Titre	Installation d'un serveur web Apache2 SSL sous Ubuntu
Propriétaire	Tavares José
Classification	Public
Date dernière modification	01 Octobre 2009
Chemin\NomFichier	\\10.1.1.1\FilesTD\Group4\Personnel\Tavares\00_EIG\Sécuriser_SSL_Apache.doc

OBJECTIF 1

1 INSTALLER UBUNTU 2

2 EFFECTUER LES MISES À JOUR SYSTÈME 2

3 CONFIGURER LES INTERFACES RÉSEAU (IP + DNS) 2

4 INSTALLER APACHE2 WEB SERVER 2

5 CONFIGURER SSL DANS APACHE2 WEB SERVER 2

6 CRÉER UNE DEMANDE DE CERTIFICAT SERVEUR 3

7 STOCKER LE CERTIFICAT SERVEUR ET SA CLÉ PRIVÉE 3

8 CRÉER UN NOUVEAU SITE WEB SSL 4

9 SÉCURISER APACHE2 WEB SERVER 5

9.1 Cacher la bannière d'Apache2 et les informations OS 5
9.2 Voir avec quel compte s'exécute Apache2 6
9.3 Droits du dossier /etc/apache2 6
9.4 Restreindre les accès web 6
9.5 Désactiver le module qui gère les scripts CGI 8
9.6 Aller plus loin 8
9.7 Quelques liens 8

Remarques

- Le **#** précédent les commandes indique qu'il faut des droits root (sudo)
- Le **$** précédent les commandes indique qu'il faut de simples droits utilisateur

Objectif

Ce document répertorie les bonnes pratiques pour l'installation et la configuration sécurisée d'un serveur web **statique** Apache2 sous Ubuntu, avec une configuration SSL

1 Installer Ubuntu

Le choix c'est porté sur la version Ubuntu Server 8.04 LTS (Long Time Support), qui offre un support de 3 ans (contre 1 an pour les versions non-LTS)

Suivre le premier § de
https://help.ubuntu.com/8.04/serverguide/C/installing-from-cd.html

Attention, après le Boot sur CD, appuyer sur F3 pour choisir la langue clavier

2 Effectuer les mises à jour système

```
# apt-get update
# apt-get upgrade
```

https://help.ubuntu.com/8.04/serverguide/C/apt-get.html

Il est conseillé d'effectuer ces mises à jour régulièrement (tous les mois)

3 Configurer les interfaces réseau (IP + DNS)

https://help.ubuntu.com/8.04/serverguide/C/network-configuration.html

Attention, il faut ouvrir les fichiers en questions avec les droits root (sudo)

4 Installer Apache2 Web Server

```
# apt-get install apache2
```

https://help.ubuntu.com/8.04/serverguide/C/httpd.html
http://httpd.apache.org/docs/2.2/

5 Configurer SSL dans Apache2 Web Server

```
# apt-get install libapache2-mod-jk
# apt-get install libapache2-mod-proxy-html
# a2enmod ssl
# /etc/init.d/apache2 stop
# /etc/init.d/apache2 start
```

http://doc.ubuntu-fr.org/tutoriel/securiser_apache2_avec_ssl

SSL est à présent activé, il reste à créer un certificat serveur.
Pour cela il faut créer une paire de clés (privée + publique)

6 Créer une demande de certificat serveur

Les commandes ci-dessous génèrent une CSR (*certificate signing request*) qui doit être soumise à une autorité de certification (CA) afin d'être signée et enfin nous délivrer le certificat serveur

```
$ openssl genrsa -des3 -out server.key 1024
$ openssl req -new -key server.key -out server.csr
```

https://help.ubuntu.com/8.04/serverguide/C/certificates-and-security.html

La partie OpenSSL est traitée en laboratoire et n'est pas expliquée dans ce document

7 Stocker le certificat serveur et sa clé privée

```
$ mv /home/eig/server.cer /etc/ssl/certs/server.csr
$ mv /home/eig/server.key /etc/ssl/private/server.key
```

Voir avant-dernier § de
https://help.ubuntu.com/8.04/serverguide/C/certificates-and-security.html

Attention, il est conseillé de déplacer la clé privée, et non pas de la copier comme dans le lien ci-dessus, cette clé doit être unique et ne peux être lue que par un compte root

