TP HYDRA

-VM-Défense et VM-Attaque créer

-Création de mon compte et ajout des droits sudo

Mon nom d'utilisateur sera cserpette avec mdp Csrp@2025!

Création du compte et mdp:

useradd -m -s /bin/bash cserpette

passwd cserpette

Ajout des droits sudo:

usermod -aG wheel cserpette

Pour vérifier que mon compte ai bien les droits sudo, je vais faire la commande :

groups cserpette

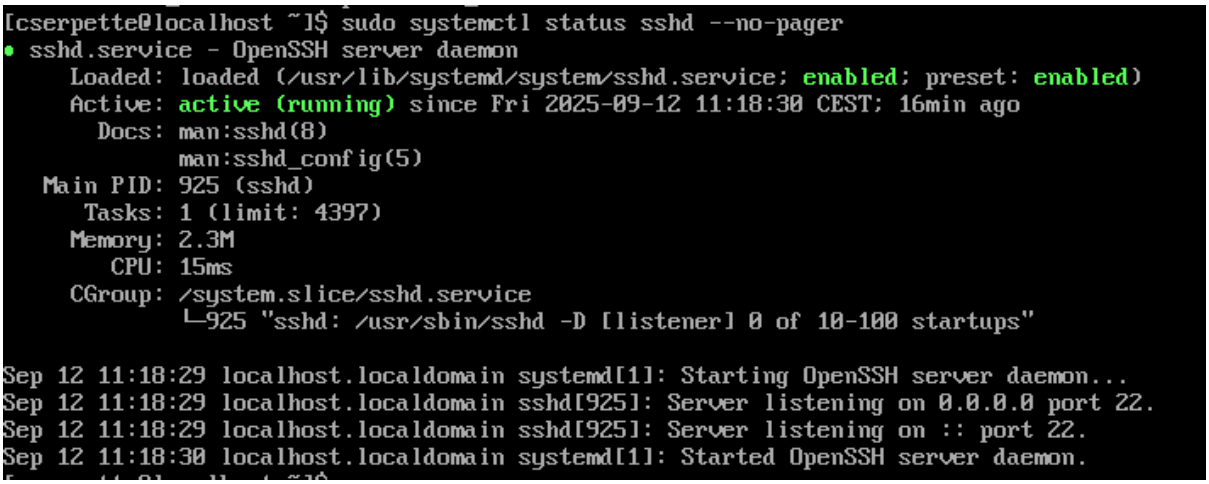
```
[root@localhost ~]# groups cserpette
cserpette : cserpette wheel
```

Mon compte cserpette est bien crée et il a les droits sudo

-Changement du port SSH vers le port 15000

Le port du SSH est pour le moment sur le port 22, je vais le prouver grâce à la commande :

sudo systemctl status sshd --no-pager



Les lignes

Server listening on 0.0.0.0 port 22.

Server listening on :: port 22.

stipule bien que mon SSH est sur le port 22.

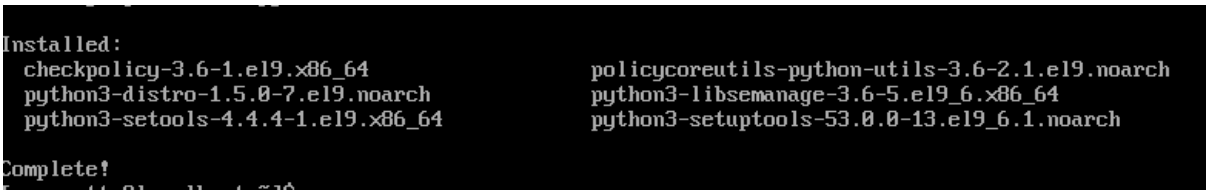
Maintenant, je vais ouvrir le port 15000 sans fermer le port 22 pour le moment pour avoir une porte de sortie au cas où j'ai un problème.

