



ETHICAL HACKING V2 LAB SERIES

Lab 17: Creating and Installing SSL Certificates

Document Version: **2020-08-24**

Material in this Lab Aligns to the Following	
Books/Certifications	Chapters/Modules/Objectives
All-In-One CEH Chapters ISBN-13: 978-1260454550	11: Cryptography 101
EC-Council CEH v10 Domain Modules	20: Cryptography
CompTIA Pentest+ Objectives	2.1: Given a scenario, conduct information gathering using appropriate techniques
CompTIA All-In-One PenTest+ Chapters ISBN-13: 978-1260135947	4: Vulnerability Scanning and Analysis 5: Mobile Device and Application Testing

Contents

Introduction	3
Objective	3
Pod Topology	4
Lab Settings	5
1 Creating a Self-Signed Certificate	6
2 Configuring the Apache SSL File.....	8
3 Testing the SSL Certificate	9

Introduction

SSL (Secure Socket Layer) is used to secure communication between a client and a web server throughout the internet. This lab demonstrates how to create a self-signed X.509 certificate and install it into a Linux web server.

Objective

In this lab, you will be conducting ethical hacking practices using various tools. You will be performing the following tasks:

1. Creating a Self-Signed Certificate
2. Configuring the Apache SSL File
3. Testing the SSL Certificate

Pod Topology



Kali

Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Kali Linux	192.168.9.2 192.168.0.2	root	toor

1 Creating a Self-Signed Certificate

1. Click on the **Kali** tab.
2. Click within the console window and press **Enter** to display the login prompt.
3. Enter **root** as the *username*. Press **Tab**.
4. Enter **toor** as the *password*. Click **Log In**.
5. Open a new terminal by clicking on the **Terminal** icon located at the top of the page, if the terminal is not already opened.
6. In the new *Terminal* window, type the command below to generate an SSL key. Press **Enter**.

```
openssl genrsa -out ca.key 2048
```

```
root@kali:~# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
root@kali:~#
```

7. Generate a new *Certificate Signing Request (CSR)*. Type the command below, followed by pressing the **Enter** key.

```
openssl req -new -key ca.key -out ca.csr
```

```
root@kali:~# openssl req -new -key ca.key -out ca.csr
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]:
```

- a. When prompted for *Country Name*, type **us**. Press **Enter**.
- b. *State*, type **North Carolina**. Press **Enter**.
- c. *Locality Name*, type **Raleigh**. Press **Enter**.
- d. *Organization Name*, type **xyz**. Press **Enter**.
- e. *Organization Unit Name*, leave blank. Press **Enter**.
- f. *Common Name*, type **sally**. Press **Enter**.
- g. *Email Address*, type **sally@xyz.corp**. Press **Enter**.
- h. *Challenge Password*, type **sally**. Press **Enter**.
- i. *Company Name*, leave blank. Press **Enter**.

```
root@kali:~# openssl req -new -key ca.key -out ca.csr
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]:North Carolina
Locality Name (eg, city) []:Raleigh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:XYZ
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Sally
Email Address []:sally@xyz.corp

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

8. Generate a self-signed key by entering the command below.

```
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

```
root@kali:~# openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = XYZ, CN = Sally, emailAddress = sally@x
yz.corp
Getting Private key
root@kali:~#
```

9. Copy the **ca.crt** file from the current directory to the **/etc/ssl/certs/** directory.

```
cp ca.crt /etc/ssl/certs/ca.crt
```

```
root@kali:~# cp ca.crt /etc/ssl/certs/ca.crt
root@kali:~#
```

10. Copy the **ca.key** file from the current directory to the **/etc/ssl/private,** directory.

```
cp ca.key /etc/ssl/private/ca.key
```

```
root@kali:~# cp ca.key /etc/ssl/private/ca.key
root@kali:~#
```

