



Al Imam Mohammad Ibn Saud Islamic University
College of Computer and Information Sciences

Network Security

PKI using SSL



Content

Requirement :	3
Task 1: Becoming a Certificate Authority (CA):	4
Task 2: Creating a Certificate for SEEDPKILab2020.com	5
Step 1: Generate public/private key pair.	5
Step 2: Generate a Certificate Signing Request (CSR)	6
Step 3: Generating Certificates.	6
Task 3: Deploying Certificate in an HTTPS Web Server	7
Step 1: Configuring DNS	7
Step 2: Configuring the web server	8
Step 3: Getting the browser to accept our CA certificate	9
Step 4. Testing our HTTPS website	10
Task 4: Deploying Certificate in an Apache-Based HTTPS Website	12
Task 5: Launching a Man-In-The-Middle Attack	15
Step 1: Setting up the malicious website	15
Step 2: Becoming the man in the middle	16
Step 3: Browse the target website	16
Task 6: Launching a Man-In-The-Middle Attack with a Compromised CA	18



Requirement :

1. Becoming a certificate authority (CA).
2. Creating a certificate.
3. Deploying the certificate in a web server.
4. Deploying certificate in an Apache-based HTTPS website.
5. Launching a Man in the Middle Attack.
6. Launching a Man in the Middle Attack using a compromised CA.



Task 1: Becoming a Certificate Authority (CA):

- First, we Set Up the Configuration File.
- Create the Necessary Directories to storing certificates and other files.
- Use OpenSSL to generate a private key and self-signed Root Certificate , during this process you will have to enter information (The Picture below) , passphrase is important (usually in this lab we make it reverse of seed "dees" but you can choose any) and will be required when we use this Certificate Authority (CA) to sign certificates. After running the command, we will have two files
 - "ca.key" : The private key of the CA
 - "cs.crt" : The self-signed certificate of the CA

```
[05/17/24]seed@VM:~/.../Abdulmajeed$ cp /usr/lib/ssl/openssl.cnf ./openssl.cnf
[05/17/24]seed@VM:~/.../Abdulmajeeds mkdir -p demoCA/newcerts
[05/17/24]seed@VM:~/.../Abdulmajeeds mkdir -p demoCA/certs
[05/17/24]seed@VM:~/.../Abdulmajeeds mkdir -p demoCA/crl
[05/17/24]seed@VM:~/.../Abdulmajeeds touch demoCA/index.txt
[05/17/24]seed@VM:~/.../Abdulmajeed$ echo 1000 > demoCA/serial
[05/17/24]seed@VM:~/.../Abdulmajeeds 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]:SA
State or Province Name (full name) [Some-State]:Riyadh
Locality Name (eg, city) []:Riyadh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:STC
Organizational Unit Name (eg, section) []:STCKSA
Common Name (e.g. server FQDN or YOUR name) []:EE
Email Address []:majeed.a.gh@gmail.com
[05/17/24]seed@VM:~/.../Abdulmajeed$
```



Task 2: Creating a Certificate for SEEDPKILab2020.com

Step 1: Generate public/private key pair.

