# COMPTE RENDU

## WIRESHARK SNIFF ROUTER TRAFFIC

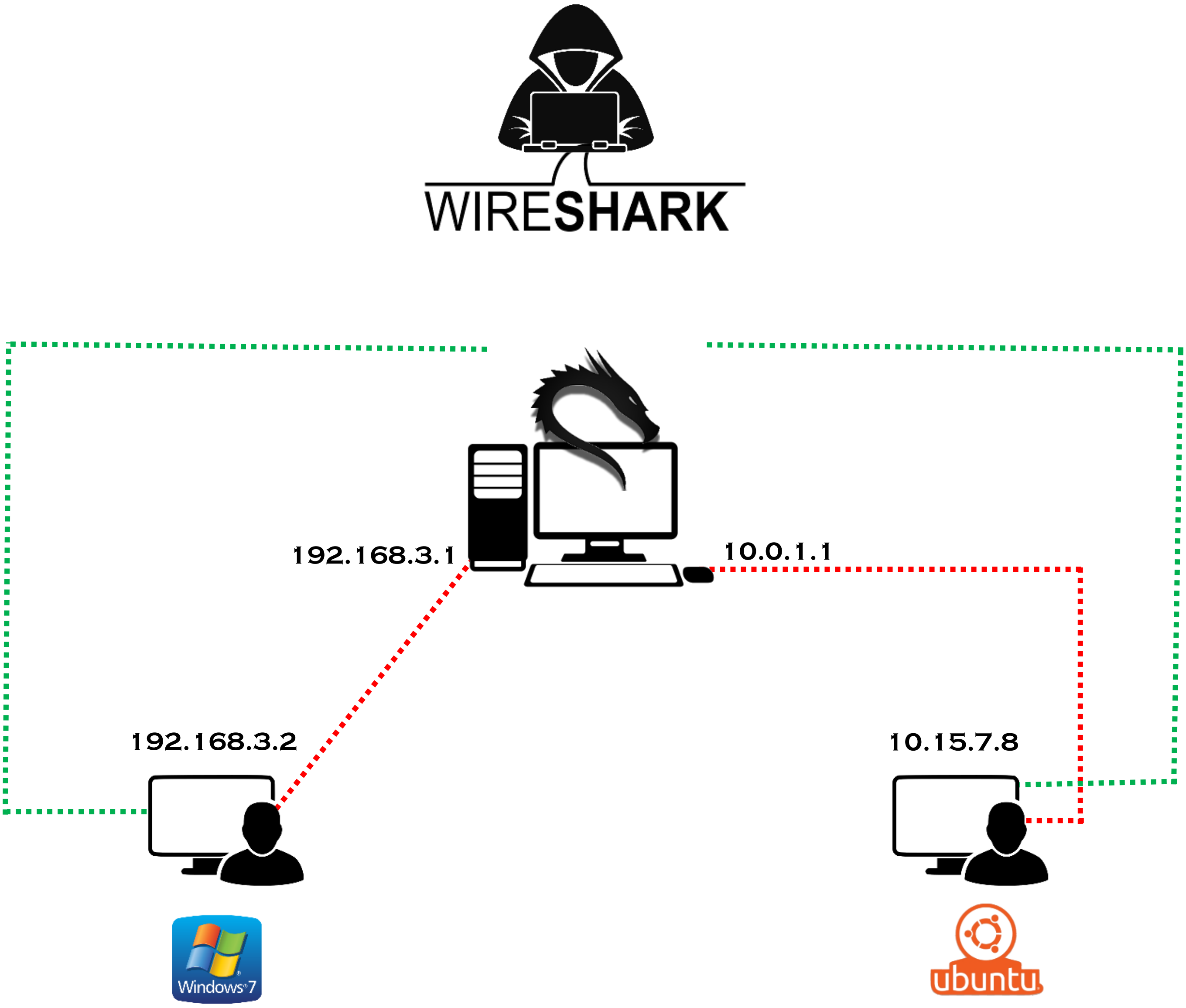
**Réalise par**

OTHMANE TAYBI

**Encadré par**

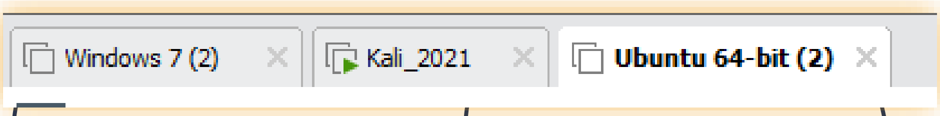
M.BENSLIMANE

## ★ Le réseau :


WIRESHARK
192.168.3.1
10.0.1.1
192.168.3.2
10.15.7.8
Windows 7
ubuntu

## Configuration des Machine Virtual :

- On créer 3 machines Virtual :



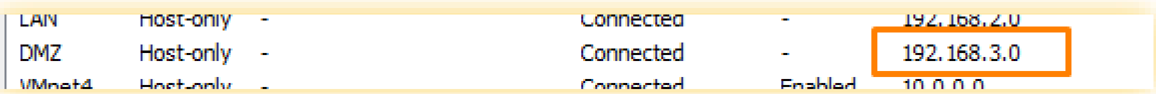
Machine 1 [Windows 7]

Machine 2 [Kali linux : joue le rôle de routeur]

Machine 3 [Ubuntu]