sudo firewall-cmd --add-port=15000/tcp --permanent

sudo firewall-cmd –reload

Installation de semanage : Cet outil n'est pas présent par défaut sur Rocky Linux, mais il est indispensable pour gérer les règles de SELinux. Comme SELinux bloque par défaut toute écoute SSH en dehors du port 22, j'ai installé policycoreutils-python-utils (qui fournit semanage) afin d'autoriser sshd à utiliser le port 15000. <- chatgpt

sudo dnf install -y policycoreutils-python-utils



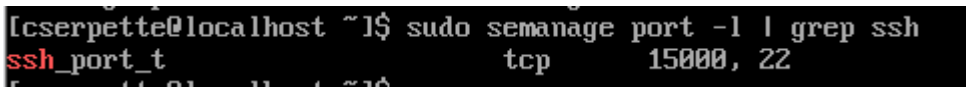
Je vais Autoriser le port côté SELinux pour éviter des problèmes suite au changement.

sudo semanage port --add -t ssh_port_t -p tcp 15000

Vérification des ports SELinux :

J'ai utilisé la commande :
sudo semanage port -l | grep ssh

Afin de lister les ports autorisés par SELinux pour le service SSH. Cette commande permet de confirmer que le port 15000 a bien été ajouté en plus du port 22.



Installation de l'éditeur "nano"

sudo dnf install -y nano

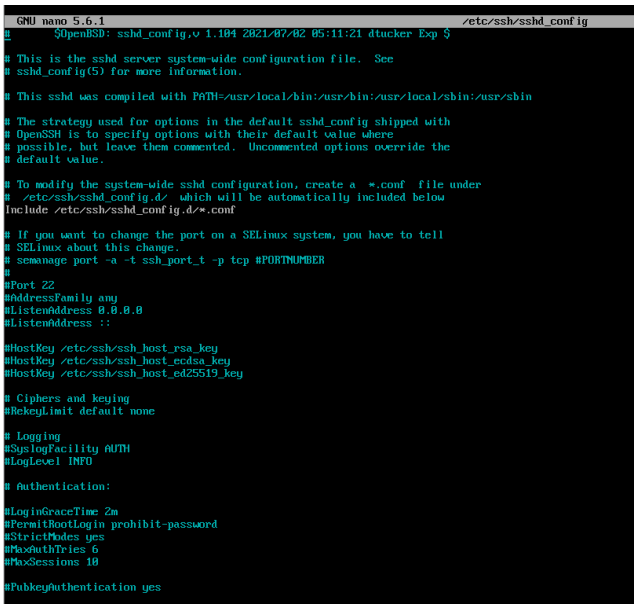
```
Installed:
  nano-5.6.1-7.el9.x86_64

Complete!
```

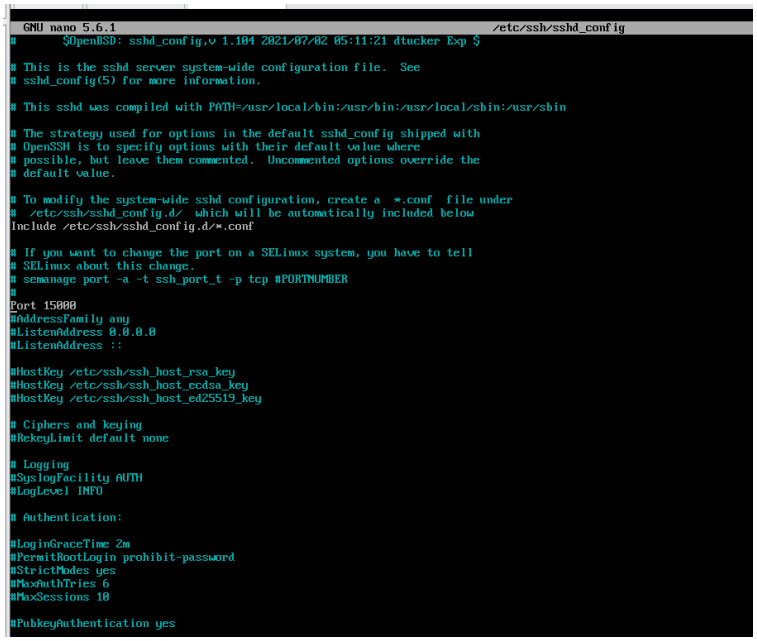
Changement du port ssh dans la configuration

sudo nano /etc/ssh/sshd_config

Dans Nano :

- Cherche la ligne #Port 22 (ou Port 22).
- Modifie-la ou ajoute en bas : Port 15000