Le dossier **/etc/ssl/private** a été spécialement conçu pour stocker la(les) clé(s) privée(s), car celui-ci peut uniquement être lu par un compte root (ce dossier est créé automatiquement lors de l'installation d'apache avec ssl)

```
$ cd /etc/ssl
$ ls -lha
drwxr-xr-x  4 root root 4.0K 2009-09-08 15:49 .
drwxr-xr-x 65 root root 4.0K 2009-09-24 15:04 ..
drwxr-xr-x  2 root root 4.0K 2009-09-24 16:31 certs
-rw-r--r--  1 root root 9.2K 2009-09-09 09:41 openssl.cnf
drwx------  2 root root 4.0K 2009-09-24 16:32 private
```

Si on regarde les permissions de sécurité des fichiers contenus dans /etc/ssl/private :

```
# cd /etc/ssl/private
# ls -lha
drwx------ 2 root root 4.0K 2009-09-24 16:32 .
drwxr-xr-x 4 root root 4.0K 2009-09-08 15:49 ..
-rw-r--r-- 1 root root  887 2009-09-14 15:55 server.key
```

Sous Linux, les permissions ne sont pas héritées !
On constate que par défaut, la clé privée peut être lues par n'importe qui, cependant, seul le compte root peut accéder au contenu du dossier /etc/ssl/private et donc la clé ne peuvent être lues que par ce dernier !

Le problème qui peut se poser :
Si une clé privée est déplacée dans un autre dossier qui autorise la lecture de simples utilisateurs, ceux-ci auront accès à la clé, surtout si celle-ci n'est pas protégée par mot de passe. Par sécurité, on modifie également les permissions des clés privées :

```
# chmod 400 /etc/ssl/private/server.key
```

Résultat

```
# ls -lha /etc/ssl/private
-r-------- 1 root root  887 2009-09-14 15:55 server.key
```

Remarque : au cas où la clé n'appartient pas à root :

```
# chown root:root /etc/ssl/private/server.key
```

Voir fin du §5 :
http://aide.sivit.fr/index.php?2005/06/27/37-mse-en-place-d-apache-2-ssl

8 Créer un nouveau site web SSL

Le nouveau site web ci-dessous sera créé sur le port TCP 1234

```
# cd /etc/apache2/sites-available
# touch ssl-1234
# nano ssl-1234
```

Dans ce fichier, on crée la configuration SSL du site web

```
Listen 1234
<VirtualHost 129.194.184.99:1234>
        ServerName mon_site_ssl.abc.ch
        SSLEngine On
        SSLCertificateFile /etc/ssl/certs/server.cer
        SSLCertificateKeyFile /etc/ssl/private/server.key
        DocumentRoot /var/www/secure1234
        ErrorLog /var/log/apache2/error.log
</VirtualHost>
```

On ajoute le site web à Apache2

```
# a2ensite ssl-1234
# /etc/init.d/apache2 stop
# /etc/init.d/apache2 start
```

Voir §HTTPS Configuration
https://help.ubuntu.com/8.04/serverguide/C/httpd.html

Divers exemples
http://httpd.apache.org/docs/2.0/vhosts/examples.html

9 Sécuriser Apache2 Web Server

9.1 Cacher la bannière d'Apache2 et les informations OS

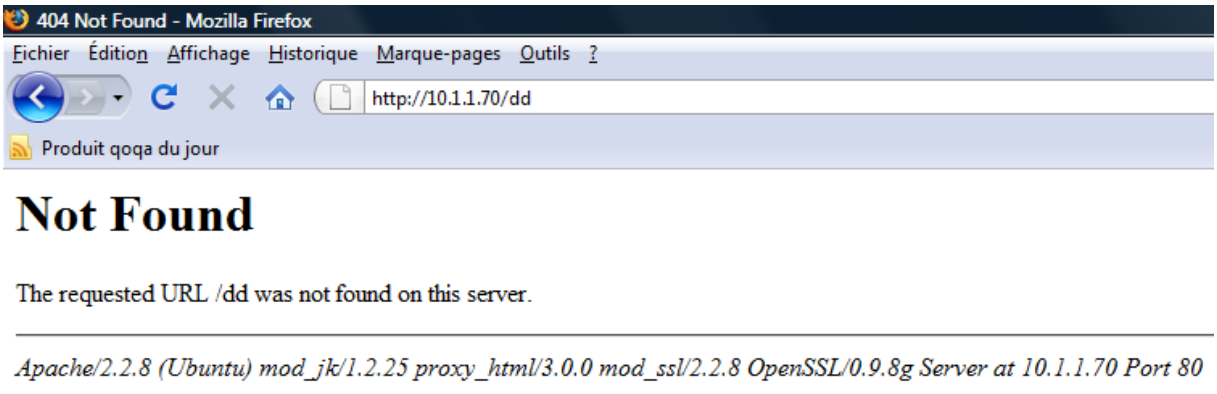
Pour cacher les informations renvoyées par le serveur, concernant la version d'Apache2 et l'OS installé :

```
# nano /etc/apache2/apache2.conf
```