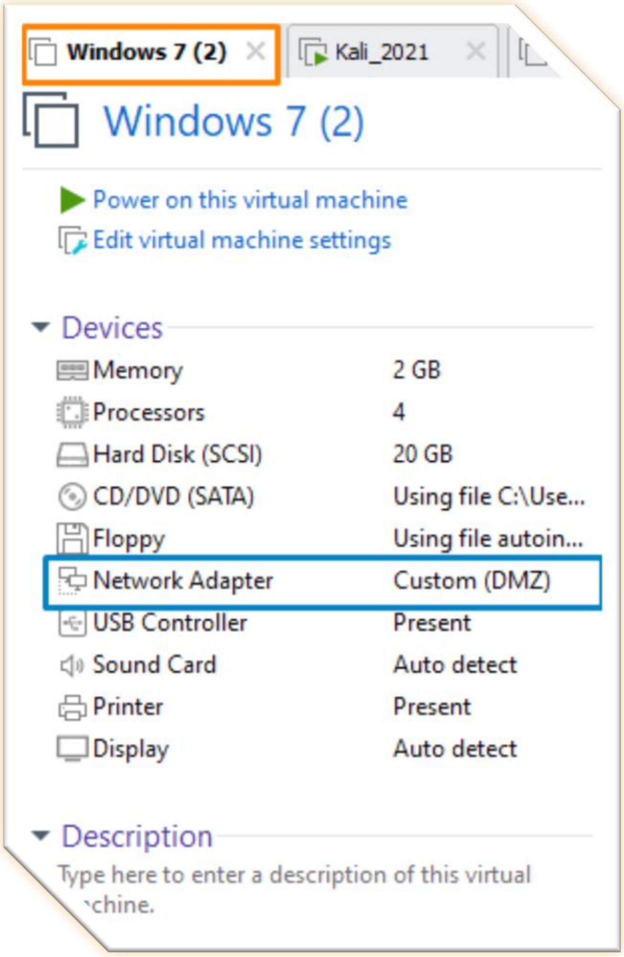
- **Configuration de Machine 1 [Windows 7] :**

  ✓ On créer une carte réseau qui s'appelle DMZ [192.168.3.0],



| | | | | | |
|---|---|---|---|---|---|
| LAN | Host-only | - | Connected | - | 192.168.2.0 |
| DMZ | Host-only | - | Connected | - | 192.168.3.0 |
| VMnet4 | Host-only | - | Connected | Enabled | 10.0.0.0 |

✓ Et en configurer l'interfaces de la machine 1 [Windows7] a cette carte réseau :



Windows 7 (2)

Power on this virtual machine
Edit virtual machine settings

Devices

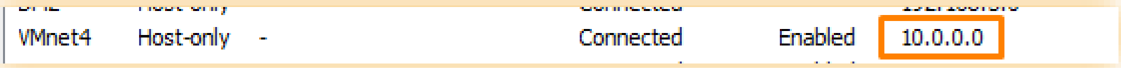
| | |
|---|---|
| Memory | 2 GB |
| Processors | 4 |
| Hard Disk (SCSI) | 20 GB |
| CD/DVD (SATA) | Using file C:\Use... |
| Floppy | Using file autoin... |
| Network Adapter | Custom (DMZ) |
| USB Controller | Present |
| Sound Card | Auto detect |
| Printer | Present |
| Display | Auto detect |

Description

Type here to enter a description of this virtual machine.

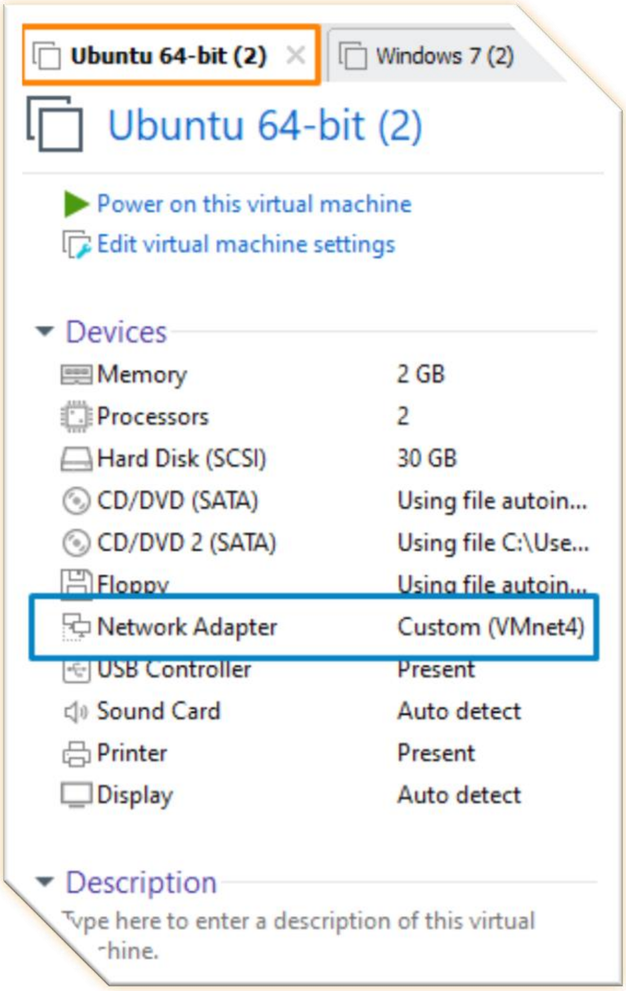
- **Configuration de Machine 3 [Ubuntu] :**

✓ On créer une carte réseau qui s’appelle VMnet4 [10.0.0.0],



| | | | | | |
|---|---|---|---|---|---|
| VMnet4 | Host-only | - | Connected | Enabled | 10.0.0.0 |

✓ Et en configurer l'interfaces de la machine 3 [Ubuntu] a cette carte réseau :



Ubuntu 64-bit (2)

