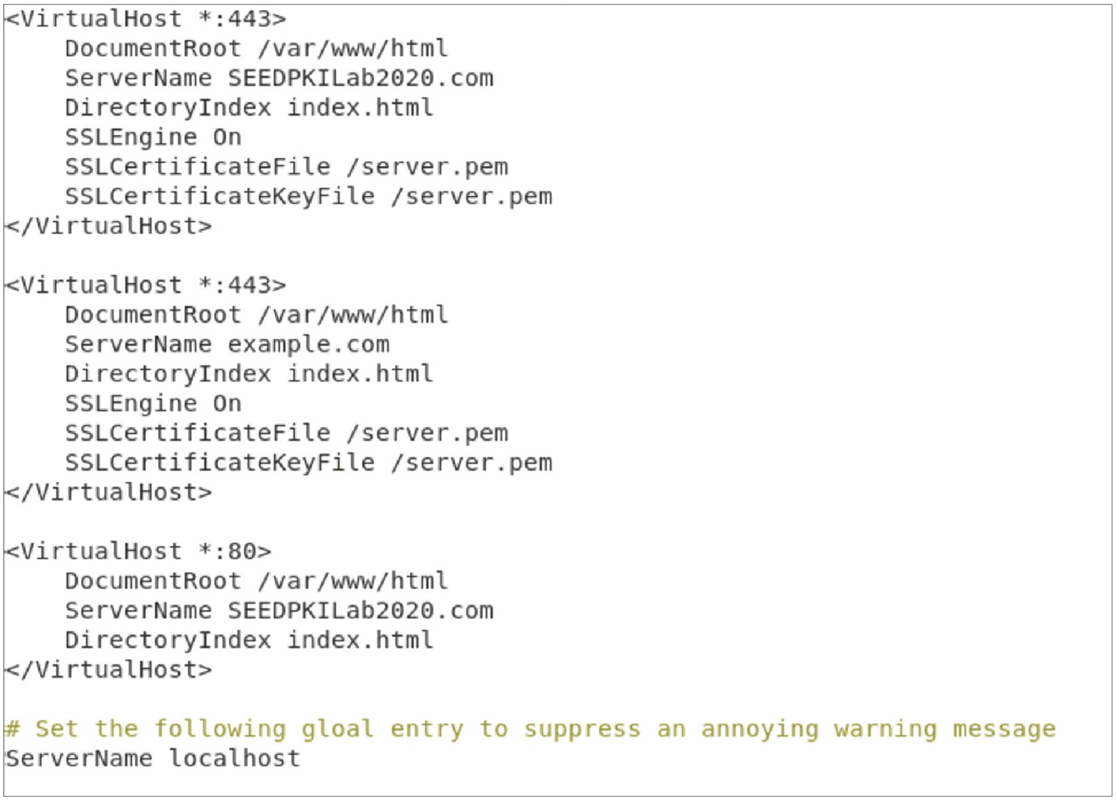
```
show_img('Task 4/mod_seedpkilab2020_apache_ssl.png')
show_img('Task 4/seedpkilab2020_apache_ssl.png')
```

Color



```
root@8660df14aad8:/etc/apache2/sites-available# cp bank32_apache_ssl.conf se
edpkilab2020_apache_ssl.conf
root@8660df14aad8:/etc/apache2/sites-available# l
000-default.conf        default-ssl.conf
bank32_apache_ssl.conf  seedpkilab2020_apache_ssl.conf
root@8660df14aad8:/etc/apache2/sites-available# nano seedpkilab2020_apache_s
sl.conf
root@8660df14aad8:/etc/apache2/sites-available# apachectl configtest
Syntax OK
```

