# Apache



---

# Sections

# Installation

## Sous Windows

### Tout-en-un

Des logiciels tout-en-un (serveur Web, base de donnée MySQL, et PHP) permettent de s'affranchir d'une installation fastidieuse et rédhibitoire pour le débutant :

1. EasyPHP téléchargement (http://www.easyphp.org) : n'a pas vocation à être installé pour de la production, mais pour le développement. Il stocke les bases de données dans C:\Program Files (x86)\EasyPHP\binaries\mysql\data.
2. WAMP téléchargement (http://www.wampserver.com) : est du même type qu'EasyPHP : ce logiciel installe facilement un serveur Web Apache, une base de données MySQL et PHP 4 et 5. Il a l'avantage de permettre de passer facilement de PHP 4 à PHP 5, sans avoir à refaire une installation ou une compilation. Tout comme EasyPHP, c'est un environnement de développement, et non un environnement de production. Attention : la résolution des noms d'hôtes se réalise séparément. Les installations WAMP servent à tester en local sur votre PC. Dans la plupart des cas, il suffit d'utiliser le fichier *Hosts* local, comme on le ferait sur une machine Linux, afin de lier des noms aux adresses IP. Dans Windows XP, Vista et 7, ce fichier se trouve dans le répertoire *systemroot\System32\Drivers\Etc*. Il peut se faire que le service ait déjà été configuré. Lorsque vous vous en doutez, contactez votre administrateur réseau. Remarque : vous trouverez une liste des possibilités de résolution de noms avec MS Windows sur Microsoft.com (http://www.microsoft.com/technet/prodtechnol/winxppro/reskit/c24621675.mspx).
3. XAMPP téléchargement (http://www.apachefriends.org/fr/xampp.html) : est du même type qu'EasyPHP ou WAMP, le deuxième P étant pour Perl. Son usage est recommandé avec PHPEclipse (http://www.phpeclipse.de/tiki-view_articles.php), et il fournit aussi un serveur Apache Tomcat par défaut.
4. The Uniform Server téléchargement (http://www.uniformserver.com) : en anglais seulement avec Apache2, Perl5, PHP5, MySQL5, phpMyAdmin.

**Attention !**

Sur Windows 10 pro, le serveur IIS est installé par défaut, et oblige Apache à changer de port (888 au lieu de 80) lors de l'installation. Pour résoudre cela il suffit de décocher *Internet Information Services* dans *Programmes et fonctionnalités*, *Activer ou désactiver des fonctionnalités Windows.*

De même, le port MySQL est susceptible de passer de 3306 à 3388.

**Attention !**

Sur Windows 10, *EasyPHP development server* (alias *Devserver*, la version rouge) ne fonctionne pas (*il manque MSVCR110.dll*), mais *EasyPHP hosting server* (alias *Webserver*, la bleue) tourne normalement. Or, elle se lance automatiquement à chaque démarrage, ce qui le ralentit significativement. Pour éviter cela, exécuter *services.msc*, puis passer les trois services ci-dessous en démarrage manuel. Ensuite pour les lancer à souhait (en tant qu'administrateur), créer un script MySQL.cmd contenant les lignes suivantes :

```
net start ews-dbserver
net start ews-httpserver
net start ews-dashboard
pause
net stop ews-dashboard
```

```
net stop ews-httpserver
net stop ews-dbserver
```

### Message d'erreur relatif à SSL

Logo EasyPHP.

Pour l'instant, WAMP ne supporte pas encore le *Secure Socket Layer* (SSL). L'installation se finit par un message qui vous informe de ce fait. Afin de pouvoir travailler sans problèmes, éditez le fichier *c:\windows\php.ini*. Cherchez dans ce fichier la ligne qui commence avec *extension=php_openssl.dll*. Commentez cette ligne en la faisant précéder d'un point-virgule :

```
;extensions=php_openssl.dll
```

Si tout se passe bien, vous pouvez ouvrir la page de test dans votre navigateur.

## Installation manuelle

- Apache est disponible sur le site Web de Apache Software Foundation apache.org (http://www.apache.org).
- PHP est téléchargeable sur le site officiel de php (http://www.php.net). Choisissez le fichier au format ZIP.
- Enfin, vous trouverez MySQL sur mysql.com (http://www.mysql.com).

### Installer Apache

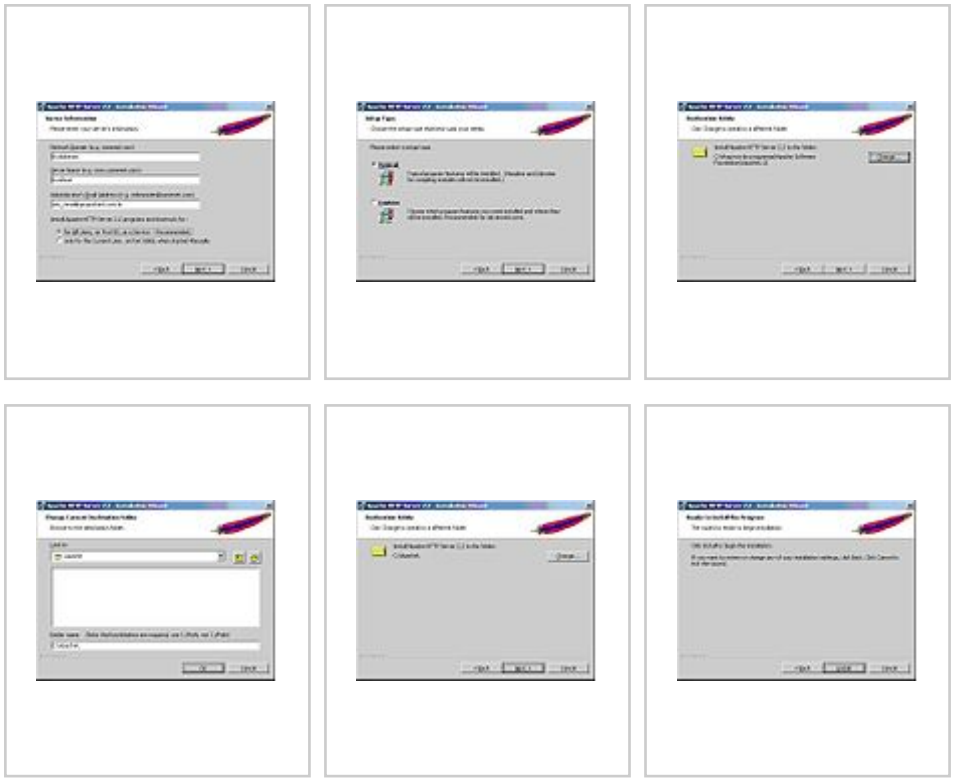
Pour installer Apache, double-cliquez sur le fichier exécutable, et suivez les instructions d'installation automatique.

Si vous installez Apache sur un ordinateur de développement, renseignez le champ "nom de domaine" avec la valeur `localhost`.

Si vous installez un serveur de production et que vous disposez d'un nom de domaine, vous devriez disposer des informations nécessaires concernant votre nom de domaine, fournies par le *registrar*.

Une fois l'installation terminée, il faut encore indiquer à Apache qu'il doit fonctionner conjointement avec PHP, car il ne sait pas les traiter par défaut. Pour cela, il faut modifier les informations de configuration d'Apache, contenues dans le fichier `httpd.conf`, qui se trouve dans le dossier d'installation d'Apache, dans le sous-dossier `conf`.





### Installer PHP

Une fois l'archive téléchargée, décompressez-la à la racine de votre disque dur et renommez le dossier en `'PHP'`. Dans le dossier `PHP`, vous trouverez deux fichiers: `php.ini-dist` et `php.ini-recommended`. Copiez `php.ini-recommended` dans votre dossier `C:\Windows` ou `C:\winnt` (le nom du dossier dépend de la version de votre système.
renommez-le en `php.ini`.

