



Lab 2 : Analyse de trafic réseau

OBJECTIFS :

- Savoir utiliser deux outils de capture de trafic réseau et d'analyse de paquets :
 - **tcpdump** qui fonctionne dans les environnements en mode texte
 - Wireshark qui fonctionne dans les environnements en mode graphique

ACTION 1 : Génération de trafic avec la commande ping

La commande **ping** permet de tester la connectivité avec une machine distante en générant une série de paquets **ICMP echo** auxquels répond la machine distante par des paquets **ICMP Reply** (Fig. 1)

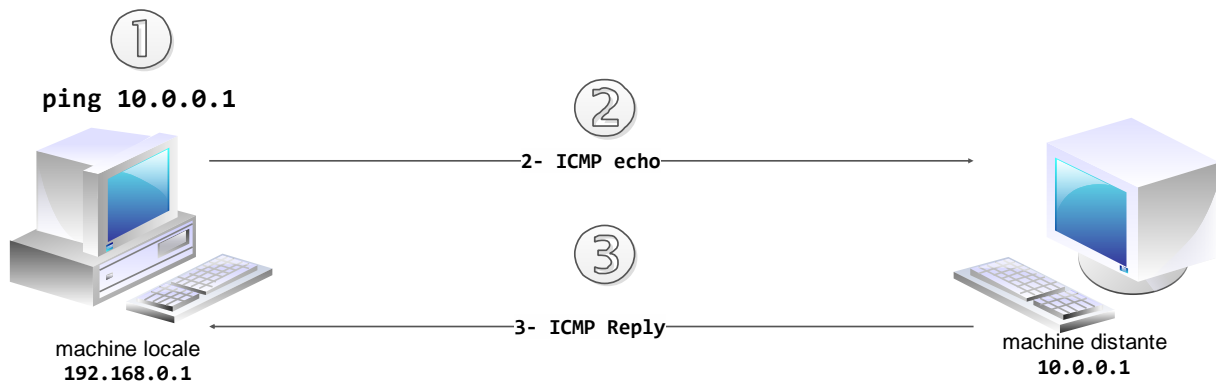


Figure 1 Phase d'exécution du ping

Cette commande est très utile pour générer du trafic lors de l'apprentissage des outils d'analyse de trafic. La syntaxe simplifiée de la commande **ping** est la suivante :

ping adresse_ip_machine_distante

Exemple :

ping 10.0.0.1

1. Réalisez la topologie de la figure 2 en vous inspirant du « *Création de réseau LAN avec VirtualBox* »

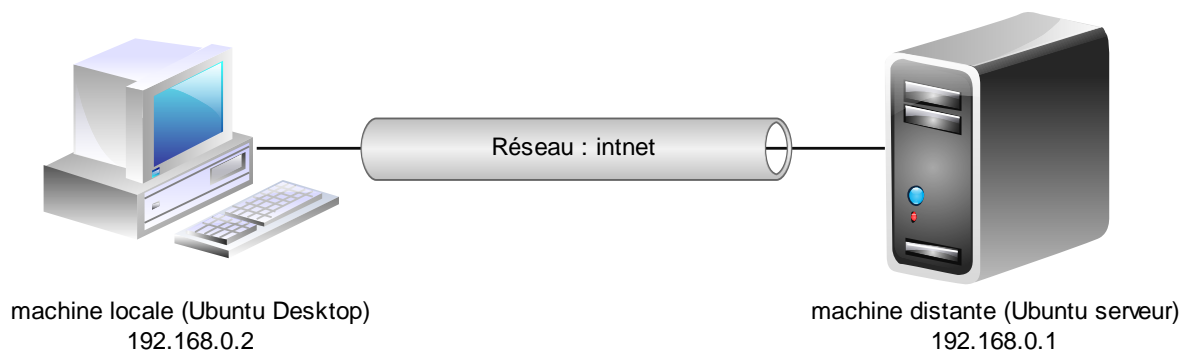


Figure 2 Topologie ping

- Exécutez la commande **ping** de la machine Ubuntu serveur vers la machine Ubuntu Desktop. Vous devriez obtenir comme résultat le délai d'aller-retour comme illustré sur la figure 3

```

Userver [En fonction] - Oracle VM VirtualBox
Machine Écran Périphériques Aide
uroot@userver:~$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data:
64 bytes from 192.168.0.2: icmp_req=1 ttl=64 time=2.14 ms
64 bytes from 192.168.0.2: icmp_req=2 ttl=64 time=0.947 ms
64 bytes from 192.168.0.2: icmp_req=3 ttl=64 time=0.740 ms
64 bytes from 192.168.0.2: icmp_req=4 ttl=64 time=0.844 ms
^C
--- 192.168.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.740/1.169/2.146/0.569 ms
uroot@userver:~$

