

0.1 Apache avec mod_ssl

Apache 1 (actuellement en version 1.3.27) est un serveur HTTP maintenu par l'Apache Group. Apache est le fruit de l'effort de webmasters qui développèrent chacun des extensions au serveur le plus populaire de l'époque (le serveur HTTP du NCSA), puis décidèrent de former l'Apache Group afin de mettre en commun leurs efforts et de produire le serveur HTTP le plus robuste et le plus stable jamais connu. Apache est conforme au protocole HTTP (utilisé pour la communication entre les navigateurs et le serveur) et reconnaît également les requêtes HTTP/1.1, dernières extensions à HTTP. Il intègre le concept de modules, ce qui rend facile le développement d'extensions par la communauté du logiciel libre. Cette facilité donne à Apache un grand nombre de fonctionnalités uniques (voir l'URL <http://modules.apache.org> pour une liste des modules développés). Associé à PHP, il devient un puissant serveur de pages web dynamiques. Dans ce document nous allons détailler l'installation d'apache avec mod_ssl, pour apache 1. Il existe un Apache 2, qui apporte de nombreuses améliorations par rapport à cette version.

0.1.1 1. Installation binaire d'apache :

J'explique ici l'installation d'apache depuis les paquets de votre distribution. Si votre distribution n'est pas étudiée, recherchez dans vos cdroms des paquets aux noms similaires, ou utilisez l'installation à partir des sources. Dans tous les cas, la configuration qui suivra, fera la part belle aux sources, par soucis de généricité. Reportez-vous à la doc de votre distribution pour les détails.

- **Pour les utilisateurs de Mandriva** : Installez les paquets apache, apache-devel, apache-conf, apache-manual, apache-mod_perl, apache-modules
- **Pour les utilisateurs de RedHat** : Installez les paquets apache-common, apache-devel, apache-modules, apache-conf, apache et apache-manual.
- **Pour les utilisateurs de Debian** : Installez les paquets apache et apache-common

0.1.2 2. Apache + ssl à partir des sources :

0.1.3 2.1 Pré-requis :

- Est-il nécessaire de préciser que tous, mais vraiment tous les paquets, se rapportant de près ou de loin à apache, doivent être désinstallés auparavant ?
- les paquets openssl, zlib, perl, flex, bison, autoconf, libtool, awk et freetype2(-devel)
- Installez expat¹

Vous trouverez la version la plus à jour d'Apache 1, ICI² ou LA³. Pour fixer les idées, je désarchiverai toutes mes sources dans un même répertoire, je vous conseille de faire pareil. Dans mon cas il s'agit de /tmp/apache, créez-le et décompressez apache :

```
# mkdir /tmp/apache
# cd /tmp/apache/
# tar -xzf /où_est/apache_1.3.27.tar.gz
```

¹ <http://www.trustonme.net/didactels/211.html#expat>

² <http://www.apache.org/dist/httpd/>

³ <http://apache.crihan.fr/dist/httpd/>

0.1.4 2.2 Installation de mm :

La librairie MM est une librairie qui simplifie l'utilisation de la mémoire partagée entre plusieurs processus "forkés" sous Unix. Vous pouvez télécharger la version la plus à jour ICI⁴. Son installation est assez simple :

```
# cd /tmp/apache
# tar -xzf /où_est/mm-1.3.0.tar.gz
# cd mm-1.3.0/
# ./configure --prefix=/usr
# make && make install
```

0.1.5 2.3 Installation de mod_ssl :

Mod_ssl est le module ssl pour apcahe. J'installe mod_ssl statiquement. Commencez par télécharger la dernière version de mod_ssl, correspondante à votre apache ICI⁵. Pour l'installation : # cd /tmp/apache # tar -xzf /où_est/mod_ssl-2.8.14-1.3.27.tar.g
cd mod_ssl-2.8.14-1.3.27/ # ./configure -with-apache=/tmp/apache/apache_1.3.27 -enable-shared=ssl -with-mm=/tmp/apache/mm-1.3.0

0.1.6 2.4 Installation d'apache :

J'ai choisi de mettre les gros fichiers d'apache dans la partition /home/, parce que c'est la plus grosse chez moi. Vous êtes libres, d'adapter à votre convenance. La compilation d'apache se résume à : # cd /tmp/apache/apache_1.3.27/ # EAPI_MM="/tmp/apache/mm-1.3.0" SSL_BASE="/usr" ./configure -prefix=/usr -exec-prefix=/usr -bindir=/usr/sbin -sbindir=/usr/sbin -libexecdir=/usr/sbin -datadir=/usr/share -includedir=/usr/share -mandir=/usr/man -sysconfdir=/etc/httpd/conf -localstatedir=/home/log/httpd -htdocsdir=/home/httpd/html -cgidir=/home/httpd/cgi-bin -iconsdir=/home/httpd/icons -enable-module=all -enable-module=so -enable-shared=max # make
Maintenant vous devez générer des certificats auto-signés. Dans la plus part des cas vous devez taper :

```
# make certificate TYPE=dummy
```

Si vous souhaitez utiliser, un ancien certificat se sera : # make certificate TYPE=existing CRT=/où_est_votre/server.crt KEY=/où_est_votre/server.key Finissez par :

```
make install
```

Vérifiez que vous avez bien les fichiers :

```
/etc/httpd/conf/ssl.crt/server.crt
/etc/httpd/conf/ssl.key/server.key
```