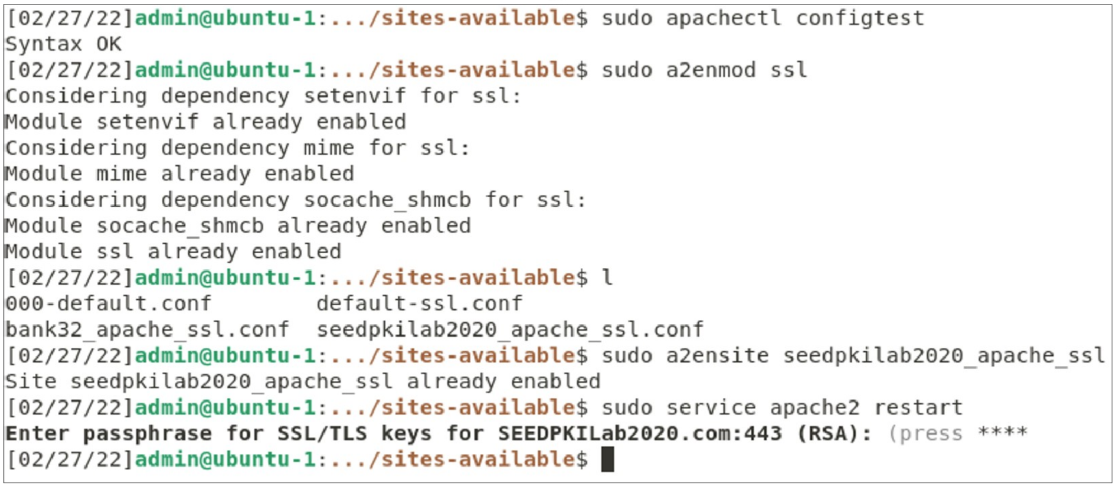
Color



```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /server.pem
    SSLCertificateKeyFile /server.pem
</VirtualHost>

<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName example.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /server.pem
    SSLCertificateKeyFile /server.pem
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
</VirtualHost>

# Set the following gloal entry to suppress an annoying warning message
ServerName localhost
```

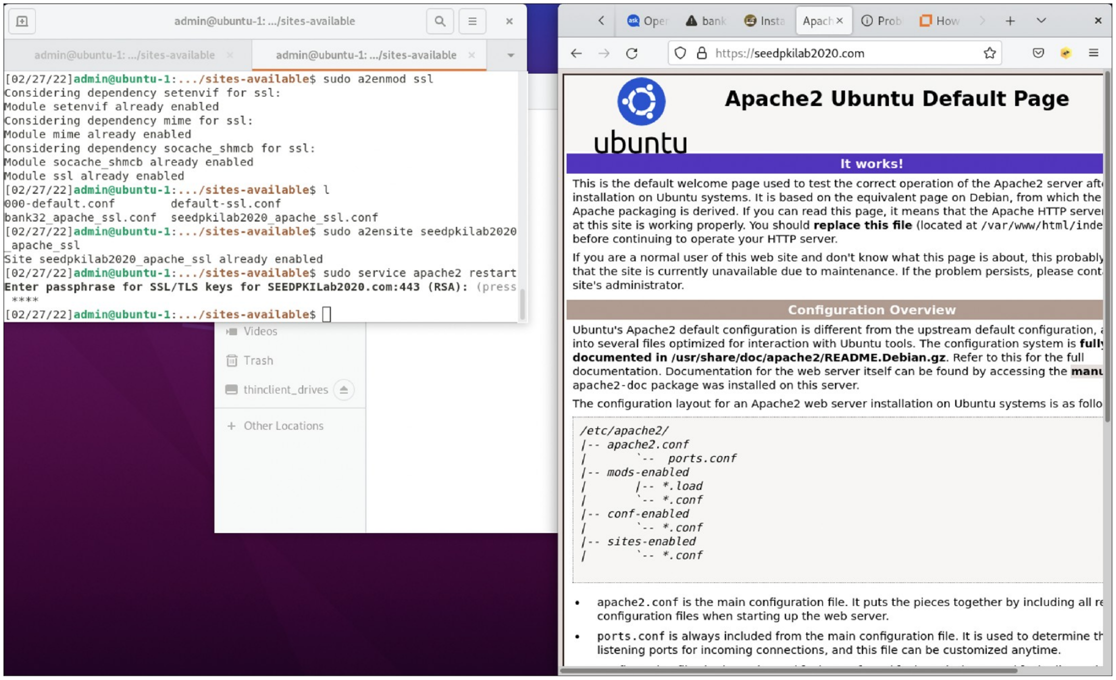
```
show_img('Task 4/deploy_server.png')
show_img('Task 4/browser_seedpiklab2020.png')
```

