TRANSFERT DE FICHIERS & PROTOCOLE FTP

Auteur: Bernard GIACOMONI
Autoentreprise GIACOMONI Bernard

Version	Date	Objet
1.0	10/05/2019	Version initiale

Table des matières

I.GÉNÉRALITÉS:

I.1.LE PROTOCOLE FTP:

I.1.1.PRINCIPE:

File Transfer Protocol (FTP), est un protocole de communication de la couche 7 de l'ISO (couche application). Il permet d'échanger des fichiers entre les hôtes d'un réseau. Il permet également de supprimer ou de modifier des fichiers ou d'inspecter les répertoires et sous répertoires d'un hôte distant.

Le protocole FTP s'appuie sur la couche transport de TCP/IP. Il possède une variante sécurisée (FTPS) implantée au dessus du protocole SSL (Secure Socket Layer) ou de son successeur TLS (Transport Layer Security).

I.1.2.ARCHITECTURE:

FTP se conforme au modèle CLIENT-SERVEUR: un logiciel CLIENT FTP envoie des requêtes à un logiciel SERVEUR FTP situé sur la même machine ou sur des machines différentes. Le SERVEUR fournit en réponse à cette requête des SERVICES qui peuvent être principalement:

- Le transfert de fichiers entre les deux postes physiques;
- La navigation dans les répertoires d'un des postes physiques et la visualisation de leurs contenus;
- Le renommage ou la suppression de fichiers dans un des postes physiques.

La principale fonction d'un serveur FTP est de rendre accessible à des clients AUTHENTIFIÉS (par Id. et mot de passe), une arborescence de fichiers située dans la machine qui l'héberge. FTP utilise pour cela la couche transport du protocole TCP. Il fonctionne donc en mode CONNECTÉ. De ce fait, seuls peuvent accéder à l'arborescence les utilisateurs disposant d'un CLIENT FTP et possédant un COMPTE CLIENT enregistré sur le serveur.

Par défaut, les connexions FTP utilisent deux ports: le port 21 pour les commandes et le port 20 pour les données (Le mode FTPS dit "implicite" utilise par défaut un seul port: le 990). Le protocole fonctionne en IPv4 ou IPv6.

Un logiciel client FTP peut accéder à un serveur FTP soit par un INTERFACE GRAPHIQUE (par exemple: le client graphique filezilla), soit par l'intermédiaire de COMMANDES EN LIGNE (commandes CMD sous Windows ou Shell linux-unix).

II.INSTALLATION DE FTP:

II.1.INSTALLATION D'UN CLIENT FTP:

Les systèmes d'exploitation récents proposent "nativement" un logiciel client FTP. C'est le cas en particulier de WINDOWS ou des distributions LINUX. Cependant, ces installation natives n'offrent en général qu'une interface en ligne de commande, basée sur la commande "ftp <@IP/nom du serveur>".

Sinon, de nombreuses implémentations gratuites sont disponibles. En particulier:

Produit	Système hôte	Description
Inetutils	linux	Paquets linux à installer, contenant en particulier un client et un serveur FTP en ligne de commande
cURL	Linux/Windows	Client en ligne de commande de Transfert de ressources sur le net (fonctionne avec des URL).
Vsftpd	Linux/Unix	Verisure FTP: client en ligne de commande et serveur FTP (paquets à installer)
Filezilla client	Linux/Windows	Client graphique (FTP, FTPS et SFTP)
Etc….		

II.2.INSTALLATION D'UN SERVEUR FTP:

En revanche, l'installation native d'un serveur FTP est assez rare: en effet, comme tout serveur, un serveur FTP doit pour remplir sa mission rester en permanence à l'écoute sur les interfaces réseaux de sa machine hôte, ce qui consomme une fraction non négligeable des ressources. D'autre part, il doit posséder une IP FIXE. Il est donc contre-productif de l'installer dans un système qui n'est pas dédié à cette fonction.

II.2.1.INSTALLATION SOUS WINDOWS:

Les logiciels gratuits TypoSoft FTP Server et FileZilla Server sont disponibles dans des versions compatibles avec WINDOWS. Leur installation ne pose aucun problème car elle suit la procédure standard d'installation des logiciels sous windows.

Ces deux logiciels proposent une interface graphique complète pour le paramétrage du serveur. L'interface de paramétrage de FileZilla permet d'implémenter le serveur en mode "FTP over TSL", ce qui correspond à FTPS.

