

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

# HTTPS

## SOMMAIRE :

I)	Objectif.....	2
II)	Prérequis.....	2
III)	Définitions.....	2
IV)	Installation du serveur Web Apache2.....	2
V)	Configuration du serveur Web en HTTPS.....	2-7
VI)	Conclusion.....	7

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

## 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 :

Nombre de postes	SE serveur	Nom serveur
1	Debian 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

- D'abord, nous mettons à jour les paquets en tapant la commande :  
« **apt-get update** ».
- Une fois la mise à jour des paquets terminée, nous installons le paquet « **apache2** ». Pour ce faire, nous tapons :  
« **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 nous saisissons le contenu suivant :

```
GNU nano 2.2.4      Fichier: apache_generate_cert.sh      Modifié
openssl genrsa -out server.key 1024
openssl req -outform PEM -new -key server.key -x509 -days 1825 -out server.crt
```

- Je lance le fichier « **apache\_generate\_cert.sh** » :

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

- Ensuite, ce script demande les propriétés du certificat :

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

```

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 (Secure Sockets Layer)** du serveur Web « **apache2** » qui permet la sécurisation des échanges sur Internet et que **SSL** fonctionne avec « **apache2** » : « **sudo a2enmod ssl** » :

```

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 : « **sudo /etc/init.d/apache2 restart** » :

```

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

```

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

```

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:~$ █

```

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

- 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 de ces 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 :

```
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** :

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

```

GNU nano 2.2.4      Fichier: /etc/apache2/ports.conf
# If you just change the port or add more ports here, you w
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default
# This is also true if you have upgraded from before 2.2.9-
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.De
# README.Debian.gz

NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also
    # the VirtualHost statement in /etc/apache2/sites-avail
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is
    # supported by MSIE oWindowsws XP.
    NameVirtualHost *:443
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

```

- 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 vérifions que ce fichier est bien présent : « **ls** » :

```

admin@ks36020:/etc/apache2/sites-available$ ls
aquaman-dts.com.conf      www.aquaman-dts.c
boutique.subpassion.com.conf  www.borelec.com.c
cyclesandco.com.conf      www.cyclesandco.c
cyclesandco.fr.conf       www.cyclesandco.f
default                   www.deadcircles.c
default-ssl               www.festivaldelag
default-ssl.bak           www.fnpsa-normand

```

- 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 **HTTPS** :

```

GNU nano 2.2.4      Fichier: default-ssl
<IfModule mod_ssl.c>
<VirtualHost *:443>

```

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

- De plus, nous pouvons tester pour vérifier si le serveur Web « **apache2** » écoute bien le port d'écoute **443** correspondant au port **HTTPS** : « **sudo lsof -i:443** » :

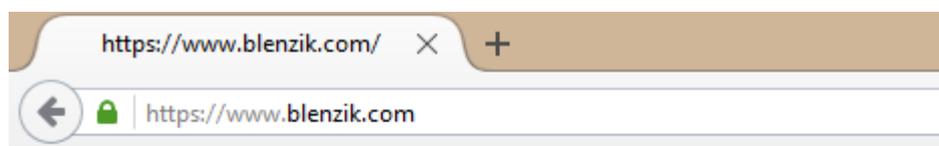
```
admin@ks36020:~$ sudo lsof -i:443
[sudo] password for admin:
COMMAND  PID    USER   FD   TYPE    DEVICE  SIZE/OFF  NODE  NAME
apache2  2497  www-data  6u   IPv6  190574975      0t0  TCP  *:https (LISTEN)
apache2  5825  www-data  6u   IPv6  190574975      0t0  TCP  *:https (LISTEN)
apache2  6751 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



**It works!**

This is the default web page for this server.

The web server software is running but no content has been added, yet.

- 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.

ETTORI Bastien	BTS SIO 2 <sup>ème</sup> année
01 février 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

- 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** » :

```
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](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. Nous pouvons dire que le serveur Web « **apache2** » est correctement configuré en **HTTPS** et qu'il permet donc de communiquer de manière sécurisée.