TP de SSH

# Principe et prérequis.

Ce fascicule traite de l’utilisation de **SSH** (Secure Shell) qui est un protocole de communication qui permet une connexion cryptée. Nous ne traiterons ses utilisations que sur un serveur Linux et depuis des postes Linux.

Il se base sur un algorithme de cryptage asymétrique, c’est à dire que la clé de chiffrement n’est pas la même que la clé de déchiffrement. Chaque client génère une paire de clé et envoie sa clé publique au serveur distant pour pouvoir s’y connecter. La clé privée ne doit jamais être déplacée pour limiter tout risque de divulgation.

.

Il faut distinguer les termes :

- **SSH** : le protocole de communication

- **ssh** : le programme client permettant de se connecter au serveur

- **sshd** : le serveur (ssh daemon) écoutant sur le port 22 par défaut

# II. Installation partie serveur (Linux).

Installation: **sudo apt install openssh-server**

Lancez le service **sshd**

**III.** **Authentification par mot de passe**

Sur le serveur et le client linux, on lance le service ssh (**ssh start**), puis on teste une connexion **ssh** depuis le serveur sur le client :

Syntaxe ssh : **ssh [options] user@adresse**

Quelques options :

-l <login> : identifiant de l’utilisateur.

-1 ou -2 : version de ssh employé. La version 2 est plus sécurisée.

-p <port> : Numéro de port utilisé (22 par défaut)

**Travail :** Testez de vous connecter en **ssh** sur les 2 machines.

# IV. Authentification par clé

**A. Approche :**

L'authentification par mot de passe n’est pas très robuste. Aussi, l'authentification par clé est une meilleure alternative. La clé permet de garantir à un système l’authenticité d’un utilisateur.

L'authentification par clé fonctionne se fait grâce à 3 éléments :

* Une **clé** **publique** : exportée sur chaque hôte sur lequel on souhaite pouvoir se connecter.
* Une **clé** **privée** : permet de prouver son identité aux serveurs.
* Une **passphrase** : permet de sécuriser la clé privée.

**NB** : La **passphrase** seule ne sert à rien sans la clé privée, et vice-versa.

Le réseau possède 2 postes (Serveur et Client) sous Linux. Le service **ssh** permet de prendre le contrôle des postes clients à distance. Nous allons configurer **ssh** de telle sorte que les connexions se réalisent grâce aux clés publiques/privées et sans mot de passe.

**B. Configuration sur le Poste Serveur :**

a) Vérifier si **ssh** est déjà présent sur votre poste : **rpm -q openssh**

**b) Si besoin** : Installation ou mise à jour : **sudo apt-get install openssh**

**c)** Le fichier de configuration est accessible sur : /**etc/ssh/sshd\_config**

L'authentification par clé est activée par défaut. Dans le fichier de configuration, toutes les valeurs commentées (précédées d'un #) sont placées à leur valeur par défaut.

Remarque : pour modifier une variable, enlever le # pour prendre en compte la nouvelle valeur.

**d) Connexion ssh vers le serveur** : ***ssh user@ip\_address***

Remarque : L'échange de clé n’est pas encore effectué, il vous sera alors demandé le mot de passe de l'utilisateur serveur.

**C. création de la paire de clé sur le client**

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 –P ’’passpass’’-b 1024**

Avec :

* -t : type de clé (DSA ou RSA)
* -b : nombre de bytes

**Travail** : Générer une clé **RSA** de 1024 bits.

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

* **id\_rsa** : contient la clé privée.
* **id\_rsa.pub** : contient la clé publique, c'est elle qui sera mise sur le serveur.

**D. Extraction de clé publique**

**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 user@ip\_address**

Cette commande envoie le 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 faire 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**

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

Remarque : Vous devez néanmoins renseigner la **passphrase** à chaque redémarrage.

**V. Sécurité**

**A. Accès au root**

Il pourrait être risqué d'autoriser une connexion vers **root**. Il est donc conseillé d'exclure cet utilisateur pour les connexions **ssh**. De même, Il est possible de changer le numéro de port **ssh** (22 par défaut) pour renforcer la sécurité.

Pour ce faire, vous devez à nouveau modifier le fichier **/etc/ssh/sshd\_config**

**Port 202 # Changer le port par défaut**

**PermitRootLogin no # Ne pas permettre de login en root**

Une fois le fichier sauvegardé, relancez le service **sshd**.

Testez une connexion sous root (sans le port 2022, puis avec).

Remarque : vous devez repasser **PasswordAuthentification à yes.**

**VI. ssh-agent sur le poste Client**

Le serveur **SSH** est sécurisé mais requiert de taper des **passphrases** à chaque connexion. L'**agent SSH** permet de le faire une seule fois et conserve la **passphrase** en mémoire pendant tout son fonctionnement. Les communications SSH fonctionneront par la suite de façon transparente. Pour activer l’agent SSH :

#Lancer le ssh-agent **: ssh-agent $SHELL**

#charger les clés présentes dans ~/.ssh **: ssh-add**

La **passphrase** est enregistrée sur le système, toutes les connexions seront désormais transparentes.

**VII. SFTP**

La commande **sftp** permet de transférer un fichier entre deux serveurs, de manière similaire à **ftp** mais sécurisée. Le chemin du serveur peut être indiqué en absolu ou relatif à partir du répertoire de base. Pour utiliser **sftp**, vous devez connaître l’arborescence exacte des répertoires de la machine distante.

Il faut que **SSH** soit installé sur les deux machines pour effectuer le transfert.

Syntaxe : **sftp user*@IP-serveur***

Cette commande donne accès à :

sftp >

Tapez **help** pour accéder à la liste des commandes internes à **sftp**.

Commandes utiles :

**quit** : pour quitter la session en cours

**get** : récupère un fichier présent sur le serveur FTP et le place sur votre machine

**put** : transfère un fichier de votre disque dur vers le serveur

**ls** : permet de lister le contenu du répertoire courant côté FTP

**pwd** : affiche le nom du répertoire courant sur le FTP

**delete** et **rm** : effacent un fichier sur le FTP

**mkdir** : créé un répertoire sur le FTP

**Exemple**:

Pour envoyer un fichier : sftp> put fichier

Pour télécharger un fichier : sftp> get fichier