

SECURE SOCKETS LAYERS.

<http://nassih.com>

© MOHAMED NASSIH.

1. Introduction.

SSL -Secure Sockets Layers- est un protocole de la couche présentation, qui permet de sécuriser la communication entre un client et un serveur en chiffrant les données échangées. Les données transmises sont chiffrées en utilisant une structure à clé publique. Le poste client utilise la clé publique du serveur pour chiffrer les messages et le serveur utilise son clé privée pour faire le déchiffrement.

Le partage des clés se fait en utilisant un annuaire électronique ou un site web. Cette méthode de partage ne garantit pas que la clé publique est bien celle de l'utilisateur à qui elle est associée. Ce problème pourra être résolu en utilisant les certificats.

Un certificat est un document électronique qui permet d'associer une clé publique à une machine afin d'en assurer la validité. Il est délivré par un organisme appelé l'autorité de certification (Certification Authority-CA-). Un certificat a toujours une date de validité.

C'est le standard X509 qui normalise la structure des certificats. Un certificat contient des informations comme son numéro de série, l'algorithme de chiffrement utilisé pour signer le certificat, le nom de l'autorité de certification, les dates de validation et la signature. L'autorité de certification utilise une fonction de hachage à ces informations et la clé publique pour signer le certificat.

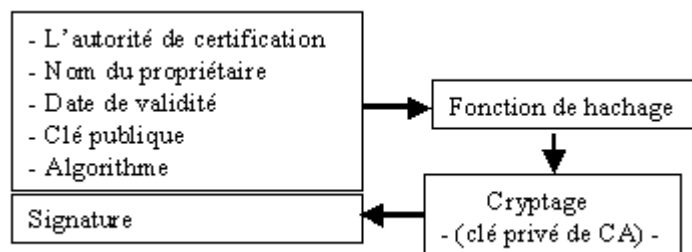


Figure 1.1 : création du certificat.

Si Bob voudra communiquer avec Alice il doit avoir le certificat de Alice qui contient sa clé publique et la signature de l'autorité de certification (AC), Bob va utiliser la clé publique de AC pour déchiffrer la signature puis appliquer la fonction de hachage, il doit obtenir la première partie du certificat, la partie qui contient les informations sur le certificat.

2. L'implémentation de SSL.

Un serveur SSL doit avoir une clé privée et un certificat, nous avons donc besoin d'une autorité de certification, pour notre laboratoire, nous allons créer une autorité de certification locale qui va jouer le même rôle d'une autorité de certification commerciale, ceci est possible avec l'outil OpenSSL.

OpenSSL est un ensemble d'outils de cryptographie qui implémente SSL (V2 et V3) et le protocole TLS (Transport Layer Security). Il permet également de créer des certificats.

Nous commençons notre implémentation par l'installation d'OpenSSL, puis nous installons le module Apache Mod_ssl, qui ajoute le support des transactions sécurisé de http (https). Et enfin nous créons l'autorité de certification locale et le certificat pour le serveur Web.

Étape 1 : compilation et installation de OpenSSL.

La figure 2.1 montre l'installation et la configuration d'OpenSSL.

```
tar -zxvf openssl-0.9.6d.tar.tar
cd openssl-0.9.6d
./config shared
make
make test
make install
echo "/usr/local/ssl/lib" >> /etc/ld.so.conf
ldconfig
```

Figure 2.1 : l'installation et la configuration de SSL.

Étape 2 : la configuration de mod_ssl avec apache.

Mod_ssl est un module de sécurité pour Apache, il utilise les outils fournis par OpenSSL.
La figure 2.2 montre la configuration de ce module.

```
tar -zxvf apache_1.3.33.tar
tar -zxvf mod_ssl-2.8.22-1.3.33.tar
cd mod_ssl-2.8.22-1.3.33
./configure --with-apache=/tmp/apache_1.3.33
```

Figure 2.2 : la configuration du mod_ssl.

Étape 3 : Installation et configuration d'Apache :

La figure 2.3 montre l'installation et la configuration d'Apache.

```
cd apache-1.3.3
SSL_BASE="/usr/local/ssl" \
./configure \
--enable-module=unique_id \
--enable-module=rewrite \
--enable-module=speling \
--enable-module=expires \
--enable-module=info \
--enable-module=usertrack \
--enable-module=proxy \
--enable-module=userdir \
--enable-module=so \
--enable-shared=ssl \
--enable-module=ssl \
make
make install
```

Figure 2.3 : l'installation et la configuration de Apache.

Étape 4 : Création du certificat de l'autorité de certification.

