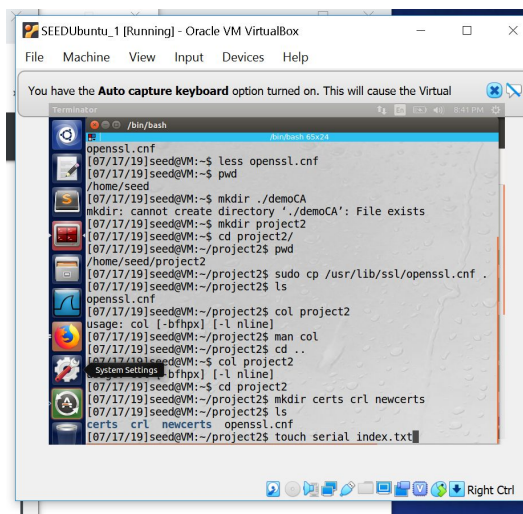


Public Key Infrastructure

Introduction:

In the Public Key Infrastructure lab, we became a root CA which is a trusted certificate authority that is able to sign certificates for other companies or websites. This lab demonstrates what a root CA is and how to sign a certificate for a company. It also shows us how to use our certificate for a web server when the browser doesn't recognize it yet. Lastly, we also see how a Man-in-the-middle attack works.

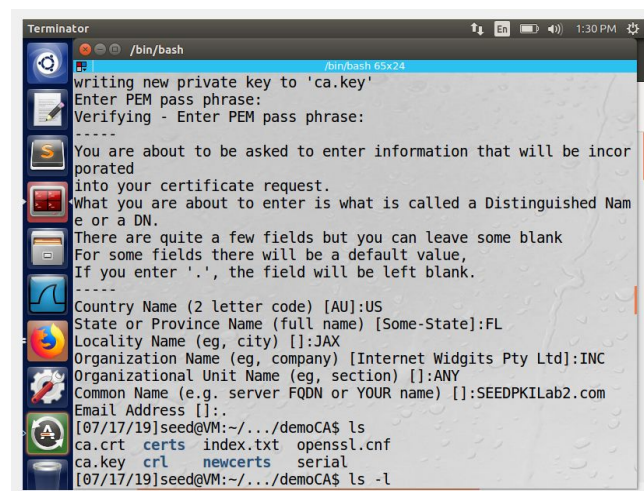
Task 1: Becoming a Certificate Authority



```
SEEDUbuntu_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

You have the Auto capture keyboard option turned on. This will cause the Virtual

Terminal
/bin/bash
openssl.cnf
[07/17/19]seed@VM:~$ less openssl.cnf
[07/17/19]seed@VM:~$ pwd
/home/seed
[07/17/19]seed@VM:~$ mkdir ./demoCA
mkdir: cannot create directory './demoCA': File exists
[07/17/19]seed@VM:~$ mkdir project2
[07/17/19]seed@VM:~/project2$ pwd
/home/seed/project2
[07/17/19]seed@VM:~/project2$ sudo cp /usr/lib/ssl/openssl.cnf .
[07/17/19]seed@VM:~/project2$ ls
openssl.cnf
[07/17/19]seed@VM:~/project2$ col project2
usage: col [-bfbpx] [-l nline]
[07/17/19]seed@VM:~/project2$ man col
[07/17/19]seed@VM:~/project2$ cd ..
[07/17/19]seed@VM:~$ col project2
[07/17/19]seed@VM:~$ cd project2
[07/17/19]seed@VM:~/project2$ touch serial index.txt
[07/17/19]seed@VM:~/project2$ touch serial index.txt
```



```
Terminator
/bin/bash
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
e 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]:FL
Locality Name (eg, city) []:JAX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:INC
Organizational Unit Name (eg, section) []:ANY
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2.com
Email Address []:.
[07/17/19]seed@VM:~/../demoCA$ ls
ca.crt  certs  index.txt  openssl.cnf
ca.key  crl    newcerts  serial
[07/17/19]seed@VM:~/../demoCA$ ls -l
```