this command will generate RSA key pair and store it in server.key file

Enter the pass phrase we choose it before

```
Terminal [05/17/24]seed@VM:~/.../Abdulmajeed$ 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:
[05/17/24]seed@VM:~/.../Abdulmajeed$ openssl rsa -in server.key -text
Enter pass phrase for server.key:|
```

```
Terminal [05/17/24]seed@VM:~/.../Abdulmajeed$ 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:
[05/17/24]seed@VM:~/.../Abdulmajeed$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
 00:b1:f9:bb:d5:81:eb:4a:ea:ca:c8:0a:c1:a7:f3:
71:33:18:3f:a2:c8:09:77:21:cf:8c:e7:16:68:04:
84:cb:e4:0f:70:1b:e2:66:6a:84:dc:25:a2:2e:0b:
1b:4a:86:2a:26:72:a8:f1:2a:f5:3d:f8:00:94:98:
74:d0:69:92:0b:38:6d:7f:f2:e2:b5:44:02:57:60:
95:a4:68:20:2f:ec:a9:f8:04:11:fd:21:19:3f:
b3:4e:36:72:09:e2:42:c2:68:56:37:64:c3:82:36:
76:80:d0:66:2f:d5:f6:f7:61:6a:ad:62:4c:ed:4b:
98:cd:75:9b:60:28:00:99:13
publicExponent: 65537 (0x10001)
privateExponent:
 2d:86:70:21:c1:45:ea:bf:7f:07:21:5a:50:cf:
57:fe:8f:e8:97:ba:b4:1a:95:f8:b8:f3:e6:4f:al:
98:76:11:ec:df:75:1e:35:89:1e:b7:11:b1:2d:6e:
21:b6:07:fa:25:c2:49:4e:6f:c5:16:fb:a3:07:0c:
7a:73:46:c3:e2:ad:50:75:df:27:ec:e6:2a:1a:87:
3b:e1:03:10:23:ca:ac:1e:8f:d1:57:cb:c8:bd:e9:
d3:41:b6:4f:a0:f8:98:92:10:27:79:b6:ff:0b:9c:
5c:58:d0:19:79:e3:ea:7f:b2:62:76:f4:53:33:0e:
66:09:da:b8:fb:4b:6d:61
prime1:
 00:df:a4:2c:b4:a1:89:2a:e4:eb:12:b4:51:65:fc:
ae:73:9f:82:14:16:83:25:29:f5:42:58:ca:59:d3:
58:91:54:ac:bl:8b:c4:bl:7c:f0:e8:15:99:f6:c3:
4a:8e:94:96:59:3e:84:94:fb:a4:c4:69:ac:4e:80:
ab:b9:4c:3a:ad
prime2:
```



Step 2: Generate a Certificate Signing Request (CSR)

Once you has the key file, it should generates a Certificate Signing Request (CSR), The CSR will be sent to the CA, who will generate a certificate for the key

use SEEDPKILab2020.com as the common name of the certificate request.

```
Terminal [05/17/24]seed@VM:~/.../Abdulmajeed$ openssl req -new -key server.key -out server.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]:SA
State or Province Name (full name) [Some-State]:Riyadh
Locality Name (eg, city) []:Riyadh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:STC
Organizational Unit Name (eg, section) []:STCKSA
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2020.com
Email Address []:majeed.a.gh@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:abc123
An optional company name []:
[05/17/24]seed@VM:~/.../Abdulmajeed$
```

Step 3: Generating Certificates.

As the CA we will sign the CSR to generate X509 certificate

```
Terminal [05/17/24]seed@VM:~/.../Abdulmajeed$ 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: May 17 22:57:48 2024 GMT
        Not After : May 17 22:57:48 2025 GMT
    Subject:
        countryName            = SA
        stateOrProvinceName     = Riyadh
        organizationName        = STC
        organizationalUnitName  = STCKSA
        commonName              = SEEDPKILab2020.com
        emailAddress            = majeed.a.gh@gmail.com
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
            8B:19:1F:9E:8E:83:10:36:28:D8:C7:4A:13:57:92:6F:F1:22:88
        X509v3 Authority Key Identifier:
            Keyid:6E:25:A7:34:56:2E:7A:D6:10:0D:16:4C:B5:D9:4C:39:81:57:FE:68
Certificate is to be certified until May 17 22:57:48 2025 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]
Write out database with 1 new entries
Data Base Updated
[05/17/24]seed@VM:~/.../Abdulmajeed$
```



Task 3: Deploying Certificate in an HTTPS Web Server

Step 1: Configuring DNS

to map the hostname "SEEDPKILab2020.com" to localhost we will use this command

```
sudo nano /etc/hosts
```

then Add the following line , Save the file and exit

```
127.0.0.1      SEEDPKILab2020.com
```

```
Terminal
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
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
127.0.0.1      SEEDPKILab2020.com

-- INSERT --
19,29-35      All
```



Step 2: Configuring the web server

Combine the secret key and certificate into one file

Then Launch the web server using server.pem

The screenshot shows a terminal window on the left and a Firefox browser window on the right. The terminal window displays the following commands and output:

```
[05/17/24]seed@VM:~/.Abdulmajeed$ cp server.key server.pem
[05/17/24]seed@VM:~/.Abdulmajeed$ cat server.crt >> server.pem
[05/17/24]seed@VM:~/.Abdulmajeed$ openssl s_server -cert server.pem
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
ACCEPT
ACCEPT
```

The Firefox browser window shows the URL <https://seedpkilab2020.com:4433>. The page title is "Problem loading page - Mozilla Firefox". The main content area displays an error message: "Secure Connection Failed". It states: "An error occurred during a connection to seedpkilab2020.com:4433. Peer's certificate has an invalid signature. Error code: SEC_ERROR_BAD_SIGNATURE". It also includes a list of bullet points: "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.", and a link "Learn more...". There is a checkbox "Report errors like this to help Mozilla identify and block malicious sites" and a blue button "Try Again".

