Project Report
Part -2

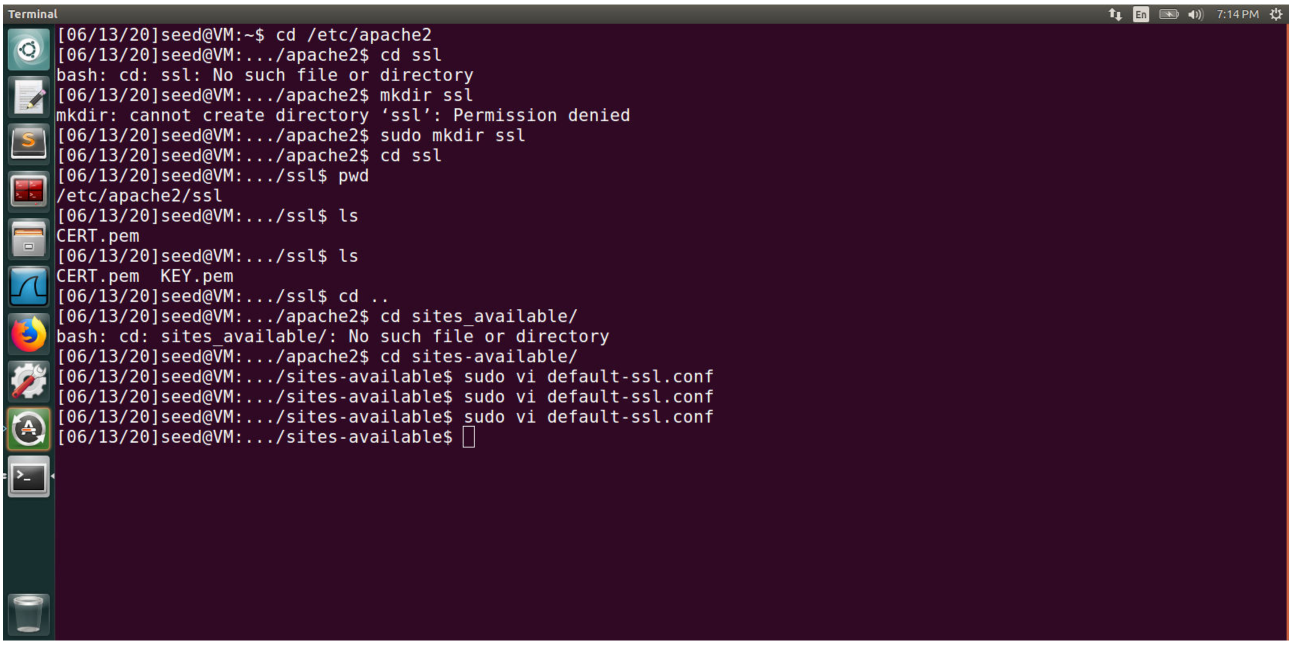
Objective: To Understand PKI and launching a Man in the Middle Attack.

Lab Environment: Ubuntu 16.04 vm downloaded from SEED website.

