TP N° 4: TP Admin à distance SSH

Belerge Jérémy BTS SIO 1

But

Pouvoir administrer une machine linux de façon sécurisée.

Expérimenter les clés asymétriques

Utiliser SSL et observer son fonctionnement

Consignes:

-Définition

Le SSH c'est quoi ?

Le protocole Secure Shell (SSH) est une méthode permettant d'envoyer en toute sécurité des commandes à un ordinateur sur un réseau non sécurisé. SSH a recours à la cryptographie pour authentifier et chiffrer les connexions entre les appareils

SSH s'exécute parallèlement à la suite de protocoles TCP/IP, sur laquelle repose une grande partie du site Internet. TCP signifie Transmission Control Protocol et IP Internet Protocol. TCP/IP associe ces deux protocoles afin de formater, acheminer et distribuer les paquets.

Source: www.cloudflare.com

Consignes:

-Comment installer le serveur

Comment installer le ssh sur notre srveur?

Pour cela il nous faut juste utiliser la commande 'apt install openssh-server' comme ci-dessous :

```
root@BELERGESSH:~# which ssh
/usr/bin/ssh
root@BELERGESSH:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
openssh-server is already the newest version (1:9.2p1-2+deb12u3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

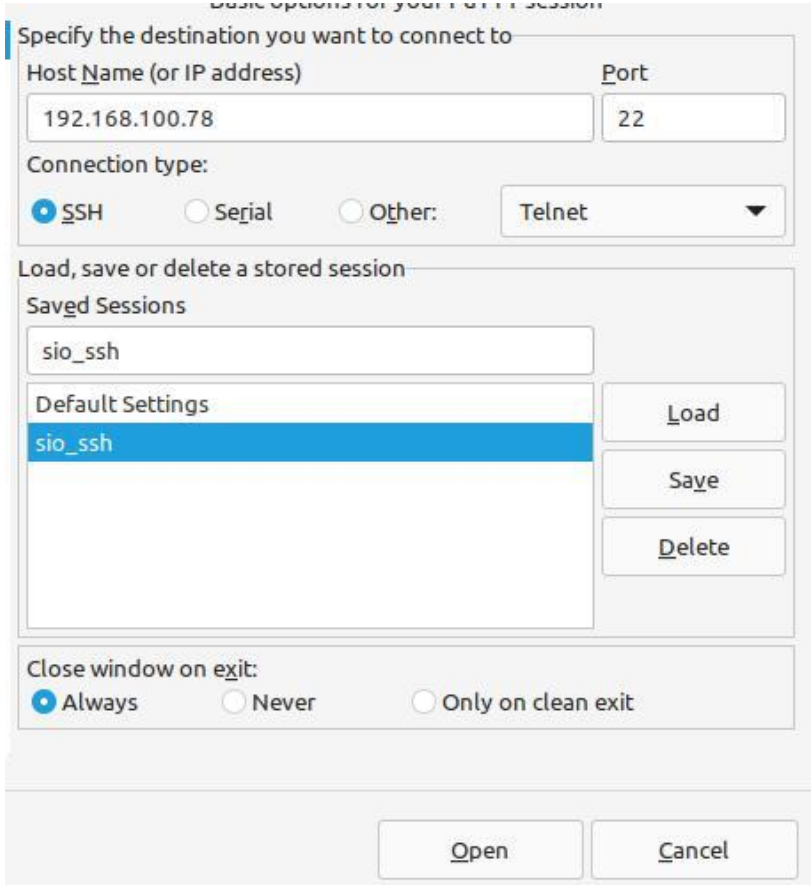
Consignes:

-Comment se connecter

On créer un user 'user' depuis notre serveur afin de pouvoir se connecter