II.2.2.INSTALLATION SOUS LINUX:

Le paquet vsftpd permet d'installer sous linux ubuntu ou debian un server FTP sécurisé (Very Securized File Transfert Protocol), grâce à la commande:

```
sudo aptitude install vsftpd
```

Après installation, la configuration du serveur est effectuée en éditant, en mode root, le fichier:

/etc/vsftpd.conf

Quelques exemples de configuration du fichier vsftpd.conf:

- Pour interdire le FTP anonyme (connexion non authentifiée): Changez la ligne: anonymous_enable=YES en anonymous_enable=NO
- Pour autoriser les utilisateurs locaux à se connecter, Ajouter: local_enable=YES (Ceci permet aux utilisateurs déclarés dans Ubuntu d'utiliser le même login/mot de passe pour accéder au serveur FTP.
- Pour Autoriser les utilisateurs à uploader des fichiers; ajouter: write_enable=YES
- Pour choisir l'interface d'écoute: par défaut le serveur FTP est en écoute sur toutes les interfaces réseaux. Il est possible de restreindre l'écoute à une interface en ajoutant par exemple dans le fichier de config: listen_address=<adresse ip lan du serveur>

Dans tous les cas, il faut redémarrer le serveur pour que les modifications soient prises en compte: sudo /etc/init.d/vsftpd restart.

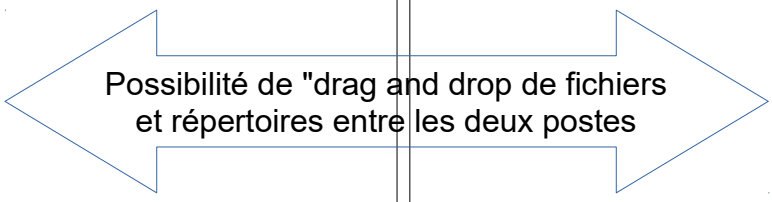
III.UTILISATION DE FTP:

III.1.UTILISATION DES CLIENTS GRAPHIQUES:

A quelques détails près, les clients FTP avec interfaces graphiques offrent tous aux utilisateur une interface ressemblant à ce qui suit:

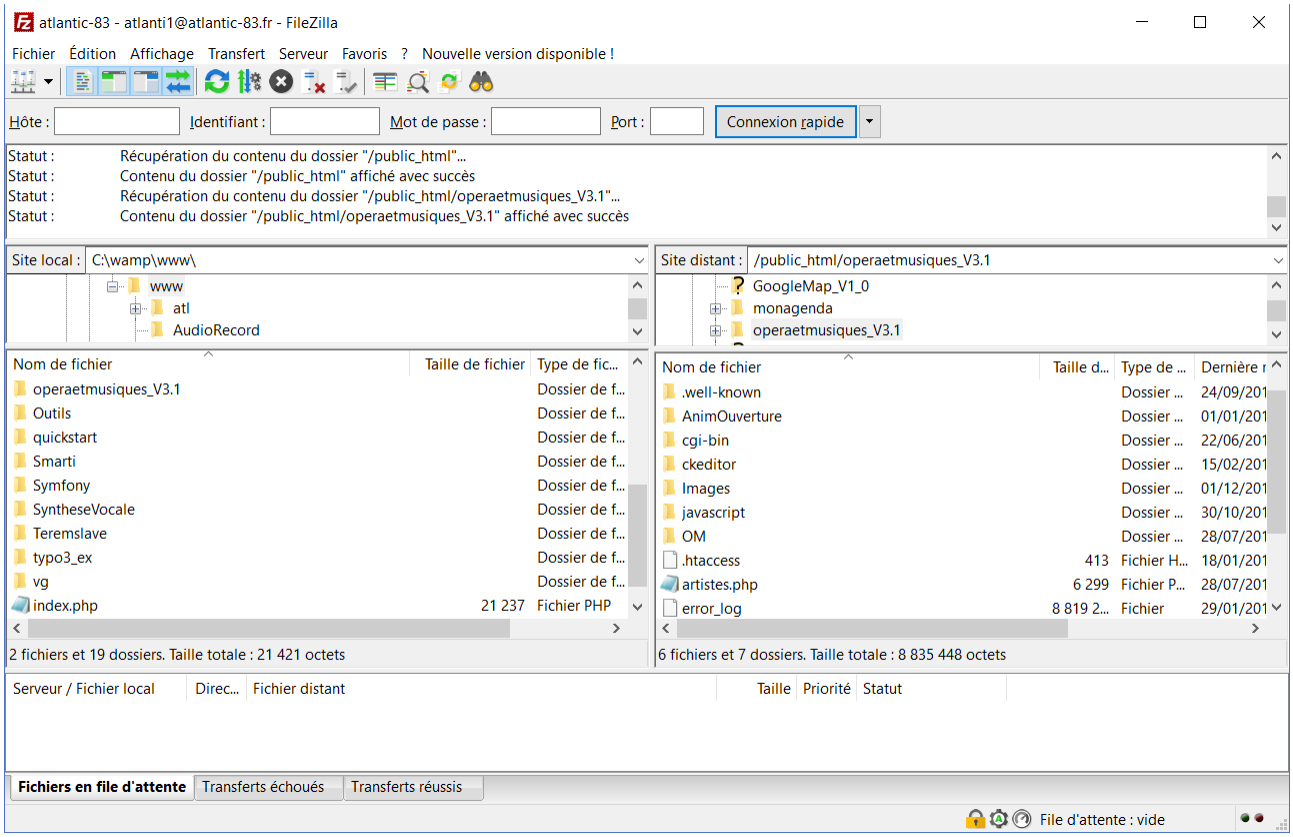
Fenêtre de paramétrage et de connexion au serveur :
- Saisie de l'adresse internet du serveur et du port à utiliser ;
- Saisie des identifiants de l'utilisateur ;
- Commande de connexion/déconnexion)
- Autres commandes...

