Mise en place d'un serveur FTP sous CentOS 6.2

Nicolau Fabien, Emanuele Prestifilippo

Le protocole *FTP* (File Transfert Protocol, en français protocole de transfert de fichiers) permet d'envoyer ou récupérer des fichiers sur un réseau TCP/IP en utilisant le modèle client-serveur.

L'un des avantages du protocole FTP est qu'il permet facilement, avec les serveurs les plus courants, d'utiliser des « utilisateurs virtuels ». Si vous avez 50 clients, inutile de créer un compte système pour chacun, les comptes seront tous stockés dans une base gérée par le serveur et un seul compte système sera utilisé pour les droits. Ces utilisateurs virtuels sont généralement dits « mappés » sur le compte local.

Dans ce tutoriel, nous expliquerons comment mettre en place le serveur vsftpd en mode standalone, avec une gestion des utilisateurs virtuels. Pour l'authentification, nous verrons deux des nombreuses possibilités offertes par PAM :

- stockage dans une base de données au format berkeley,
- stockage dans une base de données MySQL.

Présentation de vsftpd

vsftpd (Very Secure FTP Daemon) est un serveur FTP pour les systèmes UNIX, y compris Linux. Les principales qualités qui lui sont reconnues sont sa légèreté, sa configuration simple mais puissante et sa sécurité. Il supporte le chiffrement au travers de SSL intégré et gère le système d'identification des utilisateurs via *PAM* (Pluggable Authentication Modules), permettant ainsi d'utiliser tous les modes d'authentification que ce dernier propose.

vsftpd est capable de s'exécuter dans deux modes : *xinetd* ou *standalone*. Le premier sera à privilégier si les connexions clientes sont rares, le second si les connexions sont plus fréquentes.

Installation du serveur

vsftpd est disponible dans les dépôts de CentOS (les serveurs centralisés contenant des programmes) et s'installe simplement avec le groupe « FTP Server », grâce à l'utilitaire YUM.

Ouvrez un terminal et exécutez la commande suivante en tant que root :

```
# yum groupinstall "FTP Server"
```

L'installation terminée, si vous souhaitez que le service vsftpd soit lancé automatiquement au démarrage, il suffit d'utiliser la commande `chkconfig`, toujours en root :

```
# chkconfig vsftpd on
```

Lui passer « off » en paramètre permettra de désactiver le lancement automatique. Il est aussi possible de lancer, arrêter ou redémarrer le serveur avec les commandes suivantes :

```
# service vsftpd start
# service vsftpd restart
# service vsftpd stop
```

Configuration du serveur

La configuration du serveur vsftpd se fait via le fichier */etc/vsftpd/vsftpd.conf*. Toutes les clés de configuration existantes sont commentées via la page de man de vsftpd ou en-

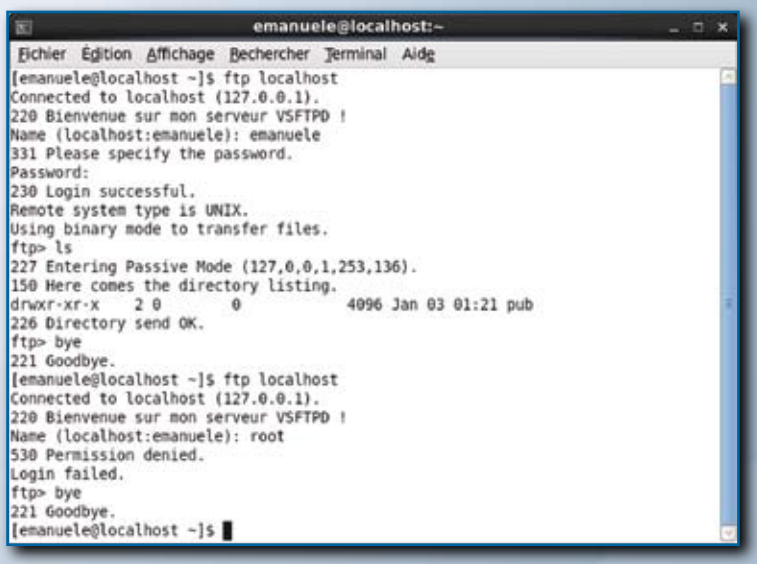

Figure 1. *Le premier essai réussi avec un utilisateur système*

core, à cette adresse : *http://vsftpd.beasts.org/vsftpd_conf.html* La première chose à faire est de créer une sauvegarde du fichier de configuration (nous utilisons ici la commande `mv` car nous partirons d'un fichier de configuration vide ensuite) :

```
$ su
# cd /etc/vsftpd
# mv vsftpd.conf vsftpd.conf.original
```

