Thibault JACQUEMIN - Michel KOT - Nicolas KNIEBIHLER

Rapport IN411 – ESIEE PARIS

TP DMZ / Firewall /Linux

Antoine FRANCOIS - Thibault JULLIAND – Mathieu MARLEIX

Table des matières

I. Câblage de la plate-forme étudiée

Partie Routeur

```
[root@linux-6 user]# mii-tool
        eth0: no link
        eth1: negotiated 100baseTx-FD flow-control, link ok
        eth2: negotiated, link ok
```

Le résultat de mii-tool nous permet de vérifier que la machine est reliée à internet via l'interface eth2 et au poste 5 (DMZ) via l'interface eth1.

```
[root@linux-6 user]# mii-tool
eth0: negotiated 100baseTx-FD flow-control, link ok
eth1: negotiated 100baseTx-FD flow-control, link ok
eth2: negotiated, link ok
```

Après le second branchement, on observe que la jonction avec le poste 4 (LAN) apparaît sur eth0.

Partie DMZ

```
[root@localhost user]# mii-tool
        eth0: no link
        eth1: no link
        eth2: negotiated, link ok (vers Internet)

[root@localhost user]# mii-tool
        eth0: negotiated 100baseTx-FD flow-control, link ok
        eth1: no link
        eth2: negotiated, link ok
```

On observe que l'interface eth0 est liée au routeur, l'interface eth2 à internet et eth1 est inutilisée.

Partie LAN

```
[root@localhost user]# mii-tool
eth0: negotiated, link ok (→ D)
eth1: no link
eth2: no link

[root@localhost user]# mii-tool
eth0: negotiated, link ok
eth1: negotiated 100baseTx-FD flow-control, link ok (→ B)
eth2: no link (→ A)
```

On a donc :

eth0 → D → internet, eth1 → B → routeur, eth2 → A → inutilisé.

II. Routage classique

Configuration des interfaces

Partie Routeur

```
[root@linux-6 user]# ifconfig eth1 194.10.20.1 netmask 255.255.255.0 broadcast 194.10.20.0
[root@linux-6 user]# ifconfig eth0 194.10.10.1 netmask 255.255.255.0 broadcast 194.10.10.0
```

Cela nous attribue une adresse dans les sous-réseaux, nous permettant de communiquer avec les autres machines.

```
echo "1" > /proc/sys/ipv4/net/ip_forward
```

L'activation du routage des paquets. icmp_echo_ignore_all fixe si la machine va ignorer ou non les ping venant d'autres machines.

