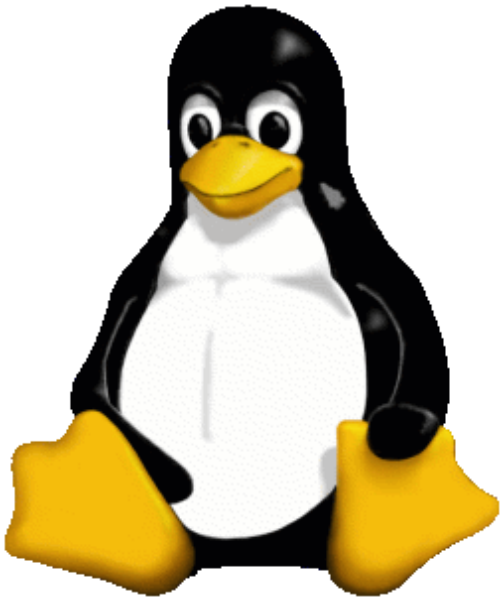
Accueil

COURS LINUX

Dernière mise à jour le 29/11/2006



Linus Torvalds, étudiant à l'université d'Hélsinki, travaille sur un projet du nom de Minix permettant d'exploiter au mieux les fonctionnalités multi-tâches du processeur i386. Ce système s'avérant trop limité, Linus Torvalds décida de développer un système d'exploitation. Ce système vu le jour en août 1991 sous le nom de Linux. Linus Torvald est le propriétaire de Linux mais le code source est gratuitement distribué selon les termes du GPL (General Public Licence). Cela signifie que tout le monde peut utiliser et même commercialiser ce système à la condition de rendre à leur tour disponible toutes les modifications apportées. De ce fait Linux appartient au monde du "logiciel libre".

Dans ce document nous allons voir comment administrer un système linux en privilégiant la manipulation des fichiers système. Cette méthode est certes primaire et nécessite une bonne maîtrise de la syntaxe de chacun des fichiers mais a le mérite d'être tout à fait exportable. Les fichiers ainsi créés pourront être utilisés sur beaucoup de distributions différentes. Par conséquent nous n'allons pas décrire l'utilisation d'outils d'administration tels que linuxconf ou webmin.

La distribution utilisée pour écrire ce document est la Debian Sarge (3.1)

SOMMAIRE

GENERALITES :

Linux est un système d’exploitation :

- Réseau,
- Multi-tâches préemptif,

- Multi-utilisateurs,
- Multi-plateformes (ALPHA, MIPS, INTEL, POWERPC, MAC,…),
- Gérant le SMP, le Clustering, le RAID,
- Capable de gérer beaucoup de systèmes de fichiers (FAT, FAT32, …)
- Possède un système de fichier très performant (ext2, ext3). Il n'y a pas de fragmentation.

Pourquoi utiliser Linux :

- Linux est un système d'exploitation bon marché, on peut même se le procurer librement (gratuitement) en téléchargement sur des sites ftp.
- C'est un système d'exploitation très stable (grande cohérence du noyau), une seule personne est à l'origine du projet.
- On peut apporter facilement des améliorations (correctifs) parce que le code source est librement distribué.
- A noter, qu'il y a très peu de virus connu sur Linux.
- Beaucoup d'applications sont développées sur cette plateforme.

Les logiciels libres : FSF, GNU, GPL :

La FSF (Free Software Fondation) a été fondée aux débuts des années 80 par Richard Stallman. Le but de cette fondation est de développer des logiciels libres (copie, utilisation, modification et redistribution libre à condition de rendre disponible les sources). Le projet GNU est un projet émanant de la FSF et dont le but est de développer un système d'exploitation. Ce système reprend un certain nombre de concept UNIX mais GNU signifie GNU's not Unix ce qui veut dire que Linux n'est pas Unix. Ce système est appelé HURD. Linux est aussi appelé GNU/Linux puisque qu'il respecte les termes de la GPL. GPL signifie General Public Licence. C'est en fait une licence qui spécifie les conditions de distribution de tous les logiciels GNU.

[Retour au menu]

LES DIFFERENTES DISTRIBUTIONS :

Une distribution se présente sous forme de fichiers téléchargeables sur des sites ftp ou sous forme d'un ou plusieurs cd-rom. Une distribution est un ensemble composé du noyau, d'applications et de documentations. Il existe beaucoup de distributions sur le marché, en voici quelques une commentées :

RedHat

Les programmes installés sont le plus souvent directement utilisables, elle convient donc parfaitement aux débutants qui bénéficient également d'interfaces graphiques pour la configuration et l'administration (linuxconf). Les utilisateurs avertis trouvent également leur compte dans la cohérence, la qualité et le dynamisme de cette distribution. L'apport principal de RedHat est le concept de paquetage (.rpm) qui comprend le logiciel ainsi que tout les utilitaires permettant sa configuration, son installation, sa désinstallation ainsi que sa mise à jour.

A ce jour version :

- Redhat Linux 9.0, Fedora Core 6
- Redhat Linux WS, ES, AS version 3

Debian

C'est la distribution d'un groupe de bénévoles. Ses adeptes reconnaissent l'esprit GNU qui anime depuis toujours le développement de Linux. Très complète et conçue de façon méticuleuse et efficace, la Debian permet de tout dimensionner selon ses besoins. D'importants intervalles de temps séparent cependant parfois deux mises à jour stables et sa prise en main est parfois délicate, surtout pour les débutants. Mais une fois passés les premiers obstacles, on ne veut paraît-il plus en changer. Debian possède son propre format de paquetage (.deb).

