Le serveur vsftpd

- Objet : Installation et configuration du serveur ftp vsftpd.
- Niveau requis :
  avisé
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Introduction

Ce wiki a pour but de détailler les différentes étapes pour mettre en place un serveur de partage de fichier FTP, en utilisant vsftpd (Very Secure FTP Daemon).

Un serveur vsftpd permet comme un serveur FTP de déterminer un le lieu sur votre serveur depuis lequel il est possible de transférer des fichiers depuis son ordinateur personnel sur lequel est installé un client ftp.

Il y a plusieurs façon de créer des espaces ftp:

- En utilisant ou en créant un utilisateur linux dont le répertoire personnel sera l'espace ftp.

- En utilisant une base de donnée de type Berkeley (mode utilisateurs viruels)

Nous allons d'abord mettre en œuvre la solution un utilisateur linux pour chaque espace ftp car c'est certainement la solution la plus simple. Elle ne convient pas pour un grand nombre d'espaces FTP.

La faille majeur d'un serveur vsftpd est que tout passe en clair. Il faudra donc installer un certificat ssl qui permet le chiffrement de la connexion au serveur.

Installation et utilisations

Installation et configuration basique de vsftpd

Installation de vsftpd

```
apt-get install vsftpd
```

Mon serveur vsftp est-il en fonction?

Pour le savoir deux méthodes :

```
netstat -a | grep ftp
```

```
tcp    0      0 *:ftp     *:*      LISTEN
```

→ C'est ok !

```
/etc/init.d/vsftpd status
```

```
vsftpd is running
```

→ C'est ok aussi !

Configuration basique

Attention il faut être derrière un routeur c'est-à-dire en local.

- On fait quelques menues modification au fichier de configuration “/etc/vsftpd.conf”

```
vim /etc/vsftpd.conf
```

```
listen=YES
anonymous_enable=NO
local_enable=YES
ftpd_banner=Bienvenue !
```

Première connexion avec le compte de l'utilisateur principal sur le serveur

Depuis le client une autre machine virtuelle par exemple que celle sur laquelle on a installé vsftpd on tape simplement :

```
ftp 192.168.0.10
```

> 192.168.0.10 : étant l'IP du serveur vsftpd

```
Connected to 192.168.0.10.
220 Bienvenue !
Name (192.168.0.10:hypathie): hypathie
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
221 Goodbye.
```

> où `Password` : est le mot de passe du compte de l'utilisateur “hypathie” sur le système Linux.

Voyons maintenant le détail des options possibles et l'installation d'un certificat ssl pour chiffrer le mot de passe de l'utilisateur.

Configuration de vsftpd avec ssl

Installation de open-ssl

```
apt-get install openssl
```

Création du certificat

Peut importe d'où la commande suivante va être lancé on déplacera le fichier qu'elle va créer.

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout vsftpd.pem -out vsftpd.pem
```

Répondre comme bon vous semblera à toutes les questions sauf :
`Common Name` : mettre l'IP du serveur ou le nom de domaine si un DNS en local est installé.

```
mv vsftpd.pem /etc/ssl/certs
```

- Pour sécuriser tout ça :

```
chown root:root /etc/ssl/certs/vsftpd.pem
```

```
chmod 600 /etc/ssl/certs/vsftpd.pem
```

Et voilà !
Il faut maintenant éditer à nouveau le fichier /etc/vsftpd.conf

Détail des options par défaut de vsftpd avec ssl

Voici les options utilisées et commentées à partir de man vsftpd, histoire de comprendre ce qu'on fait. C'est sûr, c'est long ...

Pour les options qui ne sont pas utilisées dans cette configuration, voir le man ou en ligne ici : https://security.appspot.com/vsftpd/vsftpd_conf.html

- Éditons à nouveau /etc/vsftpd.conf :

```
vim /etc/vsftpd.conf
```

```
# Pour mettre en mode standalone : ce mode permet au service FTP d'avoir
# son propre démon au lieu de fonctionner sous le démon des autres
# service du sytème, le démon xinetd
#
listen=YES
```

