

6 - CONFIGURER TCP/IP EN ENVIRONNEMENT LINUX.



Ajouter un système dans un réseau Ipv4 / Ipv6 - Les commandes de diagnostics - Le fonctionnement des systèmes INETD – Les wrappers.

6 - CONFIGURER TCP/IP EN ENVIRONNEMENT LINUX.

- 6-1. Ajouter un système (Debian, RedHat)dans un réseau Ipv4 / Ipv6.
- 6-2. Les commandes de diagnostics.
- 6-3. Le fonctionnement des systèmes INETD.
- 6-4. Les wrappers.

6-I AJOUTER UN SYSTÈME LINUX À UN RÉSEAUX IPV4 OU IPV6

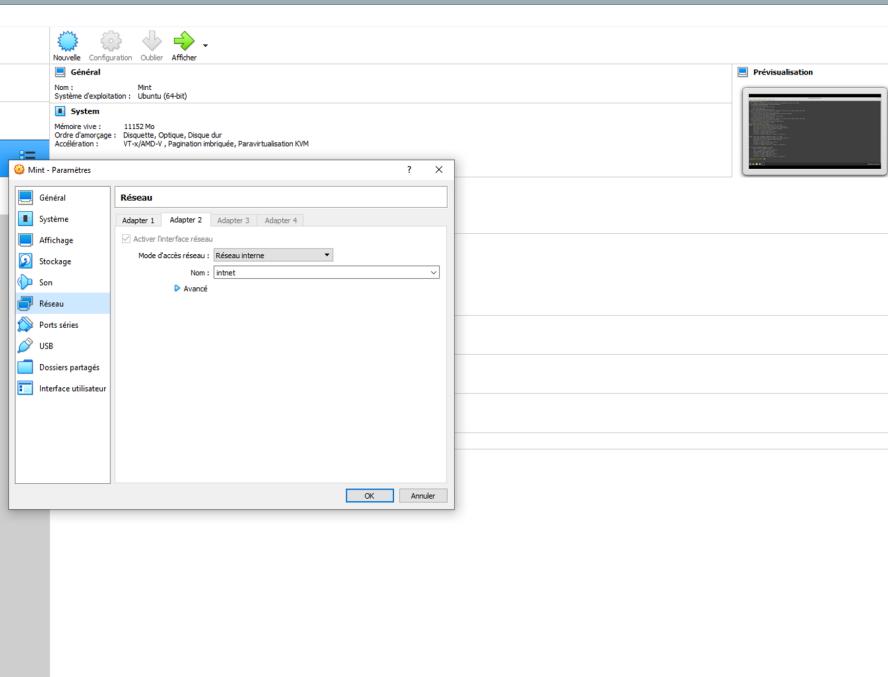
- Quelques définitions de base :
 - **Le réseau informatique** est un ensemble d'équipements reliés entre eux pour échanger des informations.
 - **Un hôte** est une machine sur un réseau.
 - **Une Adresse IP** permet d'identifier une machine (hôte) sur un réseau.
 - Il existe 2 types d'adresse IP : **Ipv4** sur 32 bits et **Ipv6** sur 64 bits.
 - Les machines communiquent entre elles via un protocole bien défini, le **protocole TCP/IP**.
 - L'adresse **127.0.0.1** représente la machine elle-même sur un réseau. On l'appelle **localhost**.
 - Un **masque réseau** est le délimiteur entre la partie réseau et la partie machine (ex : 255.255.255.0).
 - Un **réseau local** (LAN : Local Area Network) ou sous-réseau est le réseau local sur lequel une ou plusieurs machines peuvent être connectées et qui est rattaché à un réseau plus grand (**WAN : Wide Area Network**) sur lequel se trouve d'autres LAN.