Now we can access the server using this URL "<https://seedpkilab2018.com:4433/>"

usually get the error message from the browser , to solve this problem we going to step 3

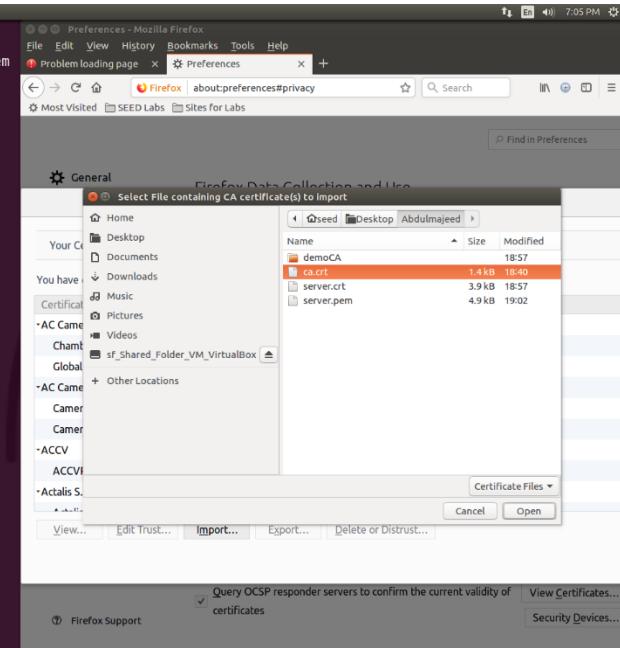


Step 3: Getting the browser to accept our CA certificate

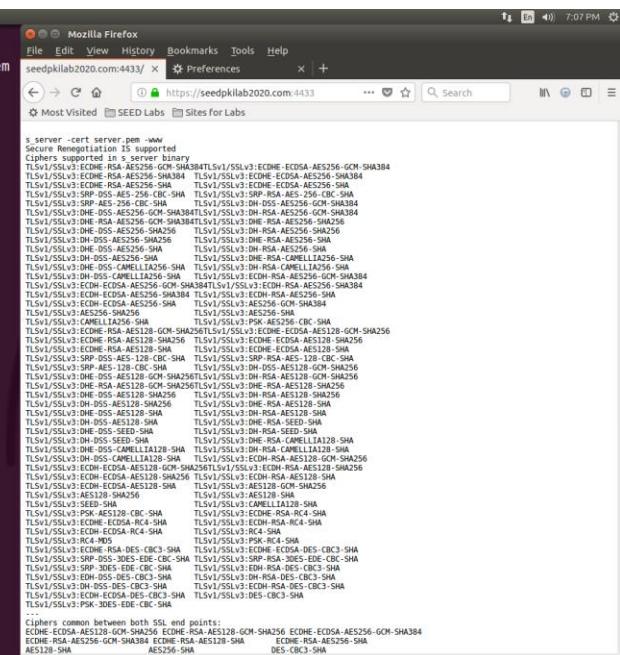
we need to add the CA to firefox manually

Edit -> Preferences -> Privacy & Security -> View Certificates

```
[Firefox Web Browser] [05/17/24]seed@VM:~/.Abdulmajed$ cp server.key server.pem  
[05/17/24]seed@VM:~/.Abdulmajed$ cat server.crt >> server.pem  
[05/17/24]seed@VM:~/.Abdulmajed$ openssl s_server -cert server.pem  
Enter pass phrase for server.pem:  
Using default temp DH parameters  
S ACCEPT
```



```
[05/17/24]seed@VM:~/.Abdulmajeed$ cp server.key server.pem  
[05/17/24]seed@VM:~/.Abdulmajeed$ cat server.crt >> server.pem  
[05/17/24]seed@VM:~/.Abdulmajeed$ openssl s_server -cert server.pem  
Enter pass phrase for server.pem:  
Using default temp DH parameters  
S  
ACCEPT  
ACCEPT
```





Step 4. Testing our HTTPS website.

