Annexe 1
HTTPS avec Apache V2

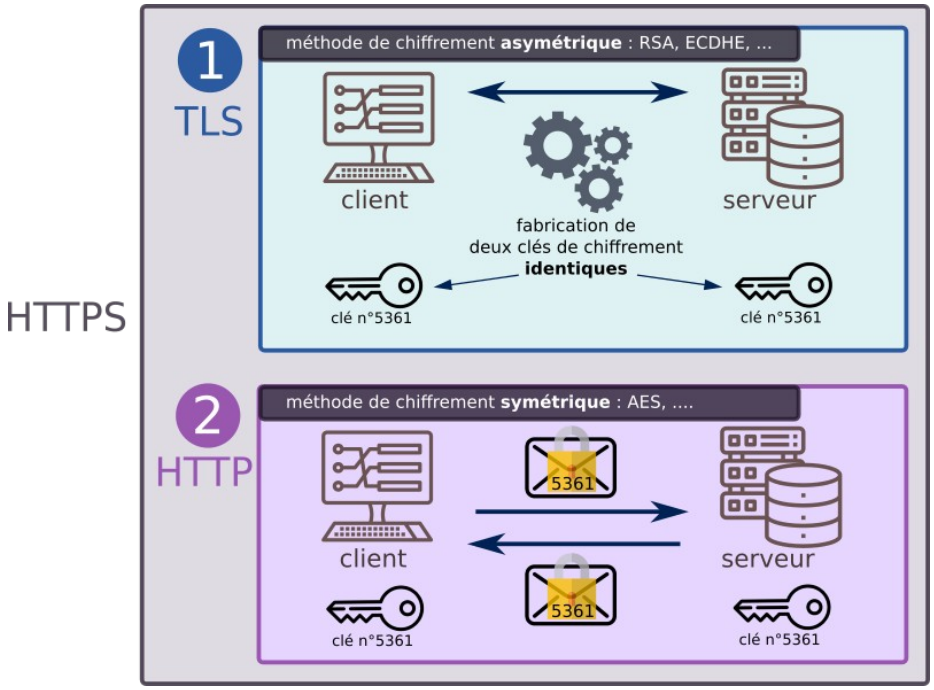
Sommaire :

I - Présentation de HTTPS

Aujourd'hui, plus de 90 % du trafic sur internet est chiffré : les données ne transitent plus en clair (protocole **http**) mais de manière chiffrée (protocole **https**), ce qui empêche la lecture de paquets éventuellement interceptés.

Le protocole **https** est la réunion de deux protocoles :

- le protocole **TLS** (**T**ransport **L**ayer **S**ecurity, qui a succédé au **SSL**) : ce protocole, basé sur du **chiffrement asymétrique**, va conduire à la génération d'une clé identique chez le client et chez le serveur.
- le (bon vieux) protocole **http**, mais qui convoiera maintenant des données chiffrées avec la clé générée à l'étape précédente. Les données peuvent toujours être interceptées, mais sont illisibles. Le **chiffrement symétrique** utilisé est actuellement le chiffrement **AES**.



Dans la majorité des cas, l'utilisateur authentifie le serveur **TLS** sur lequel il se connecte. Cette authentification est réalisée par l'utilisation d'un **certificat numérique X.509** délivré par une **Autorité de Certification** (**AC**) ou **Certificate Authority** (**CA**).

II - Mise en œuvre de HTTPS avec Apache

II.1. Présentation

Il y a 2 méthodes pour mettre en place un site **HTTPS** sous **Apache 2** :
- **Méthode rapide** : Consiste à utiliser les certificats **SSL** par défaut d'**Apache 2** ;
- **Méthode manuelle** : Consiste à générer des certificats **SSL** et de les indiquer dans la configuration d'**Apache 2**.

II.2. Méthode rapide

Par défaut **Apache 2** contient deux sites pré-configurés : « **default** » et « **default-ssl** » qui pointent tous les deux vers le répertoire « **/var/www/html** » mais le premier écoute sur le port **80** (**HTTP**) et le second sur le port **443** (**HTTPS**).