6-1 AJOUTER UN SYSTÈME LINUX A UN RÉSEAU IPV4

EN MODE MANUEL

Simulation avec Virtual Box



- Dans virtualBox, nous allons créer un nouveau point de connexion pour notre distribution.
1. Eteindre la distribution si allumée.
 2. Sélectionner votre distribution .
 3. Cliquer sur Configuration > Réseau > Adapter 2 :
 - **Active l'interface réseau : coché.**
 - **Mode d'accès réseau : Réseau interne**
 - **Validation avec ok.**

6-1 AJOUTER UN SYSTÈME LINUX A UN RÉSEAU IPV4

EN MODE MANUEL

Simulation avec Virtual Box

```
# Affiche toutes les interfaces de connexion avec les informations relatives à leurs adresse IP.  
# On apperçoit notamment la nouvelle interface réseau : enp0s8  
alain@mohamed-VirtualBox:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inetc6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:3b:a5:90 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 86287sec preferred_lft 86287sec  
    inetc6 fe80::1f53:ee4:977e:5e09/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:ec:a5:2e brd ff:ff:ff:ff:ff:ff  
    inetc6 fe80::8f4a:b4df:e4b2:97ca/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
  
# Affiche toutes les interfaces de connexion avec les informations relatives à leurs adresse IP.  
# On apperçoit notamment la nouvelle interface réseau : enp0s8  
alain@mohamed-VirtualBox:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inetc6 fe80::1f53:ee4:977e:5e09 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:3b:a5:90 txqueuelen 1000 (Ethernet)  
    RX packets 77 bytes 10380 (10.3 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 123 bytes 14096 (14.0 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inetc6 fe80::8f4a:b4df:e4b2:97ca prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:ec:a5:2e txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 84 bytes 13481 (13.4 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inetc6 127.0.0.1 netmask 255.0.0.0  
    inetc6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Boucle locale)  
    RX packets 231 bytes 20289 (20.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 231 bytes 20289 (20.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
# crée un fichier yaml pour la configuration réseau de la nouvelle interface réseau créée : enp0s8  
root@mohamed-VirtualBox:/home/alain# touch /etc/netplan/2-lan_statique.yaml
```

- Allumons la distribution modifiée.
- Vérifions que la modification a bien été prise en compte :
 - Saisir « **ip a** » ou « **ifconfig** ».
 - Un nouveau point de connexion (interface) doit apparaître :
 - **enpS08**
- On va utiliser **netplan** qui est un programme qui va nous permettre d'écrire nos configurations réseaux dans un fichier yaml.
- Création d'un fichier de configuration au format yaml :
 - « **touch /etc/netplan/2-lan_statique.yaml** »

6-1 AJOUTER UN SYSTÈME LINUX A UN RÉSEAU IPV4

```
root@mohamed-VirtualBox:/home/alain# nano /etc/hosts
root@mohamed-VirtualBox:/home/alain# ip link set enp0s3 down
root@mohamed-VirtualBox:/home/alain# ip link set enp0s8 up

# L'interface réseau enp0s8 est bien active avec une adresse IP4 : 192.168.0.10 (inet)
# L'interface réseau enp0s3 est désactivée.
root@mohamed-VirtualBox:/home/alain# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:3b:a5:90 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ec:a5:2e brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.10/24 brd 192.168.0.255 scope global noprefixroute enp0s8
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:feec:a52e/64 scope link
            valid_lft forever preferred_lft forever

# Les paquets sont bien transmis aucune perte. L'adresse ip est bien fonctionnel.
root@mohamed-VirtualBox:/home/alain# ping -c 4 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 octets de 192.168.0.10 : icmp_seq=1 ttl=64 temps=0.031 ms
64 octets de 192.168.0.10 : icmp_seq=2 ttl=64 temps=0.045 ms
64 octets de 192.168.0.10 : icmp_seq=3 ttl=64 temps=0.074 ms
64 octets de 192.168.0.10 : icmp_seq=4 ttl=64 temps=0.046 ms

--- statistiques ping 192.168.0.10 ---
4 paquets transmis, 4 reçus, 0 % paquets perdus, temps 3052 ms
rtt min/avg/max/mdev = 0.031/0.049/0.074/0.015 ms
```

