Cours PureFTPD

1) Le protocole FTP

Le protocole FTP (File Transfer Protocol) a été créé en 1971. Il s'agit du protocole de transfert de fichiers le plus répandu**.**

Caractéristiques :

Couche 7(Application) du modèle OSI
Protocole client / serveur
Authentification : login + password
Ports utilisés par défaut : 20 (données) et 21 (commandes).

2) Le serveur PureFTPd

Le serveur PureFTPd est un serveur libre, sécurisé et de qualité.

Il possède de nombreux atouts :

- Choix des ports
- Possibilité du FXP : transfert entre deux serveurs FTP
- Réglage de le bande passante
- Support de ratios
- Possibilité de cryptage avec ssl / tls
- Le « chroot » des utilisateurs dans leur répertoire.

3) Le chroot

Le chrootage permet d'emprisonner un utilisateur dans un répertoire et son arborescence inférieur.

Principe : on isole l'utilisateur du reste du système de fichiers.

Exemple : Un utilisateur toto qui est chrooté dans son répertoire personnel /home/toto

Conséquence : Le plus haut de sa hiérarchie sera son répertoire personnel.

Intérêt : Il ne pourra visualiser aucun répertoire se trouvant à la racine du système.

PureFTP, SSL/TLS, Chroot, Backup-Manager

1) Procédure d'installation

Dans cet exemple de configuration, il s'agira de mettre en place des utilisateurs virtuels, chrootés, avec des options de restrictions propres à chaque utilisateur.

Installation des paquets : dnf install ftp pure-ftpd suffira pour installer l'application.

2) Paramétrage du fichier /etc/pure-ftpd/pure-ftpd.conf

Remarque : On met en place un serveur ftp qui s'exécutera uniquement par des clients virtuels, donc non physiques --> on interdira la connexion PAM (méthode d'authentification classique sous unix basée sur des utilisateurs physiques).

2.1) Paramètres à changer

MaxClientsNumber 50 --> MaxClientsNumber 10 # Nombre total de client pouvant se connecter simultanément

MaxClientsPerIP 8 --> MaxClientsPerIP 3 # Nombre de clients pouvant se connecter simultanément avec la même adresse IP

NoAnonymous no --> NoAnonymous yes # Permet de désactiver l'accès anonyme au serveur ftp

PAMAuthentication yes --> PAMAuthentication no # interdit l'authentification par module PAM.

PureDB /etc/pure-ftpd/pureftpd.pdb --> PureDB /etc/pure-ftpd/pureftpd.pdb # on décommente la ligne, les utilisateurs seront enregistrés dans cette base de données.

2.2) Paramètres à vérifier

ChrootEveryone yes #permet de chrooter tout le monde

Daemonize yes #Permet a pureftpd de tourner en arrière-plan

SyslogFacility ftp #Messages Log FTP

DontResolve yes #on limite les résolutions DNS

2.3) Paramètres optionnels

MaxDiskUsage 99 --> MaxDiskUsage 95 #limitation de la saturation du filesystem)

DisplayDotFiles yes --> DisplayDotFiles no #On n'affiche pas les fichiers cachés automatiquement.

MaxIdleTime 15 --> MaxIdleTime 5 # 5 minutes d'inactivité avant la déconnexion

ProhibitDotFilesWrite no --> ProhibitDotFilesWrite yes #interdiction de créer ou d'écrire dans les fichiers cachés

ProhibitDotFilesRead no --> ProhibitDotFilesRead yes #interdiction de lecture des fichiers cachés

#**NoChmod** yes --> NoChmod yes #les utilisateurs ne peuvent pas changer les permissions des fichiers/dossiers

#**KeepAllFiles** yes --> KeepAllFiles yes #Les utilisateurs ne peuvent pas supprimer de fichiers

#**NoRename** yes --> NoRename yes #Les utilisateurs ne pourront pas renommer les fichiers

2.4) Service pure-ftpd

1) Relancez du service pure-ftpd
2) Activez le service au démarrage du système

3) Préparation de la racine FTP

Explication : On va créer un utilisateur physique sur la machine et lui attribuer le droit de lecture aux données du répertoire ftp.