Ce fichier est le fichier de configuration qui contrôle les options dont vous disposerez. Par exemple :

| PHP.ini | PHP | Rôle |
|---|---|---|
| `error_reporting E_ALL` | `error_reporting(E_ALL);` | Affiche tous les avertissements et erreurs directement sur le site. C'est utile pour la préproduction car cela évite de rechercher d'éventuels messages dans les logs, mais peut perturber la mise en page pour des avertissements bénins. |
| `error_reporting 0` | `error_reporting(0);` | N'affiche aucun message sur le site relatif à son exécution |
| `max_execution_time = 300` | | Définit le "timeout", c'est-à-dire le temps maximum en secondes autorisé pour exécuter un script PHP. |
| `post_max_size = 80M` | | Définit la taille maximum d'un fichier que l'on peut envoyer au serveur en HTTP. |

#### MySQL

Télécharger et installer le .msi sur http://dev.mysql.com/downloads/gui-tools/5.0.html.

Pour arrêter, démarrer, démarrer automatiquement le serveur MySQL vous devez aller dans la gestion des services (Démarrer/Exécuter/services.msc).

# Sous Linux

## LAMP

Logiciel tout-en-un pour Linux (Apache + MySQL + PHP), comme WAMP pour Windows.

commande nécessitant les privilèges root

```
# apt-get install tasksel
# tasksel install lamp-server
```

## Installation manuelle

#### Apache sur Debian / Ubuntu

commande nécessitant les privilèges root

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

Le service peut ne pas être lancé par défaut, mais même s'il l'est on peut quand-même essayer de l'activer avec :

commande nécessitant les privilèges root

```
# /etc/init.d/apache2 start
```

On peut ensuite tester le serveur, pour voir si une page s'affiche ou s'il refuse la connexion :

commande

```
$ lynx http://localhost/
```

Cette adresse est le rebouclage, elle peut aussi être rentrée directement dans tout navigateur web.

Si Apache était déjà installé vérifier le fichier pour indiquer le démarrage automatique d'Apache 2 **/etc/default /apache2** :

```
# vi /etc/default/apache2
 ...
NO_START=0
```

### PHP

PHP peut-être installé avec toutes les déclinaisons de la distribution Debian (stable, testing, unstable). Il suffit pour cela d'insérer vos lignes préférées dans le fichier */etc/apt/sources.list* :

```
deb http://ftp.fr.debian.org/debian/ stable main non-free contrib
deb-src http://ftp.fr.debian.org/debian/ stable main non-free contrib
```

Ce qui suit suppose que le serveur Web a bien été installé : exécuter les commandes suivantes :

```
sudo apt-get update && apt-get install php7.0 && apt-get install libapache2-mod-php7.0
```

Une fois ces commandes exécutées, redémarrer le serveur Web. Dans le cas d'Apache cela s'effectue avec la commande suivante :

```
/etc/init.d/apache2 restart
```

Si tout s'est bien passé, vous disposez maintenant d'un serveur Web qui a la capacité d'exécuter des scripts PHP dans votre navigateur.

Testons :

commande

```
$ lynx http://localhost/test.php
```

Pour débugger :

commande

```
$ tail /var/log/apache2/error.log
```

### Mise à jour

Pour la v7.2 :

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php7.2 php7.2-common php7.2-cli php7.2-fpm
sudo a2enmod php7.2
sudo a2dismod php7.0
```

**Attention !**

Une fois les serveurs Web installés, ils se lancent automatiquement à chaque démarrage de la machine, ce qui est souhaitable pour un serveur, mais pas toujours pour un PC. Pour éviter cela, il suffit d'y désactiver les daemons :

```
sudo update-rc.d apache2 disable
sudo update-rc.d mysql disable
```

### Apache sur Gentoo

Premièrement il faut installer Apache si ce n'est pas déjà fait :

```
emerge apache
```

Ensuite, il faut installer PHP :

```
emerge dev-lang/php
```

Puis il faut qu'apache utilise PHP dans sa configuration.

**Code:** Configuration de apache

```
# nano -w /etc/conf.d/apache2
APACHE2_OPTS="-D PHP5"
```

### MySQL seul

MySQL est disponible sur http://dev.mysql.com/downloads/gui-tools/5.0.html au format :

1. .msi (Windows)
2. .dmg (Mac)
3. .rpm (Linux)
4. .tar

En l'absence de gestionnaire de paquets, utiliser le .tar ainsi :

```
shell> groupadd mysql
shell> useradd -r -g mysql mysql
shell> cd /usr/local
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.gz
shell> ln -s full-path-to-mysql-VERSION-OS mysql
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
shell> bin/mysqld_safe --user=mysql &
```

### APT

```
$ sudo apt-get install mysql-server mysql_secure_installation
```

Puis, modifier PHP pour qu'il supporte MySQL :

```
$ sudo apt-get install php4-mysql
```

### Variante

La dénomination des paquets mentionnés peut varier légèrement selon la version. Dans un terminal, entrez :

```
$ sudo apt-get install mysql-server
```

et confirmez.

*(Remarque : il semblerait qu'en installant le paquet "mysql-server-5.0", au lieu du paquet mentionné plus haut, certaines personnes rencontrent des problèmes. Il est donc préférable d'installer ce paquet, ou d'installer la dernière*

*version 4 stable avec : $ sudo apt-get install mysql-server-4.1. Consultez le forum pour plus d'informations : [1] (http://forum.ubuntu-fr.org/viewtopic.php?id=15352))*

Lancez ensuite la commande :

```
cd && sudo mysql_secure_installation
```

Appuyez sur Entrée lorsqu'il vous demande le mot de passe root MySQL : pour le moment il n'y en a pas.

***NB :** MySQL a ses propres utilisateurs, avec leurs propres privilèges. Le root MySQL n'est donc pas le root système. Il est conseillé de ne pas mettre les mêmes mots de passes pour les utilisateurs MySQL et les utilisateur du système.*

Le script vous demande alors si vous voulez mettre un mot de passe pour l'utilisateur root. Répondez Y, et entrez (2 fois le nouveau mot de passe du root MySQL). Il vous pose ensuite une série de questions. Si vous ne savez pas quoi répondre, acceptez les choix par défaut en appuyant simplement sur Enter.

Votre serveur MySQL est prêt. Par défaut il se lance à chaque démarrage du système, si vous ne le souhaitez pas, il vous suffit de lancer :

```
$ sudo dpkg-reconfigure mysql-server
```

et de répondre "Non" à la question du démarrage systématique de MySQL.

### Sur Gentoo

```
emerge mysql
```

## Installer PhpMyAdmin

Depuis un tout-en-un, il suffit de créer un chemin accessible depuis le serveur Web :

```
sudo ln -s /usr/share/phpmyadmin /var/www/phpmyadmin
```

Sinon :

```
sudo apt-get install phpmyadmin php5
```

### Installer Apache et PHP avec PhpMyAdmin

Grâce aux dépendances des paquets, cette opération peut se faire en une seule fois : *Remarque : Vérifiez que la case "Traiter les paquets recommandés comme des dépendances" soit cochée dans Synaptic, configuration, préférences.*

```
$ sudo apt-get install phpmyadmin
```

Cela installera automatiquement apache2 + php + modules d'apache pour PHP et MySQL + PhpMyAdmin. Pour accéder à PhpMyAdmin, il faut se rendre à la page http://localhost/PhpMyAdmin.

*Note : En cas de problème d'authentification (erreur 2002 notamment) installer le paquet mysql-server peut résoudre ce dernier.*

Après l'installation, il vaut mieux modifier les droits d'accès de root, et ajouter un mot de passe pour un peu plus de sécurité. Pour cela, il faut se rendre à la page *privilèges* de PhpMyAdmin.

Remarque pour Ubuntu 5.04 (Hoary Hedgehog) : Afin que cette commande fonctionne il est nécessaire d'avoir effectué les modifications suivantes : dans /etc/apt/ éditer le fichier sources.list supprimer les # des lignes suivantes :

```
# deb http://fr.archive.ubuntu.com/ubuntu hoary universe
```

(cette ligne est dans certain cas '# deb http://archive.ubuntu.com/ubuntu/ hoary universe main restricted multiverse')

```
# deb-src http://fr.archive.ubuntu.com/ubuntu hoary universe
```

*Pour la version d'Ubuntu 5.10 (Breezy), vous pouvez effectuer ces changements avec le gestionnaire de paquets synaptic (apt) : Système ---> Administration ---> Gestionnaire de paquets Synaptic*