EN MODE MANUEL

Simulation avec Virtual Box

- Rajouter la nouvelle adresse IP (192.168.0.10/24) au fichier **/etc/hosts** avec la commande : « nano /etc/hosts »
 - **192.168.0.10 'nom du serveur'**
- Désactiver l'interface enp0s3 : **« ip link set enp0s3 down ».**
- Activer l'interface réseau que l'on a créée : **« ip link set enp0s8 up ».**
- Vérifier l'état des interfaces et s'assurer que les modifications ont bien été prise en compte : « ip a ».
- Vérifier que l'interface réseau est fonctionnel et que notre serveur est intégré au réseau local et non plus à celui de VB avec la commande : **« ping 192.168.0.10 »**

6-2 LES COMMANDES DE DIAGNOSTICS

- Il existe beaucoup de commandes qui vont permettre d'effectuer un diagnostic du réseaux sur lequel est branché votre machine.
- Nous allons voir les principales commandes qui vont permettre d'avoir une vue sous différents angles de votre réseau :
 - IFCONFIG – IP.
 - PING – HOST – DIG.
 - NMAP – TRACEROUTE.
 - NETSTAT – IFTOP.
 - DCPFUMP - NGREP



6-2 LES COMMANDES DE DIAGNOSTICS

```
root@mohamed-VirtualBox:/home/alain# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::1f53:ee4:977e:5e09 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:3b:a5:90 txqueuelen 1000 (Ethernet)
            RX packets 12222 bytes 12475822 (12.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 14203 bytes 1397349 (1.3 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Boucle locale)
            RX packets 9218 bytes 621628 (621.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 9218 bytes 621628 (621.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mohamed-VirtualBox:/home/alain# ip -4 -o addr show
1: lo    inet 127.0.0.1/8 scope host lo\      valid_lft forever preferred_lft forever
2: enp0s3  inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3\
valid_lft 85505sec preferred_lft 85505sec
3: enp0s8  inet 192.168.0.10/24 brd 192.168.0.255 scope global enp0s8\
valid_lft forever preferred_lft forever
```

LES COMMANDES : IFCONFIG – IP

IFCONFIG :

- **ifconfig** est une commande Unix qui permet de configurer et d'afficher les informations des interfaces réseau IP à partir de l'interpréteur de commandes.
- « **man ifconfig** » : liste des commandes.

IP :

- Commande qui tend à remplacer **ifconfig**.
- « **ip a** » : Affiche toutes les adresses IP d'un réseau.
- « **Ip addr add 192.168.1.5/24 dev eth0** » : attribue une adresse ip à l'interface eth0.
- « **ip -4 -o addr show** » : affiche les informations sur les interfaces réseaux avec une Ipv4 sur une ligne.
- « **ip addr del 192.168.1.5/24 dev eth0** » : supprime l'adresse ip de l'interface eth0.
- « **ip link set eth0 up** » : active l'interface réseau.
- « **ip link set eth0 down** » : désactive l'interface réseau.
- « **man ip** » : liste des commandes.

6-2 LES COMMANDES DE DIAGNOSTICS

```
root@mohamed-VirtualBox:/home/alain# ping -c 4 google.fr
PING google.fr (142.250.201.163) 56(84) bytes of data.
64 octets de par2ls23-in-f3.1e100.net (142.250.201.163) : icmp_seq=1 ttl=113 temps=33.9 ms
64 octets de par2ls23-in-f3.1e100.net (142.250.201.163) : icmp_seq=2 ttl=113 temps=65.5 ms
64 octets de par2ls23-in-f3.1e100.net (142.250.201.163) : icmp_seq=3 ttl=113 temps=48.2 ms
64 octets de par2ls23-in-f3.1e100.net (142.250.201.163) : icmp_seq=4 ttl=113 temps=69.2 ms

--- statistiques ping google.fr ---
4 paquets transmis, 4 reçus, 0 % paquets perdus, temps 3005 ms
rtt min/avg/max/mdev = 33.919/54.209/69.219/14.148 ms

root@mohamed-VirtualBox:/home/alain# host www.google.fr
www.google.fr has address 216.58.214.163
www.google.fr has IPv6 address 2a00:1450:4007:81a::2003

root@mohamed-VirtualBox:/home/alain# dig www.google.fr
<>> DiG 9.16.1-Ubuntu <>> www.google.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 32515
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.google.fr.      IN  A

;; ANSWER SECTION:
www.google.fr.    137 IN  A   216.58.206.227

;; Query time: 52 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: sam. sept. 04 11:52:54 CEST 2021
;; MSG SIZE  rcvd: 58
```

