Table des matières

TP- Admin à distance : SSH – Cousin Baptiste

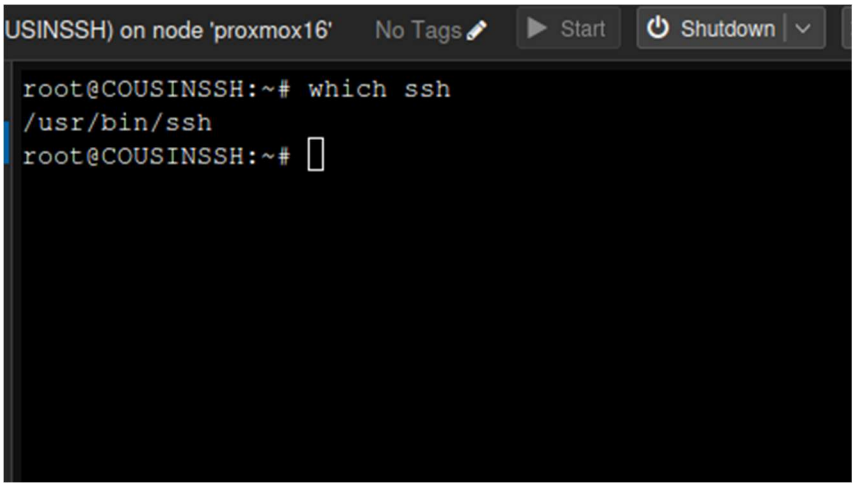
C'est quoi SSH ?

SSH (Secure Shell) est un protocole réseau sécurisé qui permet d'établir une connexion chiffrée entre un client et un serveur pour l'administration à distance et le transfert de fichiers. Il assure l'authentification, la confidentialité et l'intégrité des données.

Installation du serveur SSH sur Debian

On va vérifier que SSH n'est pas déjà installé, en utilisant la commande « which ssh ».

La commande nous renvoie : /usr/bin/ssh, cela veut dire que ssh est installé.



SSH est installé et donc déjà opérationnel.

Si SSH n'est pas installé, l'installer avec la commande : apt install openssh-server

SSH est donc utilisable. Pour nous connecter à distance depuis windows, nous allons utiliser le terminal windows « cmd ». Ouvrir cmd et taper ssh root@192.168.20.216

On peut maintenant naviguer au sein des fichiers etc... Comme si nous étions sur la machine en local.

Si on se connecte avec le login user, on ne peut pas modifier le fichier car « user » n'est pas administrateur.

Configuration et sécurisation d'un serveur SSH et d'un client

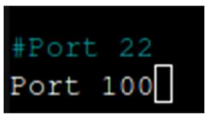
Modification du port d'écoute

Maintenant, nous allons modifier la configuration de SSH pour le sécuriser et l'adapter.

Sur le serveur, nous allons éditer le fichier /etc/ssh/sshd_config, en tapant nano /etc/ssh/sshd_config

Nous sommes bien en « root » afin de modifier ce fichier.

Une fois le fichier ouvert, on va commencer par modifier le port d'écoute de SSH, on va modifier la ligne comportant le numéro 22 et le changer en 100 par exemple.



On redémarre ensuite le service ssh : service ssh start

Pour se reconnecter, on se déconnecte en faisant exit puis on se reconnecte.

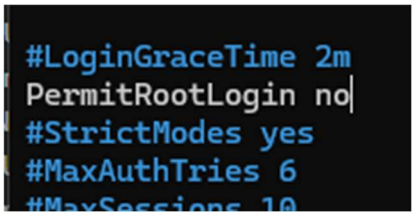
Pourquoi modifier le port d'écoute ?

Pour renforcer la sécurité du serveur, j'ai modifié le port d'écoute de SSH afin de réduire les attaques automatisées et limiter les tentatives de connexion non autorisées.

Autorisation login root

Autoriser ou non la connexion en root sur un serveur SSH est très important pour des raisons de sécurité et de contrôle d'accès. On peut ainsi sécuriser le serveur contre des attaques par mot de passe, controler l'accès...

Pour le modifier, on modifie la ligne « PermitRootLogin » sur no :



On enregistre la config et on restart le service ssh comme précédemment.

Permettre les mots de passe vide

Cette option permet à l'utilisateur de se connecter en laissant le champ mot de passe vide. Par défaut cette option est désactivée et pour garantir la sécurité, nous allons la laisser désactivée.

PermitEmptyPasswords no se rapporte aux mots de passe vides pour tous les utilisateurs, tandis que PermitRootLogin without-password concerne l'accès du compte root via SSH.

