

Introduction à proftpd

par DuF

Un exemple de configuration de proftpd

Introduction

Tout d'abord, si je rédige cet article, c'est tout simplement que j'ai moi-même eu des difficultés à me servir de ce logiciel et j'ai eu énormément de problèmes pour trouver une documentation précise, claire et en français pour ce logiciel. J'en écris donc une qui même si elle n'est pas aussi claire ou précise que voulue, a au moins le mérite d'être en français, soyez conscient que ce n'est pas si mal :) . Je tiens aussi à préciser que dans cet article je vais parler de proftpd et de sa configuration seulement par rapport aux besoins que j'ai eu à un moment donné, qui étaient avant tout liés à une utilisation personnelle et non professionnelle, afin de remplacer `wu-ftpd` qui est par défaut avec la plupart des distributions, si ce n'est tout les distributions (c'est proftpd avec la Mandrake 8.2 que j'avais ;-). Donc pour moi, le besoin par rapport à un serveur FTP, c'était de pouvoir partager des ressources que j'ai sur mon ordinateur et qui sont sur une partition FAT32 (ce fut un besoin à un instant T) mais aussi de donner la possibilité à un utilisateur de pouvoir écrire dans un endroit bien précis, restreint en essayant de suivre quelques règles de sécurité de base (qui sont vraiment de base)...

Je reprends cette documentation aujourd'hui pour prendre en compte quelques changements, remplacements de certaines directives, pour corriger des erreurs (si si il y en avait :)) et ajouter une partie concernant le `mod_tls` pour faire des transferts chiffrés (à ne pas confondre avec du transfert FTP over SSH). La version actuelle de proftpd servant de support à cette mise à jour, est la version 1.2.8.

Installation

Bon et bien là c'est vraiment simple, il faut une connexion internet (disons que c'est mieux pour récupérer la dernière version de proftpd sur le site www.proftpd.org) et sinon si jamais il vous manque quelque chose, lors de l'installation tout ce qui vous manque sera indiqué et donc vous pourrez aller récupérer tout ce dont vous avez besoin sur internet. Pour ma part il me manquait 2 choses, comme il indique à la fin précisément ce qui manque, avec même un lien internet de la ressource manquante, il faut juste quelques minutes pour répondre aux besoins de proftpd. Si je ne rentre pas plus dans le détail pour cette partie c'est tout simplement que les ressources nécessaires à proftpd dépendent totalement de votre configuration et de votre distribution, donc je pourrai en mentionner alors que vous n'en aurez pas besoin et en oublier alors que vous en aurez besoin... Donc pas la peine de vous envoyer sur de mauvaises pistes, lisez juste le message d'erreur lors de l'installation et il n'y aura pas de souci.

Je vous conseille de prendre la dernière version (actuellement la 1.2.8 est la dernière version stable), pour l'installer, c'est comme d'habitude :

```
tar zxvf proftpd-1.2.8.tar.gz
```

Ensuite on se place dans le répertoire nouvellement créé, on lit bien le fichier d'installation (INSTALL ou README) et on lance les commandes habituelles.

Il est possible de récupérer le package proftpd sous la forme de package, il n'est pas la peine d'expliquer cela.

Configuration

Tout d'abord dans la configuration de proftpd il n'y a qu'un seul fichier qui rentre en ligne de compte, c'est : `/etc/proftpd.conf` (le chemin peut être différent suivant votre distribution et/ou distribution).

C'est dans ce fichier que vous allez "tout" définir (enfin presque) concernant la configuration du serveur. Mais attention, cela concerne la configuration du serveur, ce n'est pas ici que vous allez définir les utilisateurs qui ont accès ou non au serveur (du moins en parti). Occupons nous tout d'abord des utilisateurs !

Les utilisateurs

Attention, tous les utilisateurs se connectant sur le serveur proftpd doivent exister réellement sur le système (avec un uid). Il est cependant de faire un alias d'un utilisateur n'existant vers un utilisateur existant, pour cela regarder l'exemple concernant le [contexte de configuration anonyme](#), avec l'explication sur le `UserAlias`. Noter aussi que pour ma part j'ai fait le choix de créer les utilisateurs avec un "faux shell" plutôt que de faire des alias, mais j'explique cela juste après.

Avec proftpd on a le choix dans sa stratégie. J'ai choisi d'utiliser le fichier `/etc/ftpusers` pour définir tous les utilisateurs qui n'ont pas accès au service FTP. Tous les utilisateurs présent dans ce fichier ne pourront donc en aucun cas se connecter au service FTP.

Vous pouvez indiquer à proftpd d'utiliser le fichier `/etc/ftpusers` avec la directive :

```
UseFtpUsers
```

Mais par défaut il le fait, donc vous pouvez vous servir de `/etc/ftpusers` sans dire explicitement dans le fichier `proftpd.conf` que vous souhaitez vous en servir, sans utiliser la directive `UseFtpUsers` donc (j'espère que vous suivez, toute façon j'y reviendrai après).

Pour ma part j'ai laissé le fichier `/etc/ftpusers` comme il était en ajoutant tous les utilisateurs qui ont un shell sur ma machine, voici un exemple de fichier `/etc/ftpusers` :

Exemple de fichiers ftpusers

```
root  
bin
```

```
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
gopher
postgres
squid
gdm
htdig
dhcpcd
...
...
...
nobody
anonymous
DuF
ftp
```

Vous remarquerez que l'utilisateur `anonymous` est indiqué, c'est tout simplement que je ne souhaite pas donner un accès anonyme, si vous n'avez pas utilisé d'un accès anonyme, je vous conseille d'ajouter, vous aussi, `anonymous` dans le fichier `/etc/ftpusers`.

