ProFTPD

>>> Configuration de ProFTPD sur Debian

Description :

Cet article vous explique comment installer et configurer ProFTPD. Cet article abordera seulement la configuration du FTP, deux autres articles suivront pour la configuration du FTPS Explicite et FTPS Implicite.

ProFTPD

>>> Configuration de ProFTPD sur Debian

Sommaire :

I) Introduction

Lorsque le projet a commencé, le serveur le plus couramment utilisé était wu-ftpd. Alors que wu-ftpd fournit d'excellentes performances et est généralement un bon produit. Mais il manque de nombreuses fonctionnalités trouvées dans les nouveaux serveurs FTP Win32, de plus il a un historique de sécurité médiocre. Beaucoup de gens, y compris les développeurs qui travaillent sur ProFTPD, avaient passé beaucoup de temps à corriger les bugs. Malheureusement, il est vite devenu évident qu'une refonte complète était nécessaire pour mettre en œuvre les fonctionnalités souhaitées.

En plus de wu-ftpd, il y a quelques autres serveurs FTP disponibles qui sont conçus pour être léger et sécurisé. Par exemple, Troll FTP est un excellent démon FTP qui est beaucoup plus sécurisé et moins consommateur de ressources que wu-ftpd. Malheureusement, même s'il est tout à fait approprié pour les services FTP de base, il n'offre pas le jeu de fonctionnalités requis pour les sites FTP plus sophistiqués.

ProFTPD n'est pas basé sur un autre serveur, c'est un démon indépendant depuis le début. Un certain nombre de sites bien connus et de trafic élevé utilisent ProFTPD.

Voici les points fort de ProFTPD :
- Une configuration puissante similaire à celle d'Apache
- Des serveurs virtuels
- Comptes facilement chrootés
- Pas besoin de binaire dans les prisons ou comptes chrootés