Profitons-en pour créer le répertoire qui contiendra les fichiers décrivant les droits spécifiques de chaque utilisateur :

```
# mkdir vsftpd_user_conf
```

Dans notre configuration souhaitée, nous autoriserons seulement les utilisateurs virtuels à se connecter et non pas les utilisateurs locaux de votre ordinateur (donc pas besoin du fichier *ftpusers*), nous utiliserons donc le fichier *user_list* contenant les identifiants qui seront refusés (sans même demander le mot de passe). Les utilisateurs virtuels seront mappés sur le compte système FTP et leurs fichiers de configuration seront stockés dans le dossier */etc/vsftpd/vsftpd_user_conf*. Enfin, notre serveur tournera en mode « standalone ». Voici le moment d'éditer le fichier de configuration avec votre éditeur de texte habituel, ce sera VI dans notre exemple :

```
# vi /etc/vsftpd/vsftpd.conf
```

Puisque nous avons précédemment renommé l'original, le fichier *vsftpd.conf* sera vide. Les principales valeurs à écrire sont présentées dans le Script 1, chacune d'elles est commentée.

Une fois le fichier rempli, enregistrez-le, limitez les droits le concernant, puis redémarrez le service vsftpd :

```
# chmod 600 /etc/vsftpd/vsftpd.conf
# service vsftpd restart
```

Il est déjà temps de faire un premier essai, grâce au client ftp en ligne de commande ; s'il n'est pas encore installé, vous pouvez rapidement l'obtenir en installant le paquet *ftp* avec YUM. Il suffit de passer en paramètre l'adresse ou le nom d'hôte auquel se connecter ; ici, ce sera localhost, c'est-à-dire l'ordinateur local. Un premier essai est fait avec notre utilisateur système (emanuele) et notre mot de passe. Grâce à la commande `ls`, vous devriez voir le contenu du dossier */var/ftp/*. Ensuite, une connexion est tentée avec l'utilisateur root. Celui-ci faisant partie du fichier */etc/vsftpd/user_list*, l'accès lui est refusé. Notez que le mot de passe n'est même pas demandé, ce qui évite de le faire passer inutilement en clair sur le réseau ! (Figure 1).

Script 1. *Fichier de configuration vsftpd.conf*

```
# Port d'écoute
listen_port=21
# Bannière de bienvenue
ftpd_banner=Bienvenue sur mon serveur VSFTPD !
# Fichier de configuration de PAM
pam_service_name=vsftpd
# Mode "standalone"
listen=YES
# Pas de connexion anonyme
anonymous_enable=NO
# Les utilisateurs système sont autorisés
local_enable=YES
# Fichier des utilisateurs
userlist_file=/etc/vsftpd/user_list
# Chargement de la liste userlist_file
userlist_enable=YES
# Il est indiqué ici que cette liste est celle des identifiants refusés, par ceux autorisés
userlist_deny=YES
# Un utilisateur virtuel pourra télécharger un fichier même s'il n'est pas lisible par tous
anon_world_readable_only=NO
# Refus des commandes influant sur le système de fichier (STOR, DELE, RNFR, RNTO, MKD, RMD, APPE and SITE)
write_enable=NO
# Refus des droits d'écriture pour les anonymes (et donc utilisateurs virtuels) par défaut
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
# Activation des utilisateurs virtuels et mappage sur le compte local ftp
guest_enable=YES
guest_username=ftp
# chroot des utilisateurs
chroot_local_user=YES
# Nombre maximum de connexions simultanées
max_clients=50
# Nombre maximum de connexions venant de la même IP
max_per_ip=4
# Dossier de configuration spécifique des utilisateurs
user_config_dir=/etc/vsftpd/vsftpd_user_conf
```

