

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
```

```
[cserpette@localhost ~]# sudo systemctl status sshd --no-pager
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-09-12 11:18:30 CEST; 16min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 925 (sshd)
    Tasks: 1 (limit: 4397)
   Memory: 2.3M
     CPU: 15ms
   CGroup: /system.slice/sshd.service
           └─925 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 12 11:18:29 localhost.localdomain systemd[11]: Starting OpenSSH server daemon...
Sep 12 11:18:29 localhost.localdomain sshd[925]: Server listening on 0.0.0.0 port 22.
Sep 12 11:18:29 localhost.localdomain sshd[925]: Server listening on :: port 22.
Sep 12 11:18:30 localhost.localdomain systemd[11]: Started OpenSSH server daemon.
```

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
```

```
Installed:
  checkpolicy-3.6-1.e19.x86_64                policycoreutils-python-utils-3.6-2.1.e19.noarch
  python3-distro-1.5.0-7.e19.noarch          python3-libsemanage-3.6-5.e19_6.x86_64
  python3-setools-4.4.4-1.e19.x86_64        python3-setuptools-53.0.0-13.e19_6.1.noarch

Complete!
```

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.

```
[cserpette@localhost ~]$ sudo semanage port -l | grep ssh
ssh_port_t                tcp                15000, 22
```

Installation de l'éditeur "nano"

```
sudo dnf install -y nano
```

```
Installed:
nano-5.6.1-7.e19.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

```
GNU nano 5.6.1 /etc/ssh/sshd_config
# $OpenSSH: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
include /etc/ssh/sshd_config.d/*.conf
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#SyslogFacility AUTH
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
```

```
GNU nano 5.6.1 /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
include /etc/ssh/sshd_config.d/*.conf
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 15000
AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#SyslogFacility AUTH
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
```

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+8RMZRh5BgxQrr9CZmE8iAAyQ5r4UnMKCRu6XK0.
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

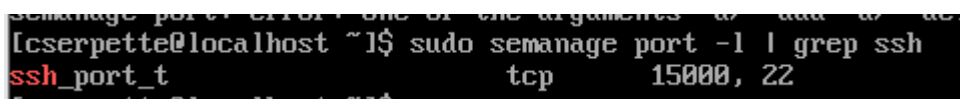
```
sudo dnf install -y polycoreutils-python-utils
```

ajouter le port 15000 pour ssh

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

vérifier

```
sudo semanage port -l | grep ssh
```



```
semmanage port -l | grep ssh  
[cserpette@localhost ~]$ sudo semanage port -l | grep ssh  
ssh_port_t tcp 15000, 22
```

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
```

```
Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.  
E.g. % export HYDRA_PROXY=socks5://1:p0127.0.0.1:9150 (or: socks4:// connect://)  
% export HYDRA_PROXY=connect_and_socks_proxylist.txt (up to 64 entries)  
% export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080  
% export HYDRA_PROXY_HTTP=proxylist.txt (up to 64 entries)  
  
Examples:  
hydra -l user -P passlist.txt ftp://192.168.0.1  
hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN  
hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5  
hydra -l admin -p password ftp://[192.168.0.0/24]/  
hydra -L logins.txt -P pws.txt -M targets.txt ssh  
[csernetfe@localhost ~]$
```

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
```

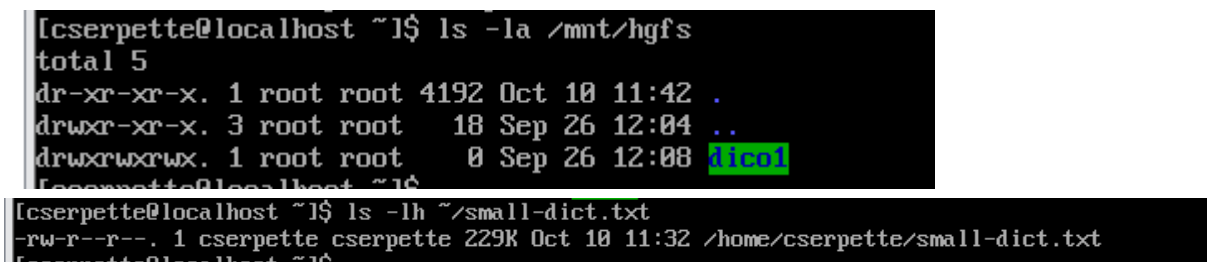
```
cp /mnt/hgfs/dico1/crackstation.txt.gz ~/
```

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