Dans la configuration d'origine, seul le site « **default** » est actif ce qui permet d'accéder à la page « **It Works !** » d'Apache tout de suite après avoir effectué l'installation.

Il suffit d'effectuer 3 choses pour rendre actif et opérationnel le site « **default-ssl** » :
- Activer le module **SSL** d'Apache ;
- Activer le site « **default-ssl** » d'Apache ;
- Relancer Apache.

Voici les commandes à saisir :

#a2enmod ssl
#a2ensite default-ssl
#service apache2 reload

Vous remarquerez qu'il n'y a pas eu besoin de générer de certificat **SSL**. En effet, il y en a déjà un par défaut (valable 10 ans) et on peut voir où il se trouve en regardant de plus près le fichier « **default-ssl** » situé dans « **/etc/apache2/sites-available** ».

II.3. Méthode manuelle

II.3.a. Activation du module SSL

Il faut installer le paquet **openssl** : **#apt-get install openssl**
Et il faut activer le module **ssl** : **# a2enmod ssl**

II.3.b. Création des clés et des certificats

Les **certificats** permettent de fournir diverses informations concernant l'identité de son détenteur. Ce certificat s'accompagne d'une **clé publique** indispensable pour que la communication entre les machines soit chiffrée. De même, afin de garantir l'authenticité du certificat, ce dernier est signé numériquement par le biais d'une **clé dite privée** provenant soit d'un organisme officiel (**Autorité de Certification - AC** ou **Certificate Authority - CA**) soit par le détenteur du certificat lui même. Dans ce dernier cas, on parlera de **certificat auto-signé**.

II.3.c. Création d'un certificat auto-signé

Pour **générer** le **certificat auto-signé**, il faut taper la commande suivante dans un terminal :

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out /etc/apache2/selfsigned.crt -keyout /etc/apache2/selfsigned.key

Explications :
- **x509 -nodes** donne le type de certificat voulu ;
- **days 365** indique la durée de validité (en jours) du certificat ;
- **newkey rsa:2048** demande une clé RSA de 2048 bits - d'après la doc apache, il est déconseillé de créer une clé plus grosse pour des histoires de compatibilité ;
- **out /etc/apache2/selfsigned.crt** est le chemin du certificat ;
- **keyout /etc/apache2/selfsigned.key** est le chemin de la clé privée.

Là on demande des informations régionales :
- **Country Name (2 letter code) [GB]**:
  Saisir **fr** si vous êtes situé en France et valider par la touche « Entrée ».
- **State or Province Name (full name) [Some-State]**:
  Saisir **Landes** et valider par la touche « Entrée ».
- **Locality Name (eg, city) []**:
  Indiquer ici le nom de votre ville. (*exemple :* **AireSurAdour**) et valider par la touche « Entrée ».
- **Organization Name (eg, company; recommended) []**:
  Indiquer le nom de votre organisation, de votre société (*exemple :* **gcrampe**) et valider par la touche « Entrée ». Si vous n'avez pas de société, vous pouvez mettre un nom fictif, le nom de notre site Web par exemple.
- **Organizational Unit Name (eg, section) []**:
  Indiquer ici le nom de la section de votre organisation, de votre société (*exemple :* **bts-informatique**). Si vous n'en avez pas, mettez la même chose que pour la question précédente.
- **Common Name (eg, YOUR name) []**:
  Ici, il convient de faire particulièrement **attention** à ce que vous allez entrer. Vous devez indiquer le *nom de domaine* que vous désirez sécuriser. En ce qui nous concerne, nous saisirons l'adresse du serveur : **192.168.X.1** puis nous validons par la touche « Entrée ».
- **Email Address []**:
  Ici, il s'agit d'indiquer l'adresse E-mail de l'administrateur.

Enfin, on empêche les curieux de lire notre clé privée :
chmod 600 /etc/apache2/selfsigned.*

II.3.d. Création d'un certificat venant d'une CA

Voir **TP3-3**.