Utilisateurs virtuels

Le serveur FTP étant maintenant prêt, nous pouvons enregistrer des utilisateurs. Ceux-ci sont dits « virtuels » car ce ne sont pas des utilisateurs disposant d'un compte système, ils sont uniquement gérés par le serveur vsftpd. Les accès étant

gérés via PAM, nous pouvons utiliser n'importe quel mode d'authentification proposé. Nous détaillerons deux d'entre eux : berkeley et MySQL.

Préparation d'une base berkeley

Construire une base de données au format berkeley est plutôt simple. Il suffit de partir d'un fichier texte, qui contiendra les identifiants et mot de passe. Un compte se compose donc d'une première ligne avec l'identifiant et d'une seconde en dessous avec le mot de passe. Les comptes se mettent les uns en dessous des autres. Il faut enfin ajouter un dernier retour à la ligne. Voici notre fichier créé donc dans */tmp* et appelé *login.txt* :

```
user1
pass1
user2
pass2
```

Il ne reste plus qu'à transformer ce fichier en une base de données berkeley, grâce à la commande `db_load` Si celle-ci n'est pas disponible sur votre CentOS 6.2, installez le paquet *db4-utils* la fournissant. Lancez donc la commande qui suit :

```
# db_load -T -t hash -f /tmp/ ↵
   login.txt  /etc/vsftpd/  ↵
   login.db
```

Puis réduisez les droits sur cette base nouvellement créée :

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

Préparation d'une base MySQL

Avec ce type de stockage, nous créerons une table contenant les utilisateurs du serveur et une pour l'enregistrement des connexions (possibilité offerte par le module pam_mysql). Les champs des tables sont libres, ils seront ensuite à préciser dans la configuration de PAM. Profitons-en donc pour ajouter un champ « active » à notre enregistrement utilisateur, ce qui permettra de suspendre l'accès à un utilisateur sans pour autant le supprimer. Ici, nous supposons que vous avez un serveur MySQL fonctionnel. Créez d'abord une base de données nommée *vsftpd*, puis exécutez le Script 2 afin de créer les deux tables. Ajoutons nos deux utilisateurs de tests (Script 3).
Deux choses sont à remarquer ici :

- les mots de passe sont chiffrés grâce à la fonction md5,
- user2 est créé, mais non activé (active vaut 0).

Configuration de PAM

Maintenant que notre base d'utilisateurs est prête, il nous reste à mettre en place le fichier PAM pour que vsftpd puisse l'utiliser.

Attention ! Dans les Scripts 4 et 5, si vous avez installé CentOS 64 bits, remplacez /lib/ avec /lib64/.

Vous avez utilisé une base berkeley

Ouvrez en root avec votre éditeur de texte habituel le fichier */etc/pam.d/vsftpd,* effacez toutes les lignes existantes et placez-y le contenu présenté dans le Script 4.

Les deux premières lignes du fichier permettent d'autoriser l'accès aux utilisateurs système, les deux suivantes indiquent que les comptes des utilisateurs virtuels utilisés pour l'authentification se trouvent dans notre base berkeley (le fichier est bien login, sans extension).

Script 2. *Script créant la structure de la base de données vsftpd*

```
CREATE TABLE  `vsftpd`.`users` (
`id_user` int(11) NOT NULL auto_increment,
`login` varchar(50) NOT NULL,
`password` varchar(100) NOT NULL,
`active` int(1) NOT NULL,
PRIMARY KEY  (`id_user`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1

CREATE TABLE  `vsftpd`.`log` (
`id_log` int(11) NOT NULL auto_increment,
`login` varchar(50) NOT NULL,
`message` varchar(200) NOT NULL,
`pid` varchar(10) NOT NULL,
`host` varchar(30) NOT NULL,
`time` datetime default NULL,
PRIMARY KEY  (`id_log`)
) ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

Script 3. *Insertion de deux utilisateurs*