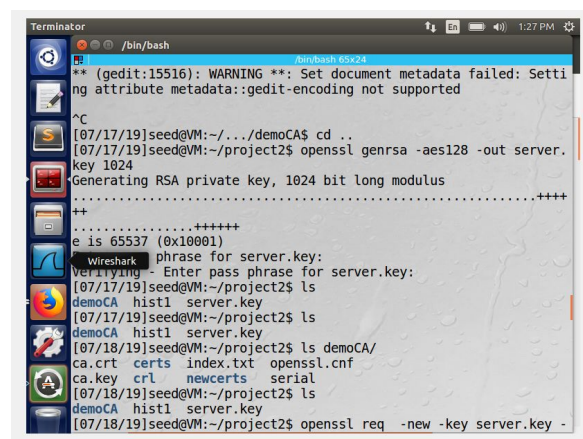
In Task 1, our goal was to become a root certificate authority (CA) in order for us to issue certificates and self-sign our own certificate. To create the certificate we need to use OpenSSL and create a configuration file. In the terminal, we made a directory to store the configuration file to use OpenSSL. Then we made subdirectories to store information such as issued certificates and new certificates. After we completed these steps, we typed in the terminal `openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf` to make our certificate a self-signed one. It asked us to create a password and fill in some information that was then stored in the files we created earlier: the private key (ca.key) and public key (ca.crt).

Task 2: Creating a Certificate for SEEDPKILab2018.com

In Task 2, we sign certificates to companies since we became a root CA in the previous task.

Step 1:

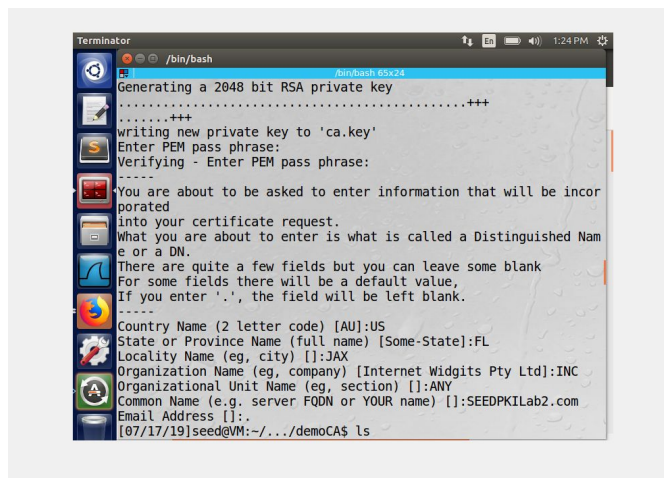
The first step to creating a certificate for SEEDPKILab2018.com is creating a public and private key for the company. We use the command `openssl genrsa -aes128 -out server.key 1024` which prompts us with a password in order to encrypt the key. This command generates an RSA private key, encrypts it using AES-128, and stores it in the encoded text file `server.key`.



```
Terminator /bin/bash
** (gedit:15516): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported
^C
[07/17/19]seed@VM:~/demoCA$ cd ..
[07/17/19]seed@VM:~/project2$ 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:
[07/17/19]seed@VM:~/project2$ ls
demoCA  hist1  server.key
[07/17/19]seed@VM:~/project2$ ls demoCA/
ca.crt  certs  index.txt  openssl.cnf
ca.key  crl    newcerts  serial
[07/17/19]seed@VM:~/project2$ ls
demoCA  hist1  server.key
[07/17/19]seed@VM:~/project2$ openssl req -new -key server.key -
```

Step 2:

The second step is to generate a certificate signing request (CSR). This is done to get the certificate for the private and public key of the company. We used the command `openssl req -new -key server.key -out server.csr -config openssl.cnf` which just requests the CSR.