Fenêtre du client (fenêtre locale)	***Fenêtre du serveur***
Visualisation de l'arborescence des fichiers du client, fonctions de navigation dans cette arborescence, possibilité de renommer ou supprimer des fichiers locaux ou de les transférer dans le serveur (si connecté).	*SI LE CLIENT EST CONNECTÉ :* Visualisation de l'arborescence des fichiers du serveur, fonctions de navigation dans cette arborescence, possibilité de renommer ou supprimer des fichiers du serveur ou de les transférer dans le client (si connecté).



Fenêtre de compte-rendu des commandes
- comptes-rendus déchanges
- etc...

III.2.EXEMPLE: INTERFACE DE FILEZILLA CLIENT:



III.3.UTILISATION DES CLIENTS FTP EN LIGNE DE COMMANDE:

III.3.1.OUVERTURE D'UNE FENÊTRE DE COMMANDE SOUS WINDOWS:

Une fenêtre "console" en mode non administrateur peut être lancée par la commande de démarrage "exécuter→cmd"

III.3.2.OUVERTURE D'UNE FENÊTRE DE COMMANDE SOUS UNIX/LINUX:

- Si l'on dispose d'une installation "graphique", un terminal peut s'ouvrir soit par un "clic droit" sur le bureau, soit en sélectionnant une application "terminal";
- Sinon, la commande xterm permet d'ouvrir une fenêtre terminal en ligne depuis un autre terminal (Attention: le logiciel xterm n'est pas forcément installé dans les installations de base de linux).

III.3.3.PRINCIPALES COMMANDES:

III.3.3.A.OUVERTURE DE SESSION:

La commande: > ftp <nom/@ip du serveur>

permet à un client FTP de se connecter au serveur hébergé par la machine dont le nom ou l'adresse IP sont spécifiés dans la commande.

EXEMPLES*:* > ftp 192.168.1.21, > ftp BUREAU_COMPTA

Le système répond par la demande de saisie des identificateurs d'un client ftp du serveur. Si la connexion au serveur a été acceptée, le prompt "ftp>" apparaît. Les commandes suivantes (commandes ftp) sont alors adressées au serveur.

REMARQUE: Autre façon d'ouvrir une session:

> ftp -n

ftp> open <Nom ou @Ip du serveur>[:<n° de port>]

III.3.3.B.FERMETURE DE SESSION:

Une session ftp se termine par une des commandes ftp> quit ou bye.

III.3.3.C.NAVIGATION DANS LES RÉPERTOIRES DU SERVEUR:

Commande	Effet
ls	Liste les noms des fichiers et sous-répertoires du répertoire courant du serveur (a l'ouverture, un serveur est positionné par défaut sur le répertoire racine de l'utilisateur authentifié)
dir	Même chose, mais les noms des éléments sont accompagnées d'informations diverses (droits d'accès, date et heure de création, volume, etc.).
pwd	Affiche le nom et le chemin d'accès complet du répertoire courant du serveur.
cd	Permet de changer le répertoire courant du serveur: - cd <nom et chemin d'accès> permet de remplacer le répertoire courant par le répertoire spécifié par <nom et chemin d'accès>; - cd .. remplace le répertoire courant par son répertoire parent; - cd ../Rep1 remplace le répertoire courant par le répertoire fils Rep1 de son répertoire parent.