Devices

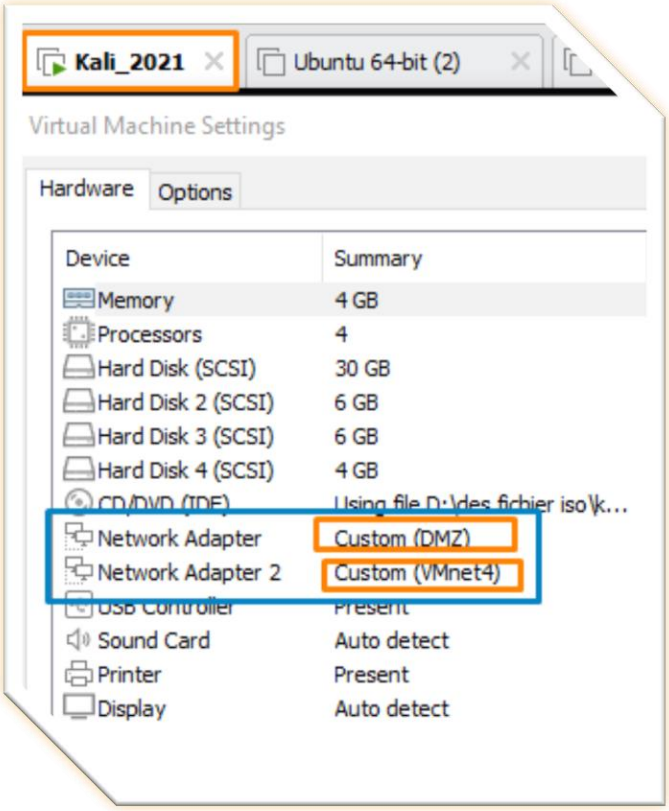
| | |
|---|---|
| Memory | 2 GB |
| Processors | 2 |
| Hard Disk (SCSI) | 30 GB |
| CD/DVD (SATA) | Using file autoin... |
| CD/DVD 2 (SATA) | Using file C:\Use... |
| Floppy | Using file autoin... |
| Network Adapter | Custom (VMnet4) |
| USB Controller | Present |
| Sound Card | Auto detect |
| Printer | Present |
| Display | Auto detect |

Description

Type here to enter a description of this virtual ~hine.

## Configuration de Machine 2 [Kali] :

✓ On configurer les l'interfaces de la machine 2 [Ubuntu] a les deux carte réseau [DMZ] et [VMnet4] :



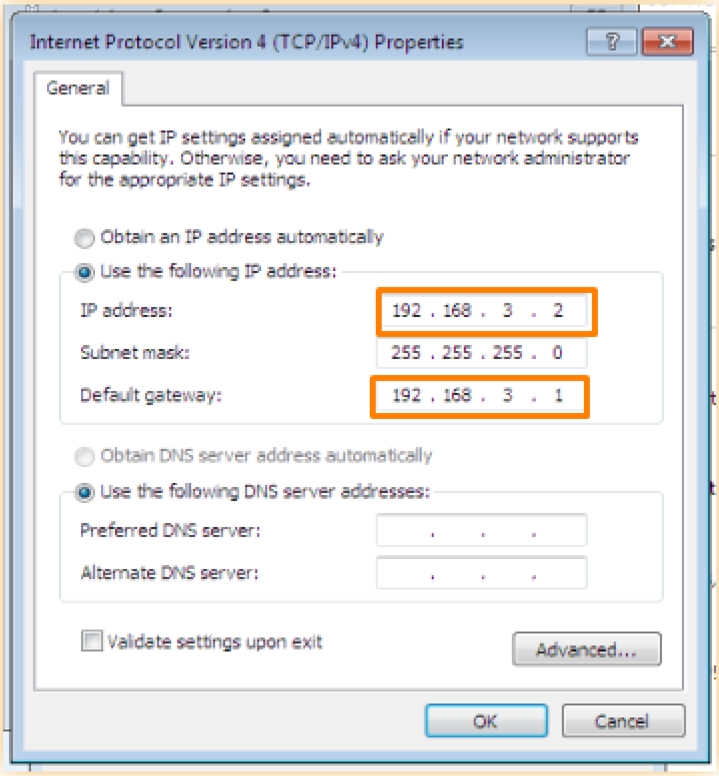
| Device | Summary |
|---|---|
| Memory | 4 GB |
| Processors | 4 |
| Hard Disk (SCSI) | 30 GB |
| Hard Disk 2 (SCSI) | 6 GB |
| Hard Disk 3 (SCSI) | 6 GB |
| Hard Disk 4 (SCSI) | 4 GB |
| CD/DVD (IDE) | Using file D:\des fichier iso\k... |
| Network Adapter | Custom (DMZ) |
| Network Adapter 2 | Custom (VMnet4) |
| USB Controller | Present |
| Sound Card | Auto detect |
| Printer | Present |
| Display | Auto detect |

## Configuration les interfaces des machines :

### Machine 1 [Windows 7] :

Adresse IP : 192.168.3.2

Gateway : 192.168.3.1 [L'interfaces eth0 de la machine 2]



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 3 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 3 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