Color



```
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo apachectl configtest
Syntax OK
[02/27/22]admin@ubuntu-1:.../sites-available$ 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
[02/27/22]admin@ubuntu-1:.../sites-available$ l
000-default.conf        default-ssl.conf
bank32_apache_ssl.conf  seedpkilab2020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo a2ensite seedpkilab2020_apache_ssl
Site seedpkilab2020_apache_ssl already enabled
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): (press ****
[02/27/22]admin@ubuntu-1:.../sites-available$ 
```

Color



admin@ubuntu-1: .../sites-available
admin@ubuntu-1: .../sites-available
admin@ubuntu-1: .../sites-available
Videos
Trash
thinclient_drives
+ Other Locations
Oper
bank
Insta
Apach
Prob
How

```
[02/27/22]admin@ubuntu-1:.../sites-available$ 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
[02/27/22]admin@ubuntu-1:.../sites-available$ l
000-default.conf        default-ssl.conf
bank32_apache_ssl.conf  seedpkilab2020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo a2ensite seedpkilab2020
_apache_ssl
Site seedpkilab2020_apache_ssl already enabled
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): (press
 ****
[02/27/22]admin@ubuntu-1:.../sites-available$ 
```

https://seedpkilab2020.com

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server aft installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Apache packaging is derived. If you can read this page, it means that the Apache HTTP serve at this site is working properly. You should **replace this file** (located at /var/www/html/inde 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 that the site is currently unavailable due to maintenance. If the problem persists, please cont site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, into several files optimized for interaction with Ubuntu tools. The configuration system is **full 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 **manu** apache2-doc package was installed on this server.

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

```
/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 re configuration files when starting up the web server.
- ports.conf is always included from the main configuration file. It is used to determine th listening ports for incoming connections, and this file can be customized anytime.

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

Step 1: Setting up the malicious website Used example.com for the index.html

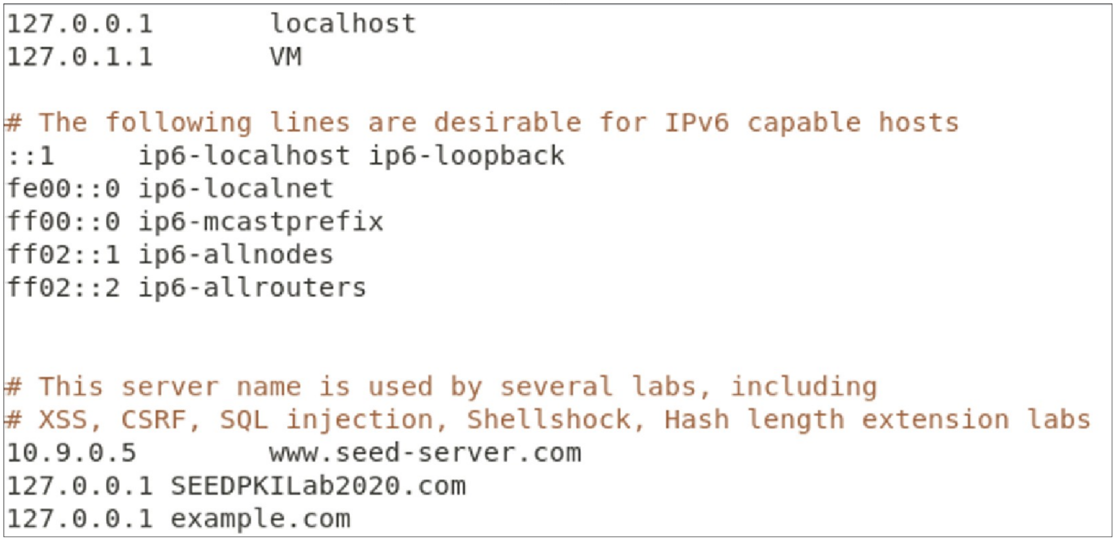
```
!cat apache2/sites-available/seedpkilab2020_apache_ssl.conf

<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /server.pem
    SSLCertificateKeyFile /server.pem
```

```
</VirtualHost>

<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName example.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /example.pem
    SSLCertificateKeyFile /example.pem
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
</VirtualHost>

# Set the following gloal entry to suppress an annoying warning
message
ServerName localhost
```

Step 2: Becoming the man in the middle Edit the /etc/hosts to emulate the DNS cache poisoning attack

