Linux Apache SSL PHP/FI frontpage mini-HOWTO

Marcus Faure, *marcus@faure.de* v1.1, July 1998

Ce document traite de la mise en place d'un **serveur WEB multifonctions** Il autorisera la création dynamique des pages avec la langage **PHP**, sécurisera l'échange des données grâce à **SSL** et fiabilisera l'exécution des scripts **CGI** et des extensions **Frontpage** de M$.

Contents

1 Introduction

Avant tout : [NdT : l'auteur explique qu'il n'est pas un Mozart de la langue anglaise. Ca tombe bien, moi non plus !]

1.1 Description des différentes composantes

Le serveur web dont vous disposerez gràce à ce Howto comprend plusieurs parties : les sources Apache d'origine avec quelques (ah, ah) patches et divers exécutables. Je vous conseille d'utiliser les mêmes versions de logiciels que moi. Elles ne devraient pas poser de problèmes et le démon résultant sera relativement stable.

Les plus téméraires essayeront les dernières versions avec de toutes nouvelles fonctionnalités mais ce n'est pas ma faute si ca ne marche pas. Les configurations utilisables gagneront à être signalées afin que je les inclut dans les nouvelles versions de ce document. Ma plate forme d'expérimentation repose sur un noyau Linux 2.0.35, d'où une certaine spécificité de ce Howto qui devrait rester applicable à d'autres Unix.

Il n'est pas nécessaire de tout compiler. J'ai essayé de structurer ce HOWTO de telle sorte que vous puissiez ignorer les parties qui ne vous concernent pas.

Ce document n'est pas un manuel utilisateur pour Apache, SSL, PHP/FI ou FrontPage. Son objectif premier est de diminuer la consommation d'aspirine des fournisseurs de service WEB lors de la mise en place de leur premier serveur et d'apporter une contribution à la communauté Linux.

PHP est un langage de script qui autorise les pages HTML dynamiques. Il ressemble au SSI d'Apache mais en bien moins compliqué. Des modules existent pour les bases de données courantes. PHP requiert les librairies GD.

SSL est une mise en oeuvre du Secure Socket Layer de Netscape qui sécurise les connections sur des réseaux douteux (typiquement pour la transmission des numéros de cartes de crédit à partir de formulaires web).

Frontpage est un outil ouiziouigue de création web qui propose des extensions, propres à un type de serveurs, appellées webbots. Certains trouvent Frontpage agréable en ce qu'il permet la mise en place de sites de discussion et de questionnaires en ignorant tout du HTML et du CGI. Il épargne même au concepteur le téléchargement du site via ftp grâce à son propre agent. Si vous souhaitez la compatibilité Frontpage sans le serveur à fenètres, les extensions du serveur Apache sont faites pour vous.

1.2 Configurations opérationelles

Bien que ce document ait été téléchargé une centaine de fois depuis sa publication, je n'ai guère eu de retour. En particulier, personne ne m'a rapporté la viabilité de nouvelles combinaisons. Voici les miennes :

- Linux 2.0.31, Apache 1.2.4, PHP 2.0.0, SSL 0.8.0, fp 98 3.0.3 (*)
- Linux 2.0.33, Apache 1.2.5, PHP 2.0.1, SSL 0.8.0, fp 98 3.0.3 (*)
- Linux 2.0.35, Apache 1.2.6, PHP 3, SSL 0.8.0, fp 98 3.0.4

(*) la version 3.0.3 est 3.6 (déconseillé)

1.3 Historique

v0.0/Apr 98: Version allégée

v1.0/Jun 98: Apache 1.2.6, section FrontPage mise à jour, corrections mineures

v1.1/Jul 98: Restructuration et passage au SGML

La dernière mouture de ce document est disponible via :

`http://www.faure.de`

2 Installation des différents éléments

2.1 Préparatifs

Il vous faudra :

- Apache 1.2.6 `http://www.apache.org/dist/apache_1_2_6.tar.gz`
- Extensions PHP/FI
  `http://php.iquest.net/files/download.phtml?/files/php-2.01.tar.gz`
- Librairie GD `http://siva.cshl.org/gd/gd.html`
- SSL 0.8.0
  `ftp://ftp.ox.ac.uk/pub/crypto/SSL/SSLeay-0.8.0.tar.gz`
- Patch SSL pour Apache 1.2.6
  `ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.6+ssl_1.17.tar.gz`
- Extensions pour serveur Frontpage 98 script d'installation
  `http://www.rtr.com/fpsupport/download.htm`

Récupérez les sources souhaitées. Détarez Apache, PHP, GD et SSL en `/usr/src`. Détarez le patch SSL vers `/usr/src/apache_1.2.6`.