```
$ mysql -u <votre_user_mysql> --password=<votre_password_ ↵
   mysql> -e "INSERT INTO vsftpd.users (login,password,active) ↵
   VALUES('user1',md5('pass1'),1)"
$ mysql -u <votre_user_mysql> --password=<votre_password_ ↵
   mysql> -e "INSERT INTO vsftpd.users (login,password,active) ↵
   VALUES('user2',md5('pass2'),0)"
```

Script 4. *Configuration de PAM pour une base berkeley*

```
#%PAM-1.0
auth sufficient pam_unix.so
account sufficient pam_unix.so
auth     required     /lib/security/pam_userdb.so db=/etc/ ↵
   vsftpd/login
account   required    /lib/security/pam_userdb.so db=/etc/ ↵
   vsftpd/login
```

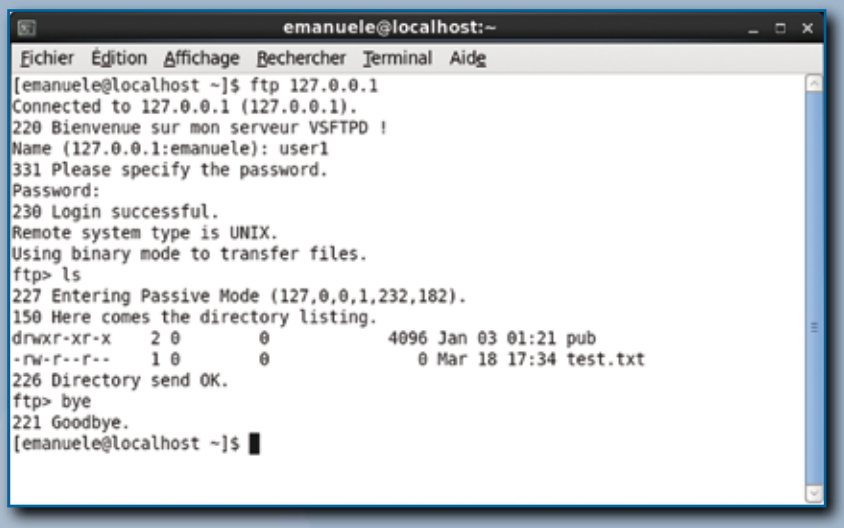

Figure 2. *La connexion avec un utilisateur virtuel*

Vous avez utilisé une base MySQL

La première chose à faire est d'installer le paquet *pam_mysql* contenant le module MySQL pour PAM. L'opération terminée, ouvrez en root avec votre éditeur de texte habituel le fichier */etc/pam.d/vsftpd*, effacez toutes les lignes existantes et placez-y le contenu visible dans le Script 5.

Les deux premières lignes du fichier permettent d'autoriser l'accès aux utilisateurs système, alors que les deux suivantes indiquent les différents champs de votre base d'utilisateurs stockés sur un serveur MySQL afin d'y accéder et d'enregistrer les connexions. Notez que *crypt=3* est utilisé car nos mots de passe sont chiffrés avec la fonction md5(). La clause *where* permet de filtrer les utilisateurs pour lesquels *active* ne vaut pas 1.

Tous les paramètres mis après *where* ne sont à utiliser que si vous vous servez du log dans la base MySQL. Nous avons utilisé 127.0.0.1 pour l'hôte, mais il faut évidemment l'adapter en mettant l'adresse IP de votre serveur MySQL. Voici en détails les paramètres :

- *verbose* : permet d'afficher plus d'informations, seulement utile lors du débogage,
- *user* : utilisateur MySQL,
- *passwd* : mot de passe MysSQL,
- *host* : hôte du serveur MySQL,
- *db* : nom de la base de données,
- *table* : nom de la table pour les utilisateurs,
- *usercolumn* : nom de la colonne contenant les identifiants,
- *passwdcolumn* : nom de la colonne contenant les mots de passe,
- *crypt* : type de chiffrement, 3 indique que l'on utilise du md5,
- *where* : close supplémentaire à rajouter dans la requête, ici nous nous en servons pour utiliser le paramètre *active*,
- *sqllog* : *yes* indique que nous voulons logger dans une table les événements,
- *logtable* : nom de la table contenant les logs,
- *logmsgcolumn* : nom de la colonne contenant le message de log,
- *logusercolumn* : nom de la colonne contenant l'identifiant pour le log,
- *logpidcolumn* : nom de la colonne contenant le pid pour le log,
- *loghostcolumn* : nom de la colonne contenant l'hôte pour le log,
- *logtimecolumn* : nom de la colonne contenant la date et l'heure pour le log.

Nous pouvons maintenant redémarrer le serveur :