Le port 22 est modifié par le port 15000 et j'ai enlevé le # pour qu'elle soit actif.

```
[cserpette@localhost ~]$ ssh -p 15000 localhost
The authenticity of host '[localhost]:15000 ([::1]:15000)' can't be established.
ED25519 key fingerprint is SHA256:k/P+8RNZRh5BgxQrr9CZMmE8iAAyQ5r4VnMKCRu6XK0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:15000' (ED25519) to the list of known hosts.
cserpette@localhost's password:
Last login: Fri Sep 26 10:59:03 2025
[cserpette@localhost ~]$ whoami
cserpette
[cserpette@localhost ~]$ exit
logout
Connection to localhost closed.
[cserpette@localhost ~]$ ssh cserpette@localhost
ssh: connect to host localhost port 22: Connection refused

You can find some explanations for typical errors at this link:
            https://red.ht/support_rhel_ssh
[cserpette@localhost ~]$
```

Test du changement de port SSH

ssh -p 15000 cserpette@localhost

->Connexion réussie, preuve que le service SSH écoute bien sur 15000.

ssh cserpette@localhost

-> Connexion échouée, preuve que le port 22 n'est plus utilisé.

Conclusion : le service SSH a bien été déplacé du port 22 vers le port 15000.

Configurer SELinux pour le port SSH 15000 (VM-DEFENSE)

But : autoriser sshd sur le port non-standard.

installer semanage si nécessaire

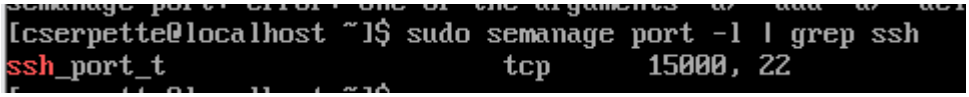
sudo dnf install -y policycoreutils-python-utils

ajouter le port 15000 pour ssh

sudo semanage port -a -t ssh_port_t -p tcp 15000

vérifier

sudo semanage port -l | grep ssh



sortie de sudo semanage port -l | grep ssh montrant 22, 15000.

Préparer VM-ATTAQUE : installer Hydra

Mon nom d'utilisateur sera cserpette avec mdp Attaque2025 !

mot de passe root : Csrp@2025!

*Le sharefolder est désactivé de base alors je dois penser à le remettre a chaque cours

Étape A — Mise à jour + installation d'hydra

1 mise à jour du système

sudo dnf clean all

sudo dnf -y update --refresh

2 installer EPEL (dépôt requis)

sudo dnf -y install epel-release

3 tenter l'installation directe d'hydra (si disponible)

sudo dnf -y install hydra openssh-clients || true

4 si hydra n'est pas fourni par le dépôt, compiler depuis la source

sudo dnf -y install git make gcc openssl-devel zlib-devel libssh2-devel libssh-devel libidn2-devel

git clone https://github.com/vanhauser-thc/thc-hydra.git

cd thc-hydra

./configure

make -j$(nproc)

sudo make install

5 vérification

hydra -h

J’ai activé et monté le dossier partagé VMware (dico1) sur la VM-ATTAQUE afin de récupérer le fichier crackstation.txt.gz depuis l’hôte. Ensuite j’ai copié l’archive dans mon répertoire personnel et j’ai extrait un échantillon rapide de 20 000 mots pour les tests afin de ne pas surcharger la VM

monter le partage (sur la VM-ATTAQUE)

sudo mkdir -p /mnt/hgfs

sudo /usr/bin/vmhgfs-fuse -o allow_other,auto_unmount .host:/ /mnt/hgfs

vérifier le contenu du partage et copier le fichier

ls -la /mnt/hgfs

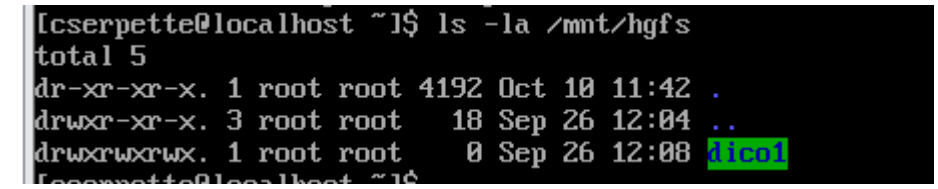
cp /mnt/hgfs/dico1/crackstation.txt.gz ~/