```
show_img('Task 5/inject_site.png')
```

Color



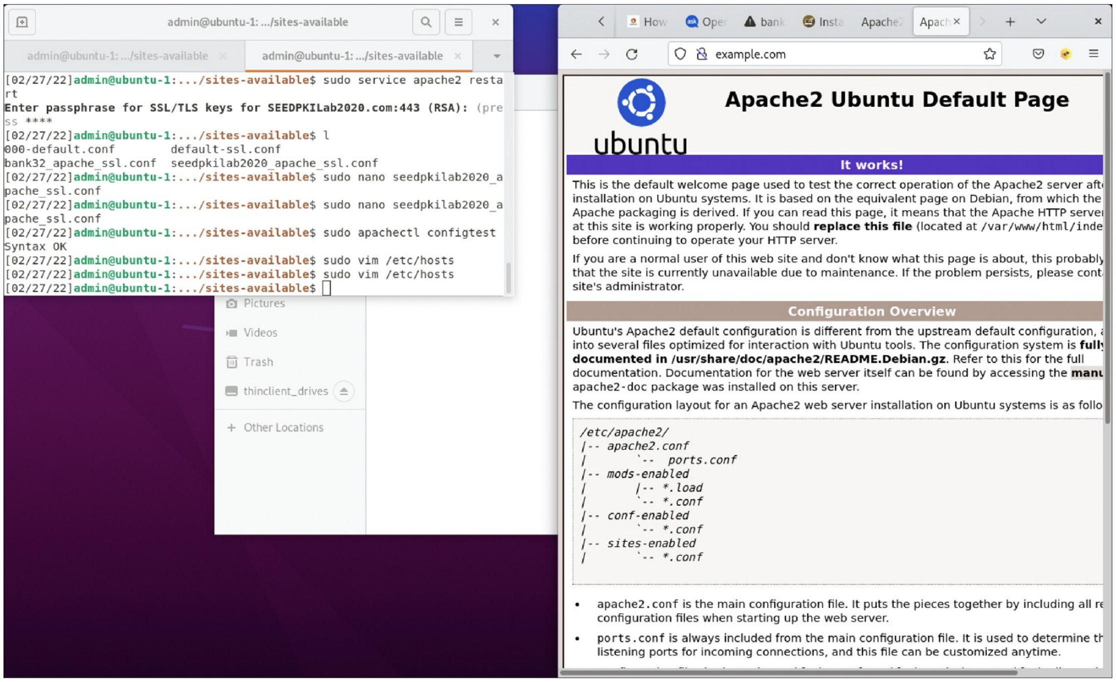
```
127.0.0.1       localhost
127.0.1.1       VM

# 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


# This server name is used by several labs, including
# XSS, CSRF, SQL injection, Shellshock, Hash length extension labs
10.9.0.5        www.seed-server.com
127.0.0.1 SEEDPKILab2020.com
127.0.0.1 example.com
```

The task of creating the csr, cert conversion of csr to crt is the same as in Task 1 and 2, except that the common name is changed to example.com, and example is used as the name for crt, csr and pem extension.

Step 3: Browse the target website

```
show_img('Task 5/browser_example.png')
```

Color


admin@ubuntu-1: .../sites-available
admin@ubuntu-1: .../sites-available
admin@ubuntu-1: .../sites-available
Pictures
Videos
Trash
thinclient_drives
Other Locations
How
Oper
bank
Insta
Apache2
Apach
example.com

