Assignment 6
Public Key Cryptography

Goal

- Generate and use public key certificate chains for web security.

1 Introduction

Students are expected to practice the use of public keys and digital certificates for the authentication of people and services.

Students will configure an Apache server to accept HTTPS secure connections.

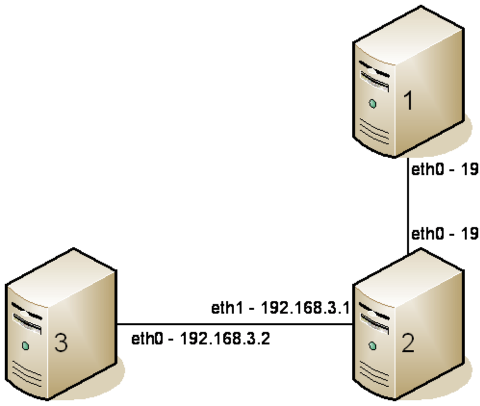
2 Preliminaries

Three entities are required to perform the job:

- The client
- The server
- The certification authority

Each of these entities are going to be run in a different computer.

We will use the architecture of Assignment 4, where machine 4 may remain shutdown. The firewall should accept all traffic and not perform address translation.



The client will run in machine 1, the server will run in machine 2 and the certification authority in machine 3.

3 HTTPS Secure Connections

It is often necessary to create private websites accessible only to authenticated users. In this section, students will learn how to do it safely.

In this process students will learn how to:

1) Create a simple Certification Authority;
2) Configure a Web Server to accept HTTP connections over TLS / SSL (aka https);
3) Configure the authentication of the website with a pair of user / password;
4) Configure website authentication with public client certificates.

3.1 Creation of a certificate authority

Run the following sequence of commands on a terminal of the virtual machine 3.

```
> sudo mkdir /root/CA
> sudo chmod 0770 /root/CA
> sudo -i
> cd /root/CA
```

The following command generates a key pair for the 2048-bit RSA cipher algorithm and encrypts it with the symmetric cipher algorithm 3DES and a password provided by the user. This will be the public and private key of the Certification Authority.

```
> sudo openssl genrsa -des3 -out my-ca.key 2048
```

After the keys are generated, it is necessary to sign the public key with the CA's private key. As this is a root certificate (i.e. self-signed) the private signature key is the public key pair to be signed. A self-signed certificate is similar to a certificate request, because in a certificate request a certificate request is also sent to the certifying entity, which is a self-signed certificate, which is why the same "req" command is used, but with the "-x509" option to generate such a certificate. In the same command it is also indicated the validity time of the certificate "10 years = 10x 365 days".

```
> openssl req -new -x509 -days 3650 -key my-ca.key -out my-ca.crt
    COUNTRY: PT
    STATE:   Lisbon
    Locality: Lisbon
    Organization: CSC-<groupname>
    Organizational Unit: CA-<groupname>
    Common Name: CA<goupname>
    Email: your email address
```

To view the contents of your CA certificate you can run the following command

```
> openssl x509 -in my-ca.crt -text  2>&1 | less
```

3.2 Creating a certificate for the WEB server

Next you will create a key pair for the Web Server.

Run the following commands in a terminal of machine 2.

```
> cd ~/csc-course/assignment6
> openssl genrsa -out csc-server.pem 1024
```

After generating the keys you must generate the certificate request in the same way as before. Note that this time the -x509 option is not used because it is intended to generate a certificate request and not a root certificate

Assuming the architecture of assignment 4, then machine 2 ip address is 192.168.1.2

```
> openssl req -new -key csc-server.pem -out csc-server.csr
        COUNTRY: PT
        STATE:   Lisbon
        Locality: Lisbon
        Organization: CSC-<groupname>
        Organizational Unit: Servidor Web
        Common Name: 192.168.1.2
        Email: your email address
```

Then you need to send the certificate request file to machine 3. Got to machine 3 and run:

```
> sudo –i
> cd /root/CA
> scp user@192.168.1.2:csc-course/assignment6/csc-server.csr .
```

