Leçon 11 : Le service FTP

11.1 Introduction

Le service FTP est un service complémentaire au service Web puisqu'il permet aux utilisateurs de charger et déposer des fichiers sur espace configuré (par exemple des fichiers *html* ou *php*). Il est parfois proposé comme moyen de backup par les entreprises vendant de l'espace web. Il existe de nombreux programmes serveurs FTP sous Linux. Ainsi les plus connus sont : *proftpd, vsftpd, wu-ftpd*. Sous Windows, d'autres logiciels peuvent remplir ce rôle comme *Microsoft FTP (intégrant IIS)* ou *FileZilla FTP Server*.

Il faut cependant être très prudent avec le service FTP afin :

- D'autoriser uniquement les utilisateurs souhaités à se connecter au service
- Limiter les dossiers et fichiers visibles

La distribution *CentOS 7* livre en standard le logiciel *vsftpd* que nous allons découvrir dans cette leçon.

11.2 FTP et la sécurité

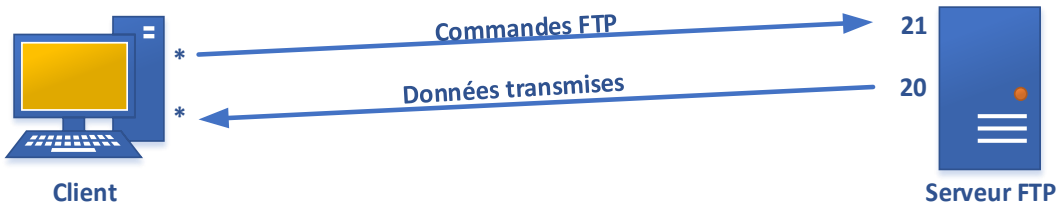
Le service FTP est un service qu'il faut tenir à l'œil car il peut être utilisé pour accéder ou découvrir des informations sensibles. Ainsi, il est possible de *scripter* des tentatives de connexion au serveur et découvrir des comptes (login et mot de passe) valides. Par conséquent, des mesures additionnelles sont souvent mises en place pour sécuriser celui-ci :

- Autoriser uniquement certains comptes à accéder au service FTP (et systématiquement bannir des logins connus : bin, daemon, root, admin, …).
- Forcer les utilisateurs à utiliser un mot de passe **fort** (composé d'au moins 10 caractères comprenant majuscules, minuscules, chiffres et caractères spéciaux) et **non issu du dictionnaire** (beaucoup d'utilisateurs choisissent des mots courants dans leur mot de passe).
- Protéger la phase d'authentification par SSL afin d'imposer un échange chiffré du login / mot de passe
- Utiliser des techniques de sécurité avancées pour limiter les tentatives de connexion

11.3 FTP Actif et FTP Passif

Le service FTP supporte 2 modes de fonctionnement : le mode **actif** et le mode **passif**. Historiquement, le mode **actif** est celui qui était présent dès le départ, cependant, avec le déploiement du NAT et des firewalls, le mode **passif** s'est vite répandu. Aujourd'hui, c'est le mode de fonctionnement par défaut. Nous allons détailler ces deux modes de fonctionnement.

10.3.1 Le mode actif

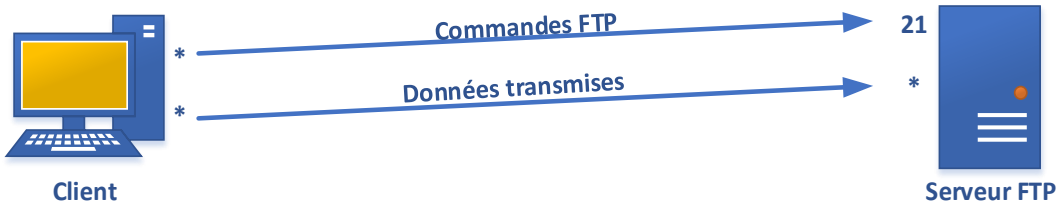


En mode actif, le client se connecte au serveur FTP sur le port 21 (couramment utilisé à cet usage). Ce canal de communication est utilisé pour transmettre les commandes FTP du client vers le serveur (transmission des requêtes). Lorsque des données doivent être transmises depuis le serveur (liste des fichiers / dossiers, contenu d'un fichier, …), celui-ci **ouvre une nouvelle connexion vers le client**.