Change the first byte of server.pem

```
Terminal: -----BEGIN RSA PRIVATE KEY-----  
Proc-Type: 4,ENCRYPTED  
DEK-Info: AES-128-CBC,347DF88CD222F90BCF37B0FB5F6DC640  
  
MjpwN7rMDyG6U0k5cEaehntokL1/vdvUXedgIra56NrXVQyqDkfQYIX8IUQgEf78Ht  
d/06ktACXEIX5oB3eLk791zA/ZTHBLbTR1YUxodH+5gTzn9wx6yJFp1p4qJBfxu  
JpMAX2subsmmsbYsNEvKvn0mk0E5e4B+fUob15Zm1NdA6CPhE4uuDBs90kV05  
C4fjn+OULApYfhfvTA9r1+J0Jq08FyZpTwEp5mq9nAafjtTh9Rrc1zfEHnGdZuE  
yvCdsdwRPEHqjnspIIIC9oHls617+iM0myruwoRlhga2vus7ic+rKanCvCFomDdr  
XzIgHxr++UjWz36Mzx5wVlt-QYc3V975AnVLHEvngyNYIQ4y3/lgCNJRYzy08  
bmqHuslCliey5+ewKnhq7zKzL1HKjV3sgnxvvg15rUX85ssxj7n6WPA40uYqZZ  
totKuOSYZ4RFVTCNREuMxEZ/PUG023yUs5zsfkUXZM6C7jxhqvTLVgWpmYpM4X  
/evWL74fLUo2z6PdauheXJE0AgI2PAIBzt5Rdnm0rt7KF03RX1DCC21vFSQk0  
hgm36xaIbLl6inJtA2Kwv5wsjq7J5r5KZd758mJNEp/yJq@0wah0d0mlNUN  
lQffFV+gFW7koZaDmq8hjuKmDc5ibjwgvcealcKH0Nj/45T287NfoT0+/rng/g  
d2sh5N+V5LzK9nBxs5z9G5zgVGALN2Fx0BsucyfWmaJ2EJ1Gfk4A7+SjWxu0L  
k+sG1B9hbIdmnfcUewDRh853tKkB8aELtj0s/b/SP3piSJh1a@PLSMWjyCfZPx  
-----END RSA PRIVATE KEY-----  
Certificate:  
Data:  
    Version: 3 (0x2)  
    Serial Number: 4096 (0x1000)  
    Signature Algorithm: sha256WithRSAEncryption  
    Issuer: C=SA, ST=Riyadh, L=Riyadh, O=STC, OU=STCKSA, CN=EE/emailAddress=majeed.a.gh@gmail.com  
    Validity  
        Not Before: May 17 22:57:48 2024 GMT  
        Not After : May 17 22:57:48 2025 GMT  
    Subject: C=SA, ST=Riyadh, O=STC, OU=STCKSA, CN=SEEDPKILab2020.com/emailAddress=majeed.a.gh@gmail.com  
    Subject Public Key Info:  
        Public Key Algorithm: rsaEncryption  
        Public-Key: (1024 bit)  
        Modulus:  
            00:b1:f9:bb:d5:81:eb:4a:ea:ca:c8:0:a:c1:a7:f3:  
            71:33:18:3f:a2:c8:09:77:21:cf:8c:e7:16:68:04:  
            84:cb:e4:0f:70:1b:e2:66:6a:84:dc:25:a2:e:0b:  
            1b:4a:86:2a:26:72:a8:f1:2a:f5:3d:fb:00:94:98:  
            74:d0:69:92:0b:38:6d:7f:f2:e2:b5:44:02:57:60:  
-----  
1,1 Top
```

Now reopen "<https://localhost:4433/>" and "<https://seedpkilab2020.com/>" :

```
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo vi server.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo vi server.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$ openssl s_server -cert server.pem  
-www  
Enter pass phrase for server.pem:  
unable to load server certificate private key file  
3070420672:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong  
tag:tasn_dec.c:1197:  
3070420672:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nest  
ed asn1 error:tasn_dec.c:374:Type:RSA  
3070420672:error:04093004:rsa routines:OLD_RSA_PRIV_DECODE:RSA lib:rsa  
ameth.c:119:  
3070420672:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong  
tag:tasn_dec.c:1197:  
3070420672:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nest  
ed asn1 error:tasn_dec.c:374:Type:PKCS8 PRIV KEY INFO  
3070420672:error:0907B00D:PEM routines:PEM_READ_BIO_PRIVATEKEY:ASN1 li  
b: pem_pkey.c:141:  
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo vi server.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$ ]
```



