Scan réseau Couche 3 (ICMP)

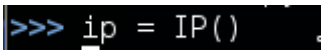
Crée son paquet ICMP avec Scapy

Lancez Wireshark sur la carte eth0 et filtrez sur ICMP

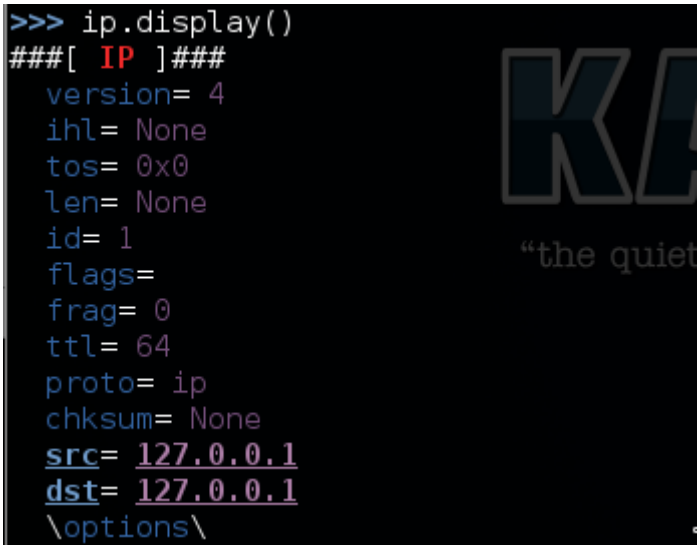
Le protocole ICMP fait appel à la couche IP il faut donc commencer par structurer un paquet IP avant de passer à l'ICMP.

Ouvrez scapy

Mettez l'objet IP dans une variable.



Affichez la structure de l'objet (en-tête)



Changez l'IP du destinataire.

```
>>> ip.dst = "192.168.2.1"
```

Vérifiez le champ dst

```
>>> ip.display()
  src= 192.168.2.5
  dst= 192.168.2.1
```

Cela change automatiquement l'adresse source.

Importer l'objet ICMP dans une seconde variable.

```
>>> ping = ICMP()
```

Affichez la variable

```
>>> ping.display()
###[ ICMP ]###
  type= echo-request
  code= 0
  chksum= None
  id= 0x0
  seq= 0x0
```

La structure est par défaut celle du ping.

Il faut donc maintenant compiler les deux structures dans une variable.

```
>>> ping_request = (ip/ping)
```

Envoyez le paquet

```
>>> sr1(ping_request)
```

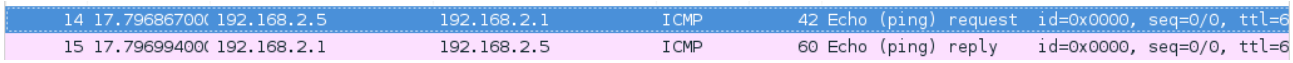
```
Begin emission:
.Finished to send 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
<IP  version=4L ihl=5L tos=0x0 len=28 id=8192 flags= frag=0L ttl=64 proto=icmp chksum=0xd58a src=192.168.2.1 dst
=192.168.2.5 options=[] |<ICMP  type=echo-reply code=0 chksum=0xffff id=0x0 seq=0x0 |<Padding  load='\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00' |>>>
```

La machine a bien répondu.
Inclure la réponse dans une variable

```
>>> rep_ping = sr1(ping_request)
```

Affichez la structure de la réponse.

Analysez votre paquet ICMP dans wireshark



Ping

Il est possible de déterminer si l'hôte est actif ou non grâce à la fonction ping (qui peut être bloquée via pare-feu)

```
root@kali:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_req=1 ttl=64 time=6.11 ms
64 bytes from 192.168.2.1: icmp_req=2 ttl=64 time=0.229 ms
^C
--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.229/3.172/6.115/2.943 ms
```

La machine 192.168.2.1 répond, elle est donc active. Comme ARPING, il faut couper le ping pour qu'il arrête l'envoi des requêtes ICMP (ctrl+c)

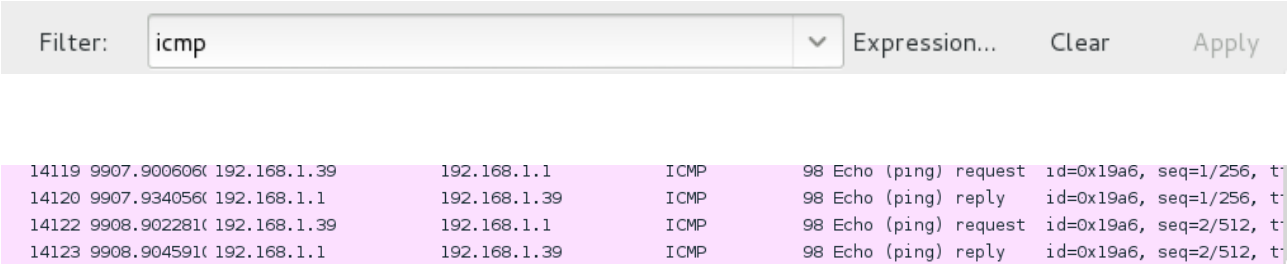
Il est donc possible de préciser le nombre de requêtes à envoyer.

```
root@kali:~# ping 192.168.2.1 -c 1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_req=1 ttl=64 time=0.536 ms
```

Le ping retourne plusieurs informations comme le TTL, chaque système a un TTL par défaut.
Windows = 128
Cisco = 255
Unix = 64

Grâce au ping il est possible de déterminer (si aucune modification du système) l'OS de la machine.