III.3.3.D.NAVIGATION DANS LES RÉPERTOIRES DU CLIENT:

Commande	Effet
!dir	Liste les noms des fichiers et sous-répertoires du répertoire courant du client (répertoire local). Ces informations sont accompagnées d'informations diverses (droits d'accès, date et heure de création, volume, etc.).
lcd	Permet de changer le répertoire courant du client (répertoire local): - lcd <nom et chemin d'accès> permet de remplacer le répertoire courant par le répertoire spécifié par <nom et chemin d'accès>; - lcd .. remplace le répertoire courant par son répertoire parent; - lcd ../Rep1 remplace le répertoire courant par le répertoire fils Rep1 de son répertoire parent.
substitut à wpd en local	En début d'affichage de !dir apparaît le nom et le chemin d'accès du répertoire local courant.

III.3.3.E.CHOIX DU MODE DE TRANSFERT DES FICHIERS:

Commande	Effet
binary	Positionne le mode de transfert sur "binaire": les contenus des fichiers sont alors transmis sous forme binaire. Ceci convient à tous les contenus "non textuels" (images, sons, etc.).
ascii	Positionne le mode de transfert sur "ascii": Les contenus des fichiers sont alors considérés comme des textes. Dans ce cas, certaines données peuvent être transformées lors du transfert pour tenir compte du système d'exploitation du récepteur (par exemple: caractères de fin de lignes). Ceci convient aux fichiers à contenus textuels (documents, fichiers html, sources de logiciles, etc.).

III.3.3.F.TRANSFERT DE FICHIERS:

Commande	Syntaxe	Effet
get	get <distant> [<local>]	Permet de transférer un fichier du serveur vers le client
mget	mget <distant> [<local>]	Permet de transférer plusieurs fichiers du serveur vers le client en utilisant des "wildcards" (caractères "*").
put	put <local> [<distant>]	permet de transférer un fichier du client vers le serveur
mput	mput <local> [<distant>]	permet de transférer plusieurs fichiers du client vers le serveur en utilisant des wildcards (caractères "*"). "wildcards".

REMARQUES:

- Dans le cas de get ou mget, si le deuxième argument de la commande est absent, le ou les fichiers sont transférés dans le répertoire courant du client sans changement de nom;
- Dans le cas de put ou mput, si le deuxième argument de la commande est absent, le ou les fichiers sont transférés dans le répertoire courant du serveur sans changement de nom;

III.3.3.G.CRÉATION/SUPPRESSION DE FICHIERS OU RÉPERTOIRES:

Commande	Syntaxe	Effet
del	del <nom+CA fichier>	Supprime un fichier du serveur
!del	!del <nom+CA fichier>	Supprime un fichier du client (fichier local)
rmdir	rmdir <nom+CA répertoire>	Supprime un répertoire du serveur
!rmdir	!rmdir <nom+CA répertoire>	Supprime un répertoire du client
mkdir	mkdir <nom+ca du répertoire>	Crée un répertoire dans le serveur
!mkdir	!mkdir <nom+ca du répertoire>	Crée un répertoire dans le client

*CA = Chemin d'accès.

IV.AUTOMATISATION DES COMMANDES FTP:

IV.1.INTRODUCTION:

Nous avons, jusqu'à maintenant, abordé l'utilisation des outils FTP en mode INTERACTIF. Cependant, comme répéter plusieurs fois la même série de commandes interactives est évidemment pénible, peu efficace et source d'erreurs, il est intéressant de regrouper des séries de commandes FTP participant à un traitement donné en procédures réutilisables et exécutables automatiquement sans autre interaction humaine que leur lancement (Ceci revient à disposer de l'équivalent des fichiers .bat de l'interpréteur de commandes cmd.exe pour l'interpréteur du client FTP en ligne).

Cette problématique doit être traitée différemment en fonction du système sur lequel le client FTP est implanté.

IV.2.PROCÉDURES DE COMMANDES FTP SOUS WINDOWS:

IV.2.1.L'ATTRIBUT -S:

Les clients FTP en ligne **sous windows** supportent un attribut de la commande "ftp" qui permet de définir une suite de commandes FTP à exécuter, enregistrée dans un fichier de commande. C'est l'option "-s:<nom de fichier commande>". Cette option indique au client FTP qu'il doit utiliser le contenu textuel du fichier spécifié pour se connecter au serveur ftp, puis pour lui envoyer des commandes.

IV.2.2.FORMAT DU FICHIER DE COMMANDES:

Les lignes du fichier de commandes doivent correspondre soit à des commandes ftp, soit aux réponses attendues par les prompts ftp résultant de ces commandes.

