ETTORI Bastien	BTS SIO 2ème année
01 Février 2016	Année scolaire : 2015/2016
Option : SISR	Version 2

HTTPS DEBIAN

SOMMAIRE :

ETTORI Bastien	BTS SIO 2ème année
01 Février 2016	Année scolaire : 2015/2016
Option : SISR	Version 2

I) Objectif

Dans cette procédure, nous allons voir comment configurer un serveur Web **Apache** en **HTTPS** sous Debian.

II) Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

OS	Distribution	Version	Nom du serveur
Debian Squeeze	Linux	6.0.6	ks36020.kimsufi.com

III) Définitions

- **Apache2** est un serveur Web qui permet de gérer de manière synchrone plusieurs arborescences Web grâce aux hôtes virtuels.
- Un serveur **HTTPS** (**H**yper**T**ext **T**ransfer **P**rotocol **S**ecure) fonctionne de la même manière qu'un serveur **HTTP** mais en plus, il permet la communication pour un accès à un serveur Web sécurisé.

IV) Installation du serveur Web Apache2

- Tout d'abord, nous mettons à jour les paquets en tapant la commande :

« **apt-get update** ».

- Ensuite, une fois la mise à jour des paquets terminée, nous installons le paquet « **apache2** » :

« **apt-get install apache2** ».

V) Configuration du serveur Web en HTTPS

- Nous créons un fichier nommé « **apache_generate_cert.sh** » :

```
admin@ks36020:~$ nano apache_generate_cert.sh
```

- Nous l'éditons et saisissons le contenu suivant :

```
  GNU nano 2.2.4        Fichier: apache_generate_cert.sh              Modifi

openssl genrsa -out server.key 1024
openssl req -outform PEM -new -key server.key -x509 -days 1825 -out server.crt
```

- Nous lançons le fichier « **apache_generate_cert.sh** » :

```
admin@ks36020:~$ sh apache_generate_cert.sh
```

- Ensuite, ce script demande les propriétés du certificat :

```
admin@ks36020:~$ sh apache_generate_cert.sh
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
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]:FRANCE
Locality Name (eg, city) []:Caen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Blenzik
Organizational Unit Name (eg, section) []:Musique
Common Name (eg, YOUR name) []:ks36020.kimsufi.com
Email Address []:
admin@ks36020:~$
```

- Nous activons le mode **SSL** (**S**ecure **S**ockets **L**ayer) du serveur Web « **apache2** » qui permet la sécurisation des échanges sur Internet et que **SSL** fonctionne avec « **apache2** » :

```
admin@ks36020:~$ sudo a2enmod ssl
[sudo] password for admin:
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
 create self-signed certificates.
Run '/etc/init.d/apache2 restart' to activate new configuration!
admin@ks36020:~$
```

- Enfin, nous redémarrons le service « **apache2** » pour prendre en compte toutes les modifications :

```
admin@ks36020:~$ sudo /etc/init.d/apache2 restart
Restarting web server: apache2 ... waiting .
admin@ks36020:~$
```

- Une fois le service « **apache2** » redémarré, nous vérifions que les fichiers « **server.key** » et « **server.crt** » ont bien été créés et constatons que c'est le cas :

```
admin@ks36020:~$ ls
apache_generate_cert.sh  server.crt  server.key
admin@ks36020:~$
```

  - « **server.crt** » est le fichier de certificat du serveur.
  - « **server.key** » est le fichier de la clé privée du serveur.

- Nous copions ces 2 fichiers de certificats **SSL** dans le dossier « **/etc/ssl/private** » :

```
admin@ks36020:~$ cp server.* /etc/ssl/private/
admin@ks36020:~$
```

- Nous devons mettre à jour les fichiers de configuration d'**apache2** pour utiliser ces fichiers. Pour ce faire, je me rends dans le fichier « **default-ssl** » qui se situe dans le répertoire « **/etc/apache2/sites-available** » :

```
admin@ks36020:~$ sudo nano /etc/apache2/sites-available/default-ssl
```

- Dans le fichier « **default-ssl** », nous modifions les 2 directives (lignes) suivantes en précisant le nom des fichiers de certificats et le dossier où ils ont été copiés :

```
SSLCertificateFile    /etc/ssl/private/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

- Ensuite, nous vérifions la copie des 2 certificats situés dans le répertoire « **/etc/ssl/private/** »