A ce jour version :

- Stable : 3.1r4 appelée Sarge
- Testing : appelée Etch

Mandrake / Mandriva

Distribution française très bien finie et bien francisée. Tout est fait pour le confort de l'utilisateur final. La version GPL tient sur un CD. La version commerciale en comprend cinq et représente 2500 packages. Son installation reprend celle de RedHat. Les paquetages sont des .rpm. Elle est à recommander à tous ceux qui veulent utiliser leur machine rapidement sans passer trop de temps à jouer le rôle de l'ingénieur système.

A ce jour version 2006

Slackware

Destinée à ceux qui souhaitent acquérir en douceur une bonne maîtrise d'Unix, qui veulent mettre en place un serveur, et ceux pour qui la facilité de déploiement est particulièrement importante. Elle compte encore de très nombreux adeptes parfois convaincus mais semble en perte de vitesse et les mises à jour se font rares. De plus sa conception ancienne lui confère quelques défauts.

A ce jour version 10.2

Il existe beaucoup d'autres distributions que nous n'allons pas détailler ici, on peut en citer quelques unes : Ubuntu, Suse, Corel Linux, Caldera, Gentoo, Connectiva, Yellow Dog, ...

Il existe aussi des distributions dites "Clés en main" qui propose en général une interface graphique accessible par http ou https et permettant de configurer les services les plus couramment utilisés. Les services sont souvent pré-configurés : les plus connues sont Ipcop, Smoothwall, SecurePoint, Mandrake MNF, E-smith SME Server, Free-EOS, Clarkconnect, Engarde Secure Linux, ...Je vous renvoie sur les sites de ces distributions pour en savoir davantage.

D'autres distributions, appelées "Linux Live", permettent d'utiliser Linux sans installation sur le disque dur. Elles se présentent sous forme de CD ou DVD bootable. Très pratique, elles permettent de se faire la main sur Linux sans pour autant installer le système sur sa machine. Certaines offrent la possibilité de réaliser des sauvegardes sur clé USB ce qui permet de retrouver ses fichiers et son environnement par la suite. Les plus connues sont : Knoppix (et ses dérivés nombreux), Mandrake Move, Mandows, GeeXBoX, Damn Small Linux, MoviX, …

[Retour au menu]

LE NOYAU

Le noyau joue le rôle d'intermédiaire entre les programmes et le matériel. Il gère la mémoire pour tous les programmes en cours d'exécution (processus), et s'assure qu'ils occupent tous une part équitable (ou non) du temps processeur. En plus, il fournit une interface aux programmes pour communiquer avec votre matériel.

La numérotation des noyaux Linux :

- La numérotation du noyau Linux est basée sur trois nombres, par exemple : 2.0.12.
- Le premier nombre (dans notre cas le "2") indique la version majeure du noyau. Aujourd'hui, le noyau en est à sa deuxième version.
- Le deuxième nombre peut être considéré comme un numéro de version mineure. Attention, les versions mineures impaires indiquent une version de développement. Actuellement, la version stable est la 2.6.
- Le dernier numéro indique les évolutions mineures. Dans un noyau stable, il s'agit souvent de corrections ; dans un noyau instable, il peut s'agir de nouvelles fonctionnalités.

A retenir : si vous n'êtes pas un bidouilleur, prenez une version stable (de préférence la dernière).

Astuce : le numéro de version de votre noyau est obtenu en tapant uname -a

Changement de noyau :

- La compilation d'un nouveau noyau est une opération délicate qui doit être réalisée uniquement si vous savez ce que vous faites.
- Votre machine est un serveur et fonctionne correctement depuis des années : ne changez que si vous voulez disposer de nouvelles fonctionnalités ou si des failles de sécurité ont été découvertes.
- Vous n'êtes pas un bidouilleur et vous installez régulièrement une nouvelle distribution : patientez jusqu'à la sortie d'une nouvelle distribution qui proposera une version de noyau plus récente.

[Retour au menu]

ARBORESCENCE ET SYSTEME DE FICHIERS

/bin : répertoire contenant les commandes UNIX pour tous les utilisateurs. Equivalent à /usr/bin.

/boot : contient le noyau du système et les fichiers nécessaires à l'amorçage de la machine.
/boot/grub/ : emplacement des fichiers utiliser par l'utilitaire « grub ».

/dev : contient les fichiers périphériques
fd : floppy disk (disquette). En général fd0
hd / sd : hard disk (disque dur)
La 1° partition du 1° disque dur IDE sera hda1
La 3° partition du 2° disque dur IDE sera hdb3
La 2° partition du 1° disque dur SCSI sera sda2

Les numéros de 1 à 4 correspondent aux partitions primaires (physiques)
A partir de 5 c'est des partitions étendues (logiques)
cdrom : pour le lecteur de cd-rom

/etc : contient les fichiers de données pour l'administration et la configuration du système.
/rc2.d/ répertoire contenant les scripts de démarrage pour le runlevel 2
/init.d/ : répertoire contenant les scripts de démarrage et d'arrêt des services.
/network/ : répertoire contenant les fichiers de configuration du réseau.

/home : contient les répertoires et fichiers utilisateurs.

/mnt : répertoire proposé pour le montage des systèmes de fichiers (CD-ROM, Floppy …).

/opt : paquetages d'applications logicielles supplémentaires.

/proc : répertoire utilisé par le système pour mémoriser les processus entre autre.

/root : répertoire de connexion de l'utilisateur "root".

/sbin : contient les commandes pour l'administration du système. Equivalent à /usr/sbin.

/tmp : répertoire utilisé par des commandes pour créer des fichiers de travail. Leur destruction n'est pas automatique mais ils peuvent être supprimés à n'importe quel moment.

/usr : répertoire contenant les programmes et les données importantes mais non vitaux au démarrage du système (manuel en ligne par exemple).