```

Figure 3 Message quand la machine distante est accessible

```

Userver [En fonction] - Oracle VM VirtualBox
Machine Écran Périphériques Aide
uroot@userver:~$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data:
From 192.168.0.1 icmp_seq=9 Destination Host Unreachable
From 192.168.0.1 icmp_seq=10 Destination Host Unreachable
From 192.168.0.1 icmp_seq=11 Destination Host Unreachable
From 192.168.0.1 icmp_seq=12 Destination Host Unreachable
From 192.168.0.1 icmp_seq=13 Destination Host Unreachable
From 192.168.0.1 icmp_seq=14 Destination Host Unreachable
^C
--- 192.168.0.2 ping statistics ---
17 packets transmitted, 0 received, +6 errors, 100% packet loss, time 16097ms
pipe 3
uroot@userver:~$

```

Figure 4 Message lorsque la machine distante est inaccessible pour des raisons multiples

- Pour arrêter **ping** sur Linux il suffit d'appuyer sur les touches **[Ctrl]+C**
- Pour limiter le nombre de paquets générés par **ping** de Linux vous pourrez utiliser l'option **-c**

Exemple : **ping -c 4 10.0.0.1** #envoyer 4 paquet ICMP echo à 10.0.0.1

```

uroot@userver:~$ ping -c 1 172.21.0.1
PING 172.21.0.1 (172.21.0.1) 56(84) bytes of data.
64 bytes from 172.21.0.1: icmp_req=1 ttl=64 time=0.072 ms

--- 172.21.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.072/0.072/0.072/0.000 ms
uroot@userver:~$ ping -c 2 172.21.0.1
PING 172.21.0.1 (172.21.0.1) 56(84) bytes of data.
64 bytes from 172.21.0.1: icmp_req=1 ttl=64 time=0.070 ms
64 bytes from 172.21.0.1: icmp_req=2 ttl=64 time=0.059 ms

--- 172.21.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.059/0.064/0.070/0.009 ms
uroot@userver:~$ ping -c 3 172.21.0.1
PING 172.21.0.1 (172.21.0.1) 56(84) bytes of data.
64 bytes from 172.21.0.1: icmp_req=1 ttl=64 time=0.084 ms
64 bytes from 172.21.0.1: icmp_req=2 ttl=64 time=0.164 ms
64 bytes from 172.21.0.1: icmp_req=3 ttl=64 time=0.055 ms

--- 172.21.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.055/0.101/0.164/0.046 ms
uroot@userver:~$

```

Figure 5 Ping avec un nombre de paquets limités

ACTION 2 : Capture de trafic avec tcpdump

TCPDUMP est un outil de capture et d'analyse de trafics réseaux en ligne de commande. Son utilisation peut être indispensable afin de bien comprendre le fonctionnement des protocoles de base de TCP/IP (surtout dans les environnements sans X Window).

ACTION 2.1 Utilisation de base

TCPDUMP¹ capture des paquets soit sur une interface soit sur toutes les interfaces mais nécessite des droits root (administrateur). Dans cette partie nous considérons la topologie sur la figure 2. Les captures de paquets se font à partir du client **Ubuntu Desktop** et la génération de trafic se fait à partir du serveur **Ubuntu Server**.