Library and commands used : OpenSSL

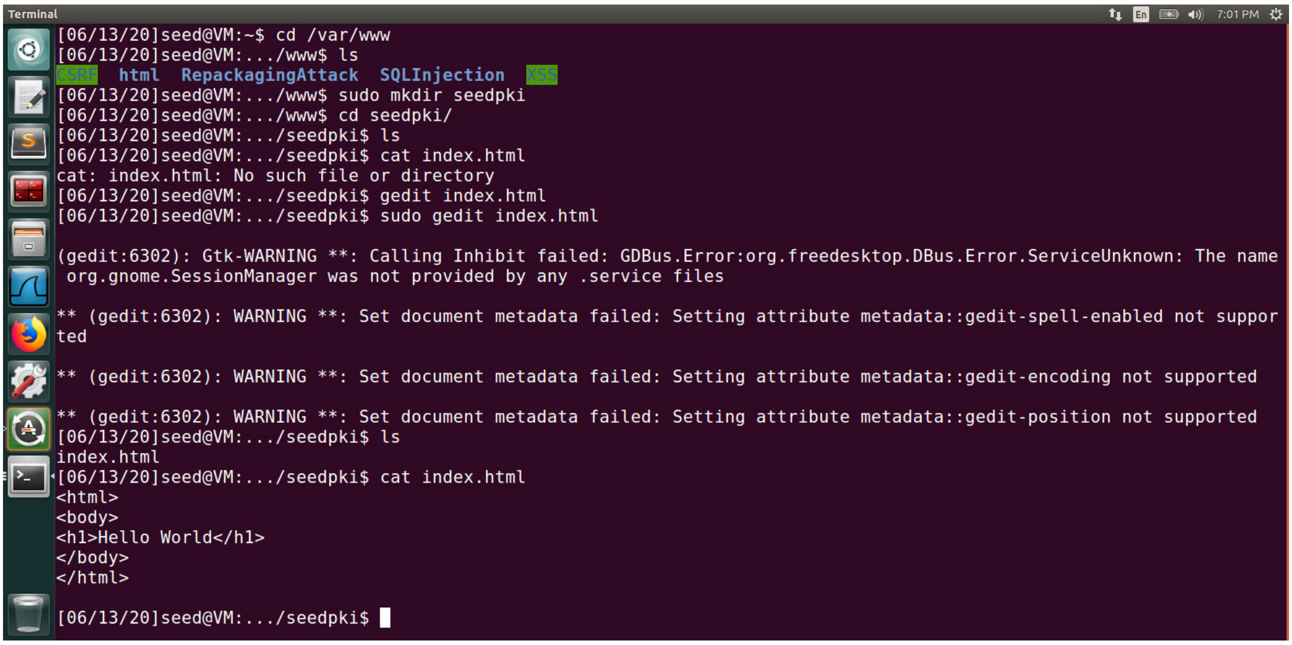
Screenshots of 2nd and 3rd Terminal:

Terminal one had many commands so i recorded the screen for it using the Free screen recorder (published by thundershare.net) so it has its water mark. The .mp4 file is Terminal-1 recording.
Terminal-2



```
[06/13/20]seed@VM:~$ cd /etc/apache2
[06/13/20]seed@VM:.../apache2$ cd ssl
bash: cd: ssl: No such file or directory
[06/13/20]seed@VM:.../apache2$ mkdir ssl
mkdir: cannot create directory ‘ssl’: Permission denied
[06/13/20]seed@VM:.../apache2$ sudo mkdir ssl
[06/13/20]seed@VM:.../apache2$ cd ssl
[06/13/20]seed@VM:.../ssl$ pwd
/etc/apache2/ssl
[06/13/20]seed@VM:.../ssl$ ls
CERT.pem
[06/13/20]seed@VM:.../ssl$ ls
CERT.pem  KEY.pem
[06/13/20]seed@VM:.../ssl$ cd ..
[06/13/20]seed@VM:.../apache2$ cd sites_available/
bash: cd: sites_available/: No such file or directory
[06/13/20]seed@VM:.../apache2$ cd sites-available/
[06/13/20]seed@VM:.../sites-available$ sudo vi default-ssl.conf
[06/13/20]seed@VM:.../sites-available$ sudo vi default-ssl.conf
[06/13/20]seed@VM:.../sites-available$ sudo vi default-ssl.conf
[06/13/20]seed@VM:.../sites-available$ 
```

Terminal-3



```
[06/13/20]seed@VM:~$ cd /var/www
[06/13/20]seed@VM:.../www$ ls
CSRF  html  RepackagingAttack  SQLInjection  XSS
[06/13/20]seed@VM:.../www$ sudo mkdir seedpki
[06/13/20]seed@VM:.../www$ cd seedpki/
[06/13/20]seed@VM:.../seedpki$ ls
[06/13/20]seed@VM:.../seedpki$ cat index.html
cat: index.html: No such file or directory
[06/13/20]seed@VM:.../seedpki$ gedit index.html
[06/13/20]seed@VM:.../seedpki$ sudo gedit index.html

(gedit:6302): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name
 org.gnome.SessionManager was not provided by any .service files

** (gedit:6302): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not suppor
ted

** (gedit:6302): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:6302): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[06/13/20]seed@VM:.../seedpki$ ls
index.html
[06/13/20]seed@VM:.../seedpki$ cat index.html
<html>
<body>
<h1>Hello World</h1>
</body>
</html>

[06/13/20]seed@VM:.../seedpki$ 
```

