Proftpd

Master 2 I2L

Année 2009/2010

D'après la version de Jean-Christophe Soulié (2007-2008)

D. Duvivier
LIL – Université du Littoral Côte d'Opale
soulie@lil.univ-littoral.fr

1 Installation

```
aptitude install proftpd proftpd-doc
```

Lors de l'installation, lancer `proftpd` depuis `inetd` suffit pour nos tests (trafic faible).

Remarque : « `apt-cache search proftpd` » liste plusieurs outils utilisables pour compléter ou faciliter l'installation de `proftpd` ; vous pouvez notamment inclure le paquet proftpd-doc (→ la documentation est placée dans /usr/share/doc/proftpd-doc/).

2 Configuration

La configuration de `proftpd` est placée dans le répertoire `/etc/proftpd`. Il faut focaliser votre attention sur le fichier `/etc/proftpd/proftpd.conf`.
C'est dans ce fichier que vous allez presque tout définir 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 partie).

3 Les utilisateurs

Attention, tous les utilisateurs se connectant sur le serveur `proftpd` doivent exister réellement sur le système (avec un `uid`). Noter aussi que dans cet exemple, nous avons fait le choix de créer les utilisateurs avec un « faux shell » plutôt que de faire des alias.
Avec `proftpd` nous avons le choix dans sa stratégie. Nous avons choisi d'utiliser le fichier `/etc/ftpusers` pour définir tous les utilisateurs qui n'ont pas accès au service FTP. Tous les utilisateurs présents 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 suivante dans le fichier de configuration `/etc/proftpd/proftpd.conf` :

```
UseFtpUsers     on
```

Cependant elle est active par défaut. Vous pouvez donc vous servir de `/etc/ftpusers` sans le dire explicitement dans le fichier `proftpd.conf`, via la directive `UseFtpUsers` donc.
Dans notre cas, nous avons utilisé le fichier `/etc/ftpusers` comme il était en ajoutant tous les utilisateurs qui ont un shell sur la machine. Comment faire ? Facile :

```
cat /etc/passwd | grep –v '/bin/false$' | cut –d: -f 1
```

ATTENTION : n'oubliez pas d'ajouter les users au fur et à mesure des installations (SGBD…).
Copiez le résultat dans `/etc/ftpusers` après avoir effectué une copie de sécurité du fichier de configuration via une commande `cp -i /etc/ftpusers /etc/ftpusers.orig`

Voici un exemple de fichier `/etc/ftpusers` :

```
root
bin
daemon
adm
lp
sync
mail
news
uucp
games
squid
dhcpd
...
nobody
anonymous
```

Remarquez que l'utilisateur `anonymous` est indiqué. Dans ce cas, il n'y a pas d'accès anonyme. Par conséquent, si vous n'avez pas utilité d'un accès anonyme, il est préférable d'ajouter, vous aussi,

anonymous dans le fichier /etc/ftpusers. Pour ce TP, comme indiqué ci-avant, nous avons choisi que tous les utilisateurs ayant un shell (ksh, bash...) n'aient pas accès au serveur FTP. Il faut donc créer des utilisateurs spécifiquement pour l'utilisation du FTP. Or par défaut seuls les utilisateurs disposant d'un shell peuvent se connecter (directive RequireValidShell placée sur « on » par défaut dans /etc/proftpd/proftpd.conf, voir les commentaires dans ce fichier). Ces utilisateurs que nous allons créer n'auront pas de shell leur permettant de se connecter en telnet (du genre telnet 127.0.0.1 21 ☹), etc... Pour cela, il faut indiquer le shell suivant :

```
/bin/false
```

Pour que ce shell soit « valide », deux solutions sont possibles :

- 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.
  J'y suis personnellement opposé car cela revient à considérer que – <u>pour toutes les applications</u> – le shell « /bin/false » est valide ☹ !
- une autre possibilité consiste à placer la directive RequireValidShell sur « off » dans /etc/proftpd/proftpd.conf, ce qui n'est pas satisfaisant car nous ne considérons plus exclusivement les shells listés dans /etc/shells. Cependant nous avons limité les accès via /etc/ftpusers. Et il y a moins d'effets de bord sur la sécurité que la première solution.

Pour la création des utilisateurs, vous pouvez faire un adduser (ou useradd) pour plus d'options c'est très simple, il faut faire man adduser. Notez qu'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 :

```
addgroup ftptest
adduser --shell /bin/false --no-create-home --ingroup ftptest user1
adduser --shell /bin/false --no-create-home --ingroup ftptest user2
```

Vérifiez les informations relatives aux utilisateurs créés :

`ls -l /home`	→ Pas de répertoire dans /home
`cat /etc/group`	→ Groupe ftpusers bien créé
`cat /etc/passwd`	→ Utilisateurs créés dans le groupe ftpusers avec shell=/bin/false
`cat /etc/shadow`	→ Utilisateurs créés avec un mot de passe « valide » et « actif »

Par contre les utilisateurs user1 et user2 ne peuvent pas se logguer (essayez !) vous pouvez vérifier avec les commandes suivantes :

`su – user1`	→ Je tente de « déguiser » root en user1...
`whoami`	→ Je suis toujours root !

Au passage vérifiez le mode de protection (=700) pour /root et /home/<votre_répertoire_home>.

Voilà déjà pour la partie spécifique aux utilisateurs (users), nous allons en reparler dans la partie du fichier proftpd.conf.