1. Listez les interfaces sur lesquelles vous pouvez capturer des paquets sur le client en tapant la commande² :

sudo tcpdump -D

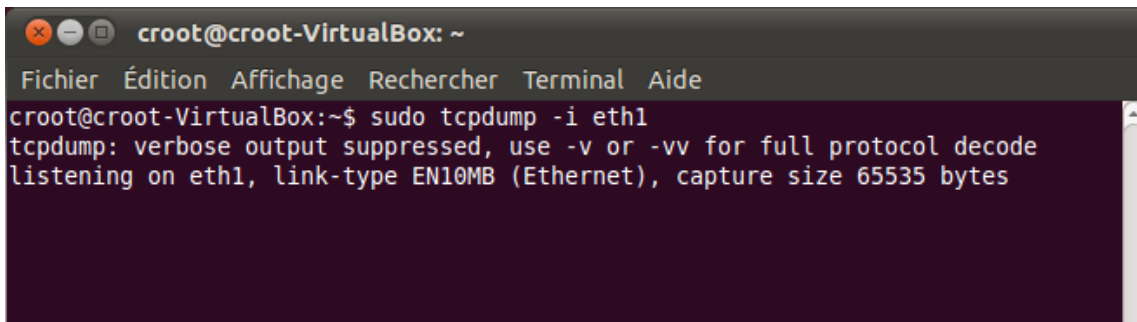
¹ Tcpdump est par défaut installé sur quasiment tous les systèmes Linux

² Les utilisateurs simples ne sont pas autorisés à faire une capture de paquets

```
croot@croot-VirtualBox:~$ sudo tcpdump -D
1.eth0
2.eth1
3.usbmon1 (USB bus number 1)
4.any (Pseudo-device that captures on all interfaces)
5.lo
croot@croot-VirtualBox:~$
```

2. Démarrez la capture sur l'interface **eth1** en tapant la commande suivante :

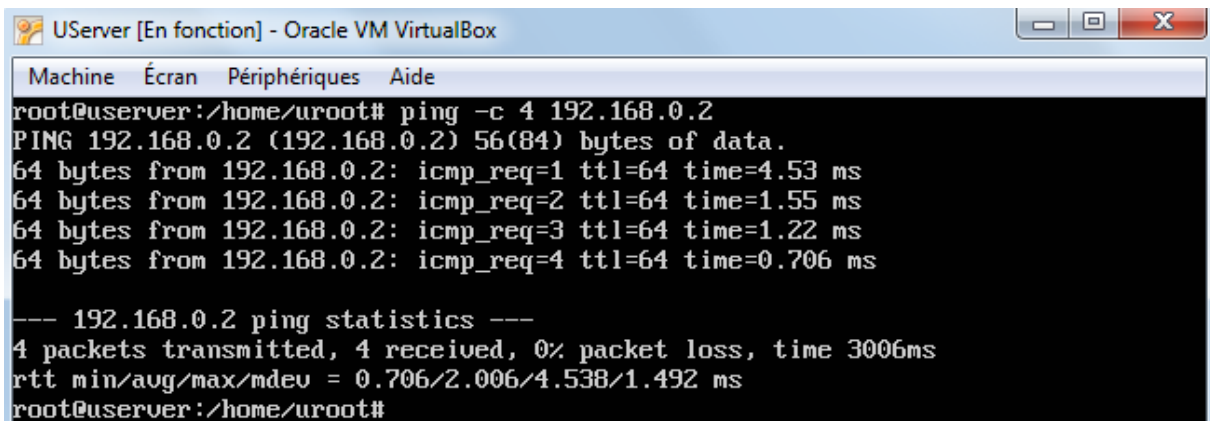
sudo tcpdump -i eth1



```
croot@croot-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide
croot@croot-VirtualBox:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
```

Il faut ensuite générer du trafic en faisant un **ping** du serveur vers le client

Server \$ sudo ping -c 4 192.168.0.2



```
UServer [En fonction] - Oracle VM VirtualBox
Machine Écran Périphériques Aide
root@userver:/home/uroot# ping -c 4 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_req=1 ttl=64 time=4.53 ms
64 bytes from 192.168.0.2: icmp_req=2 ttl=64 time=1.55 ms
64 bytes from 192.168.0.2: icmp_req=3 ttl=64 time=1.22 ms
64 bytes from 192.168.0.2: icmp_req=4 ttl=64 time=0.706 ms

--- 192.168.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.706/2.006/4.538/1.492 ms
root@userver:/home/uroot#
```