Serveurs actifs de la machine :

```
[root@linux-6 user]# netstat -l
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:41000                     *:*                         LISTEN
tcp        0      0 localhost.localdomain:11211 *:*                         LISTEN
tcp        0      0 *:sunrpc                    *:*                         LISTEN
tcp        0      0 *:49617                     *:*                         LISTEN
tcp        0      0 localhost.localdomain:7634  *:*                         LISTEN
tcp        0      0 *:5555                      *:*                         LISTEN
tcp        0      0 linux-6.local:domain        *:*                         LISTEN
tcp        0      0 localhost.localdomai:domain *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 localhost.localdomain:953   *:*                         LISTEN
tcp        0      0 *:4025                      *:*                         LISTEN
tcp        0      0 *:nfs                       *:*                         LISTEN
tcp        0      0 *:30020                     *:*                         LISTEN
tcp        0      0 localhost.localdomain:5380  *:*                         LISTEN
tcp        0      0 *:37797                     *:*                         LISTEN
tcp        0      0 *:sunrpc                    *:*                         LISTEN
tcp        0      0 *:www                       *:*                         LISTEN
tcp        0      0 *:33265                     *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 *:https                     *:*                         LISTEN
tcp        0      0 *:58622                     *:*                         LISTEN
udp        0      0 *:44891                     *:*
udp        0      0 *:1900                      *:*
udp        0      0 *:44909                     *:*
udp        0      0 *:45994                     *:*
udp        0      0 localhost.localdomain:11211 *:*
udp        0      0 *:54754                     *:*
udp        0      0 *:nfs                       *:*
```

```
udp        0      0 linux-6.local:domain    *:*
udp        0      0 localhost.locald:domain    *:*
udp        0      0 *:bootpc                *:*
udp        0      0 *:614                   *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:651                   *:*
udp        0      0 *:41647                 *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:43609                 *:*
udp        0      0 *:614                   *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:48354                 *:*

Sockets du domaine UNIX actives(seulement serveurs)
Proto RefCpt Indicatrs   Type       Etat         I-Node Chemin
unix  2      [ ACC ]     STREAM     LISTENING     13862  /tmp/.esd-500/socket
unix  2      [ ACC ]     STREAM     LISTENING     13865
/home/user/.pulse/d9337073b2860fd9e1e0825400000348-runtime/native
unix  2      [ ACC ]     STREAM     LISTENING     13913  /tmp/orbit-user/linc-2a08-0-77767d467642a
unix  2      [ ACC ]     STREAM     LISTENING     6792   /tmp/gpg-9zKoHe/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     13221  /tmp/ssh-JVKUR10594/agent.10594
unix  2      [ ACC ]     STREAM     LISTENING     14146  /tmp/orbit-user/linc-2a1a-0-27ff8acead67b
unix  2      [ ACC ]     STREAM     LISTENING     12962  @/tmp/gdm-session-PhoPdhys
unix  2      [ ACC ]     STREAM     LISTENING     13339  /tmp/.ICE-unix/10594
unix  2      [ ACC ]     STREAM     LISTENING     13338  @/tmp/.ICE-unix/10594
unix  2      [ ACC ]     STREAM     LISTENING     6895   @/tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     13361  /tmp/orbit-user/linc-29e7-0-66f91d17b2b6e
unix  2      [ ACC ]     STREAM     LISTENING     13606  /tmp/keyring-5A4Isx/ssh
unix  2      [ ACC ]     STREAM     LISTENING     6896   /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     16087  /tmp/orbit-user/linc-2bc2-0-7fb5f96c6f927
unix  2      [ ACC ]     STREAM     LISTENING     6239   /var/run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     15130  /tmp/orbit-user/linc-2a3a-0-13c7d4132a7a2
unix  2      [ ACC ]     STREAM     LISTENING     10964  /var/run/avahi-daemon/socket
unix  2      [ ACC ]     STREAM     LISTENING     13273  @/tmp/dbus-MGC2Mc6iRC
unix  2      [ ACC ]     STREAM     LISTENING     13011  /tmp/keyring-5A4Isx/control
unix  2      [ ACC ]     STREAM     LISTENING     14048  /tmp/orbit-user/linc-2a0c-0-ad0fff89b265
unix  2      [ ACC ]     STREAM     LISTENING     12188  /var/run/saslauthd/mux
unix  2      [ ACC ]     STREAM     LISTENING     13677  /tmp/orbit-user/linc-29ef-0-519e9dcb12758
unix  2      [ ACC ]     STREAM     LISTENING     8862   /dev/gpmctl
unix  2      [ ACC ]     STREAM     LISTENING     15053  /tmp/orbit-user/linc-2a1d-0-2925d99c9b455
unix  2      [ ACC ]     STREAM     LISTENING     7702   @/tmp/gdm-greeter-YVMWjsva
unix  2      [ ACC ]     STREAM     LISTENING     36751  /tmp/orbit-user/linc-ecf-0-20b26914c2522
unix  2      [ ACC ]     STREAM     LISTENING     37283
/tmp/OSL_PIPE_500_SingleOfficeIPC_2181fc8df0bd2178b8578cc2ee8e30ad
unix  2      [ ACC ]     STREAM     LISTENING     13740  /tmp/keyring-5A4Isx/pkcs11
unix  2      [ ACC ]     STREAM     LISTENING     13520  /tmp/orbit-user/linc-2962-0-6d4f0fbdbdde2
unix  2      [ ACC ]     STREAM     LISTENING     10948  /var/run/rpcbind.sock
unix  2      [ ACC ]     STREAM     LISTENING     14844  /tmp/orbit-user/linc-2a23-0-61966a6569d97
unix  2      [ ACC ]     STREAM     LISTENING     15004  /tmp/orbit-user/linc-2a33-0-4bf45a3795234
unix  2      [ ACC ]     STREAM     LISTENING     14622  /tmp/orbit-user/linc-2a1f-0-3b3ffcc73cf7c
unix  2      [ ACC ]     STREAM     LISTENING     6321   @/var/run/hald/dbus-5ZVqkQFHPl
unix  2      [ ACC ]     STREAM     LISTENING     6278   @/var/run/hald/dbus-Ouz0jafHYO
unix  2      [ ACC ]     STREAM     LISTENING     15077  /tmp/orbit-user/linc-2a25-0-1e3fa02ca41cc
unix  2      [ ACC ]     STREAM     LISTENING     37279  /tmp/orbit-user/linc-1034-0-4e17db47206c
unix  2      [ ACC ]     STREAM     LISTENING     13209  /tmp/gpg-ybORHU/S.gpg-agent
```

```
root@linux-6 user]# route -n
```

```
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
194.10.10.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.182.0   0.0.0.0         255.255.255.0   U     10     0        0 eth2
127.0.0.0       0.0.0.0         255.255.255.0   U     0      0        0 lo
194.10.20.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     10     0        0 eth2
0.0.0.0         192.168.182.1   0.0.0.0         UG    10     0        0 eth2
```