LES COMMANDES : PING – HOST - DIG

PING :

- **ping** est un outil d'administration qui permet de diagnostiquer l'accessibilité d'une machine à travers un réseau.
- Sa mission principale consiste à vérifier les connexions établies entre un ou plusieurs hôtes distants
- La commande « **ping -c 4 www.google.fr** » : va envoyer 4 paquets sur le réseau à l'adresse IP du serveur sur lequel se trouve Google.

Host :

- **Host** permet de convertir des DNS en adresse ip.
- Exemple : « **host www.google.fr** ».

DIG :

- **Dig (Domain Information Groper)** est un outil très complet pour effectuer des requêtes DNS.
- Exemple : « **dig www.google.fr** ».

6-2 LES COMMANDES DE DIAGNOSTICS

```
root@mohamed-VirtualBox:/home/alain# nmap -v www.leboncoin.fr
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-04 11:17 CEST
Initiating Ping Scan at 11:17
Scanning www.leboncoin.fr (52.222.174.122) [4 ports]
Completed Ping Scan at 11:17, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:17
Completed Parallel DNS resolution of 1 host. at 11:17, 0.06s elapsed
Initiating SYN Stealth Scan at 11:17
Scanning www.leboncoin.fr (52.222.174.122) [1000 ports]
Discovered open port 443/tcp on 52.222.174.122
Discovered open port 21/tcp on 52.222.174.122
Discovered open port 80/tcp on 52.222.174.122
Completed SYN Stealth Scan at 11:17, 4.79s elapsed (1000 total ports)
Nmap scan report for www.leboncoin.fr (52.222.174.122)
Host is up (0.016s latency).
Other addresses for www.leboncoin.fr (not scanned): 52.222.174.18 52.222.174.76 52.222.174.10
rDNS record for 52.222.174.122: server-52-222-174-122.cdg50.r.cloudfront.net
Not shown: 997 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
443/tcp   open  https

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.04 seconds
Raw packets sent: 2003 (88.100KB) | Rcvd: 6 (252B)

C:\Users\afpa>tracert -4 -h 10 www.google.fr
Détermination de l'itinéraire vers www.google.fr [142.250.179.67]
avec un maximum de 10 sauts :

 1  <1 ms   <1 ms   <1 ms  192.168.0.254
 2  6 ms    6 ms    6 ms  194.149.169.174
 3  7 ms    7 ms    6 ms  194.149.166.54
 4  7 ms    6 ms    6 ms  72.14.220.92
 5  6 ms    6 ms    6 ms  108.170.231.95
 6  6 ms    7 ms    6 ms  142.251.49.131
 7  7 ms    6 ms   12 ms  par21s19-in-f3.1e100.net [142.250.179.67]

Itinéraire déterminé.
```

LES COMMANDES : TRACEROUTE - NMAP

NMAP :

- **NMAP** est un scanner de port libre. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant.
- **« nmap -vv www.leboncoin.fr ».**
- **« man nmap »** : pour le détails d'autres commandes.

TRACEROUTE :

- **TRACEROUTE** est un programme utilitaire qui permet de suivre les chemins qu'un paquet de données (paquet IP) va prendre pour aller de la machine locale à une autre machine connectée au réseau IP.
- **« traceroute -4 10 google.com »** : affiche les 10 premiers sauts vers le DNS google.com pour les ipv4.
- **« man traceroute »** : pour le détails d'autres commandes.