11. Copy the **ca.csr** file from the current directory to the **/etc/ssl/private/** directory.

```
cp ca.csr /etc/ssl/private/ca.csr
```

```
root@kali:~# cp ca.csr /etc/ssl/private/ca.csr
root@kali:~#
```

2 Configuring the Apache SSL File

1. Using the *Terminal*, navigate to the `/etc/apache2/sites-available/` directory.

```
cd /etc/apache2/sites-available
```

2. Copy the **default-ssl.conf** file.

```
cp default-ssl.conf localhost-ssl.conf
```

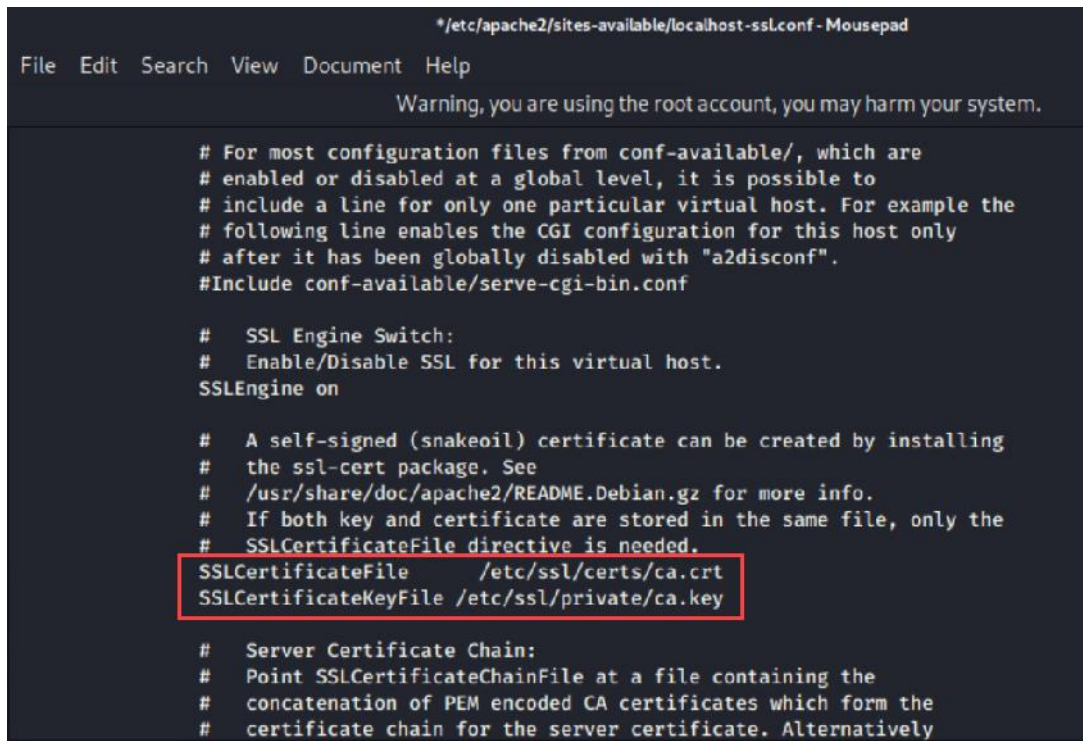
```
root@kali:/etc/apache2/sites-available# cp default-ssl.conf localhost-ssl.conf
root@kali:/etc/apache2/sites-available#
```

3. Edit the **localhost-ssl.conf** file using the *Mousepad* file editor.

```
mousepad localhost-ssl.conf
```

4. The Mousepad window appears. Change the **SSLCertificateFile** and **SSLCertificateKeyFile** paths.

```
SSLCertificateFile /etc/ssl/certs/ca.crt
SSLCertificateKeyFile /etc/ssl/private/ca.key
```



```
*/etc/apache2/sites-available/localhost-ssl.conf - Mousepad
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

# 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

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ca.crt
SSLCertificateKeyFile /etc/ssl/private/ca.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
```