/var : contient les fichiers des services.
cron/ : commandes du service cron
lib/ : répertoire utilisé par les services pour stocker les données courantes.
/var/lib/dhcpd contient les baux du service dhcp
log/ : répertoire contenant les fichiers de log
messages ou **syslog** : fichiers log des principaux messages du système. Associé au démon syslogd.
secure : fichier log des tentatives de connexions infructueuses.
maillog : fichier log du service de messagerie.
cron : fichier log du service cron
wtmp : fichier log en binaire des commandes init et login. Ce fichier est exploité par la commande last.
apache/* : répertoire contenant les fichiers log du service web.
samba/* : répertoire contenant les fichiers log du service samba.
lastlog : fichier log des dernières connexions.
dmesg : fichier contenant les messages affichés au démarrage.
mail/ : stockage des mails utilisateurs non lus
run/ : contient les PID des services actifs et le fichier utmp des utilisateurs connectés.
spool/ : contient les fichiers de données des services.
www/ : fichiers du service Web (apache).

Attributs d'un fichier :

La commande ls –l ou ll permet d'afficher les attributs pour chaque fichier ou répertoire.

-	- - -	- - -	- - -
d	r w x	r w x	r w x
directory	user	group	other
r : lecture w : écriture x : exécution			

Exemple : **chmod 600 file** 600 signifie 110 000 000 donc rw- - - - - - -

Détail des permissions pour le fichier file:
Utilisateur propriétaire : lecture écriture mais il n'a pas le droit d'exécuter ce fichier
Groupe propriétaire : aucun droit
Autres : aucun droit

Pour donner les droits de lecture, écriture et exécution à tout le monde : **chmod 777 file**

Autre exemple :

Permissions		Utilisateur propriétaire	Groupe propriétaire		Date	Nom du fichier ou du répertoire
drwxr-xr-x	5	root	root	4096	mar 29 14:47	Desktop
-rwx--x--x	1	root	root	166	mar 6 07:53	script

Desktop est un répertoire - script est un fichier

[Retour au menu]

COMMANDES UNIX

adduser, **userdel** : ajoute / supprime un compte utilisateur.
adduser –G eleves –u uid –d /home/toto toto
-G : ajoute toto au groupe eleves
-u : permet de fixer l'UID de l'utilisateur
-d : fixe l'emplacement du répertoire personnel de l'utilisateur
userdel –r toto
-r : supprime les données du compte toto (répertoire personnel)

awk, **gawk** : utilitaire permettant de faire du traitement de chaîne à partir d'un fichier et d'exécuter un bloc d'instruction.
awk –F":" 'bloc-programme' fichier
-F":" : permet de spécifier le caractère de séparation
Exemples :
awk -F":" '{ print $1 }' /etc/passwd : affiche les noms de login inscrit dans le fichier des utilisateurs

bzip2 / **bunzip2** : compresse / décompresse au format bz2

cal : affiche le calendrier du mois (cal) ou de l'année (cal 2002).

cd : permet de se déplacer dans l'arborescence.
cd /root : pour se déplacer dans le répertoire root depuis la racine
cd / : pour revenir à la racine
cd .. : pour descendre d'un niveau
cd - : pour retourner au répertoire précédent
cd ~ : retour à la maison ;-)

cfdisk : utilitaire de partitionnement de disque
cfdisk /dev/hdb, cfdisk /dev/sda

chkconfig :
permet d'automatiser le lancement des services au démarrage. Cette commande crée des liens symboliques dans les différents répertoires /etc/rc.d/rc(n) en configurant les scripts situés dans /etc/rc.d/init.d/. cf. commande ntsysv.
chkconfig --list crond : permet de savoir si le service crond est lancé au démarrage et sur quel runlevel
chkconfig --add/--del smb : ajoute/retire le lancement automatique du service samba
chkconfig --level 345 sendmail on : ajoute le lancement automatique de sendmail au démarrage sur les runlevel 3,4,5
chkconfig --level 012 sendmail off : retire le lancement automatique de sendmail au démarrage sur les runlevel 0,1,2

chmod : change les droits sur les répertoires et les fichiers.
chmod 744 file le fichier file sera :
en accès complet pour l'utilisateur propriétaire,
en lecture seule pour le groupe propriétaire,
en lecture seule pour le reste des groupes et utilisateurs.
chmod –R 777 rep :
Le répertoire rep ainsi que tous les sous-répertoires et fichiers seront en accès complet.

chown : change le propriétaire des fichiers et des répertoires.
chown root file le propriétaire du fichier file sera root

chgrp : change le groupe propriétaire des fichiers et des répertoires.
chgrp root file le groupe propriétaire du fichier file sera le groupe root

clear : efface l'écran.

cp : permet de copier des fichiers.
cp /root/file /tmp copie le fichier file dans le répertoire tmp
cp –r rep copie le répertoire rep

cut : sélection d'une colonne ou d'une chaîne de caractère séparé par un motif.
cut –c5 /etc/passwd affiche le 5° caractère de chaque ligne contenue dans le fichier spécifié
cut –d":" –f1,5 /etc/passwd affiche le 1° et le 5° champ séparé par le caractère : dans le fichier spécifié

date : gestion de la date et de l'heure.
date : affiche la date et l'heure
date [MMJJHHmmAA] : règle la date et l'heure
MM : Mois
JJ : Jours
HH : Heure
mm : minute
AA : Année
date 1106121402 : fixe la date au 6 novembre 2002 et l'heure à 12h14

dmesg : permet d'afficher le fichier /var/log/dmesg contenant l'ensemble des messages affichés au démarrage.

du : affiche la taille de tous les répertoires et sous-répertoires du répertoire courant.
du –s affiche la taille de tous les répertoires et sous-répertoires du répertoire courant.
du –a affiche la taille de tous les répertoires et sous-répertoires du répertoire courant en donnant des informations sur les fichiers.

