

Introduction

Les outils suivants sont indispensables à connaître lorsque l'on utilise un système sous Linux. On ne peut ici donner toutes les options de ces commandes. N'oubliez donc pas que l'on peut avoir plus d'aide en tapant la commande suivie de `--help`, mais aussi `man` commande ainsi que la commande `info`. Exemple `netstat --help`, `man netstat` ou `info netstat`. Enfin souvenez vous que sous GNU/Linux, on ne peut pas utiliser indifféremment les majuscules et les minuscules (la commande `ping` existe, pas la commande `Ping`).

ping

Cette commande est normalement connue de tous. Elle existe dans tous les systèmes. Elle permet de vérifier si une machine distante répond. La syntaxe est des plus simple `ping -c 5 213.186.xx.xx` pour envoyer 5 pings à la machine dont l'adresse IP est 213.186.xx.xx

On peut aussi utiliser le nom de la machine, si celle-ci est renseignée dans votre fichier `Hosts` ou dans un serveur DNS. On peut par exemple utiliser `ping` pour vérifier si la connexion est toujours active ou pour la monter. Si vous ne placez pas l'option `-c 5` pour n'envoyer que 5 pings, la commande ne s'arrête pas. Utilisez alors `Ctrl C`. Il existe un autre outil plus complet, mais qui n'est pas installé par défaut : `hping`.

ifconfig

`ifconfig` permet de connaître la configuration de vos cartes réseau, mais aussi de changer celle-ci. Pour changer la configuration de votre carte réseau, vous devez taper :

```
ifconfig eth0
```

```
213.186.xx.xx netmask 255.255.255.0 broadcast 213.186.xx.255
```

Comme les valeurs que je viens de donner sont standards, vous pouviez simplement taper `ifconfig ETH0 213.186.xx.xx` (le `netmask` et `broadcast` proposés sont ceux correspondant à une adresse de classe C).

Attention au redémarrage de la machine ce changement sera perdu. Il vous faut donc modifier en même temps le fichier `/etc/sysconfig/network-script/ifcfg-eth0`. Vous pouvez utiliser `linuxconf` pour faire plus simplement le même travail. On peut aussi désactiver une carte réseau `ifconfig eth0 down` et bien sûr la réactiver `ifconfig eth0 up`

arp

La commande `arp` permet de mettre en correspondance des adresses IP et les adresses MAC. Les options possibles importantes sont :

```
arp -a pour avoir toutes les entrées ARP de la table,
```

```
arp -d nom_de_la_machine pour supprimer une entrée de la table,
```

```
arp -s nom_de_la_machine adresses_mac pour ajouter une nouvelle entrée dans la table.
```

route

Cette commande permet de voir, d'ajouter ou d'enlever les routes se trouvant déclarées sur votre machine.

Ainsi, pour indiquer à votre machine où aller trouver les adresses qui ne sont pas les adresses de votre réseau local, vous devez lui indiquer la passerelle (ou gateway) vers laquelle elle doit envoyer tous les paquets.

Pour voir les routes indiquer `route -n` (on peut aussi utiliser `netstat -nr`). L'option `-n` permet de ne pas avoir la résolution des noms. Pour ajouter une route par défaut : `route add default gateway 192.168.0.1` (La passerelle vers qui j'envoie tous les paquets qui ne sont pas pour le réseau local). Pour détruire cette route `route del default` Pour ajouter une route vers une machine indiquer `route add -host 195.98.246.28 gateway 192.168.0.1` (Indiquer le `netmask` si celui-ci n'est pas un mask correspondant à la classe de votre adresse). Pour ajouter une

route vers un réseau indiquer route add -net 195.98.246.0 netmask 255.255.0.0 gateway 192.168.0.1. Enfin pour supprimer une de ces routes remplacer add par del. La gateway ou passerelle correspond la plupart du temps à votre routeur. Pour avoir la route que vous venez d'ajouter à chaque démarrage placer la commande dans le fichier /etc/rc.d/rc.local par exemple.

netstat

Voilà une commande moins connue et pourtant très utile. Je ne peux ici commenter toutes les options, je vous propose de lire le man netstat. Elle permet en effet de connaître les ports en écoute sur votre machine, sur quelles interfaces, avec quels protocoles de transport (TCP ou UDP), les connexions actives et de connaître les routes. Pour voir les connexions actives netstat -nt, pour les ports ouverts netstat -ntl. On peut aussi vérifier s'il existe une route par défaut, par exemple existe-t-il une route par défaut vers la machine 213.186.xx.xx utilisez alors netstat -nr | grep 213.186.xx.xx

L'option -a énumère les ports en cours d'utilisation ou ceux qui sont écoutés par le serveur. L'option -i donne des informations sur les interfaces réseau. L'option -p protocole donne beaucoup d'informations (paquets reçus, perdus, forwardés, taille options...) sur le trafic réseau dans le protocole donné Ex : netstat -p ip.

lsof

lsof permet de lister les fichiers ouverts et les processus actifs.

lsof -i indique les processus de type internet.

On peut ne demander que pour un protocole lsof -ni tcp:25 ou que vers une machine lsof -ni @213.186.xx.xx:25

Pour connaître tous les fichiers ouverts par sur /hda1 utiliser lsof /dev/hda1.

lsof -i -a -p 1234 permet de connaître tous les ports réseau ouverts par le processus 1234 (-a est interprété comme AND).

lsof -p 1234, 12345 -u 500, toto permet de connaître tous les fichiers ouverts par l'utilisateur 500 ou toto ou par le processus 1234 ou 12345.