```
$ su -lc "service vsftpd ↵
  restart"
```

Vérifier la configuration de PAM

Quelle que soit la configuration de PAM que vous avez utilisée, essayez à nouveau de vous connecter en utilisant cette fois-ci l'identifiant *user1* et le mot de passe *pass1*, la connexion doit être acceptée (Figure 2).

La connexion fonctionne mais il vous est impossible pour le moment de télécharger un fichier. La dernière étape consistera donc à attribuer des droits d'écriture, seulement pour certains utilisateurs, grâce aux fichiers placés dans */etc/vsftpd/vsftpd_user_conf*.

Configuration des droits par utilisateur

Pour attribuer des droits spécifiques à un utilisateur, il suffit de lui créer un fichier portant son identifiant dans */etc/vsftpd/vsftpd_user_conf*. Nous entrons ensuite dans ce fichier les mêmes clés que dans le fichier de configuration principal, avec des valeurs différentes bien sûr. Lorsque l'utilisateur se connectera, les clés de configuration présentent dans son fichier seront alors prioritaires par rapport à celles du fichier principal. Il est ainsi possible de modifier

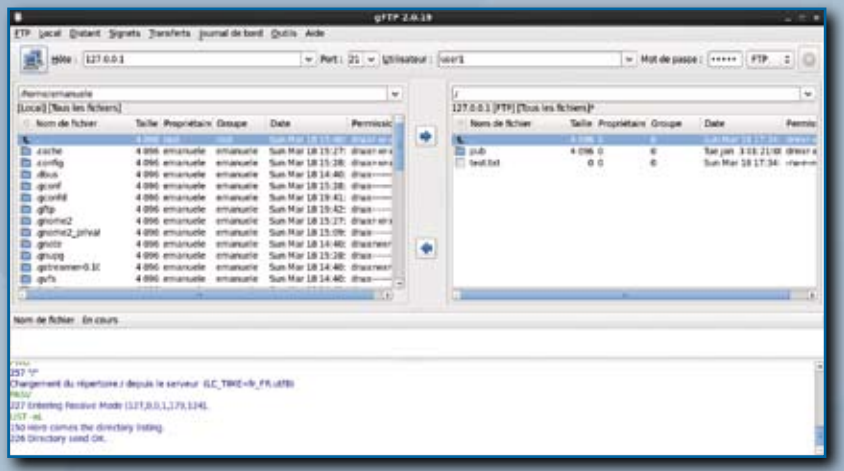

Figure 3. *La connexion à notre serveur vsftpd via gFTP*

tous les paramètres pour un utilisateur en particulier. C'est aussi une des forces de vsftpd. Prenons un exemple où nous allons changer le répertoire de l'utilisateur user1, afin qu'il lui soit propre, et donnons-lui ensuite les droits d'écriture. Notez que pour que le tout fonctionne bien, le nouveau répertoire d'accueil de user1 doit appartenir à l'utilisateur système FTP, utilisateur sur lequel user1 est mappé (Script 6).

Il est possible de mettre un chemin absolu pour la clé *local_root* ; cependant, il faut s'assurer des droits de l'utilisateur ftp sur le dossier indiqué.

Faites à nouveau un test, avec l'utilisateur *user1*. Vous voilà dans un autre dossier, dans lequel vous pouvez envoyer des fichiers.

Erreurs courantes

Voici quelques règles à mettre en place si vous rencontrez les erreurs suivantes, notamment si SELinux est activé sur votre système.

500 OOPS: reading non-root config file

Cette erreur vient de ce que l'utilisateur n'a pas de fichier spécifique dans */etc/vsftpd/vsftpd_user_conf* (même vide). Il semble que ce soit un bogue. Si vous avez ce souci, créez un fichier pour cet utilisateur. Nous verrons ensuite que nous pouvons faire cela automatiquement.

500 OOPS: cannot change directory:

Dans ce cas, entrez : `$ su -lc "setsebool -P ftp_home_dir 1"`

500 OOPS: vsftpd: refusing to run with writable anonymous root

Il faut enlever le droit d'écriture sur le répertoire root : `# chmod u-w /le/local/root` Ensuite, il faut s'assurer que le *local_root* appartient bien au user et groupe ftp.

553 Could not create file

Si cette erreur intervient et qu'une erreur SELinux apparaît, lancez alors ces commandes :

