Proftpd

Serveur FTP Proftpd avec utilisation de mod_radius pour l'authentification et ainsi utiliser des comptes LDAP disponible sur un serveur Active Directory.

Cf. Radius pour des informations complémentaires sur ce service.

Généralités

ProFTPd est un serveur FTP libre. Ses auteurs l'annoncent comme puissant et parfaitement sécurisé sur le site web dédié au logiciel.

Il est distribué selon les termes de la licence GNU GPL.

Ses partisans disent que ProFTPd est bien documenté et que la plupart des configurations seront proches de celles des exemples fournis avec le logiciel. Son unique fichier de configuration, proftpd.conf, utilise une syntaxe similaire à celle d'Apache permettant ainsi d'homogénéiser les fichiers de configuration. La directive include permet cependant de répartir les directives de configuration dans différents fichiers pour les cas plus complexes.

Le logiciel permet de configurer plusieurs serveurs FTP virtuels et a la possibilité d'être utilisé dans un environnement dédié (chroot). Il peut être lancé comme un démon ou comme service inetd. Enfin, ProFTPd est compatible IPv6.

Son architecture est modulaire, ce qui a permis d'écrire des extensions pour le support de la cryptographie SSL/TLS (protocole FTPS) et l'extension de l'authentification via des bases RADIUS, LDAP ou SQL.

Le module SQL permet, en outre, le stockage en base de données des opérations effectuées sur le serveur FTP.

La version 1.3.2 apporte également, via le module mod_sftp, le support des protocoles SFTP et SCP issus de SSH.

Installation

Ce serveur est installable par la commande rpm après l’avoir obtenu sur les sites habituels comme dag-wieers.com. Le problème ici est que nous utilisons une authentification Radius associé à une authentification Kerberos sur l’AD. Ceci permet une gestion centralisée des comptes FTP et de sécuriser les accès.

Le module `mod_radius` de proftpd n’est pas facile à obtenir directement et surtout il n’est pas intégré dans le service de base. Il faut l’ajouter à partir des sources. J'avais utilisé un paquetage rpm réalisé à la main avec le module intégré. Il était disponible pour une architecture 32 bits.

Dans le cas où on n'utilise pas Radius, utiliser le rpm disponible avec la version la plus récente.

Pour l'installer, commencer par enlever vsftpd si il est présent.

```
rpm -e vsftpd
```

Ensuite installer proftpd avec :

```
rpm -ivh proftpd...i386.rpm
```

Configuration du service proftpd

- Liste des directives

Le fichier `proftpd.conf` est le seul fichier du serveur FTP qu'on doit modifier, ensuite, tout se passe sur le serveur Radius afin de permettre l'authentification. Récupérer le fichier `proftpd.conf` d'une configuration existante par exemple ou copier la configuration qui suit.

Ce fichier de configuration est adapté à une utilisation avec le module `mod_radius` forcément compilé car non disponible par défaut. J'ai utilisé le paquet rpm de Proftpd version 1.1 créé par un personne avec le module `mod_radius` inclus.

On considère deux serveurs, un Radius et le serveur qui dispose du service FTP avec ce fichier de conf.