Or le problème se situe ici surtout depuis la généralisation du NAT et des firewalls du coté client. En effet, le firewall (par exemple celui contenu dans la box VDSL) devrait se souvenir de qu'une connexion a été demandée par le client et autoriser la connexion – sur n'importe quel port – depuis ce serveur.

Autant dire que ce mode de fonctionnement n'est pas du tout utilisable aujourd'hui. Il reste cependant parfois présent à l'intérieur des entreprises ou des réseaux internes.

11.3.2 Le mode passif



Le mode passif fonctionne un peu différemment. Comme nous pouvons le voir, le client lorsqu'il doit envoyer une commande au serveur, il ouvre un canal de connexion sur le port 21 (couramment utilisé à cet usage). Lorsque le client souhaite transmettre ou recevoir des données depuis le serveur, il ouvre un second canal vers le serveur.

Ainsi, il n'y a plus de connexion du serveur vers le client, ce qui simplifie beaucoup la gestion de la sécurité chez le client. Cependant, nous avons reporté le problème du coté serveur : en effet, le canal d'échange de donnée doit arriver jusqu'au serveur FTP sur n'importe quel port, lui aussi protégé par un firewall.

Afin de sortir de ce problème, la solution proposée est que le serveur FTP peut suggérer au client le port que celui-ci peut utiliser pour ouvrir le canal de transmission des données. Grâce à cet élément de configuration, l'administrateur du firewall coté serveur doit simplement : *ouvrir le port 21* pour le canal de commande et *ouvrir une série de ports (range)* pour le canal de transmission (par exemple : les ports TCP entre 15000 et 15500).

Ouverture de ports dans pfSense

Pour ouvrir des ports dans pfSense, il faut se connecter à l'interface web de configuration du firewall et puis aller dans le menu **Firewall > NAT**.

Nous allons devoir ajouter 2 règles[35] pour permettre la connexion sur le serveur FTP depuis le réseau de l'école :

3. Interface : **WAN** ; Protocol : **TCP** ; Destination : **WAN address** ; Destination port range from : **(other) 15000** to **(other) 15500** ; Redirect target IP : **<ip de votre serveur Linux>** ; Redirect target port : **15000** ; Cliquer sur SAVE

4. Interface : **WAN** ; Protocol : **TCP** ; Destination : **WAN address** ; Destination port range from : **FTP** to **FTP** ; Redirect target IP : **<ip de votre serveur Linux>** ; Redirect target port : **FTP** ; Cliquer sur SAVE

   Cliquer sur **Apply Changes**

   Le service FTP devrait, quand il sera configuré et activé, être accessible depuis l'extérieur du réseau virtuel.

11.4 Configuration du service FTP

La configuration du service *vsftpd* est concentrée dans le fichier `/etc/vsftpd/vsftpd.conf`. Ce fichier texte permet de préciser les options souhaitées. Nous allons, dans la suite, pointer quelques options importantes[36] :

- `anonymous_enable=`**`YES`**`|NO`
  Permet d'autoriser ou interdire la connexion en mode anonyme sur le serveur. Le mode anonyme est le mode de connexion par défaut lorsqu'on entre une URL ftp://ftp.monsite.be dans son navigateur (connexion avec l'utilisateur *anonymous* ou *ftp* sans vérification du mot de passe). Cet utilisateur ne doit pas avec un compte sur le système. Dans CentOS 7, la connexion avec ce compte donne accès au dossier `/var/ftp`.

- `local_enable=YES|`**`NO`**
  Permet d'autoriser ou interdire la connexion des utilisateurs existants sur le système (i.e. disposant d'un compte). Si l'authentification réussit, l'utilisateur est automatiquement connecté dans son dossier personnel dans `/home/login`.

- `write_enable=YES|`**`NO`**
  Permet d'autoriser ou interdire l'écriture (dépôt d'un fichier, création d'un dossier, modification des permissions, …). Attention, les permissions UNIX (droits, ACLs) doivent également permettre l'opération pour que celle-ci soit autorisée.