3. La sortie de TCPDUMP devrait commencer à montrer l'activité qui vient de se passer sur le réseau

```
croot@croot-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide
croot@croot-VirtualBox:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
04:29:12.629120 ARP, Request who-has 192.168.0.2 tell useur, length 46
04:29:12.629177 ARP, Reply 192.168.0.2 is-at 08:00:27:07:c8:29 (oui Unknown), length 28
04:29:12.630497 IP useur > 192.168.0.2: ICMP echo request, id 1366, seq 1, length 64
04:29:12.630535 IP 192.168.0.2 > useur: ICMP echo reply, id 1366, seq 1, length 64
04:29:13.629467 IP useur > 192.168.0.2: ICMP echo request, id 1366, seq 2, length 64
04:29:13.629500 IP 192.168.0.2 > useur: ICMP echo reply, id 1366, seq 2, length 64
04:29:14.631496 IP useur > 192.168.0.2: ICMP echo request, id 1366, seq 3, length 64
04:29:14.631530 IP 192.168.0.2 > useur: ICMP echo reply, id 1366, seq 3, length 64
04:29:15.633637 IP useur > 192.168.0.2: ICMP echo request, id 1366, seq 4, length 64
04:29:15.633671 IP 192.168.0.2 > useur: ICMP echo reply, id 1366, seq 4, length 64
04:29:17.640899 ARP, Request who-has useur tell 192.168.0.2, length 28
04:29:17.641691 ARP, Reply useur is-at 08:00:27:ce:3c:21 (oui Unknown), length 46
```

Figure 6 : Première sortie de la capture de TCPDUMP

4. Cherchez la signification de cette sortie dans les pages manuelles de **tcpdump** et sur Internet
5. Refaites les questions 2 à 5 en remplaçant à chaque fois la commande de la question 2 par les commandes suivantes et essayez de comprendre la sortie en cherchant sur Internet
 - a. `sudo tcpdump -ni eth1`
 - b. `sudo tcpdump -nni eth1`
 - c. `sudo tcpdump -Xi eth1`
 - d. `sudo tcpdump -XXi eth1`
 - e. `sudo tcpdump -vvvi eth1`
 - f. `sudo tcpdump -nnXXi eth1`
 - g. `sudo tcpdump -nnvvSi eth1`
 - h. `sudo tcpdump -nnvvXXSi eth1`
 - i. `sudo tcpdump -nvi eth1 -c 2`
 - j. `sudo tcpdump -enX eth1`
6. Il est possible de sauvegarder les paquets qui sont capturés au format pcap et les relire grâce aux options `-w` et `-r`
 - a. `tcpdump -s 0 -i eth1 -w filecapture.cap`
 - b. `tcpdump -r filecapture.cap`
7. Application :
 - a. Montrer qu'un ping génère 2 paquets ICMP.
 - b. Qu'est-ce qui différencie deux paires de paquets appartenant à deux ping différents ? (utilisez **tcpdump**)

ACTION 2.2 : Capture avec des filtres tcpdump

Pour filtrer (sélectionner) les paquets que **tcpdump** doit capturer, vous pouvez utiliser les nombreux paramètres réseaux et champs des entêtes disponibles mettre des conditions que doivent vérifier les paquets à capturer (adresse source, destination, numéro de port, valeur d'un champ d'entête...). Un filtre est une suite d'expressions logiques reliées par des opérateurs logiques. L'expression peut porter sur :

- L'adresse :
 - **host** *A.B.C.D* (adresse machine),
 - **net** *A.B.C.D /M* (adresse réseau),
 - **port** *num_port* (adresse de port)
- La direction
 - **src** [*A.B.C.D / num_port*] (adresse source, hôte, net ou port)
 - **dst** [*A.B.C.D / num_port*] (destination)
- Le protocole
 - **proto** *nom_proto* (nom du protocole)

Exemples à tester (en tant que root);

```
tcpump -nvi eth1 host 10.0.0.1 #les trafics impliquant le 10.0.0.1
```

```
tcpump -nvi eth1 '(src 10.0.0.1) or (dst 10.0.0.1)' # même chose avec src et dst
```

```
tcpump -nvi eth1 host www.ugb.sn #version avec le nom d'une machine
```

```
tcpump -nvi eth1 net 10.0.0.0/8 #les trafics dans le réseau 10.0.0.0/8
```

```
tcpump -nvi eth1 port 80 #les trafics dans le réseau 10.0.0.0/8
```

```
tcpump -nvi eth1 src port 80 #les trafics dans le réseau 10.0.0.0/8
```

```
tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or 172.16.0.0/16
```