Sinon comme je l'ai dit plus haut, j'ai choisi que tous mes utilisateurs ayant un shell (`ksh`, `bash`...) n'aient pas accès à mon serveur FTP, je vais donc créer des utilisateurs spécifiquement pour l'utilisation du FTP. Ces utilisateurs que je vais créer n'auront pas de shell leur permettant de se connecter en telnet, etc... Pour cela, j'indique le shell suivant :

```
/bin/false
```

Pour que ce shell soit "valide", il faut indiquer dans le fichier `/etc/shells` la ligne `/bin/false` si cela n'a pas déjà été fait. Car `proftpd` par défaut n'accepte pas la connexion si le shell de l'utilisateur n'est pas "valide" donc indiqué dans `/etc/shells` (bien sûr on se sera assuré que le shell en question existe effectivement dans l'arborescence). Si le fichier `/etc/shells` n'existe pas, par contre il accordera la connexion (personnellement j'ai trouvé ça étonnant, mais il doit y avoir des raisons à cela, il faut donc en tenir compte).

De plus vous pouvez spécifier lors dans la configuration de `proftpd.conf` si vous souhaitez que `proftpd` vérifie que les utilisateurs aient oui ou non un shell valide. Cela se fait par la directive : `RequireValidShell` (qui prend comme argument `on` ou `off` !)

Sinon concernant la création des utilisateurs, pour cela je vous renvoie aux tutoriaux déjà présents sur Léa, si je devais faire une recommandation, c'est de créer un groupe unique qui servira à placer tous les utilisateurs avec le shell `/bin/false` dedans, cela peut s'avérer utile par la suite, et de toute façon il n'est pas utile de créer un groupe pour chaque nouvel utilisateur.

Pour la création pur et dur des utilisateurs, vous pouvez faire un `adduser` ou `useradd`, pour plus d'options c'est très simple, il faut faire `man adduser` :) ou sinon maintenant, il existe des outils graphiques qui le font aussi, donc en récapitulant, vous créez les "users" par exemple `user1` et `user2` et vous les mettez dans le groupe `ftptest`.

Voilà déjà pour la partie spécifiques aux utilisateurs (users), je vais en reparler dans la partie du fichier `proftpd.conf`.

Le fichier Proftpd.conf

Contexte de configuration Server Config

On entre dans le vif du sujet.

Tout d'abord le fichier `proftpd.conf` se divise en plusieurs parties, qui ne sont pas toutes nécessaires (vitales). Je vais essayer, pour garder une cohérence avec la documentation en ligne du site www.proftpd.org de reprendre les mêmes terminologies, on a donc plusieurs contextes de configuration :

```
server config, <Global>, <Anonymous>, <VirtualHost>, <Limit>, <Directory>, .ftpassess
```

Donc pour résumé sur les contextes de configuration, on peut avoir un fichier `proftpd.conf` avec seulement le contexte : `server config`.

Si le contexte "server config" n'est pas entre `<>` c'est tout simplement qu'il est implicite, ce n'est pas utile de le mentionner. Notez aussi que les options qui sont à l'intérieur des contextes de configuration sont les directives !

Les directives les plus courantes

On commence par un fichier `proftpd.conf` ne contenant qu'une directive et le contexte de configuration obligatoire qu'est "server config".

```
#début du fichier proftpd.conf
```

```
UseFtpUsers      off
```

#fin du fichier proftpd.conf d'exemple, pour info ce fichier en l'état n'est pas valide

On peut dire que tous les [contextes de configuration](#) sont forcément inclus dans le contexte "server config" et que l'on peut avoir un contexte comme <Directory> qui soit dans un "sous" contexte comme <Anonymous>. C'est un principe de configuration imbriquée.

Les premières directives que l'on va avoir dans le fichier `proftpd.conf` vont concerner le mode de lancement du serveur et les infos le concernant.

```
ServerName       "Server FTP du DuF"
ServerType       standalone
DefaultServer    on
```

ServerName => je ne reviens pas dessus

ServerType => est important (indispensable) car il indique si le serveur est démarré par vos soins en faisant : `/etc/init.d/proftpd start` (ou `/etc/rc.d/init.d/proftpd start`). Ou alors si il a la valeur `inetd` il est démarré par le meta-daemon du même nom (qui peut selon les distributions être `xinetd` au lieu d'`inetd`, mais il faut quand même laisser `inetd` dans le fichier `proftpd.conf`, car `xinetd` sera interprété comme un mauvais paramètre).

DefaultServer => cela est utile si vous faites des virtualhost, si vous n'utilisez qu'un "server config" sans virtualhost alors ce n'est pas utile de l'indiquer. En fait cette directive vérifie qu'elle configuration du serveur sera prise en compte (soit server config ou un des virtual hosts, ou anonyme....) Si il n'y aucune configuration de prévue, "l'inconnu" aura le message suivant : "no server available to service your request" et déconnecté.

