



ETHICAL HACKING LAB SERIES

Lab 6: Creating and Installing SSL Certificates

Material in this Lab Aligns to the Following Certification Domains/Objectives	
Certified Ethical Hacking (CEH) Domains	SANS GPEN Objectives
18: Cryptography	18: Wireless Crypto and Client Attacks

Document Version: 2016-03-09

Copyright © 2016 Network Development Group, Inc.
www.netdevgroup.com

NETLAB Academy Edition, NETLAB Professional Edition, and NETLAB+ are registered trademarks of Network Development Group, Inc.

VMware is a registered trademark of VMware, Inc. Cisco, IOS, Cisco IOS, Networking Academy, CCNA, and CCNP are registered trademarks of Cisco Systems, Inc. EMC² is a registered trademark of EMC Corporation.

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

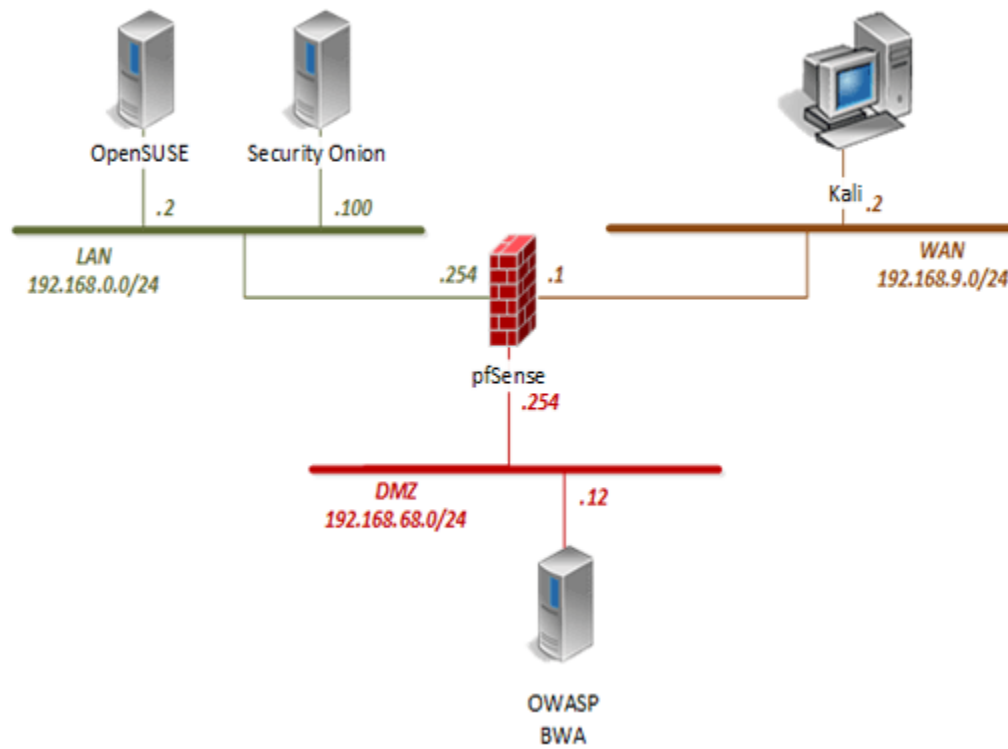
SSL (Secure Socket Layer) is used to secure communication between a client and web server throughout the internet. This lab demonstrates how to create a self-signed X.509 certificate and install it into a Linux web server.

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



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	root	toor
pfSense	192.168.0.254 192.168.68.254 192.168.9.1	admin	pfsense
OWASP Broken Web App	192.168.68.12	root	owaspbwa
OpenSUSE	192.168.0.2	osboxes	osboxes.org
Security Onion	192.168.0.100	ndg	password123

1 Creating a Self-Signed Certificate

1. Click on the **Kali** graphic on the *topology page*.
2. Click anywhere within the *Kali* console window and press **Enter** to display the login prompt.
3. Enter `root` as the *username*. Click **Next**.
4. Enter `toor` as the *password*. Click **Sign In**.
5. Open the *Terminal* by clicking on the **Terminal** icon located on the left panel.



6. In the new *Terminal* window, type the command below to generate a *SSL* key. Press **Enter**.

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

```
root@Kali2:~# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
```

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
```

- 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@Kali2:~# 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 []:
```