The screenshot illustrates a penetration testing scenario. On the left, a terminal window titled 'Firefox Web Browser' shows a user attempting to edit a server certificate (server.pem) and then opening it in a browser. The terminal output includes commands like 'sudo vi server.pem', 'openssl s_server -cert server.pem -www', and 'Enter pass phrase for server.pem'. On the right, a Mozilla Firefox window titled 'Problem loading page - Mozilla Firefox' shows an error message: 'Firefox can't establish a connection to the server at seedpkilab2020.com:4433.' Below the message, there are several troubleshooting tips:

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

A 'Try Again' button is visible at the bottom right of the Firefox window.

The reason is modifying a single byte in the server.pem file invalidates the certificate because it changes the hash and cryptographic signature and this will prevent the browser from verifying the certificate's integrity.



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

Create the apache2/ssl directory then move certificate files to it

```
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo cp server.crt /etc/apache2/ssl/seedpki_cert.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo cp server.key /etc/apache2/ssl/seedpki_key.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$ ls  
ca.crt demoCA server.crt server.key  
ca.key openssl.cnf server.csr server.pem  
[05/17/24]seed@VM:~/.../Abdulmajeed$
```

Edit the default-ssl.conf file to add a VirtualHost entry SEEDPKILab2020.com.

```
[05/17/24]seed@VM:~/.../apache2$ ls  
apache2.conf envvars mods-enabled sites-available  
conf-available magic ports.conf ssl  
conf-enabled mods-available sites-available  
[05/17/24]seed@VM:~/.../apache2$ cd ssl  
[05/17/24]seed@VM:~/.../ssl$ ls  
seedpki cert.pem seedpki_key.pem  
[05/17/24]seed@VM:~/.../ssl$ sudo nano /etc/apache2/sites-available/default-ssl.conf  
[05/17/24]seed@VM:~/.../ssl$ sudo nano /etc/apache2/sites-available/000-default.conf  
[05/17/24]seed@VM:~/.../ssl$ sudo apachectl configtest  
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist  
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message  
Syntax OK  
[05/17/24]seed@VM:~/.../ssl$ sudo service apache2 restart  
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): **  
[05/17/24]seed@VM:~/.../ssl$
```

>Edit the default-ssl.conf

```
<IfModule mod_ssl.c>  
<VirtualHost *:443>  
    ServerName SEEDPKILab2020.com  
    DocumentRoot /var/www/seedpki  
    DirectoryIndex index.html  
    SSLEngine On  
    SSLCertificateFile /etc/apache2/ssl/cert.pem  
    SSLCertificateKeyFile /etc/apache2/ssl/key.pem  
</VirtualHost>  
  
<VirtualHost _default_:443>  
    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  
    #
```

Add the VirtualHost



create simple HTML file for testing



```
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo cp server.crt /etc/apache2/s
sl/seedpki_cert.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo cp server.key /etc/apache2/s
sl/seedpki_key.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ ls
ca.crt demoCA server.crt server.key
ca.key openssl.cnf server.csr server.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo mkdir /var/www/seedpkilab202
0
[05/17/24]seed@VM:~/.../Abdulmajeed$ echo "<html><body><h1>Welcome to
SEEDPKILab2020.com</h1></body></html>" | sudo tee /var/www/seedpkilab2
020/index.html
<html><body><h1>Welcome to SEEDPKILab2020.com</h1></body></html>
```

