

0.1 Apache 2 et SSL

Apache 2 est la nouvelle version du serveur Web le plus populaire. Cette nouvelle version apporte un support amélioré des threads et une meilleure interaction avec les Systèmes d'exploitation non Unix. L'API du serveur Apache a été profondément remaniée, ce explique que la plupart des extensions pour Apache 1, ne fonctionneront pas avec Apache 2. Dans ce document, je vais indiquer comment installer et configurer Apache 2 avec support SSL. Si vous cherchez un didacticiel sur Apache 1, regardez ICI¹.

0.1.1 1. Apache + ssl avec les paquetages de sa distribution

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.

- Pour les utilisateurs de Mandriva, tapez dans un terminal en root :

```
# urpmi apache2 apache2-common apache2-devel apache2-manual apache2-  
modules apache2-conf apache2-mod_ssl
```

- Pour les utilisateurs de Fedora, tapez dans un terminal en root :

```
# yum install httpd httpd-devel httpd-manual mod_ssl
```

- Pour les utilisateurs de Debian, tapez dans un terminal en root :

```
# apt-get install apache2 apache2-common apache2-doc  
# a2enmod ssl
```

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 ou httpd, doivent être désinstallés auparavant ?
- les paquets openssl, zlib, flex, bison, autoconf, libtool, awk, expat et freetype2 doivent être présents sur votre système. Installez également les -dev(el) qui correspondent à ces paquetages lorsqu'il y'en a.
- Téléchargez la version la plus à jour d'Apache 2 sur le site officiel²

0.1.4 2.2 Installation d'apache

Décompressez l'archive d'apache que vous avez téléchargée, comme suite :

```
$ cd /tmp  
$ tar -xvzf /où_est/httpd-2.0.xx.tar.gz  
$ cd httpd-2.0.xx
```

Éditez maintenant le fichier `/tmp/httpd-2.0.xx/config.layout`. Là vous voyez plusieurs layout, par défaut la compilation utilisera le layout "Apache". Modifiez-le comme suite :

¹<http://www.trustonme.net/didactels/110.html>

²<http://httpd.apache.org/download.cgi>

```
<Layout Apache>
prefix : /usr
exec_prefix : ${prefix}
bindir : ${prefix}/bin
sbindir : ${prefix}/sbin
libdir : ${prefix}/lib
libexecdir : ${prefix}/lib/apache
mandir : ${prefix}/man
sysconfdir : /etc/httpd/conf
datadir : /home/httpd
installbuilddir : ${datadir}/build
errordir : ${datadir}/error
iconsdir : ${datadir}/icons
htdocsdir : ${datadir}/htdocs
manualdir : ${datadir}/manual
cgidir : ${datadir}/cgi-bin
includedir : ${prefix}/include/apache
localstatedir : /var
runtimedir : ${localstatedir}/run
logfiledir : ${localstatedir}/log/httpd
proxycachedir : ${localstatedir}/cache/httpd
</Layout>
```

Dans ce layout, j'ai choisi d'installer les données Web d'apache dans `/home/httpd`, car `/home` est la partition la plus large de mon disque dur. Si vous souhaitez faire différemment, modifiez la valeur attribuée à `datadir`. Notez également que la racine de mon site sera `/home/httpd/htdocs`. Finissez l'installation ainsi :

```
$. /configure --enable-so --enable-rewrite=shared --enable-deflate=shared --enable-ssl=shared --enable-cgi=shared
$ make
$ su
# make install
```

0.1.5 3. Configuration de l'environnement

0.1.6 3.1 Convention de nommage

Pour rendre le didacticiel générique, dans la suite j'appellerai :

- Le dossier de configuration d'Apache `$CONF`. Si vous avez installé Apache à partir des sources et exactement comme moi, ce sera `/etc/httpd/conf`.
- Le script d'initialisation d'Apache `$APACHECTL`. Si vous avez installé Apache à partir des sources et exactement comme moi, ce sera `/usr/sbin/apachectl`.
- Le dossier des modules d'Apache `$MODULES`. Si vous avez installé Apache à partir des sources et exactement comme moi, ce sera `/usr/lib/apache`.

