Compte Rendu TP Rsyslog

1. Installation et configuration de RSyslog sur le serveur de log

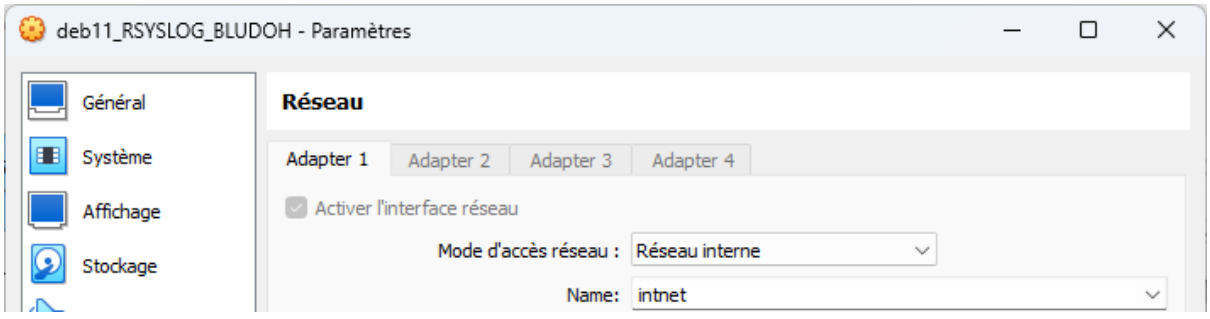
Tout d'abord, il faut s'assurer de la présence de rsyslog sur la machine en utilisant la commande "systemctl status rsyslog". Dans le cas présent, rsyslog était présent de base et actif.

```
Mot de passe :
root@debian11-Rsyslog-BLUDOH:~# sudo systemctl status rsyslog
● rsyslog.service - System Logging Service
     Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor prese>
     Active: active (running) since Mon 2025-05-19 10:50:23 CEST; 17min ago
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             man:rsyslog.conf(5)
             https://www.rsyslog.com/doc/
   Main PID: 317 (rsyslogd)
      Tasks: 4 (limit: 2321)
     Memory: 3.1M
        CPU: 49ms
     CGroup: /system.slice/rsyslog.service
             └─317 /usr/sbin/rsyslogd -n -iNONE

mai 19 10:50:23 debian systemd[1]: Starting System Logging Service...
mai 19 10:50:23 debian rsyslogd[317]: imuxsock: Acquired UNIX socket '/run/syst>
mai 19 10:50:23 debian rsyslogd[317]: [origin software="rsyslogd" swVersion="8.>
mai 19 10:50:23 debian systemd[1]: Started System Logging Service.
```

