

## Protéger son site web via htaccess et htpasswd

Voici un petit tutoriel tout bête pour comprendre comment mettre en place une demande de login et mot de passe pour un site web.

Cette méthode est très pratique si vous souhaitez empêcher l'accès à certains sites qui ne sont pas protégés eux-même par une demande de login.

J'utilise pour ce tutoriel l'OS Ubuntu-Server 8.04, avec apache2, mais il ne devrait pas y avoir de différence avec une autre distribution GNU/Linux.

Pour mettre cette protection en place, il vous faudra créer un fichier contenant les logins et mots de passe (qui seront cryptés en md5). Vous pouvez avoir le choix d'en créer un par site web, ou un pour tous, et demander au site web un login spécifique contenu dans le fichier. Pour faire plus simple, j'utiliserai la méthode un fichier par site.

Tout d'abord, créons le fichier qui contiendra le login et le mot de passe, soyez prudent, on ne met pas ce fichier n'importe où, et surtout pas en 777 hein :) Je vous conseille de pas le laisser traîner dans votre home !

Voici les commandes :

```
touch votrefichierlogins  
htpasswd -cb votrefichierlogins login motdepasse
```

Le mot de passe sera crypté, vous pouvez vérifier en faisant un :

```
cat votrefichierlogins
```

Maintenant, modifions la configuration d'apache. Tout ce passe dans le fichier **default** dans le répertoire **/etc/apache2/sites-available/** si vous définissez les alias de vos sites à cet endroit, ou alors dans le fichier du site web dans **/etc/apache2/conf.d/**.

Bref, repérez la déclaration de l'alias concerné, et rajoutez lui les lignes en italique :

```
Alias /sysinfo "/var/www/phpsysinfo/"  
<Directory "/var/www/phpsysinfo/">  
    AuthName "Login et mdp requis"  
    AuthType Basic  
    AuthUserFile "/chemin/votrefichierlogins"  
    Require valid-user  
    allow from all  
</Directory>
```

**AuthName** : petit message affectif pour quand vous aurez l'invite de login xD

**AuthUserFile** : le chemin vers votre fichier de login.

Maintenant, rechargez la config d'apache par la commande :

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

Testez l'accès à votre site, c'est magique !

Si vous avez des questions, un avis, des idées pour améliorer ce tutoriel, n'hésitez pas à me contacter via mon site <http://www.shivaserv.fr>