

2.3 Task 3: Deploying Certificate in an HTTPS Web Server

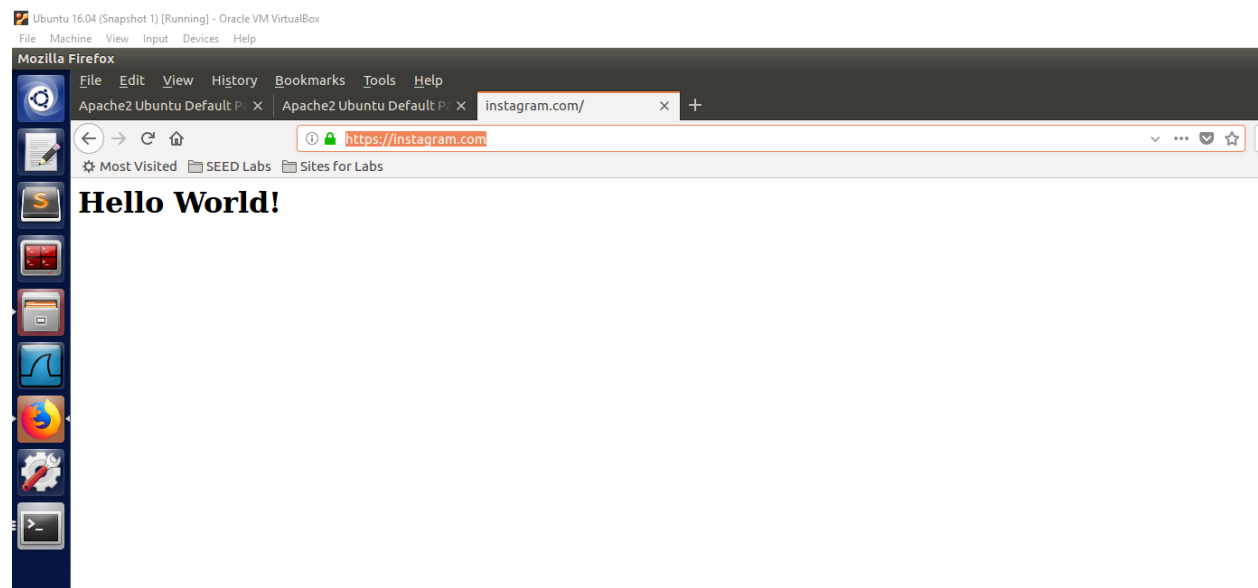
1. Modify a single byte of server.pem, and restart the server, and reload the URL. What do you observe? Make sure you restore the original server.pem afterward. Note: the server may not be able to restart if certain places of server.pem is corrupted; in that case, choose another place to modify.

After modifying the 'ST' variable under Issuer, and reloading the page, the page is still valid and displays the same thing as before when server.pem was not modified

2. Since SEEDPKILab2020.com points to the localhost, if we use <https://localhost:4433> instead, we will be connecting to the same web server. Please do so, describe and explain your observations.

We will not connect to the same website, the certification only has knowledge of domain name and the ip it is associated with in the cert file. So it will throw the error page showing that it is unable to connect

Part 3 Submission screenshots and comments



Here we have the website <https://instagram.com> compromised displaying my own html file.

Under the command **sudo nano /etc/hosts** my file looks like

```
Terminal
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 h$
::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.csrfabattacker.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      instagram.com
```

Here we modified the victims hosts file to mimic an attack on their DNS

While using the command **sudo nano default-ssl.conf** under my directory **/etc/apache2/sites-available** our file looks like

```
Terminal
GNU nano 2.5.3      File: 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 *: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>
```

Here we created a new VirtualHost with our apache server for a fake instagram.com to land on our server.

We achieved this with the following commands:

```
openssl req -new -key server.key -out instagram.csr -config openssl.cnf
openssl ca -in instagram.csr -out instagra.crt -cert ca.crt -keyfile ca.key -config
openssl.cnf
cp server.key instagram.pem
cat instagram.crt >> instagram.pem
cp instagram.crt CERT2.pem
Sudo mv "/home/seed/proj2/CERT.pm" "/etc/apache2/ssl"
sudo service apache2 restart (to end off and launch the attack)
```