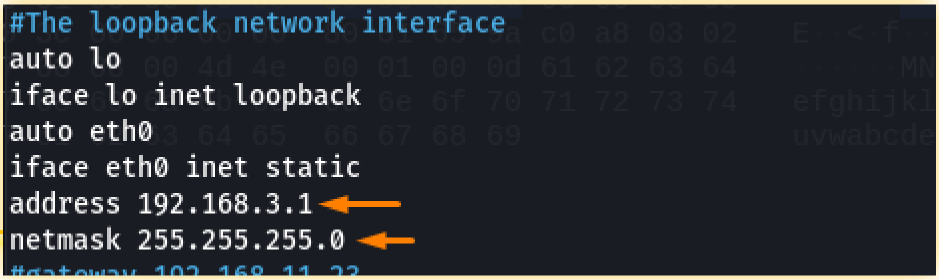
□ Validate settings upon exit

## Machine 2 [Kali 'Routeur'] :

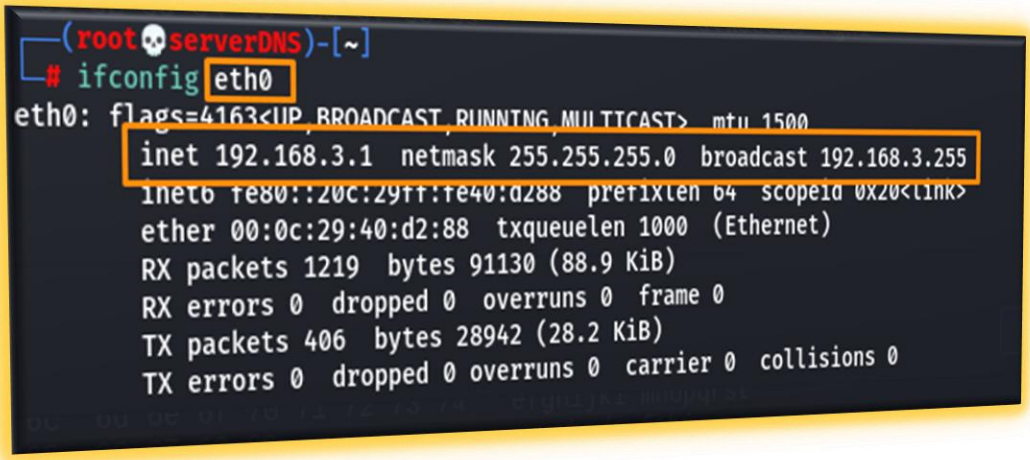
Interfaces 1 [eth0] :

Adresse IP : 192.168.3.1

Masque : 255.255.255.0



```
#The loopback network interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.3.1
netmask 255.255.255.0
#gateway 192.168.11.23
```

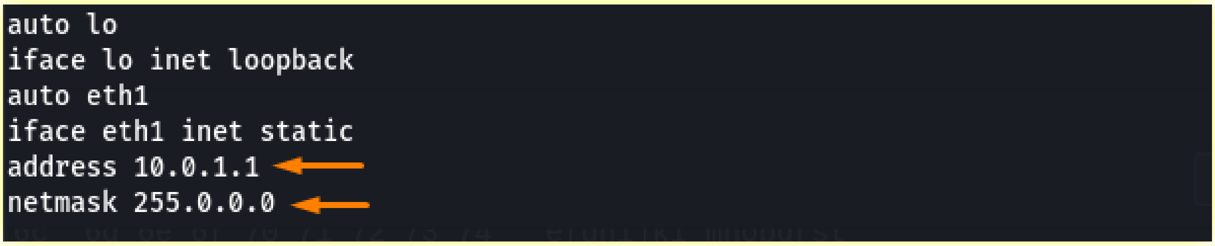


```
┌──(root💀serverDNS)-[~]
└─# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.1  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::20c:29ff:fe40:d288  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:d2:88  txqueuelen 1000  (Ethernet)
        RX packets 1219  bytes 91130 (88.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 406  bytes 28942 (28.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

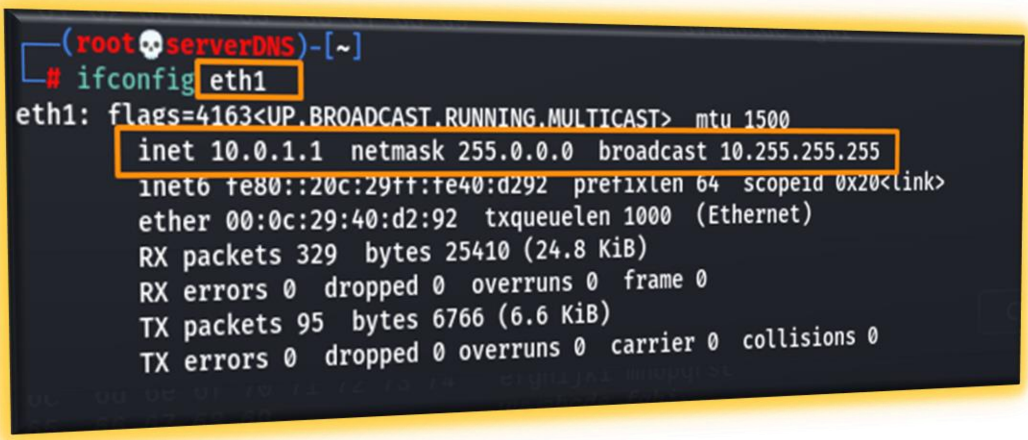
Interfaces 2 [eth1] :

Adresse IP : 10.0.1.1

Masque : 255.0.0.0



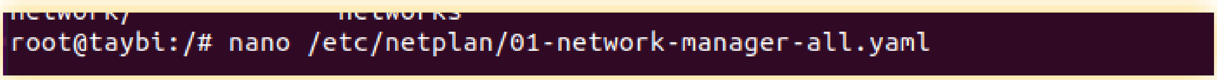
```
auto lo
iface lo inet loopback
auto eth1
iface eth1 inet static
address 10.0.1.1
netmask 255.0.0.0
```



```
┌──(root💀serverDNS)-[~]
└─# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::20c:29ff:fe40:d292  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:40:d2:92  txqueuelen 1000  (Ethernet)
        RX packets 329  bytes 25410 (24.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 95  bytes 6766 (6.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