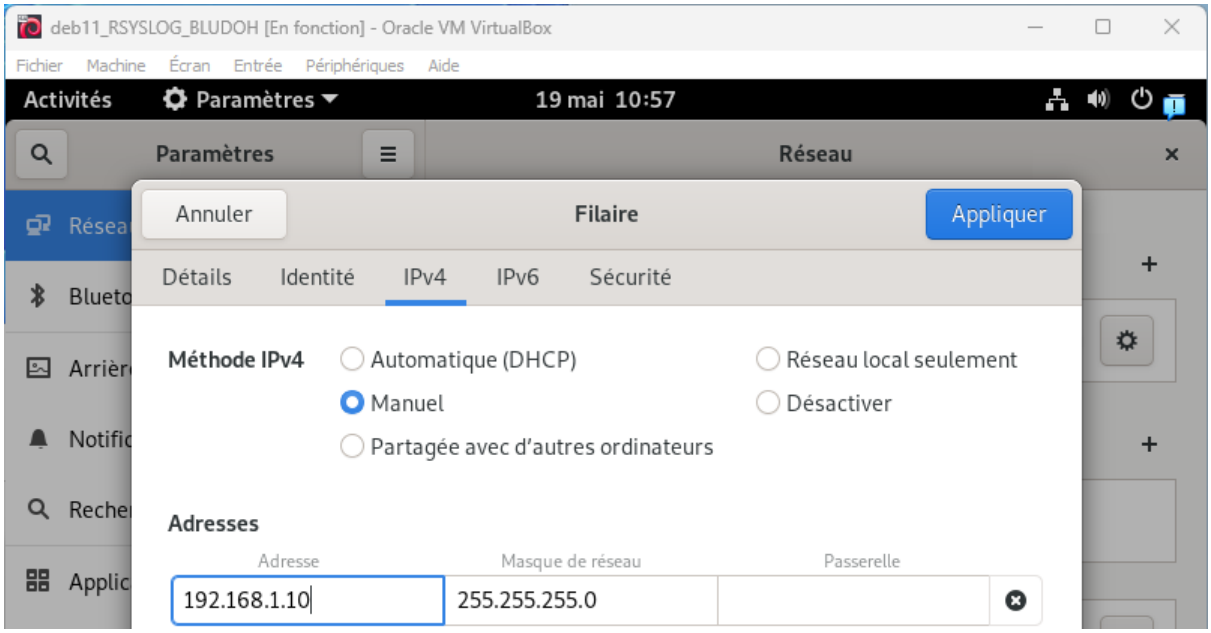
Puis on change le nom d'hôte de la machine avec la commande "hostnamectl set-hostname [nom]" (ici debian11-Rsyslog-BLUDOH) et on la redémarre :

```
eleve@debian:~$ hostnamectl set-hostname debian11-Rsyslog-BLUDOH
eleve@debian:~$ hostname
debian11-Rsyslog-BLUDOH

eleve@debian:~$ systemctl reboot
```

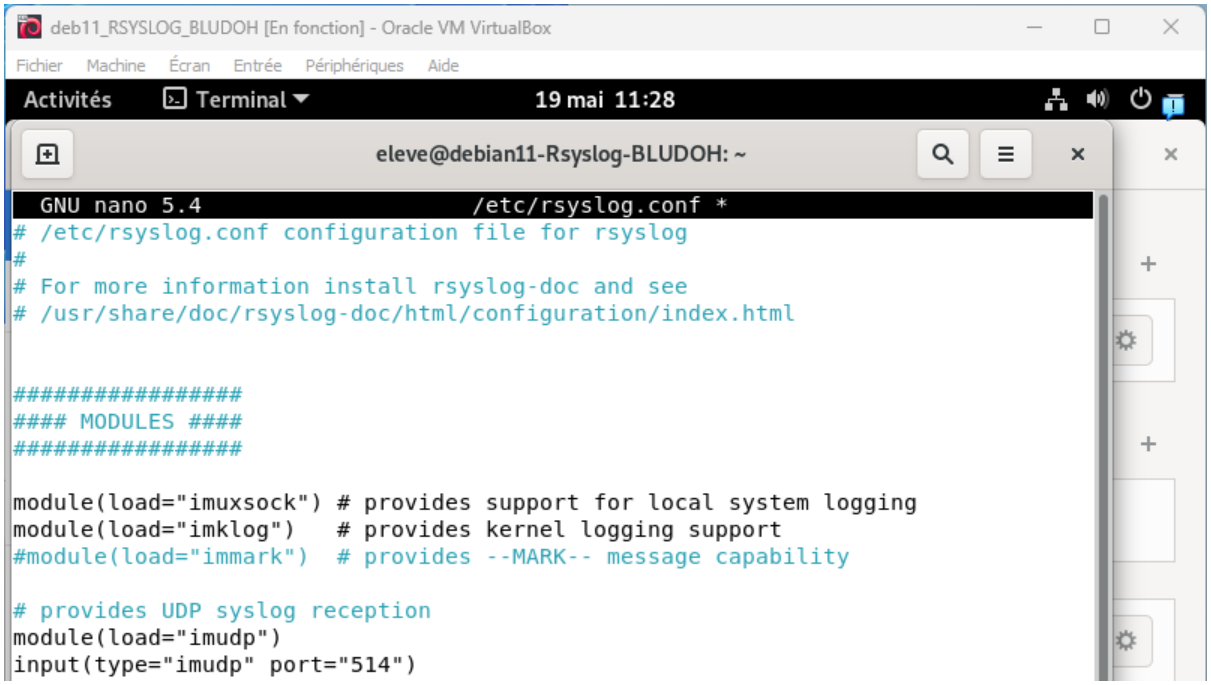
On passe la VM en réseau interne :