2.2 Ajout de PHP

Faites un `cd` vers `/usr/src/gd1.2` suivi de make pour construire la librarie GD `libgd.a` qui devra être copiée en `/usr/lib`. Allez ensuite en `php-2.0.1` et lancez `./install`.

Voici les questions-clé :

```
Would you like to compile PHP/FI as an Apache module? [yN] y
Are you compiling for an Apache 1.1 or later server? [Yn] y
Are you using Apache-Stronghold? [yN] y
Does your Apache server support ELF dynamic loading? [yN] y
Apache include directory (which has httpd.h)? [/usr/local/include/apache] /usr/src/apache_1.2.6/src
Would you like to build an ELF shared library? [yN] y
Additional directories to search for .h files []: /usr/src/gd1.2
Would you like the bundled regex library? [yN] n
```

De même que les extensions FrontPage, phtml pose un problème au niveau de la sécurité vu qu'il tourne avec l'UID du serveur Web. Enclenchez le mode sûr dans src/php.h et limitez prudemment le chemin de recherche. Vous souhaiterez peut- être changer d'autres valeurs dans le php.h. Si l'aspect sécurité est crucial pour vous, compilez php en tant que cgi. L'ensemble sera cependant moins performant que la version module.

Tapez `make` pour compiler les sources. Une fois la compilation achevée, copiez `mod_php.*` et `libphp.a` en `/usr/src/apache_1.2.6/src`. Ajoutez une ligne

```
Module php_module mod_php.o
```

à la fin de `/usr/src/apache_1.2.6/src/Configuration` et

```
-lphp -lm -lgdbm -lgd
```

à `EXTRA_LIBS` (même fichier), ainsi que

```
application/x-httpd-php phtml
```

au `mime.types` d'Apache et

```
AddType  application/x-httpd-php .phtml
```

au `srm.conf` d'Apache.

Peut-être ajouterez vous aussi `index.phtml` à `DirectoryIndex` dans ce même fichier de façon à ce qu'un fichier index.phtml soit automatiquement chargé lors d'une requête sur son répertoire.

2.3 Ajout de SSL

`cd /usr/src/SSL-0.8.0; ./Configure linux-elf; make; make rehash` Ceci compilera les librairies requises par Apache. `make test` vous permettra de vérifier le bon déroulement de la compilation. Il faut à présent patcher Apache. Il est primordial d'appliquer ce patch avant le patch Frontpage sinon ce dernier sera inefficace. Faites un `cd` vers `/usr/src/apache_1.2.6/src` et lancez `patch < /usr/src/apache_1.2.6/SSLpatch`. Positionnez `SSL_BASE=/usr/src/SSLeay-0.8.0` in `Configuration`. Assurez vous de ce que `Module proxy_module` est désactivé sinon Apache ne compilera pas. Si vous avez besoin d'un proxy, Squid vous tend les bras :

`squid.nlanr.net`

`make certificate` génerera `SSLconf/conf/httpsd.pem`.

2.4 Ajout de FrontPage

Renommez le fichier `fp30.linux.tar.Z` en `fp30.linux.tar.gz` ou bien le script d'install ne le verra pas. Lancez `./fp_install` ainf de copier les fichiers d'extension en `/usr/local/frontpage`. zcat devrait s'exécuter par /usr/bin/zcat.

On applique à présent le patch FrontPage : allez en `/usr/src/apache_1.2.6/src` et tapez `patch < /usr/src/frontpage/version3.0/apache-fp/fp-patch-apache_1.2.5` Ceci créera les fichiers `mod_frontpage.*` et en modifiera d'autres (`Configuration`, ...). Le patch 1.2.5 fonctionne aussi bien avec Apache 1.2.5 qu'avec 1.2.6. Passez pour l'instant sur ce qui se rapporte à l'installation des serveurs ouaibe.