## Machine 3 [Ubuntu] :

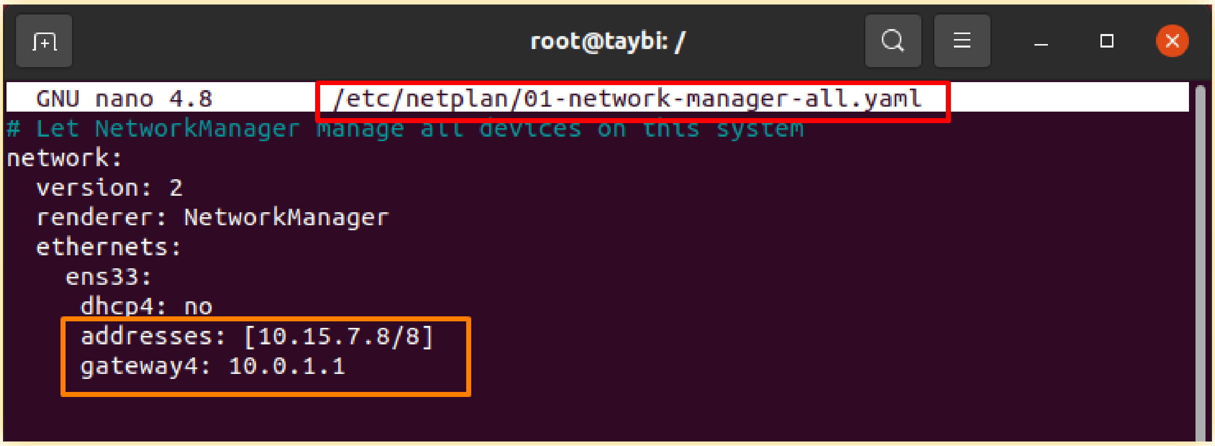
- On configurer le fichier `01-network-manager-all.yml`



```
root@taybi:/# nano /etc/netplan/01-network-manager-all.yaml
```

Adresse IP : 10.15.7.8

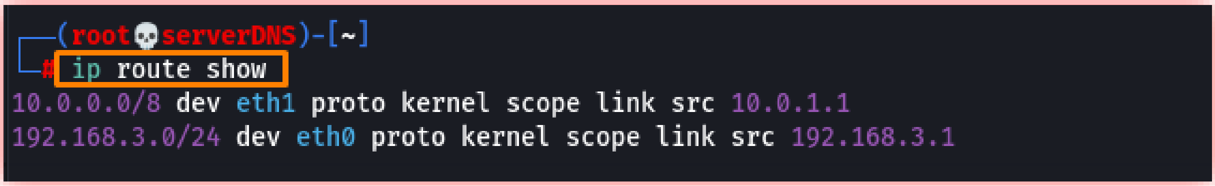
Gateway : 10.0.1.1 [L'interfaces eth1 de la machine 2]



```
  GNU nano 4.8          /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
     dhcp4: no
     addresses: [10.15.7.8/8]
     gateway4: 10.0.1.1
```

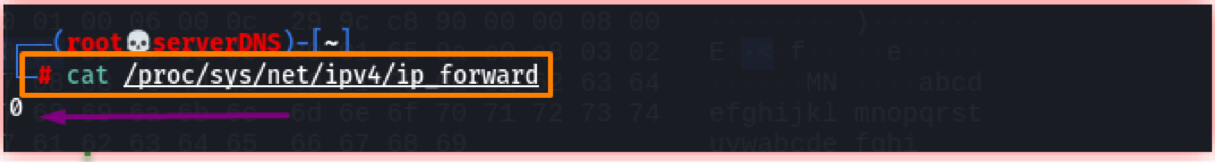
## ★ Connectivité entre Machine1 et Machine 3 :

- Voilà Le table de routage :



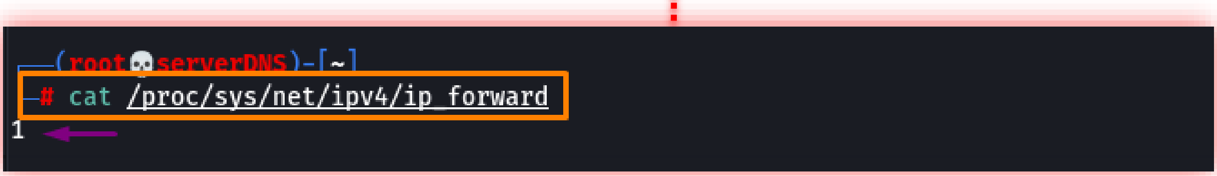
```
┌──(root💀serverDNS)-[~]
└─# ip route show
10.0.0.0/8 dev eth1 proto kernel scope link src 10.0.1.1
192.168.3.0/24 dev eth0 proto kernel scope link src 192.168.3.1
```

- On activer le transfert IP [forwarding] :



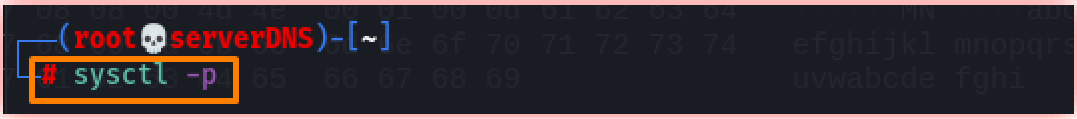
```
┌──(root💀serverDNS)-[~]
└─# cat /proc/sys/net/ipv4/ip_forward
0
```

Pour active le transfert IP nous devons changer la valeur 0 sur le Fichier `/proc/sys/net/ipv4/ip-forward` a 1 :