```
#
# On n'utilisera pas ipv6
#listen_ipv6=YES
#
# On ne veut pas de connexions en mode anonymous qui permet à quiconque
# de se connecter au serveur
#
anonymous_enable=NO
#
# On veut que les utilisateurs locaux puissent se connecter
local_enable=YES
#
# On veut que les utilisateurs puissent remonter des fichiers sur le serveur
write_enable=YES
#
# Umask par défaut pour les utilisateurs locaux est 077
# On peut changer cela en 022, si les utilisateurs s'attendent à ce que
# (022 soit utilisé par la plupart des autres serveurs FTP).
local_umask=022
#
#
# On interdit l’upload anonyme
anon_upload_enable=NO
#
# On interdit l'upload anonyme
anon_mkdir_write_enable=NO
#
# Lorsque cette option est activée, un message apparaît chaque fois
# qu'un utilisateur ouvre un répertoire avec un fichier message.
# Ce message se trouve dans le répertoire qui est ouvert.
# Le nom de ce fichier est spécifié dans la directive
# message_file et part défaut prend la valeur .message.
# man page précise que par défaut la valeur est NO mais qu'une simple
# configuration peut mettre la valeur YES. Cela varie selon les
distributions.
dirmessage_enable=NO
#
# Les heures d’enregistrement des fichiers seront affichées à l’heure
locale.
use_localtime=YES
#
# Pour que les actions des utilisateurs soient loggées.
xferlog_enable=YES
#
# On vérifie que la demande de port provienne forcément du port 20
# de la machine cliente.
connect_from_port_20=YES
#
# Lorsque cette option est activée, tous les fichiers téléléchargés
# vers les serveurs par des utilisateurs anonymes deviennent
# la propriété de l'utilisateur spécifié dans la directive chown_username.
```

```
chown_uploads=yes
# Spécifie la propriété de fichiers téléchargés anonymement
# vers le serveur si la directive chown_uploads est activée.
chown_username=hypathie
#
# Les logs seront enregistres dans le fichier /var/log/vsftpd.log
xferlog_file=/var/log/vsftpd.log
#
#Lorsque l'option "xferlog_std_format=YES"  est activée de concert avec
# xferlog_enable, alors un seul un journal de transfert de fichiers
# compatible avec wu-ftpd est enregistré dans le fichier spécifié
# dans la directive xferlog_file (par défaut /var/log/xferlog).
# Il est important de noter ici que ce fichier journalise seulement
# les transferts de fichiers et n'enregistre pas les connexions au serveur.
#
#Lorsque l'option "xferlog_std_format=NO", on peut ajouter les directives
# "log_ftp_protocol=YES" et "xferlog_enable=YES"
# ainsi toutes les commandes et réponses FTP seront journalisées.
# Cela est très utilise lors d'opérations de débogage.
xferlog_std_format=YES
#
# On declare les valeurs de timeout. Spécifie la durée maximale pouvant
# s'écouler entre des commandes depuis un client distant.
# Une fois cette durée écoulée, la connexion au client distant est fermée.
idle_session_timeout=600
#
# Spécifie la durée maximale exprimée en secondes,
# pendant laquelle les transferts de données peuvent s'arrêter.
data_connection_timeout=120
#
#On ajoute une indication sur la durée maximale exprimée en secondes,
# donnée à un client utilisant un mode actif pour répondre
# à une connexion de données.
connect_timeout=60
#On ajoute une indication sur la durée donnée à un client
# utilisant une connexion passive pour se connecter.
accept_timeout=60
# Il est recommandé que vous définissez sur votre système un utilisateur
# unique que le serveur FTP peut utiliser comme un utilisateur non
# privilégié et totalement isolé.
# Cela restreint complétement les privilèges d'exécution du processus vsftpd
# non privilégié (lancé au moment de l'attente d'authentification
# puis lors de l'automutilation) en lui dédiant un utilisateur (par défaut
# nobody). Mais il faut de créer cet utilisateur non-privilégié (par
# exemple vsftpd) et de le spécifier par la directive nopriv_user=vsftpd.
# Si on ne crée pas cet utilisateur plus personne peut être accepté.
# man page indique que par défaut la valeur est "Default: nobody".
#nopriv_user=ftpsecure
nopriv_user=nobody
#
# Par sécurité, on interdit la commande ABOR
```