Use the certificate from our CA to sign the WebServer certificate.

```
> openssl x509 -req -in csc-server.csr -out csc-server.crt -sha1 -CA my-ca.crt
-CAkey my-ca.key  -CAcreateserial -days 3650
```

make certificates accessible to non-root users:

```
> chmod 444 *.crt
> cp *.crt /home/user/csc-course/assignment6/
```

Once again to view the issued certificate

```
> openssl x509 -in csc-server.crt -text -noout  2>&1 | less

>exit
```

3.3 Configuring the Apache server on the Web server

Install the ssl module in apache at machine 2

```
> sudo a2enmod ssl
```

Install the private key the server certificate and the certificate from your CA

At machine 2 run:

```
> cd ~/csc-course/assignment6
> scp 192.168.3.2:~/csc-course/assignment6/csc-server.crt .
> scp 192.168.3.2:~/csc-course/assignment6/my-ca.crt .
> chmod 0400 *.crt
```

Edit the Apache WebServer configuration file.

```
> gedit /etc/apache2/sites-available/default-ssl.conf
```

and change the lines below to:

```
DocumentRoot /var/www/SSL

#  Server Certificate
SSLCertificateFile /home/user/csc-course/assignment6/csc-server.crt

#   Server Private Key:
SSLCertificateKeyFile /home/user/csc-course/assignment6/csc-server.pem

#   Server Certificate Chain:
SSLCertificateChainFile /home/user/csc-course/assignment6/my-ca.crt

#   Certificate Authority (CA):
SSLCACertificateFile /home/user/csc-course/assignment6/my-ca.crt
```

Create the directory that will contain the protected content

```
> sudo mkdir /var/www/SSL
> sudo chmod 0775 /var/www/SSL
> cd /var/www/SSL
```

Create an index.html file in this directory with the following code

```
> sudo gedit index.html
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"> <title>HTTPS Default
Page</title>
```

```
</head>
<body>
<h2>SSL default index page<br /><br /></h2>
<h4><a href="Passneeded/">Password Protected Directory</a></h4>
<h4><a href="Certneeded/">Special Directory - Client Certificate Required</a></h4>
<h4><a href="PassAndCert/">Special Directory - Client Cert AND Password Required</a></h4>
</body>
</html>
```

You can find the above text in the file ~/csc-course/assignment6/index1.html

```
>sudo cp ~/csc-course/assignment6/index1.html index.html
```

Create three other directories and place index.html files

```
> sudo mkdir Passneeded
```

```
> sudo gedit Passneeded/index.html
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"> <title>HTTPS Default
Page</title>
</head>
<body>
<h2> You are seeing the <b>Password</b> Protected Area<br /><br /></h2>
</body>
</html>
```

You can find the above text in the file ~/csc-course/assignment6/index2.html

```
>sudo cp ~/csc-course/assignment6/index2.html Passneeded/index.html
```

```
> sudo mkdir Certneeded
```

```
> sudo gedit Certneeded/index.html
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"> <title>HTTPS Default
Page</title>
</head>
<body>
<h2> You are seeing the <b>Certificate<\b>  Protected Area<br /><br /></h2>
</body>
</html>
```

You can find the above text in the file ~/csc-course/assignment6/index3.html

```
>sudo cp ~/csc-course/assignment6/index3.html Certneeded/index.html
```

```
> sudo mkdir PassAndCert
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"> <title>HTTPS Default
Page</title>
</head>
<body>
<h2> You are seeing the <b>Certificate and Password</b>  Protected Area<br /><br /></h2>
</body>
</html>
```

You can find the above text in the file ~/csc-course/assignment6/index4.html

>sudo cp ~/csc-course/assignment6/index4.html PassAndCert/index.html

Restart the web server

```
> a2ensite default-ssl.conf
> service apache2 restart
```

check that apache2 is listening on ports 80 and 443

```
> sudo netstat -tulpn
```

3.4 Configuration of authentication mechanism with user / password

Create the directory where the password file will be

```
>sudo mkdir /etc/apache2/www
```

Create the first account with access to protected pages

```
> sudo htpasswd -c -m /etc/apache2/www/.htpasswd <username_1>
```

Create other accounts with access to protected pages

```
> htpasswd -m /etc/apache2/www/.htpasswd <username_n>
```