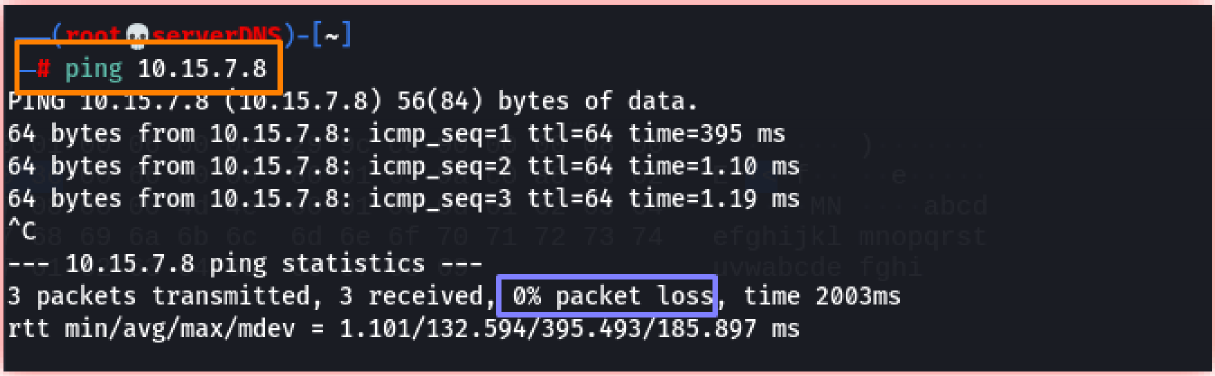
```
┌──(root💀serverDNS)-[~]
└─# cat /proc/sys/net/ipv4/ip_forward
1
```

- Après avoir modifié le fichier, vous pouvez exécuter la commande suivante pour que les modifications prennent effet immédiatement. `Sysctl -p`.

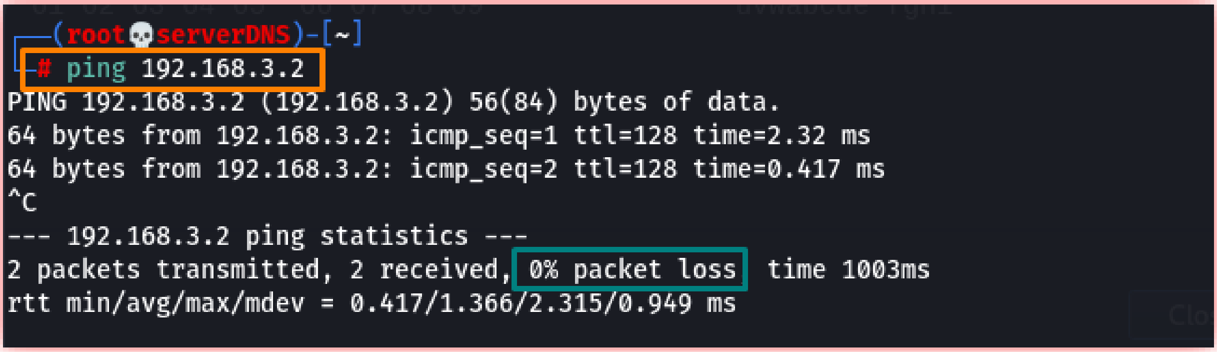


```
┌──(root💀serverDNS)-[~]
└─# sysctl -p
```

- On test la Connectivity entre les machines et les interfaces de Kali[routeur].



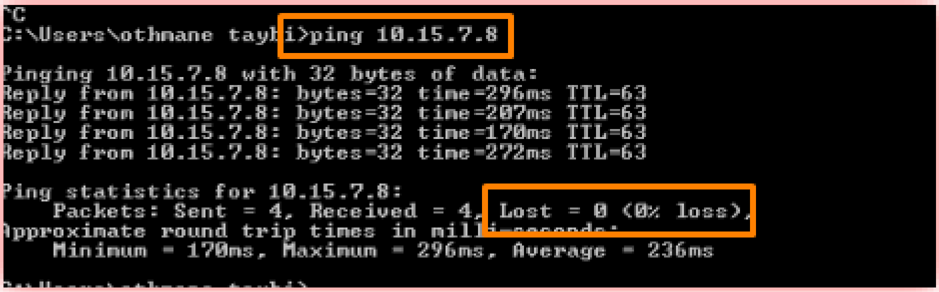
```
┌──(root💀serverDNS)-[~]
└─# ping 10.15.7.8
PING 10.15.7.8 (10.15.7.8) 56(84) bytes of data.
64 bytes from 10.15.7.8: icmp_seq=1 ttl=64 time=395 ms
64 bytes from 10.15.7.8: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 10.15.7.8: icmp_seq=3 ttl=64 time=1.19 ms
^C
--- 10.15.7.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.101/132.594/395.493/185.897 ms
```



```
┌──(root💀serverDNS)-[~]
└─# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=128 time=2.32 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=128 time=0.417 ms
^C
--- 192.168.3.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.417/1.366/2.315/0.949 ms
```

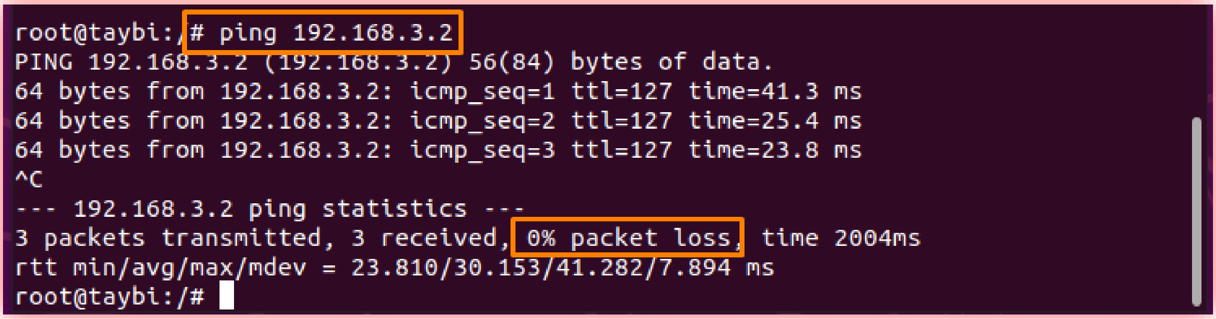
Maintenant en test la Connectivity entre la machine 1 et la machine 3 :