3 Assemblons les morceaux du puzzle !

3.1 Modules Apache à essayer

J'utilise les modules suivants en plus de SSL, PHP et Frontpage :

```
Module env_module          mod_env.o
Module config_log_module   mod_log_config.o
Module mime_module         mod_mime.o
Module negotiation_module  mod_negotiation.o
Module dir_module          mod_dir.o
Module cgi_module          mod_cgi.o
Module asis_module         mod_asis.o
Module imap_module         mod_imap.o
Module action_module       mod_actions.o
Module alias_module        mod_alias.o
Module rewrite_module      mod_rewrite.o
```

```
Module access_module       mod_access.o
Module auth_module         mod_auth.o
Module anon_auth_module    mod_auth_anon.o
Module digest_module       mod_digest.o
Module expires_module      mod_expires.o
Module headers_module      mod_headers.o
Module browser_module      mod_browser.o
```

3.2 Sécuriser CGI

Si vous êtes un FAI (c'est surement le cas si vous lisez ces lignes), vous voudrez améliorer la sécurité. L'utilitaire suexec vous le permettra dans la mesure où il exécutera les CGI avec l'UID du propriétaire du Web plutôt qu'avec celle du serveur. Allez en `/usr/src/apache_1.2.6/support` et lancez `make suexec`. Faites un `chmod 4711 suxec` et copiez le à l'emplacement précisé dans `../src/httpd.h`, soit `/usr/local/etc/httpd/sbin/suexec` par défaut. Si le chemin vous semble un peu obscur - ce fût le cas pour moi - éditez `httpd.h` et positionnez le à quelque chose de plus raisonnable.

3.3 Compilation et installation du démon

Allez en `/usr/src/apache_1.2.6/src` et éditez `Configuration` afin de choisir les modules que vous voulez inclure dans votre démon Apache. Ceci effectué, lancez `./Configure` et `make`. Il s'agit là de la dernière étape de compilation, la plus compliquée. Croisez les doigts. Si tout se déroule normalement, copiez `httpsd` en `/usr/sbin`. Ne négligez pas la taille du démon. Créez le répertoire `/var/httpd` avec les sous-répertoires `cgi-bin`, `conf`, `htdocs`, `icons`, `virt1`, `virt2` et `logs`. Editez `access.conf-dist`, `mime.types` et `srm.conf-dist` (répertoire `/usr/src/apache_1.2.6/conf`) selon vos besoins et copiez les en `var/httpd/conf/access.conf`, `srm.conf` et `mime.types`. Enfin, copiez le fichier `httpsd.pem` créé par `make certificate` en `/var/httpd/conf`. Utilisez le `httpd.conf` suivant :

```
ServerType standalone
Port 80
Listen 80
Listen 443
User wwwrun
Group wwwrun
ServerAdmin webmaster@yourhost.com
ServerRoot /var/httpd
ErrorLog logs/error_log
TransferLog logs/access_log
PidFile logs/httpd.pid
ServerName www.yourhost.com
MinSpareServers 3
MaxSpareServers 20
StartServers 3

SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/ssl.log

<VirtualHost www.virt1.com>
```

```
SSLDisable
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi-bin/ /var/httpd/virt1/cgi-bin/
ServerName www.virt1.com
ErrorLog logs/virt1-error.log
TransferLog logs/virt1-access.log
User virt1admin
Group users
</VirtualHost>

<VirtualHost www.virt1.com:443>
ServerAdmin webmaster@virt1.com
DocumentRoot /var/httpd/virt1
ScriptAlias /cgi-bin/ /var/httpd/virt1/cgi-bin/
ServerName www.virt1.com
ErrorLog logs/virt1-ssl-error.log
TransferLog logs/virt1-ssl-access.log
User virt1admin
Group users
SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/virt1-ssl.log
SSLVerifyClient 0
SSLFakeBasicAuth
</VirtualHost>

<VirtualHost www.virt2.com>
SSLDisable
ServerAdmin webmaster@virt2.com
DocumentRoot /var/httpd/virt2
ScriptAlias /cgi-bin/ /var/httpd/virt2/cgi-bin/
ServerName www.virt2.com
ErrorLog logs/virt2-error.log
TransferLog logs/virt2-access.log
</VirtualHost>
```

Les directives dépendent des modules compilés. Leur liste s'obtient avec `httpsd -h`.