Pour les sites professionnels, un certificat doit être acheté chez une autorité de certification. Dans notre cas on va créer une autorité de certification locale en utilisant OpenSSL (dans le répertoire `/root/CA`). On doit créer une clé privée, puis utiliser cette clé pour créer le certificat.

```
[root@web_server root]# chmod 0770 /root/CA
[root@web_server root]# cd CA
[root@web_server CA]# openssl genrsa -des3 -out my-ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
Enter pass phrase for my-ca.key:
Verifying - Enter pass phrase for my-ca.key:
[root@web_server CA]# _
```

Figure 2.4 : la création de la clé privée pour l'autorité de certification.

La figure 2.4 montre la création de la clé privée. Le système demande une phrase de passage, c'est comme un mot de passe que le système demande à chaque utilisation de cette clé. Nous pourrions maintenant créer le certificat en utilisant la commande `openssl`. Nous utilisons le standard X509 pour définir la structure du certificat. (Voir figure 2.5).

```
[root@web_server CA]# openssl req -new -x509 -days 3650 -key my-ca.key -out my-ca.crt
Enter pass phrase for my-ca.key:
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) [GB]:CA
State or Province Name (full name) [Berkshire]:QC
Locality Name (eg, city) [Newbury]:MTL
Organization Name (eg, company) [My Company Ltd]:POLY
Organizational Unit Name (eg, section) []:GI
Common Name (eg, your name or your server's hostname) []:MASSPROJ
Email Address []:
[root@web_server CA]# _
```

Figure 2.5 : création du certificat avec 10 ans de validité.

la commande «`openssl x509 -in my-ca.crt -text -noout`» permet d'afficher le certificat créé.

```
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 0 (0x0)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=CA, ST=QC, L=MTL, O=POLY, OU=GI, CN=NASSPROJ
  Validity
    Not Before: Aug 22 15:29:20 2005 GMT
    Not After : Aug 20 15:29:20 2015 GMT
  Subject: C=CA, ST=QC, L=MTL, O=POLY, OU=GI, CN=NASSPROJ
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
    Modulus (2048 bit):
      00:b9:2f:eb:ee:98:ec:67:db:8f:72:30:3b:23:d5:
      75:06:1d:46:fb:34:39:0e:6c:1e:e0:65:54:18:a9: ...
    Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      58:B9:73:64:EB:0B:09:B0:F6:BF:A4:A6:5E:9E:54:7E:0E:56:A4:6C
    X509v3 Authority Key Identifier:
      keyid:58:B9:73:64:EB:0B:09:B0:F6:BF:A4:A6:5E:9E:54:7E:0E:56:A4:6C
      DirName:/C=CA/ST=QC/L=MTL/O=POLY/OU=GI/CN=NASSPROJ
      serial:00
    X509v3 Basic Constraints:
      CA:TRUE
  Signature Algorithm: md5WithRSAEncryption
  b0:23:0e:3e:8c:f6:3e:80:fc:3d:99:89:18:d4:48:5d:c4:e8:
  41:c5:13:e5:83:09:c9:cd:0f:96:ce:c8:4a:e1:b5:76:94:c8: ...
```

Figure 2.6 : le certificat de l'autorité de certification.

Étape 5 : Création du certificat pour le serveur web.

Le serveur Web a besoin d'une clé privée et une demande de certificat (Certificate Signing Request, CSR). Cette demande (qui contient une copie de la clé privée) doit être signée en utilisant la clé privée de l'autorité de certification (my-ca). La figure 2.7 montre la création de la clé privée d'une longueur de 1024 bit.

```
[root@web_server CA]# openssl genrsa -des3 -out web-server.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for web-server.key:
Verifying - Enter pass phrase for web-server.key:
[root@web_server CA]# _
```

Figure 2.7: création de la clé privé pour le serveur web.

Nous allons utiliser la clé privée web-server.key pour créer une demande de certificat (Certificate Signing Request, CSR).

```

[root@web_server CA]# openssl req -new -key web-server.key -out web-server.csr
Enter pass phrase for web-server.key:
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) [GB]:CA
State or Province Name (full name) [Berkshire]:QC
Locality Name (eg, city) [Newbury]:MTL
Organization Name (eg, company) [My Company Ltd]:POLY
Organizational Unit Name (eg, section) []:GI
Common Name (eg, your name or your server's hostname) []:web_server
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

```

Figure 2.8: création de la demande de certificat.

La commande de la figure 2.8 demande un certain nombre d'information, comme le nom de domaine, ce nom doit correspondre au nom du site si nous avons un nom de domaine valide. Il nous reste la signature de la demande avec la clé privée de l'autorité de certification my-ca.key. Dans le cas où on a un site professionnel, la demande de certificat doit être envoyée à l'autorité de certification pour la signature (durant le procédure d'inscription). (Voir figure 2.9).

```

[root@web_server CA]# openssl x509 -req -in web-server.csr -out web-server.crt -
sha1 -CA my-ca.crt -CAkey my-ca.key -CAcreateserial -days 3650
Signature ok
subject=/C=CA/ST=QC/L=MTL/O=POLY/OU=GI/CN=web_server
Getting CA Private Key
Enter pass phrase for my-ca.key:
[root@web_server CA]# _

```

Figure 2.9 : signature du certificat.

