Implémentation de service WebFTP sécurisé

Arnaud A. A. AMELINA

AUF-CNF de Lomé

15 Décembre 2010

Sommaire

Objectif

Touver une application web pouvant permettre aux abonnés des CNF d'accéder via n'importe quel navigateur à leurs données se trouvant dans leur répertoire personnel à travers internet et sans avoir à installer une application particulière. En prenant soin d'éviter que le mot de passe se fasse sniffer en cours de route, donc une couche de sécurité via SSL/TLS est nécessaire.

Le Public cible

Cette application ou interface sera mise à la disposition des abonnés des campus numérique et des campus partenaires ou centre d'accès à l'information des institutions partenaires de l'Agence.

Fonctions primordiales

- Open source
- Installateur simple
- Pas de base de données pour la gestion des fichiers
- Simple d'usage
- Projet actif
- Gestion correcte des caractères accentués dans le nom du fichier/dossier
- Gestion correcte des utilisateurs pouvant accéder aux fichiers via FTP.
- Sécurité d'authentification des utilisateurs
- Pouvoir uploader plusieurs fichiers en même temps
- Compatible avec Firefox

Fonctions qui seraient un plus

- Visualisation/ouverture des documents sans téléchargement
- Possibilité de diaporama sur les fichiers
- Pouvoir choisir un thème simplement
- Pas de pub trop envahissante sur le logiciel dans l'usage de celui-ci
- Un menu spécifique dans le clic droit

Les applications conçues en PHP simple

- Net2Ftp : dernière M-à-J : 2009-09-06 Version : 0.98
- eXtplorer : dernière M-à-J : 2010-06-11 Version : 2.0 RC1
- Agency4Net : dernière M-à-J : 2008-05-20 Version : 1.1
- WebFTP : dernière M-à-J : 2006-12-31 Version : 2.0
- Web-FTP : dernière M-à-J : 2004-01-12 Version : 2.2.1

Les applications conçues en PHP + Javascript

- Ajaxplorer : dernière M-à-J : 2010-11-29 version : 3.1.1
- PhpXplorer : dernière M-à-J : ??-??-?? Version : 0.9.37
- webftp-ajax : dernière M-à-J : 2006-12-31 Version : 2.0
- Webshare : dernière M-à-J : 26 juillet 2009 Version : 0.8.2 Alpha

Les applications conçues sous d'autres langages

- yawebftp : dernière M-à-J :2004-09-23 version : 1.0 en Java
- Web-FTP : dernière M-à-J : 2004-01-12 Version : 2.2.1 en Perl

La démonstration de toutes les applications testées se fera en direct et en ligne. Pour éviter d'alourdir la présentation par les captures d'écran.

Projets en abandon ou stationnaires

- Web-FTP(Perl) : utilise CGI et demande trop de bidouille dans le php.ini
- yawebftp : Utilise un applet Java, fonctionne après bidouille
- webftp-ajax : pas de gestion d'utilisateurs, ...
- Web-FTP(PHP) : Nécessite des bidouilles php.ini

Projets qui méritent qu'on s'y intéresse

- Webshare : Trop de fonctions j'ai pas eu le temps de tout parcourir
- PhpXplorer : Gestion interne des utilisateurs
- Ajaxplorer : Configuration modulaire, gestion interne des utilisateurs
- eXtplorer : projet intéressant rivalise avec le cheval gagnant, gestion interne et externe des utilisateurs, ...

L'application choisie : Net2FTP

Le choix ne fut pas aisé en effet beaucoup parmi ces applications méritent une attention particulière. Cepedant celle qui répond à peu près à notre cahier de charge est Net2FTP.

Présentation de Net2FTP

Net2FTP est un client FTP en ligne, qui vous permet d'effectuer toutes les manipulations sur un serveur distant que permet généralement un logiciel FTP installé : envoyer, télécharger, modifier, supprimer, renommer, zipper, dézipper, chmoder, déplacer des fichiers, etc

Avantage de Net2FTP

Net2FTP est une application web de service FTP très avancé offrant une interface et des fonctionalités proche d'une application de type desktop. Les fonctionnalités FTP fourni par Net2FTP sont : "chmodage", renommage, suppression, téléchargement, le téléversemment peut se faire par plusieurs méthode, à savoir : via un formulaire standard de téléversement, téléversement-décompression, ou via une applet java.

Net2FTP offre la possibilité d'éditer un fichier de type HTML ou PHP en ligne, avec prise en charge de la coloration syntaxique à l'aide d'application WYSIWYG (FCKeditor, TinyMCE)

Inconvenient de Net2FTP

A vrai dire je n'en ai pas trouvé, cependant on peut juste lui reprocher la non prise en compte du SFTP, mais c'est une autre histoire.

Présentation de VsFTP

vsFTPd est un serveur FTP sous license GPL pour les systèmes UNIX. “vs”, mis pour “VerySecure” rappelle quelle a été l’orientation de son auteur Chris Evans, lors de la conception de cette application.

- Privilégiant une architecture modulaire,
- s’appuyant sur des composants externes tels que PAM ou xinetd,

Ces caractéristiques remarquables font de vsFTPd un serveur FTP sécurisé, performant et stable, qui fait l’unanimité de sites tels que

- ftp ://ftp.debian.org/
- ftp ://ftp.openbsd.org/

(la liste est loin d’être exhaustive) et des grands noms du libre.