Ensuite nous passons à des options dont l'utilité est très simple à comprendre :

```
AllowStoreRestart on
Port              45000
Umask             022
MaxInstances      30
```

AllowStoreRestart => permet d'autoriser les clients à reprendre les uploads vers vous, ce n'est pas cette directive qui leur permet de reprendre quand ils téléchargent depuis votre serveur. Donc cette option, AllowStoreRestart n'est utile que si vous autorisez au moins une personne à écrire chez vous.

Port => je passe, simple à comprendre sinon je me demandes ce que vous faites là :) c'est bien évidemment le numéro de port sur lequel le client se connecte.

Umask => c'est comme dans unix en général, 022 est une valeur qui est bien, donc laissez-là à 022, si vous voulez en savoir plus sur umask en général, consultez notre ami à tous : www.google.fr/linux

MaxInstances => comme c'est indiqué dans le fichier par défaut de `proftpd.conf`, cela sert à spécifier le nombre de processus fils maximum que va gérer (utiliser) proftpd, et comme indiqué, au delà d'une valeur de 30 vous être vulnérable à des attaques de type Ddos, donc laissez à 30, pour une utilisation, sincèrement c'est largement suffisant.

Ensuite viennent 2 paramètres très important, on va indiquer sous quel utilisateur le serveur FTP est lancé, il ne s'agit donc pas de mettre root :)

Le user `nobody` et le group `nogroup` sont les paramètres par défaut et à mon avis ils sont très bien, donc on n'y touche pas car il faut que l'utilisateur qui lance le serveur n'ai pas trop de privilèges.

Ensuite il est question de l'option `PersistentPasswd`, dont l'utilité ici est pas évidente à expliquer étant donné que je n'ai pas compris l'explication sur le site de proftpd, je l'ai laissé à `off`. En fait si je me trompes pas, en le mettant à `on` cela permet à proftpd de chercher lui même dans `/etc/passwd` la validité des mots de passe, mais c'est à vérifier.

Ensuite vient une option qui a mon avis est très importante pour les personnes qui comme moi veulent partager une petite connexion adsl, il faut limiter le nombre de tentatives de logins (l'adsl ça sature vite :) :

```
MaxLoginAttempts 3
```

Ensuite nous avons une option pas du tout vitale mais sympa je trouve, cela concerne la personnalisation de votre serveur, tout du moins le message d'accueil :

```
AccessGrantMsg    "Bienvenue %u chez DuF...."
```

Vous remarquerez le `%u`, c'est un paramètre qui récupère le user qui se connecte et le remplace en lieu et place de `%u`. Cette option indique le message de bienvenue quand l'utilisateur a réussi à se connecter.

Pour ne pas donner d'info précise sur le serveur, je conseil de mettre à on l'option suivante :

```
DeferWelcome      on
```

Contexte de configuration <Limit> appliqué à notre cas

Maintenant on va indiquer un [contexte de configuration](#) Limit qui va s'appliquer au [contexte de configuration](#) server config (partie générale du serveur) plus concrètement, c'est à dire tous ceux qui vont se connecter et qui ne seront pas concernés par un virtualhost. En fait ce [contexte de configuration](#) est utile (très important) pour tous ceux qui souhaitent partager des données sur des partitions FAT32 par exemple tout en limitant les possibilités (écriture, création de répertoires...)

Pour cela j'ai utilisé le [contexte de configuration](#) Limit avec les directives suivantes :

```
<Limit MKD RNFR RNT0 DELE RMD STOR CHMOD SITE_CHMOD SITE XCUP WRITE XRMD XPWD>
  DenyAll
</Limit>
```

Bon moi j'en ai mis beaucoup pour l'exemple, je vais juste en expliquer certaines :

- MKD : création de répertoire
- RNFR : (rename from) empêche de pouvoir renommer
- RNT0 : (rename to) c'est la suite de RNFR en fait, donc si RNFR est interdit, ce n'est pas utile de le mettre, mais bon
- DELE : suppression de fichiers
- STOR : écriture de fichiers depuis un client vers notre serveur proftpd
- CHMOD : changement de permission sur les fichiers (et répertoires)
- RMD : suppression de répertoire

Il est aussi possible d'utiliser des mots clefs comme READ et WRITE qui englobent plusieurs commandes, et vont limiter l'accès en lecture et en écriture. Pour le reste des options vous pouvez consulter les commandes de la section Limit sur le site de www.proftpd.org

Contexte de configuration Global

Là on arrive à une section relativement importante, le [contexte de configuration](#) <Global>

En effet ce contexte de configuration peut être utilisé à l'intérieur de la "server config" et du contexte de configuration <VirtualHost>

Tout ce qui va être défini dans <Global> va être appliqué à l'ensemble du contexte de configuration dans laquelle <Global> se trouve. Cela est donc très pratique lorsque l'on a défini des <VirtualHost> car nous n'aurons pas à redéfinir plusieurs fois les mêmes paramètres.