```
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo service apache2 resta
rt
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): (pre
ss ****
[02/27/22]admin@ubuntu-1:.../sites-available$ l
000-default.conf        default-ssl.conf
bank32_apache_ssl.conf  seedpkilab2020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo nano seedpkilab2020_a
pache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo nano seedpkilab2020_a
pache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo apachectl configtest
Syntax OK
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo vim /etc/hosts
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo vim /etc/hosts
[02/27/22]admin@ubuntu-1:.../sites-available$ 
```

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server aft installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Apache packaging is derived. If you can read this page, it means that the Apache HTTP serve at this site is working properly. You should **replace this file** (located at `/var/www/html/inde` 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 that the site is currently unavailable due to maintenance. If the problem persists, please cont site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, 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 **manu** apache2-doc package was installed on this server.

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

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

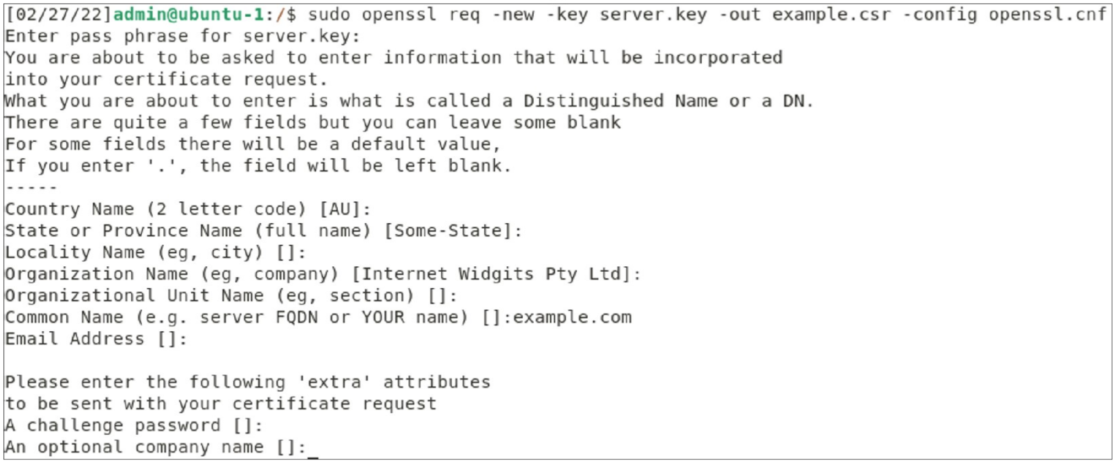
- `apache2.conf` is the main configuration file. It puts the pieces together by including all re configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine th listening ports for incoming connections, and this file can be customized anytime.

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

If the root CA is compromised, the attacker can create sign as my certificates as they want and issue to random sites like example.com, and such fictitious sites would not be flagged by web browsers after deploying to the Apache HTTP server seen in Task 6.

```
show_img('Task 6/create_new_csr.png')
```

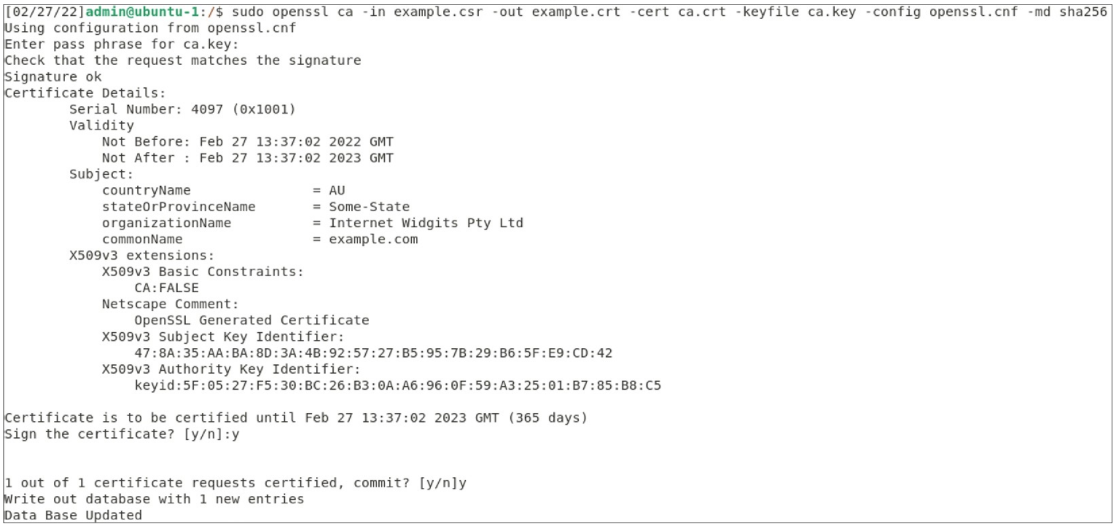
Color



```
[02/27/22]admin@ubuntu-1:/$ sudo openssl req -new -key server.key -out example.csr -config openssl.cnf
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]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:

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

```
show_img('Task 6/convert_crt.png')
```