Une petite description de l'arborescence d'apache : Vos pages web seront dans : **/home/httpd/html/**, les exécutable d'apache dans **/usr/bin/**, les mods sont dans **/usr/sbin/**, les logs dans **/home/log/httpd/**, les fichiers de conf dans **/etc/httpd/conf/**, les scripts cgi dans **/home/httpd/cgi/** et les icônes dans **/home/httpd/icons/** Apache ne doit pas être lancé par root, il est préférable d'utiliser

⁴<http://www.osspp.org/pkg/lib/mm/>

⁵<http://www.modssl.org>

pour cela un utilisateur sans droit. Pour ce faire, nous utiliserons : nobody, présent sur la plus part des distributions. Vérifiez sa présence par :

```
# cat /etc/passwd | grep nobody
```

Si cette commande restait sans réponse c'est que vous n'avez pas d'utilisateur nobody. Vous pouvez le créer par :

```
# groupadd nobody
# useradd -g nobody -d /home/httpd nobody
```

Terminez en attribuant le répertoire /home/httpd et son contenu à nobody :

```
chown -R nobody.nobody /home/httpd/
```

0.1.7 2.5 Configuration d'apache :

- le fichier : /etc/httpd/conf/httpd.conf C'est le fichier de configuration d'apache, il est assez illisible. Pour le nettoyer , fermez-le, puis tapez les lignes suivantes dans un terminal :

```
# cd /etc/httpd/conf/ # mv httpd.conf httpd.conf.old
# awk '! (/^#/ || /$/ ) { print $0 }' httpd.conf.old > httpd.conf
```

Maintenant éditez votre /etc/httpd/conf/httpd.conf, c'est pas plus lisible ainsi ? vous pouvez revenir à tout moment à la forme avec commentaire en écrasant httpd.conf par httpd.conf.old. Vérifiez maintenant que vous avez bien les options suivantes (les autres options peuvent conserver leur valeur par défaut) :

```
# le serveur est un daemon à part entière
ServerType standalone
# répertoire racine pour les chemin relatifs
ServerRoot "/usr"
# fichier contenant le pid d'apache
PidFile /var/log/httpd/logs/httpd.pid
# temps limite des transactions http
Timeout 300
# le port http
Port 80
# laissez la valeur par défaut
User nobody
# laissez la valeur par défaut
Group nobody
# adresse mail du webmaster
ServerAdmin Dieu@trustonme.net
# répertoire ou sont disposés les pages web
DocumentRoot "/home/httpd/html"
# journal d'erreur
ErrorLog /var/log/httpd/logs/error_log
# full permet d'avoir plus de détails
CustomLog /var/log/httpd/logs/access_log full
```

Pour les Options entre balises :

```
# voir plus bas
UserDir public_html
# Configuration des index
DirectoryIndex index.html index.htm
```

Rajoutez également, ceci en fin de fichier :

```
# pour avoir des infos à chaud sur le serveur
<Location /server-status>
SetHandler server-status
Order Deny,Allow
Allow from localhost
</Location>
```

Ceci permet d'avoir un apache fonctionnel, pour un usage avancé, reportez-vous à la documentation d'apache, ou à ce qui suit (public_html).

- public_html : Grâce à la directive "**UserDir public_html**" (dans le fichier /etc/httpd/conf/httpd.conf) chaque utilisateur de votre PC a la possibilité, d'avoir un site perso, accessible à `http://localhost/~le_login/`⁶ (une fois qu'apache est lancé). Chaque utilisateur souhaitant en profiter devra, taper les lignes suivantes :

```
$ mkdir ~/public_html
$ chmod 755 ~
$ chmod 755 ~/public_html
```

Il pourra ensuite y mettre les fichiers de son site web. Apache étant, lancé avec les droits de nobody, il convient de de lui donner le droit de lire les fichiers dans ~/public_html/, chaque utilisateur devra donc après y avoir mis des fichiers taper :

```
$ chmod -R 755 ~/public_html/*
```

Notez que, sur la plus part des distributions, lorsque vous créez un répertoire, il est déjà en 755, seuls les fichiers peuvent nécessiter un ajustement des droits. Apache permet bien plus de chose, comme les virtuals hosts qui vous permettent d'héberger plusieurs site avec la même adresse IP ou encore restreindre l'accès à certains fichiers/répertoires. N'hésitez pas à lire la doc pour en connaître toutes les subtilités.

0.1.8 2.6 Lancement d'apache :

Pour lancer apache sans support ssl tapez maintenant :

```
# /usr/sbin/apachectl start
```

Pour lancer apache avec support ssl tapez :

```
# /usr/sbin/apachectl startssl
```

/usr/bin/apachectl accepte également les arguments, "stop" pour l'arrêt et "restart" pour le redémarrer. Vous pouvez également utiliser les arguments "status" ou "fullstatus" si et seulement si vous avez le navigateur lynx installé. Pour tester votre installation d'Apache, tapez dans un terminal :

```
# cd /home/httpd/html/
# mv manual/index.html.fr index.html
```

⁶http://localhost/~le_login/

Puis dans un navigateur tapez <http://localhost/>⁷ Pour qu'apache soit lancé à chaque démarrage, rajoutez les lignes suivantes dans votre /etc/rc.d/rc.local :

```
if [ -x /usr/sbin/apachectl ]; then
  echo "Demarrage d'Apache ..."
  /usr/sbin/apachectl start
fi
```

Adaptez, si vous souhaitez activer le support SSL. Vous trouverez de la documentation sur apache [ICI](#)⁸

⁷<http://localhost/>

⁸http://www.geldenhuis.com/Manuels/Apache_1.3/