df : affiche des informations sur les disques (espace libre, partitions montées).
df –T : affiche les systèmes de fichiers
df –i : affiche l'utilisation des inodes
df –h : affiche les informations en utilisant les multiples

exit : pour quitter les droits du super-utilisateur. Permet aussi de sortir de certains services comme ftp.

file : renseigne sur la nature d'un fichier.

finger : permet d'avoir des renseignements sur un utilisateur.

free : affiche les quantités de mémoires libres et utilisées.

fuser : identifie les activités en cours sur un disque
fuser –u /dev/hda2

ftp : ftp est un outil qui permet de télécharger des fichiers entre machine.
ftp <serveur> permet de se connecter en ftp sur la machine. Il faut ensuite s'identifier en tant qu'utilisateur connu

grep : permet de rechercher une chaîne de caractères dans un fichier.
grep [options] motif [fichier]
grep ftp /etc/services : recherche le mot ftp dans le fichier spécifié
more /etc/services | grep ftp : Cette commande aboutie au même résultat (utilisation d'un tube)
[options] :
-c : compte le nombre de ligne
-l : donne le nom des fichiers
-v : donne les lignes ne correspondant pas au critère
-i : permet de ne pas tenir compte de la casse
-n : affiche le numéro des lignes
-w : impose que le motif corresponde à un mot entier

groupadd, **groupdel** : ajoute / supprime un groupe.

groups : affiche les groupes auxquels appartient un utilisateur.
groups toto affiche les groupes auquel appartient l'utilisateur toto

gzip / **gunzip** : compresse / décompresse au format gz.

halt : pour arrêter le système. Equivalent à shutdown -h

id : affiche le n° utilisateur (uid), le n° de groupe (gid) et les groupes auxquels l'utilisateur appartient.

ifconfig : permet de configurer les interfaces réseaux
ifconfig : liste l'ensemble des interfaces réseaux et affiche les informations sur celles-ci
ifconfig eth0 : affiche des informations sur eth0
ifconfig eth0 up/down : active/désactive l'interface eth0
ifconfig eth0 192.168.1.1 : attribue l'adresse spécifiée à eth0. Par défaut le netmask et le broadcast affectés seront ceux de la classe correspondante
! En cas de redémarrage du service réseau ou de la machine, les changements sont perdus.

last : permet de connaître les dernières connexions sur la machine.

locate : permet de chercher un fichier ou un répertoire.
La base de données doit auparavant être mise à jour. Cela se fait par la commande **updatedb**

logout : pour se déconnecter (raccourci : "Ctrl + d")

ln : permet de créer des liens entre fichiers
ln –s /dev/ttyS1 /dev/modem

ls : permet de lister l'ensemble des objets d'un répertoire.
ls –l : affiche les liens des fichiers. Equivalent à la commande ll. Permet aussi l'affichage des droits
ls –a : affiche tous les types de fichiers y compris les fichiers cachés.
ls –i : affiche les inodes.

kill : permet de tuer un processus
kill numero_pid tue le processus correspondant.
kill –9 numero_pid opération plus radicale.
killall nom_processus tue tous les processus portant ce nom

mc : Midnight Commander. Utilitaire de gestion de fichiers identique à Norton Commander ou Pctools.

mkbootdisk : permet de créer une disquette de démarrage
Se placer dans le répertoire /lib/modules puis taper la commande suivante :
mkbootdisk –device /dev/fd0 "n° du noyau" (sans les " ")

mkdir / **rmdir** : permet de créer/supprimer un répertoire.

mkfs : permet de formater en choisissant un système de fichiers.
mkfs –t fstype /dev/sdb
Par défaut le fstype est ext2.
Les différents système de fichiers pris en charge sont : Minix, ext, ext2, msdos, hpfs, iso9660, nfs, ntfs, smbfs, swap, vfat.

mk2fs : formate un périphérique au format ext2.
mk2fs /dev/fd0

more : permet d'afficher page par page le contenu d'un fichier.

mount, **umount** : monte / démonte un système de fichiers.
mount : liste tous les système de fichiers actuellement montés
mount –a : monte tous les systèmes au démarrage
mount /dev/fd0 /mnt/floppy : monte la disquette
mount /dev/cdrom /mnt/cdrom : monte le cd-rom
mount /dev/hdb1 /mnt/windows : monte une partition
mount –t vfat /dev/hda2 /mnt/disque1 : monte une partition en indiquant le système de fichier
umount /mnt/floppy : démonte le système de fichiers attaché à la disquette

mv : pour déplacer un fichier

netstat : commande réseau multiple.
netstat –nr : affiche la table de routage. Identique à route –n
netstat –nt : affiche les connexions actives
netstat –ntl : affiche les ports ouverts par les différents services
netstat –a : affiche les ports ouverts ou ceux écoutés par le serveur
netstat –i : identique à ifconfig

pico / **nano** : Editeur de fichier.

ping : permet de vérifier si une machine distante répond. Utile pour vérifier s'il existe un lien physique entre 2 machines.
ping 192.168.1.253
ping –c 4 192.168.1.253

ps : affiche la liste des processus.
ps : affiche la liste des processus utilisateur en cours.
ps a : affiche la liste complète des processus en cours.
ps u : affiche la liste des processus en cours en donnant leur appartenance utilisateur.
ps x : affiche la liste des processus en cours en prenant en compte ceux ne dépendant d'aucun terminal.
ps aux |grep squid affiche tous les processus contenant la chaîne de caractère squid. Cela permet de savoir si le programme squid tourne et surtout connaitre son pid.