1) Créez un groupe ftpgroup

2) Créez l'utilisateur ftpuser avec pour home /dev/null et comme invite de commandes /sbin/nologin. : # **useradd -g ftpgroup -d /dev/null -s /sbin/nologin ftpuser**

Explication : Cela interdit à l'utilisateur *ftpuser* la possibilité de se connecter à un terminal, car l'utilisateur n'aura aucun répertoire de travail et n'aura le droit d'exécuter aucune commande

3) Vérifiez que l'utilisateur s'est bien créé

4) Créez le répertoire où seront stockées les données : **# mkdir /ftp**

5) Mettez ftpuser comme propriétaire et ftpgroup comme groupe

6) Enlevez le droit d'écriture à tout le monde. De plus, la catégorie « autres utilisateurs » n'aura aucun droit sur le répertoire.

4) Gestion des utilisateurs virtuels

Explication : Les utilisateurs virtuels seront mappés à l'UID de ftpuser et au GID de ftpgroup.

Par ailleurs, notre serveur ftp aura pour racine /ftp

4.1 Ajout d'un utilisateur

 # pure-pw useradd toto -u ftpuser -d /ftp -m

- -u : login de l'utilisateur ftp/machine que l'on utilisera pour les utilisateurs virtuels
- -d : répertoire racine de l'utilisateur virtuel (sans possibilité de remonter d'un niveau car il est chrooté)
- -m : mise à jour automatique de la base de données des utilisateurs virtuels (/etc/pure-ftpd/pureftpd.pdb)

Explication : l'utilisateur toto est un utilisateur virtuel mappé sur l'utilisateur physique ftpuser

4.2 Visualisation des informations

pure-pw mkdb : permet de mettre à jour la base de données.

 # pure-pw show nom_login : permet de visualiser l'ensemble des informations sur un utilisateur virtuel

Exemple : **# pure-pw show toto**

Remarque : Le /./ à la fin du répertoire d'accueil permet de spécifier que l'utilisateur ne pourra pas remonter au-delà de /ftp (il ne pourra donc pas aller dans /ftp).

4.3 Changement du mot de passe

On peut changer à tout moment le mot de passe d'un utilisateur : **# pure-pw passwd toto -m**

4.4 Connexion d'un utilisateur

1) Pour se connecter sur sa machine, on tape la commande :

ftp localhost

Puis on saisit un login d'un utilisateur et un mot de passe.

1. On peut aussi se connecter depuis un navigateur web : on tape : ftp://localhost/
2. Pour se connecter depuis un client on tape l'adresse ip du serveur : **ftp://192.168.1.13/**

Remarque : Vérifiez que votre pare-feu autorise les requêtes ftp et que selinux soit désactivé.

4.5 Ajout d'un utilisateur spécial pour l'upload

Cet utilisateur pourra écrire des fichiers dans le dossier upload du serveur FTP (/ftp/upload du serveur). Ce sera son dossier d'accueil mais pourra remonter à la racine du FTP (/ftp/upload).

1. Créez un répertoire d'upload dans /ftp :

2. Créez un nouvel utilisateur pureftp : **# useradd -g ftpgroup -d /dev/null -s /sbin/nologin ftpupload**

3. Créez un utilisateur virtuel : **# pure-pw useradd upload -u ftpupload -D /ftp/./upload -m**

Remarque : -D : Ne pas chrooter dans le dossier racine (la racine sera explicitement définie par /./)

4. Changez les droits du répertoire /ftp/upload/ : le propriétaire aura le droit d'écriture sur le répertoire, le groupe pourra lire les données et les autres n'aurons aucun droit.

5. Mettre ftpupload en tant que propriétaire et ftpgroup en tant que groupe du répertoire.

6. Connectez-vous avec upload et vérifiez qu'il a bien le droit d'écriture dans le répertoire.

7. Connectez-vous avec toto et vérifiez qu'il peut lire les données du répertoire upload, mais qu'il ne possède pas les droits d'écriture.