```
           Catégories ---> Dépôts ----> Ajouter et ensuite, sélectionner : maintenu
par la communauté universe...
```

Lancer le chargement des nouvelles sources :

```
$ sudo apt-get update
```

Puis lancer l'installation de PhpMyAdmin comme décrit ci-dessus.

## Extensions

Pour activer des modules complémentaires :

```
a2enmod Nom_du_module # passe dans /etc/apache2/mods-enabled/
```

Ex :

```
a2enmod rewrite
```

Pour les désactiver :

```
a2dismod Nom_du_module # passe dans /etc/apache2/mods-available/
```

Pour activer des sites :

```
a2ensite Nom_du_site # passe dans /etc/apache2/sites-enabled/
```

Pour les désactiver :

```
a2dissite Nom_du_site # passe dans /etc/apache2/sites-available/
```

Les extensions PHP nécessitent une autre commande. Ex :

```
phpenmod mbstring
```

# Problème d'encodage d'Apache2

Si vous rencontrez un problème d'encodage des caractères de vos pages, par exemple les caractères accentués apparaissant sous la forme "" (<?>), c'est probablement parce qu'Apache2 déclare dans les en-têtes HTTP qui accompagnent les pages visionnées un encodage par défaut en Unicode (UTF-8) :

```
Content-Type: text/html; charset=UTF-8
```

Tandis que les pages visionnées utilisent un autre encodage des caractères, comme par exemple Latin1 (ISO-8859-1). Même si vos documents indiquent le jeu de caractères utilisé, le paramètre donné par le serveur dans les en-têtes HTTP est prioritaire !

Pour corriger ce problème, il faudra éditer /etc/apache2/apache2.conf :

```
$ sudo gedit /etc/apache2/apache2.conf
```

### Encodage par défaut en Latin1 (ISO-8859-1)

Cherchez la ligne suivante :

```
#AddDefaultCharset     ISO-8859-1
```

Décommentez-la en enlevant le # :

```
AddDefaultCharset      ISO-8859-1
```

Pour ceux qui ont la locale iso-8859-15 (sinon vous pouvez faire "sudo dpkg-reconfigure locales" pour l'ajouter) et qui désirent l'utiliser par défaut, ajoutez un 5 en fin de ligne :

```
AddDefaultCharset      ISO-8859-15
```

ainsi que la ligne suivante dans le paragraphe en-dessous :

```
AddCharset ISO-8859-15 .iso8859-15  .latin15 .fr
```

Il ne vous reste plus qu'à mettre "fr" en première position dans la ligne LanguagePriority (juste au-dessus), et à demander à apache de relire sa configuration :

```
$ sudo /etc/init.d/apache2 reload
```

### Aucun encodage par défaut

Il est également possible de s'affranchir de tout encodage par défaut, de la manière suivante :

Cherchez la directive *AddDefaultCharset* :

```
AddDefaultCharset      ISO-8859-1
```

Remplacez l'attribut par la valeur *Off* :

```
AddDefaultCharset      Off
```

Là encore, on demandera à Apache de relire sa configuration :

```
$ sudo /etc/init.d/apache2 reload
```

Maintenant, les en-têtes HTTP ne contiendront plus d'indication d'encodage des caractères. Attention : il faudra alors que chaque page indique l'encodage utilisé, car s'en remettre à la détection automatique par les navigateurs peut s'avérer assez aléatoire !

# Test des bases de données

## MySQL

Pour tester si les comptes utilisateurs peuvent se connecter au serveur de bases de données il suffit de lancer un fichier de testBDD.php comme suit[1] :

```php
<?php
	$connexion = mysql_connect("localhost", "mysql_user", "mysql_password");
	if ($connexion) {
		print ("Connexion OK");
	}
	else {
		print ("Connexion KO");
	}
?>
```

En cas d'erreur :

- 1045 *échec de la connexion* : le compte "mysql_user" n'a pas le droit de se connecter au serveur, il faut le recréer dans PHPmyadmin. Le faire en cochant "créer une base à son nom", même si elle existe déjà (cela ne l'effacera pas).
- 1044 : le compte "mysql_user" n'a pas le droit de se connecter à la base, il faut le modifier dans PHPmyadmin.
- 1449 *The user specified as a definer does not exist* :

```
grant all privileges on NomBase to `NomUtilisateur`@`localhost`
```

## MS-SQL

On distingue plusieurs pilotes PHP pour MS-SQL Server :

- mssql (désuet en PHP7).
- sqlsrv
- PDO
  - pdo_sqlsrv
  - pdo_dblib (Sybase)
  - pdo_odbc (ODBC).

## Windows

Pour se connecter au serveur MS-SQL à partir d'un tout-en-un comme EasyPHP, il suffit de télécharger les pilotes .dll[2] correspondant à sa version de PHP, puis d'indiquer leurs chemins dans le PHP.ini :

- Sous PHP 4, copier le fichier `php_mssql.dll` dans les extensions.
- Pour PHP 5.4 :
  1. Télécharger les .dll SQL30
  2. Les copier dans `C:\PROGRA~2\EasyPHP\binaries\php\php_runningversion\ext`
  3. Les ajouter dans `C:\PROGRA~2\EasyPHP\binaries\php\php_runningversion\php.ini` via les lignes suivantes[3] :
     ```
     extension=php_sqlsrv_54_ts.dll
     extension=php_pdo_sqlsrv_54_ts.dll
     ```
- Dans PHP 5.5 on obtient toujours `Fatal error: Call to undefined function`

`sqlsrv_connect()`, donc upgrader ou downgrader PHP.
- Dans PHP 7 : cela fonctionne.

Pour vérifier l'installation, redémarrer le serveur Web, puis vérifier que la ligne `pdo_sqlsrv` s'est bien ajoutée dans la configuration (ex : `http://127.0.0.1/home/index.php?page=php-page&display=extensions`).

### Linux

Pour se connecter au serveur MS-SQL, il suffit de télécharger les pilotes .so[4] correspondant à sa version de PHP, puis d'indiquer leurs chemins dans le PHP.ini[5] :

```
extension=php_pdo_sqlsrv_7_nts.so
extension=php_sqlsrv_7_nts.so
```

Pour vérifier l'installation, redémarrer le serveur Web, puis vérifier que la ligne `pdo_sqlsrv` s'est bien ajoutée dans la configuration (ex : `php -r "phpinfo();" |grep sql`).

### Erreurs

Certaines fonctions permettent d'afficher des erreurs plus précises :

- MySQL : `mysql_error()`.
- MS-SQL : `sqlsrv_errors()`.

## Références

1. http://www.phpsources.org/tutoriel-connection.htm
2. http://www.microsoft.com/en-us/download/details.aspx?id=20098
3. http://www.php.net/manual/fr/ref.pdo-sqlsrv.php
4. https://docs.microsoft.com/en-us/sql/connect/odbc/linux-mac/installing-the-microsoft-odbc-driver-for-sql-server
5. https://www.barryodonovan.com/2016/10/31/linux-ubuntu-16-04-php-and-ms-sql

# Sites

## Unix/Linux

### Raccourcis

Tous les sites installés sur le serveur peuvent simplement être publiés depuis un lien symbolique qui pointe vers eux :

```
$ ln -s /usr/share/phpmyadmin /var/www/phpmyadmin

$ ln -s /usr/share/mediawiki /var/www/mediawiki
```

Ensuite ils sont accessibles par le même nom de domaine :

- //monAdresse/phpmyadmin
- //monAdresse/mediawiki

### apache2.conf

Les répertoires accessibles par Apache peuvent aussi être paramétrés dans[1] :

commande nécessitant les privilèges root

```
# vim /etc/apache2/apache2.conf
```

Lignes :

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

### 000-default.conf

Pour ajouter un site, éditer :

commande nécessitant les privilèges root

```
vim /etc/apache2/sites-available/000-default.conf
```

### .htaccess

Un fichier .htaccess peut aussi gérer les sites[2].

commande nécessitant les privilèges root