- Le dossier contenant les données Web d'apache \$WEB. Si vous avez installé Apache à partir des sources et exactement comme moi, ce sera /home/httpd. Notez que la racine de votre serveur Apache est suivant cette convention \$WEB/htdocs ou \$WEB/html. Dans mon cas c'est \$WEB/htdocs.

A chaque fois que vous verrez l'une de ces variables dans la suite, vous la remplacerez par sa valeur chez vous.

0.1.7 3.2 Gestion des droits

Apache ne doit pas être lancé par root, il est préférable d'utiliser pour cela un utilisateur sans droit. Pour ce faire, nous utiliserons : nobody, présent sur la plupart des distributions. Vérifiez sa présence par :

```
# grep nobody /etc/passwd
```

A partir de là, deux cas possibles :

- Si cette commande reste sans réponse, c'est que vous n'avez pas d'utilisateur nobody. Vous pouvez le créer par (remplacez \$WEB par sa valeur) :

```
# groupadd nobody
```

```
# useradd -g nobody -d $WEB nobody
```

- Si cette commande donnait une réponse mais que le dossier attribué à cet utilisateur était autre chose que \$WEB vous pouvez changer cela en tapant (remplacez \$WEB par sa valeur) :

```
# usermod -d $WEB nobody
```

Terminez en attribuant le répertoire \$WEB et son contenu à nobody (remplacez \$WEB par sa valeur) :

```
# chown -R nobody.nobody $WEB
```

0.1.8 4. Le fichier de configuration d'Apache

Le fichier de configuration d'apache est \$CONF/httpd.conf, c'est à dire chez moi /etc/httpd/conf/httpd.conf.

0.1.9 4.1 Nettoyage du fichier de configuration d'Apache

\$CONF/httpd.conf est le fichier de configuration d'Apache, il est assez illisible. Pour le nettoyer, fermez-le (si vous l'aviez ouvert), puis tapez les lignes suivantes dans un terminal (remplacez \$CONF par sa valeur) :

```
# cd $CONF
```

```
# mv httpd.conf httpd.conf.old
```

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

Maintenant éditez votre \$CONF/httpd.conf, n'est-t-il 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) :

```
User nobody
Group nobody
# adresse mail du webmaster (à adapter)
ServerAdmin Dieu@trustonme.net
```

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>
```

0.1.10 4.2 Déplacer la racine de son serveur Apache

Ceci concerne essentiellement ceux qui souhaitent déplacer la racine de leur serveur sur un autre répertoire. Pour ceux qui ignorent toujours la valeur de leur variable `$WEB` chez eux, sachez qu'elle est indiquée par l'option "DocumentRoot" du fichier `$CONF/httpd.conf`. Concrètement, chez moi j'ai :

```
DocumentRoot "/home/httpd/htdocs"
```

Cette variable indique la racine de mon serveur Apache, ceci implique que `$WEB` vaut chez moi `/home/httpd`. Chez vous, vous aurez par exemple :

```
DocumentRoot "/var/www/html"
```

Si vous souhaitez le changer en `/nouveau/site/htdocs`, il suffit de faire trois choses :

- Copiez le contenu de `/var/www`, cela peut se faire en root par :

```
# cp -rfd /var/www/* /nouveau/site/
# cd /nouveau/site
# mv html htdocs
# chown -R nobody.nobody /nouveau/site
```

- Dans les fichiers `$CONF/httpd.conf` et `$CONF/ssl.conf`, à l'aide de votre éditeur de texte favori, remplacez toutes les occurrences de `/var/www` par `/nouveau/site`.
- Assurez-vous que l'option qui suit est ainsi dans votre `$CONF/httpd.conf` :

```
DocumentRoot "/nouveau/site/htdocs"
```

0.1.11 4.3 public_html

Dans `$CONF/httpd.conf`, grâce à la directive :