4.6 Création d'un utilisateur sans droit

Travail : Créer l'utilisateur ftp « test » qui n'aura aucun droit sur le répertoire.

4.7 Modification des propriétés de l'utilisateur :

#pure-pw usermod login options : permet de modifier les propriétés d'un utilisateur**.**

Voici les principales autres options :

- -c : commentaire
- -t : limitation de la bande passante du téléchargement
- -T : limitation de la bande passante de l'envoi
- -n : quota en nombre de fichiers max
- -N : quota en taille maximale utilisable
- -r : IP (ou noms d'hôtes) depuis lesquelles l'utilisateur est autorisé à se connecter
- -R : IP (ou noms d'hôtes) depuis lesquels l'utilisateur n'est pas autorisé à se connecter
- -z : limitation horaire pendant laquelle l'utilisateur peut se connecter
- -y : nombre de sessions simultanées autorisées

1. Limitez la bande passante en téléchargement à 8 ko/s à toto

2. Limitez la connexion d'un utilisateur à une plage horaire

Remarque : Après la modification, saisir la commande **pure-pw mkdb** pour mettre à jour la bdd

3. Limitez à une seule connexion simultanée :

4.8 Suppression d'un utilisateur :

1. Supprimez l'utilisateur toto # pure-pw userdel toto

2. Vérifiez que toto a bien été supprimé : # pure-pw show toto
   ->Unable to fetch info about user [toto] in file [/etc/pure-ftpd/pureftpd.passwd]

5) Suivi de l'activité du serveur

1. Le suivi se fait par la commande : **#pure-ftpwho**.

Remarque : On y trouve le PID, l'utilisateur, la durée, la vitesse, son action, le fichier et l'adresse IP de l'utilisateur.

2) Regarder les informations dans /var/log/messages : **# tail -f /var/log/messages | grep ftp**

6) Problèmes sécurité : Mise en place de SSL

Approche : Le protocole FTP présente une importante faille de sécurité : Le login et le password circulent en clair !

Solution : Pour parer à ce problème, nous allons intégrer le protocole TLS (anciennement SSL) qui va crypter toutes les trames FTP.

1) Installez le logiciel d'analyse de trames wireshark et wireshark-gnome

2. Lancez l'analyseur de trame sur l'interface lo

3. Constatez que le login et le mot de passe circulent en clair

4. Installez le paquet openssl

5. Créez un certificat ssl pour pureftp : openssl req -x509 -nodes -newkey rsa:1024 -keyout /etc/pki/pure-ftpd/pure-ftpd.pem -out /etc/pki/pure-ftpd/pure-ftpd.pem

   Remarque : /etc/pki/pure-ftpd/pure-ftpd.pem est le fichier qui va gérer ssl pour pure-ftp

   Remplir les informations demandées pour générer le certificat.

6. Modifiez le fichier de configuration pureftp : Affectez la valeur 1 à la ligne TLS

7. Relancez le service pure-ftpd

8. Pour se connectez avec un certificat ssl, installez le logiciel Filezilla

9. Cliquez sur Fichier / Gestionnaire de Sites/Nouveau site

10. Lancez Wirershark

11. Remplir les informations : Hôte (ip du serveur), type de serveur (FTP avec TLS explicite), type d'authentification (normale), identifiant et mot de passe.

12. Acceptez le certificat SSL

13. Vérifiez que les informations sont cryptées lorsque vous vous connectez par ssl.

7) Problèmes rencontrés

En cas de dysfonctionnement, il faut regarder dans les logs (journactl -xe). Le pare feu de Fedora bloque par défaut les ports utilisés par FTP (20 et 21), il faut penser à le configurer. Si le serveur FTP doit être utilisé derrière un routeur, il faut penser à faire une translation de port. En cas de problème de connexion, il faut essayer en local, puis depuis une machine dans le réseau local (LAN) et terminer enfin par une machine sur Internet afin de vérifier où le blocage se situe.