```
admin@ks36020:~$ ls /etc/ssl/private/
server.crt  server.key  ssl-cert-snakeoil.key
admin@ks36020:~$
```

- Nous activons la nouvelle configuration du fichier « **default-ssl** » :

```
admin@ks36020:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
Run '/etc/init.d/apache2 reload' to activate new configuration!
admin@ks36020:~$
```

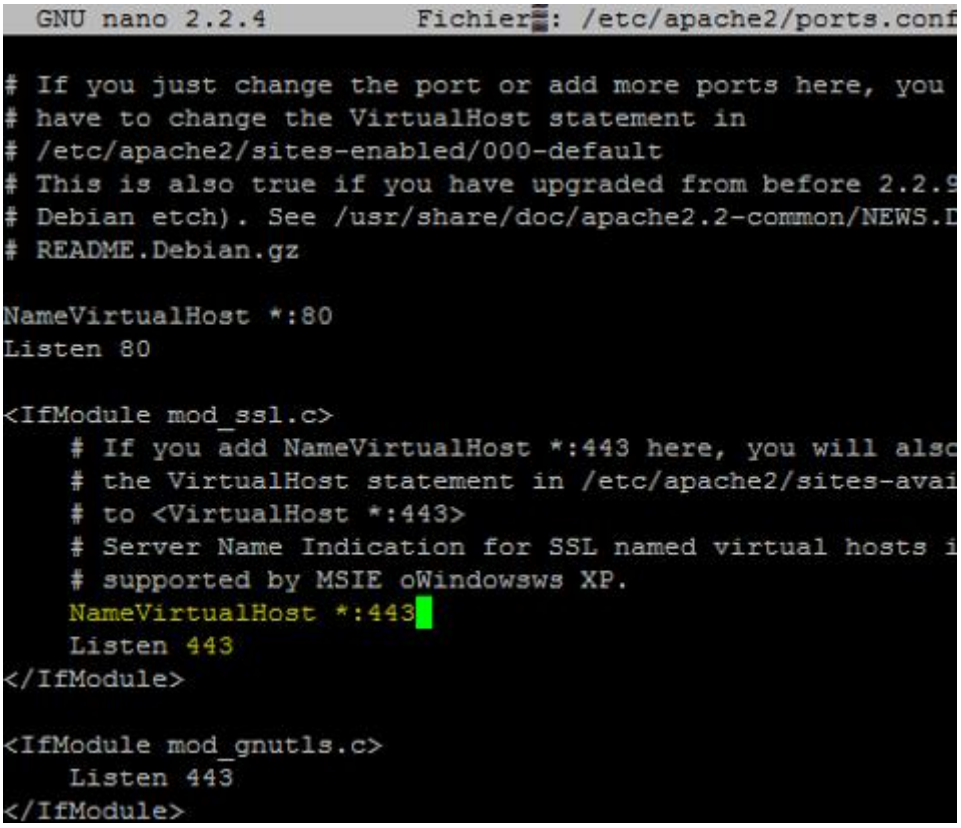
- Nous redémarrons le service « **apache2** » pour confirmer toutes les modifications et vérifier qu'il est activé et remarquons que c'est le cas :

```
admin@ks36020:~$ sudo /etc/init.d/apache2 restart
Restarting web server: apache2 ... waiting .
admin@ks36020:~$ sudo /etc/init.d/apache2 status
Apache2 is running (pid 17768).
admin@ks36020:~$
```

- Nous nous rendons dans le fichier « **ports.conf** » qui se situe dans le répertoire « **/etc/apache2** » :

```
admin@ks36020:~$ sudo nano /etc/apache2/ports.conf
```

- Et, dans ce fichier, nous ajoutons la ligne « **NameVirtualHost *:443** » qui correspond au numéro de port **HTTPS** :



- Nous faisons une copie du fichier « **default-ssl** » au préalable à modifier :

```
admin@ks36020:/etc/apache2/sites-available$ sudo cp default-ssl default-ssl.bak
admin@ks36020:/etc/apache2/sites-available$
```

- Nous nous rendons dans le fichier « **default-ssl** » :