Et on change l'adresse IP en 192.168.1.10 avec le masque 255.255.255.0 :



Pour permettre au serveur de recevoir les logs des autres serveurs, on va décommenter les lignes “module(load=”imudp”)” et “input(type=”imudp” port=”514”)”.

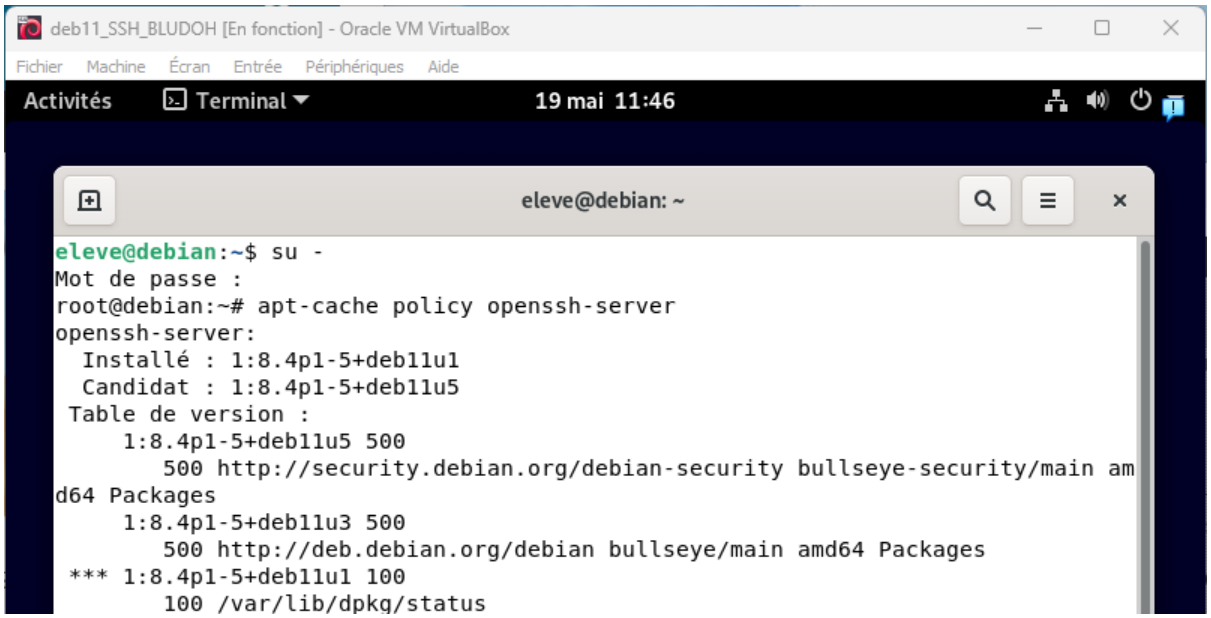


On redémarre Rsyslog avec ”systemctl restart rsyslog” et on vérifie bien que Rsyslog est en écoute sur le port 514 :

```
root@debian11-Rsyslog-BLUDOH:~# sudo systemctl restart rsyslog
root@debian11-Rsyslog-BLUDOH:~# sudo ss -tunlp | grep 514
udp   UNCONN 0      0            0.0.0.0:514        0.0.0.0:*    users:(("rsyslo
gd",pid=1805,fd=6))
udp   UNCONN 0      0               [::]:514           [::]:*    users:(("rsyslo
gd",pid=1805,fd=7))
```