8) Backup Manager

Backup Manager est un gestionnaire de sauvegardes simple, automatique et sûre. Il permet :

- de sauvegarder vos données sous forme d'archive tar/dar
- d'effacer les vieilles sauvegardes.
- de compresser les sauvegardes, les découper en plusieurs fichiers de taille déterminée.
- de ne sauvegarder que les différences entre les sauvegardes (sauvegarde incrémentale)
- de graver automatiquement les sauvegardes sur CD/DVD
- d'exporter les sauvegardes sur une machine distante via FTP ou SSH
- d'exécuter une commande avant et après la sauvegarde
- de créer plusieurs méthodes de sauvegarde grâce à divers outils de sauvegardes

Dans notre cas, nous allons sauvegarder les fichiers du répertoire /va/www sur le serveur FTP précédemment créé.

- Nous allons procéder à la configuration d'une sauvegarde incrémentielle (seul ce qui a été modifié sera sauvegardé). La sauvegarde se réalisera tous les 5 jours via le protocole Rsync

Rsync (pour **r**emote **sync**hronization ou synchronisation à distance), est un logiciel de synchronisation de fichiers. Il est fréquemment utilisé pour mettre en place des systèmes de sauvegarde distante.

Rsync travaille de manière unidirectionnelle c'est-à-dire qu'il synchronise, copie ou actualise les données d'une source (locale ou distante) vers une destination (locale ou distante) en ne transférant que les octets des fichiers qui ont été modifiés.

Rsync utilise le protocole SSH pour la connexion vers le serveur distant.

SSH (Secure SHell) est un protocole de communication (écoutant sur le port 22) permettant de se connecter de façon sécurisée à un système Unix, Linux et Windows.

SSH permet de garantir :

- La confidentialité : le cryptage des paquets permet de garantir celle-ci. Les services tels que telnet, rlogin envoient les données en clair.
- L'intégrité : SSH permet de garantir que les pastartquets circulant d'un hôte vers un autre ne sont pas altérés.
- L'authentification : chaque connexion SSH vérifie l'identité du serveur (par sa clé d'hôte ~/.ssh/known_hosts) puis celle du client (par mot de passe ou clé publique ~/.ssh/authorized_keys).
- L'autorisation : il est possible avec SSH de limiter les actions autorisées à l'utilisateur (~/ssh/.authorization).
- Tunneling : SSH permet de sécuriser un service dont les informations circulent habituellement en clair (POP, IMAP, VNC, ...).

Face à la faiblesse de l'authentification par mot de passe, l'authentification par clé se révèle être très efficace.

La clé permet de garantir à un système qu'un utilisateur est bien celui qu'il prétend être... en deux mots : « Je jure et je prouve que c'est bien moi ».

L'authentification par clé fonctionne grâce à 3 composants :

3. Une clé publique : elle sera exportée sur chaque hôte sur lequel on souhaite pouvoir se connecter.
4. Une clé privée : elle permet de prouver son identité aux serveurs.
5. Une passphrase : Permet de sécuriser la clé privée (notons la subtilité, passphrase et pas password ... donc « phrase de passe » et non pas « mot de passe »).

Travail : Dans un premier temps, nous allons tester la connexion ssh vers le serveur FTP.

1. **Sur le Client**

Connexion ssh vers le serveur : **ssh root@192.168.1.1 (par exemple).**

Remarque : Le service sshd doit être lancé sur le serveur et le pare-feu doit accepter les requêtes.

Première connexion : Lors de la première connexion SSH vers un hôte, ce message peut apparaitre :

The authenticity of host 'serveur (192.168.1.1)' can't be established.

RSA key fingerprint is c6:ee:c6:e4:9a:b6:7e:46:4c:17:b4:d0:7b:80:af:2c.

Are you sure you want to continue connecting (yes/no)? Il faut répondre yes lors de cette première connexion.

Warning: Permanently added 'serveur' (RSA) to the list of known hosts.

La clé est maintenant conservée (fichier ~/.ssh/known_hosts) .