Procedure/Tasks and Observation :

Task-4:

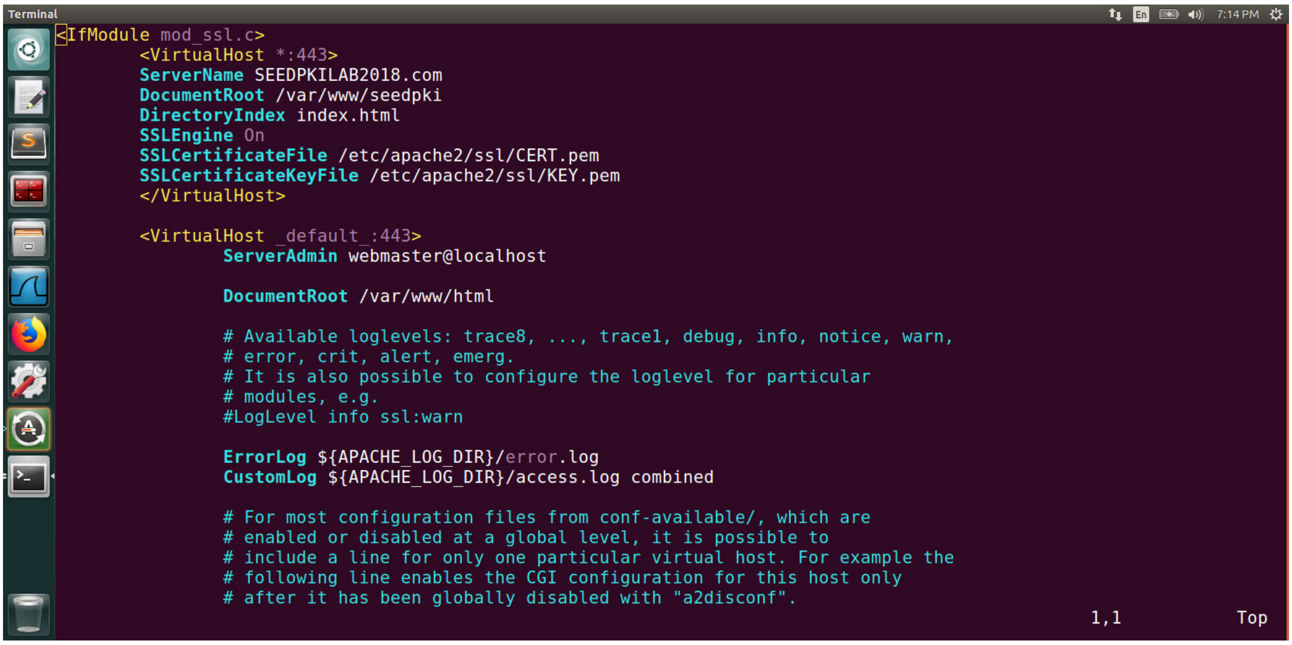
Deploying Certificate in an Apache-Based HTTPS Website:

The HTTPS server setup using openssl’s s server command is primarily for debugging and demonstration purposes. So, I set up a real HTTPS web server based on Apache which is preinstalled in the VM.

To create an HTTPS website, I just need to configure the Apache server, so it knows where to get the private key and certificates.