Première ligne:

La première ligne du fichier doit être la commande d'ouverture de la session:

open <nom ou @ ip du serveur>

Cette commande va déclencher en retour la demande de saisie du nom d'utilisateur:

Connecté à <nom ou @ip du serveur>
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
Utilisateur (*******:(none)) :

Deuxième ligne:

La deuxième ligne du fichier doit donc être le nom de l'utilisateur. Le client FTP va alors utiliser ce nom pour envoyer une commande user <nom d'utilisateur> au serveur, qui va

répondre en demandant la saisie du mot de passe:

331 Please specify the password.

Mot de passe :

Troisième ligne:

La troisième ligne du fichier doit donc contenir le mot de passe.

Lignes suivantes:

Les lignes suivantes contiendront les commandes ftp que l'on veut effectuer automatiquement (la dernière ligne sera la commande de déconnexion "bye" ou "quit"). Cependant, si l'on veut éviter de perturber l'exécution séquentielle de ces commandes par des prompts en retour de certaines commandes (par exemple, des demandes de confirmation), il est nécessaire de commencer la séquence par la commande "prompt off", qui les désactive.

Contenu type:

Compte tenu de ce qui précède, la forme la plus courante d'un fichier de commandes ftp est donc:

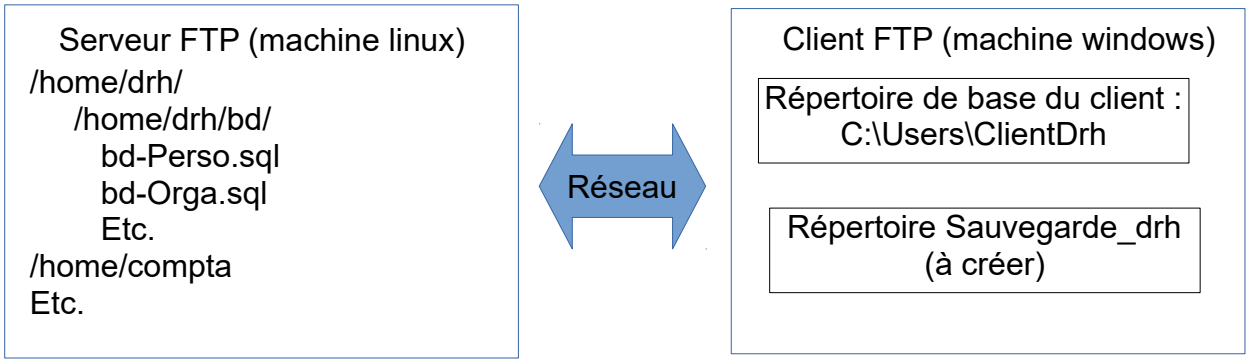
```
open <nom ou @ip du serveur>
<nom d'utilisateur>
<mot de passe>
prompt off
<commande ftp 1>
< - - - - - - - - - - - >
<commande ftp n>
bye (ou quit)
```

Remarques:

- La commande "prompt off" ne doit évidemment pas être située avant la saisie du mot de passe (sinon, l'identification échoue);
- La grande faiblesse de ce procédé est qu'il oblige à libeller les identifiants de connexion dans le texte du fichier de commande. Nous verrons qu'il est possible d'atténuer cet inconvénient;
- Le contenu du fichier ne doit contenir aucun espace inutile en fin de ligne sous peine d'être mal interprété.

IV.2.3.EXEMPLE N°1-SOLUTION NON SÉCURISÉE:

Supposons la configuration suivante:



Nous voulons sauvegarder par une procédure automatique tous les fichiers d'extension .sql du répertoire /home/drh/ du SERVEUR dans un répertoire Sauvegarde_drh que nous allons créer à cet effet dans le répertoire courant du client ftp. Supposons que l'utilisateur Admin du serveur FTP a pour mot de passe %AdmSys%:

Fichier Commande.ftp:

```
1. open 192.168.1.57
2. Admin
3. %AdmSys%
4. prompt off
5. !mkdir .\Sauvegarde_drh_230119
6. lcd ./Sauvegarde_drh_230119
7. cd /home/drh
8. mget *.sql
9. bye
```

Commande d'exécution du fichier:

La commande > ftp -s:Commande.ftp, lancée par l'exécuteur de commandes cmd de Windows provoque l'exécution de la série de commandes ftp contenue dans le fichier texte Commande.ftp.