passwd : permet de changer le mot de passe d'un utilisateur.
passwd : change le mot de passe de l'utilisateur courant
passwd toto : change le mot de passe de l'utilisateur toto
passwd –d toto : supprime le mot de passe de l'utilisateur toto
passwd –l toto : verrouille le compte toto
passwd –u toto : déverrouille le compte toto

passwd --stdin toto : le système attend le mot de passe sur l'entrée standard. Très utile dans les scripts couplé avec un tube : echo password | passwd –stdin utilisateur

pwd : indique le path.

reboot : permet de redémarrer le système.

rm : permet de supprimer des fichiers et des répertoires.
rm /root/file : supprime le fichier file se trouvant dans le répertoire rep
rm –d /home/rep : supprime le répertoire rep
rm –df rep pour forcer la suppression du répertoire toto
rm –rf rep supprime le répertoire rep même s'il n'est pas vide

route : affiche, ajoute ou enlève une route
route ou route –n : affiche les routes
route add -net 192.168.0.0 netmask 255.255.255.0 gw 172.16.0.1 : ajoute une route
route del -net 192.168.0.0 netmask 255.255.255.0 gw 172.16.0.1 : supprime une route
route add default gw 172.16.0.1 : ajoute une route par défaut
ou encore : route add -net 0.0.0.0 netmask 0.0.0.0 gw 172.16.0.1
route del default : supprime la route par défaut

ssh : permet de lancer une session ssh.
ssh 192.168.0.253
ssh www.bruno-simonet.net -l admin

startx : permet de lancer l'interface graphique.

su : permet de passer super-utilisateur c'est à dire prendre les droits de root.
A partir de n'importe quel utilisateur su permet de s'approprier les droits de root (il faut connaître bien sûr le password root). su est surtout utiliser pour l'administration à distance (telnet, ssh). En effet, il n'est pas possible, par défaut, de se loguer sous root par telnet. Il faut donc se loguer sous un nom d'utilisateur puis s'approprier les droits de root.

tar : Commande d'archivage.
tar x : pour extraire le contenu d'une archive
tar c : pour créer une archive
tar v : mode bavard
tar f : affiche le contenu d'une archive
tar z : compresse ou décompresse en utilisant gzip
tar y : compresse ou décompresse en utilisant bgzip2
tar xvf fichier.tar
tar zxvf fichier.tar.gz
tar zcvf fichier.tar.gz repertoire/

tcpdump : outils d'observation réseaux.
tcpdump : lance la capture (ici, tout est capturé !)
src : spécifie la source
dst : spécifie la destination
host : spécifie un hôte
port : spécifie le port
udp/tcp : spécifie le protocole de la couche transport
Exemple :
tcpdump src host 192.168.0.2 and dst host 172.17.1.251 and port 53 and udp
tcpdump -x -X -s 0 dst host 192.168.0.1 and port 110 and tcp
tcpdump -i eth1 port 520 -v

telnet : permet l'administration à distance d'une machine. Attention, toutes les informations passent en clair. Pour plus de sécurité utilisez ssh.
telnet 192.168.0.252 : connexion au service telnet (port 23)
telnet 192.168.0.252 110 : connexion au service pop3 (port 110). Permet de lire ses mails en ligne de commande.

traceroute : permet de déterminer la route prise par un paquet pour atteindre la cible. traceroute @IP ou nom d'hôte.

tty : affiche le numéro de la console.

type : pour savoir si un programme est installé et où il se trouve.
type vi réponse si vi est installé vi is /bin/vi

uname : affiche des informations sur le système et la machine.
uname –a : affiche toutes les informations disponibles par cette commande

usermod : modifie les propriétés d'un compte utilisateur.

vi : éditeur de fichier. Un exemple d'utilisation est vu dans la configuration de la crontab.
vipw : visualisation et édition du fichier /etc/passwd
vigr : même chose pour le fichier /etc/group

wc : affiche le nombre d'octets, de mots et de lignes d'un fichier
-c : octet
-w : mot
-l : ligne

wget : permet de télécharger un fichier en ligne de commande sans lancer de session ftp.
wget ftp://192.168.1.230/partage/debian-31r1a-i386-netinst.iso

whereis, **which** : permet de chercher l'emplacement d'une commande.

who : permet de connaître les utilisateurs d'une session.
who –u pour connaître les utilisateurs ayant ouvert une session
who am i pour afficher les renseignements en rapport avec sa propre session

[Retour au menu]

PRINCIPAUX FICHIERS DE CONFIGURATION

/etc/fstab : table de montage et système de fichiers.

Format de la table :
/dev/hda5 / ext2 defaults 1 1
/dev/hda6 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,ro 0 0

/dev/cdrom : fichier périphérique
/mnt/cdrom : point de montage
iso9660 : type du système de fichier
noauto,ro : options de montage
0 0 : le premier chiffre est utile à la commande dump, le deuxième à fsck

/etc/hosts : table de correspondance entre nom et @ IP.

/etc/hosts.allow et **/etc/hosts.deny** : fichiers utilisés par le wrapper tcpd.

/etc/host.conf : défini l'ordre de priorité pour la recherche de nom
order hosts,bind
multi on lecture du fichier hosts en 1°, puis requête DNS

/etc/hostname : fichier de définition du nom de la machine.

/etc/inetd.conf : fichier de configuration du super démon réseau inetd.

/etc/init.d/ : répertoire des scripts de démarrage et d'arrêt des services.

/etc/inittab : configuration du niveau de démarrage.

/etc/issue, **/etc/issue.net** : texte affiché avant une connexion locale / distante.

/etc/lilo.conf : fichier de configuration du boot (lilo : LInux LOader).

/etc/motd : texte affiché après une connexion.