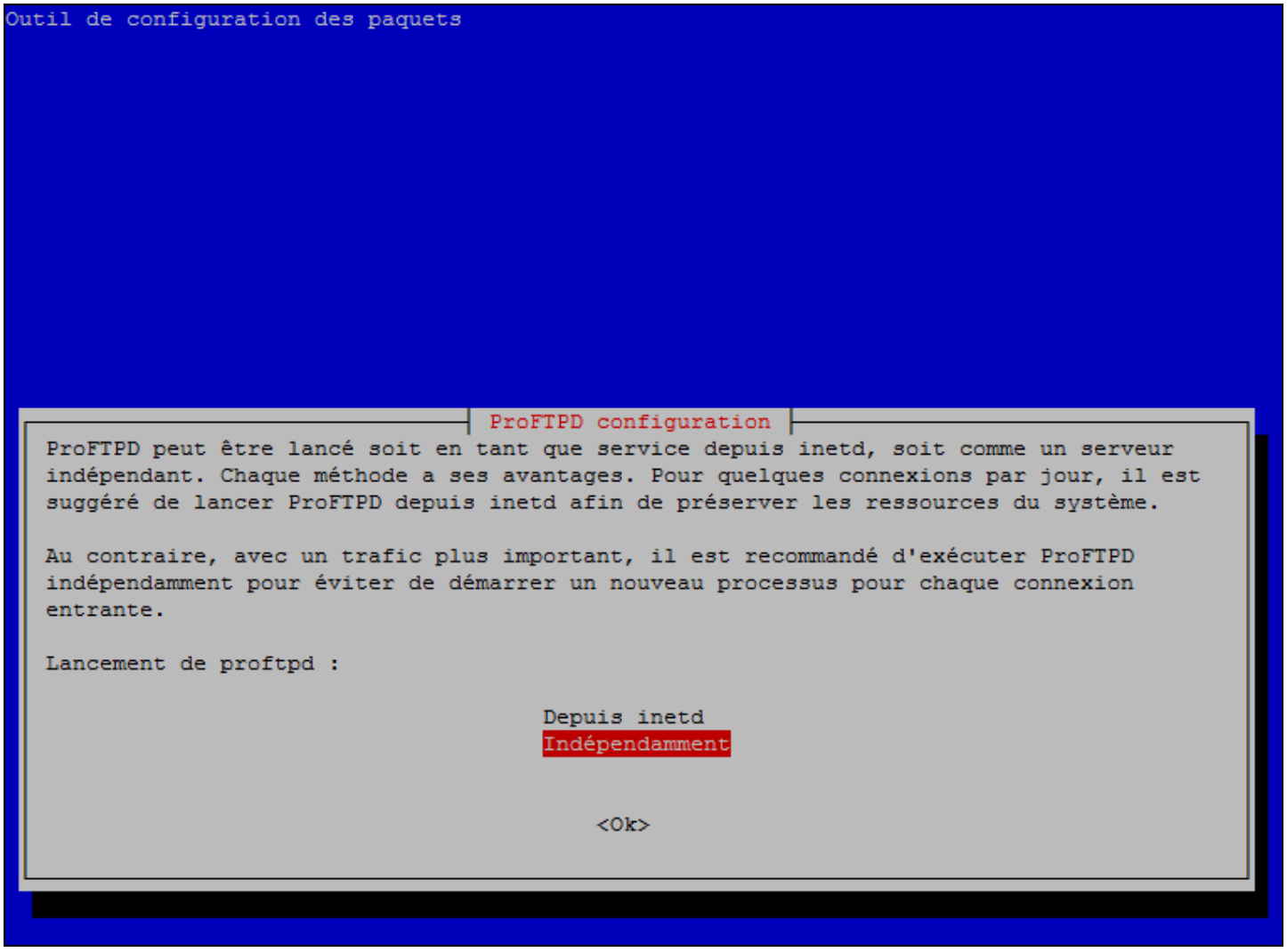
II) Installation

Pour installer le paquet "**Proftpd**", tapez la commande suivante :

```
aptitude install proftpd
```

Lors de l'installation, ProFTPD vous propose deux modes :
- Inetd
- Standalone

ProFTPD peut être lancé soit en tant que service depuis inetd, soit comme un serveur indépendant. Chaque méthode a ses avantages. Pour quelques connexions par jour, il est suggéré de lancer ProFTPD depuis inetd afin de préserver les ressources du système.
Au contraire, avec un trafic plus important, il est recommandé d'exécuter ProFTPD indépendamment pour éviter de démarrer un nouveau processus pour chaque connexion entrante.



Le serveur FTP fonctionne par défaut après installation. Vous trouverez le fichier de configuration dans le répertoire "/etc/proftpd/".

III) Configuration de ProFTPD

1) Configuration de base

- Pour activer/désactiver l'utilisation de IPv6 :

- off Désactive l'identification distante

```
UseIPv6 on
```

- Protocole pour tenter d'identifier le nom d'utilisateur distant. L'activation de cette option peux engendrer des temps de réponse plus long.

- off Désactive l'identification distante

```
IdentLookups off
```

- Nom du serveur FTP :

```
ServerName "Debian"
```

- Mode de fonctionnement du serveur :

```
ServerType standalone
```

- Affichage du message de bienvenue dans les logs du client FTP.

```
DeferWelcome off
```

- Limiter le nombre de processus simultanés autorisés

```
MaxInstances 30
```

- Le message de bienvenue est stocké dans un fichier. Pour définir l'emplacement du message vous devez configurer la ligne :

```
DisplayLogin welcome.msg
```

- Si vous n'indiquez pas de chemin comme ci-dessus, alors le serveur cherchera le fichier "welcome.msg" dans le répertoire personnel de l'utilisateur.
- Sinon vous devez indiquer le chemin du fichier.
- Vous pouvez utiliser le fichier exemple :

```
DisplayLogin /srv/ftp/welcome.msg
```

- Pour activer/désactiver l'affichage des liens symboliques :

- off Désactive l'identification distante

```
ShowSymlinks on
```

- Pour bloquer les utilisateurs dans leur répertoire personnel (décommenter la ligne suivante) :

```
# DefaultRoot ~
```

- Pour bloquer tous les utilisateurs dans leur répertoire personne SAUF user1 :

```
# DefaultRoot ~ !user1
```

- Si vous voulez bloquer tous les utilisateurs dans un répertoire, par exemple : /srv/ftp/ :

```
DefaultRoot /srv/ftp/
```