Serveur avec utilisateurs virtuels

Plusieurs méthodes de configuration s'offre à nous. Cette première configuration va nous permettre de mettre en évidence des concepts fondamentaux de vsFTPd.

- mise en place d'un serveur FTP en mode standalone(autonome),
- restriction de l'accès(grâce à PAM) aux utilisateurs virtuels d'une base de donnée.

Les utilisateurs virtuels ne disposent pas de véritables comptes sur la machine, mais d'un compte virtuel dans lequel ils sont "chrootés".

Serveur avec utilisateurs virtuels

Dans ce qui suit, nous allons être amené à créer différents fichiers de configuration. Pour plus de commodité, nous les rangerons dans /etc/vsftpd/ . Créons par conséquent ce répertoire :

```
# mkdir /etc/vsftpd
```

Si ce n'est pas déjà fait, on prend soin de sauvegarder la configuration par défaut de vsFTPd (on sait jamais..) :

```
# cp /etc/vsftpd.conf /etc/vsftpd.conf.default.bak
# cp /etc/pam.d/vsftpd /etc/pam.d/vsftpd.default.bak
```

Créons notre base de données d'utilisateurs virtuels

Le mécanisme d'authentification PAM utilise une base de données au format "Berkeley db", un format très répandu. Nous allons donc la créer grâce cet outil :

```
# apt-get install libdb3-util
```

Pour créer ce fameux fichier au format “db”, créons un fichier texte login.txt où apparîtront alternativement par paire de lignes, le nom d’utilisateur et le mot de passe, comme ceci :

login.txt

```
tom
foo
fred
bar
```

Les 2 utilisateurs sont donc ici ”’tom”’ et ”’fred”’ et leurs mots de passe respectifs ”’foo”’ et ”’bar”’.

Attention !

Veillez à ce que votre fichier se termine bien par un retour à la ligne et respectez scrupuleusement le format “login puis passwd” sans ajouter le moindre commentaire !

puis convertissons-le au format “db” :

```
# db3_load -T -t hash -f login.txt /etc/vsftpd/login.db
```

voilà, on dispose maintenant d’une base de données au format “db” : /etc/vsftpd/login.db

```
# chmod 600 /etc/vsftpd/login.db
```

Créons un fichier PAM

Créons maintenant un fichier vsftpd.pam qui dira à PAM d'utiliser notre base de données pour authentifier les utilisateurs :

```
auth required /lib/security/pam_userdb.so db=/etc/vsftpd/login
required /lib/security/pam_userdb.so db=/etc/vsftpd/login
```

Copions finalement ce fichier dans le dossier de configuration de PAM :

```
# cp vsftpd.pam /etc/pam.d/vsftpd
```

Ca y est, PAM dispose désormais de tous les éléments nécessaires pour assurer l'authentification des utilisateurs virtuels.

Un méta-utilisateur virtuel : l'utilisateur système

Tous les utilisateurs virtuels de notre base de données(tom, fred. . .) vont en réalité être représenté par UN même utilisateur(réel cette fois ci) système : l'utilisateur virtual(que l'on aurait très bien pu appeler autrement cela dit).
Ajoutons cet utilisateur à notre système, en définissant son groupe primaire à ftp(groupe que l'on va créer) et en le liant à son futur home /home/virtual/, qu'il ne restera alors plus qu'à créer :

```
# groupadd ftp \
# useradd -g ftp -d /home/ftp/virtual/ virtual
```

- Créons maintenant physiquement le répertoire /home/virtual/ en prenant soin de lui procurer des permissions correctes :
- le répertoire ne doit **pas lui appartenir** !
- il ne doit **pas pouvoir y écrire** !(pour cet usage, on créera, un peu plus tard, un sous répertoire dédié : /home/virtual/upload/)

```
# mkdir -p /home/ftp/virtual
# chown root.ftp /home/virtual/
# chmod 2750 /home/virtual/
```

vsftpd.conf

voir fichier annexe vsftp.conf

Copions-le dans /etc/ :

```
# cp vsftpd.conf /etc/
```

Relancement du serveur

```
# /etc/init.d/vsftpd stop
# /etc/init.d/vsftpd start
```

Test

```
$ ftp localhost 21
Connected to localhost.localdomain.
220 (vsFTPd 2.0.3)
Name (localhost:to_): tom
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Transfer done (but failed to open directory).
ftp> size hosts
213 147
ftp> quit
221 Goodbye.
```

On remarque que la commande ”ls” a échoué : **failed to open directory**. Ceci est tout à fait normal, car les utilisateurs virtuels ont en réalité les même permissions que les utilisateurs anonymes, pour lesquels vsFTPd applique une politique très restrictive.
En effet, par défaut le serveur n’autorise que la lecture des répertoires dont les permissions sont définies sur *world_readable* (lisible par le reste du monde). Or ce n’est pas le cas du répertoire /home/virtual/ qui a été ”chmodé” ainsi : 2750.

On pourrait remédier à ce problème en contrant cette politique dans /etc/vsftpd.conf par un anon_world_readable_only=NO, mais il serait beaucoup plus judicieux de pouvoir affiner les permissions de chacun des utilisateurs virtuels. . .