- Serveur Radius : 172.20.15.1
- Serveur Proftpd : 172.20.15.80

Le compte alice et adminbob sont des comptes créé sur l'annuaire LDAP. Dans mon cas c'était Active Directory.

```
# This is the ProFTPD configuration file
# $Id: proftpd.conf,v 1.1 2004/02/26 17:54:30 thias Exp $

ServerType                      inetd

# Ne pas renvoyer des ident ou des resolutions DNS en reponse.
# (Timeouts en cas de filtrage sur ces ports).
IdentLookups                    off
UseReverseDNS                   off

# Emplacement du fichier "tableau de bord" du processus.
# Cette directive en necessaire pour que la limitation du nombre de clients
simultanes
# fonctionne correctement.
ScoreboardFile                  /var/run/proftpd.scoreboard

# Utilise la RFC 2228 pour les reponses multilignes au lieu de la RFC 959.
# En principe meilleure compatibilite avec certains clients.
MultilineRFC2228                on

# nombre maximum de connections en une seconde.
```

```
MaxConnectionRate               10

# Format des logs du serveur.
LogFormat                       default "%h %l %u %t \"%r\" %s %b"
LogFormat                       auth    "%v [%P] %h %t \"%r\" %s"
LogFormat                       write   "%h %l %u %t \"%r\" %s %b"

SyslogLevel                     warn

port                            0

<IfModule mod_cap.c>
        CapabilitiesEngine on
        CapabilitiesSet +CAP_CHOWN
</IfModule>


<Global>

        # Repertoire de chroot (et definition des groupes utilisateurs
autorises).
        # Ceci empeche tout utilisateur a remonter au dessus du repertoire
specifie.
        # La fonction definissant les groupes autorises n est pas utilisee
ici mais
        # definie dans la LIMIT LOGIN de chaque serveur virtuel
        DefaultRoot                     /data/Uploads

        # Utilisation du mecanisme PAM pour l authentification des
utilisateurs
        AuthPAM                         off
        RadiusEngine                    on
        RadiusAcctServer                172.20.15.2:1813 KMR32L64ZXX31
        RadiusAuthServer                172.20.15.2:1812 KMR32L64ZXX31
        RadiusUserInfo 501 100 /data/Uploads /sbin/nologin

        # limitation du nombre de clients simultanes.
        MaxClients      20      "Désolé, nombre maximum de clients autorisés
atteint (%m)"

        # utilisation du fichier /etc/ftpusers contenant les logins de
utilisateurs
        # interdits de FTP.
        # Cette technique n est pas retenue dans cette configuration ou la
limitation est
        # imposee par un LIMIT LOGIN dans le contexte "server config" qui
permet de ne
        # specifier que les utilisateurs autorises ( la liste est plus
courte ).
        USeFtpUsers                     off
```

```
        # nombre maximum de tentatives de login avant deconnexion.
        MaxLoginAttempts                3

        # nombre maximal de connexion clientes par hote client.
        MaxClientsPerHost               5

        # Autorise la reprise de connexion dans les recuperations de
fichiers sur le serveur.
        AllowRetrieveRestart            on

        # Autorise la reprise de connexion dans les transferts de fichiers
vers le serveur.
        AllowStoreRestart               on

        # Interdit a root tout acces au FTP pour des questions de securite.
        RootLogin                       off
        RootRevoke                      off
        # Demande un mot de passe meme si l identifiant seul suffirait a ne
pas autoriser la
        # connexion. Ceci evite de donner une information supplementaire a
un attaquant eventuel.
        LoginPasswordPrompt             on

        # Interdit des commandes PORT de transfert de fichiers entre le
serveur et le client
        # si ce dernier n est pas la source IP de ce fichier. (Empeche les
"bounce attack").
        AllowForeignAddress             off

        # Utilisateur et groupe sous l identite duquel le processus du
serveur FTP tournera.
        User                            root
        Group                           root

        # Seule directive de securite du serveur exterieure a ce fichier de
configuration:
        # le nombre maximum de processus enfants possible pour ce service.
Cette limitation
        # n existe que pour un serveur en mode standalone (MaxInstances). En
cas de service
        # gere par un super-demon (inetd ou xinetd), elle devient
inopérante. Par contre
        # xinetd permet la definition du nombre d instances maximum dans sa
configuration
        # ce qui permet de realiser de cette maniere cette limitation.

        # Autorizer l ecrasement de fichiers deja existants.
        AllowOverwrite                yes

        # Default to show dot files in directory listings
        ListOptions                     "-a"
```

```
        # Umask 117 pour les fichiers (rw-rw----) et 007 pour les dossiers
(rwxrwx---) pour avoir
        # les droits a l utilisateur et au groupe  et les autres sans aucun
droit
        Umask                           0117 0007

</Global>

<Virtualhost 172.20.15.80>

ServerName                      "Service FTP du serveur
SERVERFTP.DOMAINE.LOCAL"
Port                            21
ServerIdent                     on "Bienvenue sur le serveur administratif
du SERVEURFTP.DOMAINE.LOCAL"
DefaultRoot                     /data/Uploads
ServerAdmin                     root@localhost
AccessGrantMsg                  "Utilisateur %u: Acces Autorisé."
DeferWelcome                    on

# Utilisateurs autorises a s authentifier sur ce serveur FTP virtuel.
# Filtrage par username. Seule L'authentification Active Directory est
autorisee
<Limit LOGIN>
order                           allow,deny
AllowUser                       alice
AllowUser                       adminbob
DenyAll
</Limit>
        <Directory /data/Uploads/*>
                <Limit READ DIRS WRITE CHMOD SITE_CHMOD>
                        AllowAll
                </Limit>
                UserOwner uploaduser
                GroupOwner users
        </Directory>
allowOverwrite                  yes
</VirtualHost>

# pas d acces anonyme sinon il aurait fallu creer un block de configuration
Anonymous.
# Fin de la configuration.
```