Si vous avez l'arborescence suivante :
- /srv/ftp
- /srv/ftp/www/

- Vous voulez que l'utilisateur atterrisse dans le répertoire **/srv/ftp/www/** et qu'il ne puisse pas remonter au-delà du répertoire **/srv/ftp/** :

```
DefaultRoot /srv/ftp/
DefaultChdir /srv/ftp/www/
```

- Autoriser/refuser aux fichiers nouvellement transférés d'écraser les fichiers existants.

```
AllowOverwrite on
```

- Emplacement des fichiers de logs :

```
TransferLog /var/log/proftpd/xferlog
SystemLog /var/log/proftpd/proftpd.log
```

2) Autres options

Voici quelques options à rajouter qui ne sont pas dans le fichier de base, mais qui peuvent être utiles.

- Autoriser la connexion avec l'utilisateur **"root"**.

```
RootLogin on
```

- Activez l'utilisation du fichier /etc/ftpusers qui donne la liste des utilisateurs n'ayant **pas** accès au serveur ftp.

```
UseFtpUsers on
```

- Autorisez les clients à reprendre les uploads vers le serveur.

```
AllowStoreRestart
```

- Autorisez les clients à reprendre les téléchargements.

```
AllowRetrieveRestart on
```

- Cachez tous les éléments inaccessibles aux utilisateurs.

```
HideNoAccess
```

- Pour ne pas afficher le nom du serveur FTP et son numéro de version

```
ServerIdent off
```

3) Configuration accès anonymes

Nous allons maintenant mettre en place un accès Anonymous à notre serveur FTP. L'accès anonymous autorise n'importe quelle personne connaissant le serveur à se connecter sans login/password. Je vais donc isoler les utilisateurs Anonymous dans un répertoire.

- Commencez par créer un répertoire :

```
mkdir /srv/ftp/Anonymous
```

- Editez ensuite le fichier de configuration.

```
vim /etc/proftpd/proftpd.conf
```

- Décommentez la partie ""
- Modifier la première ligne, pour indiquer le dossier réserver au Anonymous. Vous devez obtenir ceci :

```
User ftp
Group nogroup
# We want clients to be able to login with "anonymous" as well as "ftp"
UserAlias anonymous ftp
# Cosmetic changes, all files belongs to ftp user
DirFakeUser on ftp
DirFakeGroup on ftp

RequireValidShell off

# Limit the maximum number of anonymous logins
MaxClients 10

# We want 'welcome.msg' displayed at login, and '.message' displayed
# in each newly chdired directory.
DisplayLogin welcome.msg
DisplayChdir .message

# Limit WRITE everywhere in the anonymous chroot


DenyAll



# Uncomment this if you're brave.
#
# # Umask 022 is a good standard umask to prevent new files and dirs
```

```
# # (second parm) from being group and world writable.
# # Umask 022 022
# #
# # DenyAll
# #
# #
# # AllowAll
# #
# #
#
```

Voilà votre accès Anonymous fonctionne. Les utilisateurs anonymes peuvent seulement télécharger les fichiers.

4) Minutes Sécurités

Voici quelques conseils pour sécuriser votre serveur FTP :

- Cachez les informations retournées par le serveur FTP :

```
ServerIdent off
IdentLookups off
```

- Désactiver l'affichage des liens symboliques :

```
ShowSymlinks off
```

- Si vous n'utilisez pas IPv6, désactivé le :

```
UseIPv6 off
```

- Utilisez le fichier **"/etc/ftpusers"** pour interdire les connexions FTP aux utilisateurs non autorisés et aux utilisateurs systèmes :

```
UseFtpUsers on
```

- Limiter le nombre de Clients connecté simultanément :

```
MaxClients 20
```

- Désactivez le reverse DNS :

```
UseReverseDNS off
```

IV) Configuration de "Limites"

Les sections de configuration **"Limit"** de ProFTPD permettent un contrôle puissant et fin sur la personne autorisée à utiliser les commandes FTP. Cette puissance est au prix de la complexité, cependant ce document décrit certaines des choses à garder à l'esprit lors de l'écriture des sections **"Limit"**.