Détail de l'exécution:

- La commande ftp -s:Commande.ftp utilise d'abord les contenus des trois premières lignes pour se connecter au serveur 192.168.1.57 en tant qu'utilisateur Admin muni du mot de passe%AdmSys%;
- La commande "prompt off" indique alors au client ftp qu'il faut quitter le mode interactif;
- la commande ftp !mkdir .\Sauvegarde_drh_230119 crée le sous-répertoire

Sauvegarde_drh_230119 dans le répertoire courant du CLIENT (ici, le client est sous windows: il faut donc utiliser un anti slash et non un slash dans la commande);

- La commande ftp lcd ./Sauvegarde_drh_230119 choisit pour le répertoire courant du CLIENT le nouveau répertoire;
- La commande ftp cd /home/drh positionne le répertoire courant du SERVEUR sur le sous-répertoire drh de /home;
- La commande mget *.sql ./Sauvegarde_drh_230119 transfère tous les fichiers d'extension sql contenus dans le répertoire courant du serveur dans le répertoire de sauvegarde Sauvegarde_drh_230119 que nous avons créé à cet effet;
- La commande bye déconnecte le client du serveur.

IV.2.4.EXEMPLE N°2-SOLUTION POUR ÉVITER D'EXPOSER LES IDENTIFIANTS DE CONNEXION:

La faiblesse du type de solution proposé au paragraphe précédent est d'exposer les identifiants de connexion dans une procédure texte.

S'il était possible de passer des arguments à la procédure FTP contenue dans le fichier appelé par l'attribut -s, une solution de sécurisation consisterait à lui passer en arguments l'Id. de l'utilisateur et le mot de passe de connexion, ainsi que l'identifiant du serveur.

Malheureusement, cette possibilité n'est pas prévue par la commande ftp -s. La solution consiste à créer un fichier batch qui reçoit en arguments les identifiants de connexion et effectue successivement les actions suivantes:

- Construire le fichier de commandes ftp à exécuter en utilisant les arguments d'entrée;
- Exécuter la commande ftp -s:<fichier commandes ftp>;
- Supprimer le fichier de commandes ftp en fin d'exécution (pour éviter de conserver un fichier contenant les identifiants de connexion).

Fichier batch à créer: BuildCmdFTP.bat

```
1. set /P SRV=Adresse serveur:
2. set /P USR=Id.Utilisateur:
3. set /P PASS=Mot de passe:
4. echo open %SRV%> Commande.ftp
5. echo %USR%>> Commande.ftp
6. echo %PASS%>> Commande.ftp
7. echo prompt off>> Commande.ftp
8. echo !mkdir .\Sauvegarde_drh_230119>> Commande.ftp
9. echo lcd ./Sauvegarde_drh_230119>> Commande.ftp
```

```
10. echo cd /home/drh>> Commande.ftp
11. echo mget *.sql>> Commande.ftp
12. echo bye>> Commande.ftp
13. ftp -s:Commande.ftp
14. del Commande.ftp
```

Explication:

- Les lignes 1 à 3 permettent de saisir en ligne l'adresse du serveur et les identifiants de connexion. Une variante est de saisir ces données en paramètres du fichier batch;
- Les lignes de 4 à 12 permettent de créer le contenu du fichier Commande.ftp en utilisant les valeurs des paramètres d'entrée. On obtient le même contenu que précédemment pour ce fichier. Remarquer que les opérateurs de redirection > et >> sont "collés" au texte des commandes echos pour éviter d'ajouter des espaces en fin de ligne dans le texte obtenu;
- La ligne 13 déclenche la connexion au serveur ftp et l'exécution des commandes ftp contenues dans le fichier Commandes.ftp;
- La ligne 14 efface le fichier Commande.ftp qui contient les paramètres de connexion en clair.

Remarque:

Cette solution est plus sécurisée que la solution précédente, puisque le fichier Commande.ftp, qui renferme les identifiants de connexion en clair, est détruit dès que la commande ftp a été exécutée. Cependant, elle présente l'inconvénient d'exiger une intervention humaine pour saisir les arguments d'entrée. Il n'est donc pas possible de déclencher automatiquement ce type de procédure en utilisant le mécanisme de planification des tâches de windows.

IV.2.5.EXEMPLE N°3-MASQUAGE DES IDENTIFIANTS DE CONNEXION-TRANSFORMATION D'UN BATCH EN EXE:

Cette solution est basée sur la possibilité de transformer des fichiers .bat en fichiers .exe qu'offrent certains outils sous windows (par exemple, l'outil gratuit "bat to exe converter"). Contrairement à un fichier .bat, le fichier .exe obtenu par cette transformation ne peut pas être édité par un traitement de texte "classique". Il est donc beaucoup plus difficile d'accéder à d'éventuels identifiants de connexion.