Le mieux est de passer directement à un exemple de [contexte de configuration](#) <Global> :

Exemple de contexte de configuration Global

```
<Global>
DefaultRoot      ~
AllowOverwrite   yes
MaxClients       3
MaxClientsPerHost 1
UseFtpUsers      on
AllowForeignAddress on
ServerIdent      on      "ProFTP DuF's Server Ready"
AccessGrantMsg   "Bienvenue %u sur le serveur du DuF"
</Global>
```

Explications de ce qui est à l'intérieur de <Global>

DefaultRoot => Limite le user à son home directory, si son home directory est par exemple /home/user, il pourra se ballader dedans, mais ne pourra remonter plus haut, il ne pourra pas aller dans /home par exemple et quand il se connecte, le user voit comme path dans son client FTP le chemin /

AllowOverwrite => Cela permet de remplacer d'anciens fichiers par les nouveaux, option inutile si vous interdisez l'écriture. J'indique différentes possibilités pour l'option, mais c'est à vous d'être cohérent. De toutes façons, si vous interdisez l'écriture, cette option ne prendra pas le dessus, vous ne pourrez pas écraser les fichiers.

MaxClients => C'est pour dire le nombre de clients différents qui peuvent se connecter en même temps sur le serveur, si vous avez une connexion ADSL, pas la peine de mettre 50...

MaxClientsPerHost => Option que je trouve très utile, elle limite le nombre de clients pour la même personne, si vous utilisez l'option MaxClients, il faut forcément que MaxClientsPerHost soit strictement inférieur (ou <=) à MaxClients sinon cela ne sert à rien.

UseFtpUsers => C'est dire que l'on utilise ou non le fichier /etc/ftpusers pour savoir qui a le droit d'utiliser le service FTP. Par défaut proftpd utilise le fichier, donc l'option n'est pas utile si on la met à on, mais moi j'ai préféré la mettre, c'est un choix ;-)

AllowForeignAdress => Alors cette option sert à autoriser ou non le fait que quelqu'un envoie ou télécharge des fichiers sur notre serveur FTP depuis un autre ordinateur que le sien. Pour faire simple, on va dire que la personne A veut transférer des fichiers entre le serveur B et notre serveur C car A n'a pas de serveur mais il a accès à B. Sans cette option mise à on cela n'est pas possible que A puisse passer les commandes.

ServerIdent => Cette option permet d'indiquer quel sera le premier message affiché quand quelqu'un essaiera de se connecter sur notre serveur, et cela même si sa connexion échoue. Si vous mettez cette option à "off" le client verra le message suivant : "[hostname] FTP server ready.". Le hostname sera souvent localhost.localdomain si vous ne l'avez pas modifié. Moi je vous invite à mettre cette option à on et mettre la chaîne de

caractere que vous souhaitez mais qui ne donne pas trop d'indication non plus sur votre serveur. Dans mon exemple j'ai mis un message explicite, mais c'est juste un exemple, un message comme "Server Ready" sera tout aussi bien.

`AccessGrantMsg` => C'est là que vous définissez le message d'accueil lorsque la connexion a réussie, donc si vous le mettez dans un [contexte de configuration](#) `<Global>` pas la peine de le mettre à un autre endroit (server config, virtual host....)

Voilà pour la partie Globale, avec déjà toutes ces infos, vous êtes en mesure de partager grâce à votre serveur FTP des données en ayant bien le contrôle de ce qui se passe. Et surtout vous pouvez donner l'accès à des données qui sont sur des partitions FAT32 (mais aussi n'importe quel type de partition ext2, reiserfs etc...), partitions qui normalement vous empêchent de définir une stratégie utilisateur, car si vous avez besoin d'écrire sur des partitions FAT32, et donc que vous les montez en lecture/écriture, vous seriez embêté lors de l'accès par FTP car tout le monde pourrait écrire, supprimer, créer, faire ce qu'il veut en somme sur ces partitions, ce que pas grand monde souhaite. Donc grâce au [contexte de configuration](#) `<Limit>` des commandes, vous empêchez que l'on puisse toucher à vos données (autrement qu'en lecture) ce qui est intéressant pour ceux qui ont encore un multiboot.

Maintenant vous vous dites mais j'aimerais quand même qu'une personne puisse accéder en écriture chez moi, même sur une partition ext2, mais vous dites que maintenant ce n'est plus possible, car on ne peut plus passer les commandes comme MKD, STOR, DELE.... Et bien trompez vous, nous allons créer un `VirtualHost`, terme que certains doivent connaître car c'est le même principe pour le serveur Web Apache.

Contexte de configuration `VirtualHost`

Maintenant on sait comment marche le fichier, donc dans `<VirtualHost>` le principe va être le même que pour `Global` etc... On va définir des options à l'intérieur du [contexte de configuration](#) `<VirtualHost>`.

Premièrement il faut savoir que `<VirtualHost>` à la base est prévu pour un serveur par exemple qui pourra être accessible d'un côté par des personnes s'y connectant depuis internet et d'autres qui s'y connecteront depuis le réseau local et l'on souhaite que ceux qui s'y connectent depuis internet n'aient pas accès aux mêmes données que ceux qui s'y connectent depuis le réseau local. Donc ces différents personnes vont se connecter sur le serveur en indiquant une adresse différente. Par exemple l'utilisateur A est chez lui sur internet, il veut se connecter, pour cela il indique l'adresse suivante : `ftp.serveur_proftpd_internet.com`

Une autre personne, B, va elle se connecter sur ce serveur depuis le lieu de travail, donc depuis le réseau local, elle va utiliser l'adresse interne qui sera l'adresse : `ftp.serveur_proftpd_local.com`

On a donc `ftp.serveur_proftpd_local.com` qui dirige vers l'adresse IP 192.168.10.10 et le `ftp.serveur_proftpd_internet.com` qui dirige vers l'adresse IP 159.159.159.159. Ces 2 adresses IP dirigeant vers la même machine. Il est possible de spécifier le même port (ou de ne pas le spécifier, le port sera celui défini dans le [contexte de configuration](#) `server config`, cela ne posant pas de problème car proftpd va regarder dans son `<VirtualHost>` et tout se fera par rapport à l'adresse qui aura été indiquée pour la connexion.

