Activité : Serveur FTP (Transfert de fichiers)

Thierry Vaira <tvaira@free.fr>

2013-2016 (rev. 1)

Table des matières

Serveur FTP (Transfert de fichiers)

Introduction

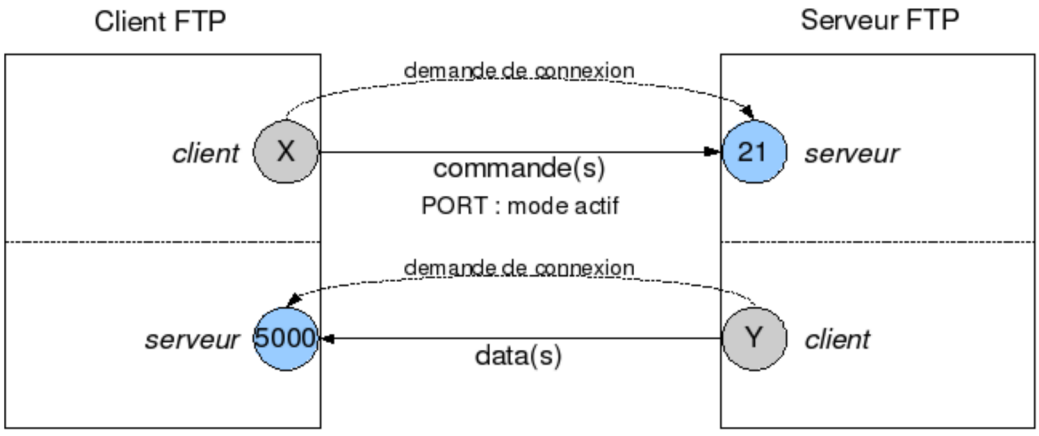
Le protocole **FTP** (*File Transfer Protocol*) est un protocole de transfert de fichier (RFC959) de la couche Application. Le protocole FTP s'utilise de façon standard sur le port 21 du serveur en mode TCP. Par contre, FTP ne fonctionne que sur du TCP. Il existe un protocole **TFTP** (*Trivial FTP*) qui est lui basé sur UDP.

Lors d'une connexion FTP, deux canaux de transmission sont ouverts :

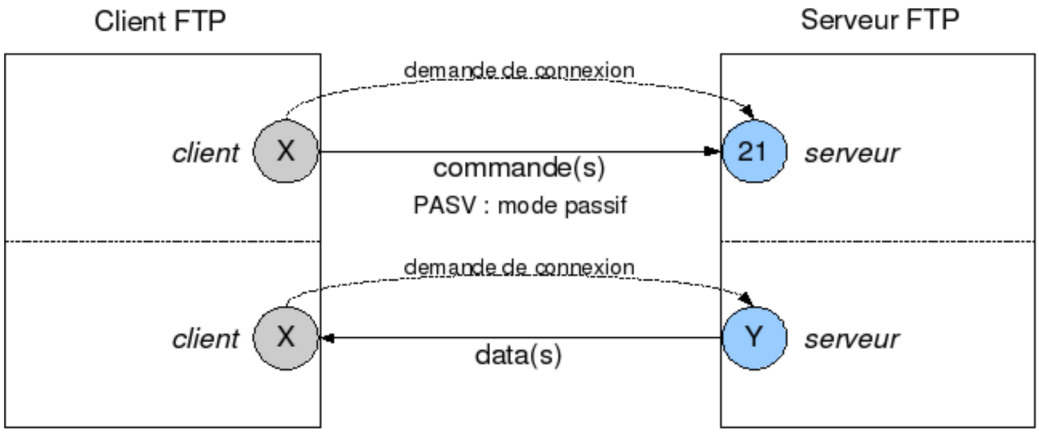
- Un canal pour l'échange des commandes (canal de contrôle) : USER, PASS, LIST, RETR, STOR, ...
- Un canal pour l'échange des données

L'échange de données fonctionnant suivant le modèle client/serveur, il existe donc deux possibilités : le mode actif et le mode passif (qui est le plus utilisé).

Dans le mode actif, le client FTP (en utilisant la commande `PORT`) détermine le port d'écoute et joue le rôle de serveur pour le canal de données :