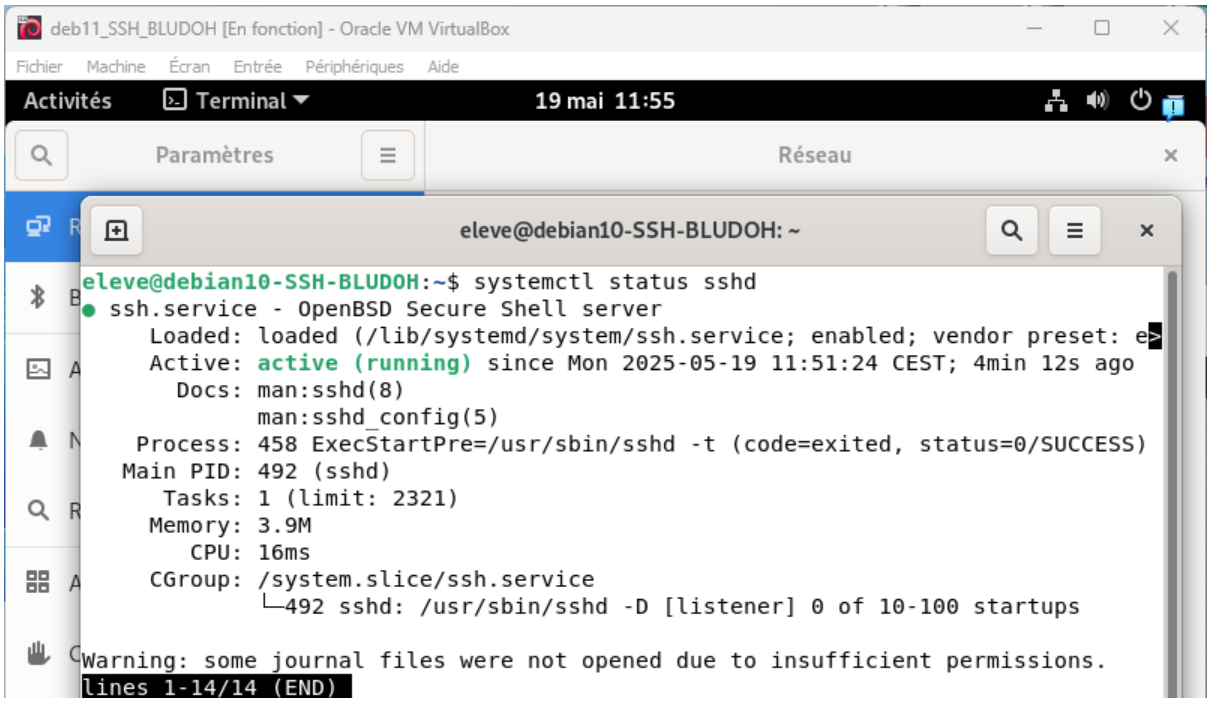
2. Configuration du serveur SSH

Les commandes “apt-cache policy openssh-server" et “ls –al /etc/ssh/sshd_config” sont deux manières de vérifier que SSH est bien présent sur la machine.