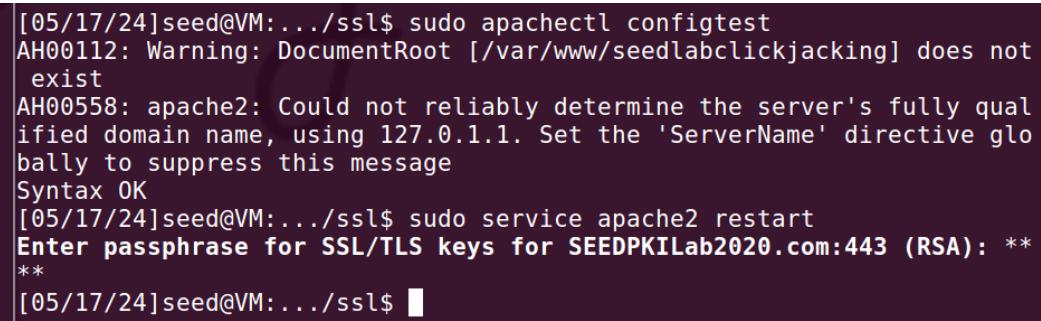
Enable SSL and the Site Configuration

```
// Test the Apache configuration file for errors
$ sudo apachectl configtest

// Enable the SSL module
$ sudo a2enmod ssl

// Enable the site we have just edited
$ sudo a2ensite default-ssl

// Restart Apache
$ sudo service apache2 restart
```



```
[05/17/24]seed@VM:.../ssl$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not
exist
AH00558: apache2: Could not reliably determine the server's fully qual
ified domain name, using 127.0.1.1. Set the 'ServerName' directive glo
bally to suppress this message
Syntax OK
[05/17/24]seed@VM:.../ssl$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): **
**
[05/17/24]seed@VM:.../ssl$
```



Navigate to <https://SEEDPKILab2020.com> and its work

The screenshot shows a Firefox browser window displaying the Apache2 Ubuntu Default Page. The page content includes a heading 'It works!', a 'Configuration Overview' section, and a file tree diagram of the Apache2 configuration directory. To the right of the browser is a terminal window showing command-line logs related to Apache2 configuration and testing.

Apache2 Ubuntu Default Page: It works - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Apache2 Ubuntu Default... X Problem loading page X +

Most Visited SEED Labs Sites for Labs

ubuntu

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.htm) 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`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed

```
[05/17/24]seed@VM:.../apache2$ sudo vi sites-available/default-ssl.conf
[05/17/24]seed@VM:.../apache2$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00526: Syntax error on line 7 of /etc/apache2/sites-enabled/default-ssl.conf:
SSLCertificateFile: file '/etc/apache2/ssl/cert.pem' does not exist or is empty
Action 'configtest' failed.
The Apache error log may have more information.
[05/17/24]seed@VM:.../apache2$ sudo service apache2 restart
Job for apache2.service failed because the control process exited with error code. See "systemctl status apache2.service" and "journalctl -xe" for details.
[05/17/24]seed@VM:.../apache2$ ls
apache2.conf  envvars      mods-enabled  sites-available
conf-available  magic      ports.conf    ssl
conf-enabled   mods-available sites-available
[05/17/24]seed@VM:.../apache2$ cd ssl
[05/17/24]seed@VM:.../ssl$ ls
seedpki cert.pem seedpki.key.pem
[05/17/24]seed@VM:.../ssl$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[05/17/24]seed@VM:.../ssl$ sudo nano /etc/apache2/sites-available/000-default.conf
[05/17/24]seed@VM:.../ssl$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[05/17/24]seed@VM:.../ssl$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): ***
[05/17/24]seed@VM:.../ssl$ 
```

in previous we cerate simple HTML file so if we change the Document Root in virtual Host to our file

The screenshot shows a Firefox browser window displaying the 'Welcome to SEEDPKILab2020.com' page. The page content is identical to the previous screenshot. To the right of the browser is a terminal window showing command-line logs related to Apache2 configuration and testing.

Firefox Web Browser

File Edit View History Bookmarks Tools Help

seedpkilab2020.com/ X +

Most Visited SEED Labs Sites for Labs

Welcome to SEEDPKILab2020.com

```
[05/17/24]seed@VM:.../ssl$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[05/17/24]seed@VM:.../ssl$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[05/17/24]seed@VM:.../ssl$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): ***
[05/17/24]seed@VM:.../ssl$ 
```



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

Step 1: Setting up the malicious website.

setting up the malicious website , by do same instructions from task 4 to set up a Virtual Host

Enter , but in ServerName we will choose anywebsite for example in my case I use youtube.com

The screenshot shows a Linux desktop environment with several windows open:

- A terminal window titled "Terminal" showing the command "GNU nano 2.5.3 File: .../sites-available/default-ssl.conf Modified". The content of the file is displayed, including configuration for virtual hosts for "seedpkilab2020.com" and "youtube.com".
- A Firefox browser window titled "Mozilla Firefox" showing the URL "seedpkilab2020.com". The page content "Welcome to SEEDPKILab2020.com" is visible.
- A screenshot of the terminal window, which is also part of the desktop environment, showing the same nano editor session and configuration details.