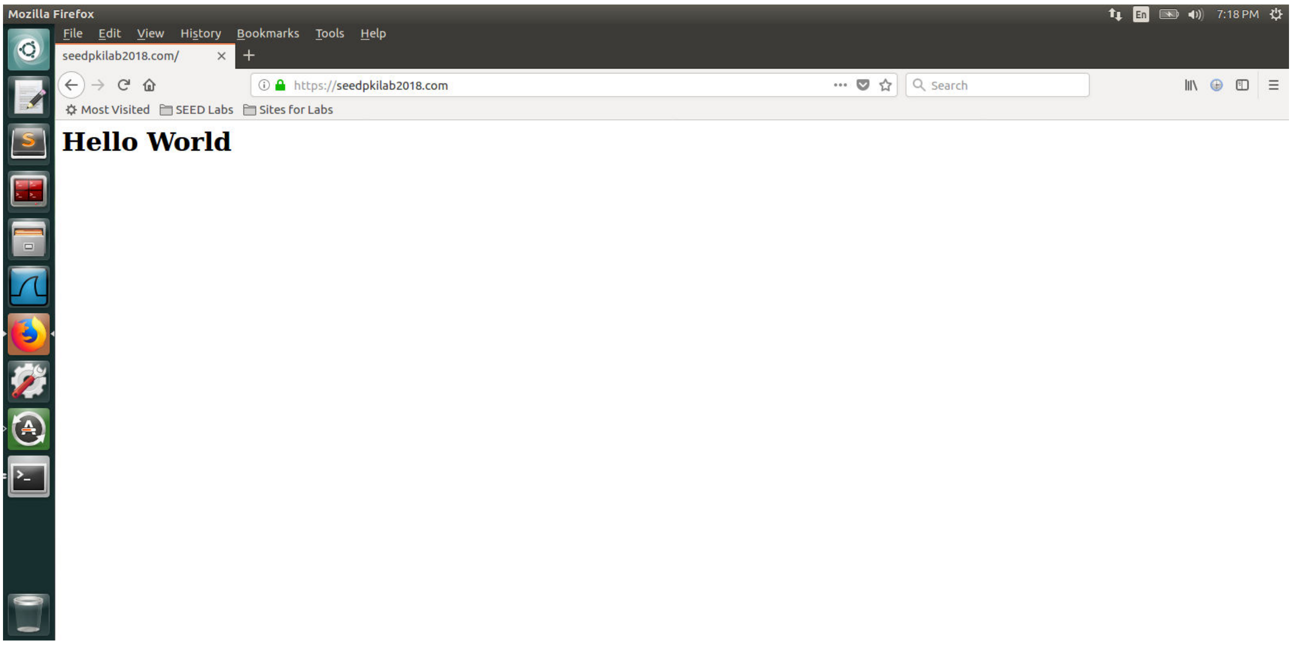
An Apache server can simultaneously host multiple websites. It needs to know the directory where a website’s files are stored. This is done via its VirtualHost file, located in the “/etc/apache2/sites-available” directory. To add an HTTPS website, I add a VirtualHost entry to the file “000-default.conf.”

To add an HTTPS website, I need to add a VirtualHost entry to the “default-ssl.conf” file in the same folder.(I use vim editor to do so)



```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
        ServerName SEEDPKILAB2018.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
                # after it has been globally disabled with "a2disconf".
                                                                      1,1           Top
```

After this I ran a series of commands to enable SSL:
(See the recording of Terminal 1 the .mp4 file)
(i) sudo apachectl configtest
(Test the Apache configuration file for errors)
(ii) sudo a2enmod ssl
(Enable the SSL module)
(iii) sudo a2ensite default-ssl
(Enable the site I have just edited)
(iv) sudo service apache2 restart
(Restart Apache)