```
# vim /var/www/.htaccess
```

*Pour plus de détails voir : **Apache/.htaccess**.*

### UserDir

UserDir est un module Apache permettant d'accéder à un site différent par dossier d'utilisateur Unix (/home/...) en tapant son nom dans l'URL :

*Pour plus de détails voir : **Apache/UserDir**.*

### Fichier hosts

Cette solution est la plus adaptée quand plusieurs domaines (donc URL) pointent vers le même serveur (adresse IP).

Dans un premier temps, il faut associer les noms des sites, à la manière des DNS [3] :

commande nécessitant les privilèges root

```
# vim /etc/hosts
```

Le contenu du fichier va servir à renvoyer les connexions de l'une des deux colonnes de chaque ligne, vers l'autre colonne de la même ligne :

```
127.0.0.1 mediawiki
127.0.0.1 phpmyadmin
```

## Windows

Une première solution est que chaque site soit un sous-répertoire de `localhost`.

### Fichier host

Même principe qu'en Unix-like dans C:\Windows\System32\drivers\etc\host.

## VirtualHost

Une fois que les noms des sites sont connus de l'hôte, des Vhost peuvent être définis avec.

*Pour plus de détails voir : **Apache/Serveurs virtuels**.*

### Plusieurs comptes

Parfois on a besoin d'un compte utilisateur par site, par exemple pour y conférer des accès FTP différents. Dans ce cas il faut installer le module suExec[4] et ajouter une ligne aux vhost de chaque site avec le nom et le groupe de leur compte Unix :

```
SuexecUserGroup MonUtilisateur MonGroupe
```

Puis :

```
apt-get install apache2-suexec
a2enmod suexec
```

Par ailleurs, le compte par défaut utilisé par Apache (www-data) peut être modifié dans :

```
sudo vim /etc/apache2/envvars
```

# Références

1. http://doc.ubuntu-fr.org/apache2
2. httpd.apache.org (http://httpd.apache.org/docs/trunk/)
3. http://doc.ubuntu-fr.org/tutoriel/virtualhosts_avec_apache2
4. https://httpd.apache.org/docs/2.4/fr/suexec.html

# Serveurs virtuels

## Principe

Apache peut gérer plusieurs sites web simultanément. Ils seront tous accessibles à partir de la même adresse IP et du même port.

Pour les différencier, Apache se sert de l'adresse demandée par le navigateur.

Par exemple si site1.com et site2.com pointent sur la même adresse IP, les URL http://site1.com/ et http://site2.com/ aboutiront sur le même serveur.

Mais au moment de la requête, le navigateur précise qu'il a demandé l'adresse http://site1.com/ ou http://site2.com/.

Apache se sert de cette information pour savoir quel site afficher. On parle de *serveur virtuel* ou *virtual host*.

## Configuration

Pour indiquer à Apache quel site correspond à un nom de domaine, on utilise une section `<VirtualHost *>`. Sous Debian, il y a généralement un fichier par section `VirtualHost` dans le répertoire `/etc/apache2/sites-available`.

La section devra contenir une directive ServerName[1] qui indiquera le nom associé à ce *serveur virtuel*.

Elle pourra également contenir une directive ServerAlias si on veut que d'autres noms aboutissent à ce site.

Par exemple[2] :

- En Windows éditer `C:\Program Files (x86)\EasyPHP\binaries\conf_files\httpd.conf`
- En Unix-like : `/etc/apache2/httpd.conf` ou `/etc/apache2/apache2.conf`

```
<VirtualHost MonIP:80>
  ServerAdmin admin@site1.com
  DocumentRoot /home/site1/public_html
  ServerName site1.com
  ServerAlias www.site1.com
</VirtualHost>

<VirtualHost MonIP:80>
  ServerAdmin admin@site2.com
  DocumentRoot /home/site2/public_html
  ServerName site2.com
  ServerAlias www.site2.com
  AccessLog /home/site2/access.log
  ErrorLog /home/site2/error.log
  <Directory /home/site2/public_html>
    AllowOverride All
  </Directory>
</VirtualHost>
```

Pour affecter tous les sites et ports, remplacer ceux-ci dans la première balise par *.

En cas d'erreur Apache d'ajouter une "directive" lors de sa relance, ajouter une ligne *NameVirtualHost MonIP:MonPort*.

La documentation d'Apache sur les serveurs virtuels[3] contient des informations détaillées sur le sujet.

Pour que ce serveur virtuel fonctionne, il est impératif que les noms site1.com et www.site1.com soient connus par la machine qui tente d'y accéder (celle qui lance le navigateur).

Pour cela il y a plusieurs méthodes :

- acheter le nom de domaine en question et le configurer pour qu'il pointe sur la bonne adresse IP
- utiliser un serveur DNS qui renverra la bonne IP pour ce domaine
- modifier le fichier `hosts` sur la machine cliente pour faire correspondre ce domaine à la bonne adresse IP (voir le livre Installation et configuration d'une carte réseau)

# Références

1. http://httpd.apache.org/docs/2.2/mod/core.html#servername
2. https://httpd.apache.org/docs/2.4/fr/vhosts/examples.html
3. http://httpd.apache.org/docs/2.2/vhosts/

# UserDir

Le module UserDir permet à tous les utilisateurs de la machine de publier des documents. Pour cela ils ont un sous-répertoire appelé *public_html* dans leur répertoire personnel. Les fichiers mis dans ce répertoire sont accessibles à *http://serveur/~utilisateur*.

Sous Debian : le module UserDir est installé par défaut. Si ce n'est pas le cas :

commande nécessitant les privilèges root

```
# a2enmod userdir && /etc/init.d/apache2 reload
```

Créons-nous une page utilisateur :

commande

```
$ mkdir ~/public_html && echo "<html><body>Notre test du module UserDir</body>
</html>" > ~/public_html/index.html
```

et testons

commande

```
$ lynx http://localhost/~admin
```

# URL Rewriting

## Principe

Sur Internet on distingue globalement quatre types de redirection automatique[1] des navigateurs vers d'autres adresses que celles sur lesquelles ils arrivent :

1. HTML (`<meta http-equiv="refresh" content="1; URL=http://destination.fr">`).
2. PHP (`header('Location: http://destination.fr');`).
3. Javascript[2].
4. HTTP, proposant elle-même plusieurs techniques décrites ci-dessous.

Les règles de réécriture appliquées sur tout le serveur Apache se situent dans `/etc/apache2/sites-available/default`[3] ou `/usr/local/apache/conf/httpd.conf`.

Les fichiers .htaccess peuvent également en contenir[4] pour personnaliser chaque répertoire.

D'une manière générale, Apache permet de rediriger vers une autre page grâce à :

1. Dans les systèmes Unix, la ligne `Options +FollowSymLinks` permet au navigateur de suivre les liens symboliques, en naviguant dans le système de fichier.
2. `Alias`[5].
3. `AliasMatch` (pareil avec du regex).
4. `RewriteRule` réécrit l'URL selon des règles pouvant être précisées en regex[6]. Il nécessite d'être installé sous peine d'erreurs 500, via la commande Unix `a2enmod rewrite`.
5. `Redirect` renvoie simplement ailleurs.
6. `RedirectMatch` (pareil avec du regex).

> **Attention !**
>
> Définir une redirection définitive (`Redirect permanent`) ne s'annule pas seulement en changeant le code qui l'a définit, mais dure jusqu'à expiration du cache serveur. Pour forcer sa suppression immédiate, lancer par exemple :
>
> ```
> /usr/local/apache/bin/htcacheclean -p/var/cache
> /edb/dep/usr/local/portage-ovh/www-apache -l500M
> /etc/init.d/apache restart
> ```

### AllowOverride

Attention, si vous n'avez pas créé de VirtualHost, un VirtualHost par défaut est utilisé par apache2 qui ignore les fichiers .htaccess

Pour y remédier :

```
sudo vi /etc/apache2/sites-enabled/000-default
```

et remplacez AllowOverride none par AllowOverride All (normalement 2 fois), puis relancez apache :

```
sudo /etc/init.d/apache2 reload
```

# Exemples

Dans la syntaxe suivante, le point seul représente le répertoire courant, et le slash seul la racine de l'URL (le domaine).

Pour changer le domaine vers *localhost*, tout en gardant la même URL :