Cependant, comme il est impossible d'appliquer cette transformation à un fichier .ftp, Il faudra l'appliquer au fichier BuildCmdFTP.bat, en le modifiant pour initialiser directement les variables SRV, USR et PASS dans la procédure:

Par exemple, si l'on se connecte à l'utilisateur Admin du serveur 192.168.1.21, avec le mot de passe %AdmSys%, le contenu du fichier BuildCmdFTP.bat sera:

```
echo open 192.168.1.21> Commande.ftp
echo Admin>> Commande.ftp
echo %AdmSys%>> Commande.ftp
echo prompt off>> Commande.ftp
echo !mkdir .\Sauvegarde_drh_230119>> Commande.ftp
echo lcd ./Sauvegarde_drh_230119>> Commande.ftp
echo cd /home/drh>> Commande.ftp
echo mget *.sql>> Commande.ftp
echo bye>> Commande.ftp
ftp -s:Commande.ftp
del Commande.ftp
```

Après connexion en BuildCmdFTP.exe, les paramètres de connexion ne seront plus lisibles par un éditeur de texte, ce qui limite beaucoup le risque de piratage des identificateurs de connexion.

Un autre avantage est que ce fichier .exe pourra être déclenché par une tâche périodique sans intervention d'un opérateur. Par exemple, la sauvegarde des fichiers *.sql pourra être effectuée tous les vendredis soirs à 22 heures par la commande:

```
schtasks /create /tn "SVG" /tr "C:\Users\ClientDrh\ BuildCmdFTP.bat"
    /sc weekly /d wed /st 22:00
```

IV.3.AUTOMATISATION DES PROCÉDURES FTP SOUS UNIX-LINUX:

IV.3.1.GÉNÉRALITÉS SUR LE CLIENT LFTP:

Sous UNIX-LINUX, les clients ftp en ligne de commande disponibles ne supportent pas l'attribut /S. De ce fait, on ne peut les utiliser pour automatiser les procédures d'échange FTP. La solution la plus courante est alors d'utiliser le logiciel lftp.

LFTP est un client FTP en ligne, qui a l'avantage de permettre l'exécution automatique de fichiers de commandes ftp et permet également des échanges en mode sécurisé.

Lftp accepte toutes les commandes d'un client ftp courant, plus quelques commandes supplémentaires. On peut noter en particulier:

- La commande lftp -u <utilisateur>,<mot de passe> <adresse serveur> qui permet de se connecter à un serveur avec une seule ligne de commandes (l'adresse serveur peut être une URL);

- La commande lftp -f <fichier de commandes lftp> qui permet d'exécuter automatiquement les commandes lftp contenues dans le fichier spécifié;
- La commande lftp -e "liste de commandes séparées par;" qui permet d'exécuter directement la liste des commandes libellée.
- Toutes les commandes de navigation qui s'adressent au serveur (ls, dir, pwd,etc…), s'adressent au client si elles sont précédées de! (!ls,!dir,!pwd,etc…);
- lftp supporte également des commandes spécifiques telles que mirror qui permet de dupliquer des arborescences de fichiers.

IV.3.2.INSTALLATION DE LFTP (ubuntu, debian):

> sudo apt-get install lftp

IV.3.3.EXÉCUTION AUTOMATIQUE D'UN FICHIER DE COMMANDES FTP:

On utilise l'attribut -f de la commande lftp pour spécifier le fichier de commandes à exécuter:

> lftp -f <nom du fichier de commandes>

Ce fichier de commande doit contenir des commandes lftp écrites ligne à ligne.

IV.3.4.EXEMPLE N°1-SCRIPT FTP BASIQUE:

Le script ftp suivant (fichier SauvePerso.lftp)

```
open 192.168.1.21
user admin %adm%
mget /admin/Perso/*.sql -o /Sauvegardes
bye
```

- Ouvre le serveur FTP situé à l'adresse 192.168.1.21;
- Se connecte à ce serveur sous l'utilisateur admin (mot de passe %adm%);
- Rapatrie tous les fichiers .sql du répertoire /admin/Perso dans le répertoire /Sauvegardes du client.
- Se déconnecte du serveur.

L'exécution de ce script sera déclenchée par la commande:

> lftp -f SauvePerso.lftp

IV.3.5.EXEMPLE N°2-SOLUTION POUR ÉVITER D'EXPOSER LES IDENTIFIANTS DE CONNEXION:

La présence des identifiants de connexion en clair dans le fichier SauvePerso.lftp constitue évidemment une faille de sécurité importante. Pour corriger cette faille, nous pouvons envisager, comme dans le cas de windows, de créer un script shell qui dans un

premier temps créera le contenu du fichier de commandes lftp à partir de saisies en ligne, puis lancera l'exécution de ce fichier et le détruira dès que l'exécution sera terminée:

Fichier BuildSauvePerso.bash:

```
read -p 'Adresse serveur:' SRV
read -p 'Utilisateur:' USR
read -p 'Mot de passe:' PASS
echo open $SRV> SauvePerso.lftp
echo user $USR $PASS>> SauvePerso.lftp
echo mget /admin/Perso/*.sql -o /Sauvegardes>> SauvePerso.lftp
echo bye>>SauvePerso.lftp
lftp -f SauvePerso.lftp
rm SauvePerso.lftp
```

Cette solution a pour avantage de ne pas laisser les identifiants de connexion visibles, uisque le fichier qui les contient est détruit automatiquement en fin de traitement.