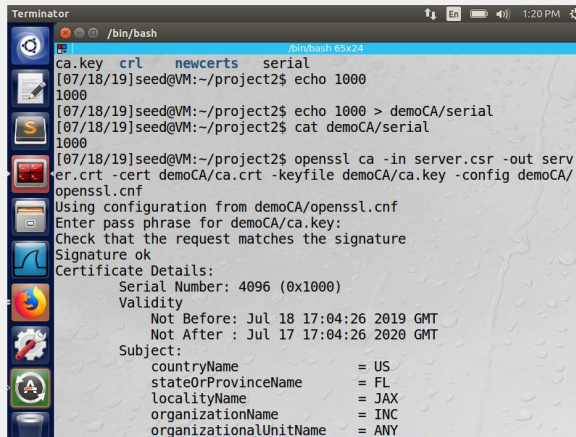


```
Terminator /bin/bash
Generating a 2048 bit RSA private key
.....+++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter 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]:US
State or Province Name (full name) [Some-State]:FL
Locality Name (eg, city) []:JAX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:INC
Organizational Unit Name (eg, section) []:ANY
Common Name (e.g. server FQDN or YOUR name) []:SEEDPKILab2.com
Email Address []:.
[07/17/19]seed@VM:~/demoCA$ ls
```

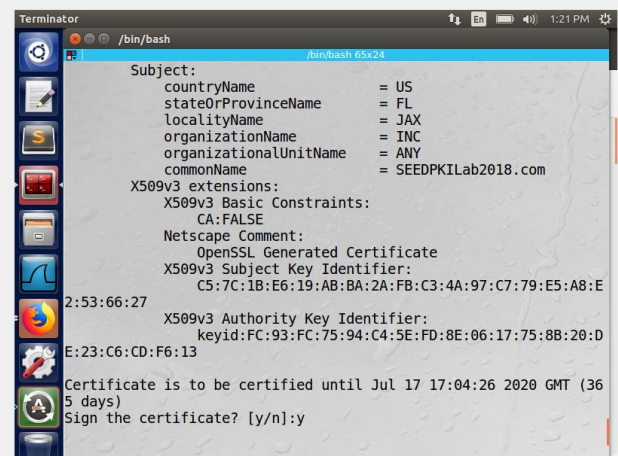
Step 3:

The third step is to generate the certificate. We opened the configuration file to change the matching rules to `policy_match` in order for the request file of the company's to match our CA.

We used the command `openssl ca -in server.csr -out server.crt -cert demoCA/ca.cert -keyfile demoCA/ca.key -config demoCA/openssl.cnf` which makes server.csr to server.crt by using our own public and private keys. Then it checks if the request server.csr matches our signature and prompts us to sign the certificate.



```
Terminator /bin/bash
ca.key crt newcerts serial
[07/18/19]seed@VM:~/project2$ echo 1000
1000
[07/18/19]seed@VM:~/project2$ echo 1000 > demoCA/serial
[07/18/19]seed@VM:~/project2$ cat demoCA/serial
1000
[07/18/19]seed@VM:~/project2$ openssl ca -in server.csr -out serv
er.crt -cert demoCA/ca.crt -keyfile demoCA/ca.key -config demoCA/
openssl.cnf
Using configuration from demoCA/openssl.cnf
Enter pass phrase for demoCA/ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4096 (0x1000)
  Validity
    Not Before: Jul 18 17:04:26 2019 GMT
    Not After : Jul 17 17:04:26 2020 GMT
  Subject:
    countryName           = US
    stateOrProvinceName   = FL
    localityName          = JAX
    organizationName       = INC
    organizationalUnitName = ANY
```