6-2 LES COMMANDES DE DIAGNOSTICS

```
root@mohamed-VirtualBox:/home/alain# netstat -e
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale      Adresse distante     Etat      Utilisatr Inode
tcp      0      0 mohamed-VirtualBo:41010  ec2-44-239-97-185:https ESTABLISHED alain    119584
udp      0      0 mohamed-VirtualB:bootpc _gateway:bootps   ESTABLISHED root     102447
Sockets du domaine UNIX actives (sans serveurs)
Proto RefCnt Flags     Type      State          I-Node  Chemin
unix  2      [ ]  DGRAM    CONNECTE      29968  /run/user/1001/systemd/notify
unix  3      [ ]  DGRAM    CONNECTE      15857  /run/systemd/notify
unix  2      [ ]  DGRAM    CONNECTE      15875  /run/systemd/journal/syslog
unix 19      [ ]  DGRAM    CONNECTE      15885  /run/systemd/journal/dev-log
unix  8      [ ]  DGRAM    CONNECTE      15889  /run/systemd/journal/socket
unix  3      [ ]  STREAM   CONNECTE    92880
unix  3      [ ]  STREAM   CONNECTE    31407  @/tmp/dbus-RD1V5rSxFx
unix  2      [ ]  DGRAM    CONNECTE    42652
unix  3      [ ]  STREAM   CONNECTE    31690  /run/user/1001/pulse/native
unix  3      [ ]  STREAM   CONNECTE    20283  /run/systemd/journal/stdout
unix  2      [ ]  STREAM   CONNECTE    94041
unix  3      [ ]  STREAM   CONNECTE    32249
unix  3      [ ]  STREAM   CONNECTE    33105
unix  3      [ ]  STREAM   CONNECTE    31085  /run/systemd/journal/stdout
unix  3      [ ]  STREAM   CONNECTE    118526  /run/dbus/system_bus_socket
unix  3      [ ]  STREAM   CONNECTE    92895
unix  3      [ ]  STREAM   CONNECTE    92870
```

```
Fichier Édition Affichage Rechercher Terminal Aide
1,91Mb      3,81Mb      5,72Mb      7,63Mb      9,54Mb
mohamed-VirtualBox  => par21s17-in-f14.1e100.net  476b  1,58Kb  1,26Kb
mohamed-VirtualBox  <=                           476b  417b  279b
mohamed-VirtualBox  => par21s17-in-f3.1e100.net  320b  96b   48b
mohamed-VirtualBox  <=                           320b  96b   48b
mohamed-VirtualBox  => par21s20-in-f22.1e100.net  476b  95b   24b
mohamed-VirtualBox  <=                           476b  95b   24b
mohamed-VirtualBox  => 93.184.220.29           0b    0b   40b
mohamed-VirtualBox  <=                           0b    0b   40b
mohamed-VirtualBox  => ec2-52-50-19-116.eu-west- 0b    0b   25b
mohamed-VirtualBox  <=                           0b    0b   25b
mohamed-VirtualBox  => par10s39-in-f1.1e100.net  0b    0b   24b
mohamed-VirtualBox  <=                           0b    0b   24b
mohamed-VirtualBox  => fra15s10-in-f6.1e100.net  0b    0b   24b
mohamed-VirtualBox  <=                           0b    0b   24b
mohamed-VirtualBox  => 104.19.183.2            0b    0b   24b
mohamed-VirtualBox  <=                           0b    0b   24b
mohamed-VirtualBox  => 221.209.102.34.bc.googleu 0b    0b   24b
mohamed-VirtualBox  <=                           0b    0b   24b

TX:          cum:  500KB  peak:  11,6Kb  rates:  1,24Kb  1,77Kb  1,76Kb
RX:              1,88MB  5,48Kb  1,24Kb  608b   797b
TOTAL:        2,37MB  17,1Kb  2,48Kb  2,36Kb  2,54Kb
```

LES COMMANDES : NETSTAT - IFTOP

NETSTAT :

- **netstat**, pour « network statistics », est une ligne de commande affichant des informations sur les connexions réseau, les tables de routage et un certain nombre de statistiques.
- « **netstat -e** » : affiche les statistiques ethernet.

IFTOP :

- **iftop** fait partie des commandes "top" mais pour le réseau qui permet de visualiser en temps réel le débit par adresses contactées.
- « **iftop** » : Affiche les informations concernant les débits consommés lors des appels d'adresses IP.