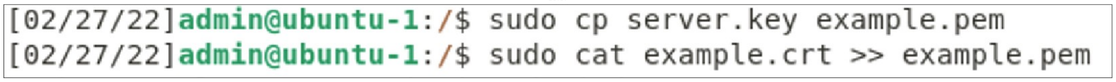
Color



```
[02/27/22]admin@ubuntu-1:/$ sudo openssl ca -in example.csr -out example.crt -cert ca.crt -keyfile ca.key -config openssl.cnf -md sha256
Using configuration from 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 27 13:37:02 2022 GMT
            Not After : Feb 27 13:37:02 2023 GMT
        Subject:
            countryName               = AU
            stateOrProvinceName       = Some-State
            organizationName          = Internet Widgits Pty Ltd
            commonName                = example.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                47:8A:35:AA:BA:8D:3A:4B:92:57:27:B5:95:7B:29:B6:5F:E9:CD:42
            X509v3 Authority Key Identifier:
                keyid:5F:05:27:F5:30:BC:26:B3:0A:A6:96:0F:59:A3:25:01:B7:85:B8:C5

Certificate is to be certified until Feb 27 13:37:02 2023 GMT (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
```

```
show_img('Task 6/create_pem.png')
```

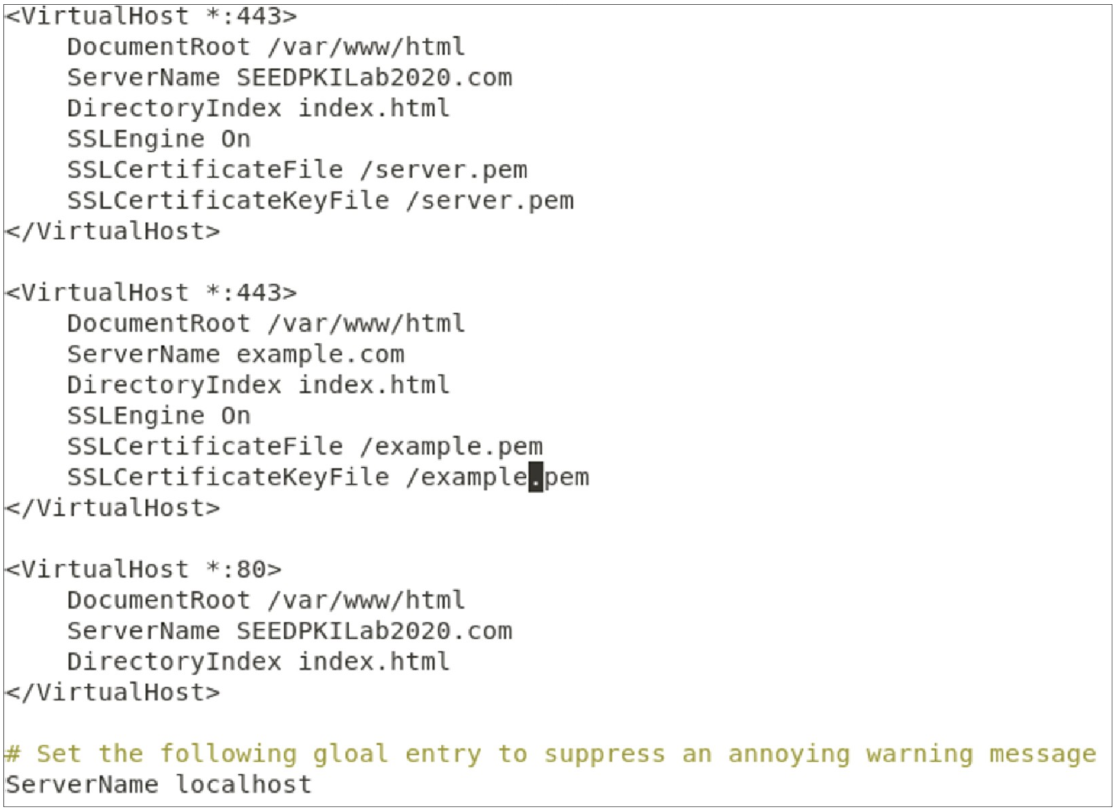
Color



```
[02/27/22]admin@ubuntu-1:/$ sudo cp server.key example.pem
[02/27/22]admin@ubuntu-1:/$ sudo cat example.crt >> example.pem
```

```
show_img('Task 6/edit_dns.png')
```