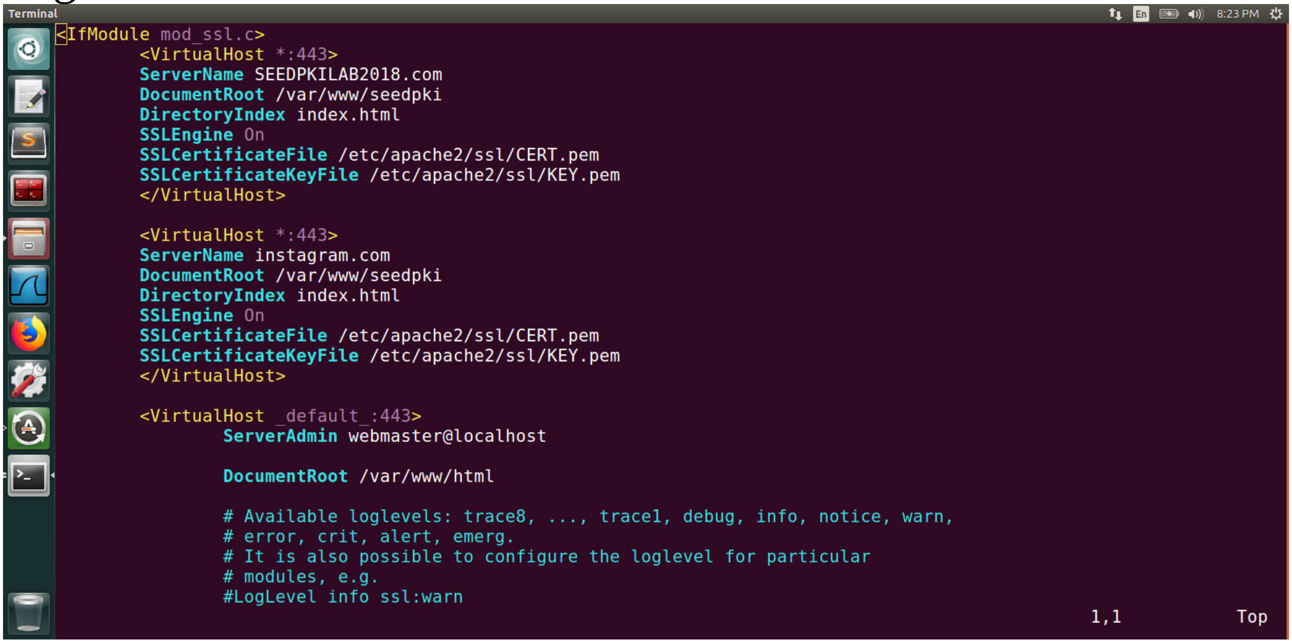


Hello World

Task-5:

Launching a Man-In-The-Middle Attack:

1) I will use the https website created in Task 4 to be the fake website where the user will land, I are using instagram.com as the target website.

