LIGNIERE
Timothée
BTS2 SIO
Date de création
23/09/23
Date de maj
16/04/24

SSH

Sommaire

1 - Problématique....2
2 - Notice d'installation....3
2.1.1 – Installation Serveur SSH....3
2.1.2 – Installation Client SSH....4
2.1.3 – Configuration Serveur SSH....5
2.1.3.1 – Modification écoute du serveur....5
2.1.3.2 – Connexion par clé....5
3 - Notice d'utilisation....7
3.1.1 – SSH basique....7
3.1.2 – Sélection de Port....7
3.1.3 – Sélection de clé....7
3.1.4 – Tunneling....7
3.1.5 – Connexion à un hôte connecter à un domaine....8
3.2 - SCP....9
3.3 – Automatiser les authentifications....10
4 - Annexes....11

1 - Problématique

Dans certains cas, il est impossible d’avoir un accès physique à une machine pour la configurer ou simplement la mettre en place.

SSH résout ce problème en permettant un contrôle à distance crypté avec une panoplie de commandes pour assurer la sécurité de la prise en main par le réseau.

2 - Notice d’installation

/!\ - L’installation suivante a été fait sous Debian 11, des différences peuvent exister selon les OS.

2.1.1 – Installation Serveur SSH

Le serveur SSH doit être positionné sur la machine qui a besoin d’être contrôlé à distance.

L’installation du serveur SSH n’est pas compliqué, c’est la configuration qui nous prendra du temps :

```
apt update
apt install openssh-server
```

L’installation est complète, désormais, nous allons passer à la configuration du serveur.

Nous allons établir une adresse IP fixe pour faciliter vos connexions. Si la machine venait à changer d’adresse à cause du DHCP alors que vous n’avez plus aucun moyen de retrouver son adresse, vous seriez dans une position délicate.

```
nano /etc/network/interfaces
```

Un fichier devrait s’ouvrir, vous allez y trouver une configuration similaire à la suivante :

```
allow-hotplug <Nom de l’interface réseau>
iface <Nom de l’interface réseau> inet dhcp
```

Mettez-la en commentaire en mettant des # au début des lignes, puis positionnez ceci en dessous :

```
allow-hotplug <Nom de l’interface réseau>
iface <Nom de l’interface réseau> inet static
address <Adresse IPV4 que vous voulez donner à cette machine>
netmask <Le masque réseau>
gateway <La passerelle du réseau>
```

Puis redémarrer le service du réseau :

```
systemctl restart networking.service
```

2.1.2 – Installation Client SSH

Le client SSH n’est qu’une commande, nous verrons plusieurs de ses paramètres plus tard dans le tutoriel. De manière générale, elle est déjà installée sous Linux ; si vous utilisez Windows, je vous recommande Tabby.

Dans le cas où elle ne serait pas installée sur votre machine Linux, tapé la commande suivante :

```
apt install ssh
```

2.1.3 – Configuration Serveur SSH

Pour la configuration, nous nous concentrerons sur les paramètres les plus courants, étant donné que certains sont destinés à une utilisation de niche.

L'ensemble de ces paramètres sont accessibles par :

```
nano /etc/ssh/sshd_config
```

Vous devriez avoir un fichier répertoriant les paramètres réglés par défaut du serveur SSH. Si vous désirez remettre un paramètre à sa valeur initiale après modification, placez un # au début de la ligne qui contient le paramètre en question.

Lorsque vous serez satisfait de vos modifications, faites :

```
systemctl restart sshd
```

2.1.3.1 – Modification écoute du serveur

Les paramètres initiaux du serveur SSH lui donne comme consigne qu'il doit écouter le port 22 et que toutes les adresses sont acceptées.

```
Port <Numéro du port virtuel voulu>
AddressFamily <any pour tout type, inet pour IPV4, inet6 pour IPV6>
ListenAddress <Adresse IPV4 autorisé (0.0.0.0 signifie que toutes les adresses sont autorisées)>
ListenAddress <Adresse IPV6 autorisé ( :: signifie que toutes les adresses sont autorisés )>
```

2.1.3.2 – Connexion par clé

Pour une connexion par clé, une clé publique et une clé privée doivent être créé au préalable du côté Client, ce que nous allons donc faire puis envoyé la clé publique au Serveur pour permettre les connexions.