6-2 LES COMMANDES DE DIAGNOSTICS



```
root@mohamed-VirtualBox:/home/alain# tcpdump -i enp0s3 host 142.250.178.131
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
10:29:01.535635 IP mohamed-VirtualBox.42070 > par21s22-in-f3.1e100.net.443: UDP, length 1357
10:29:01.611182 IP par21s22-in-f3.1e100.net.443 > mohamed-VirtualBox.42070: UDP, length 1357
10:29:01.611394 IP par21s22-in-f3.1e100.net.443 > mohamed-VirtualBox.42070: UDP, length 1357
10:29:01.611652 IP par21s22-in-f3.1e100.net.443 > mohamed-VirtualBox.42070: UDP, length 1357
10:29:01.612195 IP mohamed-VirtualBox.42070 > par21s22-in-f3.1e100.net.443: UDP, length 86
10:29:01.614215 IP par21s22-in-f3.1e100.net.443 > mohamed-VirtualBox.42070: UDP, length 1357
10:29:01.614309 IP mohamed-VirtualBox.42070 > par21s22-in-f3.1e100.net.443: UDP, length 42
10:29:01.614679 IP par21s22-in-f3.1e100.net.443 > mohamed-VirtualBox.42070: UDP, length 446
10:29:01.617932 IP mohamed-VirtualBox.42070 > par21s22-in-f3.1e100.net.443: UDP, length 43
```

LES COMMANDES : TCPDUMP - NGREP

TCPDUMP :

- **TCPDUMP** capture et affiche les paquets réseaux échangés par une ou plusieurs interfaces réseaux.
- On peut rediriger ce flux vers un fichier texte.
- « **tcpdump -i enps03** » : écoute l'échange de paquets entre l'interface enps03 et les autres interfaces.
- « **tcpdump -i enps03 host 142.250.178.131** » : écoute l'échange de paquets entre l'interface enps03 et l'hôte à l'adresse ip (142.250.178.131).
- « **tcpdump src 13.225.25.50** » : Ecoute les échanges de paquets en prévision de l'adresse ip 13.225.25.50.
- « **tcpdump dst 13.225.25.125** » : Ecoute les échanges de paquets à destination de 13.255.25.125

NGREP :

- **NGREP** fonctionne comme tcpdump à la seule différence qu'il ne va afficher que les strings des paquets.
- « **ngrep -d enps03 host 142.250.178.131** » : écoute l'échange de paquets entre l'interface enps03 et l'hôte à l'adresse ip (142.250.178.131).

6-3 LE FONCTIONNEMENT DES SYSTÈMES INETD (XINETD)

- XINETD (e**X**tend Inter**N**ET **D**aemon) version plus sécurisée d'INETD est un démon qui a comme rôle de piloter l'accès à un ou plusieurs réseaux.
- En fonction de sa configuration, il pourra sécuriser et contrôler l'accès à votre server en contrôlant et vérifiant les requêtes qui transitent sur le réseau vers votre server.
- Ses principales fonctionnalités sont :
 - Paramétrage d'accès par service et non pas de manière globale.
 - Paramétrage d'accès par créneaux ou plages horaires pour des services.
 - Possibilité de limiter les attaques de type deny of service par exemple ou autres.
 - Possibilité d'affiner les logs des services gérés.
- S'il est pas installé ou présent: « **apt install xinetd** ».



6-3 LE FONCTIONNEMENT DES SYSTÈMES INETD (XINETD)

- Il existe 2 types de configuration d'accès au fichier :
 - I. Un accès via **/etc/xinetd.conf** avec un seul fichier qui prendra la configuration globale et par service.
 - I. Un accès via /etc/ mais avec 1 fichier et 1 dossier :
 - Un fichier **/etc/xinetd.conf** : configuration générale.
 - Un dossier **/etc/xinetd.d** : avec plusieurs fichiers de configuration dédiés chacun d'eux à un service.
- **La seconde configuration avec les 2 fichiers est la plus courante dans les distributions Linux.**