3.4 Ajout des fonctionnalités Frontpage au serveur Web

Allez en `/usr/local/frontpage/version3.0/bin` et lancez `./fpsrvadm`. Choisissez `install` et `apache-fp`. Répondez comme suit aux questions :

```
Enter server config filename: /var/httpd/conf/httpd.conf
Enter host name for multi-hosting []: www.virt2.com
Starting install, port: www.virt2.com:80, web: ""
Enter user's name []: virt2admin
Enter user's password:
```

```
Confirm password:
Creating root web
Recalculate links for root web
Install completed.
```

Le nom d'utilisateur doit correspondre au login Unix du ppropriétaire du web. Le mot de passe peut différer. Ajoutez à la main `sendmailcommand:/usr/sbin/sendmail %r` au fichier `/usr/local/frontpage/www.virt2.com:80.conf` ou vos utilisateurs ne pourront envoyer de courriers générés via le web. Faites un `kill -HUP` sur votre `httpsd` de façon à ce que fp réexamine sa configuration. Votre client FrontPage peut à présent accéder à `www.virt2.com`

Dans certains cas, `fpsrvadm` se plaint de l'absence d'un web à la racine. Cela ne sert guère, mais vous devrez surement le faire pour que `fpsrvadm` se taise.

3.5 Démarrage du démon

Lancez Apache via `httpsd -f /var/httpd/conf/httpd.conf`. On accède à présent à `www.virt1.com` aussi bien via http que via https. C'est chouette la vie. Naturellement, vous acheterez un véritable certificat afin d'offrir un service Web SSL qui ne fasse pas rigoler les clients.

Copiez un des fichiers du répertoire d'exemples de php en `virt1` pour essayer phtml.

3.6 Considérations diverses

N'utilisez pas les extensions FrontPage 97. Elles ne fonctionnent pas sous Linux. A l'installation, les bibliothèques C++ semblent fonctionner mais vos fichiers de logs se rempliront de `premature end of script headers` tandis que votre boite au lettre débordera de réclamations. Evitez les extensions FrontPage 98 avec des versions antérieures à la 3.0.2.1330. Les numéros de version manquent singulièrement d'homogénéité. Faites un telnet sur le port 80, tapez "get / http/1.0" suivi de deux retours à la ligne et vous obtiendrez un numéro de version de 3.0.4 pour FrontPage.

Des informations plus précises sont disponibles en exécutant : `/usr/local/frontpage/currentversion/exes/_vti_bin/shtml.exe -version`. Les versions plus anciennes sont buggées en ce qu'elles exigent du httpd.conf qu'il puisse être écrit sous le gid du serveur web. Voila de quoi faire hurler ceux qui tiennent à la sécurité. Les versions postérieures à la 3.0.2.1330 fonctionnent mieux.

3.7 Bugs connus

L'utilisation de `Recalculate Links` par le client Frontpage lance au niveau du serveur un process qui consomme 99% du cpu et 10 Mo de mémoire. Pour des webs de taille moyenne avec des machines rapides, le client reçoit parfois un timeout et ce alors même que le calcul parvient normalement à son terme. Demandez aux adeptes de FrontPage de rester patients et de ne pas activer `Recalculate Links` à répétition. Pensez à mettre au moins 64 Mo.

A l'instant où j'écris ces lignes, SSL et Frontpage fonctionnent mais pas de façon simultanée. Vous ne pouvez donc ni mettre à jour votre web via SSL, ni utiliser des webbots avec https. Vous pouvez actualiser votre web via le port 80 et y accéder en chiffré via le port 443 mais les compteurs par exemple seront faux. Je considère ceci comme un bug. Il devrait disparaitre avec SSL 0.9.0.

3.8 Le mot de la fin

Pour ceux qui trouvent que le titre de ce HOWTO est presque aussi long que son contenu : avez vous jamais écouté Meat Loaf ?

Bien, amis lecteurs, c'est tout pour aujourd'hui. N'hésitez pas à me transmettre vos expériences, votre reconnaissance éternelle, des fleurs, de l'argent, des voitures, des gisements de pétrole, ou des ratons laveurs.