Partie DMZ

Contenu du dossier ipv4 :

```
[root@localhost user]# ls -a /proc/sys/net/ipv4
./                              tcp_dma_copybreak
../                             tcp_dsack
cipso_cache_bucket_size         tcp_ecn
cipso_cache_enable              tcp_fack
cipso_rbm_optfmt                tcp_fin_timeout
cipso_rbm_strictvalid           tcp_frto
conf/                           tcp_frto_response
icmp_echo_ignore_all            tcp_keepalive_intvl
icmp_echo_ignore_broadcasts     tcp_keepalive_probes
icmp_errors_use_inbound_ifaddr  tcp_keepalive_time
icmp_ignore_bogus_error_responses  tcp_low_latency
icmp_ratelimit                  tcp_max_orphans
icmp_ratemask                   tcp_max_ssthresh
igmp_max_memberships            tcp_max_syn_backlog
igmp_max_msf                    tcp_max_tw_buckets
inet_peer_gc_maxtime            tcp_mem
inet_peer_gc_mintime            tcp_moderate_rcvbuf
inet_peer_maxttl                tcp_mtu_probing
inet_peer_minttl                tcp_no_metrics_save
inet_peer_threshold             tcp_orphan_retries
ip_default_ttl                  tcp_reordering
ip_dynaddr                      tcp_retrans_collapse
ip_forward                      tcp_retries1
ipfrag_high_thresh              tcp_retries2
ipfrag_low_thresh               tcp_rfc1337
ipfrag_max_dist                 tcp_rmem
ipfrag_secret_interval          tcp_sack
ipfrag_time                     tcp_slow_start_after_idle
ip_local_port_range             tcp_stdurg
ip_nonlocal_bind                tcp_synack_retries
ip_no_pmtu_disc                 tcp_syncookies
neigh/                          tcp_syn_retries
route/                          tcp_timestamps
rt_cache_rebuild_count          tcp_tso_win_divisor
tcp_abc                         tcp_tw_recycle
tcp_abort_on_overflow           tcp_tw_reuse
tcp_adv_win_scale               tcp_window_scaling
tcp_allowed_congestion_control  tcp_wmem
tcp_app_win                     tcp_workaround_signed_windows
tcp_available_congestion_control  udp_mem
tcp_base_mss                    udp_rmem_min
tcp_congestion_control          udp_wmem_min
tcp_cookie_size                 xfrm4_gc_thresh
```