Change and configure the web server to protect the directory protected with those accounts.

```
> chown www-data.www-data /etc/apache2/www/.htpasswd
> chmod 0460 /etc/apache2/www/.htpasswd
```

Edit the /etc/apache2/sites-available/default-ssl.conf file and add:

```
<Directory "/var/www/SSL/Passneeded">
        AuthType Basic
        AuthName "Username and Password Required"
        AuthUserFile /etc/apache2/www/.htpasswd
        Require valid-user
</Directory>
```

Restart the webserver

```
> service apache2 restart
```

On machine 1 open firefox and go to page

https://192.168.1.2/Passneeded

3.5 Creating the Client Certificate

Go to machine 3 to generate a key pair for the client. Notice that for the client the key is not created by the user. The user receives the certificate and the key in a single packet. This is less secure but simpler for the common user.

```
> sudo -i
> cd /root/CA
> openssl genrsa -des3 -out client-cert.key 1024

> openssl req -new -key client-cert.key -out client-cert.csr

> openssl x509 -req -in client-cert.csr -out client-cert.crt -sha1 -CA my-ca.crt -CAkey my-ca.key -CAcreateserial -days 3650

> openssl pkcs12 -export -in client-cert.crt -inkey client-cert.key -name "User Cert" -out client-cert.p12

> openssl pkcs12 -in client-cert.p12 -clcerts -nokeys -info
```

Make the bundle accessible to other users:

```
> chmod 444 client-cert.p12

> cp client-cert.p12 /home/user/csc-course/assignment6/
```

3.6 Import the client and CA certificates into the browser

The import into the browser is more easily performed by first importing into the server.

Go to machine 2, the server, and import the CA certificate and the bundle with the certificate and private key of the client

```
> sudo scp user@192.168.3.2:/home/user/csc-course/assignment6/my-ca.crt /var/www/html/
> sudo scp user@192.168.3.2:/home/user/csc-course/assignment6/client-cert.p12 /var/www/html/
```

Go to to the browser in machine 1, the client, a navigate to

http://192.168.1.2/my-ca.crt

Choose to open the certificate, and then choose to install the certificate.
When you are asked to choose the place where to place the certificate, choose **trusted root certificate.**

.

Then go to

http://192.168.1.2/client-cert.p12

Choose to open the certificate. When you are asked to choose the place where to place the certificate, choose **personal**
Otherwise, save the certificate, then install the certificate in the Firefox menu option
Preferences/Advanced/Certificates/ViewCerificates/YourCertificates/Import.

3.7 Apache server configuration to accept authentication with certificates

In machine 2, add the following code to /etc/apache2/sites-available/default-ssl.conf

```
<Directory /var/www/SSL/Certneeded>
            SSLVerifyClient require
            SSLVerifyDepth 1
</Directory>
```

Restart the web server

> service apache2 restart

Go to machine 1 and test

https://192.168.1.2/Certneeded

3.8 Apache server configuration to accept authentication with certificates and passwords

Add the following code to /etc/apache2/sites-available/default-ssl.conf

```
<Directory "/var/www/SSL/PassAndCert">
            SSLVerifyClient require
            SSLVerifyDepth 1
            AuthType Basic
            AuthName "Restricted Area"
            AuthUserFile /etc/apache2/www/.htpasswd
            Require valid-user
</Directory>
```

Restart the web server

> service apache2 restart

Go to machine 1 and test

https://192.168.1.2/PassAndCert

3.9 [Optional] Configure the web server to accept citizen card

You will need to:

1) Configure the server to accept certificates signed by the Portuguese National Authority.
2) Configure the client to read the key and certificate from a smartcard.

For the first step, you need to go to

http://pki.cartaodecidadao.pt/publico/certificado/cc_ec_cidadao/

and download the certificates there. You will need all three root certificates because you are not sure with which one was your card signed.

Then concatenate all the files into one single file together with your old root CA.

In machine 2

```
> cd /home/user/csc-course/assignment6
> cat Cartao\ do\ Cidadao\ 00?.cer >> my-ca.crt
```

For the second step, follow the instructions in

https://www.autenticacao.gov.pt/cc-software

in machine 1