Consultez la page manuelle de **pcap-filter(7)** et **tcpdump(8)** de pour compléter votre lecture. Inspirez-vous de ces exemples pour donner les commandes qui permettent de capturer :

- a. Seulement les paquets du protocole ARP (mot clé arp)
- b. Tous les paquets sauf le numéro de port 22
- c. Tous les paquets qui sont destinés à **www.ugb.sn** sur le port 80 ou sur le port 22 lorsque le réseau source n'est pas 192.168.0.0/16
- d. Capturez les paquets du protocole ICMP
- e. Tous les paquets destinés à la machines 192.168.0.1 et qui sont destinés à des numéros de port entre 1 et 300
- f. Les paquets dont la taille est supérieure à 300octets
- g. Les paquets utilisant le protocole TCP

ACTION 3 : Wireshark : installation et lancement

Wireshark est aussi un outil de capture et d'analyse de trafic comme tcpdump qui offre une interface graphique (ne fonctionne que dans les environnements graphiques) et permet de faire bien plus de choses.

Installation :

Wireshark n'étant pas installé par défaut sur Ubuntu, vous pourrez l'installer sur le client Ubuntu Desktop. Pour cela, il suffit de taper la commande suivante :

```
sudo apt-get install wireshark
```

Lancement de wireshark (Gnome)

Pour ouvrir Wireshark sur Gnome, vous avez deux possibilités.

- Soit aller sur le menu Applications/Internet/Wireshark (comme indiqué sur la figure 5).

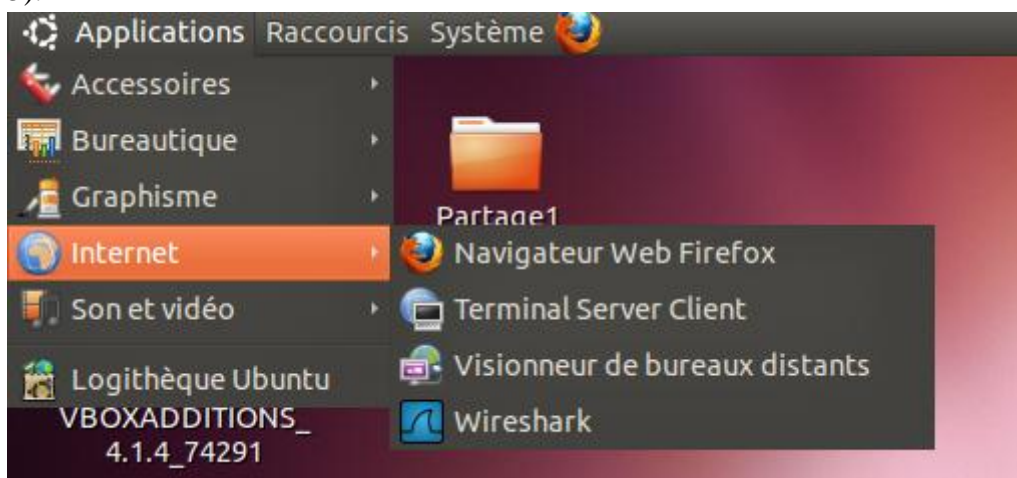


Figure 7 Ouvrir Wireshark sur le menu

- Vous pouvez aussi l'ouvrir en utilisant la ligne de commande en tapant tout simplement **wireshark**