- `local_umask=022` umask par défaut : `local_umask=077`
  Détermine le masque appliqué sur les permissions. Cette option est utilisée lors du dépôt d'un fichier ou la création d'un dossier afin de savoir quelle permission UNIX définir sur le fichier / dossier déposé. Ainsi, les permissions `777 – 022`, soit `755` seront appliquées.

[35] Les ports proposés sont donnés à titre d'exemple (21 et ceux compris entre 15000 et 15500)
[36] En **gras** : la valeur par défaut si l'option est absente du fichier de configuration. En rouge, la valeur présente dans le fichier `vsftpd.conf` à l'installation.

- `anon_upload_enable=YES|`**NO**
  `anon_mkdir_write_enable=YES|`**NO**
  Détermine si un utilisateur non authentifié (en mode anonyme) peut déposer des fichiers ou créer des dossiers. ***Ces deux options devraient toujours être désactivées !***

- `dirmessage_enable=YES|`**NO**
  Permet d'activer ou non la prise en charge des fichiers .message présents dans les dossiers. Le contenu du fichier texte est alors envoyé au client FTP qui peut décider de l'afficher.

- `connect_from_port_20=YES|`**NO**
  Impose que la commande FTP PORT émane du port 20 du serveur. Ce comportement est complètement obsolète mais certains clients primitifs peuvent l'exiger.

- `ftpd_banner=GodSwila FTP Serveur version 2025`
  Permet de changer l'annonce par défaut du serveur. Cette option est utile pour cacher le type de serveur et la version qui est installée.

- `chroot_local_user=YES|`**NO**
  `chroot_list_enable=YES|`**NO**
  `chroot_list_file=/etc/vsftpd/chroot_list`
  `allow_writeable_chroot=YES|`**NO**
  L'option `chroot_local_user` permet d'emprisonner l'utilisateur qui se connecte dans son dossier personnel. En conséquence, il ne peut pas en sortir et naviguer sur tous les dossiers accessibles du système. Cette option devrait toujours être activée. Afin d'augmenter la flexibilité dans la configuration du serveur FTP, il est possible de déterminer modifier quelque peu ce comportement :

chroot_local_user	chroot_list_enable	Explication
YES	NO	Tous les utilisateurs sont emprisonnés dans leurs dossiers personnels
YES	YES	Tous les utilisateurs sont emprisonnés dans leurs dossiers personnels excepté ceux listés dans le fichier pointé par `chroot_list_file`
NO	YES	Seuls les utilisateurs listés dans le fichier pointé par `chroot_list_file` sont emprisonnés dans leurs dossiers personnels.

  Lorsqu'on active l'option `allow_writeable_chroot`, *vsftpd* ne vérifie pas si l'utilisateur à le droit d'écrire dans son dossier personnel. A activer en cas d'erreur du type « *refusing to run with writable root inside chroot()* ».

- `listen=YES|`**NO**
  Permet de déterminer si le serveur FTP fonctionne en mode *standalone*. Le mode *standalone* devrait toujours être utilisé (soit via `listen=YES` ou `listen_ipv6=YES`). Cette option démarre l'écoute uniquement sur les adresses IPv4 de la machine. **Elle ne peut être utilisée en même temps que `listen_ipv6`.**

- `listen_ipv6=YES|NO`
  Permet de déterminer si le serveur FTP fonctionne en mode *standalone*. Le mode *standalone* devrait toujours être utilisé (soit via `listen_ipv6=YES` ou `listen =YES`). Cette option démarre l'écoute sur les adresses IPv4 et IPv6 de la machine (pour autant que l'adresse d'écoute ne soit pas modifiée). **Elle ne peut être utilisée en même temps que `listen`.**
- `pam_service_name=vsftpd`
  Cette option détermine le nom sous lequel les modules PAM (pluggable authentication modules) reconnaissent le serveur FTP. Grâce à ceux-ci, il est possible de diversifier les méthodes d'authentification (comme utilisé des mots de passe à usage unique, …).
- `userlist_enable=YES|NO`
  `userlist_deny=YES|NO`
  `userlist_file=/etc/vsftpd/user_list`
  Permet de contrôler les utilisateurs autorisés à se connecter au serveur. Ainsi, nous avons les combinaisons possibles pour ces options :

userlist_enable	userlist_deny	Explication
YES	YES	Tous les utilisateurs connus sont autorisés à se connecter sur le serveur excepté ceux listés dans le fichier pointé par `userlist_file`.
YES	NO	Seuls les utilisateurs connus et listés dans le fichier pointé par `userlist_list_file` peuvent se connecter sur le serveur.
NO	-	Tous les utilisateurs connus peuvent se connecter sur le serveur FTP.