```
setsebool -P allow_ftpd_anon_write=1
chcon -R -t public_content_rw_t <local_root de l'utilisateur>
```

Connexions graphiques

Dans les exemples de ce tutoriel, nous avons utilisé la commande `ftp` pour les essais. Il vous est évidemment possible d'utiliser n'importe quel client graphique pour vous connecter à votre serveur, comme par exemple, le très répandu gFTP (Figure 3) ou le multi-plate-forme FileZilla.

Conclusion

Voilà votre serveur FTP en place. Il vous est maintenant possible de distribuer des comptes et d'appliquer finement des droits à chacun d'eux comme, par exemple, avoir un compte dédié à la mise à jour de votre site Internet. Parfaitement intégré à CentOS, ce serveur est facilement administrable. Si vous souhaitez interdire des utilisateurs système, il vous suffit d'entrer leurs identifiants dans le fichier *user_list*. Enfin, si les bases berkeley ou MySQL ne vous conviennent pas, vous pouvez aller plus loin en choisissant, parmi les différentes possibilités de PAM, une solution de stockage qui vous convient mieux et ainsi profiter un peu plus de la flexibilité offerte par vsftpd !

En plus de vsftpd, il existe d'autres serveurs utilisant le protocole FTP pour Linux, parmi lesquels ProFTPd et Pure-FTPd. Si la configuration d'un serveur FTP en utilisant la ligne de commande ne vous convient pas, utilisez une des nombreuses interfaces graphiques disponibles :

- System-Config-vsftpd et KVsftpdManager pour vsftpd,
- GProftpd et jProftpd pour ProFTPd,
- PureAdmin et KcmPureftpd pour Pure-FTPd.

Script 5. *Configuration de PAM pour une base MySQL*

```
#%PAM-1.0
auth sufficient pam_unix.so
account sufficient pam_unix.so
auth required /lib/security/pam_mysql.so verbose=0
   user=<votre_user_mysql> passwd=<votre_pass_mysql>
   host=127.0.0.1 db=vsftpd table=users usercolumn=login
   passwdcolumn=password crypt=3 where=users.active=1
   sqllog=yes logtable=log logmsgcolumn=message
   logusercolumn=login logpidcolumn=pid loghostcolumn=host
   logtimecolumn=time
account required /lib/security/pam_mysql.so verbose=0
   user=<votre_user_mysql> passwd=<votre_pass_mysql>
   host=127.0.0.1 db=vsftpd table=users usercolumn=login
   passwdcolumn=password crypt=3 where=users.active=1
   sqllog=yes logtable=log logmsgcolumn=message
   logusercolumn=login logpidcolumn=pid loghostcolumn=host
   logtimecolumn=time
```

Script 6. *Personnalisation des droits pour l'utilisateur user1*

```
# mkdir /var/ftp/user1
# chown ftp:ftp /var/ftp/user1
# echo "local_root=user1" > /etc/vsftpd/vsftpd_user_conf/user1
# echo "write_enable=YES" >> /etc/vsftpd/vsftpd_user_conf/user1
# echo "anon_upload_enable=YES" >> /etc/vsftpd/vsftpd_user_conf/user1
# echo "anon_mkdir_write_enable=YES" >> /etc/vsftpd/vsftpd_user_conf/user1
# echo "anon_other_write_enable=YES" >> /etc/vsftpd/vsftpd_user_conf/user1
```