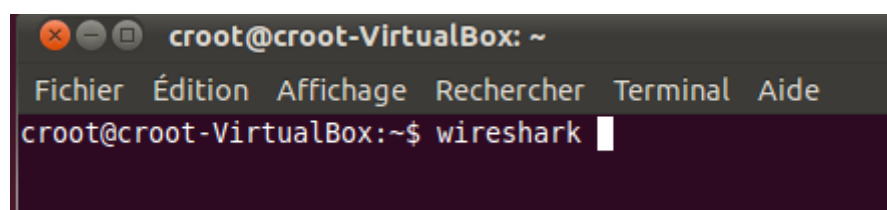


Figure 8 Lancement de Wireshark en mode ligne de commande

Vous obtenez la fenêtre principale de Wireshark

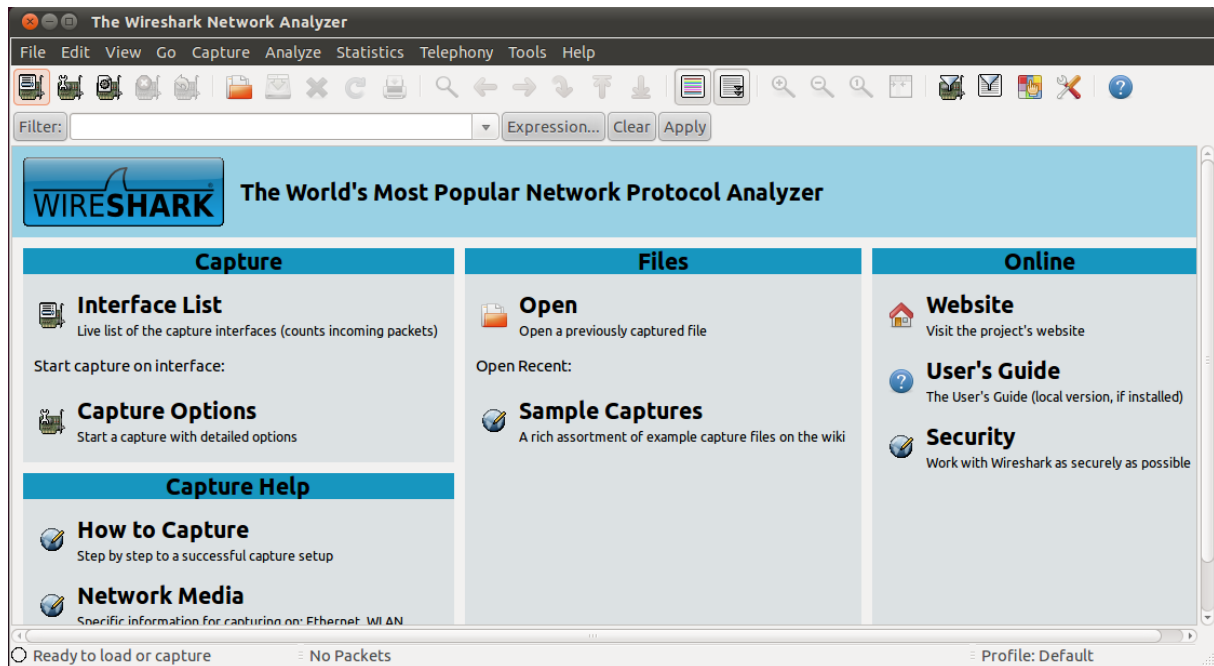


Figure 9 Fenêtre principale de Wireshark

Lorsque wireshark est ouvert en mode utilisateur simple, il n'est pas possible de capturer des paquets. Ce mode d'ouverture est utilisé pour lire des traces de paquets sauvegardées au format pcap par exemple. Pour faire des captures il faut ouvrir Wireshark en tant que root.

Pour reconnaître les différents formats il suffit de regarder la partie **interface list** qui contient la liste des interfaces sur lesquelles vous pouvez capturer des paquets. Lorsqu'elle est vide il s'agit du mode utilisateur simple.

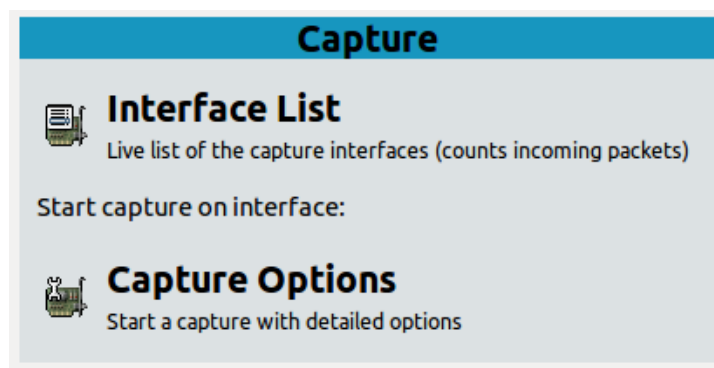


Figure 10 Liste des interfaces en mode utilisateur simple

Pour ouvrir en tant que root wireshark il suffit taper la commande (vous aurez un message de notification):