1) Priorité

Peut-être la partie la plus difficile de l'utilisation de **"Limit"** est de comprendre ses règles de priorités, qui dictent quelles restrictions de **"Limit"** s'appliquent quand la priorité est discutée. Tout d'abord, il existe trois types de paramètres dans une directive **"Limit"** :
- Les commandes FTP **"brutes"**
- Les groupes de commandes FTP
- Le mot-clé "ALL"

Les commandes FTP **"brutes"** sont énumérées ici, y compris les commandes FTP commandées par RFC, qui sont souvent absentes d'une configuration **"Limit"** complète.

Les groupes de commandes FTP sont :

- ALL
  - Enveloppe : Toutes les commandes FTP (but not LOGIN)
- DIRS
  - Enveloppe : CDUP, CWD, LIST, MDTM, MLSD, MLST, NLST, PWD, RNFR, STAT, XCUP, XCWD, XPWD
- LOGIN
  - Enveloppe : client logins
- READ
  - Enveloppe : RETR, SIZE
- WRITE
  - Enveloppe : APPE, DELE, MKD, RMD, RNTO, STOR, STOU, XMKD, XRMD

Les **"Limit"** qui utilisent des commandes FTP **"brutes"** ont la priorité la plus élevée, suivies par les **"Limit"** qui utilisent les groupes de commandes et ayant la plus basse priorité, le mot clé **"ALL"**. Si un **"Limit"** est à la fois des commandes **"brutes"** et des groupes de commandes, il se résume à l'ordre d'apparition dans ProFTPD.conf.

2) Commandes SITE

Pour appliquer un **"Limit"** à une commande **"SITE"**, combinez **"SITE"** et la commande (par exemple "CHMOD") par un underscore "_", comme suit :

```
```

Ainsi, afin de fixer une limite au SITE CHMOD, on aurait :

```
    DenyAll
```

3) Héritage

La plupart des sections **"Limit"** apparaissent dans les sections **"Directory"** de proftpd.conf. Cela signifie que, comme les autres effets de configuration **"Directory"**, les **"Limit"** seront hérités par tous les sous-répertoires qui apparaissent dans le chemin **"Directory"**, sauf si explicitement remplacés par une section **"Limit"**. Cela signifie que l'on peut configurer une section **"Limit"** refusant toutes les commandes FTP pour tous les répertoires, puis autoriser explicitement les groupes de commandes READ ou WRITE FTP dans les sous-répertoires appropriés.

Utilisation de AllowUser et DenyUser
L'utilisation de la directive de configuration AllowUser est gênante, surtout lorsqu'une seule directive AllowUser est utilisée pour autoriser l'accès à certaines commandes FTP uniquement à certains utilisateurs. ProFTPD utilise la même fonction pour analyser les directives AllowUser et AllowGroup (et autres). Cette fonction analyse la liste des noms de ces directives comme une liste ET booléenne, ce qui signifie que chaque nom de la liste doit correspondre à TRUE (doit correspondre) à l'utilisateur actuel pour que la directive s'applique. Pour AllowGroup, cela est logique et permet une grande souplesse. Toutefois, cela n'a pas de sens pour AllowUser, car un utilisateur ne peut pas être plusieurs utilisateurs en même temps. Il s'agit d'un problème connu, et une bonne solution approfondie est en cours de développement. En attendant, cependant, il existe une solution de contournement pour permettre à plusieurs utilisateurs via la directive AllowUser. Plutôt que d'inscrire les utilisateurs à l'aide d'un seul AllowUser, en utilisant un AllowUser séparé pour chaque utilisateur. Par exemple, au lieu de :

4) AllowUser bob, dave, wendy

Essayez d'utiliser :
- AllowUser bob
- AllowUser dave
- AllowUser wendy

Tout cela s'applique également à la directive DenyUser.

Un autre point important à garder à l'esprit est que les noms utilisés dans les sections **"Limit"**, utilisant AllowUser, DenyUser, AllowGroup et DenyGroup, ne sont pas résolus à un ID puis appliqués. Les limites ne s'appliquent qu'aux noms. Pourquoi est-ce important ? Considérez le cas où le site utilise des utilisateurs virtuels, où deux noms d'utilisateur différents sont affectés au même UID. Différentes limites peuvent être appliquées à chaque nom séparément. Ne présumez pas que les limites sont appliquées aux identifiants sous-jacents.