/etc/network/ : répertoire de configuration des paramètres réseaux.
/etc/network/interfaces : fichier de configuration des interfaces réseaux.
auto eth0 eth1 -> montage des interfaces au démarrage
iface eth0 inet dhcp -> configuration de eth0 en dhcp
iface eth1 inet static -> configuration manuelle de l'interface eth0
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1

/etc/passwd - **/etc/group** - **/etc/shadow** : fichier de configuration des comptes utilisateurs et des groupes.

/etc/printcap : fichier de configuration des imprimantes.

/etc/resolv.conf : domaine de recherche et @ des serveurs DNS.

/etc/services : fichier utilisé par le super démon réseau xinetd / inetd. Ce fichier associe les différents services et les ports tcp,udp correspondants

/etc/xinetd.d/ : répertoire contenant les fichiers de configuration du super daemon réseau xinetd

[Retour au menu]

GESTION DES PACKAGES

dpkg : commande permettant la gestion des packages
-i | **--install** : installation d'un packages
dpkg –i dhcp3-server_3.0.1-2_i386.deb

-r | **--remove** : supprime un package
dpkg –r dhcp3-common_3.0.1-2_i386.deb
dpkg –r –purge dhcp3-common_3.0.1-2_i386.deb

-l | **--list** : liste les packages installés sur la machine

Logiciel "APT" :

/etc/apt/sources.list :
deb http://host/debian distribution section1 section2 section3
deb-src http://host/debian distribution section1 section2 section3

Exemple :
deb ftp://192.168.0.230/mirrors/debian sarge main
deb ftp://ftp.fr.debian.org/debian stable main contrib non-free
#deb-src ftp://ftp.fr.debian.org/debian stable main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

apt-get update : met à jour la liste des packages disponible sur les serveurs spécifiés dans le sources.list

apt-get install : télécharge et installe le ou les packages nécessaires à l'installation. Apt gère les dépendances.
apt-get install proftpd

apt-get remove –purge : supprime une application. L'option purge permet de supprimer les fichiers et en particulier les fichiers de configuration.
apt-get remove –purge samba
apt-get remove samba-common

apt-cache search : realise un recherche dans le cache (liste des packages) pour savoir si l'application recherchée existe sur les serveurs mirroirs.
apt-cache search vsftpd : recherche l'application vsftpd
apt-cache search ftp : recherché les application correspondent au motif passé en argument, ici, ftp. Cette commande permet d'afficher l'ensemble des applications ayant un rapport avec le protocole ftp.

[Retour au menu]

TCP_WRAPPERS et super-démon réseau INETD, XINETD

Un wrapper offre une couche de protection supplémentaire. Il contrôle l'accès aux services à l'aide de fichiers de configuration contenant des listes de contrôle d'accès. Par exemple, il est possible d'activer ce contrôle sur le service ftp (wu-ftpd); Lorsqu'un client se connecte au serveur ftp, le système vérifie d'abord s'il est autorisé à accéder au serveur puis lance le service demandé. Les fichiers consultés sont /etc/hosts.allow et /etc/hosts.deny. Si ces fichiers sont manquants ou vides, il n'y a pas de contrôle d'accès.
Exemple pour le fichier /etc/hosts.allow :
in.ftpd : .domain.com
ici tous les clients du domaine domain.com et seulement eux ont le droit d'utiliser le service ftp

L'administrateur a le choix d'utiliser ou non le wrapper. Prenons l'exemple du service ftp (wu-ftpd). Si vous souhaitez utiliser le contrôle d'accès par le wrapper il vous faudra configurer le super-démon réseau inetd.

INETD :

inetd utilise les fichiers /etc/services et /etc/inetd.conf
Le fichier **/etc/services** permet d'associer un port à une application (à un nom)
Le fichier **/etc/inetd.conf** est le fichier de configuration du super-démon. Il permet de lancer les services qui seront contrôlés par le wrapper. Il est possible dans ce fichier d'affiner le paramétrage des services.

Structure du fichier /etc/inetd.conf :
[nom] [service] [protocole] [état] [utilisateur] [chemin]
[nom] : nom du service tel qu'il est déclaré dans /etc/services (ftp, telnet, ...)
[type] : type de service de transport de données
stream pour tcp
dgram pour udp
raw pour IP
[protocole] : nom du protocole tel qu'il existe dans /etc/protocols
[état] : état d'attente peut prendre les valeurs wait et nowait
wait : il y aura un seul serveur pour l'ensemble des clients
nowait : il y aura un serveur par client
[utilisateur] : Nom de l'utilisateur sous lequel sera exécuté l'application
[chemin] : Chemin d'accès au programme lancé par inetd. Il est possible ici d'ajouter les options de démarrage du programme.

Extrait du fichier /etc/inetd.conf :
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
Ici les services ftp et telnet sont lancés par inetd et sont sous contrôle du wrapper.

XINETD :

Si vous décidez d'utiliser xinetd il faut arrêter le service inetd. On utilise soit l'un soit l'autre. xinetd est le remplacant de inetd et tcp_wrappers. Il permet d'apporter plus de possibilités de configuration.
Le service est configuré par le fichier /etc/xinetd.conf puis chacune des applications que l'on souhaite exécuter par xinetd est décrite par un fichier se trouvant dans le répertoire /etc/xinetd.d/.