```
SetEnv PHP_VER 5
Redirect / http://localhost/
```

Plus subtile, on peut changer l'ordre des paramètres initiaux dans l'URL de destination[7] :

```
RewriteEngine on
RewriteBase /
RewriteRule ^/xtools/ec/*$ /xtools/pcount/index.php
```

Pour rediriger les requêtes sous certaines condition il existe *RewriteCond*. Par exemple pour que toutes les pages non trouvées renvoient vers l'accueil au lieu d'afficher une erreur :

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php
```

# Références


1. https://openclassrooms.com/courses/la-redirection-http
2. http://ntt.cc/2008/01/21/5-ways-to-redirect-url-with-javascript.html
3. http://saintcarre.dyndns.org/saintcarre/content/tux/Activer-la-r-criture-dURL-sous-Debian-Squeeze
4. http://craym.eu/tutoriels/referencement/url_rewriting.html
5. http://httpd.apache.org/docs/current/fr/mod/mod_alias.html
6. http://www.illiweb.com/manuel/Apache_1.3_VF/mod/mod_rewrite.html#RewriteRule
7. http://www.fbollon.net/node/110

# .htaccess

## Principe

Pour protéger un répertoire en particulier (et ses sous-répertoires), il suffit de placer un fichier nommé `.htaccess` dedans. Apache appliquera instantanément ensuite les règles qu'il contient, uniquement dans cette arborescence.

> **Attention !**
>
> L'explorateur de fichiers de Windows ne permet pas de rebaptiser des fichiers commençant par des points, il faut donc passer par un éditeur de texte.

Par exemple, pour interdire de visualiser les fichiers d'un répertoire qui n'a pas d'index (ex : .html, .php), ajouter le code : `Options -Indexes`.

## Protection par provenance

De nombreux robots tentent quotidiennement de pirater des bases de données (par exemple via PhpMyAdmin). Pour s'en prémunir on peut n'autoriser que deux IP à lire ce répertoire :

```
deny from all
allow from 127.0.0.1
allow from 127.0.0.2
```

Si les plages d'autorisation chevauchent celles d'interdiction, il est possible de préciser leur précédence (l'ordre des lignes dans le fichier ne change rien) :

**`order allow, deny`**
: commence par les autorisation puis démarre les interdictions au risque d'interdire ce qui était autorisé.

**`order deny, allow`**
: le contraire est moins restrictif.

## Protection par mot de passe

### Configuration de l'authentification

Il est impératif que la modification des paramètres d'*authentification* soit autorisée dans la configuration d'Apache.

Il faut que la directive AllowOverride (http://httpd.apache.org/docs/2.2/mod/core.html#allowoverride) d'un répertoire parent contienne l'option `AuthConfig`.

Les directives à placer dans le `.htaccess` sont les suivantes :

**`AuthType basic`**
: type d'authentification communément adopté mais peu sécurisé

**`AuthName "Mon message"`**
: affichera le texte comme invite dans la boîte de dialogue

**`AuthUserFile /etc/apache2/my_passwd`**
: indique où vont se trouver les mots de passe

**`Require valid-user`**
: précise qu'il faut un compte dans le fichier de mots de passe pour accéder au répertoire

On peut aussi utiliser `Require user toto sasa` pour n'autoriser que les comptes *toto* et *sasa*.

Le type d'authentification *basic* fait circuler les mots de passe en clair. Il existe d'autres types plus sécurisés comme *digest*, qu'il est recommandé de combiner à HTTPS. Voir l'article sur wikipédia pour plus de détails sur le fonctionnement.

La première requête adressée à ce répertoire protégé provoquera l'affichage d'une boîte de dialogue par laquelle l'utilisateur devra s'identifier (nom et mot de passe) :

- Si le mot de passe saisi est invalide, la boite de dialogue s'affichera de nouveau.
- S'il est valide, le navigateur l'enregistre et ne le demandera plus.

Il faudra relancer le navigateur pour qu'il le demande de nouveau.

## Fichier de mots de passe

Pour créer un fichier stockant les mots de passe permettant de lire un site, nommé `/etc/apache2/default-passwd` avec comme 1er utilisateur *toto*, on utilisera la commande

```
htpasswd -c /home/user/www/.htpasswd toto
```

Pour ajouter ou modifier un utilisateur à un fichier de mots de passe existant :

```
htpasswd /home/user/www/.htpasswd sasa
```

Pour que le .htaccess active le .htpasswd, y ajouter les directives :

```
AuthName "Page protégée"
AuthType Basic
AuthUserFile "/home/user/www/.htpasswd"
Require valid-user
```

**Attention !**

> Cette protection ne tient pas compte des robots qui essaient tous les mots de passe un par un. Il convient donc de l'utiliser en complément d'un bon pare-feu (ex : iptables).

## Redirections

La syntaxe est la même que dans le fichier de configuration générale d'Apache, sauf que cela n'affectera que le répertoire du fichier .htaccess.

# Cache

## Principe

Le cache web enregistre des pages visitées sur un serveur[1] pour les ré-afficher plus rapidement ensuite. Il doit comporter une date d'expiration qui dépend de la fréquence de changement des pages.

> **Attention !**
>
> Ne pas utiliser de cache pour une préproduction sous peine de ne pas voir immédiatement ses modifications.

## Configuration du serveur

La mise en cache peut être configurée au moyen de plusieurs modules Apache :

```
a2encode expires
a2encode cache
a2encode file_cache
a2encode mem_cache
a2encode cache_disk # pour Apache version 2.4
a2encode disk_cache # pour Apache version 2.2
```

Puis ajouter à apache2.conf :

```
<IfModule mod_expires.c>
        ExpiresActive On
        ExpiresDefault "access plus 1 month"
        <filesMatch "\.(ico|jpg|jpeg|png|gif)$">
                ExpiresDefault "access plus 1 year"
        </filesMatch>
        ExpiresByType image/x-icon "access plus 1 day"
        ExpiresByType text/css "access plus 1 day"
        ExpiresByType application/javascript "access plus 1 day"
</IfModule>

<IfModule mod_cache.c>
    <IfModule mod_cache_disk.c>
        CacheRoot   "/var/cache/apache2/"
        CacheEnable disk /
        CacheDirLevels 2
        CacheDirLength 1
    </IfModule>
    <IfModule mod_mem_cache.c>
        CacheEnable mem /
        MCacheSize 4096
        MCacheMaxObjectCount 100
        MCacheMinObjectSize 1
        MCacheMaxObjectSize 2048
```

```
    </IfModule>
    <IfModule mod_file_cache.c>
        mmapfile /var/cache/apache2/index.html # Liste de page à mettre en cache
    </IfModule>
</IfModule>
```

Enfin, recharger Apache :

```
service apache2 reload
```

# Configuration du site

Côté HTML, on distingue trois balises méta.

Syntaxe sans cache :

```
<meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate, proxy-
revalidate, max-age=0, s-maxage=0, post-check=0, pre-check=0" />
<meta http-equiv="Pragma" content="no-cache, no-store" />
<meta http-equiv="Expires" content="0" />
```

Syntaxe avec cache[2] :

```
<meta http-equiv="Cache-Control" content="Private" />
<meta http-equiv="Pragma" content="" />
<meta http-equiv="Expires" content="" />
```

Pour voir la configuration du cache d'un site, il faut regarder l'entête HTTP, par exemple avec :

```
curl -I http://example.org
```

La durée d'expiration de la mémoire cache doit dépendre de la fréquence de rafraichissement du contenu du site. Toutefois d'une manière générale, il est recommandé de la définir à une valeur comprise entre 48 h et un an.