```
admin@ks36020:/etc/apache2/sites-available$ sudo nano default-ssl
```

- Et, dans ce fichier, nous modifions la deuxième ligne en ajoutant le port d'écoute **443** pour le service **HTTPS** :

- De plus, nous pouvons tester pour vérifier si le serveur Web « **apache2** » écoute le port d'écoute **443** correspondant au port **HTTPS** et remarquons que c'est le cas :

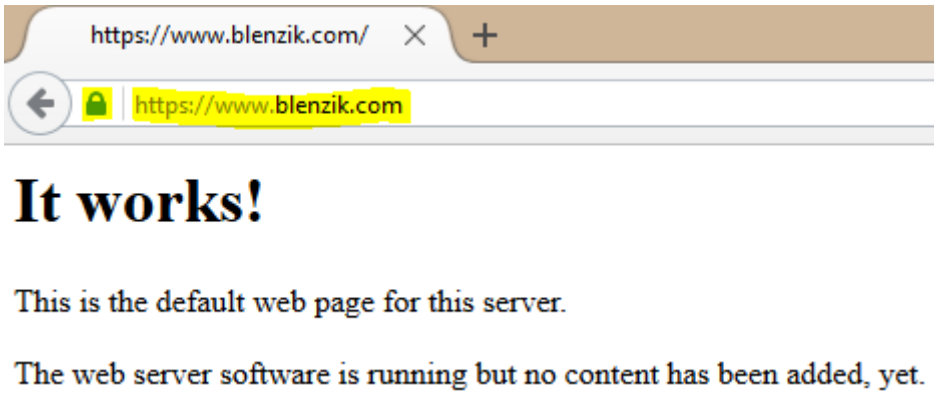
```
admin@ks36020:~$ sudo lsof -i:443
[sudo] password for admin:
COMMAND   PID     USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
apache2  2497 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  5825 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  6751 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  8000 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  8004 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  8005 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  8046 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2  8166 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2 27161 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2 31404 www-data    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
apache2 32117     root    6u  IPv6 190574975      0t0  TCP *:https (LISTEN)
admin@ks36020:~$
```

- Enfin, nous testons en saisissant dans l'URL d'un navigateur Web sous cette forme :

  - « https://nom_du_serveur ».

Ici, la configuration en **HTTPS** fonctionne mais la connexion sécurisée au serveur n'est pas encore certifiée.

- Voici le résultat dans le navigateur **Mozilla Firefox** :



- Dans **VirtualHost** (**Webmin**), nous créons une entrée « **dev.blenzik.fr** » sur le port **443** (**SSL**).

- Dans **Webmin**, nous cliquons sur l'icône **SSL**, je l'active et je lie les certificats **SSL** du répertoire « **/etc/ssl/private** ».

- Puis, nous activons le module « **Rewrite** » : « **a2enmod rewrite** » pour permettre au serveur Apache de gérer la réécriture afin d'améliorer le référencement des pages d'un site Web.

- Ensuite, nous nous rendons dans le fichier « **.htaccess** » qui se situe dans le répertoire « **/home/blenzik/wwwdev** » :

```
admin@ks36020:~$ sudo nano /home/blenzik/wwwdev/.htaccess
```

- Dans ce fichier, nous ajoutons la ligne « **Options +SymLinksIfOwnerMatch** » qui permet de vérifier les liens symboliques si le fichier ou le répertoire racine appartient au même utilisateur que le lien.

- Nous activons le « **Rewrite** » en ajoutant la ligne « **RewriteEngine On** » pour lancer le module de réécriture d'URL :

```
  GNU nano 2.2.4      Fichier : /home/blenzik/wwwdev/.htaccess

# Activation du Rewriting
#
RewriteEngine On
```

- Nous ajoutons la ligne « **RewriteCond %{HTTP_HOST} ^dev.blenzik\.fr [NC]** » pour ainsi rediriger l'URL « **dev.blenzik.fr** » vers le protocole **HTTPS**, soit « **https://dev.blenzik.fr/$1** ».

- Nous ajoutons la ligne « **RewriteCond %{SERVER_PORT} 80** » pour rediriger le port **HTTP** : **80** du serveur.

- Nous ajoutons la ligne « **RewriteRule ^(.*)$ https://dev.blenzik.fr/$1 [R,L]** » pour mettre en place une règle de sécurité. Cette règle permet de valider la condition et de renvoyer vers le même URL sur le protocole **HTTPS**.

- Enfin, nous devons acheter le certificat **SSL** adapté pour confirmer la configuration du **HTTPS** et sécuriser le site Web.

VI) Conclusion

Une fois que le certificat **SSL** est acheté, la configuration **HTTPS** sera complète. Donc, en conclusion, nous pouvons dire que le serveur Web « **apache2** » est correctement configuré en **HTTPS** et que celui-ci permet de communiquer de manière sécurisée.