Sur le **Client**, créé la clé :

```
ssh-keygen -t <Format choisi> -f <Nom de votre clé>
```

En format pour votre clé, vous pouvez utiliser les formats suivants, tenez-vous au courant vis-à-vis de celui d'actualités et lesquelles sont obsolètes :

dsa / ecdsa / ecdsa-sk / ed25519 / ed25519-sk / rsa

Vous pouvez aussi préciser la longueur de votre clé avec l'option **-b**, cependant, selon le format, cette option peut être ignorée (par exemple avec l'ed25519)

Votre machine va ensuite, vous demandez où vous souhaitez positionner la clé dans les dossiers puis un mot de passe pour permettre l'utilisation de la clé.

Faites attention, si vous avez déjà une clé à cet emplacement avec le même nom, elle sera écrasée et les serveurs SSH où vous l'aviez paramétré deviendront inaccessibles.

Puis envoyé la clé publique au serveur de la manière suivante :

```
ssh-copy-id <IP de votre serveur>
```

Entrer le mot de passe que vous avez défini pour la clé.

Bravo, vous avez terminé cette partie de la configuration côté client.

Il est désormais possible de se connecter au serveur sans le mot de passe.

Maintenant, sur le **Serveur**, nous allons désactiver la connexion par mot de passe.

Pour cela, redirigez-vous vers /etc/ssh/sshd_config pour modifier deux paramètres :

```
PasswordAuthentication No
PubkeyAuthentication Yes
```

« Pubkey Authentication » devrait déjà avoir la valeur « Yes » mais vérifiez tout de même par sécurité.

Redémarrez le service du serveur SSH :

```
systemctl restart sshd_config
```

Désormais, la connexion à votre serveur depuis le serveur avec un mot de passe devrait être impossible.

Et depuis le client, vous n'aurez qu'à donner à la commande ssh l'IP du serveur sans l'utilisateur.

3 - Notice d'utilisation

3.1.1 – SSH basique

L'utilisation la plus courante de ssh est la suivante :

```
ssh <Nom d'utilisateur présent sur le serveur>@<Ip du serveur>
```

3.1.2 – Sélection de Port

Dans le cas où le port SSH du serveur a été changé, rajoutez l'option P :

```
ssh -p <Port du serveur> <Nom d'utilisateur présent sur le serveur>@<Ip du serveur>
```

Dans certains cas, l'option P peut varier en écriture et sera plutôt :

```
ssh -p <Nom d'utilisateur présent sur le serveur>@<Ip du serveur> :<Port du serveur>
```

3.1.3 – Sélection de clé

Parfois, il est nécessaire de préciser que le client doit se connecter à l'aide d'une clé, on utilise donc le paramètre i :

```
ssh -i <Emplacement absolu de la clé sur la machine> <Nom d'utilisateur présent sur le serveur>@<Ip du serveur>
```

3.1.4 – Tunneling

Dans le cas où un service n'est accessible que depuis le point du serveur, il est possible de faire du Tunneling, ce qui permet de se connecter à une machine qui serait normalement Indisponible.

Pour cela, on utilise les options f,N et L

```
ssh -f <Nom d'utilisateur présent sur le serveur>@<Ip du serveur> -L <Port d'entrée du Tunnel>:<Adresse de la machine à laquelle vous voulez accéder>:<Port de sortie du tunnel> -N
```

-f permet de faire tourner la commande en arrière plan

-L permet la mise en place du Tunnel

-N va de pair avec -L

Je vous recommande de faire divers tests pour prendre en main le tunneling, ce n'est pas un principe très clair lors de son introduction.