Fichier **/etc/xinetd.conf** :

```
defaults
    {
    disable = yes    Tout est désactivé par défaut
    instances = 10
    per_source = 3    On n'autorise que 3 connexions en provenances de la même machine.
    log_type = SYSLOG authpriv
    log_on_success = HOST PID
    log_on_failure = HOST RECORD
    }
includedir /etc/xinetd.d
```

Voici l'exemple du fichier **wu-ftpd** :

```
service ftp
    {
    disable = no / yes    no pour activer le service, yes pour le désactiver
    flags = REUSE
    instances = UNLIMITED    Pas de limitation sur le nombre de requêtes possibles (à éviter).
    instances = 5    On n'autorise que 5 connexions simultanées.
    only_from = 192.168.0.0/24    On n'autorise la connexion que depuis le réseau 192.168.0.0
(masque:255.255.255.0).
    only_from = .domain.com    On n'autorise la connexion que depuis une machine du domaine domain.com
    only_from = 192.168.0.{1,2,3}    On n'autorise la connexion que depuis les machines 192.168.0.1,
192.168.0.2 et 192.168.0.3
    socket_type = stream    Type de service.
    wait = no / yes    Etat d'attente.
    user = root    Nom de l'utilisateur sous lequel le démon tourne.
    server = /usr/sbin/in.ftpd    Chemin d'accès au programme in.ftpd.
    server_args = -l -a    les options de démarrage du programme
    protocol = tcp    Type de protocole
    port = 21    Port associé au service
    log_on_success += DURATION USERID
    log_on_failure += DURATION USERID
    }
```

[Retour au menu]

SERVEUR PROXY-CACHE : SQUID

Fichier de configuration : **/etc/squid/squid.conf**
Fichiers de log : **/var/log/squid/access.log** | **cache.log** | **store.log**

Gestion du service: **/etc/init.d/squid start** | **stop** | **restart**

squid -z : Création du cache sur le disque dur
squid -k reconfigure : Relecture de fichier squid.conf. Cela permet de prendre en compte des modifications dans le fichier de configuration sans avoir à relancer squid.

Description du fichier de configuration squid.conf :

http_port 3128
#
cache_mem 40 MB
 # Taille mémoire allouée à SQUID. Il est recommandé d'affecter 1/3 de la mémoire à SQUID. Ici 40 MB pour 128 Mo de RAM.

cache_swap_low 75
cache_swap_high 90
 # Lorsque le cache est occupé à 90% il se vide jusqu'à atteindre la valeur de 75%.

maximum_object_size 8192 KB
 # Taille max des objets stockés en cache.

cache_dir ufs /cache1 2000 16 256
cache_dir ufs /cache2 2000 16 256
 # Emplacement et taille du cache.

cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log
 # Emplacement des fichiers de log

cache_effective_user proxy
 # Pour éviter que squid soit lancé par root. Ici les processus appartiennent à l'utilisateur 'proxy'

logfile_rotate 2
 # Pour faire tourner les logs toutes les 2 semaines.
 # très utile car si squid est beaucoup utilisé les fichiers de log peuvent être volumineux

acl QUERY urlpath_regex cgi-bin ? .cgi .pl .php3 .php4 .asp
acl deny QUERY
 # Type de page à ne pas stocker dans le cache (formulaire par exemple)

acl all src 0.0.0.0/0.0.0.0
acl lan src 192.168.0.0/24
acl am time MTWHF 08:30-12:30
acl pm time MTWHF 13:30-17:30
http_access allow lan am
http_access allow lan pm
http_access deny all
 # ACCESS LIST : ici seul le réseau local est autorisé à utiliser le proxy
 # Accès possible du lundi au vendredi de 8h30 à 12h30 et de 13h30 à 17h30

http_port 8080
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
 # Accélerateur http

log_fqdn on | off
 # Permet d'utiliser les noms pleinement qualifiés ou les adresses IP dans les fichiers de log. Préférez l'option 'off' pour optimiser les performances, cela évite les consultations par un serveur DNS.

redirect_children 20
 # Nombre de processus pouvant être lancés simultanément

redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
 # Redirige les requêtes vers squidGuard. Utile si vous souhaitez mettre en place du filtrage

Si vous souhaitez activer l'authentification par le proxy, c'est-à-dire que les utilisateurs doivent entrer un login et un mot de passe pour accéder à Internet, il faut ajouter les lignes suivantes dans le fichier squid.conf :

acl auth proxy_auth REQUIRED
http_access allow auth
auth_param basic children 5

auth_param basic realm Squid Authentification

Ensuite, il faut choisir la méthode d'authentification :

auth_param basic program /usr/lib/squid/ncsa_auth /usr/lib/squid/passquid
 # Authentification de type NCSA (fichier au format login:password)
 # Pour ajouter les utilisateurs que vous souhaitez autoriser à accéder à internet :
 # 'htpasswd -c /usr/lib/squid/passquid util' si le fichier n'existe pas sinon lancez la commande sans -c

auth_param basic program /usr/lib/squid/ldap_auth
-D "cn=admin,dc=mynetcourse,dc=info" -w password
-b "dc=mynetcourse,dc=info"
 # Authentification LDAP sur un serveur linux (OpenLDAP) :
 # La requête LDAP est effectué avec l'utilisateur admin (mot de passe : password)
 # Si le serveur LDAP n'est pas spécifié, la requête est effectuée en local (127.0.0.1)

auth_param basic program /usr/lib/squid/ldap_auth
-R -D "CN=internet,CN=Users,DC=mrim-combs,DC=com" -w mdp
-b "dc=mrim-combs,dc=com" -f sAMAccountName=%s -h 192.168.1.253
 # Authentification LDAP sur un serveur Windows 2000/2003 (Active Directory) :
 # La requête LDAP est effectuée par l'utilisateur 'internet' qui se trouve dans le conteneur 'Users'
 # du domaine 'mrim-combs.com'. Il a pour mot de passe 'mdp'. Le proxy se connecte sur l'active directory
 # dont le serveur a pour adresse IP 192.168.1.253 et vérifie que le couple login/password entré par
 # l'utilisateur existe dans l'active directory.

