Apache – Authentification, Autorisation et contrôle d'accès

Apache est le serveur Web le plus répandu de la planète ; dans ce petit document, nous verrons comment le paramétrer pour authentifier des utilisateurs lorsqu'ils naviguent sur votre serveur web, interdire/autoriser/contrôler l'accès à certains répertoires du serveur à des utilisateurs ou groupe d'utilisateurs bien définis. Ceci peut être particulièrement intéressant si vous avez des informations sensibles sur votre serveur web destinées uniquement à un petit groupe d'utilisateurs ; vous pourrez ainsi n'autoriser que ceux-ci à avoir accès aux données sensibles, et vous saurez également par qui et quand est-ce que les données ont été utilisées. En général, les informations sensibles sont placées dans un répertoire, et c'est ce répertoire qui sera protégé.

Deux possibilités existent : utiliser des directives <directory> dans le fichier de configuration de apache, ou bien à travers des fichiers « .htaccess ». Cependant, pour utiliser les fichiers « .htaccess », il faudrait que dans la directive <directory> du répertoire ou du répertoire parent il n'y ait pas cette ligne : **AllowOverride None**. Mettez tout ce que vous voudrez (exples : **AllowOveride** `AuthConfig` ou **AllowOverride All**) mais pas le None. Il n'y a vraiment pas une grande différence sur le plan de la configuration, parce que le fichier .htaccess qu'on place dans le répertoire à protéger et la directive <directory> contiennent les mêmes informations. Dans la suite du document, je n'utilise que la directive <directory>, pour avoir le .htaccess, c'est du copier/coller.

Alors pour ces deux techniques, il existe plusieurs méthodes pour avoir la liste des utilisateurs qui ont accès aux fichiers du répertoire à protéger. Je décrirai les 3 méthodes que j'ai déjà implémentées et testées : les utilisateurs du système (/etc/passwd et /etc/shadow), les utilisateurs pris dans un fichier créé spécialement pour ça (avec la commande htpasswd), les utilisateurs pris dans une base de données mysql. Avec ces 3 méthodes, nous protègerons le répertoire /var/www/html/vacance (comme exemple).

I – Avec utilisation de htpasswd (fichier spécial) :

Coller les lignes suivantes dans la section principale du fichier de config de Apache, ou bien placer les dans un fichier que vous placerez dans un répertoire d'inclusion de config de Apache. Dans ma config, je crée le fichier vacance.conf que je place dans le répertoire /etc/httpd/conf.d/.

```
<Directory "/var/www/html/vacance">
   AllowOverride AuthConfig
   Order allow,deny
   Allow from all
   AuthType Basic
   AuthName "Test de protection"
```

```
   AuthUserFile /etc/httpd/conf.d/.passwd.vacance
   <Limit GET POST>
   require valid-user
   </Limit>
</Directory>
```

Ces lignes précises que seuls des utilisateurs valides auront l'accès au répertoire /var/www/html/vacance, et que la liste de ces utilisateurs avec leurs mots de passe correspondants se trouve dans le fichier /etc/httpd/conf.d/.passwd.vacance.
Pour créer et remplir le fichier .passwd.vacance, il faut utiliser la commande htpasswd.
Ajoutons 2 utilisateurs dans ce fichier :
]# htpasswd -bc /etc/httpd/conf.d/.passwd.vacance roger monpasswd
]# htpasswd -b /etc/httpd/conf.d/.passwd.vacance roger2 monpasswd2
Notez bien la petite difference entre les 2 commandes. La première crée le fichier avec l'option -c, la deuxième n'a plus besoin de créer le fichier qui existe déjà, donc pas d'option –c.

Mon répertoire vacance est ainsi accessible que par roger et roger2.

II – Utilisation des utilisateurs du système :

Pour cela, il faut que le module mod_auth_pam soit installé (http://pam.sourceforge.net/mod_auth_pam/download.html).

Dans ce cas, voici le contenu de notre fichier /etc/httpd/conf.d/vacance.conf :

```
<Directory "/var/www/html/vacance">
   AllowOverride None
   Order allow,deny
   Allow from all
   AuthPAM_Enabled on
   AuthType Basic
   AuthName "Test de protection"
   <Limit GET POST>
   require valid-user
   </Limit>
</Directory>
```

Avec ces directives, les accès au répertoire vacance sont gérés par le fichier /etc/shadow.

III – Utilisation d'une base de données mysql :

Pour ce cas, on dispose d'une base de données mysql vacusersdb ; l'utilisateur de la base de données est vacuseradm et son mot de passe d'accès à la base est vacpasswd. La table de la

base de données contenant la liste des utilisateurs est vacusers, et ses champs username et password contiennent respectivement les noms d'utilisateur et mots de passe des différents utilisateurs.

Le fichier /etc/httpd/conf.d/vacance.conf sera donc :

```
<Directory "/var/www/html/vacance">
        AllowOverride All
        AuthType Basic
        AuthName "Test de protection"
        AuthMYSQLEnable on
        AuthMySQLDB vacusersdb
        AuthMySQLUser vacuseradm
        AuthMySQLPassword vacpasswd
        AuthMySQLUserTable vacusers
        AuthMySQLNameField username
        AuthMySQLPasswordField  password
#         AuthMySQLCryptedPasswords On
        require valid-user
</Directory>
```

Sources :

http://httpd.apache.org/docs/1.3/howto/htaccess.html
http://cchatelain.developpez.com/articles/web/apache/htaccess/
http://eric.extremeboredom.net/2004/10/
http://httpd.apache.org/docs/1.3/howto/auth.html

Roger YERBANGA
www.rogeryerbanga.fr.st