```
# Elle permet de stopper un téléchargement asynchrone en cours.
# Elle est considérée comme complexe et inélégante, parce qu'async ABOR
# produit des effets différents en fonction des clients. On l'utilise quand
# on ne sait pas quel type de client est susceptible de se connecter.
async_abor_enable=NO
#
# Les transferts en ASCII sont souvent source de confusions.
ascii_upload_enable=NO
ascii_download_enable=NO
#
# On change la bannière pour reconnaître note serveur et c'est plus joli !
ftpd_banner=Bienvenue !
#
# Utile pour prévenir certaines attaques : Lorsque cette option est activée,
# tout utilisateur anonyme employant des mots de passe de messagerie
# spécifiés dans /etc/vsftpd.banned_emails se voit refuser l'accès au
serveur.
# Le nom du fichier référencé par cette directive peut être spécifié
# à l'aide de la directive banned_email_file
# On l'utilise avec l'autorisation des connexions anonymes, ce qu'on a
refusé.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
### Si on veut limiter les utilisateurs à leur repertoire
chroot_local_user=NO
#
# Mais il est à noter d'abord que :
# 1) Si on a choisi "chroot_local_user=YES"
# alors on doit mettre "chroot_list_enable=NO"
# sinon la liste_enable indique les utilisateurs qui ne sont pas chrootés.
#
# 2) Une mesure de sécurité ajoutée à vsftpd dans sa version 2.3.5, refuse
# de chrooter un client dans un répertoire pour lequel l'utilisateur possède
# les droits d'écriture. Du coup avec :"chroot_local_user=YES"
# un refus de connexion à lieu lorsque un utilisateur tente de se logger :
# "500 OOPS: vsftpd: refusing to run with writable root inside chroot()
#Login failed.
#
# Avec "chroot_local_user=YES" la méthode pour contourner cette sécurité:
#"chroot_local_user=YES"
#"allow_writeable_chroot=YES"
# Et ""chroot_list_enable=NO"
# ne fonctionne pas
#
### Il en va de même pour  l'option "chroot_list_enable=YES"
# (avec "chroot_local_user=NO") qui devrait permettre de fournir
# une liste des utilisateurs locaux dont le répertoire personnel peut
# placé dans un chroot () lors de la connexion.
# Par défaut, le fichier contenant cette liste devrait être
```

```
# /etc/vsftpd.chroot_list
# Mais cela ne fonctionne pas.
#chroot_list_file=/etc/vsftpd.chroot_list
#
## Désactive le listage récursif des répertoires par la commande "ls -R",
# afin d'éviter trop d'appels sur le système de fichier.
# Certain clients FTP comme "ncftp" ou "mirror" réclame l'option "-R"
# pour fonctionner.
# ls_recurse_enable=NO
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# Cette option doit être le nom d'un répertoire qui est vide.
# En outre, le répertoire ne doit pas être accessible en écriture
# par l'utilisateur ftp.
# Ce répertoire est utilisé comme un chroot sécurisé.
# Par exemple pour emprisonner le démon vsftpd :
# "secure_chroot_dir=/var/run/vsftpd".
# Mais tout va bien par défaut, à l'installation /var/run/vsftpd/empty
# possède les bons droits et il est vide.
secure_chroot_dir=/var/run/vsftpd/empty
#
# Spécifie le nom du service PAM pour vsftpd:
# Le PAM est système d'authentification utilisé en standard sous Linux.
# DOC en français : http://www-igm.univ-mlv.fr/~dr/XPOSE2003/augereau/2.html
pam_service_name=vsftpd
#
###mise en place du chiffrement ssl
#
#Remarque sur l'utilisation de ssl:
#Soit on met certificat et clé privée dans /etc/ssl/private/vstfpd.pem
#Soit on met le certificat (copie) dans
#  et dans ce cas, on ajoute la directive
#  "rsa_private_key_file=/fichier/clé/privée"
# Cette option spécifie l'emplacement de la clé privée RSA à
# utiliser pour les connexions cryptées SSL.
# emplacement par défaut : Default: (none)
ssl_enable=YES
validate_cert=NO
ssl_ciphers=HIGH
require_ssl_reuse=NO
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
# Emplacement du certificat RSA à utiliser pour les connections SSL.
```