5) Utilisation de l'ordre

Une chose qui parfois désactive certains administrateurs est la différence entre les directives de configuration de ProFTPD et de commande d'Apache. Pour Apache, un ordre de "Autoriser", "Refuser" signifie que l'accès est refusé par défaut, à moins qu'une directive "Autoriser" autorise explicitement l'accès. Un ordre de "Refuser", "Autoriser" signifie que l'accès est autorisé par défaut, à moins qu'une directive "Refuser" refuse explicitement l'accès. Ceci est différent de ProFTPD, où un ordre de "Autoriser", "Refuser" permet l'accès par défaut, sauf si refusé par une directive "Refuser". "Refuser", "Autoriser" refuse l'accès par défaut, sauf accord explicite d'une directive "Autoriser".

6) Exemples

L'explication ci-dessus extrait du site ProFTPD, est complexe. Voici quelques exemples pour illustrer le chapitre par des cas concrets :

- On souhaite autoriser seulement **"user1"** à ajouter et écrire dans le répertoire **/srv/ftp**.

```
DenyAll


AllowAll


AllowAll
```

```
AllowUser user1
```

- Limit All : Permet de tout interdire
- Limit LOGIN : Permet d'autoriser tous les utilisateurs de se connecter
- Limit DIRS : Permet d'autoriser toutes les commandes de dossier
- Limit WRITE : Permet d'autoriser toutes les commandes d'écriture mais seulement pour l'utilisateur **"user1"**

- On souhaite autoriser **"user1"** à ajouter/écrire des fichiers/dossier dans le répertoire. Mais on souhaite que **"user2"** n'est le droit que de créer des dossiers.

```
DenyAll

AllowAll

AllowAll

AllowUser user2

AllowUser user1
```

- Limit All : Permet de tout interdire
- Limit LOGIN : Permet d'autoriser tous les utilisateurs de se connecter
- Limit DIRS : Permet d'autoriser toutes les commandes de dossier
- Limit MKD : Permet d'autoriser seulement la commande de création de dossier mais seulement pour l'utilisateur **"user2"**
- Limit WRITE : Permet d'autoriser toutes les commandes d'écriture mais seulement pour l'utilisateur **"user1"**

- On souhaite que seulement l'utilisateur **"user2"** ne puissent pas ajouter/écrire dans le répertoire. Mais autoriser tous les autres utilisateurs.

```
DenyAll

AllowAll

AllowAll

AllowAll

DenyUser user2
```

- Limit All : Permet de tout interdire
- Limit LOGIN : Permet d'autoriser tous les utilisateurs de se connecter
- Limit DIRS : Permet d'autoriser toutes les commandes de dossier
- Limit WRITE : Permet d'autoriser toutes les commandes d'écriture pour tous les utilisateurs.
- Limit WRITE : Permet d'interdire toutes les commandes d'écriture mais seulement pour l'utilisateur **"user2"**

V) Configuration des quotas

1) Configuration des quotas méthode par fichier

- Editez le fichier **"/etc/proftpd/proftpd.conf"**

```
vim /etc/proftpd/proftpd.conf
```

- Recherchez la ligne correspondant au module **"mod_quotatab.c"** et ajoutez les lignes suivantes :

```
QuotaEngine on
QuotaLimitTable file:/etc/proftpd/ftpquota.limittab
QuotaTallyTable file:/etc/proftpd/ftpquota.tallytab
```

- Tapez la commande suivante pour créer une table de type "limit" :

```
ftpquota --create-table --type=limit
```

- Créez ensuite une table de type "tally" :

```
ftpquota --create-table --type=tally
```

- Redémarrer le service ProFTPD.

```
service proftpd restart
```

- Pour gérer/déclarer des quotas la commande à utiliser est **"ftpquota"**. Voici les arguments disponibles :

- **—add-record** : Permet d'ajouter un nouveau quota. Les autres choix sont :
  - **—update-record** : Met à jour un enregistrement de quota avec les limites spécifiées. Toutes Les limites laissées non précisées seront mises à jour. (Cette option requiert le nom "—name" et "—quota-type")
- **—type=limit** : Permet de dire qu'on souhaite définir un type "limit"
  - **—type=tally**
- **—bytes-upload=x** : Déclare qu'on souhaite limiter le nombre d'octets en uploads et qu'on limite l'upload à x (x étant une valeur). Les autres choix sont :
  - **—bytes-download** : Déclare qu'on souhaite limiter le nombre d'octets en downloads.
  - **—bytes-xfer** : Déclare qu'on souhaite limiter le nombre d'octets en uploads **ET** en downloads.
  - **—files-upload** : Déclare qu'on souhaite limiter le nombre de fichiers en uploads.
  - **—files-download** : Déclare qu'on souhaite limiter le nombre de fichiers en downloads.