6-3 LE FONCTIONNEMENT DES SYSTÈMES INETD (XINETD)

```
# Affiche le dossier de configuration (xinetd.d) des services et le fichier de configuration générale (xinetd.conf) Xinetd
root@mohamed-VirtualBox:/home/alain# ls /etc/xinetd/
/etc/xinetd.conf

/etc/xinetd.d:
chargen    daytime    discard    echo    servers    time
chargen-udp  daytime-udp  discard-udp  echo-udp  services  time-udp

# Fichier de configuration générale avec exemple : 60 requêtes maximal - journalisation
/var/log/secure - nom hôte et numéro processus en cas de succès de connexion - nom de l'hôte en cas d'échec - 25 connexions par seconde sinon blocage 30 secondes.
root@mohamed-VirtualBox:/home/alain# cat /etc/xinetd.conf
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances          = 60
    log_type           = SYSLOG authpriv
    log_on_success     = HOST PID
    log_on_failure     = HOST
    cps                = 25 30
}

includedir /etc/xinetd.d
```

CONFIGURATION GLOBALE

/etc/xinetd.conf

- Ces paramètres de configurations vont influencer l'ensemble des services gérées par **xinetd**.
- **Les attributs du fichier de configuration générale /etc/xinetd.conf :**
 - I. **instances** : nombre maximal de requête qu'un service peut gérer à un moment donné.
 - I. **log_type** : localisation journalisation.
 - I. **log_on_success** : donnée de journalisation si la connexion est établie avec le service.
 - I. **log_on_failure** : donnée de journalisation si la connexion a échoué.
 - I. **cps** : Limite le nombre de connexion par seconde pour chacun des services gérés par xinetd et le retire pendant une période définie.
 - I. **includedir /etc/xinetd.d/** : Inclut des options stipulées dans les fichiers de configuration spécifiques aux services qui se trouvent dans le répertoire /etc/xinetd.d/

6-3 LE FONCTIONNEMENT DES SYSTÈMES INETD (XINETD)

```
# Fichier de configuration pour le service "servers"
#(service interne : fonctionnement du Daemon xinetd)
root@mohamed-VirtualBox:/home/alain# cat /etc/xinetd.d/servers
# default: off
# description: An internal xinetd service, listing active servers.

service servers
{
    type      = INTERNAL UNLISTED
    port      = 9099
    socket_type = stream
    protocol   = tcp
    wait       = no
    disable    = yes
    only_from  = 127.0.0.1
}
```

CONFIGURATION D'UN SERVICE

/etc/xinetd.d/nom du service

- Ces paramètres de configurations vont impacter uniquement le service concerné.
- **Les attributs du fichier (non exhaustifs) de configuration du service /etc/xinetd.d/ :**

Attribut	Définition
socket-type	Type de socket utilisé pour le service : dgram s'il utilise le protocole UDP, stream s'il utilise le protocole TCP - consulter le fichier /etc/services pour avoir l'information.
user	Identité sous laquelle le service sera lancé
server	chemin et nom du serveur
wait	Définit le comportement du service dans le traitement des threads : yes pour un service mono-thread (une connexion simultanée par service et une seule), no pour un service multithread (possibilité d'avoir plusieurs connexions simultanées au service)
protocol	Protocole utilisé par le service. Si rien n'est précisé, c'est le protocole spécifié dans le fichier /etc/services qui sera utilisé.
rpc_version rpc_number	Ne concerne que les services basés sur les RPC (exemple : NFS)
port	Port associé au service. Là encore, s'il n'est pas précisé, c'est le port spécifié pour le service dans le fichier /etc/services.

6-3 LE FONCTIONNEMENT DES SYSTÈMES INETD (XINETD)

```
root@mohamed-VirtualBox:/home/alain# cat /etc/xinetd.d/servers
# default: off
# description: An internal xinetd service, listing active servers.

service servers
{
    type      = INTERNAL UNLISTED
    port      = 9099
    socket_type = stream
    access_times = 09:45-16:15
    nice      = -19
    protocol   = tcp
    no_access  = 10.0.1.0/24
    wait      = no
    disable    = yes
    only_from  = 127.0.0.1
}
```