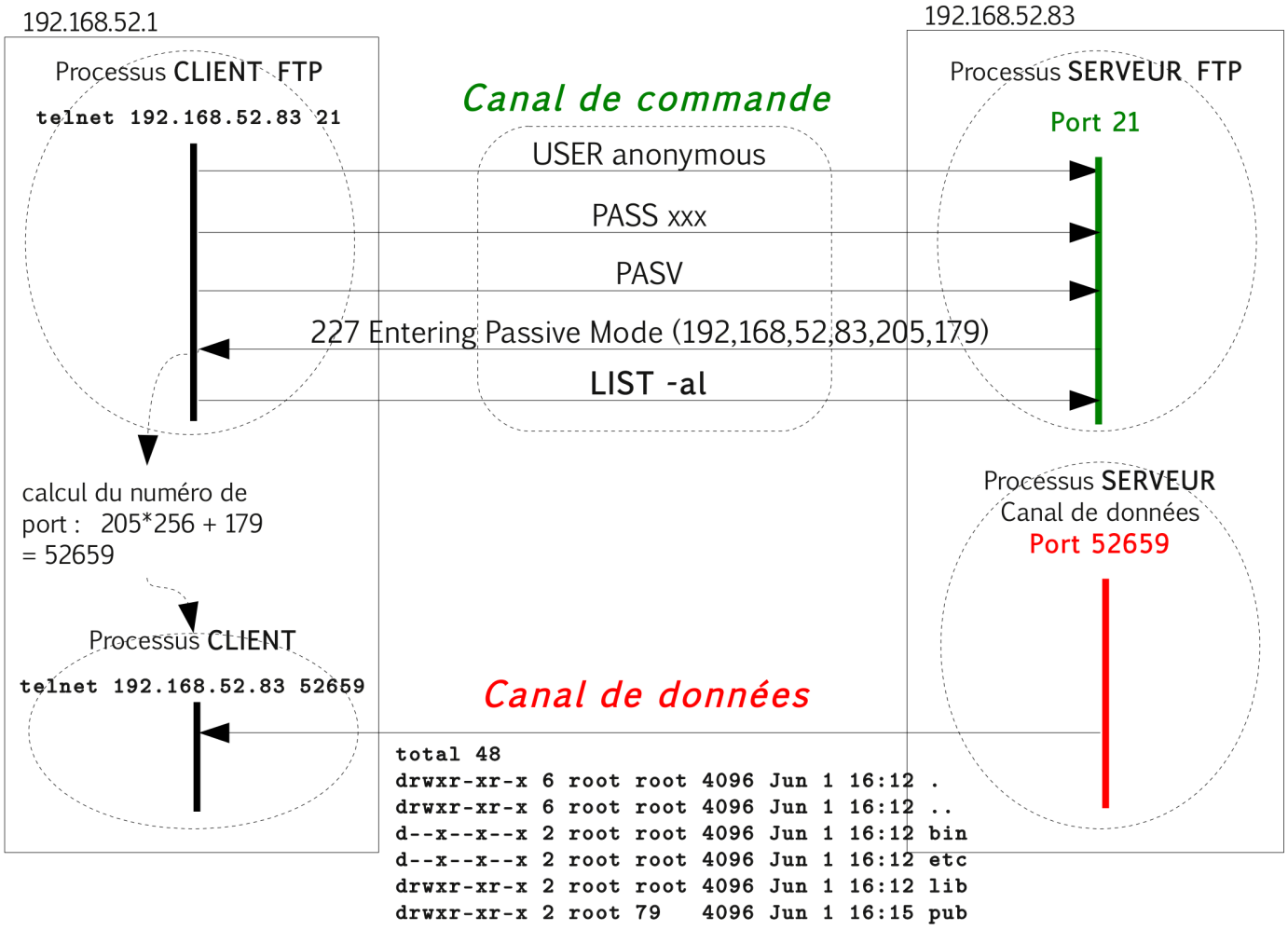
Dans le mode passif, le client FTP (en utilisant la commande `PASV`) choisit le mode passif et c'est le serveur FTP qui détermine le port d'écoute et joue le rôle de serveur pour le canal de données :



Les commandes `PORT` et `PASV` permettent donc de déterminer l'adresse IP et le numéro de port de la machine qui jouera le rôle de serveur pour le canal de données. L'échange de ces deux informations aura le format suivant : une chaîne de caractère du type "xxx ... ... (IP1,IP2,IP3,IP4,PORT1,PORT2)"

Il faudra donc :

- reconstituer l'adresse IP -> IP1.IP2.IP3.IP4
- déterminer le numéro de port -> (PORT1 x 256) + PORT2



La notion de mode actif et passif est extrêmement importante pour l'utilisation de FTP avec des pare-feux (*firewall*) :

– Si le client a un *firewall* (généralement celui-ci bloque toutes les demandes entrantes), le mode actif ne fonctionnera probablement pas car le serveur n'arrivera jamais à se connecter au client pour transférer les données (même la commande `LIST` a besoin d'un canal de données).
– Si le serveur a un *firewall*, il faut configurer celui ci pour qu'il laisse passer le port du serveur (21) et une plage de ports pour les transferts si le serveur accepte le mode passif.

Remarque : Le serveur doit avoir une plage de ports (entre X et Y) pour les transferts même s'il n'a qu'un seul client, car chaque commande `LIST` utilise un port de données, et un port de données ne peut plus être utilisé pendant presque une minute dans certains cas. Donc pour éviter tout problème à chaque fois que le client ou le serveur se mettent en attente, ils changent de port sans essayer de réutiliser le port précédent.