Il existe des commandes pour faire cela (fuser, ps, netstat...), mais celle-ci est très complète.

traceroute

Traceroute permet de déterminer la route prise par un paquet pour atteindre la cible sur Internet. On peut utiliser soit l'adresse IP, soit le nom d'hôte. Attention certains FireWall ou routeurs ne se laissent pas voir avec la commande traceroute.

La commande traceroute est très utile pour savoir où peut se trouver un blocage (plutôt ralentissement). Il existe un grand nombre d'options, entre autre il est possible de choisir les gateway (jusqu'à 8) pour atteindre une machine. Je vous conseille donc encore une fois de lire le man traceroute.

telnet

Telnet est l'outil indispensable à connaître. Il existe en tant que client sur tous les systèmes et comme serveur sur les Unix. Attention il est de moins en moins installé par défaut sur les nouvelles distributions, dans sa version serveur. Dans sa version serveur il permet de donner un accès distant à la machine et donc un shell pour administrer celle-ci. Toutefois, et pour des raisons de sécurité, il n'y a plus de raison de l'utiliser, préférez-lui SSH, car alors les mots de passe ne se promènent pas en clair sur le réseau. Il existe des clients SSH pour Windows.

Par contre le client telnet permet de faire d'autres choses. En tant que client, telnet vous permet d'envoyer et de lire vos messages. Il permet aussi de tester les autres protocoles. Par exemple, on peut très bien faire un telnet Mon_serveur_ftp 21 pour se connecter sur un serveur FTP. Idem avec un serveur Web. telnet

Mon_serveur_web 80.

ftp

ftp est un outil qui permet de télécharger des fichiers entre machines. Vous connaissez les clients ftp comme ws_ftp.

Sous Linux il existe un serveur ftp, que vous activez dans /etc/inetd.conf. Il est installé par défaut dans toutes les distributions. Ce serveur ftp n'est pas lié à l'installation d'apache, comme pour les systèmes Microsoft où vous devez installer IIS pour bénéficier de ce service. Attention toutefois le serveur ftp pose un problème de sécurité important, utilisez plutôt SFTP, qui est disponible avec SSH.

Voici les commandes que vous allez utiliser le plus :

dir : pour lister un répertoire.

cd nom_du_répertoire : pour changer de répertoire.

get mon_fichier : pour copier un fichier vers votre client (obtenir). Il se place alors dans le répertoire où vous vous trouviez.

mget * : copier tous les fichiers du répertoire vers votre station.

put mon_fichier : pour copier un fichier vers le serveur.

mput * : pour copier les fichiers se trouvant dans votre répertoire.

binary : pour copier en mode binaire.

exit : pour quitter.

Il existe un grand nombre d'autres commandes. Mais vous avez là les principales, pour copier des fichiers entre machines. La commande ftp vous rendra un grand nombre de services, car elle permet assez simplement d'échanger des fichiers entre linux et windows, sans avoir à installer un client ftp ou à configurer samba.

nslookup

L'utilitaire nslookup permet d'interroger un serveur de nom (serveur dns) afin d'avoir des informations sur un domaine ou sur une machine. Par défaut nslookup utilise le serveur de nom configuré sur votre machine, vous pouvez toutefois interroger un autre serveur de nom.

```
[root@xxxxx /] nslookup
```

host

Commande presque équivalente, mais dont l'usage est plus simple.

host 213.186.xx.xx Donne de informations sur le domaine

host ?v ?t mx votre domaine Donne des informations sur les MX du domaine

host ?l ?t any votre domaine pour obtenir toutes les machines du domaine.

who

Cette commande permet de connaître les personnes qui sont loguées sur votre machine.

last

Cette commande vous permet de voir les dernières connexions ayant eu lieu sur votre machine (en fait il lit le fichier /var/log/wtmp).

last sans rien, vous affiche toutes les informations.

last david toutes les connexions de l'utilisateur david.

last reboot tous les reboot de la machine avec la date.

lastb est une variante de last, dans la mesure ou il ne cherche que les mauvais login (il lit le fichier /var/log/btmp)

finger

finger est un service qui vous permet d'obtenir des informations sur les comptes utilisateurs de votre machine.

Ce service est à proscrire. Sa syntaxe est toutefois assez simple finger toto@la_machine_distante

Pour avoir plus d'informations utiliser l'option -l

Cet exemple montre les deux connexions ouvertes sur la machine.

```
[root@xxxxx] finger -l
```

Netcat

Outil permettant de créer des connexions (socket) entre machines. Il se comporte comme un client netcat mon_serveur.fr 200 (connexion sur le port 23 de la machine mon_serveur.fr) ou comme serveur netcat -l -p 80 (il écoute sur le port 80). Il permet aussi de faire du scan de ports.

Voici quelques exemples :

netcat -t ns213.186.xx.xx 23 se comporte comme un client telnet

netcat -l -p 23 > espionne.log écoute sur le port 23 (telnet) et enregistre dans espionne.log tout ce qui est tapé par le client.

netcat -l -p 23 < mes_commandes exécute les commandes qui sont dans mes_commandes.

netcat -l -p 23 -e ma_commande exécute la commande après connexion.

netcat -vv la_machine_a_scanner 1-100 permet de lancer un scan sur des machines distantes.

netcat -vv -z -i 10000 -r 127.0.0.1 1-200 permet de scanner aléatoirement les ports de 1 à 100 avec un timeout. On évite la détection.