5. Once modified, select **File > Save**.
6. Close the **Mousepad** window.

3 Testing the SSL Certificate

1. Using the *Terminal*, enter the following command to change to the sites-enabled directory:

```
cd /etc/apache2/sites-enabled/
```

```
root@kali:/etc/apache2/sites-available# cd /etc/apache2/sites-enabled/
root@kali:/etc/apache2/sites-enabled#
```

2. Enter the following command to enable the localhost-ssl site:

```
ln -s /etc/apache2/sites-available/localhost-ssl.conf
```

3. Enter the following command to enable the SSL Apache Mod:

```
a2enmod ssl
```

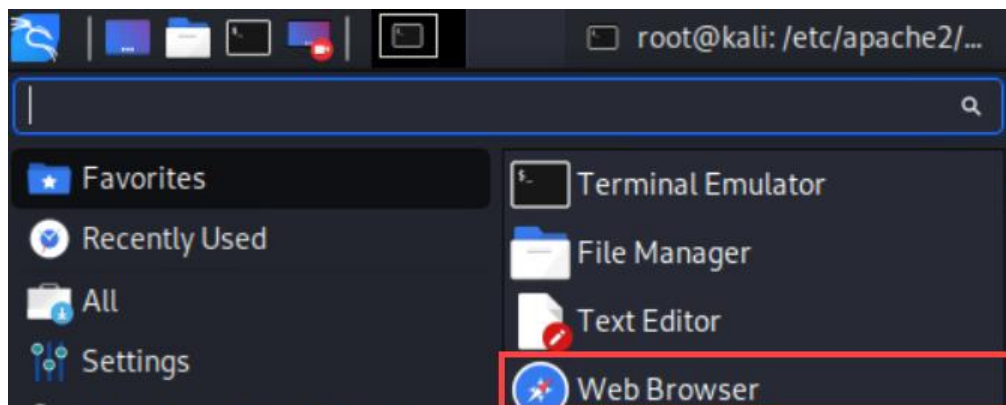
```
root@kali:/etc/apache2/sites-enabled# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@kali:/etc/apache2/sites-enabled#
```

4. Restart the Apache service.

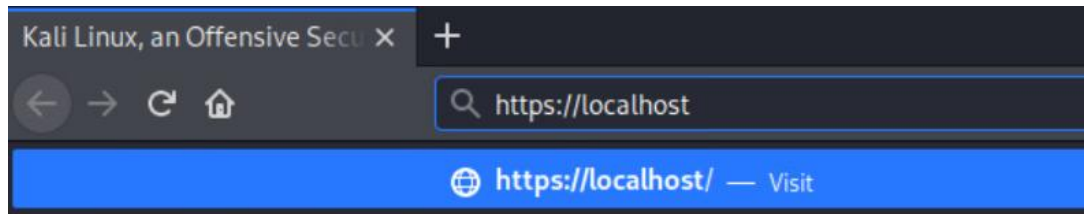
```
service apache2 restart
```

```
root@kali:/etc/apache2/sites-enabled# service apache2 restart
root@kali:/etc/apache2/sites-enabled#
```

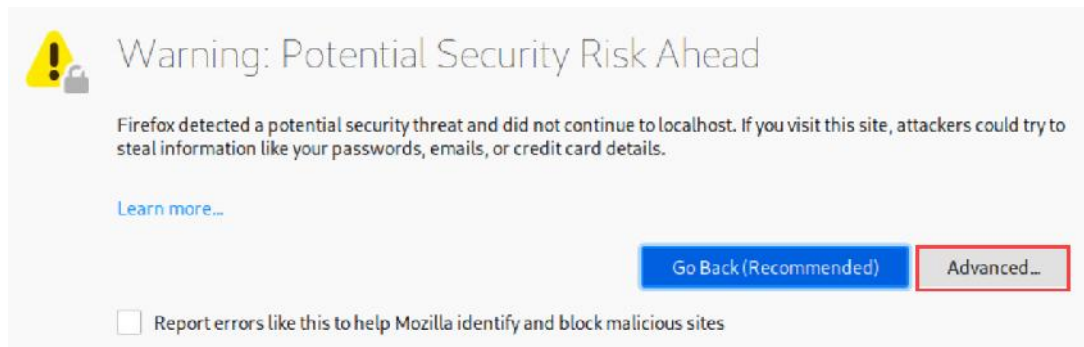
5. Open the *Mozilla Firefox* browser by clicking on the **Application Menu > Web Browser** from the top.