Il est aussi possible de faire plusieurs `<VirtualHost>` qui travaillent sur la même adresse IP mais qui ont un port différent ce qui peut être très pratique aussi. Mais attention, cela d'après le site www.proftpd.org est incompatible avec "ServerType inetd" c'est à dire lorsque l'on lance le serveur par le démon inetd, personnellement je n'ai pas testé donc je ne pourrai pas vous en dire plus, moi je l'ai juste testé en "ServerType standalone"

Exemple de `<VirtualHost>` :

```
# Serveur Virtuel pour écriture
<VirtualHost ftp.serveur_proftpd.com>
ServerName      "Mon serveur FTP virtuel"
Port            46000
Maxclients      3
MaxClientsPerHost 1
DefaultRoot     ~
AccessGrantMsg  "Bienvenue %u sur le serveur virtuel du DuF"
<Limit LOGIN>
  AllowUser     ToTo
  DenyAll
</Limit>
</VirtualHost>
```

Bon alors les options vous les connaissez toutes, par contre je vais expliquer le [contexte de configuration](#) `<Limit LOGIN>` :

Avec ce contexte de configuration on va indiquer quel(s) utilisateur(s) va(ont) pouvoir se connecter dans ce virtual Host, c'est à dire que ceux qui ont accès par exemple au service FTP en lecture par une connexion sur le port 45000 ne peuvent pas se connecter sur ce virtualhost, dans notre exemple, il n'y a que l'utilisateur `ToTo` qui puisse le faire.

Donc pour que `ToTo` se connecte il doit indiquer l'adresse IP `ftp.serveur_proftpd.com` et le port 46000, ce sont les deux conditions à remplir, sinon cela ne marchera pas. Il faut savoir donc qu'il y a un ordre entre `AllowUser` et `DenyAll` à ne pas négliger. Mais il est possible de l'inverser.

En fait ProFTPD va examiner les autorisations explicites, puis les interdictions. Si une connexion ne correspond à aucun des critères, elle est **autorisée**. Il est possible d'inverser cela en utilisant la commande `Order deny,allow`. Si elle est présente, le serveur va d'abord prendre en compte les commandes `Deny`, suivi des commandes `Allow`, et interdire toute autre connexion ce qui peut être pratique, car dans ce cas là, si jamais votre règle n'est pas infallible ou que vous avez oublié une possibilité, et bien vous êtes sûrs que de toute façon la connexion sera refusée.

A noter aussi qu'il existe des options comme `AllowGroup` etc.... je vous invite à visiter le site de proftpd pour plus de renseignements.

Sinon juste pour l'exemple, un autre virtualhost, dans le cas où le serveur est en type `standalone` et qu'on a donc sur la même IP plusieurs virtualhost

mais sur des ports différents.

```
<VirtualHost ftp.serveur_proftpd.com>
  ServerName      "Mon serveur FTP virtuel"
  Port            47000
  MaxClients      3
  MaxClientsPerHost 1
  DefaultRoot     ~
  <Limit LOGIN>
    AllowUser     Foo
    DenyAll
  </Limit>
</VirtualHost>
```

Les autres contextes de configuration

Contextes de configuration : `<Anonymous>` et `<Directory>`

Le [contexte de configuration](#) `<Anonymous>` comme son nom l'indique sert à configurer un accès anonyme au service FTP et le [contexte de configuration](#) `<Directory>`, permet de définir un contexte pour les répertoires, il est possible de les utiliser comme suit :

```
<Anonymous /home/ftp>
MaxClients 5 "Nombre de clients maximum atteints : 5"
User ftp
Group ftp
<Limit WRITE>
  DenyAll
</Limit>
<Directory uploads/>
<Limit READ>
  DenyAll
</Limit>
<Limit STOR>
  AllowAll
</Limit>
</Directory>
</Anonymous>
```

Suite à une remarque de Drinou, voici une précision sur le contexte de configuration `<Anonymous>` et surtout sur la directive `UserAlias`, si jamais vous voulez que la connexion anonyme se fasse avec un user qui n'existe pas sur votre système, vous devez utiliser les `UserAlias`, vous pouvez vérifier votre configuration avec l'exemple suivant :

```
<Anonymous /home/e-smith/files/primary/html/download>
Group public
User public
UserAlias anonymous public
UserAlias ftp public
AnonRequirePassword off
MaxClients 10
<Limit WRITE>
  DenyAll
</Limit>
<Directory upload/*>
<Limit READ>
  DenyAll
</Limit>
<Limit STOR>
  Allow All
</Limit>
</Directory>
</Anonymous>
```

Donc dans cette exemple nous voyons que les users `anonymous` et `ftp` n'existant pas sous notre OS sont "aliasés" avec le user `public` qui lui existe réellement. Cela permet donc aux clients de se connecter avec le compte `anonymous` sans que celui-ci n'existe réellement sur le système.

Contexte de configuration : `.ftppass`

Pour ce dernier je n'aurai pas d'info à vous donner n'ayant pas trouvé son utilité et la manière de s'en servir (surtout que je n'en ai pas eu besoin...)