Ces entrées servent à activer ou désactiver des protocoles ipv4.

```
[root@linux-5 user]# ifconfig eth0 194.10.20.2 netmask 255.255.255.0
```

La carte eth0 est maintenant configurée pour faire partie du même sous-réseau que celui de l'interface eth1 du routeur. On ajoute maintenant le routeur comme passerelle par défaut.

```
[root@linux-5 user]# route add default gw 194.10.20.1 dev eth0
```

Vérification de la configuration de l'interface eth0

```
[root@localhost user]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:B7:7E:B0
          inet adr:194.10.20.2 Bcast:194.10.10.0  Masque:255.255.255.0 .0
```

Note sur la commande route :

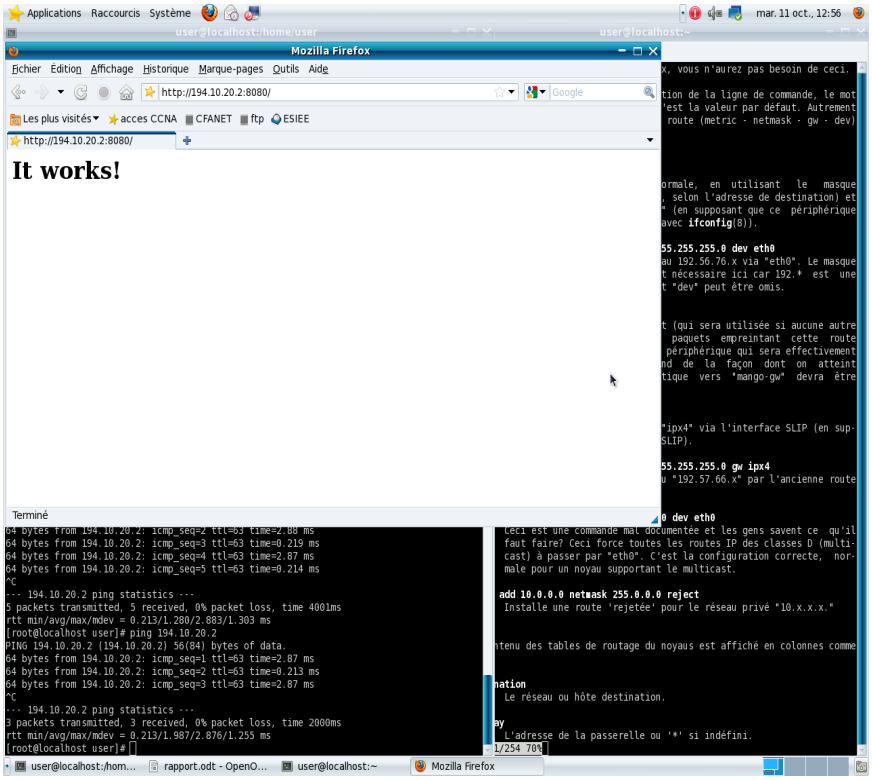
-n Affiche les adresses numériques, au lieu d'essayer de déterminer les noms d'hôtes. C'est utile si vous essayez de savoir pourquoi la route vers votre serveur de nom a disparu.

```
service httpd restart
```

Redémarrage du service du serveur web après modification du port d'écoute pour écouter le port 8080.

```
[root@localhost user]# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
0.0.0.0         194.10.20.1     0.0.0.0             UG    0      0        0 eth0
192.168.182.0   0.0.0.0         255.255.255.0   U     10     0        0 eth2
194.10.20.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     10     0        0 eth2
```

Affichage de la page par défaut du serveur web :



Partie LAN

```
[root@localhost user]# ipconfig eth0 194.10.10.2 gw 194.10.10.1 netmask 255.255.255.0
[root@localhost user]# route add default gw 194.10.10.1 dev eth0
```

La carte eth0 est maintenant configurée pour faire partie du même sous réseau que celui du routeur. Et la gateway à été configurée comme le routeur.

```
[root@localhost user]# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
194.10.10.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.182.0   0.0.0.0         255.255.255.0   U     5      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     5      0        0 eth0
0.0.0.0         194.10.10.1     0.0.0.0         UG    0      0        0 eth1
```

On observe bien que toutes les machines peuvent se pinger entre elles.