```
cd ~
```

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

```
ls -lh small-dict.txt
```



```
[cserpette@localhost ~]# ls -la /mnt/hgfs
total 5
dr-xr-xr-x. 1 root root 4192 Oct 10 11:42 .
drwxr-xr-x. 3 root root  18 Sep 26 12:04 ..
drwxrwxrwx. 1 root root   0 Sep 26 12:08 dico1
[cserpette@localhost ~]#

[cserpette@localhost ~]# ls -lh ~/small-dict.txt
-rw-r--r--. 1 cserpette cserpette 229K Oct 10 11:32 /home/cserpette/small-dict.txt
[cserpette@localhost ~]#
```

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

```
tail: no files remaining
[cserpette@localhost ~]$ sudo tail -f /var/log/secure
Oct 10 11:12:18 localhost polkitd[922]: Loading rules from directory /etc/polkit-1/rules.d
Oct 10 11:12:18 localhost polkitd[922]: Loading rules from directory /usr/share/polkit-1/rules.d
Oct 10 11:12:18 localhost polkitd[922]: Finished loading, compiling and executing 3 rules
Oct 10 11:12:18 localhost polkitd[922]: Acquired the name org.freedesktop.PolicyKit1 on the system bus
Oct 10 11:45:52 localhost systemd[1487]: pam_unix(systemd-user:session): session opened for user cserpette(uid=0)
Oct 10 11:45:52 localhost login[880]: pam_unix(login:session): session opened for user cserpette(uid=0)
Oct 10 11:45:52 localhost login[880]: LOGIN ON tty1 BY cserpette
Oct 10 11:46:52 localhost sudo[1529]: cserpette : TTY=tty1 ; PWD=/home/cserpette ; USER=root ; COMMAND=/usr/bin/sudo
Oct 10 11:46:52 localhost sudo[1529]: pam_unix(sudo:session): session opened for user root(uid=0)
Oct 10 11:46:52 localhost sudo[1529]: pam_unix(sudo:session): session closed for user root
Oct 10 11:48:00 localhost sudo[1534]: cserpette : TTY=tty1 ; PWD=/home/cserpette ; USER=root ; COMMAND=/usr/bin/sudo
Oct 10 11:48:00 localhost sudo[1534]: pam_unix(sudo:session): session opened for user root(uid=0)
```

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

```
Verifying... epel-release 9-7
Installed:
  epel-release-9-7.e19.noarch
Complete!
```

Ensuite:

sudo dnf install fail2ban

```
Installed:
  esmtp-1.2-19.e19.x86_64
  fail2ban-sendmail-1.1.0-6.e
Complete!
```

On lance le service:

sudo systemctl start fail2ban

sudo systemctl enable fail2ban

sudo systemctl status fail2ban

```
fail2ban.service - Fail2Ban Service
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-10-31 10:23:13 CET; 27s ago
```

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

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

```
GNU nano 5.6.1 /etc/fail2ban/jail.d/custom.conf
[sshd]
enable = true
port = 15000
logpath = /var/log/secure
maxretry = 5
```

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"

```
enter same passphrase again.
Your identification has been saved in /home/cserpette/.ssh/id_rsa
Your public key has been saved in /home/cserpette/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8+hMDumpIzTQShpIcknFob7Ze5vrewMLB2qb+2dxhEo cserpette@localhost.localdomain
The key's randomart image is:
----[RSA 4096]-----+
  ..+o. |
  ..o.. |
+o. . |
+o. E . |
o+. .S. |
o o+ ..o.+ |
 .o.+ + +o. |
  ...=.%* |
  .*BXB=o |
-----[SHA256]-----+
cserpette@localhost ~1$
```