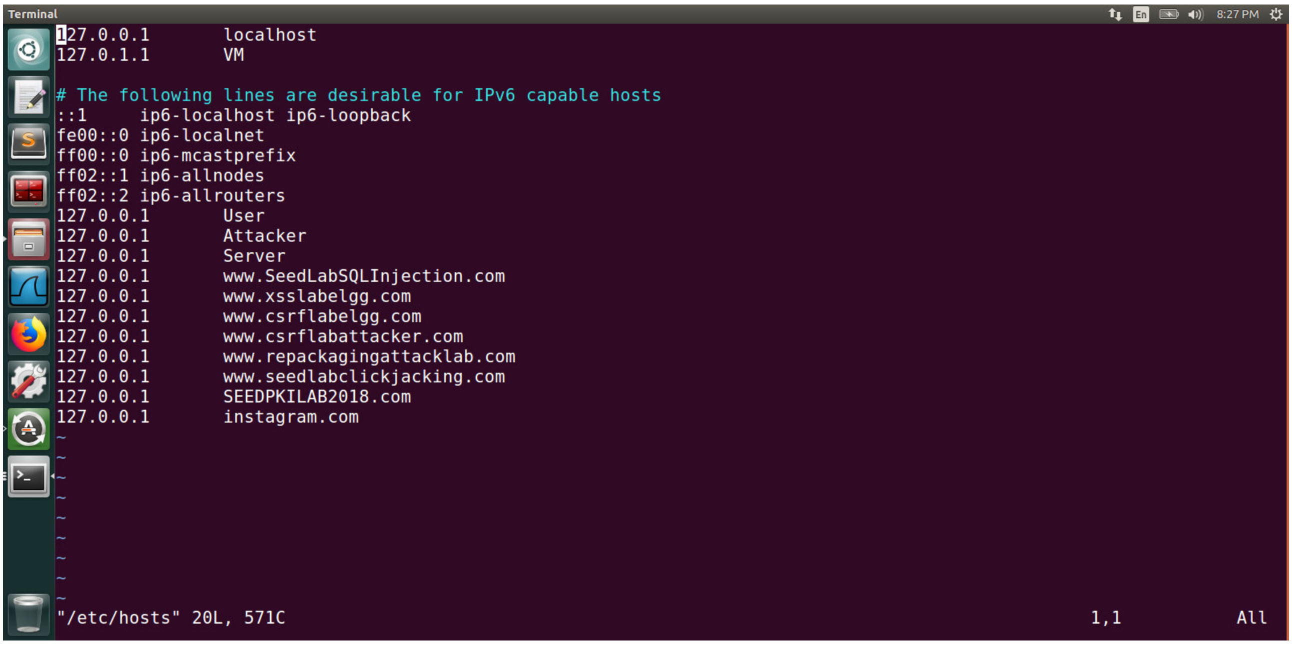


```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
        ServerName SEEDPKILAB2018.com
        DocumentRoot /var/www/seedpki
        DirectoryIndex index.html
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/CERT.pem
        SSLCertificateKeyFile /etc/apache2/ssl/KEY.pem
        </VirtualHost>

        <VirtualHost *:443>
        ServerName instagram.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
                                                                        1,1           Top
```

I only change the servername to instagram.com while the rest of the configurations are the same.

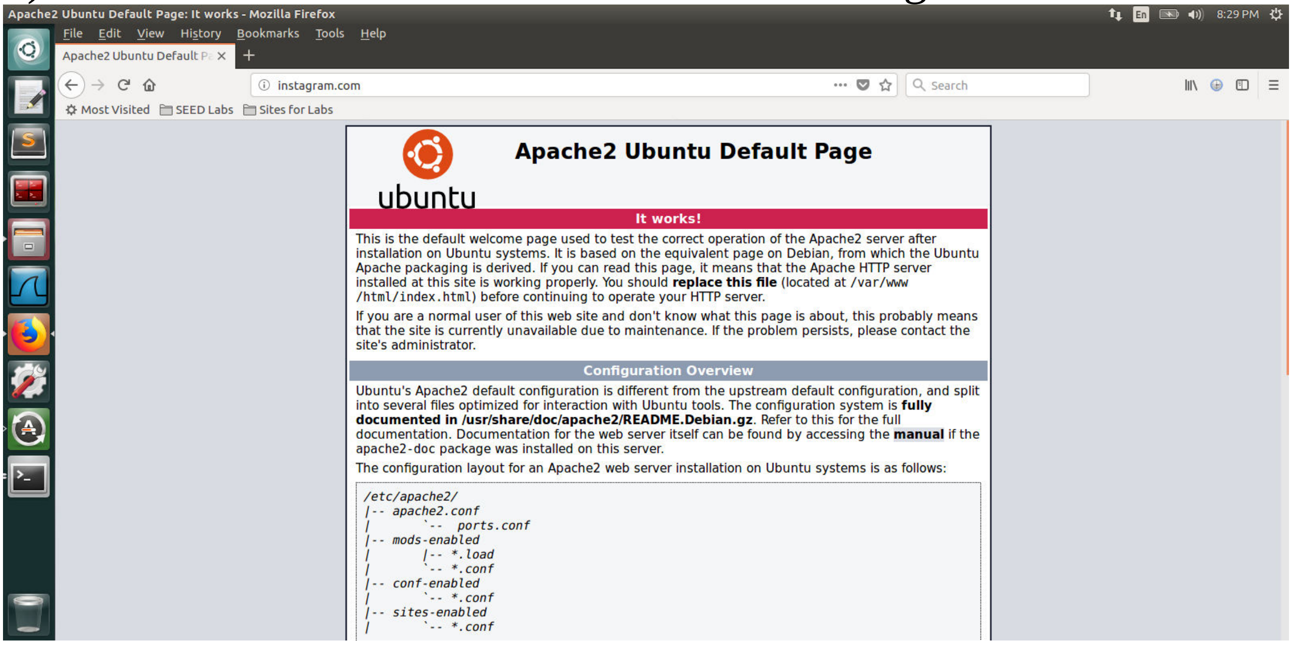
2) I now become the man in the middle by the “attack DNS” approach I simply modify the victim’s machine’s “/etc/hosts” file