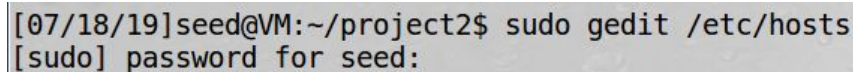


```
Terminator /bin/bash
Subject:
  countryName           = US
  stateOrProvinceName   = FL
  localityName          = JAX
  organizationName       = INC
  organizationalUnitName = ANY
  commonName            = SEEDPKILab2018.com
X.509v3 extensions:
  X.509v3 Basic Constraints:
    CA:FALSE
  Netscape Comment:
    OpenSSL Generated Certificate
  X.509v3 Subject Key Identifier:
    C5:7C:1B:E6:19:AB:BA:2A:FB:C3:4A:97:C7:79:E5:A8:E
2:53:66:27
  X.509v3 Authority Key Identifier:
    keyid:FC:93:FC:75:94:C4:5E:FD:8E:06:17:75:8B:20:D
E:23:C6:CD:F6:13
Certificate is to be certified until Jul 17 17:04:26 2020 GMT (36
5 days)
Sign the certificate? [y/n]:y
```

Task 3: Deploying Certificate in an HTTPS Web Server

Task 3 demonstrates how CA's can sign a certificate to an HTTPS web server for security.

Step 1:



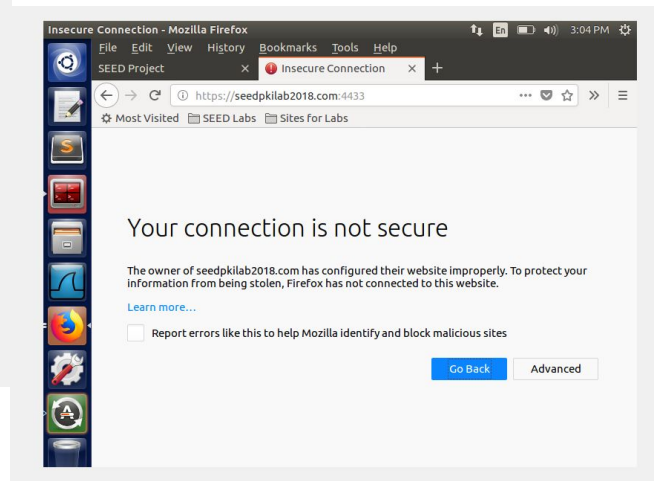
```
[07/18/19]seed@VM:~/project2$ sudo gedit /etc/hosts
[sudo] password for seed:
```

In step one we needed to make the computer recognize the name of our website which we named SEEDPKILab2018.com, which we want to add to the follow /etc/hosts so it will go to the localhost.

Step 2:

For step 2 we want to configure the web server. By using the

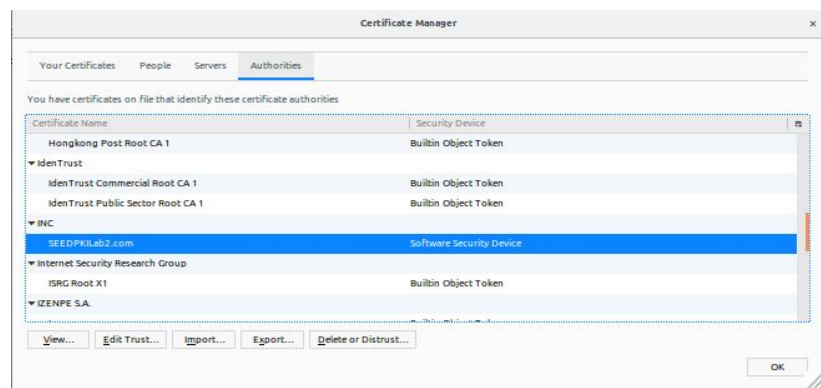
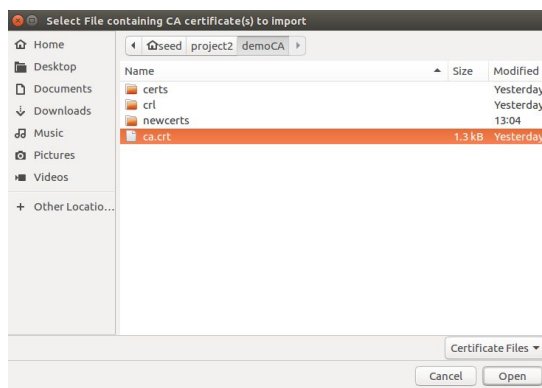
```
Terminator
/bin/bash
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe80::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 User
127.0.0.1 Attacker
127.0.0.1 Server
127.0.0.1 www.SeedLabSQLInjection.com
127.0.0.1 www.xsslabegg.com
127.0.0.1 www.csrflabelgg.com
127.0.0.1 www.csrflabelattacker.com
127.0.0.1 www.repackagingattacklab.com
127.0.0.1 www.seedlabclickjacking.com
[07/18/19]seed@VM:~/project2$ sudo vi /etc/hosts
[07/18/19]seed@VM:~/project2$ cp server.key server.pem
[07/18/19]seed@VM:~/project2$ cat server.crt >> server.pem
[07/18/19]seed@VM:~/project2$ openssl s_server -cert server.pem -
System Settings
www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

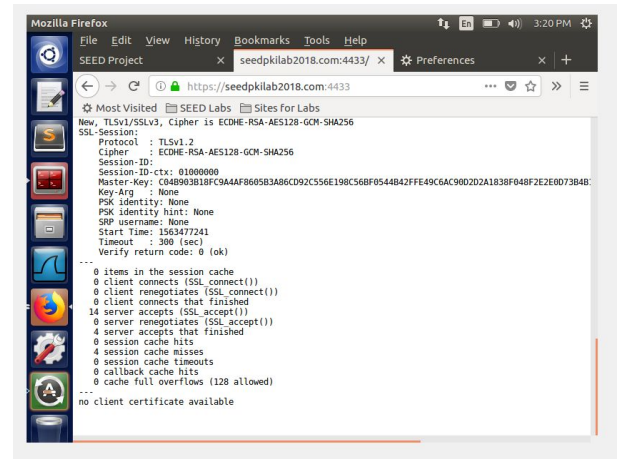


openssl s_server -cert server.pem -www command to launch a web server with the certificate we made. Then we can go to the website <https://seedpkilab2018.com:4433> and find that it says our connection is not secure. Therefore, shows that it hasn't accepted our certificate yet.

Step 3:

In the final step, we need the web browser to accept our certificate because our own CA is not recognizable to it and that's why we generate a security error when we try to go to the webpage. To do this we have to add our certificate (ca.crt) to the browser.



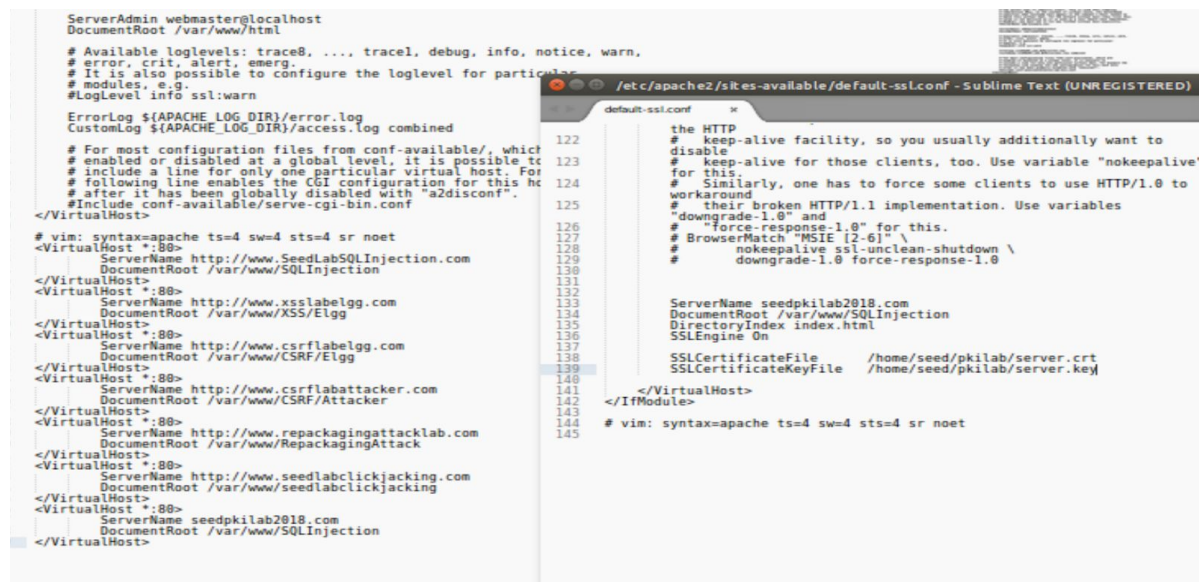