```
root@BELERGESSH:/etc# adduser user
Adding user `user'
```

Pour se connecter en ssh depuis Putty il faut entrer l'ip de notre serveur (visible depuis un 'ip a' depuis le serveur) ainsi que son port de connexion



Consignes:

-Remarque

Remarque :

On peut afficher le fichier etc/ssh/sshd_config mais nous ne pouvons pas écrire dessus

Car il s'agit d'une mesure de sécurité , 'user' n'a pas les permissions dites suddoers

Comment changer le port d'écoute du SSH ?

Consignes:

-Changer le port d'ecoute

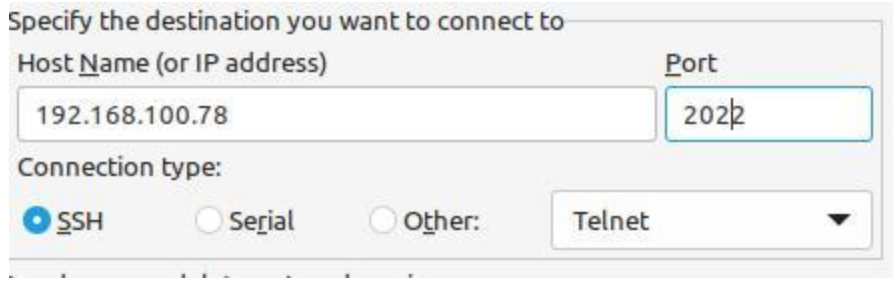
```
Include /etc/ssh/sshd_config.d/*.conf

Port 2022
#AddressFamily any
#ListenAddress 0.0.0.0
```

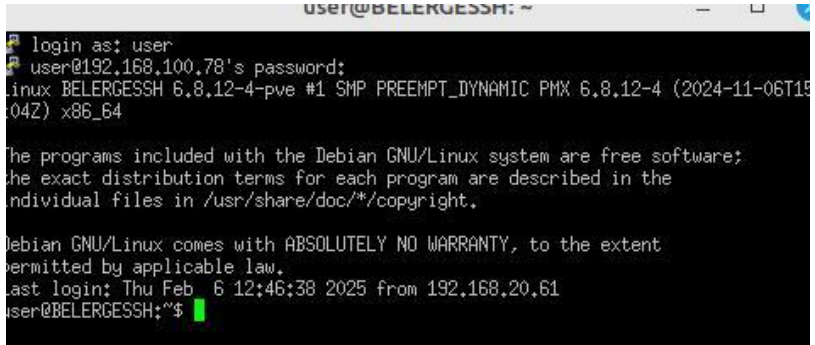
On modifie dans le fichier le port 22 en 2022 et en supprimant le commentaire (#)

```
root@BELERGESSH:/etc# systemctl restart ssh
```

Puis on redemarre notre service



On essaye de se connecter depuis Putty via notre nouveau port



On remarque alors que le port est opérationnel

Consignes:

-L'interet de changer le port

L'intérêt de changer le port ?

Changer le port d'écoute d'un serveur Linux permet de :

-Améliorer la sécurité (éviter les attaques sur les ports connus).
-Éviter les conflits entre services utilisant le même port.

On peut savoir quel est notre port grâce a la commande nmap:

```
Nmap scan report for 192.168.100.78
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE
2022/tcp open  down
```

Consignes:

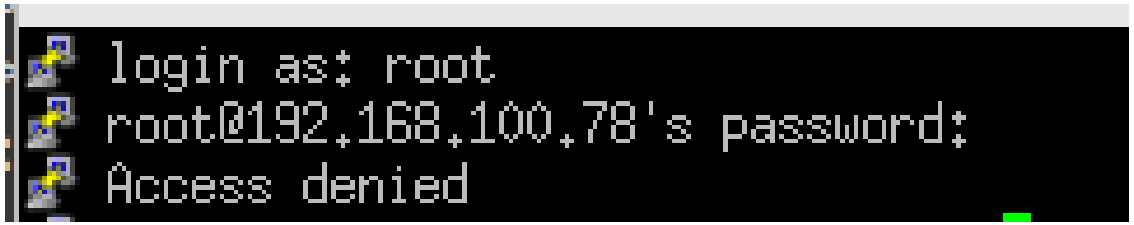
-L'importance des permissions root

Pourquoi la permission root est importante a maitriser?

Cela permet d'éviter tout accés direct a root depuis le SSH.

Donc d'évite des erreurs ou la création de panne ,etde renforcer la sécurité

Exemple depuis putty lorsqu'on essaye de se connecter avec le root:



Consignes:

-PermitEmptyPassword no et PermitRootLogin without-passwor

PermitEmptyPassword no et PermitRootLogin without-password?

-PermitEmptyPasswords no permet est utiliser pour interdire l'utilisation des mots de passe vide lors d'une connexion vide

-PermitRootLogin without-password permet lui d'accéder directement a l'utilisateur root sans mot de passe depuis le SSH mais uniquement depuis une clé SSH

Une clé SSH **(Secure Shell)** est une méthode d'authentification permettant d'établir une connexion sécurisée entre des ordinateurs distants, par le biais du protocole SSH

Consignes:

-La création des user et des groupes

Comment créer des groupes et des user ?

```
root@BELERGESSH:/etc# groupadd etudiant
root@BELERGESSH:/etc# groupadd ssh
```

On créer un groupe grace a groupadd "nom" on peut créer nos groupes.

```
root@BELERGESSH:/etc# useradd -g etudiant -G ssh -m user1
```

-Useradd : permet de créer l'user
`-g etudiant` : Ajouter l'user `user1` au groupe `etudiant`
`-G ssh` : Ajoute `user1` également au groupe `ssh`
`-m` : Crée un répertoire personnel pour l'utilisateur

```
root@BELERGESSH:/etc# useradd -g ssh -m user2
root@BELERGESSH:/etc# useradd -g etudiant -m user3
```

-useradd : permet de créer l'user
-g ssh ou –g etudiant : permet d'ajouter un user au groupe ssh ou etudiant
-m: crée un reêrtoire personnel a l'user selectionner (user2 / user3)

Consignes:

-Changer les mots de passe

Comment changer les mots de passes des utilisateurs depuis le serveur?

```
root@BELERGESSH:/etc# echo "user1:Password1" | chpasswd
root@BELERGESSH:/etc# echo "user2:Password1" | chpasswd
root@BELERGESSH:/etc# echo "user3:Password1" | chpasswd
```

-echo : nous permet d'écrire dans une chaine de texte dans le terminal
-"user1:Password1" : est la chaine de texte
-user1: est l'utilisateur cibler
-"Password1" : est le nouveau mot de passe
- | : nous permet de rediriger vers une autre commande
-chpasswd : nous permet de modifier le mot de passe

Consignes:

-Créer un répertoire d'échange de clé publique

Comment créer un répertoire d'échange de nos clés publiques?

Pour créer un dossier .ssh dans le dossier user1 , user 2 et user3

```
t@debiansio:~# mkdir /home/user1/ssh /home/user2/.ssh /home/user3/.
```

Pour changer les droits:

```
ot@debiansio:~# chmod 0770 /home/user1/.ssh /home/user2/.ssh /home/user3/
```

Créer un dossier .ssh et changer les droits sur notre serveur :

```
root@debiansio:/home# mkdir .ssh
```

```
root@debiansio:/home# chmod 0770 .ssh
```

Mkdir nous permet de créer le dossier et chmod nous permet de changer les droits

770 correspond au droit attribuer , ici les droits sont :

7 (rwx) → Lecture (r), écriture (w) et exécution (x) pour le propriétaire.
7 (rwx) → Lecture, écriture et exécution pour le groupe.
0 (---) → Aucun accès pour les autres

Consignes:

-Comment générer nos clés

Comment générer les clé publiques et priver de notre machine ?

Pour générer les clés publiques et priver on utilise la commande :
Ssh-keygen –t dsa –f ~/.ssh/id_dsa