to emulate the result of a DNS cache positing attack (the IP Address in the following should be replaced by the actual IP address of the malicious server).



```
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       SEEDPKILAB2018.com
127.0.0.1       instagram.com
~
~
~
~
~
~
~
~
~
"/etc/hosts" 20L, 571C                                        1,1           All
```

3) I now launch the website to see the following result :



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:

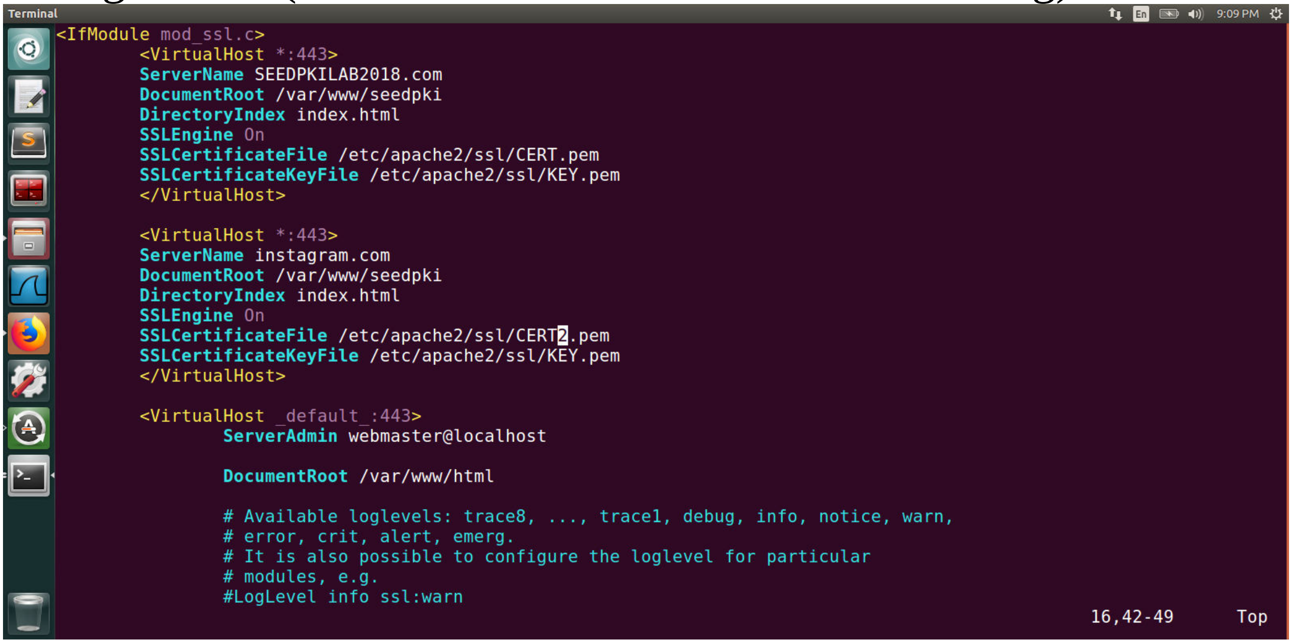
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

This is the default webpage of apache2 server and hence it raises concerns for the user as it is not the website they intended to open. Therefore, if the CA is not compromised I can at least be aware of the wrong website which opens.