III. Mise en œuvre de la plate-forme sécurisée

Script Iptable :

```
# Question 3.a
iptables -F -t nat
iptables -F
# Question 3.b
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Question 5
iptables -A INPUT -s 194.10.10.0/24 -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -d 194.10.10.0/24 -p icmp --icmp-type echo-reply -j ACCEPT
# Question 6
iptables -A INPUT -s 194.10.20.0/24 -d 194.10.20.1 -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -s 194.10.20.1 -d 194.10.20.0/24 -p icmp --icmp-type echo-reply -j ACCEPT
# Question 7
iptables -A FORWARD -s 194.10.10.0/24 -d 194.10.20.0/24 -p icmp --icmp-type echo-request -j ACCEPT
iptables -A FORWARD -s 194.10.20.0/24 -d 194.10.10.0/24 -p icmp --icmp-type echo-reply -j ACCEPT
# Question 8
iptables -t nat -A POSTROUTING -s 194.10.10.0/24 -d 194.10.20.0/24 -j SNAT --to-source 194.10.20.1
iptables -t nat -A PREROUTING -i eth0 -d 194.10.10.0/24 -j  DNAT –to-destination 194.10.20.2
# Question 9
iptables -A INPUT -s 194.10.10.0/24 -p tcp -m multiport --dports 8080 -j ACCEPT
iptables -A OUTPUT -d 194.10.10.0/24 -p tcp -m multiport --dports 8080 -j ACCEPT
#
iptables -A INPUT -s 194.10.20.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
iptables -A OUTPUT -d 194.10.20.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
#
iptables -A FORWARD -s 194.10.20.2 -d 194.10.10.2 -p tcp -m multiport --dports 443,80 -j ACCEPT
iptables -A FORWARD -s 194.10.10.2 -d 194.10.20.2 -p tcp -m multiport --dports 8080 -j ACCEPT
```

Retours :

```
iptables -F

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Après question 4

```
[root@linux-6 ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
```

Après question 5

```
[root@linux-6 ~]# iptables -n -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.0/24       194.10.10.1         icmp type 8

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.1         194.10.10.0/24      icmp type 0
```

Après question 6

```
[root@linux-6 ~]# iptables -n -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.0/24       194.10.10.1         icmp type 8
ACCEPT     icmp --  194.10.20.0/24       194.10.20.1         icmp type 8

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.1         194.10.10.0/24      icmp type 0
ACCEPT     icmp --  194.10.20.1         194.10.20.0/24      icmp type 0
```

Après question 7

```
[root@linux-6 ~]# iptables -n -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.0/24       194.10.10.1         icmp type 8
ACCEPT     icmp --  194.10.20.0/24       194.10.20.1         icmp type 8

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.0/24       194.10.20.0/24      icmp type 8
ACCEPT     icmp --  194.10.20.0/24       194.10.10.0/24      icmp type 0