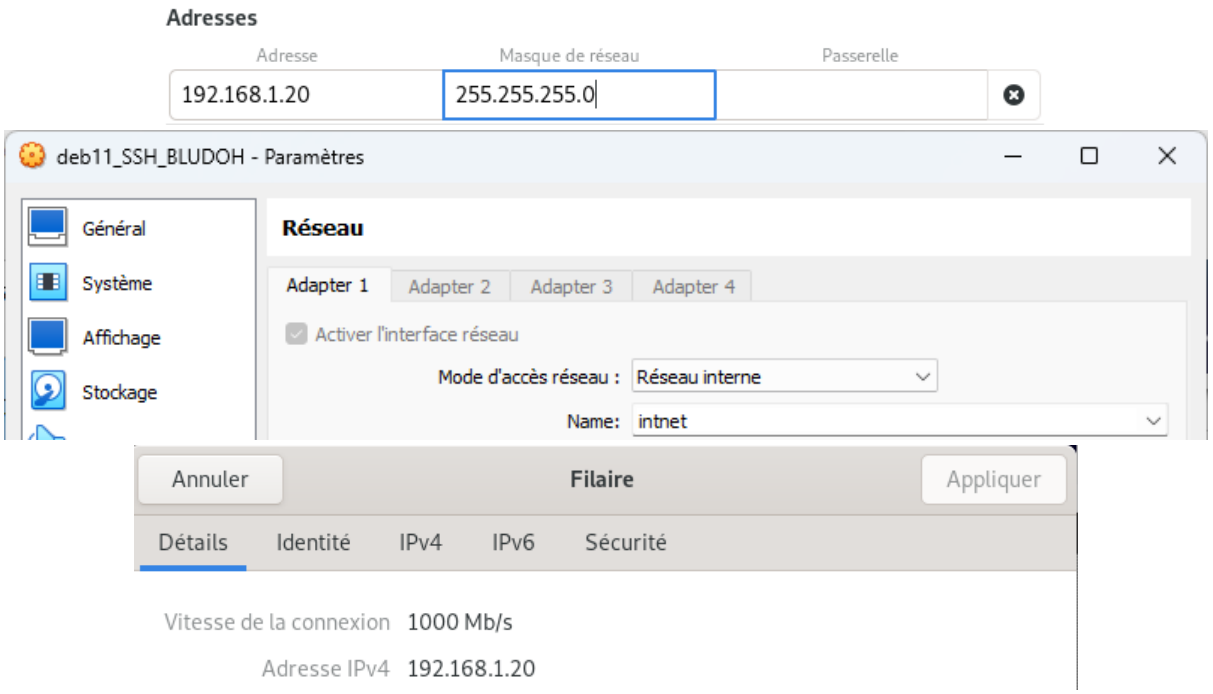

```
root@debian:~# ls -al /etc/ssh/sshd_config
-rw-r--r-- 1 root root 3289  2 juil.  2022 /etc/ssh/sshd_config
```

Et “systemctl status sshd” pour vérifier si SSH est actif ou non.



On change le nom d’hôte et on redémarre la machine, on passe en réseau interne et on met une adresse IP fixe :

```
eleve@debian:~$ hostnamectl set-hostname debian10-SSH-BLUDOH
eleve@debian:~$ hostnamectl
   Static hostname: debian10-SSH-BLUDOH
         Icon name: computer-vm
           Chassis: vm
        Machine ID: e678f67136624fc19e0566d2fe22d0f8
           Boot ID: 7b76cd1098f84595a3f25a7438e919bb
    Virtualization: oracle
  Operating System: Debian GNU/Linux 11 (bullseye)
            Kernel: Linux 5.10.0-20-amd64
      Architecture: x86-64

eleve@debian:~$ systemctl reboot
```