Installation

De nombreux serveurs FTP sont disponibles sous Linux (ftpd, proftpd, wu-ftpd, pure-ftpd, ...).

vsFTPd (*Very Secure FTP Daemon*), créé en 2000, est un serveur FTP qui mise beaucoup sur la sécurité, ce qui n'a rien d'étonnant puisqu'il est développé par Chris Evans chargé de la sécurité de Google Chrome. Il est distribué selon les termes de la licence Licence publique générale GNU.

Site officiel : https://security.appspot.com/vsftpd.html

Il suffit d'installer le paquet vsftpd :

```
$ sudo apt-get install vsftpd
...

$ sudo  service vsftpd status
vsftpd start/running, process 1805
```

Recherche du fichier de configuration de vsftpd :

```
$ find /etc -name "vsftpd.conf"
/etc/vsftpd.conf
/etc/init/vsftpd.conf
```

L'installation du paquet entraîne la création d'un utilisateur système `ftp`. Ce compte est systématiquement employé pour gérer les connexions FTP anonymes, et son répertoire personnel (`/srv/ftp/`) est la racine de l'arborescence mise à disposition des utilisateurs se connectant sur le service.

```
$ cat /etc/passwd | grep ftp
ftp:x:107:115:ftp daemon,,,:/srv/ftp:/bin/false

$ cat /etc/group | grep ftp
ftp:x:115:

$ ls -l /srv/ | grep ftp
drwxr-xr-x  2 root ftp  4096 mars  25 07:47 ftp/
```

Configuration

La configuration de **vsFTPd** est réalisée à partir du fichier `vsftpd.conf`.

Le fichier `vsftpd.conf` propose un grand nombre d'options dont les plus importantes sont :

– listen : permet de définir si le démon est en standalone (YES) ou dirigé par (x)inetd (NO)
– anonymous_enable : permet d'accepter les connexions anonymes
– local_enable : oblige les personnes à s'identifier avec un compte utilisateur
– write_enable : donne la permission d'écriture
– xferlog_file : écriture d'un log des fichiers
– ftpd_banner : bannière d'affichage à la connexion FTP
– chroot_local_user : permet de "chrooter" la connexion de l'utilisateur

La documentation en ligne : https://security.appspot.com/vsftpd/vsftpd_conf.html


```
chroot (change root) est une commande UNIX/Linux permettant de changer le répertoire racine d'un processus de la
     machine hôte. Cette commande permet d'isoler l'exécution d'un programme et d'éviter ainsi la compromission
     complète d'un système lors de l'exploitation d'une faille. Si un pirate utilise une faille présente sur l'
    application chrootée, il n'aura accès qu'à l'environnement isolé et non pas à l'ensemble du système d'
    exploitation. Cela permet donc de limiter les dégâts qu'il pourrait causer.
```

La configuration par défaut de vsFTPd est très restrictive :

- Le compte anonyme n'est pas autorisé à se connecter au serveur (anonymous_enable=NO)
- Les utilisateurs ne peuvent accéder qu'à leur compte, et en lecture seule (local_enable=YES)

```
$ cat /etc/vsftpd.conf | grep -e "^\(listen\|anonymous\|local\|write\|chroot\)"
listen=YES
anonymous_enable=NO
local_enable=YES
```

Tests

On peut utiliser le client `ftp` en ligne de commande.

```
$ ftp 192.168.52.9
Connected to 192.168.52.9.
220 (vsFTPd 2.3.5)
Name (192.168.52.9:tv): tv
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated.  Commands are:
...
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
...
ftp> bye
221 Goodbye.
```

On peut aussi utiliser un client graphique comme **FileZilla**.

On peut alors vérifier les restrictions d'accès :

```
// Envoi d'un fichier
Commande :      STOR test.txt
Réponse :       550 Permission denied.

// Création d'un répertoire
Commande :      MKD /home/tv/test
Réponse :       550 Permission denied.

// Renommage d'un fichier
Commande :      RNFR fichier.txt
Réponse :       550 Permission denied
```

Liste des commandes ftp : http://fr.wikipedia.org/wiki/Liste_des_commandes_ftp

```
La version 2.3.5 livrée en standard avec Ubuntu est boguée. L'utilisation du mode chrooté provoque une erreur :
    500 OOPS: vsftpd: refusing to run with writable root inside chroot(). L'utilisateur ne peut pas accéder à
    son répertoire personnel. Pour contourner ce bug, quelqu'un a réalisé un backport de l'option
    allow_writeable_chroot depuis vsftpd 3 en attendant que celle-ci soit présente dans les dépôts.
```

```
// vérification de la version installée
```



```
$ vsftpd -v
vsftpd: version 2.3.5

$ wget http://ppa.launchpad.net/thefrontiergroup/vsftpd/ubuntu/pool/main/v/vsftpd/vsftpd_2.3.5-1
    ubuntu2ppa1_amd64.deb

$ sudo dpkg --install vsftpd_2.3.5-1ubuntu2ppa1_amd64.deb
```

Maintenant, l'option allow_writeable_chroot=YES est utilisable dans `vsftpd.conf`.