```
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
#
pasv_address=192.168.0.10
```

C'était long ! En voici encore un peu..

Pour la signification des options concernant ssl :

ssl_enable: Si elle est activée, vsftpd utilise OpenSSL, et vsftpd soutiendra les connexions sécurisées via SSL.
Cela s'applique à la connexion de contrôle (y compris login) et les connexions de données.
Vous aurez besoin d'un client avec le support SSL.
REMARQUE !! Méfiez-vous de cette option.
vsftpd ne donne aucune garantie quant à la sécurité des bibliothèques OpenSSL. En activant cette option, vous déclarez que vous faites confiance à la sécurité de votre bibliothèque OpenSSL.

validate_cert : Si définie à “yes”, tous les certificats clients SSL reçus doivent validés. Certificats auto-signés ne constituent pas une validation. (Dans ce wiki ont a auto-signé, pour pouvoir chiffrer, mais l'utilisation est locale !)

ssl_ciphers : Cette option peut être utilisée pour sélectionner les chiffrements SSL vsftpd qui permettront des connexions SSL chiffrées.
Voir la page de man sur les chiffrement pour plus de détails.
Par défaut: DES-CBC3-SHA

require_ssl_reuse : Si définie à “yes”, toutes les connexions de données SSL dépendent de la session SSL. Mieux vaut la désactiver. Pour une discussion sur les conséquences, voir http://scarybeastsecurity.blogspot.com/2009/02/vsftpd-210-released.html (Ajouté à v2.1.0).
Par défaut: OUI

allow_anon_ssl : Ne s'applique que si “ssl_enable” est actif. Si la valeur est “YES”, les utilisateurs anonymes seront autorisés à utiliser des connexions SSL sécurisées.
Default: NO

force_local_data_ssl : Ne s'applique que si “ssl_enable” est activée. Si elle est activée, toutes les connexions non anonymes sont contraints d'utiliser une connexion SSL sécurisée pour envoyer et recevoir des données sur les connexions de données.
Par défaut: OUI

force_local_logins_ssl : Ne s'applique que si “ssl_enable” est activée. Si elle est activée, toutes les connexions non anonymes sont contraints d'utiliser un mot de passe.
Par défaut: OUI

ssl_tlsv1 : Ne s'applique que si “ssl_enable” est activée. Si elle est activée, cette option permettra les *TLS* connexions de protocole v1. Les connexions TLS de sont préférables.

Par défaut: OUI

ssl_sslv2 : Ne s'applique que si “ssl_enable” est activée. Si elle est activée, cette option permettra les connexions de protocole SSL v2. Les connexions TLS de sont très bonnes.
Par défaut: NON

ssl_sslv3 : Ne s'applique que si ssl_enable est activée. Si elle est activée, cette option permettra les connexions de protocole SSL v3. Les connexions TLS de sont très bonnes.
Par défaut: NON

pasv_address : Utilisez cette option pour remplacer l'adresse IP que vsftpd affichera en réponse de la commande PASV. Fournir une adresse IP numérique, à moins pasv_addr_resolve soit activée, dans ce cas, vous pouvez fournir un nom d'hôte qui résolu par un DNS mis en place lors du démarrage.
Valeur par défaut: (aucune)

Connexion au serveur vsftpd avec ssl

Installation sur le client de ftp-ssl

```
apt-get install ftp-ssl
```

Cela va désinstaller ftp client !
Donc il faudra forcement passer par ftp-ssl, et on ne pourra plus se connecter au serveur comme dans la première partie du wiki.
Mais ça veut mieux !

Connexion avec ftp-ssl

```
ftp-ssl 192.168.0.10
```

> où `192.168.0.10` est à remplacer par l'IP de votre serveur vsftpd

```
Connected to 192.168.0.10.
220 Bienvenue !
Name (192.168.0.10:hypathie): hypathie
234 Proceed with negotiation.
[SSL Cipher AES256-GCM-SHA384]
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
ftp> exit
221 Goodbye.
```

> où `Password` : est le mot de passe du compte de l'utilisateur “hypathie” sur le système Linux.

Et voilà !

Apprendre à utiliser les commandes FTP

Utilisation