

Rapport du TP sur l'installation d'Open SSL et d'Apache 2 :

1. Mise en place :

Tout d'abord il fallait configurer une VM (machine virtuelle) Ubuntu sur VirtualBox et une fois lancée, il fallait la mettre à jour en effectuant la commande : « `sudo apt update && sudo apt upgrade -y` » puis installer les services Open SSL et Apache 2 avec la commande : « `sudo apt install openssl apache2 -y` ».

2. Génération d'un certificat auto-signé avec Open SSL :

Une fois la mise en place terminée, nous passons à la création du répertoire « my certs » qui contiendra toutes les informations nécessaires pour la création du certificat.

```
user@ubuntu-server:~$ sudo mkdir -p /etc/ssl/mycerts
user@ubuntu-server:~$ cd /etc/ssl/mycerts
user@ubuntu-server:/etc/ssl/mycerts$
```

Ensuite, nous nous déplaçons dans ce répertoire afin de réaliser les actions suivantes :

La génération d'une clé privée de 2048 bits avec la commande ci-dessous :

```
user@ubuntu-server:/etc/ssl/mycerts$ sudo openssl genrsa -out ligma.key 2048_
```

Puis nous lançons la création de la demande de signature du certificat (CSR) :

```
user@ubuntu-server:/etc/ssl/mycerts$ sudo openssl req -new -key ligma.key -out ligma.csr_
```

On nous demande de rentrer les différentes informations pour le certificat et voici ce que nous avons entrés :

```
user@ubuntu-server:/etc/ssl/mycerts$ sudo openssl req -new -key ligma.key -out ligma.csr
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]:FR
State or Province Name (full name) [Some-State]:Loire-Atlantique
Locality Name (eg, city) []:Nantes
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Ynov
Organizational Unit Name (eg, section) []:Info
Common Name (e.g. server FQDN or YOUR name) []:ligma.local
Email Address []:user@test.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:root
An optional company name []:
user@ubuntu-server:/etc/ssl/mycerts$ _
```

Après ça, nous générons le certificat auto-signé qui aura une durée de validité d'un an :

```
user@ubuntu-server:/etc/ssl/mycerts$ sudo openssl x509 -req -days 365 -in ligma.csr -signkey ligma.key -out ligma.crt
Certificate request self-signature ok
subject=C=FR, ST=Loire-Atlantique, L=Nantes, O=Ynov, OU=Info, CN=ligma.local, emailAddress=user@test.com
```

On vérifie que nous avons toutes les informations au sein du répertoire :

```
user@ubuntu-server:/etc/ssl/mycerts$ ls -l /etc/ssl/mycerts/
total 12
-rw-r--r-- 1 root root 1371 Jan 21 14:57 ligma.crt
-rw-r--r-- 1 root root 1082 Jan 21 14:54 ligma.csr
-rw----- 1 root root 1708 Jan 21 14:50 ligma.key
```

3. Configuration du serveur Apache2 pour HTTPS :

On active le module SSL d'Apache2 :

```
user@ubuntu-server:/$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Puis on redémarre Apache2 afin qu'il prenne en compte l'activation du module SSL :

```
user@ubuntu-server:/$ sudo systemctl restart apache2
```

On modifie le fichier de configuration du site pour y déposer notre configuration :

```
GNU nano 8.1 /etc/apache2/sites-available/ligma-ssl.conf *
<VirtualHost *:443>
    ServerName ligma.local
    DocumentRoot /var/www/html

    SSLEngine on
    SSLCertificateFile /etc/ssl/mycerts/ligma.crt
    SSLCertificateKeyFile /etc/ssl/mycerts/ligma.key

    <Directory /var/www/html>
        AllowOverride All
    </Directory>
</VirtualHost>
```

On active la configuration puis on redémarre une nouvelle fois le service :

```
user@ubuntu-server:/$ sudo a2ensite ligma-ssl.conf_
systemctl restart apache2
user@ubuntu-server:/$ sudo systemctl restart apache2
```

4. Configuration réseau et accès au site :

On modifie le fichier des hôtes pour y ajouter notre site avec son adresse afin de pouvoir y accéder en local :

```
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu-apache2
127.0.0.1 ligma.local
# 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
```

[Read 9 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

Puis nous ouvrons notre navigateur en entrant l'adresse : <https://ligma.local>, le navigateur nous transmettra une alerte et il suffit de cliquer sur continuer en acceptant les risques pour être redirigé sur le site en question comme ci-dessous :



Nous pouvons maintenant jeter un œil au certificat :

Certificate

ligma.local

Subject Name

Country

FR

State/Province

Loire-Atlantique

Locality

Nantes

Organization

Ynov

Organizational Unit

Info

Common Name

ligma.local

Email Address

user@test.com

Issuer Name

Country

FR

State/Province

Loire-Atlantique

Locality

Nantes

Organization

Ynov

Organizational Unit

Info

Common Name

ligma.local

Email Address

user@test.com

Validity

Not Before

Tue, 21 Jan 2025 17:31:44 GMT

Not After

Wed, 21 Jan 2026 17:31:44 GMT

Public Key Info

Algorithm

RSA

Key Size

2048

Exponent

65537

Modulus

CD:35:A0:F2:ED:93:C9:4C:BF:59:71:F4:10:3A:07:5B:7E:64:B8:80:40:CD:41:53:...

Miscellaneous

Serial Number

38:F3:02:88:FF:E6:23:6B:11:5D:E1:3A:0D:FE:27:50:96:AC:60:09

Signature Algorithm

SHA-256 with RSA Encryption

Version

1

Download

[PEM \(cert\)](#) [PEM \(chain\)](#)

Fingerprints

SHA-256

CC:7A:36:48:A4:A8:40:40:24:BB:B6:E6:E4:61:4B:09:BB:69:E3:07:C5:5F:1D:55:...

SHA-1

9C:CE:3B:D7:5B:76:1C:EF:75:B1:88:F1:40:13:A3:B7:A2:46:6F:9E

5. Conclusion :

Par le biais de ce TP, nous avons pu créer et configurer un serveur Web en HTTPS avec le service Apache2 tout en gérant les certificats à l'aide d'Open SSL. Nous aurions pu étendre les fonctionnalités en ajoutant une redirection directe de l'HTTP vers l'HTTPS via la configuration du serveur Apache2 et faire le test de connexion avec le site avec Open SSL.