Step 2: Becoming the man in the middle

modify the victim /etc/hosts file to redirect request for the target website to our malicious

IP server use command 'sudo nano /etc/hosts' and Add the ip and malicious website we chose

```
Terminal Terminal 8:00 PM
GNU nano 2.5.3 File: /etc/hosts
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
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
127.0.0.1 SEEDPKILab2020.com
127.0.0.1 youtube.com

[05/17/24]seed@VM:.../ssl$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[05/17/24]seed@VM:.../ssl$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[05/17/24]seed@VM:.../ssl$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): ***
[05/17/24]seed@VM:.../ssl$ sudo nano /etc/apache2/sites-available/default-ssl.conf
[05/17/24]seed@VM:.../ssl$
```

Step 3: Browse the target website

Navigate to the target website its will redirect to our malicious server

```

Terminal
GNU nano 2.5.3 File: .../sites-available/default-ssl.conf

<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName SEEDPKILab2020.com
    DocumentRoot /var/www/seedpkilab2020
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/seedpki_cert.pem
    SSLCertificateKeyFile /etc/apache2/ssl/seedpki_key.pem
</VirtualHost>

<VirtualHost *:443>
    ServerName youtube.com
    DocumentRoot /var/www/seedpkilab2020
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/seedpki_cert.pem
    SSLCertificateKeyFile /etc/apache2/ssl/seedpki_key.pem
</VirtualHost>

<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

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

```

Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell

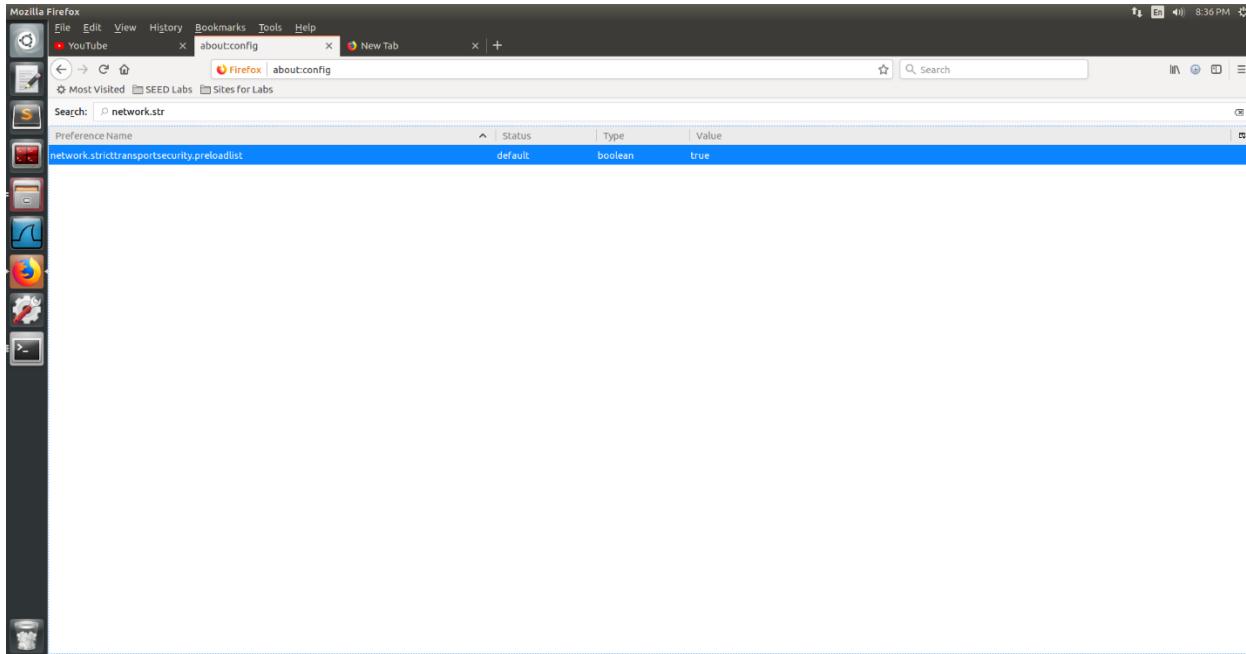
In some case we will facing some issue whenever navigate to youtube.com



So I search for the reason and I found this :