La création de la paire de clé se fait avec la commande : **ssh-keygen**

Il existe 2 types de clés : **RSA et DSA** correspondent à deux algorithmes différents. On peut utiliser l'un ou l'autre.

Chacune pouvant être de longueur différente : 1024, 2048, 4096 bits.

Commande pour une clé RSA de 1024 bits : **ssh-keygen -t rsa -b 1024 -C clessh**

Explication :

-t : type de clé (DSA ou RSA)

-b : nombre de bytes

-C : commentaire sur la clé

Travail : Générez une clé RSA de 1024 bits.

Création de la clé :

Generating public/private dsa key pair.
Enter file in which to save the key (/home/client/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/client/.ssh/id_dsa.
Your public key has been saved in /home/client/.ssh/id_dsa.pub.
The key fingerprint is:
cb:61:48:6b:b4:53:00:9b:d1:2a:cf:44:88:79:c2:19 **clessh**

Deux fichiers ont été créés (dans le dossier ~/.ssh/) du home directory de client :

2. **id_rsa** : contient la clé privée et ne doit pas être dévoilé ou mis à disposition
3. **id_rsa.pub** : contient la clé publique, c'est elle qui sera mise sur les serveurs dont l'accès est voulu.

Sur le poste Client : Il faut envoyer la clé générée par le client sur le poste du serveur. Pour ce faire, on tape la commande suivante : **ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.1.1**

Cette commande envoie le contenu de la clé publique dans le fichier /home/serveur/.ssh/authorized-key (qui sera modifié ou créé) afin que le serveur accepte le nouveau client.
Remarque : il vous sera demandé le mot de passe du compte serveur.

Sur le poste Serveur: On interdit tout accès avec mot de passe pour des raisons de sécurité. Une connexion ssh ne pourra se réaliser qu'avec une paire de clé publique/privée.

Désactivation de l'authentification par mot de passe dans le fichier : **/etc/ssh/sshd_config**
PasswordAuthentication no

Relancez le service sshd

Activez le service sshd au démarrage du système.

Test : On peut désormais se connecter sans à avoir à renseigner le mot de passe.

- Installez le paquet **backup-manager** sur le poste client.

- La configuration se réalise dans le fichier suivant : **/etc/backup-manager.conf**

Plusieurs paramètres sont à modifier :

```
//chemin où seront stockées vos sauvegardes  :
export BM_REPOSITORY_ROOT="/backup"

//Les dossier à sauvegarder
BM_TARBALL_TARGETS[0]="/va/www"
#BM_TARBALL_TARGETS[1]="/boot"

//nombre de jours de rétention de vos sauvegardes :
export BM_ARCHIVE_TTL="5"

//Le nom de la machine sera aussi le nom de vos fichiers de sauvegarde :
export BM_ARCHIVE_PREFIX="$HOSTNAME"

//sauvegarde incrémentielle (enregistre que ce qui n'a été modifié depuis la dernière
sauvegarde).
export BM_ARCHIVE_METHOD="tarball-incremental"

//methode d'export, pour nous ftp
export BM_UPLOAD_METHOD="rsync"

# the user to use for the SSH connections/transfers
export BM_UPLOAD_SSH_USER="root"

# The private key to use for opening the connection
export BM_UPLOAD_SSH_KEY="/root/.ssh/id_rsa"

# Which directories should be backuped with rsync
export BM_UPLOAD_RSYNC_DIRECTORIES="/var/www"

# Destination for rsync uploads (overrides BM_UPLOAD_DESTINATION)
export BM_UPLOAD_RSYNC_DESTINATION="/ftp/upload"

# The list of remote hosts, if you want to enable the upload
export BM_UPLOAD_RSYNC_HOSTS="192.168.1.1"
```

- Sauvegardez le fichier, puis taper la commande dans le terminal : **backup-manager -v**

Le transfert des fichiers doit se réaliser.

- Vérifiez que les dossiers ont bien été uploadés sur le serveur.

- Réaliser un script permettant d'automatiser la sauvegarde/synchronisation tous les matins à 8h00