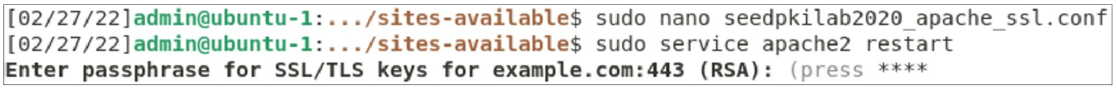
Color



```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /server.pem
    SSLCertificateKeyFile /server.pem
</VirtualHost>

<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerName example.com
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /example.pem
    SSLCertificateKeyFile /example.pem
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /var/www/html
    ServerName SEEDPKILab2020.com
    DirectoryIndex index.html
</VirtualHost>

# Set the following gloal entry to suppress an annoying warning message
ServerName localhost
```

```
show_img('Task 6/restart_server.png')
```

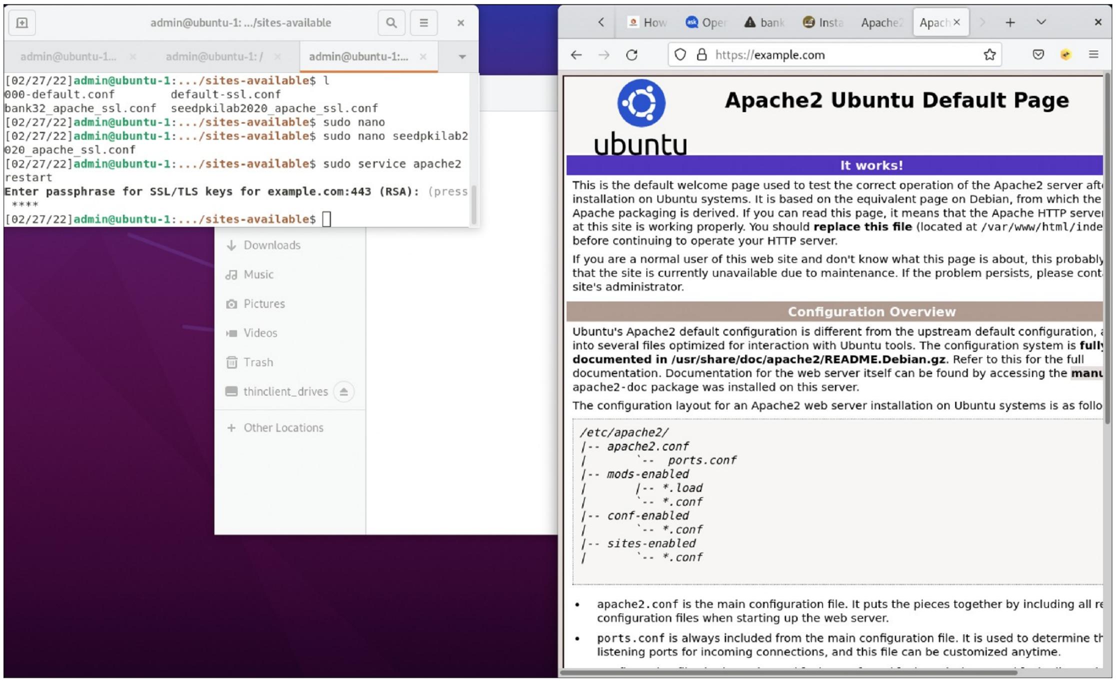
Color



```
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo nano seedpkilab2020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for example.com:443 (RSA): (press ****
```

```
show_img('Task 6/browser_example.png')
```

Color



```
[02/27/22]admin@ubuntu-1:.../sites-available$ l
000-default.conf        default-ssl.conf
bank32_apache_ssl.conf  seedpkilab2020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo nano
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo nano seedpkilab2
020_apache_ssl.conf
[02/27/22]admin@ubuntu-1:.../sites-available$ sudo service apache2
restart
Enter passphrase for SSL/TLS keys for example.com:443 (RSA): (press
 ****
[02/27/22]admin@ubuntu-1:.../sites-available$ 
```

ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server aft installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Apache packaging is derived. If you can read this page, it means that the Apache HTTP serve at this site is working properly. You should **replace this file** (located at `/var/www/html/inde` 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 that the site is currently unavailable due to maintenance. If the problem persists, please cont site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, 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 **manu** apache2-doc package was installed on this server.

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

```
/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 re configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine th listening ports for incoming connections, and this file can be customized anytime.