

Partage de fichiers avec FTP

1. Le protocole FTP

a. Historique

FTP : *File Transfer Protocol* est un protocole client-serveur assez ancien qui fut l'un des premiers à permettre le partage de fichiers entre deux ordinateurs. Il a un passé glorieux, et fut par exemple employé avant la création du protocole SMTP pour transférer les messages électroniques d'un ordinateur à un autre.

Aujourd'hui, son âge et une certaine rigidité le rendent moins apte à un partage de fichiers confortable. Il reste néanmoins très utilisé, notamment par les hébergeurs Internet qui proposent généralement à leurs clients de mettre à jour les sites web hébergés par FTP.

b. Paramètres techniques

FTP est transporté par TCP et fonctionne sur le port 21 pour la transmission des commandes. Le port 20 est historiquement utilisé pour passer les données téléchargées mais ça n'est plus un comportement universel.

FTP supporte l'authentification des clients, mais avec un degré de sécurité faible le rendant inapte au transfert de fichiers sensibles. En effet, FTP est bien connu pour transporter le mot de passe de ses clients en clair sans aucun cryptage. Pour ces raisons, FTP est généralement utilisé aujourd'hui dans un usage spécifique : le mode anonyme. Les serveurs FTP peuvent reconnaître un compte unique anonyme et lui autoriser un accès limité, généralement en lecture seule sur certains répertoires. Le compte doit obligatoirement s'appeler **anonymous**, et le serveur a la possibilité de demander un mot de passe, qui pourra être n'importe quelle suite de caractères. Le mot de passe sera alors conservé pour des raisons de traçabilité même si le client n'a aucune obligation sur ce mot de passe.

c. Mode FTP actif et FTP passif

Historiquement, les clients FTP travaillaient en **mode actif** où la session est établie sur le port 21 du serveur, et où les données sont envoyées depuis le port 20 et à l'initiative du serveur vers un port quelconque du client. Ce fonctionnement qui date d'avant la généralisation des pare-feu ne va pas sans poser de problème dans la mesure où il est vu par le pare-feu comme une session ouverte depuis le serveur sur un port imprévisible du client.

Le **mode passif** est venu corriger cet état de fait en faisant établir les deux sessions par le client. Le port utilisé pour les données est alors quelconque, annoncé par le serveur en mode commande, et utilisé par le client pour l'ouverture de la session de données.

2. Les clients FTP

a. Les clients FTP graphiques

Les clients FTP graphiques sont nombreux et existent pour toutes les plates-formes. On peut citer filezilla qui est un produit open source très populaire sur les systèmes Windows. La configuration et l'usage des clients FTP graphique varient selon les produits et ne présentant pas de difficulté majeure, leur utilisation ne sera pas traitée ici.

b. Le client FTP en lignes de commandes

La plupart des systèmes incluent un client FTP en lignes de commandes. Le mode de fonctionnement de ces clients peut les rendre inconfortables pour un usage fréquent mais ils sont extrêmement pratiques pour tester la configuration d'un serveur FTP.

Le chargement de ces clients se fait le plus simplement du monde par la commande **ftp**.

L'avantage principal du client FTP en ligne de commande est qu'il permet de réaliser toutes les opérations voulues une à une, et donc de comprendre en cas de dysfonctionnement où se situe l'échec. Au contraire, les clients graphiques ont tendance à automatiser un grand nombre d'opérations. Pour une connexion FTP avec Internet Explorer par exemple, la connexion est automatiquement anonyme, et un mot de passe standard est automatiquement envoyé.

open	Ouvre une session FTP vers le serveur donné en référence. Le client demandera interactivement l'adresse du serveur.
close	Ferme une session FTP en cours.
ls	Affiche les fichiers contenus dans le répertoire courant distant.
cd	Change le répertoire courant distant. La syntaxe est la même que dans un shell Linux.
get	Télécharge (récupère) un fichier du répertoire courant distant dans le répertoire courant local.
put	Télécharge (envoie) un fichier du répertoire courant local vers le répertoire courant distant.

3. Le serveur Pure-FTPd

Pure-FTPd est un serveur FTP qui vise à proposer un service de transfert de fichier simple, stable et efficace. Il se veut adapté aussi bien aux débutants qu'aux situations de production en entreprise. Sa principale caractéristique est de pouvoir être lancé facilement en ligne de commande sans s'appuyer sur un fichier de configuration.

a. Fonctionnement pour accès des utilisateurs à leurs répertoires personnels

C'est le fonctionnement par défaut, et les utilisateurs possédant un compte et un répertoire personnel peuvent accéder à leurs données avec leur identifiant et leur mot de passe habituel. Attention, ce mode de fonctionnement est généralement déconseillé dans la mesure où le transit du mot de passe en clair met en danger le mot de passe Linux des utilisateurs.

Lancement du service

```
pure-ftpd
```

b. Fonctionnement en accès anonyme

L'accès anonyme est possible si un compte utilisateur **ftp** a été créé sur le serveur. Les clients connectés en mode anonyme travaillent alors dans le répertoire **/home/ftp**.

Il est possible de travailler en mode anonyme seul en appelant pure-ftpd avec l'option **--anonymously**.

c. Options de fonctionnement

Pure-ftpd fonctionnant généralement sans fichier de configuration, la ligne de commande lançant le service sera enrichie d'options de configuration en fonction du résultat voulu. Certaines implémentations toutefois exploitent un ou plusieurs fichiers de configuration qui sont interprétés par le script de lancement du service. La liste ci-dessous présente certaines des options les plus courantes.

pure-ftpd : options courantes	
--help	Affiche les options possibles.
--displaydotfile	Affiche aussi les fichiers cachés aux clients.
--anonymously	Fonctionnement en serveur anonyme uniquement (si le compte ftp existe).
--noanonymous	Empêche toute connexion anonyme (même si le compte ftp existe).
--maxidletime	Temps maximum d'inactivité avant déconnexion forcée.
--anonymouscantupload	Empêche les utilisateurs anonymes de transférer des fichiers vers le

	serveur.
--anonymouscancreatedirs	Permet aux utilisateurs anonymes de créer des répertoires.