```
UserDir public_html
```

Chaque utilisateur de votre PC a la possibilité, d'avoir un site perso, accessible à l'adresse `http://localhost/~le_login`³ (une fois qu'apache est lancé). Chaque utilisateur souhaitant en profiter devra, taper les lignes suivantes :

³http://localhost/~le_login/

```
$ 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 l'autoriser à lire les fichiers dans ~/public_html/. Chaque utilisateur devra donc, à chaque fois qu'il y ajoute des fichiers taper :

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

0.1.12 4.4 Les Virtual Hosts

"Virtual Hosts" est la possibilité qu'offre un serveur Apache n'ayant qu'une unique adresse IP, d'héberger plusieurs sites. Il y'a beaucoup à dire dessus, c'est pourquoi je vais partir d'un cas simple qui, je l'espère, sera suffisamment didactique pour être adapté à vos besoins. **Situation** : Je développe deux sites Web, qui ne seront accessibles que dans mon réseau local. Ces sites sont <http://trustonme.portableslk>⁴ et <http://toomuch.portableslk>⁵. Il me suffit donc d'ajouter à la fin du fichier \$CONF/httpd.conf, ceci :

```
NameVirtualHost trustonme.portableslk
<VirtualHost trustonme.portableslk>
Options Indexes FollowSymLinks
ServerAdmin trustonme@localhost
DocumentRoot /home/httpd/sites/trustonme
ServerName trustonme.portableslk
ErrorLog /var/log/httpd/trustonme.portableslk-error_log
TransferLog /var/log/httpd/trustonme.portableslk-access_log
</VirtualHost>
NameVirtualHost toomuch.portableslk
<VirtualHost toomuch.portableslk>
Options Indexes FollowSymLinks
ServerAdmin toomuch@localhost
DocumentRoot /home/httpd/sites/toomuch
ServerName toomuch.portableslk
ErrorLog /var/log/httpd/toomuch.portableslk-error_log
TransferLog /var/log/httpd/toomuch.portableslk-access_log
</VirtualHost>
```

Je suppose dans la suite que la machine sur laquelle est installée Apache a pour adresse IP locale : 192.168.0.4. Pour associer l'adresse IP du serveur Apache aux différents noms de site : il suffit d'éditer les fichiers /etc/hosts de toutes les machines de votre réseau local. Pour Windows 95/98/Me c'est généralement C:\windows\hosts et pour Windows NT/200/XP c'est C:\windows\system32\drivers\etc\hosts. Ajoutez-y ceci :

```
192.168.0.4 trustonme.portableslk
192.168.0.4 toomuch.portableslk
```

⁴<http://trustonme.portableslk>

⁵<http://toomuch.portableslk>

Après redémarrage d'Apache, désormais, je pourrai accéder à mes deux sites de tout mon réseau local, en tapant comme adresse `http://trustonme.portableslk`⁶ et `http://toomuch.portableslk`⁷. Pour associer l'adresse IP du serveur Apache aux différents noms de site, sur Internet ou sur un réseau local d'envergure. Il faut un serveur nom (DNS) et si nécessaire réserver un nom auprès d'un bureau d'enregistrement (registrar). Mais le principe est le même. Pour plus de détails sur les "Virtual Hosts" consultez ceci⁸

0.1.13 4.5 Les Alias

La directive `Alias` du fichier `$CONF/httpd.conf`, permet de stocker des documents accessibles dans d'autres emplacements du système de fichier que la sous-arborescence définie à partir du répertoire `DocumentRoot`. En clair, lorsque quelqu'un entre comme adresse `http://localhost/toto/index.html`⁹ Apache lui enverra par défaut le fichier `$WEB/htdocs/toto/index.htm`. En supposant que `DocumentRoot` ait pour valeur `$WEB/htdocs`. Si vous souhaitez qu'Apache affiche plutôt le fichier `/home/moi/index.html`, il suffit d'ajouter à la fin du fichier `$CONF/httpd.conf` ceci :

```
Alias /toto /home/moi
```

