

Commandes de base Administration Systèmes et Réseaux

- **TcpDump**

Sniffing en mode verbeux :

```
> tcpdump -v tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

22:00:11.625995 IP (tos 0x0, ttl 128, id 30917, offset 0, flags [none], proto: UDP (17), length: 81) 192.168.1.2.1034 > valve-68-142-64-164.phx3.llnw.net.27014: UDP, length 53
22:00:20.691903 IP (tos 0x0, ttl 128, id 31026, offset 0, flags [none], proto: UDP (17), length: 81) 192.168.1.2.1034 > valve-68-142-64-164.phx3.llnw.net.27014: UDP, length 53
22:00:21.230970 IP (tos 0x0, ttl 114, id 4373, offset 0, flags [none], proto: UDP (17), length: 64) valve-68-142-64-164.phx3.llnw.net.27014 > 192.168.1.2.1034 UDP, length 36
22:00:26.201715 arp who-has 192.168.1.2 tell 192.168.1.1
22:00:26.201726 arp reply 192.168.1.2 is-at 00:04:11:11:11:11 (oui Unknown)
22:00:29.706020 IP (tos 0x0, ttl 128, id 31133, offset 0, flags [none], proto: UDP (17), length: 81) 192.168.1.2.1034 > valve-68-142-64-164.phx3.llnw.net.27014: UDP, length 53
22:00:38.751355 IP (tos 0x0, ttl 128, id 31256, offset 0, flags [none], proto: UDP (17), length: 81) 192.168.1.2.1034 > valve-68-142-64-164.phx3.llnw.net.27014: UDP, length 53
```

Connaître les interfaces sur lesquelles on peut écouter :

```
> tcpdump -D
1.eth0
2.any (Pseudo-device that captures on all interfaces)
3.lo
```

Afficher les IP et non les résolutions DNS :

```
#tcpdump -n tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

22:02:36.111595 IP 192.168.1.2.1034 > 68.142.64.164.27014: UDP, length 53
22:02:36.669853 IP 68.142.64.164.27014 > 192.168.1.2.1034: UDP, length 36
```

Afficher les paquets en sortie :

```
> tcpdump -q  
22:03:55.594839 IP a213-22-130-46.cpe.netcabo.pt.3546 > 192.168.1.2.9501:  
tcp 0  
22:03:55.698827 IP 192.168.1.2.9501 > a213-22-130-46.cpe.netcabo.pt.3546:  
tcp 0  
22:03:56.068088 IP a213-22-130-46.cpe.netcabo.pt.3546 > 192.168.1.2.9501:  
tcp 0  
22:03:56.068096 IP 192.168.1.2.9501 > a213-22-130-46.cpe.netcabo.pt.3546:  
tcp 0
```

Sniffer sur une interface en particulier :

```
> tcpdump -i eth0
```

Sniffer uniquement les paquets UDP :

```
> tcpdump udp
```

Sniffer les paquets d'un service en particulier :

```
> tcpdump port http
```

Sniffer un certain nombre de paquets puis arrêter :

```
> tcpdump -c 20
```

Enregistrer le contenu du dump dans un fichier

```
> tcpdump -w capture.log
```

Afficher les paquets qui ont pour source et destination www.serveur-monty.net

```
> tcpdump host www.serveur-monty.net
```

Afficher les paquets FTP ayant pour source 192.168.1.100 et destination 192.168.1.2 :

```
> tcpdump src 192.168.1.100 and dst 192.168.1.2 and port ftp
```

- Iperf

Tester rapidement le debit entre sa machine et une machine du réseau :

```
iperf -c 192.168.1.1 -p 22 -fM
```

- Netstat

Afficher les routes de la machine :

```
> netstat -rn (ou route tout simplement :o) )
```

Afficher des informations rapides sur les interfaces de la machine :
(trafic entrant, sortant, paquet echangés ...)

```
> netstat -i
```

Voir tout les socket ouvert (u=udp, t=tcp, a=all):

```
> netstat -uta
```

Explication sur les differents état de connexion des sockets :

```
ESTABLISHED : The socket has an established connection.  
SYN_SENT : The socket is actively attempting to establish a connection.  
SYN_RECV : A connection request has been received from the network.  
FIN_WAIT1 : The socket is closed, and the connection is shutting down.  
FIN_WAIT2 : Connection is closed, and the socket is waiting for a shutdown from the remote end.  
TIME_WAIT : The socket is waiting after close to handle packets still in the network.  
CLOSED : The socket is not being used.  
CLOSE_WAIT : The remote end has shut down, waiting for the socket to close.  
LAST_ACK : The remote end has shut down, and the socket is closed. Waiting for acknowledgement.  
LISTEN: The socket is listening for incoming connections. Such sockets are not included in the output unless you specify the --listening (-l) or --all (-a) option.  
CLOSING : Both sockets are shut down but we still don't have all our data sent.  
UNKNOWN : The state of the socket is unknown.
```

Informations plus étendues sur les sockets en cours (e=extended) :

```
> netstat -aute
```