3.1.5 – Connexion à un hôte connecter à un domaine

Dans certains cas, la machine est reliée à un domaine et on peut chercher à se connecter à l'un des utilisateurs de ce domaine. Pour cela, il faut faire la commande suivante :

```
ssh -L <Nom d'utilisateur du domaine>@<Nom du domaine en majuscule>
<Adresse de la machine>
```

3.2 - SCP

Scp est une commande parallèle à ssh qui permet le transfert de fichier par le système de ssh.

Son utilisation est la suivante :

```
scp  <utilisateur>@<Ip de la machine>:<emplacement fichier> <utilisateur>@<Ip de la deuxième machine>:<emplacement fichier>
```

Ses options de connexion sont identiques à ssh.

Cependant, elle a des options en plus qui concernent la gestion des fichiers :

-q les modifications sur les fichiers (telle que les horaires de création) sont conservés

-r permet de copier de manière récursive les fichiers et sous dossiers d'un dossier

-C compresse le fichier envoyé

-i permet de spécifier une clé pour se connecter

-P spécifie le port auquel se connecter (Identique à la commande ssh)

-p indique qu'il faut écraser le fichier ayant le même nom pour le remplacer par celui transféré

-q mets la commande en mode silencieux, aucune indication sur la progression ne sera donné ; seuls les messages d'erreurs le seront.

-v permet d'afficher le debug.

3.3 – Automatiser les authentifications

Si les machines auquelles vous vous connectez sont très souvent les mêmes et qu'elles ont reçu une clé de connexion alors, vous pouvez automatiser la précision de l'adresse IP, de l'utilisateur, de la clé utiliser et enfin du port d'écoute ; pour cela, dirigez-vous dans le dossier .ssh qui devrait se trouver dans le dossier de votre utilisateur sur votre propre machine.

```
cd /home/<utilisateur>/.ssh
```

Dans ce dossier, vous trouverez l'ensemble des clés que vous avez générées auparavant, les machines auquel vous vous êtes connecté dans « know_hosts » et enfin un fichier de configuration.

C'est celui-ci qui nous intéresse, ouvrez-le donc avec un éditeur de texte, puis pour chaque machine dont vous souhaitez faciliter la connexion, faites l'équivalent du texte suivant :

```
Host <Nom qui indiquera à la commande ssh de suivre les consignes suivantes>
Hostname <Adresse ou nom de la machine dans votre DNS>
Port <Port d'écoute de la machine>
User <Utilisateur pour se connecter>
IdentityFile <Chemin absolue vers la clé de connexion>
```

4 - Annexes

Fiche de recette

Vérification de l'opérationnalité de la solution mise en œuvre : *SSH*

Description du test :

1. *Se connecter à une machine distante avec son mot de passe*
2. *Se connecter à une machine distante à l'aide d'une clé*
3. *Transféré un fichier à l'aide d'un SCP*

Résultats Attendus :

1. *Connexion réussie*
2. *Connexion réussie*
3. *Transfert réussi*

Réception Globale : SSH **Date:** ***23/09/23***
Auteurs: Timothée LIGNIERE

Reçu : ☐

Reçu avec réserve : ☐ ……

Refusé : ☐ ……
Commentaire :

Recette étape par étape * ******* *(pour chaque étape, vous devez élaborer dans un fichier distinct un scénario détaillé à faire appliquer au « client » venant valider votre solution)*

Réception Etape 1: Se connecter à une machine ayant un serveur ssh à l'aide d'un nom d'utilisateur et du mot de passe

Reçu : ☐

Reçu avec réserve : ☐ ……

Refusé : ☐ ……
Commentaire :

Réception Etape 2 : *Se connecter à une machine toujours à l'aide d'un nom d'utilisateur mais en plus avec la clé qui a été fourni au serveur si c'est le cas*

Reçu : ☐

Reçu avec réserve : ☐ ……

Refusé : ☐ ……
Commentaire :

Réception Etape 3 : *Transféré un fichier depuis une machine vers une autre à l'aide de SCP*

Reçu : ☐

Reçu avec réserve : ☐ ……

Refusé : ☐ ……
Commentaire :