"When you navigate to youtube.com , it automatically redirects to <https://youtube.com> because YouTube, like many modern websites, enforces HTTPS for all connections to ensure secure communication between the user and the website. This is achieved through HTTP Strict Transport Security (HSTS), which instructs browsers to only interact with the site over a secure connection. "

I disable the HSTS and its now work correctly





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

generate a certificate using a Certificate Signing Request (CSR) file named "instagram.csr" and compromised CA private key to sign the CSR and generate the certificate

```
[05/17/24]seed@VM:~/.../Abdulmajeed$ openssl req -new -key server.key -out instagram.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]:SA
State or Province Name (full name) [Some-State]:Riyadh
Locality Name (eg, city) []:Riyadh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:STC
Organizational Unit Name (eg, section) []:STCKSA
Common Name (e.g. server FQDN or YOUR name) []:instagram.com
Email Address []:majeed.a.gh@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:abc123
An optional company name []:
[05/17/24]seed@VM:~/.../Abdulmajeed$ openssl ca -in instagram.csr -out
instagram.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: 4097 (0x1001)
    Validity
        Not Before: May 18 01:47:17 2024 GMT
        Not After : May 18 01:47:17 2025 GMT
    Subject:
        countryName      = SA
```

combine the secret key and certificate

```
[05/17/24]seed@VM:~/.../Abdulmajeed$ cp server.key instagram.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ cat instagram.crt >> instagram.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ cp instagram.crt cert2.pem
[05/17/24]seed@VM:~/.../Abdulmajeed$ sudo mv cert2.pem /etc/apache2/ssl/
[05/17/24]seed@VM:~/.../Abdulmajeed$ [05/17/24]seed@VM:~/.../ssl$ ls
cert2.pem  seedpki_cert.pem  seedpki_key.pem
[05/17/24]seed@VM:~/.../ssl$
```



Configuring DNS as we did before , and in the Virtualhost we add the website (notice SSLCertificateFile)
the path navigation to new cert2 we create for Instagram

The image shows two terminal windows. The left terminal displays the Apache configuration file /etc/apache2/sites-available/000-default.conf, which includes sections for youtube.com, instagram.com, and default sites. The right terminal shows the output of the Nmap command, listing various ports and services for the IP address 127.0.0.1.

```
[05/17/24]seed@VM:~/.Abdulmajed$ cp server.key instagram.pem
[05/17/24]seed@VM:~/.Abdulmajed$ cat instagram.crt >> instagram.pem
m
[05/17/24]seed@VM:~/.Abdulmajed$ cp instagram.crt cert2.pem
[05/17/24]seed@VM:~/.Abdulmajed$ sudo mv cert2.pem /etc/apache2/ssl/
l
[05/17/24]seed@VM:~/.Abdulmajed$ 
```

```
<VirtualHost *:443>
    ServerName youtube.com
    DocumentRoot /var/www/seedpkilab2020
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/seedpki_cert.pem
    SSLCertificateKeyFile /etc/apache2/ssl/seedpki_key.pem
</VirtualHost>

<VirtualHost *:443>
    ServerName instagram.com
    DocumentRoot /var/www/seedpkilab2020
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/cert2.pem
    SSLCertificateKeyFile /etc/apache2/ssl/seedpki_key.pem
</VirtualHost>

<VirtualHost *:443>
    ServerName SEEDPKILab2020.com
    DocumentRoot /var/www/seedpkilab2020
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/seedpki_cert.pem
    SSLCertificateKeyFile /etc/apache2/ssl/seedpki_key.pem
</VirtualHost>

<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
-- INSERT --
17,46 Top
```

navigate in the browser Instagram.com

Firefox Web Browser

[05/17/24]seed@VM:~/.Abdulmajeed\$ cp server.key instagram.pem
[05/17/24]seed@VM:~/.Abdulmajeed\$ cat instagram.crt >> instagram.pem
m
[05/17/24]seed@VM:~/.Abdulmajeed\$ cp instagram.crt cert2.pem
[05/17/24]seed@VM:~/.Abdulmajeed\$ sudo mv cert2.pem /etc/apache2/ssl
l
[05/17/24]seed@VM:~/.Abdulmajeed\$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for SEEDPKILab2020.com:443 (RSA): **
**
[05/17/24]seed@VM:~/.Abdulmajeed\$

[05/17/24]seed@VM:~\$ sudo vi /etc/hosts
[05/17/24]seed@VM:~\$