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
```

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

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

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

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

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

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

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.

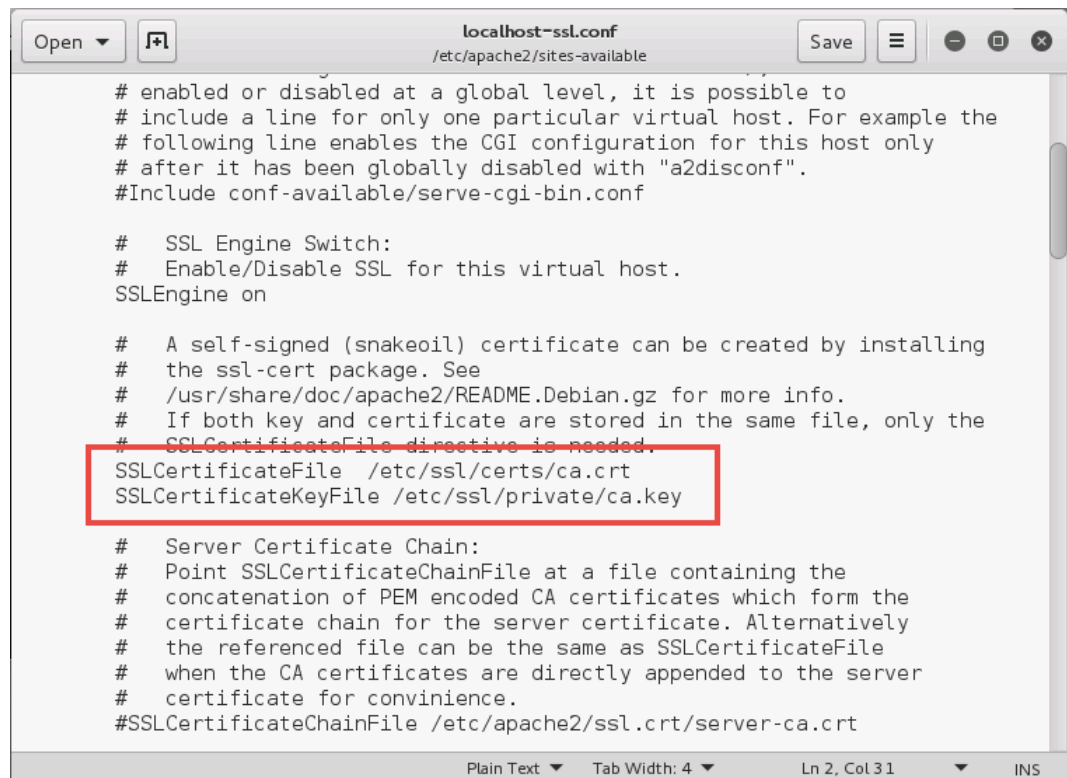
```
cp default-ssl.conf localhost-ssl.conf
```

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

```
gedit localhost-ssl.conf
```

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

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



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

# 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
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt
```

5. Once modified, click the **Save** button located towards to the top-right corner of the *Gedit* window.
6. Close the **Gedit** window.

3 Testing the SSL Certificate

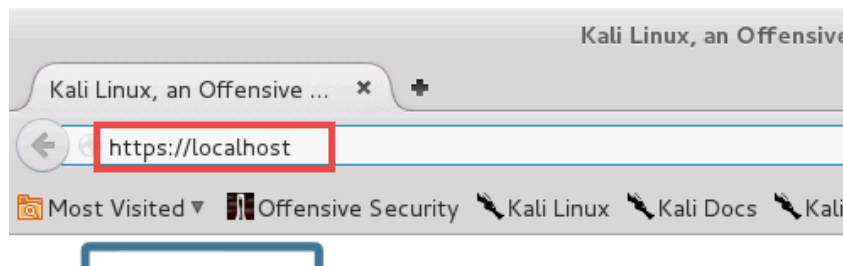
1. Using the Terminal, restart the Apache service.

```
service apache2 restart
```

2. Open the *Iceweasel* browser by clicking on the **Iceweasel** icon located on the left panel.

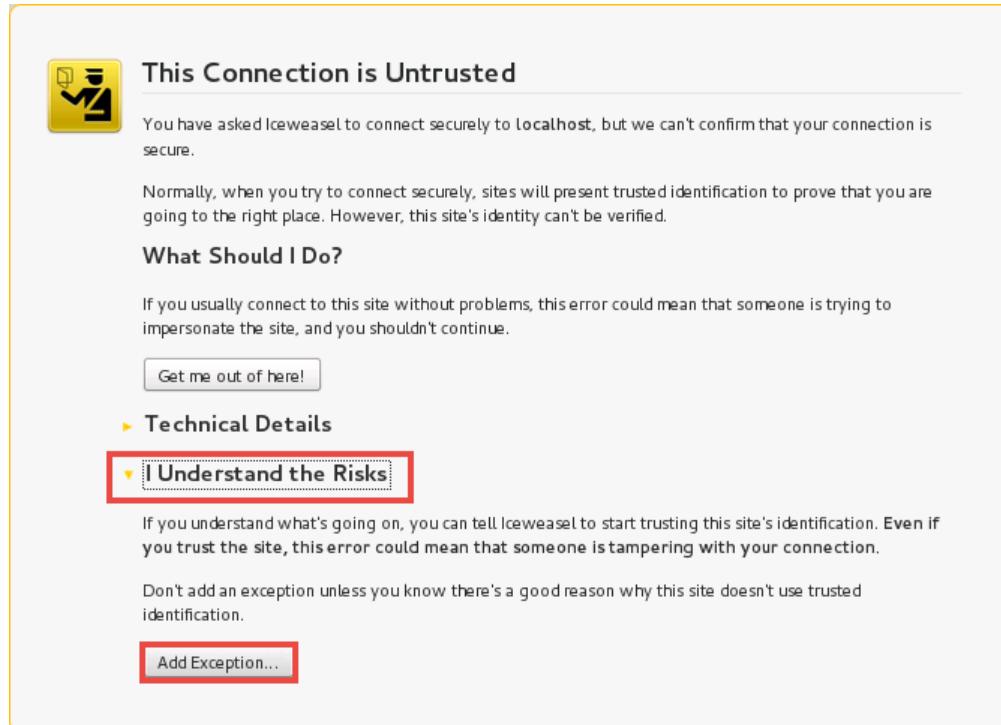


3. In the address field located towards the top, type `https://localhost` and press the **Enter** key.



4. Notice the warning message. Click on **I Understand the Risks**.

5. Click the **Add Exception** button.



6. On the *Add Security Exception* window, click **Confirm Security Exception**.



7. Once the page finishes loading, notice the lock icon in the address field. A self-signed SSL certificate has been successfully implemented.
8. Close the **Kali** PC viewer.