PARAMETRE SUPPLEMENTAIRE DE SECURITE POUR UN SERVICE

Limiter les attaques Deny of Service

Action	nom paramètre	Définition
Contrôle de la charge CPU	<i>rlimit_cpu = seconds.</i>	Cet attribut vous permet de limiter le temps CPU utilisé par un ou plusieurs services
Priorité du processus	<i>nice = level</i>	l'attribut permet de fixer une priorité d'ordonnancement pour le serveur. Le level peut prendre les valeurs de -20 (le plus prioritaire) à 19 (le moins prioritaire).
Limite nbre connexion par service	<i>instances = value</i>	L'attribut détermine le nombre d'instances simultanées du serveur qui seront autorisées. Préciser un nombre
Limite nombre de connexion avec la même origine	<i>per_source = value.</i>	Non seulement vous pouvez filtrer les adresses IP clientes, le nombre d'instances du serveur mais vous pouvez aussi limiter le nombre de connexions à un serveur donné provenant d'une même adresse IP
Ip blacklist	<i>Ex :192.168.0.12 flags = SENSOR deny_time = minutes</i>	Il est possible blacklister des adresses IP qui tenteraient des connexions sur des services
Période d'accès	<i>access_time</i>	On pourra choisir le moment auquel vous autoriserez les accès à tout ou partie de vos services réseaux.
Limiter l'accès à certaines	<i>only_from</i>	On va aussi pouvoir filtrer les clients qui vont pouvoir ou non se connecter à vos

6-4 LES WRAPPERS

- **TCP-Wrapper** est un outil de sécurité réseau qui permet de contrôler les accès, les tentatives de connexion sur une machine donnée.
- Il permet à tout instant :
 - De filtrer les accès.
 - De tracer (journalisation syslog) les connexions et tentatives de connexions à la machine.
- **Le Wrapper** va s'intercaler entre le super daemon xinetd et le serveur.
- Le daemon xinetd va passer pour le wrapper au lieu d'activer directement le service.
- Xinetd va lancer le daemon de Wrapper (**TCPD**) qui va se charger des contrôles et de vérifier les mécanismes de contrôle mis en place.



6-4 LES WRAPPERS

- On va pouvoir utiliser 2 fichiers: **/etc/hosts.allow et /etc/hosts.deny** pour filtrer les accès à sa machine.
 - **/etc/hosts.deny**: on indique dans ce fichier les services et les hôtes pour lesquels l'accès est interdit.
 - **/etc/hosts.allow**: on indique dans ce fichier les services et les hôtes pour lesquels l'accès est autorisé.
- Principe de fonctionnement : « *si c'est autorisé, c'est autorisé. Si c'est interdit, c'est interdit. Si ce n'est pas interdit, c'est autorisé* ».
- La stratégie la plus sûre : « *Interdire tout et autorisé explicitement (relation service / clients)* ».



6-4 LES WRAPPERS

```
# /etc/hosts.allow
vsftpd: 192.168.1.
in.telnetd, portmap: poste1, poste2

# /etc/hosts.deny
ALL : .hacker.org except white-hacker.org
vsftpd,in.telnetd,portmap : ALL
dovecot : 192.168.0. EXCEPT 192.168.1.5
```

AUTORISATION / INTERDICTION TCP-WRAPPER

Syntaxe au sein des fichiers **/etc/hosts.allow** et **/etc/hosts.deny**

daemon_list : client_list [:options]

- **daemon_list** : liste des exécutables (PAS DES SERVICES) séparés par des virgules. Vous pouvez mettre ALL pour spécifier tous les services.
- **client_list** : clients autorisés ou interdits pour ce service. On peut spécifier l'adresse IP, le nom, le masque de réseau, le nom du réseau, etc.
- **La client_list admet une syntaxe particulière :**
 1. ALL : correspondance systématique.
 2. LOCAL : tous les hôtes dont le nom ne contient pas de point (poste1, poste2, etc.).
 3. UNKNOWN : hôtes dont le nom ne peut pas être résolu.
 4. KNOWN : hôtes dont le nom peut être résolu.
 5. PARANOID : hôtes dont le nom ne peut être résolu ou dont l'IP n'a pas de résolution inverse.
 6. EXCEPT : permet d'exclure certains hôtes.