Complément sur la configuration de proftpd

Filtrage par Adresse IP

Il faut savoir qu'il est possible de filtrer pas adresse IP, cela est pratique dans un réseau local à IP fixe ou lorsque le client a une IP fixe, mais je ne saurai que trop vous déconseiller de mettre une filtre sur un nom de domaine, ou un redirecteur pour des raisons évidentes de sécurité même si cela peut paraître une solution de facilité.

Voici un exemple de filtrage par adresse IP sur une IP (172.16.18.5) et une classe d'adresse IP (192.168.10.x) :

```
<Limit LOGIN>
  Allow 172.16.18.5 192.168.10.
  Deny all
</Limit>
```

Gestion de la Bande Passante

Depuis la version 1.2.8 de proftpd, la gestion de la Bande Passante n'est plus la même. Auparavant on utilisant des directives comme `RateReadBPS` et `RateWriteBPS` notamment (il y en avait d'autres), maintenant il existe en fait une seule directive (`TransferRate`) qui sert à la fois à définir l'upload et le download par exemple.

Voici un exemple de gestion de la Bande Passante avant la version 1.2.8 :

```
(.....)
MaxClientsPerHost 1
RateReadBPS 12000
RateWriteBPS 63000
(.....)
```

A noter que la valeur était défini en octets, mais maintenant cela a changé, depuis la version 1.2.8 c'est `TransferRate` qu'il faut utiliser. Plutôt que de parler longuement, voici un exemple comment l'utiliser :

```
(.....)
MaxClientsPerHost 1
TransferRate RETR 12
TransferRate APPE,STOR 63
(.....)
```

Pour essayer de faire clair, en fait les 2 exemples font la même chose, le premier dans le cas des versions strictement inférieures à proftpd-1.2.8 et dans le second exemple, c'est pour les versions supérieures ou égales à la version proftpd-1.2.8. Donc RETR signifie Retrieve, ce qui correspond au fait de "récupérer" un fichier depuis le serveur, donc c'est le cas lorsqu'un utilisateur download. Pour APPE et STOR cela correspond à append et store, ce qui correspond au fait de "résumer" et "enregistrer" un fichier sur le serveur. Vous remarquez aussi que maintenant la valeur est en KiloOctets, et sachez que cette directive est valable dans tous les contextes de configuration.

Il faut noter que `TransferRate` ajoute d'autres options très intéressantes, comme le fait de d'allouer un seuil d'octets transférés avant que le contrôle du taux de transfert soit appliqué. Cela permet pour les clients transférant de petit fichier de ne pas être touché, mais ceux qui transfèrent de gros fichiers d'être limités pour donner la priorité à ceux qui transfèrent les petits fichiers. En gros ceux qui vont transférer des fichiers textes ne seront pas contrôlés à l'inverse de ceux transférant des fichiers de type iso par exemple. N'ayant pas testé je ne peux vous dire concrètement si le transfert est stoppé ou seulement limité. Il est aussi possible de créer des groupes d'utilisateurs et de définir des limites de transferts pour ces groupes seulement. Cela permet de limite la BP pour certains mais pas par exemple l'administrateur, ce qui évite de faire plusieurs contextes de configuration.

Chiffrement des transferts

Cette partie est totalement optionnelle et non nécessaire pour la plupart d'entre nous. Donc ici il va être question de chiffrer les communications (commandes, transferts...) lors de la session FTP, mais il ne s'agit pas réellement de crypter de bout en bout la connexion avec SSL, car dans l'exemple suivant il n'y aura pas d'échanges de certificats (clés) entre le client et le serveur, mais seulement authentifier, lister et transférer le tout de manière chiffré.

Première étape, créer les certificats, pour cela je vous renvoi vers d'autres articles que vous pourrez trouver en cherchant avec [google](#) ou alors directement en lisant le [How-TO SSL Certificates](#). Une fois que vous avez créer votre certificat (le fichier .pem) nous pouvons configurer proftpd.

Tout d'abord il faut indiquer dans le contexte de configuration "Server Config" le type du protocole TLS ainsi que le chemin vers le certificat et aussi ajouter la section concernant le module `mod_tls`. Voici en exemple le type de configuration que vous devez avoir :

```
(...)
TLSProtocol          SSLv23
TLSRSACertificateFile /home/proftpd/certs/cert-rsa-duf.pem
TLSRSACertificateKeyFile /home/proftpd/certs/cert-rsa-duf.pem
TLSCACertificateFile /home/proftpd/certs/cert-rsa-duf.pem
(...)
<IfModule mod_tls.c>
  TLSEngine          on
  TLSLog             /var/log/proftpd-tls.log

# Are clients required to use FTP over TLS when talking to this server?
TLSRequired         on
</IfModule>
```

Là nous avons donc spécifier les différents chemins nécessaires vers le certificat, le protocole utilisé, l'activation du moteur TLS, le chemin vers le log spécifique au `mod_tls` et à quel type de requête le client doit répondre.