Avoir des statistiques sur chaque protocoles en cours :

```
> netstat -s
```

Avoir le maximum d'informations :

```
> netstat -tunap
```

- **Dig**

Intérogation simple d'un domaine :

```
> dig serveur-monty.net
```

Résolution inverse (reverse DNS) d'un domaine :

```
> dig -x 82.246.231.218  
[...]  
;; ANSWER SECTION:  
218.231.246.82.in-addrarpa. 86400 IN PTR  serveur-monty.net.  
[...]
```

- **Nslookup**

Intérogation avec les DNS de la machine locale :

```
> nslookup www.serveur-monty.net
```

Intérogation en utilisant des DNS externes :

```
> nslookup www.serveur-monty.net -ns1.orange.fr
```

- **Nmap**

Voir tous les ports TCP ouverts sur une machine, utilisation de messages SYN, donc pas de log sur la machine cible :

```
nmap -sS 127.0.0.1
```

Voir tous les ports UDP ouverts sur une machine :

```
nmap -sU 127.0.0.1
```

Voir si une machine est sur le réseau (scan Ping) :

```
nmap -sP 127.0.0.1
```

Scanner une plage d'adresses. Ici toutes les adresses de 192.168.0 à 192.168.255 :

```
nmap 192.168.0,0-255
```

Connaitre le système d'exploitation de la machine (TCP/IP fingerprint) :

```
nmap -O 127.0.0.1
```

Si nmap n'arrive pas à déterminer la version, on pourra lui demander de nous donner une liste des systèmes qui pourraient potentiellement correspondre :

```
nmap -O --osscan-guess 127.0.0.1
```

Scanner un port précis. Ici, c'est le port http :

```
nmap -p 80 127.0.0.1
```

Scanner une plage de ports. Ici on scan du port 0 au 80 et tous ceux supérieurs à 60000) :

```
nmap -p 0-80,60000 127.0.0.1
```

Scanner des serveurs web au hasard sur le réseau :

```
nmap -v -sS -iR 0 -p 80
```

Désactiver la résolution DNS inverse des hôtes, augmente la rapidité :

```
nmap -n 127.0.0.1
```

Scan par rebond ftp, permet de demander à un serveur FTP de scanner les ports à

votre place (envoi des fichiers pour tester les ports ouverts). Cette fonctionnalité est souvent désactivée des serveurs FTP afin d'éviter les abus. Ici on passe par le serveur ftp qui a pour adresse 127.0.0.1 pour scanner une plage d'adresses ip :

```
nmap -b 127.0.0.1 192.168.0.,0-255
```

Usurper l'adresse ip source. Ici on scan 127.0.0.1, par l'interface réseau eth0, en se faisant passer pour 10.0.0.0 depuis le port 80 :

```
nmap -S 10.0.0.0 -g 80 -e eth0 -P0 127.0.0.1
```

Usurper l'adresse MAC :

```
nmap --spooof-mac 01:02:03:04:05:06 127.0.0.1  
nmap --spooof-mac Cisco 127.0.0.1
```

Choisir un fichier de sortie pour y écrire les résultats du scan :

```
nmap -oN resultat 127.0.0.1  
nmap -oX resultat.xml 127.0.0.1
```

Trace les paquets et les données envoyés et reçus. Pratique pour vérifier qu'une usurpation fonctionne :

```
nmap --packet-trace -S 10.0.0.0 -eth0 127.0.0.1
```

2. Solution

Empêcher le balayage des ports d'une machine reste assez difficile en soi. En effet, même en rajoutant des règles à iptables, les techniques de scan étant tellement diverses, cela ne sera ne fonctionnera pas à 100%.

En revanche, on peut très bien utiliser des outils spécialisés dans la détection de ces derniers comme [scanlogd](#) par exemple.

Pour s'en servir, nous allons devoir récupérer les sources sur le site officiel et lancer les commandes suivantes :

```
cd /usr/local/src/  
tar zxvf scanlogd-*.tar.gz  
rm -f scanlogd-*.tar.gz  
cd scanlogd-*/
```

```
make linux
adduser scanlogd
```

On pourra ensuite le lancer manuellement, via la commande **scanlogd**. Toutes les tentatives de scan sur la machine seront alors visibles dans le fichier **/var/log/messages** :

```
# tailf /var/log/messages | grep scanlogd
Dec 3 17:54:43 localhost scanlogd: 192.168.0.188 to 192.168.0.175
ports 80, 554, 256, 21, 22, 23, ..., TOS 00, TTL 64 @18:54:43
```

- **Ethtool**

Obtenir des informations sur les capacités de sa carte réseau :

```
> ethtool eth0
```

Ajustement manuel de la vitesse de transmission :

```
> ethtools speed 10|100|1000
```

Ajustement full ou half duplex :

```
> ethtool dupleix half|full
```

Autonegociation :

```
> ethtool autoneg on|off
```

Changer l'adresse physique :

```
> ethtool phyad 00:11:22:33:44:55
```