4 Le fichier Proftpd.conf

Contexte de configuration Server Config

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

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

Donc pour résumer 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 profptd.conf
UseFtpUsers off|on
#fin du fichier proftpd.conf d'exemple, pour info ce fichier en l'état n'est pas valide
```

On peut dire que toutes les directives sont forcément incluses 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 I2L"
ServerType inetd
DefaultServer on
```

`ServerName` → je ne reviens pas dessus

`ServerType` → est important (indispensable)

- S'il est suivi du mot-clef « `standalone` » il indique que le serveur est démarré par vos soins en faisant : `/etc/init.d/proftpd start`
- S'il est suivi par le mot-clef « `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 quelle 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 sera déconnecté.

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

```
AllowStoreRestart on
Port 21
Umask 022 022
MaxInstances 30
User proftpd
Group nogroup
```

`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, c'est bien évidemment le numéro de port sur lequel le client se connecte (21, 45000 …).

`Umask` → c'est comme dans unix en général, la première valeur pour les fichiers et la seconde pour les répertoires. `022` est une valeur qui fonctionne bien (interdire aux utilisateurs non-propriétaires de modifier les fichiers, ce masque complémenté à 1 transforme un mode 0666 en 0644 et un 0777 en 0755), donc laissez-là à `022`. Si vous voulez en savoir plus sur `umask` en général, consultez le man.

`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, en mode standalone (et non en mode inetd). Comme indiqué, au-delà d'une valeur de 30 vous être vulnérable à des attaques de type Ddos, donc laissez à 30 (ou moins), pour une utilisation, sincèrement c'est largement suffisant.

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

Le `user proftpd` et le `group nogroup` sont les paramètres par défaut et à mon avis ils sont très bien, donc on n'y touche pas pour limiter les privilèges de l'utilisateur qui lance le serveur.

D. Duvivier : Mon illustre prédécesseur (J.-C. Soulié) indiquait ceci pour la version 2007/2008 :

Ensuite il est question de l'option `PersistentPasswd`, dont l'utilité ici n'est pas évidente à expliquer étant donner que je n'ai pas compris l'explication sur le site de `proftpd`, je l'ai laissé à off. En fait si je me trompe 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.

D. Duvivier : pour la version 2008/2009 :

Je précise ce qui suit… Par défaut cette option est sur « on ». En fait, il faut placer cette option sur « off » si vous utilisez NIS ou LDAP pour récupérer les mots de passe, sinon proftpd garde des descripteurs de fichiers ouverts sur les fichiers /etc/passwd et /etc/group en supposant une authentification « locale » (*cf.* la doc).

Ensuite vient une option qui permet de limiter le nombre de tentatives de logins :

```
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 moi...."
```

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'information précise sur le serveur, il est préférable de mettre à `on` l'option suivante de manière à ne diffuser cette information qu'aux utilisateurs authentifiés :

```
DeferWelcome on
```

Contexte de configuration `<Limit>`

Maintenant on va indiquer un `Limit` qui va s'appliquer au `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 cette commande est utile (très importante) 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 on peut utiliser le Limit avec les directives suivantes :

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

MKD : création de répertoire
RNFR : (rename from) empêche de pouvoir renommer
RNTO : (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 consultez les commandes de la section Limit sur le site de www.proftpd.org. Ou en utilisant la documentation disponible via l'URL locale file:///usr/share/doc/proftpd-doc/Configuration.html#LIMIT

Contexte de configuration Global
Là on arrive à une section relativement importante, le <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 <Global> :

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

DefaultRoot → Limite le user à son home directory, si son home directory est par exemple /home/user, il pourra se balader 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 /
D. Duvivier pour la version 2009/2010 :
Si vous créez les utilisateurs « ftp » sans répertoire d'accueil, comme je l'ai fait (option --no-create-home de adduser), il faut spécifier ceci :

```
DefaultRoot /home/ftp
```

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 toute façon, 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.
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 caractères 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 <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 empêche 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 <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.

Contextes de configuration : <Anonymous> et <Directory>

<Anonymous> comme son nom l'indique sert à configurer un accès anonyme au service FTP et <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>
```

Complément sur la configuration de proftpd

Filtrage par Adresse IP

Il faut savoir qu'il est possible de filtrer par 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 un 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 `ReadWriteBPS` 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 définie 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.

Pour tester :

```
touch /home/ftp/vide.txt
ftp 127.0.0.1
```

Entrez le nom d'un utilisateur valide (user1 ou user2 dans notre exemple) et le mot de passe associé, vous devez obtenir le prompt d'accueil « ftp> », vous pouvez visualiser le contenu du répertoire d'accueil (ici /home/ftp pour notre configuration) en tapant la commande « ls » :

```
ftp> ls
```

Ce qui va lister le contenu du répertoire (il y a normalement deux fichiers : welcome.msg et vide.txt). Utilisez la commande « quit » pour quitter.

Testez maintenant les connexions sur le serveur FTP « de votre voisin », en supposant que son adresse IP soit 192.168.0.10, voici la commande à taper:

```
ftp 192.168.0.10
```

Entrez le nom d'un utilisateur valide (user1 ou user2 dans notre exemple) et le mot de passe associé, vous devez obtenir le prompt d'accueil « ftp> », vous pouvez visualiser le contenu du répertoire d'accueil (ici /home/ftp pour notre configuration) en tapant la commande « ls » :

```
ftp> ls
```

Ce qui va lister le contenu du répertoire (il y a normalement deux fichiers : welcome.msg et vide.txt). Utilisez la commande « quit » pour quitter.

En cas de problème :

- allez voir du côté du répertoire /var/log/proftpd

Pour information, allez-donc voir les pages du manuel a/s les commandes suivantes :

- ftpwho, ftpcount, ftpstats, ftpasswd, ftpdctl, ftptop, ftpshut

Voyez les commandes suivantes :

- ftpquota --help
- netstat –apv | grep ftp