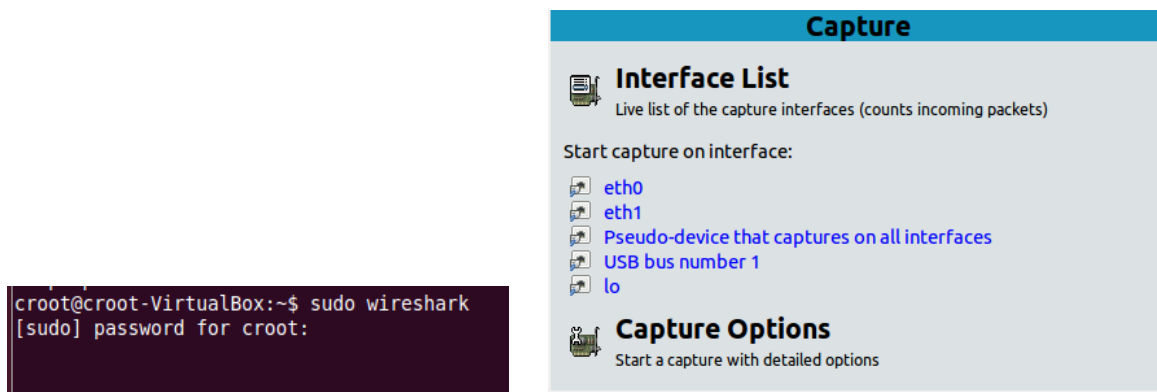


Figure 11 Lancement en mode administrateur et liste des interfaces

ACTION 4 : Wireshark : Capture et filtres

1. Pour capturer le trafic sur une interface, Allez sur la liste des interfaces et cliquez sur l'interface en question (`eth1` par exemple)³. Puis, générez du trafic avec ping