Task-6:

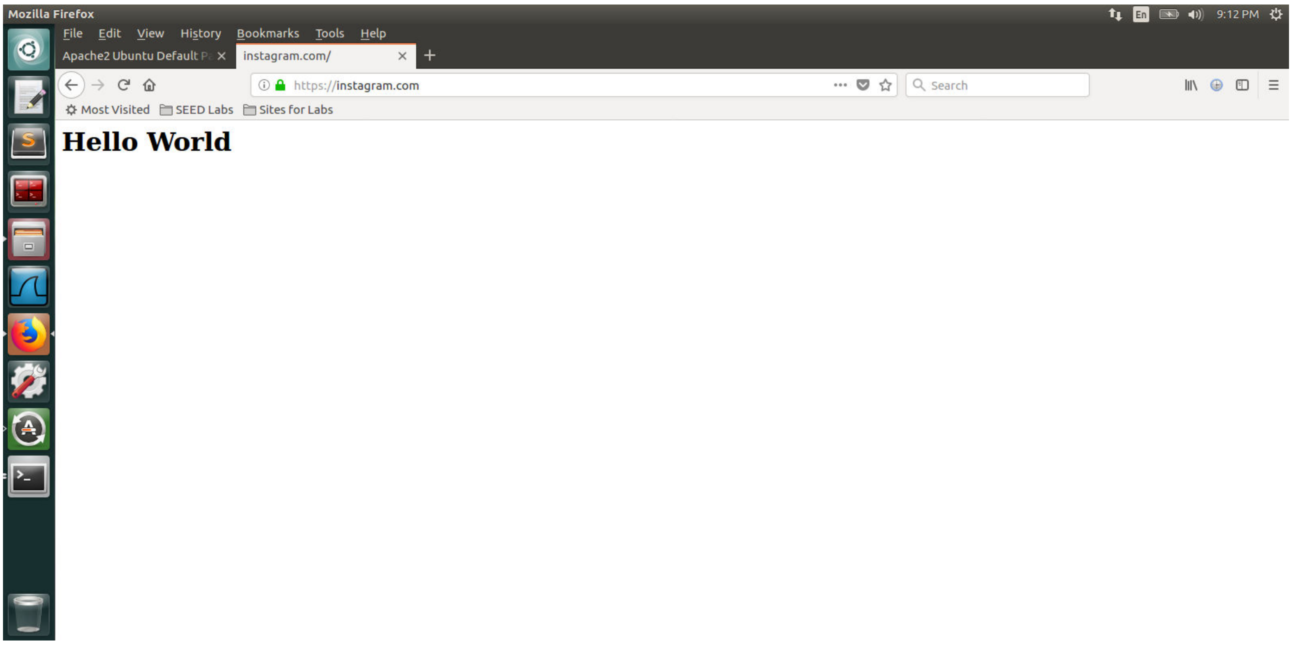
Launching a Man-In-The-Middle Attack with a Compromised CA:

1) I now assume that I access to the CA's private key and so I generate fake certificates for our malicious website and use it to make the user land on our malicious page from where I can steal the user credentials which is disastrous for the user as he may never suspect it being a fake malicious webpage.
Basically I generate certificates as in task 1 and copy them over to apache as CERT2 and modify the VirtualHost file configurations(The commands in Terminal 1 recording).



```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
        ServerName SEEDPKILAB2018.com
        DocumentRoot /var/www/seedpki
        DirectoryIndex index.html
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/CERT.pem
        SSLCertificateKeyFile /etc/apache2/ssl/KEY.pem
        </VirtualHost>

        <VirtualHost *:443>
        ServerName instagram.com
        DocumentRoot /var/www/seedpki
        DirectoryIndex index.html
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/CERT2.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
                                                                  16,42-49      Top
```

Now I can see the results in the Browser:



Mozilla Firefox
9:12 PM
File Edit View History Bookmarks Tools Help
Apache2 Ubuntu Default P
instagram.com/
https://instagram.com
Search
Most Visited
SEED Labs
Sites for Labs

Hello World