    - **—files-xfer** : Déclare qu'on souhaite limiter le nombre de fichiers en uploads **ET** en downloads.
- **—units=y** : y permet de définir l'unité de la valeur x. Les valeurs possibles :
    - **"B"** or **"byte"**
    - **"Kb"** or **"kilo"**
    - **"Mb"** or **"mega"**
    - **"Gb"** or **"giga"**
    - Par défaut la valeur est **"byte"**
- **—name=user1** : Définit le nom "user1"
- **—quota-type=user** : Permet de définir que le nom "user1" est un username. Les autres valeurs possibles sont :
    - **"user"**
    - **"group"**
    - **"class"**
    - **"all"**

Voici quelques exemples de quota :

- Définir un quota d'upload à 2Gb :

```
ftpquota --add-record --type=limit --bytes-upload=2 --units=Gb --name=user1 --quota-type=user
```

- Définir un quota de download à 200Gb :

```
ftpquota --add-record --type=limit --bytes-download=2 --units=Gb --name=user1 --quota-type=user
```

- Définir un quota de 2 fichiers maximum en upload :

```
ftpquota --add-record --type=limit --files-upload=2 --name=user1 --quota-type=user
```

- Si vous souhaitez modifier le quota, par exemple mettre le quota à 4Gb :

```
ftpquota --update-record --type=tally --bytes-upload=4 --units=Gb --name=user1 --quota-type=user
```

- Pour réinitialiser les compteurs :

```
ftpquota --update-record --type=tally --name=user1 --quota-type=user
```

- Effacer toutes les données de quota d'un utilisateur :

```
ftpquota --delete-record --type=limit --name=user1 --quota-type=user
ftpquota --delete-record --type=tally --name=user1 --quota-type=user
```

- Pour voir les quotas consommés par les utilisateurs :

```
ftpquota --show-records --type=tally
```

- Vous devez obtenir ceci :

```
----------------------------------------
```

```
Name: user1
Quota Type: User
Uploaded bytes: 652367872.00
Downloaded bytes: unlimited
Transferred bytes: unlimited
Uploaded files: 2
Downloaded files: 0
Transferred files: 0
```

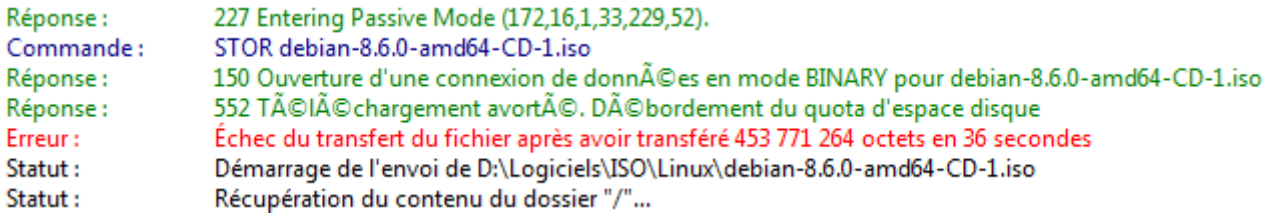
- Pour voir les quotas du serveur :

```
ftpquota --show-records --type=limit
```

- Vous devez obtenir ceci :

```
----------------------------------------
Name: foo
Quota Type: User
Per Session: False
Limit Type: Hard
Uploaded bytes: 214748364800.00
Downloaded bytes: unlimited
Transferred bytes: unlimited
Uploaded files: unlimited
Downloaded files: unlimited
Transferred files: unlimited
```

Voici un test du quota : L'utilisateur **"user1"** souhaite transférer un fichier de 3Go alors qu'il est limité à 2Go. Voici les logs de son client FTP :



2) Configuration des quotas méthode par MySQL

La méthode étant assez longue, vous trouverez donc un article dédié : Cliquez ici