Gérer les clés d'authentification sur le serveur et le client

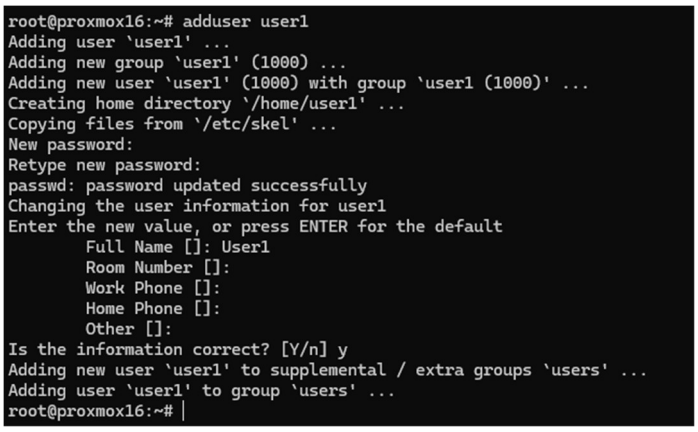
On va d'abord créer 3 comptes utilisateurs en procédant comme ceci :

User1, dans les groupes etudiant et ssh

User2, dans le groupe ssh

User3, dans le groupe étudiant

Pour créer les utilisateurs, on va utiliser la commande adduser nom_d_utilisateur



On procède de même avec les user2 et user3. On a entré comme mot de passe pour chaque utilisateur Password1, comme demandé

Une fois les utilisateurs créés, on va les ajouter à leurs groupes en utilisant la commande :

usermod -aG nom_du_groupe nom_utilisateur

Avant cela, il faut créer les groupes :

Groupadd nom_du_groupe

```
root@proxmox16:~# groupadd etudiant
root@proxmox16:~# groupadd ssh
root@proxmox16:~#
```

Les groupes sont créés, on va maintenant pouvoir mettre les utilisateurs dans les groupes :

```
root@proxmox16:~# usermod -aG etudiant user1
root@proxmox16:~# usermod -aG etudiant user3
root@proxmox16:~# usermod -aG ssh user1
root@proxmox16:~# usermod -aG ssh user2
root@proxmox16:~#
```

Gérer les échanges des clés publiques

Nous allons créer le répertoire home de chaque utilisateur (/home/user1, etc..).

On se place dans le répertoire « home » : cd /home.

On créé le répertoire de chaque utilisateur et on met en place les permissions :

mkdir /home/user1 /home/user2 /home/user3

mkdir /home/user1/.ssh /home/user2/.ssh /home/user3/.ssh

chmod 700 /home/user1/.ssh /home/user2/.ssh /home/user3/.ssh

chown user1:user1 /home/user1/.ssh

chown user2:user2 /home/user2/.ssh

chown user3:user3 /home/user3/.ssh

Dans le répertoire home de l'utilisateur root, on va créer un sous-dossier .ssh et changer la permission :

cd /root

Mkdir.ssh

Chmod 700 .ssh

Chown root :root .ssh

Nous allons maintenant nous connecter sur la machine cliente avec chaque utilisateur afin de créer une clé publique dans le dossier /home/user1/.ssh.

Pour cela, on tape la commande : ssh-keygen sur chaque session cliente. On entrera comme mot de passe : sio2025

```
user2@debiansio:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user2/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user2/.ssh/id_rsa
Your public key has been saved in /home/user2/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:x6XkNWz/dR7nsP23CkEWSrlmCfb7NHbT/Z4JbPRLEYU user2@debiansio
The key's randomart image is:
+---[RSA 3072]----+
|         ...   ..|
|       o..... E .|
|      . o.+o*  . |
|         Xo= o  .|
|        S *. .+o=|
|         o ++o.OB|
|          +.o+oo*|
|           .o o B|
|             ..*=|
+----[SHA256]-----+
user2@debiansio:~$
```

Ici, on a créé la clé pour le user2. On obtient 2 fichiers, idrsa et idrsa.pub, c'est les deux clés, une publique et une privée.

```
root@debiansio:/home/user2/.ssh# ls
id_rsa  id_rsa.pub
root@debiansio:/home/user2/.ssh#
```

On peut donc maintenant se connecter au serveur en ssh !

Autoriser la connexion uniquement aux membres des groupes root et ssh

Pour autoriser uniquement les membres des groupes ssh et root, on modifie la configuration de ssh (/etc/ssh/sshd_config), on modifie la ligne suivante : AllowGroups ssh root

```
#ListenAddress 0.0.0.0
#ListenAddress ::
AllowedGroups ssh
```

Maintenant on redémarre le service ssh : systemctl restart sshd

Pour vérifier le bon fonctionnement, on se connecte en ssh avec le user1, et le user3 ne devrait pas fonctionner puisqu'il n'appartient pas au groupe ssh. On testera aussi avec root.

Il peut être nécessaire d'ajouter au root au groupe « ssh ».

Connexion avec clés

Pour se connecter sans mot de passe et par le biais des clés, on met la variable PasswordAuthentication sur « no ». Désormais nous n'aurons plus besoin de mot de passe mais il faut se connecter avec un utilisateur possédant la clé publique.

```
AllowGroups ssh
PermitRootLogin yes
PasswordAuthentication no
```

Il est nécessaire de redémarrer le service ssh.

Un utilisateur n'ayant pas la clé publique, ne pourra pas se connecter.

```
PS C:\Users\bapti> ssh user3@192.168.100.113
Bienvenue sur la machine SSH de Baptiste Cousin !
user3@192.168.100.113: Permission denied (publickey).
PS C:\Users\bapti>
```