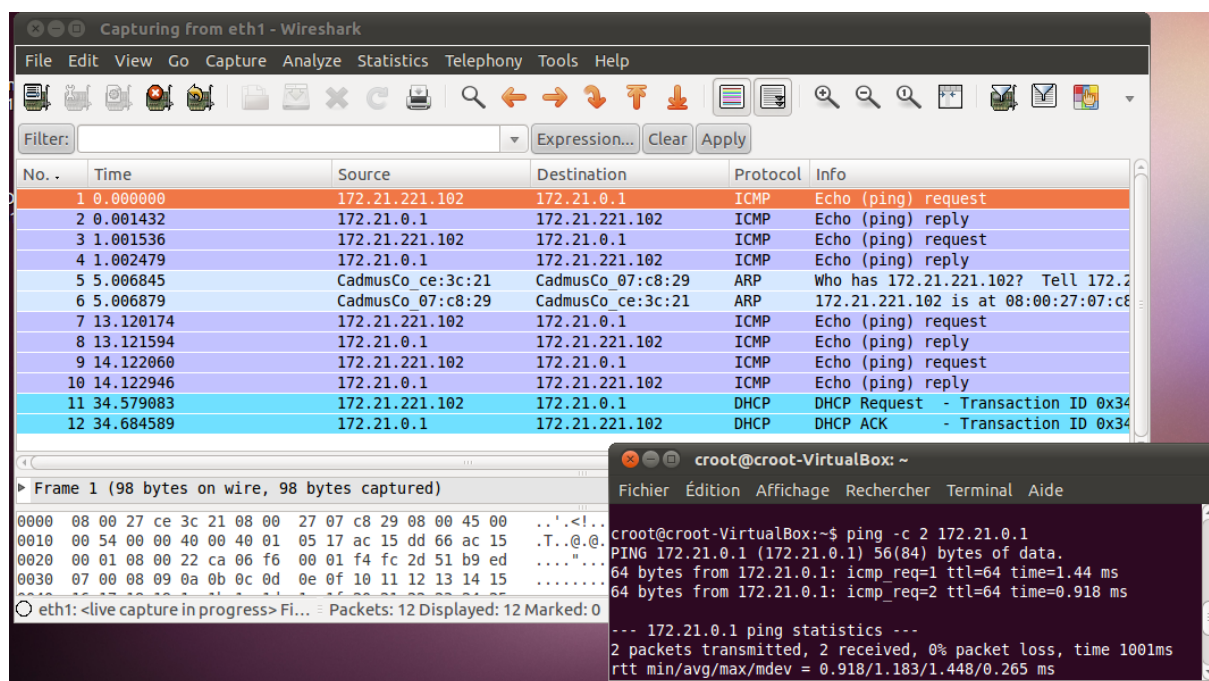
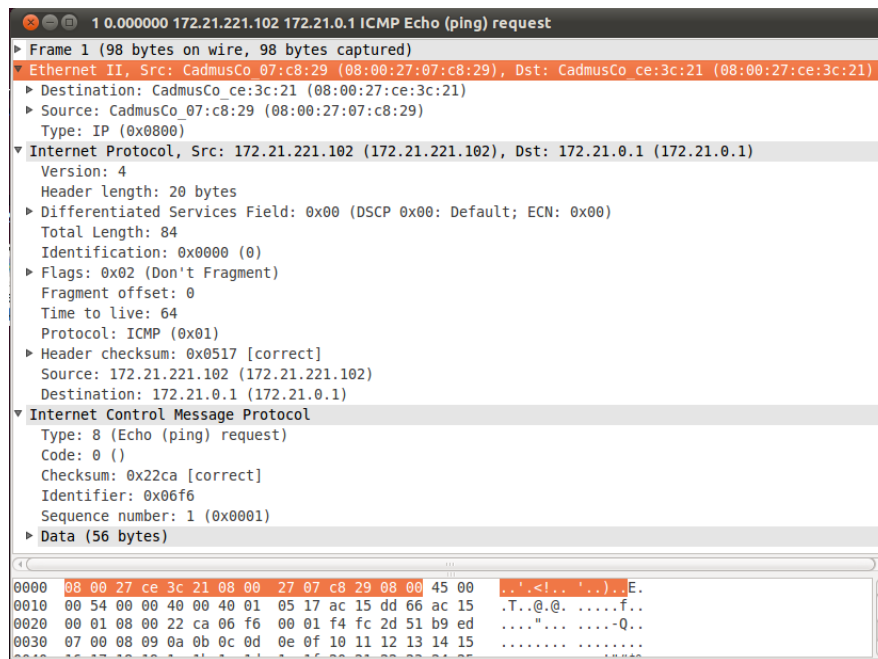


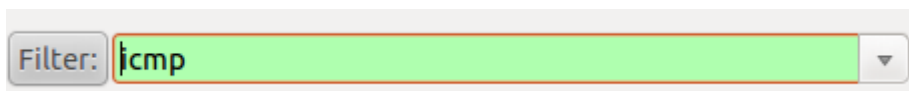
Figure 12 Capture de base d'un paquet

2. Double Cliquez sur un paquet ICMP (une des lignes ou la colonne Protocol contient ICMP)

³ Si aucune interface n'apparaît alors que vous êtes en mode root, vérifiez que vous l'avez bien activé l'interface sur VirtualBox et configurée avec **ifconfig**



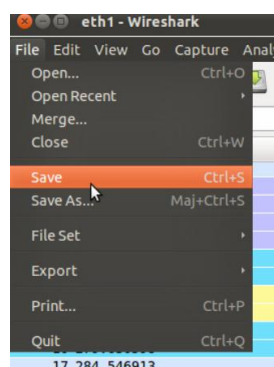
3. Pour filter les paquets, vous pouvez utiliser les mêmes expressions que tcpdump dans le champs filter . Taper **icmp** comme filtre comme sous l'image



Gérez des *pings* et observez les paquets affichés



4. Vous pourrez arrêter la capture à tout moment avec le bouton
5. Sauvegardez la capture au format PCAP avec le nom /home/croot/capture1.pcap



6. En utilisant le menu File/Open ouvrez le fichier que vous aviez capturé à la question Action 2 question 6 avec TCPDUMP
7. Téléchargez la documentation de Wireshark à l'adresse <http://www.wireshark.org/docs/> pour vous familiariser les différents objets de la barre d'outils.