# Références

1. http://httpd.apache.org/docs/2.4/caching.html
2. http://www.i18nguy.com/markup/metatags.html

# HTTPS

## Généralités

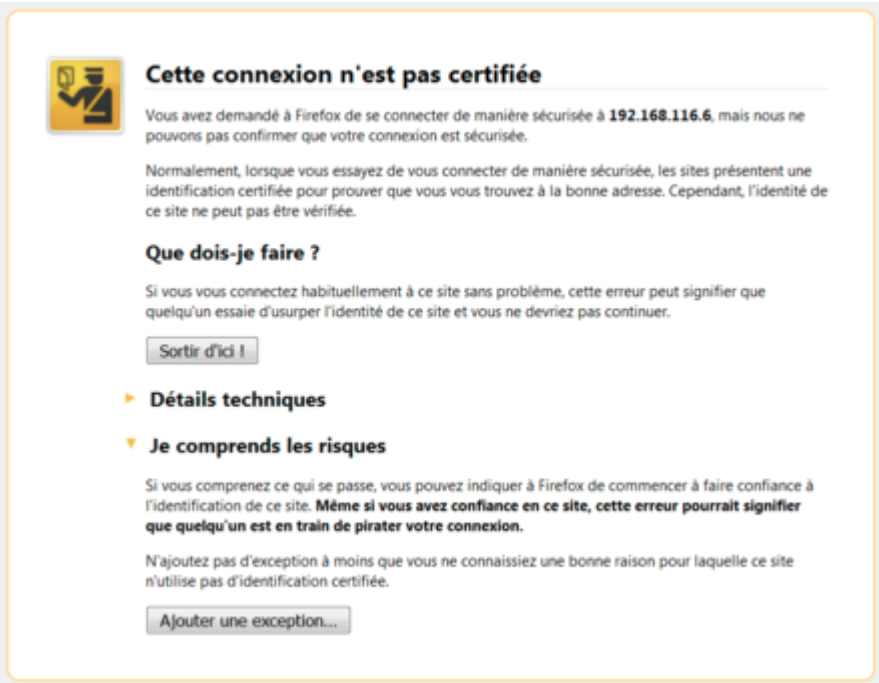
Contrairement au protocole HTTP, HTTPS garantit la confidentialité et l'intégrité des données échangées entre un serveur web et ses clients, et par conséquent il convient mieux aux transactions sensibles comme les flux bancaires. En effet il permet de se prémunir de l'attaque de l'homme du milieu en cryptant les communications.

Pour mettre en place ce protocole, il faut juste activer l'extension Apache et ajouter une directive pour le port 443[1] avec un certificat électronique.

## Types de clé

Un certificat électronique (.crt) est issu d'une demande d'identification (.csr pour Certificate Signing Request[2]). Ce dernier est généralement payant, à renouveler chaque année, car délivré par une autorité de certification, mais :

- Il est possible de le créer soi-même[3], ce qui aura pour effet d'afficher un avertissement d'exception de sécurité aux visiteurs comme celui de l'image ci-contre.
- Sur Ubuntu, il existe déjà /etc/ssl/private/ssl-cert-snakeoil.key, mais l'avertissement sera le même.
- Certains sites comme GeoTrust en propose un valide, mais valable seulement 30 jours[4].



Avertissement à accepter par les visiteurs en cas de certificat autosigné.

- La meilleure solution gratuite est https://letsencrypt.org/. En effet, elle permet de créer et configurer (ou renouveler) des sites HTTPS en une minute chacun grâce à https://certbot.eff.org/.

## Linux

**OpenSSL** est une implémentation open source des protocoles SSL et TLS[5].

## Prérequis

Apache + Mod SSL + OpenSSL (disponible depuis Synaptic).

Ajouter le module SSL à Apache 2[6] :

commande nécessitant les privilèges root

```
# a2enmod ssl
```

Ajouter *Listen 443* à */etc/apache2/ports.conf*

commande nécessitant les privilèges root

```
# echo "Listen 443" >> /etc/apache2/ports.conf
```

Générer un certificat autosigné :

commande nécessitant les privilèges root

```
# apache2-ssl-certificate
```

Si la commande est introuvable :

commande nécessitant les privilèges root

```
# apt-get install ssl-cert
# /usr/sbin/make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
```

On configure ssl :

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl
sudo ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

Éditez le fichier de configuration ****/etc/apache2/sites-enabled/ssl**** pour qu'il accepte les connections sur le port 443 :

```
NameVirtualHost *:443
<VirtualHost *:443>
(...les répertoires et autres configurations si désiré)
```

Éditez le fichier de configuration ****/etc/apache2/sites-available/default**** pour qu'il accepte les connections sur le port 80 :

```
NameVirtualHost *:80
```

```
<VirtualHost *:80>
(...les répertoires et autres configurations si désiré)
```

Dans le fichier ****/etc/apache2/ports.conf****, ajoutez :

```
Listen 443
```

et dans le milieu du fichier ****/etc/apache2/sites-available/ssl**** ajoutez :

```
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/apache.pem
```

Puis redémarrez apache :

```
sudo /etc/init.d/apache2 restart
```

Pour rendre possible la connexion en SSL, la configuration Apache suivante :

```
vim /etc/apache2/apache2.conf
# ou
vim /etc/apache2/sites-available/default-ssl
a2ensite default-ssl
```

doit comprendre dans chaque vhost concerné :

Fichier : le fichier de configuration

```
<VirtualHost *:443>
  SSLEngine on
  #SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
  #SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
  SSLCertificateFile /etc/apache2/ssl/apache.crt
  SSLCertificateKeyFile /etc/apache2/ssl/apache.key
  ...
```

Puis on relance Apache :

```
service apache2 reload
```

Pour tester :

```
curl https://monURL
```

Si la clé nécessite un mot de passe :

```
SSLPassPhraseDialog exec:/etc/ssl/nomdedomaine.fr.pwd
```

### Personnalisation

Pour personnaliser les directives (ex : ajouter *SuexecUserGroup*), il suffit de copier le contenu de :

```
vim /etc/apache2/apache2.conf
```

dans

```
vim /etc/apache2/sites-available/default-ssl.conf
```

en remplaçant `:80` par `:443`.

## Création de la clé

La structure des commandes est celle du shell Unix :

```
openssl command [ command_opts ] [ command_args ]
```

## Exemples

### Générer un fichier ".key"

On choisit de générer une clef de 2048 bits, car c'est le minimum pour les vendeurs de certificats SSL. Voici la commande :

```
openssl genrsa -out nomdedomaine.fr.key 2048
```

### Demande de chiffrement

Générer une demande de certificat avec la commande :

```
openssl req -new -key nomdedomaine.fr.key -out nomdedomaine.fr.csr
```

Pour la question *State or Province Name*, mettre son département (français).

Puis on peut l'envoyer à une autorité de certification, ou bien l'autosigner avec :

```
openssl x509 -req -days 365 -in nomdedomaine.fr.csr -signkey nomdedomaine.fr.key -out
```

```
nomdedomaine.fr.crt
```

Récapitulatif testé sur Ubuntu[7] :

```
openssl genrsa -des3 -out server.key 2048
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Ensuite il faut dire à Apache d'utiliser le .crt.

# Windows

## Prérequis

Ajouter le module SSL en décommentant la ligne suivante de `httpd.conf` :

```
#LoadModule ssl_module modules/mod_ssl.so
```

La directive sur le port 443 s'effectue ensuite dans le fichier suivant (vide par défaut), selon le logiciel :

- `C:\Program Files (x86)\EasyPHP\binaries\apache\conf\inc_virtual_hosts.conf`
- `C:\Program Files (x86)\WAMP\bin\apache\Apache2.2.21\conf\extra\httpd-vhosts.conf`

Mais `httpd.conf` contient un commentaire avec : `#<VirtualHost _default_:443>`.

Il faut y renseigner l'emplacement du `ssl.crt` ci-dessous à créer avant de décommenter, sous peine d'erreur Apache.

## Création du certificat autosigné

Lancer une console DOS :

```
>cd "C:\Program Files (x86)\EasyPHP\binaries\apache\bin"
>openssl req -config "C:\Program Files (x86)\EasyPHP\binaries\php\php_runningversion
\extras\ssl\openssl.cnf" -new -out Certificat_1.csr
WARNING: can't open config file: c:/openssl-1.0.1e/ssl/openssl.cnf
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.........++++++
.........++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
...
>openssl rsa -in privkey.pem -out Certificat_1.key
>openssl x509 -in Certificat_1.csr -out Certificat_1.cert -req -signkey
Certificat_1.key -days 365
```

Cela a généré les fichiers à renseigner dans la directive :

- `C:\Program Files (x86)\EasyPHP\binaries\apache\bin\Certificat_1.csr`.
- `C:\Program Files (x86)\EasyPHP\binaries\apache\bin\Certificat_1.key`.
- `C:\Program Files (x86)\EasyPHP\binaries\apache\bin\Certificat_1.cert`.

Le CSR n'a pas besoin d'être renseigné dans les directives, le certificat fichier expire après 365 jours[8].

# Autres aspects de sécurité

Le module `headers`[9] peut assurer la protection contre le XSS[10] et le clickjacking. Exemple :