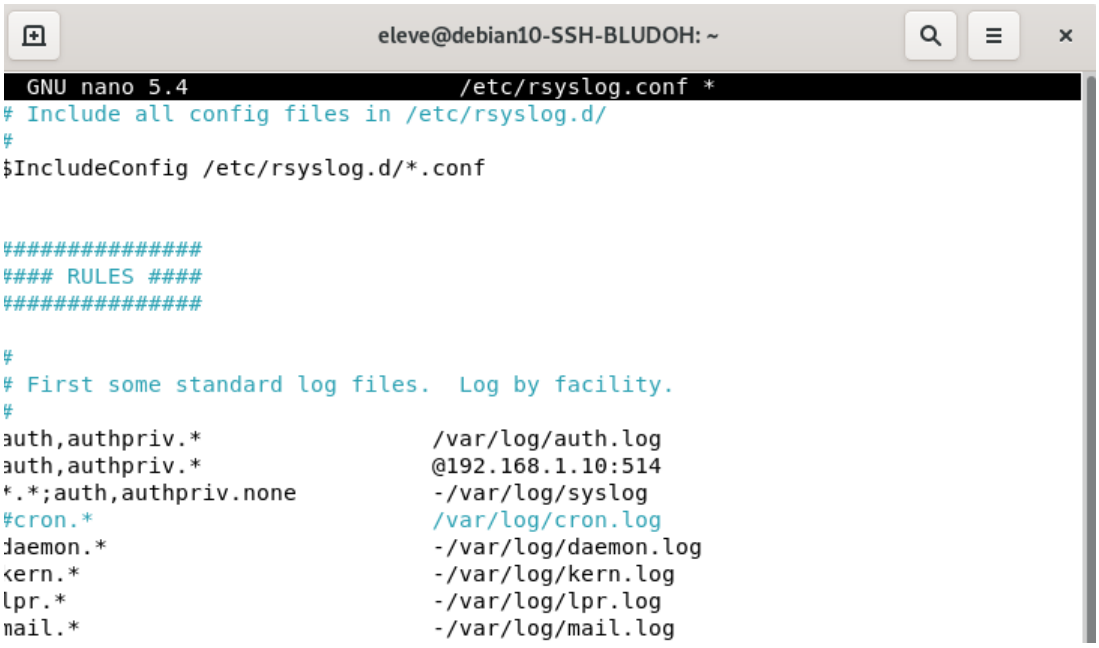


Pour autoriser la connexion a SSH avec un mot de passe, on décommente la ligne “PasswordAuthentication yes”:

On vérifie que l’écoute se fait bien sur le port 22 :

```
root@debian10-SSH-BLUDOH:~# ss -tuln | grep :22
tcp   LISTEN 0      128          0.0.0.0:22         0.0.0.0:*
tcp   LISTEN 0      128             [::]:22            [::]:*
```

Pour permettre l’envoi des logs a Rsyslog, on ajoute dans le fichier rsyslog.conf une ligne supplémentaire pour “auth, authpriv.*” avec @192.168.1.10:514 ; l’arobase est pour le protocole UDP, 514 est le port et l’adresse IP est celle du serveur Rsyslog.



On crée un nouvel utilisateur appelé johndoe qui servira de test pour l'attaque Hydra.

```
root@debian10-SSH-BLUDOH:~# adduser johndoe
Ajout de l'utilisateur « johndoe » ...
Ajout du nouveau groupe « johndoe » (1002) ...
Ajout du nouvel utilisateur « johndoe » (1001) avec le groupe « johndoe » ...
Création du répertoire personnel « /home/johndoe »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for johndoe
Enter the new value, or press ENTER for the default
        Full Name []: John Doe
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Cette information est-elle correcte ? [O/n]O
root@debian10-SSH-BLUDOH:~# nano /etc/ssh/sshd_config
root@debian10-SSH-BLUDOH:~# service ssh restart
```