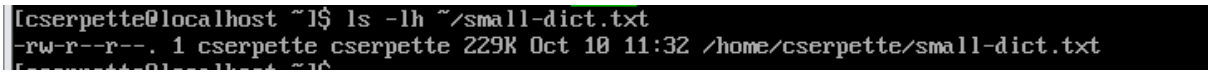
créer un petit dictionnaire d'essai (20k mots)

cd ~

zcat crackstation.txt.gz | head -n 20000 > small-dict.txt

ls -lh small-dict.txt





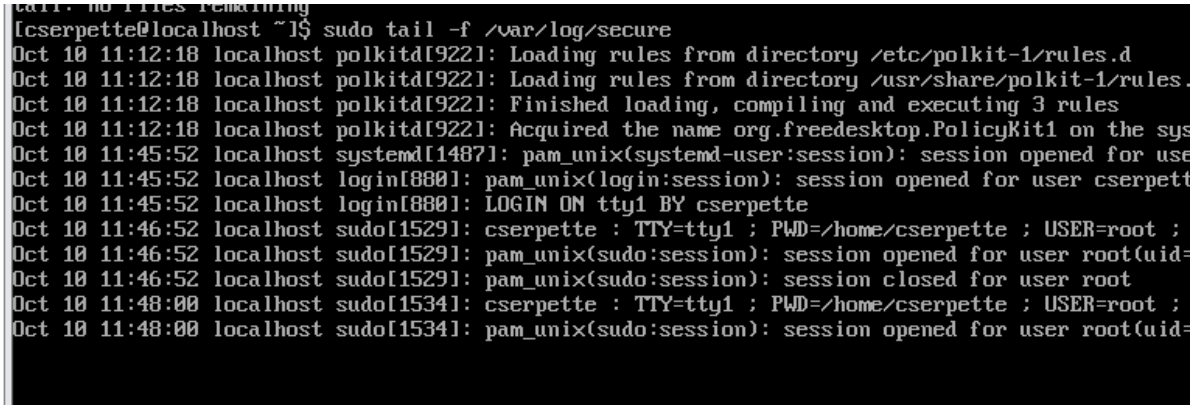
Début de l’attaque

ip -4 addr show

Ip vm defense 192.168.137.131

Sur VM-DEFENSE (ouvrir un terminal et laisser ouvert) :

sudo tail -f /var/log/secure



Sur VM-ATTAQUE (lancer Hydra) :

hydra -l cserpette -P ~/small-dict.txt -s 15000 -V 192.168.137.131

Ssh

- -l cserpette = compte visé
- -P ~/small-dict.txt = dictionnaire
- -s 15000 = port SSH

Fail2ban

Pour renforcer la sécurité, il est possible de mettre en place **Fail2Ban**. En effet, même si le port SSH a été modifié, un attaquant qui parvient à le découvrir pourra toujours tenter d'accéder à la machine.

Fail2Ban permet de bloquer automatiquement les adresses IP après plusieurs tentatives de connexion échouées.

Pour l'installer, connectez-vous en **root** puis exécutez la commande suivante :

sudo dnf install epel-release

Ensuite:

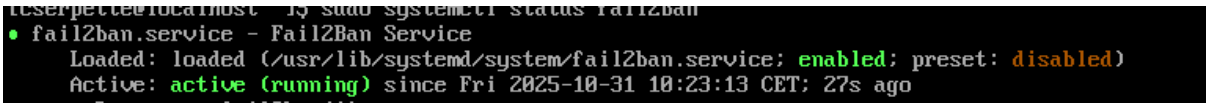
sudo dnf install fail2ban

On lance le service:

sudo systemctl start fail2ban

sudo systemctl enable fail2ban

sudo systemctl status fail2ban



Afin de configurer le fail2ban, on va créer un fichier custom.conf:

sudo nano /etc/fail2ban/jail.d/custom.conf

On active **Fail2Ban** pour le service **SSH** configuré sur le port **15000**.

Ainsi, toute adresse IP effectuant plus de **5 tentatives de connexion** échouées sera automatiquement **bannie**.

Authentification par clé SSH

La mise en place d'une authentification par clé SSH consiste à utiliser une paire de clés — une clé publique et une clé privée — afin de renforcer la sécurité des connexions SSH.

Cette méthode permet d'éviter l'utilisation de mots de passe, rendant ainsi l'accès au serveur plus sécurisé.

Pour générer une clé SSH, utilisez la commande suivante :

ssh-keygen -b 4096

On met la passphrase "Clement2025"