- `tcp_wrappers=YES|NO`
  Permet de vérifier que les connexions sont vérifiées en tenant compte des fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Il est possible d'autoriser par ces fichiers les connexions par adresse IP, préfixe IP, … Cette méthode de sécurisation est souvent trop limitée par rapport aux possibilités des firewalls.

- `pasv_min_port=borne1`
  `pasv_max_port=borne2`
  `pasv_promiscuous=YES|NO`
  Permet de déterminer, pour le mode passif, les ports qui sont suggérés par le serveur FTP au client pour le transfert des données. Il est nécessaire d'*ouvrir tous les ports compris entre ces 2 bornes* au niveau du firewall. L'option `pasv_promiscuous` doit être activée pour permettre la connexion derrière un système NAT

- `ssl_enable=YES|NO`
  `rsa_cert_file=/path/to/certificate.pem`
  `rsa_private_key_file=/path/to/private_key.pem`
  `ssl_ciphers=HIGH`
  Ces options permettent d'activer le support SSL dans le serveur FTP. Pour ce faire, il faut placer, dans un seul fichier, le certificat du serveur et tous les certificats intermédiaires – par exemple dans `/etc/pki/tls/certs/vsftpd-allcerts.pem` – et faire pointer `rsa_cert_file` vers ce fichier. Il faut également placer la clé privée dans un fichier – par exemple `/etc/pki/tls/private/vsftpd-privatekey.pem` – et faire pointer

`rsa_private_key_file` vers ce fichier. Enfin, il faut également déterminer les algorithmes de chiffrement autorisés, `HIGH` désigne les algorithmes de sécurité élevés.

Une fois la configuration de *vsftpd* terminée, le service peut être démarré au moyen de la commande :

```
$ systemctl start vsftpd
```

En cas d'erreur, il est possible d'obtenir des informations complémentaires via l'argument `status` :

```
$ systemctl status vsftpd
```

Il est parfois plus simple de lancer le service *vsftpd* à la main et voir si un message d'erreur survient :

```
$ vsftpd /etc/vsftpd/vsftpd.conf
```

Quand il n'y a plus d'erreur, on peut alors tuer le processus *vsftpd* lancé à la main et redémarrer le service *vsftpd* normalement :

```
$ killall vsftpd
$ systemctl start vsftpd
```

11.5 Le client FTP

Suivant les systèmes d'exploitation clients utilisés, il existe bon nombre de programme client. Ainsi *FileZilla* est un client FTP connus (et disponible sous Windows, Linux et Mac OS X). Il y a également *FireFTP* qui est une extension pour Mozilla Firefox qui est également un client FTP intéressant et courant.

En ligne de commande, des clients existent également : le programme *ftp* est probablement le client le plus connu (une version très limitée existe aussi sous Windows). Nous découvrirons également *lftp* qui permet de scripter les échanges FTP facilement.

Si nécessaire, l'installation de ces outils se fait au moyen de yum :

```
$ yum install ftp lftp
```