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     icmp --  194.10.10.1         194.10.10.0/24      icmp type 0
ACCEPT     icmp --  194.10.20.1         194.10.20.0/24      icmp type
```

Partie DMZ

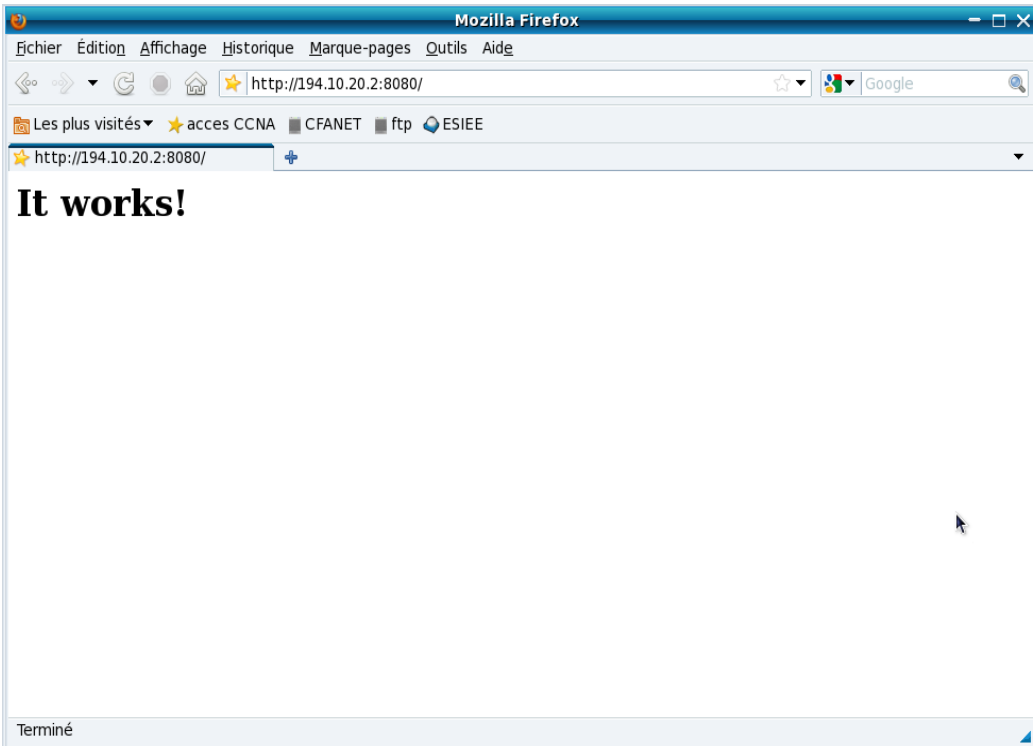
Question 8 :

```
[root@linux-8 user]# tcpdump -n -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
13:34:32.950244 IP 194.10.20.1 > 194.10.20.2: ICMP echo request, id 12732, seq 1, length 64
13:34:32.950274 IP 194.10.20.2 > 194.10.20.1: ICMP echo reply, id 12732, seq 1, length 64
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

On peut observer que la DMZ répond bien aux différents ping de la LAN, cependant, grâce au filtrage effectué et à la translation d'adresse, la DMZ voit la demande de ping venir du routeur et non de la LAN, elle répond alors au routeur qui se charge de répondre à la LAN.

Question 9 :

On observe avec tcpdump que les trames sont envoyées émises par le routeur et retransmises au à celui-ci suite à la translation d'adressage sur les ports http, 8080 et https.



Filtrage LAN et INTERNET

Question 10 :

```
iptables -A INPUT -s 194.10.10.0/24 -p icmp icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -d 194.10.10.0/24 -p icmp icmp-type echo-reply -j ACCEPT
#
iptables -A INPUT -s 194.10.10.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
iptables -A OUTPUT -d 194.10.10.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
iptables -A INPUT -s 192.168.0.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
iptables -A OUTPUT -d 192.168.0.0/24 -p tcp -m multiport --dports 443,80 -j ACCEPT
#
iptables -t nat -A POSTROUTING -s 194.10.10.0/24 -d 192.168.0.0/24 -j SNAT --to-
source 192.168.0.1
```

La table NAT est indispensable puisqu'on souhaite masquer les machines du réseau LAN, il faut donc faire une translation d'adresse, ce qui ne peut se faire que grâce au NAT et à un POSTROUTING.

Question 13 :

Exemple de trames obtenues avec Wireshark :