- `TLSProtocol` : Cela sert à définir quelle version de protocole `mod-tls` doit utiliser. A noter que cette directive ne peut être utiliser que dans le contexte de configuration 'Server Config'. Les choix possibles ici sont TLSv1 (autorise seulement TLSv1), SSLv3 (autorise seulement SSLv3) et SSLv23 qui autorise les deux (SSLv3 et TLSv1).
- `TLSEngine` : Cela permet d'activer ou non le `mod_tls` avec "on" pour "actif" et "off" pour "inactif" naturellement. Par défaut il est désactivé à la fois pour le serveur principal et les virtualhosts.
- `TLSLog` : Chemin vers le fichier log spécifique au `mod-tls`.
- `TLSRequired` : Cela sert à définir une politique sécuritaire basique, si il est à "ctrl" alors SSL/TLS est requis sur le canal de contrôle, si il est à "data" alors SSL/TLS est requis sur le canal de données (transfert des données), si il est à "on" alors SSL/TLS est requis sur les 2 canaux (données et contrôle) et si il est à "off" alors SSL/TLS n'est requis sur aucun des 2 canaux.

Voilà pour la partie configuration par défaut dans le contexte `Server Config`, maintenant si vous avez en plus des virtualhosts et que vous voulez faire des configurations spécifiques, par admettons que le `mod_tls` soit désactivé pour l'ensemble du serveur, mais que vous voulez juste l'activer pour un virtualhost précis, vous aurez alors une configuration proche de celle qui suit :

```
(...)
<VirtualHost ftp.duf.tls.com>
  ServerName      "FTP duf"
  Port            6240
  MaxClients      2
  MaxClientsPerHost 1
  TransferRate    APPE,STOR 63
  AllowForeignAddress on
  AllowStoreRestart on
  DefaultRoot     ~

  TLSRSACertificateFile /home/proftpd/certs/cert-rsa-duf.pem

<IfModule mod_tls.c>
  TLSEngine      off
  TLSLog         /var/log/proftpd-tls.log

</IfModule>

<Limit LOGIN>
  AllowUser      user_tls
  DenyAll
</Limit>
<LIMIT MKD RNFR RNTD DELE RMD STOR WRITE SITE XCUP>
  AllowAll
</Limit>
</VirtualHost>
```

Vous remarquez donc que la valeur de `TLSProtocol` n'est pas indiqué dans le contexte "virtualhost" car comme j'ai dit sa place est exclusivement dans le contexte "Server Config". Sinon comme vous pouvez le constater, afin de chiffrer une communication FTP, cela ne demande pas une grosse configuration dans le fichier `proftpd.conf`.

Attention, cette partie est très très succincte sur le sujet, et à moins que vous ne soyez pas déjà familier avec la création de certificats, l'utilisation de `mod_tls` avec apache par exemple, cette partie sera très grandement insuffisante. Donc ne vous aventurez dans cette configuration que si vous vous y connaissez déjà un minimum et documentez-vous au maximum avec les liens fournis tout en bas de cette page. De même un lien est fourni pour obtenir une liste de client permettant de se connecter sur un serveur avec le `mod_tls` activé.

Exemple de fichier proftpd.conf

Pour finir sur le fichier `proftpd.conf` voici en grande partie le mien modifié :) .

Je rappelles que moi je cherchais à la base (il y a longtemps maintenant :)) pouvoir donner accès par service FTP à des ressources sur partition de type FAT32 et cela à la fois en lecture pour certains et en écriture pour d'autres et que je me suis vu confronté au problème qu'il n'est pas possible de définir de droits pour les utilisateurs sur les partitions FAT32.

Exemple de fichier `proftpd.conf` :

```
# This is a basic ProFTPD configuration file (rename it to # 'proftpd.conf' for actual use. It establishes a single server # It assumes that you have a user/group # "nobody" for normal operation.
```

```
ServerName      "ProFTPD Linux DuF Service"
ServerType      standalone
DefaultServer   on
```

```
# Pour autoriser les clients à résumer les téléchargements, très utile. # Remember to set to off if you have an incoming ftp for upload.
AllowStoreRestart on
```



```
# Port 21 is the standard FTP port.
Port          45000

# Umask 022 is a good standard umask to prevent new dirs and files # from being group and world writable.
Umask        022

# Limitation de la bande passante en lecture :
TransferRate  RETR 11

# To prevent DoS attacks, set the maximum number of child processes # to 30. If you need to allow more than 30 concurrent connections # at once,
simply increase this value. Note that this ONLY works # in standalone mode, in inetd mode you should use an inetd server # that allows you to limit
maximum number of processes per service # (such as xinetd)
MaxInstances  30

# Set the user and group that the server normally runs at.
User          nobody
Group         nogroup

# Nombre maximum de clients
#MaxClients   3

# Number of Max Clients per host
# MaxClientsPerHost  1

# Nombre maximums de tentatives de login
MaxLoginAttempts  3

# Message d'accueil après une connexion réussie
AccessGrantMsg  "Bienvenue %u chez moi !"

# Pour ne pas donner d'info sur le serveur
DeferWelcome    on

#Regles pour limiter les commandes...
<Limit MKD RNFR RNT0 DELE RMD STOR CHMOD SITE_CHMOD SITE XCUP WRITE XRMD PWD XPWD>
  DenyAll
</Limit>

<Global>
DefaultRoot    ~
AllowOverwrite yes
MaxClients     3
MaxClientsPerHost  1
UseFtpUsers    on
AllowForeignAddress on
ServerIdent    on "ProFTP DuF's Server Ready"
AccessGrantMsg "Bienvenue %u sur le serveur du DuF"
</Global>

# Serveur Virtuel pour écriture
<VirtualHost ftp.duf.com>
ServerName     "Mon serveur FTP virtuel numero 1"
Port           46000
Maxclients    3
MaxClientsPerHost  1
DefaultRoot    ~
AccessGrantMsg "Bienvenue %u sur le serveur virtuel du DuF"
<Limit LOGIN>
  AllowUser    ToTo
  DenyAll
</Limit>
</VirtualHost>

<VirtualHost ftp.duf.com>
ServerName     "Mon serveur FTP virtuel numero 2"
Port           47000
MaxClients    3
MaxClientsPerHost  1
DefaultRoot    ~
<Limit LOGIN>
  AllowUser    Foo
  DenyAll
</Limit>
</VirtualHost>
```