```
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:

/bin/bash
[07/29/19]seed@VM:~/pkilab$ sudo 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:
Module socache_shmcb already enabled
Module ssl already enabled
[07/29/19]seed@VM:~/pkilab$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for seedpkilab2018.com:443 (RSA): ****
[07/29/19]seed@VM:~/pkilab$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for example.com:443 (RSA): ****
[07/29/19]seed@VM:~/pkilab$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
ACCEPT
^C
[07/29/19]seed@VM:~/pkilab$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
unable to load certificate
3070891712:error:0906D064:PEM routines:PEM_read_bio:bad base64 decode:pem_lib.c:823:
[07/29/19]seed@VM:~/pkilab$
```


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

For the apache server to host the website **seedpkilab2018.com** we need to add it into a virtual host in the 000-default.conf folder. Apache server can host several websites, to host this one we had to create a virtual host. That is for a http website but for a https website where keys and certificates are used we need to edit the default-ssl-config file and add into the virtual host a new servername, document root, certificate and certificate key so it actually will be able to host the website. The certificate file paths point to the certificate location which we made in task 2.



```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., tracel, 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 disabled at a global level, it is possible to
# include a line for only one particular virtual host. For
# example, to enable the CGI configuration for this host
# it is necessary to place the following line at the end of
# the file, substituting the value of the <VirtualHost> tag
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<VirtualHost *:80>
    ServerName http://www.SeedLabSQLInjection.com
    DocumentRoot /var/www/SQLInjection
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.xsslabelgg.com
    DocumentRoot /var/www/XSS/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrflabelgg.com
    DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.csrfattacklab.com
    DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.repackagingattacklab.com
    DocumentRoot /var/www/RepackagingAttack
</VirtualHost>
<VirtualHost *:80>
    ServerName http://www.seedlabclickjacking.com
    DocumentRoot /var/www/seedlabclickjacking
</VirtualHost>
<VirtualHost *:80>
    ServerName seedpkilab2018.com
    DocumentRoot /var/www/SQLInjection
</VirtualHost>
```

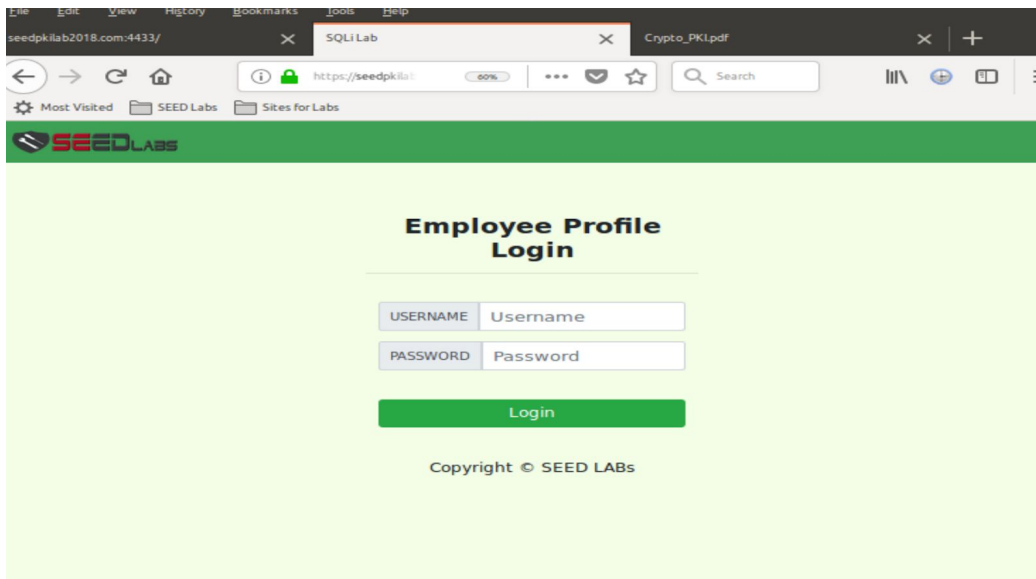
```
the HTTP
# keep-alive facility, so you usually additionally want to
# disable
# keep-alive for those clients, too. Use variable "nokeepalive"
# for this.
# Similarly, one has to force some clients to use HTTP/1.0 to
# their broken HTTP/1.1 implementation. Use variables
"downgrade-1.0" and
# "force-response-1.0" for this.
# BrowserMatch "MSIE [2-6]" \
#     nokeepalive ssl-unclean-shutdown \
#     downgrade-1.0 force-response-1.0
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