IV.3.6.EXEMPLE N°3-MASQUAGE DES IDENTIFIANTS DE CONNEXION-TRANSFORMATION D'UN SCRIPT EN EXE:

La solution de l'exemple n° 2 a pour inconvénient de ne permet pas de déclencher automatiquement les traitements avec cron, puisqu'elle exige une interaction avec l'opérateur. Pour améliorer la sécurité, il est possible de transformer le script shell en exécutable .exe, en utilisant le logiciel libre SHC. Ce logiciel peut être installé facilement sous ubuntu-debian par:

> sudo apt-get install shc

Il faut alors dans le shell précedent, remplacer la saisie en ligne des éléments de connexion par des déclarations dans le texte:

Fichier BuildSauvePerso.bash:

```
echo open 192.168.1.21> SauvePerso.lftp
echo user admin %adm%>> SauvePerso.lftp
echo mget /admin/Perso/*.sql -o /Sauvegardes>> SauvePerso.lftp
echo bye>>SauvePerso.lftp
lftp -f SauvePerso.lftp
rm SauvePerso.lftp
```

Puis, convertir BuildSauvePerso.bash en exécutable .exe:

> shc -f BuildSauvePerso.bash

On obtient alors le fichier BuildSauvePerso.bash.x qui n'est plus éditable directement par un traitement de texte et qui peut être déclenché périodiquement sans intervention opérateur par le mécanisme CRON de linux.

REMARQUE: Suivant les versions de linux et de shc, et en particulier lorsque l'on utilise une machine virtuelle, l'exécutable obtenu par shc ne fonctionne pas forcément si on le lance directement (apparemment, il s'agit d'une incompatibilité du code compilé avec la structure machine). Pour contourner le problème, on peut lancer l'exécutable en utilisant la commande: >setarch $(uname -m) -R <chemin d'accès à l'exécutable>

Commentaires:

- "uname -m" retourne un identificateur de l'architecture hardware de la machine (ex: x86_64);
- "setarch <architecture matérielle> -R <Logiciel à exécuter>" adapte l'environnement d'exécution à l'architecture matérielle.

Exemple: > setarch $(uname -m) -R ./BuildSauvePerso.bash.x

V.ANNEXE - LISTE DES COMMANDES FTP:

Commande	DESCRIPTION
help	Affiche l'ensemble des commandes supportées par le serveur FTP
status	Permet de connaître les paramètres ftp de la machine cliente
binary	Bascule du mode ASCII (envoi de documents textes) au mode binary (envoi de fichiers en mode binaire)
ascii	Bascule du mode binary au mode ascii (mode par défaut)
type	Afficher le mode de transfert courant (binary ou ascii)
user	Permet de ré-ouvrir une session sur le site FTP en cours avec un nom d'utilisateur différent.
ls	Liste le répertoire courant du serveur (ls -l pour plus de détails)
pwd	Affiche le nom complet du répertoire courant du serveur.
cd	Change le répertoire courant du serveur (change directory)
lcd	Comme cd, mais agit sur le répertoire local
dir	Liste le répertoire courant du serveur
!dir	En local
mkdir	Permet de créer un répertoire dans le serveur
rmdir	Permet de supprimer un répertoire dans le serveur
!mkdir	Permet de créer un répertoire dans le client
!rmdir	Permet de supprimer un répertoire dans le client
get	Permet de récupérer un fichier présent sur le serveur: get <fichier distant> <fichier local>
mget	Transfert de plusieurs fichiers, grâce à des "wildcards".
put	Permet d'envoyer un fichier local sur le serveur: get <fichier distant> <fichier local>
mput	Transfert de plusieurs fichiers, grâce à des "wildcards".
open	Ferme la session en cours et ouvre une nouvelle session sur le même serveur ou sur un autre serveur FTP
close	Ferme la session en cours, en laissant le logiciel FTP client actif
bye	Déconnecte le logiciel client du serveur FTP et le met en état inactif
quit	Déconnecte le logiciel client du serveur FTP et le met en état inactif