```
 <IfModule mod_headers.c>
       # Filtre sur les provenances des scripts, séparées par des espaces
       Header set Content-Security-Policy "default-src 'self' *.nomdedomaine.fr
*.nomdedomaine.com"
       # Indication anti-XSS pour les navigateurs
       Header always set X-XSS-Protection "1; mode=block"
       # Anti-clickjacking
       Header always set X-FRAME-OPTIONS "SAMEORIGIN"
       # Antivol de cookie
       Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
 </IfModule>
```

> **Attention !**
>
> Cela bloque les iFrames.

Pour tester une configuration, mieux vaut commencer par un .html plutôt que de la déployer sur tout le serveur en le redémarrant puis de faire machine arrière. Si en définissant la règle la plus permissive Firefox bloque tout de même quelque chose (Content Security Policy: Les paramètres de la page ont empêché le chargement d'une ressource à self ), il s'agit d'un bug connu qui n'affecte pas les autres navigateurs :

```
<head>
    <meta http-equiv="Content-Security-Policy" content="default-src *">
</head>
```

Par ailleurs, des sites d'audit gratuits peuvent ensuite révéler s'il reste des failles[11].

# Rediriger le flux HTTP vers HTTPS

Entrer la configuration suivante dans le fichier *https.conf* :

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

Pour le webhosting, le réglage doit être effectué à l'aide d'un fichier .htaccess, avec la configuration suivante :

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

# Références


1. https://www.startssl.com/?app=21
2. https://www.isicca.info/certificat-ssl-generer-son-certificat/
3. http://doc.ubuntu-fr.org/tutoriel/comment_creer_un_certificat_ssl
4. https://www.ssl247.fr/certificat-ssl-gratuit
5. http://www.openssl.org/
6. http://doc.ubuntu-fr.org/tutoriel/securiser_apache2_avec_ssl
7. https://doc.ubuntu-fr.org/tutoriel/comment_creer_un_certificat_ssl
8. http://www.finalclap.com/faq/414-certificat-ssl-https
9. https://httpd.apache.org/docs/trunk/fr/mod/mod_headers.html
10. http://content-security-policy.com/
11. https://www.dareboost.com/fr
    - https://commaster.net/content/how-setup-lets-encrypt-apache-windows

# CGI

Le CGI (Common Gateway Interface) est une norme permettant à Apache d'exécuter des programmes écrits en n'importe quel langage (Bash, C, Java, Perl, PHP, Python...), du moment qu'il est exécutable et qu'il respecte certaines contraintes d'entrées/sortie.

## Configurer l'accès aux scripts CGI

Pour qu'Apache prenne en charge les scripts, il est nécessaire d'effectuer un minimum de paramétrage dans la configuration du site.

### Activer le module

```
a2enmod cgi
```

### ScriptAlias

La directive (de httpd.conf) :

```
ScriptAlias /cgi-bin/ /chemin des scripts/
```

précise le nom du répertoire où Apache est autorisé à exécuter des scripts CGI[1].

Exemple Unix :

```
ScriptAlias /cgi-bin/ /var/www/cgi-bin/
```

Exemple Windows, utiliser le format URL (pas d'antislash) :

```
ScriptAlias /cgi-bin/ "C:/wamp/bin/apache/apache2.2.27/cgi-bin/"
```

En fait le chemin `/cgi-bin/` n'existe pas vraiment, il est dirigé vers le chemin des scripts défini par la directive, et cela permet d'écrire des URL comme `http://serveur/cgi-bin/mon_script`.

### ExecCGI

La clause suivante active l'option `ExecCGI` dans `/var/www/cgi-bin`, ce qui autorise Apache à exécuter les scripts sur le serveur :

```
<Directory /var/www/cgi-bin>
  Options ExecCGI
```

```
</Directory>
```

Par exemple : vous écrivez un script `essai.cgi`, et vous voulez que `/home/httpd/cgi-bin` contienne les scripts.

Il faut donc au moins écrire :

```
<Directory /home/httpd/cgi-bin>
  Options ExecCGI
</Directory>
```

L'appel à un script essai.cgi sera effectué par l'URL : `http://serveur/cgi-bin/essai.cgi`

### AddHandler

Cette clause permet de choisir les extensions de fichiers qui seront autorisés, ex :

```
AddHandler cgi-script .cgi .exe .pl .py .vbs
```

### Récapitulatif

Exemple complet sur Windows, dans la configuration Apache :

```
ScriptAlias /cgi-bin/ "E:/www/cgi-bin/"
<Directory "E:/www/cgi-bin/">
  Options FollowSymLinks Indexes
  AllowOverride All
  Order deny,allow
  Allow from all
  Require all granted
</Directory>
```

Dans `E:/www/cgi-bin/.htaccess` :

```
AddHandler cgi-script .cgi .exe .pl .py .vbs
```

## Écrire un programme CGI

La contrainte principale concerne la sortie du programme. Si un programme CGI génère des données sur sa sortie standard, il doit les précéder d'un en-tête HTTP permettant de les identifier.

### Bash

Voici un exemple de programme CGI écrit en bash :

```
#!/bin/bash

# Header
echo "Content-type: text/html"

# Fin du header
echo ""

# Contenu à afficher dans le navigateur
echo "<html><body>Hello World!</body></html>"
```

Ce script génère une page HTML.

### Perl

```
#!c:/perl/perl/bin/perl.exe -w
use CGI;
my $query = new CGI;
my $Name = $query->param('Name');
print $query->header();
print "Hello World!"
```

### Python

```
#!C:\Program Files (x86)\Python\python.exe
# -*- coding: UTF-8 -*-
print "Content-Type: text/plain;charset=utf-8"
print
print "Hello World!"
```

*Pour plus de détails voir :* ***Programmation Python/L'interface CGI***.

### VBS

Pour Windows[2].

```
''!c:/windows/system32/cscript //nologo
Wscript.Echo "Content-type: text/html" & vbLF & vbLF
WScript.Echo "Hello World!"
Wscript.Quit 0
```

## Références

1. http://httpd.apache.org/docs/current/fr/howto/cgi.html
2. http://wiki.uniformserver.com/index.php/CGI:_VBScript_CGI

</noinclude>

# Débogage

## Logs

Contrairement à ce qui est dit quand Apache bloque au démarrage, il ne faut pas utiliser `journalctl -xe` pour lire les logs car il tronque chaque ligne à 125 caractères. Il faut lancer :

```
tail /var/log/apache2/error.log
```

## Lecture seule

S'il est impossible de créer des fichiers ou répertoires depuis un navigateur c'est que le serveur n'autorise pas l'utilisateur *apache* à le faire dans le répertoire du site.

## Les fonctions PHP s'affichent sur la page au lieu de s'exécuter

Si `a2enmod php7.2` indique que le module est déjà installé, c'est peut-être lié à `a2enmod userdir`. Cela peut se régler avec :

```
vim /etc/apache2/mods-enabled/php7.2.conf
```

Commenter les lignes :

```
<IfModule mod_userdir.c>
...
</IfModule>
```

Et relancer Apache.

## 403 forbidden, client denied by server configuration

Ce message apparait dans les logs Apache quand le fichier auquel on tente d'accéder est protégé dans apache2.conf, par un `deny`, un `require` ou une absence de ce dernier. Dans ce cas, il faut l'ajouter :

```
<Directory /home>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

## Configuration error: No MPM loaded

Restaurer le *apache2.conf* d'origine, il doit y avoir une erreur dans la directive `ServerRoot`.