```
$ ssh-keygen -t dsa -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/.ssh/id_dsa
Your public key has been saved in /home/user1/.ssh/id_dsa.pub
The key fingerprint is:
SHA256:Qgp3r5HUF128XZAHRTtWb3eafCQAOZENS5EI51pHUxc user1@debiansio
The key's randomart image is:
+---[DSA 1024]----+
|       ....X%ooEB+|
|        +.o=+ooo *|
|   . . + + +.   .BB|
|    o = = o    .o*=|
|     . = S      + .|
|        +         . |
|       .           |
|                   |
|                   |
+----[SHA256]-----+
```

Consignes:

-Comment générer nos clés

Les fichiers généré apres cela sont :

```
root@debiansio:/home/user1/.ssh# ls
id_dsa  id_dsa.pub
```

Id_dsa représente la clé privée

Id_dsa.pub représente la clé publique

Ceci est normal car :
-La clé **privée** est utilisée pour signer les requêtes d'authentification SSH et elle reste sur votre machine. Elle doit être protégée car si quelqu'un d'autre y a accès, ils pourraient se faire passer pour vous.

-La clé **publique** est utilisée pour authentifier l'utilisateur auprès du serveur. Elle peut être librement partagée avec d'autres systèmes ou serveurs, et elle est ajoutée dans le fichier `~/.ssh/authorized_keys` sur le serveur pour vous permettre de vous y connecter sans mot de passe

Consignes:

-Transferer nos clés vers le serveur

Pour envoyer notre clé vers le serveur il nous faut utiliser la commande
' Ssh-copy-id –i ~/.ssh/id_dsa.pub user@ip du serveur
Comme ci-dessous

```
Connection to 192.168.100.111 closed.
root@debian:~/.ssh# ssh-copy-id -i ~/.ssh/id_dsa.pub user1@192.168.100.111
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_dsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
user1@192.168.100.111's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user1@192.168.100.111'"
and check to make sure that only the key(s) you wanted were added.
```

Pour vérifier que notre ficher est arriver il faut utiliser la commande
'cat ~/.ssh/id_rsa.pub' depuis notre serveur

Consignes:

-Comment se connecter en ssh depuis notre machine client

Comment se connecter en ssh depuis notre client ?

Pour cela on utilise la commande
'ssh user@ip du serveur -p le port '
Comme ci-dessous :

```
root@debian:~/.ssh# ssh user1@192.168.100.111 -p 22
user1@192.168.100.111's password:
Linux debiansio 6.1.0-30-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar  3 14:27:49 2025 from 192.168.20.84
$
```

Consignes:

-Comment les groupes et user

Autoriser des groupes et users?

On peut autoriser des groupes ou bien meme des utilisateurs afin de pouvoir accéder au SSH , pour cela on ce rend dans notre fichier de configuration puis on ajoute les lignes suivantes au fichier

'AllowGroups root ssh' qui permet d'autorisé les groupes
'AllowUsers user1 user2 root' qui permet d'autoriser les user.

Comme ci-dessous:

```
root@debiansio:/home/.ssh# nano /etc/ssh/sshd_config
```

```
AllowGroups root ssh
AllowUsers user1 user2 root
```

Consignes:

-Comment changer l'authentification

Changer l'authentification ?

Pour changer ceci , il nous faut retourner dans notre fichier de configuration:

```
root@debiansio:/home/user1# nano /etc/ssh/sshd_config
```

Puis trouver '#PasswordAuthentucation yes' :

```
#PasswordAuthentication yes
```

Il nous faut alors enlever le commentaire et le faire basculer sur no:

```
PasswordAuthentication no
```

Puis relancer notre service:

```
root@debiansio:/home/user1# systemctl restart sshd
```

Maintenant il nous faut la clé publique envoyer précédemment afin de pouvoir se connecter a distance sur notre serveur afin de pouvoir se connecter .