Ping machine 1[Windows 7] -----------→ machine 2[Ubuntu]:



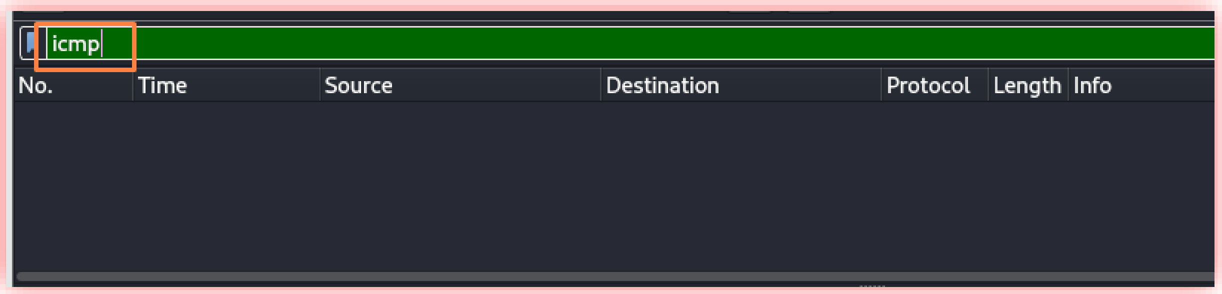
```
^C
C:\Users\othmane taybi>ping 10.15.7.8

Pinging 10.15.7.8 with 32 bytes of data:
Reply from 10.15.7.8: bytes=32 time=296ms TTL=63
Reply from 10.15.7.8: bytes=32 time=207ms TTL=63
Reply from 10.15.7.8: bytes=32 time=170ms TTL=63
Reply from 10.15.7.8: bytes=32 time=272ms TTL=63

Ping statistics for 10.15.7.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 170ms, Maximum = 296ms, Average = 236ms

C:\Users\othmane taybi>
```

Ping machine 2[Ubuntu] -----------→ machine 1[Windows 7] :



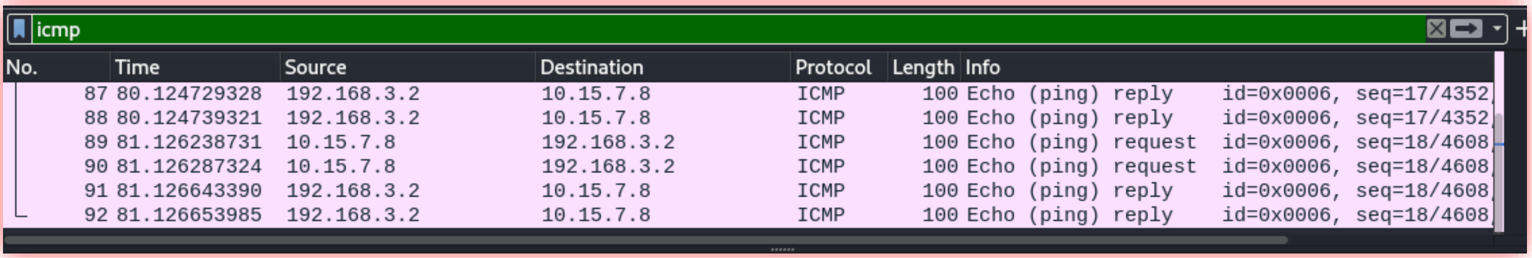
```
root@taybi:/# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=127 time=41.3 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=127 time=25.4 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=127 time=23.8 ms
^C
--- 192.168.3.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 23.810/30.153/41.282/7.894 ms
root@taybi:/# 
```

- Maintenant on sniffer le trafic réseau sur le routeur avec Wireshark :

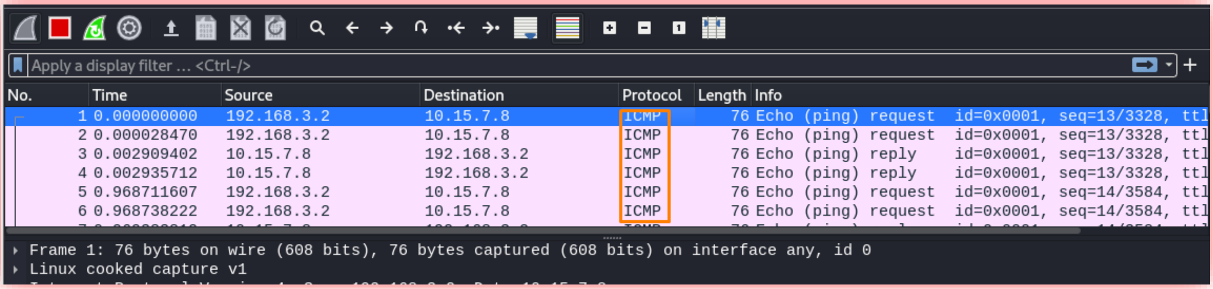


```
No.    Time    Source    Destination    Protocol    Length    Info
```

- On ping entre les machines :