# Load denied by X-Frame-Options: ... does not permit cross-origin framing

Il faut juste autoriser les iFrames vers votre site, en commentant dans `apache2.conf` la ligne qui commence comme :

```
Header always set X-FRAME-OPTIONS "SAMEORIGIN"
```

# Missing suexec binary

Installer le module :

```
sudo apt-get install apache2-suexec-custom
```

# suEXEC is disabled

Vérifier que le module est activé.

# Erreurs vhost

Les problèmes suivants peuvent survenir lors des relances Apache.

### Invalid command 'SuexecUserGroup'

Vérifier que le module est activé.

### apache2: bad user name Utilisateur1

Un utilisateur Unix appelé dans la configuration n'existe pas. Il faut donc le créer :

```
useradd Utilisateur1
```

### apache2: bad group name Groupe1

Un groupe Unix appelé dans la configuration n'existe pas. Il faut donc le créer :

```
groupeadd Groupe1
```

### No such file or directory:… Cannot access directory '/etc/apache2/logs/'… Configuration check failed

Un répertoire Unix appelé dans la configuration n'existe pas. Il faut donc le créer :

```
mkdir /etc/apache2/logs
```

### exit signal Segmentation fault (11)

Cela peut survenir quand PHP rencontre une erreur. Pour la connaitre précisément, il faut lancer le .php en shell (sans Apache). Exemple :

```
su www-data

php5 -q SendMail.php
  SMTP Error: Could not connect to SMTP host.

# Ou encore en écrivant le script sans .php :
php5 -r "chown('/home/Compte2', 'Compte1');"
  PHP Warning:  chown(): Operation not permitted in Command line code on line 1
# Vérification en shell
chown Compte1 /home/Compte2
  chown: modification du propriétaire de «/home/Compte2»: Opération non permise
```

## Erreurs HTTPS

### Échec de la connexion sécurisée. SSL a reçu un enregistrement qui dépasse la longueur maximale autorisée. (Code d'erreur : ssl_error_rx_record_too_long)

Le module d'Apache est activée mais son vhost est absent ou sa configuration ne contient pas `SSLEngine on`.

### curl: (35) error:140770FC:SSL routines:SSL23_GET_SERVER_HELLO:unknown protocol

Il faut refaire comme il faut la configuration Apache ci-dessus.

### curl: (60) SSL certificate problem: self signed certificate

C'est que la connexion fonctionne, les visiteurs doivent juste acquitter le message de leur navigateur *Cette connexion n'est pas certifiée*.

### Enter passphrase for SSL/TLS keys for à chaque relance Apache

Vérifier la présence de la ligne suivante dans la configuration : `SSLPassPhraseDialog exec:/etc/ssl/nomdedomaine.fr.pwd`. Sinon, fdaire sauter le mot de passe :

```
openssl rsa -in nomdedomaine.fr.key -out nomdedomaine.fr.key.nopass
```

### blocage du contenu mixte actif (mixed active content)

Il faut remplacer HTTP par HTTPS dans le code source du site Web.

## Dans les logs SSL

### RSA certificate configured for 127.0.0.1:443 does NOT include an ID which matches the server name

Il faut générer la clé en utilisant un FQDN.

### CSR contains unsupported extensions

Le mot de passe du .csr contient des caractères spéciaux incompatibles, comme ! ou _. Il faut se contenter de l'alphanumérique.

## Dans les logs Apache

### Server should be SSL-aware but has no certificate configured

Réinstaller clé SSL.

### Init: Unable to read server certificate from file ...csr

Le .cert est introuvable.

### SSL Library Error: error:0906D06C:PEM routines:PEM_read_bio:no start line (Expecting: TRUSTED CERTIFICATE) -- Bad file contents or format - or even just a forgotten SSLCertificateKeyFile?

Il faut convertir le certificat SSL ainsi :

```
openssl x509 -inform der -in /etc/apache2/ssl/apache.crt -outform PEM -out
/etc/apache2/ssl/apache.pem
vim /etc/apache2/sites-enabled/default-ssl.conf
# remplacement du .crt par le .pem
service apache2 restart
```

Sinon il faut retirer les BOM du certificat.

Sinon il faut retirer les autres fichiers du dossier contenant la clé et de celui du certificat (un autre), en leur conférant au maximum du `chmod -R 644`[1].

### Certificate and private key do not match

Le certificat doit être un fichier .pem et la clé .key. Il est possible de le vérifier avec la commande suivante[2] :

```
$ (openssl x509 -noout -modulus -in /etc/ssl/certs/ssl-cert-snakeoil.pem | openssl md5
;openssl rsa -noout -modulus -in /etc/ssl/private/ssl-cert-snakeoil.key | openssl md5)
| uniq
(stdin)= 8cf9b840c3239f653be542149497f047
```

Quand les deux certificats correspondent il n'y a qu'une seule ligne, comme ci-dessus. Il faut donc retrouver le bon ou régénérer la paire. Pour y voir plus clair, la commande `file` permet de les identifier :

```
$ file /etc/ssl/*
nomdedomaine.fr.crt:      PEM certificate request
nomdedomaine.fr.csr:      PEM certificate request
nomdedomaine.fr.key:      PEM RSA private key
nomdedomaine.fr.pem:      PEM certificate
nomdedomaine.fr.pwd:      ASCII text
certs:                    directory
openssl.cnf:              ASCII text
private:                  directory
apache.key:               PEM RSA private key
apache.crt:               PEM certificate
```

### AH01909: nomdedomaine.fr:443:0 server certificate does NOT include an ID which matches the server name

Le certificat installé est prévu pour un autre domaine ou sous-domaine. Il faut peut-être en acheter un pour tous les sous-domaines, ce qui se dit *certificats SSL à validation de domaine* (*wildcard certificate* en anglais).

### Pass phrase incorrect for key

Soit on peut regénérer le fichier de la directive `SSLPassPhraseDialog`, soit on fait sauter le mot de passe demandé par la clé : `openssl rsa -in privateAvecPassPhrase.key -out private.SansPassPhrase.key`.

# Erreurs .htaccess

## Inopérant

Si le .htaccess ne produit aucune redirection, vérifier que le module Apache est bien activé :

```
a2enmod rewrite
```

Et que la directive suivante figure au moins dans un répertoire parent du .htaccess :

```
AllowOverride All
```

Puis relancer Apache.

Dans le vhost du site, remplacer "*:80" par "IP_du_serveur:80".

### Request exceeded the limit of 10 internal redirects due to probable configuration error. Use 'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace.

Il existe une redirection circulaire.

## Erreurs CGI

### Error 500 *Erreur du serveur!*

Remplacer un `Deny from all` par un `Allow from all`. Sinon, regarder les logs. Par exemple, cela peut provenir de `suexec policy violation` => commenter la directive `SuexecUserGroup`.

### Error 403 *Accès interdit*

Lister ce répertoire est interdit, il faut donc connaitre l'URL des fichiers qu'il contient.

- Le code source du fichier à exécuter s'affiche dans le navigateur, ou ce dernier propose de le télécharger : le `Addhandler` est manquant (exemple dans le .htaccess ci-dessus). Sinon c'est le module qui n'est pas activé (a2enmod cgi).

### *couldn't create child process*

Remplacer le chemin après le shebang. Par exemple :

- `#!/usr/bin/perl` par `#!c:/perl/perl/bin/perl.exe -w`.
- `#!/usr/bin/env python` par `#!C:\Program Files (x86)\Python\python.exe`.

### *End of script output before headers*

En-tête manquante dans le contenu affiché par le script (ex : déplacer l'importation avant `print "Content-Type: text/plain;charset=utf-8"`). Mais cela peut aussi être dû à un message d'erreur dans le script à exécuter.

### *malformed header from script: Bad header:*

L'en-tête n'est pas adaptée (ex : remplacer `#print "Content-Type: text/plain;charset=utf-8"` par `print "Content-type: text/html\n\n"` s'il y a un `print "<html>"` après).

Sinon consulter les logs Apache...

# Références


1. https://www.certificats-ssl.com/support/Installation-dun-certificat-avec-Apache-httpd.html
2. http://stackoverflow.com/questions/23652680/postfix-cannot-get-rsa-private-key-from-file-etc-ssl-private-server-key-disabl




---