Voilà pour ce qui est du fichier `proftpd.conf`.

Utilisation de proftpd

Alors là c'est plutôt simple, deux cas de figure ! Soit vous avez indiqué dans le fichier `proftpd.conf` que le serveur démarre en `standalone` ou par `inetd`. Si c'est en `inetd` il faut relancer le démon `inetd` ou `xinetd` suivant votre distribution. Pour le redémarrer faites :

```
/etc/rc.d/init.d/xinetd restart ; ftp localhost
```

Si votre serveur est en `standalone` vous faites simplement :

```
/etc/init.d/proftpd start
```

Sachez que vous pouvez passer les commandes `start`, `restart`, `stop`, `status`... à `/etc/init.d/proftpd`.

Sinon pour ceux qui veulent utiliser `proftpd` avec `xinetd`, voilà une marche à suivre :

il faut copier le fichier `ftp.d` vers `proftpd` en faisant :

```
cp /etc/xinet.d/wu-ftpd /etc/xinet.d/proftpd
```

Ensuite il faut éditer le fichier `/etc/xinet.d/proftpd`, comme l'exemple suivant :

```
service ftp
{
  disable = no
  flags = REUSE
  socket_type = stream
  instances = 10
  wait = no
  protocol = tcp
  user = root
  server = /usr/local/sbin/in.proftpd
}
```

Cette partie là je ne l'ai pas testé, donc si vous avez un problème renseignez-vous sur l'utilisation de `xinetd`, personnellement je n'en sais pas plus.

Problèmes rencontrés

Lors de l'installation si vous rencontrez un problème de librairie, avant de vous jeter sur votre navigateur web et de les chercher, il se peut que cela vienne juste du fait de ne pas avoir `/usr/local/lib` dans votre chemin des librairies, ce qui fut mon cas. Dans ce cas ajoutez-le donc dans votre `/etc/ld.so.conf` et une fois `ld.so.conf` édité faites un `ldconfig -v`.

Si vous n'avez pas pris `proftpd` en package `rpm/debian` mais en source, et bien les chemins sont différents, le chemin d'installation est `/usr/local` et le fichier de configuration se retrouve dans `/usr/local/etc/proftpd.conf`.

"J'ai des erreurs quand je veux installer `proftpd`, il me manque plein de choses..." Comme je l'ai dit, lisez les messages, ils sont très explicites, allez récupérer ce qui vous manque, n'essayez pas de forcer quoi que ce soit, ce ne sera jamais bon pour votre système de faire ça, et vous risqueriez de mordre la queue en somme.

Vous êtes sûr d'avoir correctement créé vos utilisateurs mais ils ne peuvent pas se connecter, l'accès au service FTP leur est interdit : vérifiez qu'ils ne soient pas dans la liste du fichier `/etc/ftpusers`.

Vos utilisateurs ne sont pas dans le fichier `/etc/ftpusers` mais ils ne peuvent toujours pas se connecter, ils ont une erreur d'accès refusé au password, cela peut venir soit d'un mot de passe erroné, ou de leur répertoire home (leur path où ils sont censés se connecter et arriver) qui est soit invalide, ou même non existant sur votre système.

Conclusion

Je rappelle que cet article n'est pas la solution unique à l'utilisation de `proftpd`, surtout pas. Si vous voulez donner accès à des centaines d'utilisateurs, ayez une stratégie d'ensemble et utilisez les paramètres liés aux systèmes de fichiers, les permissions etc... Ce sera beaucoup plus simple à gérer et à mettre en place. Je le répète, cet article vous donne une idée des informations nécessaires à la configuration de `proftpd` et donne à mon avis toutes les informations nécessaires à la mise en place d'un serveur personnel pour le partage de ressources qui ne sont pas forcément toutes sur des partitions `ext2` (linux). Voilà j'espère que cet article vous aura aidé !

Si jamais vous avez des précisions, des corrections à ajouter sur la partie concernant le `mod_tls` n'hésitez pas car j'ai juste fait une présentation très très légère du sujet, très incomplète et qui mérite d'être améliorée, alors si vous êtes expert en la matière, n'hésitez pas à corriger, améliorer cette partie.

Ressources

<http://www.proftpd.org> (Vous y trouverez toutes les informations nécessaires et surtout toutes les directives y sont listées et expliquées)
<http://www.webring-adsl.com>



<http://www.ze-linux.org/>
<http://frlinux.net/index.php>

Ressources concernant le `mod-tls` :

http://www.castaglia.org/proftpd/modules/mod_tls.html
<http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html>
<http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html> Clients ftp ayant le support SSL/TLS