Chercher et corriger les lignes suivantes dans le fichier :

```
ServerTokens Prod
ServerSignature Off
```

Test avant :



Test après :

On constate que le serveur est beaucoup moins parlant, il indique tout de même qu'il s'agit d'un serveur apache. Ceci peut être masqué en recompilant Apache, cependant ceci empêchera les futures mises à jour d'Apache !

Il faudra en effet recompiler chaque nouvelle version d'apache si l'on souhaite masquer ce paramètre, la mise à jour via apt-get n'étant plus possible…

Voir §1
http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

9.2 Voir avec quel compte s'exécute Apache2

```
# cat /etc/apache2/envvars
```

Qui donne :

```
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
```

L'utilisateur www-data possède uniquement des droits utilisateur, si un hacker arrive à compromettre le service d'Apache2, il n'aura qu'un impact limité

Voir §2
http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

9.3 Droits du dossier /etc/apache2

Par défaut :

```
# ls -lha /etc | grep apache2
drwxr-xr-x  7 root root   4.0K 2009-08-31 16:09 apache2
drwxr-xr-x  2 root root   4.0K 2009-09-01 09:48 libapache2-mod-jk
```

Le compte root peut lire écrire exécuter, le groupe root peut lire exécuter, tout le reste ne doit avoir aucun accès :

```
# chmod -R 750 /etc/apache2
# ls -lha /etc | grep apache2
drwxr-x---  7 root root   4.0K 2009-08-31 16:09 apache2
drwxr-xr-x  2 root root   4.0K 2009-09-01 09:48 libapache2-mod-jk
```

Voir §6
http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

9.4 Restreindre les accès web

On commence par restreindre les accès qui peuvent se faire à la racine du site web :

```
# nano /etc/apache2/sites-available/default
```

On cherche la partie racine (options par défaut) :

```
<Directory />
        Options FollowSymLinks
        AllowOverride None
</Directory>
```

On applique les restrictions:

```
<Directory />
        Order Deny,Allow
        Deny from all
        Options None
        AllowOverride None
</Directory>
```

Ceci empêche à un utilisateur de parcourir la racine du site, désactive toutes les options (telles que les scripts CGI, le suivi de liens symboliques, …)

On se doit aussi de restreindre tous les autres VirtualHost, comme le site web disponible par défaut à l'installation sur le port 80.

```
<Directory /var/www/>
        Order Allow,Deny
        Allow from all
        Options None
        AllowOverride None
</Directory>
```

Par défaut, quand il n'y a pas de page index.html, on peut lister le contenu du site web :

Après les restrictions :

Le même principe est à effectuer pour chaque VirtualHost

Voir §3-4
http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

9.5 Désactiver le module qui gère les scripts CGI

```
# a2dismod cgi
```

Voir §5
http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

9.6 Aller plus loin

Il est possible d'effectuer un chroot d'Apache2, censé encore améliorer la sécurité.

L'inconvénient de cette méthode, c'est qu'il n'est plus possible d'effectuer les mises à jour d'Apache2 via apt-get

Voir
http://www.linux-pour-lesnuls.com/chroot.php

Il semble aussi possible de modifier les fichiers renvoyés en cas d'erreur, voir
http://www.brakstar.com/forum/braktopic_106.html

9.7 Quelques liens

http://pro.anapivirtua.com/index.php/2009/06/29/le-petit-guide-de-lapache2-securise-en-15-points/

http://doc.ubuntu-fr.org/tutoriel/securiser_apache2

http://www.system-linux.eu/index.php?category/Apache2