Analysez dans Wireshark le flux ICMP



Optimisation du ping :

Lancez la commande.

```
root@kali:~# for addr in $(seq 1 254); do ping 192.168.2.$addr -c 1 |grep "bytes from" | cut -d " " -f 4 | cut -
d ":" -f 1 >> ipliste_ping.txt & done
```

Essayez d'expliquer la commande :

Listez les réponses :

```
cat ipliste_ping.txt
```

```
192.168.2.5
192.168.2.1
192.168.2.10
```

Hping3

Hping a la particularité de pouvoir scanner sur la couche 3 et 4.

Lancez la commande hping3

```
root@kali:~# hping3 192.168.2.1 --icmp
HPING 192.168.2.1 (eth0 192.168.2.1): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.2.1 ttl=64 id=43631 icmp_seq=0 rtt=8.7 ms
len=46 ip=192.168.2.1 ttl=64 id=43632 icmp_seq=1 rtt=0.3 ms
len=46 ip=192.168.2.1 ttl=64 id=43633 icmp_seq=2 rtt=0.8 ms
len=46 ip=192.168.2.1 ttl=64 id=43634 icmp_seq=3 rtt=0.2 ms
```

Précisez le nombre de paquets à envoyer.

```
root@kali:~# hping3 192.168.2.1 --icmp -c 1
HPING 192.168.2.1 (eth0 192.168.2.1): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.2.1 ttl=64 id=43635 icmp_seq=0 rtt=0.3 ms
```

Vérifiez une requête non aboutie.

```
root@kali:~# hping3 192.168.2.100 --icmp -c 1
HPING 192.168.2.100 (eth0 192.168.2.100): icmp mode set, 28 headers + 0 data bytes

--- 192.168.2.100 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

La chaîne de caractères présente seulement dans les réponses est len=

Optimisation :

Lancez la commande

```
root@kali:~# for addr in $(seq 1 254); do hping3 192.168.2.$addr --icmp -c 1 | grep len | cut -d " " -f 2 | cut
-d "=" -f 2 >> ipliste_hping.txt & done
```

Listez les réponses

```
cat ipliste_hping.txt
```

```
192.168.2.1
192.168.2.10
```

Expliquez la commande

Fping

La fonction fping ressemble au ping sauf qu'il permet nativement de scanner plusieurs adresses ou une plage complète.

```
root@kali:~# fping 192.168.2.10
192.168.2.10 is alive
```

fping envoie seulement une requête ICMP et attend de recevoir la réponse si il y en a une.

Lancez un fping sur une adresse non utilisée.

```
root@kali:~# fping 192.168.1.150
ICMP Host Unreachable from 192.168.1.39 for ICMP Echo sent to 192.168.1.150
ICMP Host Unreachable from 192.168.1.39 for ICMP Echo sent to 192.168.1.150
ICMP Host Unreachable from 192.168.1.39 for ICMP Echo sent to 192.168.1.150
ICMP Host Unreachable from 192.168.1.39 for ICMP Echo sent to 192.168.1.150
192.168.1.150 is unreachable
```

En cas de non réponse à la première requête il essaye d'en envoyer 3 autres avant de valider que la machine n'est pas active ou bloque le ping.

Il est donc possible de lui préciser de lancer seulement une requête et d’arrêter, même en cas de non réponse, ce qui peut éviter une détection potentielle.

```
root@kali:~# fping 192.168.1.150 -c 1

192.168.1.150 : xmt/rcv/%loss = 1/0/100%
```

Il a donc effectué une seule tentative

Lancez un scan sur plusieurs adresses IP

```
root@kali:~# fping 192.168.2.10 192.168.2.1 -c 1
192.168.2.10 : [0], 96 bytes, 0.32 ms (0.32 avg, 0% loss)

192.168.2.10 : xmt/rcv/%loss = 1/1/0%, min/avg/max = 0.32/0.32/0.32
192.168.2.1  : xmt/rcv/%loss = 1/0/100%
```

Lancez un scan sur une plage réseau

```
root@kali:~# fping -g 192.168.1.0/24 -c 1
```

```
root@kali:~# fping -g 192.168.2.0/24 -c 1
192.168.2.5   : [0], 96 bytes, 0.07 ms (0.07 avg, 0% loss)
192.168.2.10  : [0], 96 bytes, 0.28 ms (0.28 avg, 0% loss)
ICMP Host Unreachable from 192.168.2.5 for ICMP Echo sent to 192.168.2.1
ICMP Host Unreachable from 192.168.2.5 for ICMP Echo sent to 192.168.2.2
```

Optimisez fping.