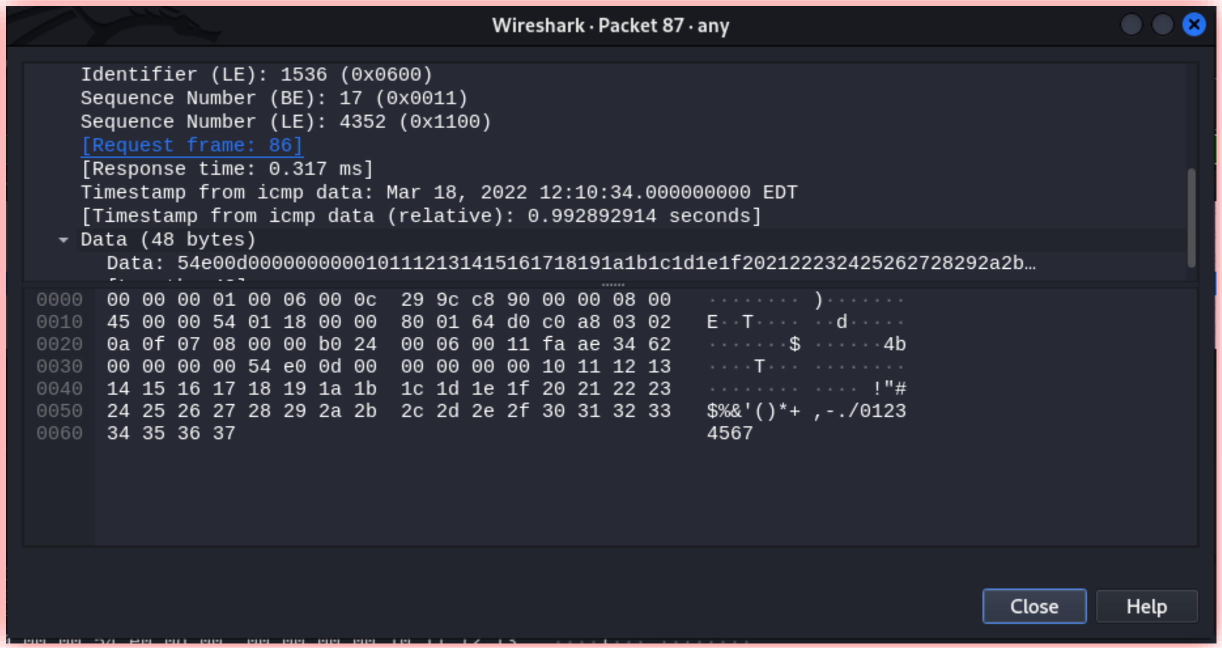


```
No.  Time          Source        Destination   Protocol  Length Info
  87 80.124729328  192.168.3.2   10.15.7.8     ICMP       100 Echo (ping) reply    id=0x0006, seq=17/4352
  88 80.124739321  192.168.3.2   10.15.7.8     ICMP       100 Echo (ping) reply    id=0x0006, seq=17/4352
  89 81.126238731  10.15.7.8     192.168.3.2   ICMP       100 Echo (ping) request  id=0x0006, seq=18/4608
  90 81.126287324  10.15.7.8     192.168.3.2   ICMP       100 Echo (ping) request  id=0x0006, seq=18/4608
  91 81.126643390  192.168.3.2   10.15.7.8     ICMP       100 Echo (ping) reply    id=0x0006, seq=18/4608
  92 81.126653985  192.168.3.2   10.15.7.8     ICMP       100 Echo (ping) reply    id=0x0006, seq=18/4608
```



```
No.  Time         Source        Destination   Protocol  Length Info
   1 0.000000000  192.168.3.2   10.15.7.8     ICMP        76 Echo (ping) request  id=0x0001, seq=13/3328, ttl
   2 0.000028470  192.168.3.2   10.15.7.8     ICMP        76 Echo (ping) request  id=0x0001, seq=13/3328, ttl
   3 0.002909402  10.15.7.8     192.168.3.2   ICMP        76 Echo (ping) reply    id=0x0001, seq=13/3328, ttl
   4 0.002935712  10.15.7.8     192.168.3.2   ICMP        76 Echo (ping) reply    id=0x0001, seq=13/3328, ttl
   5 0.968711607  192.168.3.2   10.15.7.8     ICMP        76 Echo (ping) request  id=0x0001, seq=14/3584, ttl
   6 0.968738222  192.168.3.2   10.15.7.8     ICMP        76 Echo (ping) request  id=0x0001, seq=14/3584, ttl
```

```
▸ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
▸ Linux cooked capture v1
```



```
    Identifier (LE): 1536 (0x0600)
    Sequence Number (BE): 17 (0x0011)
    Sequence Number (LE): 4352 (0x1100)
    [Request frame: 86]
    [Response time: 0.317 ms]
    Timestamp from icmp data: Mar 18, 2022 12:10:34.000000000 EDT
    [Timestamp from icmp data (relative): 0.992892914 seconds]
  ▾ Data (48 bytes)
      Data: 54e00d0000000000101112131415161718191a1b1c1d1e1f202122232425262728292a2b…
```

```
0000  00 00 00 01 00 06 00 0c  29 9c c8 90 00 00 08 00   ········ )·······
0010  45 00 00 54 01 18 00 00  80 01 64 d0 c0 a8 03 02   E··T···· ··d·····
0020  0a 0f 07 08 00 00 b0 24  00 06 00 11 fa ae 34 62   ·······$ ······4b
0030  00 00 00 00 54 e0 0d 00  00 00 00 00 10 11 12 13   ····T··· ········
0040  14 15 16 17 18 19 1a 1b  1c 1d 1e 1f 20 21 22 23   ········ ···· !"#
0050  24 25 26 27 28 29 2a 2b  2c 2d 2e 2f 30 31 32 33   $%&'()*+ ,-./0123
0060  34 35 36 37                                        4567
```