

MEMBRE DU GROUPE

Prénoms : Aniamba Amaye

NOM : SAMBOU

Prénoms : Aboubacar Sadikh

NOM : BODIANG

Prénom : Youssef

NOM : BADDOUJ

RAPPORT Projet Réseaux 2019 – MASTER 1 INFORMATIQUE - LUMINY

1. Configuration Réseau

1.1. Topologie et Adressage

1. Mise en place des 6 Vms (cf fichiers de configurations **Vagrantfile** et **salt** de chaque VM)

Quelques test :

⇒ PING VM1 VERS VM3

```
root@VM1:~# ping 172.16.2.163
```

```
PING 172.16.2.163 (172.16.2.163) 56(84) bytes of data.
```

```
64 bytes from 172.16.2.163: icmp_seq=1 ttl=63 time=2.42 ms
```

```
64 bytes from 172.16.2.163: icmp_seq=2 ttl=63 time=2.12 ms
```

```
64 bytes from 172.16.2.163: icmp_seq=3 ttl=63 time=1.68 ms
```

```
^C
```

```
--- 172.16.2.163 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
```

```
rtt min/avg/max/mdev = 1.685/2.078/2.422/0.302 ms
```

⇒ PING VM1-6 VERS VM3-6

1.2. Un grand malheur !

2. L'interface virtuelle TUN

2.1. Création de l'interface

⇒ Voir Contenu du fichier **tunalloc.c** dans le dossier « partage »

2.2. Configuration de l'interface

1. Script **configure-tun.sh**

```
ip link set dev tun0 up
```

```
ip addr add 172.16.2.1/28 dev tun0
```

2. Suite à la disparition de VM2 il faut modifier les informations de routage sur VM1 et ne pas toucher à celles de VM1-6. Cela est au fait que l'interface **eth1** de VM2 était la passerelle de VM1 pour communiquer avec un hôte du LAN2, en l'occurrence VM3.

3. Faisons un **ping 172.16.2.1**

```
root@VM1:~# gcc /mnt/partage/tunalloc.c -o tun0
```

```
root@VM1:~# cp /mnt/partage/configure-tun.sh .
```

```
root@VM1:~# ./tun0 tun0
```

Création de tun0

Faire la configuration de tun0...

Appuyez sur une touche pour continuer

Interface tun0 Configurée:

Appuyez sur une touche pour terminer

Pendant que le programme est toujours en cours d'exécution, nous pouvons vérifier que le tunnel est bien créé.

```
root@VM1:~# ip addr
```

```
8: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast  
state UNKNOWN group default qlen 500  
    link/none  
    inet 172.16.2.1/28 scope global tun0  
        valid_lft forever preferred_lft forever
```

Faisons à présent un ping vers tun0

```
root@VM1:~# ping 172.16.2.1
```

```
PING 172.16.2.1 (172.16.2.1) 56(84) bytes of data.
```

```
64 bytes from 172.16.2.1: icmp_seq=1 ttl=64 time=0.087 ms
```

```
64 bytes from 172.16.2.1: icmp_seq=2 ttl=64 time=0.083 ms
```

```
64 bytes from 172.16.2.1: icmp_seq=3 ttl=64 time=0.096 ms
```

```
64 bytes from 172.16.2.1: icmp_seq=4 ttl=64 time=0.090 ms
```

```
^C
```

```
--- 172.16.2.1 ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
```

```
rtt min/avg/max/mdev = 0.083/0.089/0.096/0.004 ms
```

On constate que le ping marche bien, cependant aucun paquet n'est capturé par wireshark sur l'interface tun0.

4. En faisant un ping 172.16.2.10 on constate que le ping ne passe pas.

```
root@VM1:~# ping 172.16.2.10
```

```
PING 172.16.2.10 (172.16.2.10) 56(84) bytes of data.
```

```
^C
```

```
--- 172.16.2.10 ping statistics ---
```

```
7 packets transmitted, 0 received, 100% packet loss, time 6048ms
```

Cependant des paquets sont capturés par wireshark sur tun0.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.2.1	172.16.2.10	ICMP	84	Echo (ping) request id=0x077c, seq=1/256, ttl=64 (no response found!)

Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0

Raw packet data

Internet Protocol Version 4, Src: 172.16.2.1 (172.16.2.1), Dst: 172.16.2.10 (172.16.2.10)

Internet Control Message Protocol

5. Les paquets capturés par wireshark nous emmènent à conclure que VM1 essaye de joindre **172.16.2.10** en passant par **tun0 (172.16.2.1)**. tun0 est dès la passerelle pour communiquer avec un autre hôte.

2.3. Récupération des paquets :