II.3.e. Configuration du site virtuel

Par défaut, **Apache2** est configuré pour écouter sur le port **80**. Il s'agit là de la configuration usuelle d'un Serveur Web. Cependant, le protocole **SSL** a besoin d'un port spécifique pour pouvoir fonctionner. Il s'agit du **port 443**. Ce port d'écoute est rajouté automatiquement au fichier **/etc/apache2/ports.conf**.

Il va donc falloir ajouter un **site virtuel** accessible sur le **port 443**. Ce dernier contenant des directives particulières qui sont les suivantes :
- **ServerName** : Cette directive permet de spécifier le nom du site virtuel. On précisera l'adresse du serveur si on utilise le site par défaut **default-ssl** ;
- **SSLEngine** : Cette directive permet d'activer le moteur SSL au sein d'un hôte virtuel, Elle peut prendre deux arguments –> **on/off** ;
- **SSLCertificateFile** : Cette directive définit le **certificat** authentifiant le Serveur auprès des clients. L'argument est le chemin d'accès au certificat. En ce qui nous concerne, le certificat se trouve dans le répertoire **/etc/apache2/** ;
- **SSLCertificateKeyFile** : Cette directive définit la **clé privée** du Serveur utilisée pour signer l'échange de clé entre le client et le serveur. Elle prend en argument le chemin

d'accès à la clé (fichier). Dans notre cas, la clé se trouve dans le répertoire **/etc/apache2/**.

Remarque : On pourra utiliser le site virtuel par défaut **default-ssl** ou créer un nouveau site virtuel.

Par ailleurs, comme nous l'avons déjà fait pour notre hôte virtuel accessible sur le **port 80**, nous allons devoir rajouter une directive **NameVirtualHost** qui permettra que l'adresse nommée par le nom de notre hôte virtuel accessible sur le **port 443** soit résolue correctement.

L'adresse IP du serveur étant ***192.168.40.1*** *(dans votre cas 192.168.<X>.1), n*ous rajouterons donc la directive ***NameVirtualHost 192.168.40.1:443*** au début de notre fichier de configuration. Voici donc le contenu de notre fichier **/etc/apache2/sites-available/www** une fois modifié :

```
NameVirtualHost  192.168.40.1:443
<VirtualHost 192.168.40.1:80>
        ServerName  www
        DocumentRoot /var/www/html/
</VirtualHost>
<VirtualHost 192.168.40.1:443>
        ServerName www
        DocumentRoot /var/www/
        SSLEngine on
        SSLCertificateFile /etc/apache2/server.crt
        SSLCertificateKeyFile /etc/apache2/server.key
</VirtualHost>
```

D'autre part, si l'on veut accéder à 2 dossiers différents (**/var/www** et **/var/www/private**) suivant que l'on accède au serveur en mode sécurisé (**https://www**) ou non (**http://www**), il suffit de renseigner la directive **DocumentRoot** dans chaque section.
Si nous voulons aussi réglementer l'accès à l'espace **/var/www/private** uniquement aux utilisateurs autorisés, il suffira de rajouter une section identique à celle rajoutée précédemment dans le fichier **/etc/apache2/httpd.conf**.
Enfin, si on veut que les clients puissent continuer d'accéder au site Web sécurisé en tapant une **url** de type **http** et non **https**, nous pouvons modifier l'hôte virtuel accessible sur le **port 80** en remplaçant la directive **DocumentRoot** par une directive de redirection (**Redirect**).

Voici donc le contenu de notre fichier **/etc/apache2/sites-available/www** une fois modifié :

```
NameVirtualHost  192.168.40.1:443
<VirtualHost 192.168.40.1:80>
        ServerName  www
        DocumentRoot /var/www/html/
        Redirect / https://www/
</VirtualHost>
<VirtualHost 192.168.40.1:443>
        ServerName www
        DocumentRoot /var/www/private
        SSLEngine on
        SSLCertificateFile /etc/apache2/server.crt
        SSLCertificateKeyFile /etc/apache2/server.key
        <Directory /var/www/private>
                AuthName "Acces Prive au Site www"
                AuthType basic
                AuthUserFile /etc/apache2/.htpasswd
                Require valid-user
        </Directory>
</VirtualHost>
```