11.5.1 Le programme FTP en ligne de commande

```
[root@localhost pam.d]# ftp
ftp> open ftp.belnet.be
Trying 193.190.67.98...
Connected to ftp.belnet.be (193.190.67.98).
220-Welcome to the Belnet public FTP server ftp.belnet.be !

 This server is located in Brussels, Belgium and operated by Belnet, ...

220 ProFTPD 1.3.4a Server (Belnet FTP Server) [193.190.67.98]
Name (ftp.belnet.be:root): ftp
331 Anonymous login ok, send your complete email address as your password
Password: test@test.com
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
```

```
ftp> ls
227 Entering Passive Mode (193,190,67,98,141,42).
150 Opening ASCII mode data connection for file list
lrw-r--r--   1 ftp      ftp            28 Nov 14  2012 debian ->
mirror/ftp.debian.org/debian
lrw-r--r--   1 ftp      ftp            31 Nov 14  2012 debian-cd ->
mirror/ftp.debian.org/debian-cd
drwxr-xr-x  89 ftp      ftp          4096 Feb 10 08:29 mirror
lrw-r--r--   1 ftp      ftp             7 Nov 10  2012 mirrors -> mirror/
lrw-r--r--   1 ftp      ftp             6 Nov 14  2012 pub -> mirror
226 Transfer complete
ftp> ?
ftp> quit
221 Goodbye.
```

Les commandes FTP autorisées sont nombreuses. Pour les connaître, il faut entrer **?** à l'invite FTP. Parmi les commandes courantes, citons :

- `passive` qui permet d'activer / désactiver le mode passif
- `get` qui permet de télécharger **un** fichier du serveur et de le placer dans le dossier courant
- `put` qui permet de déposer **un** fichier sur le serveur à partir du dossier courant
- `prompt` qui permet d'activer / désactiver le mode interactif. Par défaut, ce mode est activé et demande une confirmation lors de l'utilisation des commandes *multiples mget* et *mput*.
- `mget` qui permet de recevoir **des** fichiers dans le dossier courant. Si le mode interactif est activé, une confirmation sera demandée pour chaque fichier.
- `mput` qui permet de déposer **des** fichiers sur le serveur à partir du dossier courant. Si le mode interactif est activé, une confirmation sera demandée pour chaque fichier.
- `chmod` qui permet de modifier les permissions d'un fichier ou d'un dossier
- `cd` qui permet de changer de répertoire sur le serveur
- `lcd` qui permet de changer le dossier courant (sur le poste client). `lcd` sans paramètre affiche le dossier local dans lequel on se trouve.
- `pwd` qui affiche le répertoire distant dans lequel on se trouve
- `exit` qui permet de fermer la connexion FTP et quitter le client.

11.5.2 Le programme lftp

Le programme *lftp* dispose d'une option intéressante : il est possible de scripter les commandes FTP à envoyer au serveur. Ainsi, le programme peut être utilisé pour planifier et automatiser une sauvegarde par FTP sur un serveur distant.

Ainsi, on peut, par exemple, transférer un fichier automatiquement avec la commande :

```
$ lftp monlogin:monpass@NomOuIPserveur -e "cd dossier; put
/home/swila/monfichier; exit;"
```

Cette commande se connectera au serveur `NomOuIPserveur` (pour lequel on peut mentionner son nom complet ou sont adresse IP) et exécutera, successivement, les commandes `cd`, `put` et `exit` avec les paramètres indiqués.

Il est évidemment possible d'intégrer cette commande dans un script PERL.

11.6 Exercices

On vous demande de :

1. Créer un utilisateur backup. Définir son mot de passe.

2. Configurer votre serveur FTP afin de :
    a. Ne pas autoriser l'accès anonyme
    b. Permettre à l'utilisateur *swila* et *backup* de se connecter au serveur FTP
    c. Permettre à *votre voisin* (pour lequel un compte a été créé lors d'une leçon précédente) de se connecter au serveur FTP
    d. Emprisonnez tous les utilisateurs, excepté l'utilisateur *backup*, dans leurs dossiers personnels
    e. Activer le mode passif avec les ports compris entre `45000` et `45500`
    f. Vérifier votre configuration en vous connectant à votre serveur FTP en local

3. Configurer pfSense pour ouvrir les ports nécessaires au fonctionnement FTP. Vérifier votre configuration en utilisant le poste Windows pour vous connecter (et en utilisant l'adresse IP externe en 192.168.[128-143].x)

4. Créer un script PERL :
    a. qui compresse, sous la forme d'une archive ZIP, le contenu du dossier `/etc`
    b. qui transfert le fichier ZIP créé sur la machine de votre voisin, en utilisant le compte qu'il vous a créé

    Planifiez l'exécution du script PERL tous les semaines, durant les séances de laboratoire (à 10h ou à 15h par exemple).