La clé partagée doit être sensiblement la même dans les deux directives suivantes que celle qui est présente dans le fichier `/etc/raddb/clients.conf` de la configuration du serveur Radius.

```
        RadiusAcctServer                172.20.15.1:1813 KMR32L64ZXX31
        RadiusAuthServer                172.20.15.1:1812 KMR32L64ZXX31
```

Cf. explication dans la partie Radius. Fix Me! mettre référence.

Créer le compte uploaduser membre du groupe users nécessaire au fonctionnement du serveur FTP au niveau Linux. En fait, on peut se connecter avec n'importe quel compte sur le serveur FTP à partir du moment ou on l'a configuré dans le radius et que le compte existe sur l'AD mais un seul compte est utilisé au niveau Linux : uploaduser

Créer les dossiers nécessaires aux scripts qui utilisent ce serveur FTP pour déposer les fichiers.

```
mkdir /data/Uploads
mkdir /data/Uploads/proxy
```

```
groupadd users
useradd -g users -d /data/Uploads -s /sbin/nologin uploaduser
```

Récupérer les données si toutefois il y en a et les positionner dans les bons dossiers. J'entends par bon dossier les dossiers utilisés par les scripts de sauvegarde.

Positionnement des droits concernant les services qui sont sauvegardés par le biais du serveur FTP. Pour ceux qui utilisent TFTP cf. la partie dédié à cet effet.

```
chown -R uploaduser:users /data/Uploads
chmod 700 /data/Uploads
chmod 755 /data/Uploads/proxy
chmod 666 /data/Uploads/backup
```

Le serveur proftpd est configuré pour fonctionner en xinetd. Ceci s'oppose au mode standalone. Le serveur FTP n'est en fait lancé que lorsqu'il sert (xinetd). Le reste du temps, il n'est pas visible dans la liste des processus avec un ps -ef. En mode standalone, le service est toujours un processus.

```
vi /etc/xinetd.d/xproftpd
```

Il faut faire passer la valeur disable à no au lieu de yes pour activer le service.

```
service ftp
{
        disable                 = no
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/in.proftpd
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        nice                    = 10
}
```

Redémarrer le service xinetd pour prendre en compte la modification.

```
service xinetd restart
```

Configuration Radius

Aller sur le serveur Radius, présent ici sur le serveur 172.20.15.1.

Il y a 3 fichiers à modifier.

Scripts	Fonction
/etc/raddb/users	Définition de l'utilisateur ou d'un groupe d'utilisateur qui peuvent se connecter.
/etc/raddb/clients.conf	Liaison de ce fichier avec le mod_radius notamment avec le secret partagé.
/etc/raddb/huntgroups	Définition de l'adresse IP qui peut se connecter.