Pour afficher le certificat créé, Nous utiliserons la commande «openssl x509 -in web-server.crt -text -noout» (figure 2.10). Il est très conseillé de protéger les clés en utilisant la commande «chmod 0400 *.key».

```

Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 4 (0x4)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=CA, ST=QC, L=MTL, O=POLY, OU=GI, CN=NASSPROJ
    Validity
      Not Before: Aug 22 19:44:47 2005 GMT
      Not After : Aug 20 19:44:47 2015 GMT
    Subject: C=CA, ST=QC, L=MTL, O=POLY, OU=GI, CN=web_server
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:c9:88:9f:89:43:16:46:d9:d3:31:d2:37:84:b7:
        fe:a1:60:b6:97:8b:4f:0e:31:ae:f5:5a:05:16:ed: .....
      Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
    a8:d2:da:24:1c:95:90:91:d9:b7:81:21:8b:f3:af:ce:a4:d1:
    a6:bc:64:7a:04:0c:38:9e:03:a7:ea:34:79:6f:fa:32:d1:ac: ....

```

Figure 2.10 : le certificat du serveur web.

Nous devons faire la mise en place des clés et des certificats créés dans les répertoires du serveur Web. Le répertoire par défaut du serveur Web est /www/htdocs, il va contenir les fichiers qui vont être transférés sans chiffrage. Nous allons créer un autre répertoire /www/SSL qui sera le répertoire par défaut de notre serveur Web sécurisé. (Voir figure 2.11)

```

[root@web_server CA]# chmod 0400 *.key
[root@web_server CA]# cp web-server.crt /usr/local/apache/conf/ssl.crt
[root@web_server CA]# cp my-ca.crt /usr/local/apache/conf/ssl.crt
[root@web_server CA]# cp web-server.key /usr/local/apache/conf/ssl.key
[root@web_server CA]#
[root@web_server CA]# mkdir /usr/local/apache/SSL
[root@web_server CA]# chmod 0775 /usr/local/apache/SSL
[root@web_server CA]# mkdir /usr/local/apache/SSL/Passneeded
[root@web_server CA]# mkdir /usr/local/apache/SSL/Certneeded
[root@web_server CA]# mkdir /usr/local/apache/SSL/PassAndCert
[root@web_server CA]# _

```

Figure 2.11 : La mise en place des clés et certificats.

Étape 6: configurer le serveur Web Apache.

La figure 2.12 montre les modifications que nous avons faites au fichier httpd.conf.

```

# Le répertoire par défaut
DocumentRoot "/usr/local/apache/SSL"
# Le nom du serveur et l'adresse de l'administrateur
ServerName web_server.com
ServerAdmin webmaster@web_server.com
# Le serveur doit écouter sur les deux ports 80 et 443
Listen 192.168.0.150:80
Listen 192.168.0.150:443
# Le chemin pour le certificat du serveur
SSLCertificateFile /usr/local/apache/conf/ssl.crt/web-server.crt
# le chemin pour la clé privée
SSLCertificateKeyFile /usr/local/apache/conf/ssl.key/web-server.key
# la chaîne du certificat du serveur
SSLCertificateChainFile /usr/local/apache/conf/ssl.crt/my-ca.crt
# le certificat du AC
SSLCACertificateFile /usr/local/apache/conf/ssl.crt/my-ca.crt

```

Figure 2.12 : le fichier /usr/local/apache/SSL/httpd.conf.

Nous démarrons le serveur Web en utilisant le scripte apachectl (Voir figure 2.13).

```

[root@web_server root]# /usr/local/apache/bin/apachectl startssl
[Mon Aug 22 13:06:31 2005] [alert] httpd: Could not determine the server's fully
qualified domain name, using 192.168.0.150 for ServerName
Apache/1.3.33 mod_ssl/2.8.22 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide us with the pass phrases.

Server 192.168.0.150:443 (RSA)
Enter pass phrase:

Ok: Pass Phrase Dialog successful.
/usr/local/apache/bin/apachectl startssl: httpd started
[root@web_server root]# netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:32768           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:32769        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:443           0.0.0.0:*               LISTEN
[root@web_server root]# _

```

Figure 2.13 : le démarrage de SSL.

En utilisant la commande netstat, nous pourrions nous assurer que le serveur web écoute bien sur les deux ports 80 et 443.

Lorsqu'on essaie d'accéder, à partir d'un explorateur, au serveur https. un message de type «Le certificat de sécurité a été émis par une société à laquelle vous n'avez pas choisi de faire confiance.. Voulez-vous continuer?», ce qui veut dire que le certificat n'a pas été choisi avec une autorité de certification. En acceptant ce type de certificat, on pourra accéder au serveur Web sécurisé.