0.1.14 5. Configuration et Activation de SSL

Cette partie n'intéresse que les personnes qui souhaitent activer le support des pages commençant par "https ://". Les autres peuvent passer au 6. Pour accepter les requêtes SSL, Apache a besoin de deux fichiers : une clé pour le serveur (`trustonme.key`) et un certificat signé (`trustonme.crt`). Les noms des fichiers n'ont pas d'importance. Dans les lignes qui suivent remplacez `trustonme.crt` et `trustonme.key` par `votre_site.crt` et `votre_site.key`. La signature de ce certificat est réalisée par un organisme de certification tiers (tel que Verisign ou Thawte). Cependant vous pouvez signer vous même votre certificat, la seule différence sera un avertissement par le navigateur lors de l'accès à une ressource SSL de votre serveur. La sécurité est la même, que le certificat soit signé par vous même ou pas un organisme. Je suppose dans la suite que le fichier `openssl.cnf` est dans `/etc/ssl/`. Si ce n'est pas ainsi chez vous : adaptez !

0.1.15 5.1 Création des clé et certificat

Pour créer votre clé et votre certificat, tapez commandes les suivantes dans un terminal :

```
$ mkdir /tmp/ssl_conf
$ cd /tmp/ssl_conf
$ openssl req -config /etc/ssl/openssl.cnf -new -out trustonme.csr
```

Là il vous demande un passphrase, entrez un mot de passe dont vous vous souviendrez. Finissez, en entrant des informations régionales. Ensuite, tapez :

```
$ openssl rsa -in privkey.pem -out trustonme.key
$ openssl x509 -in trustonme.csr -out trustonme.crt -req -signkey trustonme.key
-days 3650
```

⁶<http://trustonme.portableslk>

⁷<http://toomuch.portableslk>

⁸<http://httpd.apache.org/docs-2.0/vhosts/>

⁹<http://localhost/toto/index.html>

```
$ openssl x509 -in trustonme.crt -out trustonme.der.crt -outform DER
```

Notez que mon certificat est valable 3650 jours (presque 10 ans)

0.1.16 5.2 Configuration d'Apache pour le SSL

En tant que root, copiez la clé et le certificat dans les dossiers ssl d'Apache (vous devrez peut-être les créer) :

```
# cd /tmp/ssl_conf
# cp trustonme.crt $CONF/ssl.crt/
# cp trustonme.key $CONF/ssl.key/
```

Nettoyez le fichier ssl.conf :

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

Editez le fichier \$CONF/ssl.conf et modifiez les lignes qui suivent ainsi (n'oubliez pas de remplacer \$CONF par sa valeur) :

```
SSLCertificateFile $CONF/ssl.crt/trustonme.crt
SSLCertificateKeyFile $CONF/ssl.key/trustonme.key
```

0.1.17 6. Lancement d'Apache

Pour lancer apache sans support ssl tapez maintenant (remplacez \$APACHECTL par sa valeur) :

```
# $APACHECTL start
```

Pour lancer apache avec support ssl tapez (remplacez \$APACHECTL par sa valeur) :

```
# $APACHECTL startssl
```

\$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 (remplacez \$WEB par sa valeur) :

```
# cd $WEB/htdocs/
# mv manual/index.html.fr index.html
```

Puis dans un navigateur entrez l'adresse <http://localhost/>¹⁰

0.1.18 7. Lancement automatique d'Apache

Ce qui suit ne concerne que ceux qui ont installé Apache à partir des sources. En effet, pour ceux qui ont utilisé les paquetages de leur distribution tout est fait pour qu'Apache se lance au démarrage de la machine. Pour qu'Apache soit lancé à chaque démarrage, rajoutez les lignes suivantes dans votre /etc/rc.d/rc.local (remplacez \$APACHECTL par sa valeur) :

¹⁰<http://localhost/>

```
if [ -x $APACHECTL ]; then
echo "Demarrage d'Apache ..."
$APACHECTL start
fi
```

Adaptez, si vous souhaitez activer le support SSL. Vous trouverez de la documentation sur apache2 ICI¹¹.

¹¹<http://httpd.apache.org/docs-2.0/fr/>