`/etc/raddb/users`

```
# login pour serveur FTP pour administratif web infodb et autres
DEFAULT                 Huntgroup-Name == "ftp_server_lan",Auth-Type :=
Kerberos
```

Fichier pour ajouter des utilisateurs qui seront acceptés. Ils sont acceptés à travers une authentification kerberos si le client Radius (ici notre serveur srv-in03) fait partie du huntgroup (cf. fichier suivant) « ftp_server_lan ». Ce groupe contient l'adresse IP de notre serveur. Ce fichier n'est pas à modifier si on utilise les mêmes comptes qu'un autre serveur identique.

`/etc/raddb/clients.conf`

```
client 172.20.15.80 {
        secret          = KMR32L64ZXX31
        shortname       = SERVEUR
        nastype     = other
}
```

C'est ce fichier qui fait le lien avec le module radius du serveur proftpd. On y trouve l'IP et le secret partagé de notre serveur.

La clé partagée est une clé présente de manière identique sur les deux machines. Sur le serveur proftpd elle est positionnée dans le fichier de configuration /etc/proftpd.conf et sur le serveur radius, c'est dans le fichier /etc/raddb/clients.conf. IL FAUT VEILLER A CE QU'ELLE SOIT IDENTIQUE SUR LES DEUX FICHIERS

`/etc/raddb/huntgroups`

```
#SERVEUR
ftp_server_lan          NAS-IP-Address == 172.20.15.80
```

Fichier qui contient l'association d'un groupe aux adresses IP que nous voulons.

Recharger la configuration du serveur Radius pour ne pas perturber la production avec un : service radiusd reload

Tuning performance proftpd

- http://www.proftpd.org/docs/howto/BCP.html

Resource Limits

One of the most common requests on the mailing list is to be able to limit the number of connections, in various ways, a given user may make to the proftpd daemon. There are different configuration directives for doing so, depending on the situation:

```
MaxInstances: Limits the overall number of connections. Default: none
MaxClients: Limits the number of connections on a per-server/vhost basis.
Default: none
MaxClientsPerHost: Limits the number of clients that may be connecting from
the same host. Default: none
MaxClientsPerUser: Limits the number of clients that may be logged in at one
time using the same username. Default: none
MaxHostsPerUser: Limits the number of hosts from which clients may be logged
in at one time using the same username. Default: none
```

Intensive use of CPU and/or memory by FTP sessions can be restricted by use of RLimitCPU and RLimitMemory. Here's an example:

```
RLimitCPU session 10
RLimitMemory session 4096
```

This applies CPU and memory resource limits to session processes. In order to limit the use of such resources by the daemon, should this be a concern, the directives can be used similarly:

```
RLimitMemory daemon 8192 max
```

Valeur déjà testé sur d'autres serveurs.

```
# nombre maximum de connections en une seconde.
MaxConnectionRate               10
# limitation du nombre de clients simultanes.
MaxClients      20      "Désolé, nombre maximum de clients autorisés atteint
(%m)"
# nombre maximum de tentatives de login avant deconnexion.
MaxLoginAttempts                3
# nombre maximal de connexion clientes par hote client.
MaxClientsPerHost               5
# Autorise la reprise de connexion dans les recuperations de fichiers sur le
serveur.
AllowRetrieveRestart            on
# Autorise la reprise de connexion dans les transferts de fichiers vers le
serveur.
AllowStoreRestart               on
```

Performance Tuning

When configuring a proftpd for performance, here are some settings to try. First, make sure that you have ident and reverse DNS lookups disabled:

```
IdentLookups off
UseReverseDNS off
```

Directive Log

- Log management.

```
ExtendedLog
LogFormat
WtmpLog : on
TransferLog : /var/log/xferlog
SyslogFacility
tcpBackLog
SyslogLevel
ServerLog
RewriteLog
```

Pour différencier les logs proftpd des logs systèmes (par défaut dans `messages`), ajouter l'option suivante à l'extérieur de la directive Global.

```
SystemLog /var/log/proftpd.log (pour ne pas utiliser /var/log/messages par défaut.
```

Test FTP

- Cf. FTP.