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server

AllowUsers eleve johndoe
```

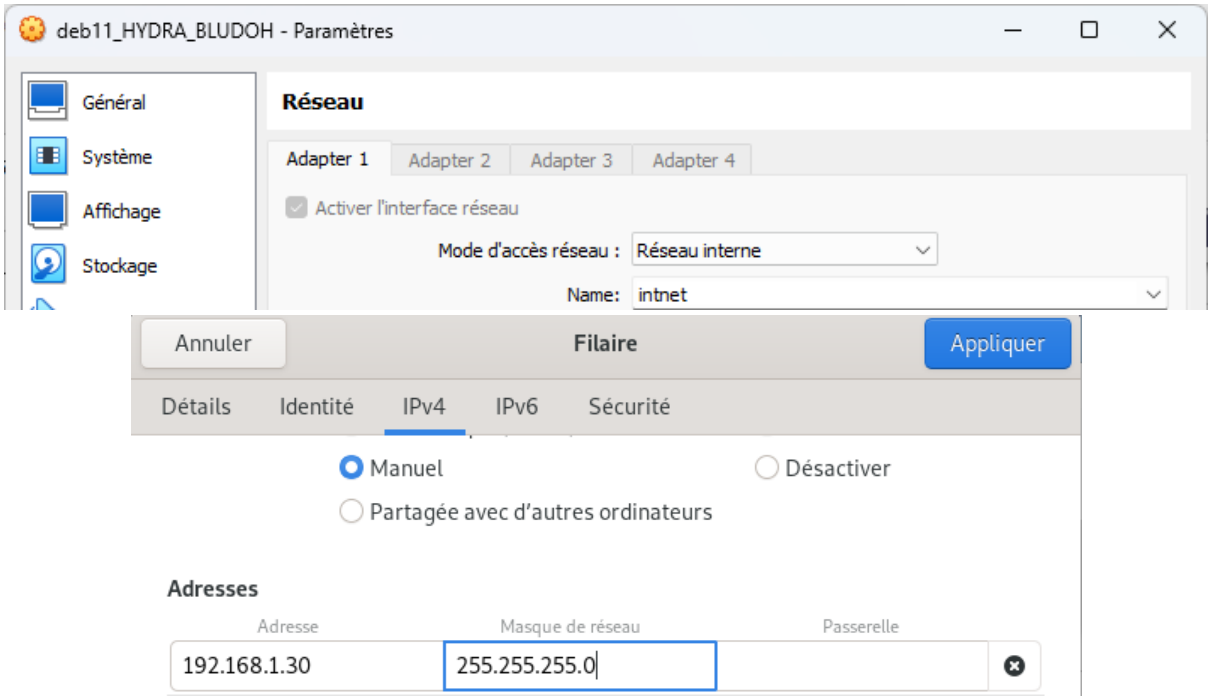
3. Simulation d'une attaque brute force avec Hydra

On exécute toutes les manœuvres précédentes concernant le nom d'hôte, le reseau interne, le changement d'adresse IP, etc.

```
eleve@debian:~$ hostnamectl set-hostname debian11-Hydra-BLUDOH
eleve@debian:~$ hostname
debian11-Hydra-BLUDOH
```

Et on installe Hydra avec la commande "sudo apt install hydra" :

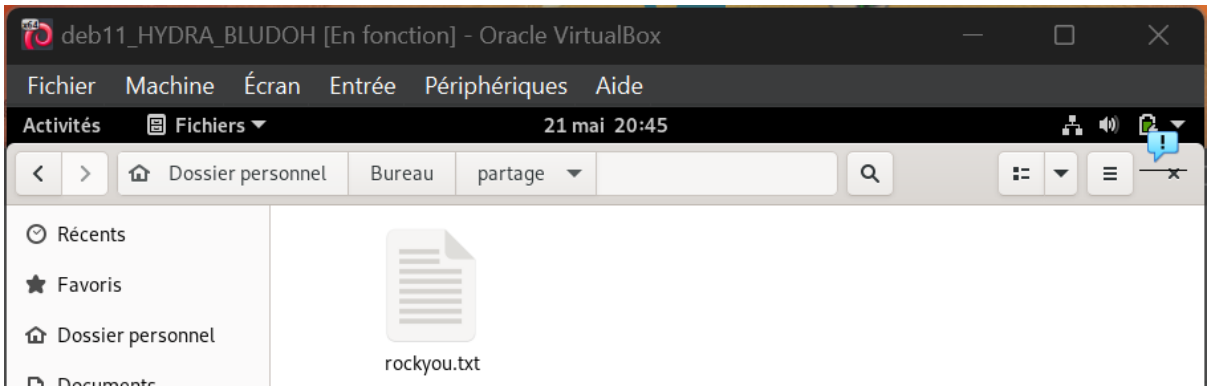
```
root@debian11-Hydra-BLUDOH:~# sudo apt install hydra
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessair
es :
  libdbus-glib-1-2 libopengl0 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  firebird3.0-common firebird3.0-common-doc libbson-1.0-0 libfbclient2
  libmariadb3 libmemcached11 libmongoc-1.0-0 libmongocrypt0 libpq5 libserf-1-1
  libssh-4 libsvn1 libtommath1 libutf8proc2 mariadb-common mysql-common
Paquets suggérés :
  hydra-gtk
Les NOUVEAUX paquets suivants seront installés :
  firebird3.0-common firebird3.0-common-doc hydra libbson-1.0-0 libfbclient2
  libmariadb3 libmemcached11 libmongoc-1.0-0 libmongocrypt0 libpq5 libserf-1-1
  libssh-4 libsvn1 libtommath1 libutf8proc2 mariadb-common mysql-common
0 mis à jour, 17 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 3 644 ko dans les archives.
Après cette opération, 11,8 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```