ServerName seedpkilab2018.com
DocumentRoot /var/www/SQLInjection
DirectoryIndex index.html
SSLEngine On

SSLCertificateFile /home/seed/pkilab/server.crt
SSLCertificateKeyFile /home/seed/pkilab/server.key
</VirtualHost>
</IfModule>
```

When the ssl file changes need to be enabled and for that we had to run a series of commands. *Sudo apachectl configtest* tested the config file for errors. *Sudo a2enmod ssl* enabled the ssl module. *Sudo a2ensite default-ssl* enabled the default file which enabled the site we created there. *Sudo service apache2 restart*, restarted the server after asking for the seed password and

for the password on the certificate.



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

```
the HTTP
# keep-alive facility, so you usually additionally want to
# disable
# keep-alive for those clients, too. Use variable "nokeepalive"
# for this.
# Similarly, one has to force some clients to use HTTP/1.0 to
# workaround
# their broken HTTP/1.1 implementation. Use variables
# "downgrade-1.0" and
# "force-response-1.0" for this.
# BrowserMatch "MSIE [2-6]" \
#     nokeepalive ssl-unclean-shutdown \
#     downgrade-1.0 force-response-1.0

ServerName seedpkilab2018.com
DocumentRoot /var/www/SQLInjection
DirectoryIndex index.html
SSLEngine On

SSLCertificateFile    /home/seed/pkilab/server.crt
SSLCertificateKeyFile /home/seed/pkilab/server.key

#this is the fake
ServerName seedpkilab2018.com
DocumentRoot /home/seed/pkilab/test
DirectoryIndex index.html
SSLEngine On

SSLCertificateFile    /home/seed/pkilab/server.crt
SSLCertificateKeyFile /home/seed/pkilab/server.key

</VirtualHost>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

To impersonate seedpkilab2018.com we have to make another virtual host and we used example.com which will route the user to the seedlabpki2018.com. When the user goes to the website they will be redirected to the fake malicious website. After doing all of that and trying to go to the website what we end up getting is that the connection is not secure. This is because of certificate issues therefore the browser does not allow the connection because it sees it as malicious.

Your connection is not secure

The owner of example.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

[Learn more...](#)

☐

Report errors like this to help Mozilla identify and block malicious sites

Go Back

Advanced

Conclusion:

Throughout this lab we learned how certificates are set up with the CA and the client. How to set up the CA. How to get the browser to accept other certificates. How to host a webpage using the certificate and key and also how to initiate a man in the middle attack. Step one is where CA makes certificate, server requests certificate from CA to make a certificate. The CA also makes a key for signing the certificate. Step two is to make a certificate for seedpkilab2018.com. Step three is to get the browser to recognize that certificate. Step 4 is to host a webpage on that webpage using the certificate for seedpki2018.com. Step 5 is to initiate a man in the middle attack but it won't work because the certificates don't work for the malicious website.