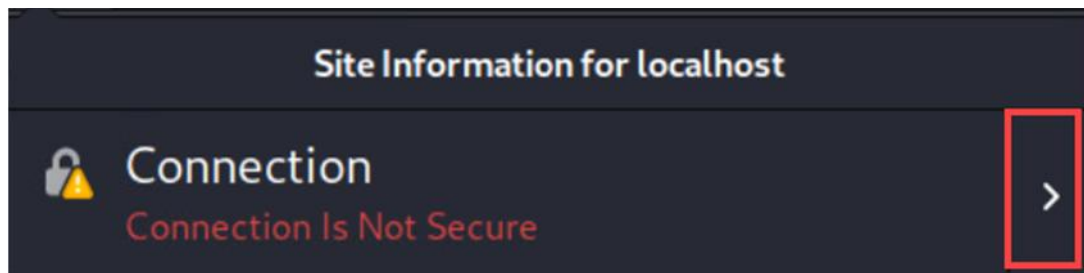
6. In the *address* field located towards the top, type `https://localhost` and press the **Enter** key.



7. Notice the warning message. Click on **Advanced**.

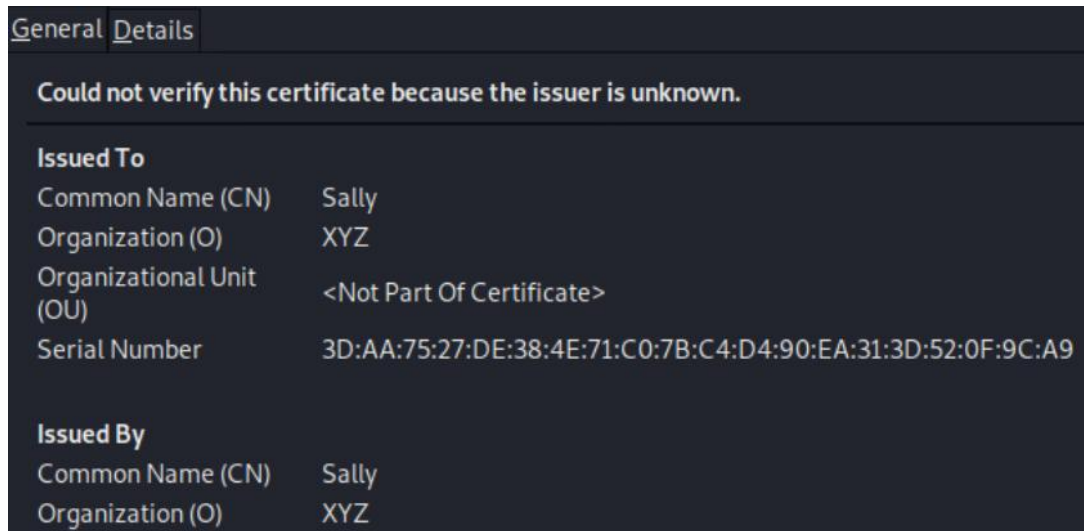


8. Scroll down and click **Accept the Risk and Continue**.
9. Once the page finishes loading, notice the padlock icon in the *address* field. A self-signed SSL certificate has been successfully implemented. Click on the **padlock** icon in the address bar.
10. Click on the arrow next to Connection.



11. Click on **More Information**.
12. In the page info popup window, click on **View Certificate**.

13. Notice the information on the General tab matches the information you entered when creating the certificate.



14. You may now end your reservation.