Un test de ping pour voir si la machine communique bine avec les deux autres sont à faire ensuite :

```
eleve@debian11-Hydra-BLUDOH:~$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.844 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=1.18 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.974 ms
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.844/0.999/1.181/0.138 ms
eleve@debian11-Hydra-BLUDOH:~$ ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=64 time=0.735 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=64 time=0.922 ms
^C
--- 192.168.1.20 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.735/0.828/0.922/0.093 ms
```

On récupère le fichier rockyou et on le partage pour l'obtenir sur la machine Hydra :



Et on lance hydra avec le chemin d'accès au fichier et le nom d'utilisateur qui va se connecter au SSH :

```
joachim@debian11-Hydra-BLUDOH:~$ hydra -l johndoe -P /home/joachim/Bureau/partag
e/rockyou.txt ssh://192.168.1.20
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-20 17:28:
06
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:
14344398), ~896525 tries per task
[DATA] attacking ssh://192.168.1.20:22/
```

On vérifie les logs sur Rsyslog avec la commande "tail –f /var/log/auth.log" :

```
root@debian11-Rsyslog-BLUDOH:~# tail -f /var/log/auth.log
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2030]: PAM 5 more authentication failur
es; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.30  user=johndoe
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2030]: PAM service(sshd) ignoring max r
etries; 6 > 3
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2033]: error: maximum authentication at
tempts exceeded for johndoe from 192.168.1.30 port 59806 ssh2 [preauth]
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2033]: Disconnecting authenticating use
r johndoe 192.168.1.30 port 59806: Too many authentication failures [preauth]
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2033]: PAM 5 more authentication failur
es; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.30  user=johndoe
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2033]: PAM service(sshd) ignoring max r
etries; 6 > 3
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2037]: error: maximum authentication at
tempts exceeded for johndoe from 192.168.1.30 port 59822 ssh2 [preauth]
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2037]: Disconnecting authenticating use
r johndoe 192.168.1.30 port 59822: Too many authentication failures [preauth]
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2037]: PAM 5 more authentication failur
es; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.30  user=johndoe
May 20 17:29:16 debian10-SSH-BLUDOH sshd[2037]: PAM service(sshd) ignoring max r
etries; 6 > 3
```

4. Analyse des logs centralisés

On note qu'il y a un grand nombre de connexions du compte johndoe avec l'adresse IP 192.168.1.30 qui est celle du serveur Hydra. Il y a des lignes indiquant que le nombre d'authentifications maximum a été dépassé.

5. Conclusion

La centralisation des logs permet d'avoir toutes les informations au même endroit et ainsi si plusieurs systèmes sont affectés, suivre la logique de l'attaque (d'où elle part, ce qu'elle affecte, etc.) plus facilement.

Les mesures de sécurité pouvant être employées sont de verrouiller le compte pendant un certain temps après un trop grand nombre d'échecs.

6. Implémentation de fail2ban pour sécuriser le serveur SSH