FILTRAGE avec SquidGuard
SquidGuard est un utilitaire permettant de filtrer les requêtes web des clients passant par le proxy. Il n'a d'utilité que si l'on impose les clients à passer par le proxy. Nous verrons un peu plus tard comment faire cela (cf. rubrique Configurer un pare-feu avec NETFILTER).

Exemple simple de configuration du fichier **/etc/squid/squidGuard.conf** :

dbhome /var/lib/squidguard/db
 # Emplacement des fichiers de base de squidGuard
logdir /var/log/squid
 # Emplacement des fichiers de logs

src profs {
user simonet delellis roche chamelot cassinelli thenieres thibault
}
 # Définition du groupe "profs"

destination interdit {
domainlist interdit/domains
urllist interdit/urls
}
 # Définition du filtrage "interdit"

dest porn {
domainlist porn/domains
urllist porn/urls
expressionlist porn/expressions
}
 # Définition du filtrage "porn"

acl {
profs {
pass all
}
 # Le groupe 'profs' cité ci-dessus est autorisé à accéder à Internet sans aucune restriction

default {
pass !interdit !games !interdit all
redirect http://www.mrim-combs.com
}
 # Par défaut, tout autre utilisateur authentifié peut accéder à Internet mais se verront refusés les sites contenus dans les filtres 'interdit' et 'porn'. Toutes requêtes refusées seront redirigées vers le site www.mrim-combs.com
}

[Retour au menu]

SERVEUR WEB : APACHE

Il est souvent utile d'installer en même temps mysql et php.

- Fichier de configuration : **/etc/apache/httpd.conf**
- Répertoire par défaut des sites virtuels : **/etc/apache/conf.d/**
- Répertoire racine du serveur Web : **/var/www/**
- Gestion du service: **/etc/init.d/apache** {**start**|**stop**|**restart**|**status**}

Pour que APACHE soit opérationnel on doit donner un nom au serveur web par défaut. Pour cela il faut décommenter la ligne #servername dans le fichier httpd.conf. Une fois que le serveur Web a un nom, il faut redémarrer le service. L'accès au serveur se fait en tapant dans le navigateur d'un poste se trouvant sur le réseau : http://nom-du-serveur ou http://adresse-IP-du-serveur.

Confiuration du serveur :

Le fichier de configuration principal est httpd.conf qui se trouve dans /etc/apache/. Nous allons voir les principales directives de ce fichier.

ServerType
Spécifie le mode de lancement du serveur. Par défaut, standalone. Sinon, inetd ce qui signifie que le serveur est lancé par inetd.

ServerName
Nom réseau du serveur.

ServerAdmin
Adresse email du responsable du site.

ServerRoot
Spécifie le répertoire racine d'administration du serveur.

User, **Group**
Utilisateur et groupe auxquels appartiennent les processus httpd.

Port
Port de communication associé au service web. Par défaut, 80.
Port 80

Listen
Cette directive est associée à l'utilisation des sites virtuels. Elle permet de spécifié qu'il existe plusieurs adresse IP et/ou plusieurs ports sur lesquels le serveur doit écouter.
Listen 192.168.0.253:80
Listen 192.168.0.253:8000

DirectoryIndex
Spécifie les noms de fichier d'index d'un répertoire. Par exemple, index.htm, index.html, index.php, accueil.htm,...

VirtualHost
Directive de type bloc (<VirtualHost></VirtualHost>) permettant de déclarer des sites virtuels.

Directory
Directive de type bloc permettant la configuration des répertoires.

Très souvent, un même serveur web héberge plusieurs sites. Nous allons voir comment héberger plusieurs domaines. Dans l'exemple ci-dessous nous allons créer 2 sites (site1 et site2) associés respectivement aux domaines site1.com et site2.com. Un serveur web fonctionne de la façon suivante : Il existe un serveur web par défaut accessible par l'adresse IP du serveur ou par son nom. Si l'on souhaite ajouter des sites, il faut créer des sites virtuels (virtual host).

Deux sites associés à deux adresses IP :

```
<VirtualHost 192.168.0.253>
    ServerName www.site1.com
    DocumentRoot /var/www/html/site1/
</VirtualHost>
<VirtualHost 192.168.0.252>
    ServerName www.site2.com
    DocumentRoot /var/www/html/site2/
</VirtualHost>
```

Deux sites qui diffèrent par leur n° de port :

```
Listen 192.168.0.253:80
Listen 192.168.0.253:8000
<VirtualHost 192.168.0.253:80>
    ServerName www.site1.com
    DocumentRoot /var/www/html/site1/
</VirtualHost>
<VirtualHost 192.168.0.253:8000>
    ServerName www.site2.com
    DocumentRoot /var/www/html/site2/
</VirtualHost>
```

Deux sites qui diffèrent par leur nom :

```
NameVirtualHost 192.168.0.253
<VirtualHost 192.168.0.253>
    ServerName www.site1.com
    DocumentRoot /var/www/html/site1/
</VirtualHost>
<VirtualHost 192.168.0.253>
    ServerName www.site2.com
    DocumentRoot /var/www/html/site2/
</VirtualHost>
```

Depuis la version 2 (httpd-2.0...), on peut utiliser l'écriture suivante. Cela signifie que toutes les interfaces, quelques soit l'adresse IP, sont utilisées.

```
NameVirtualHost *:80
NameVirtualHost *:443
<VirtualHost *:80>
    ...
    ...
</VirtualHost>
<VirtualHost *:443>
    ...
    ...
</VirtualHost>
```

Alias et Redirect :

```
<VirtualHost * :80>
    ServerName www.site1.com
    DocumentRoot /var/www/html/site1/
    Alias /public /var/www/html/site2/   à crée un répertoire virtuel
    Redirect /private https://private.site1.com  à redirige la requête vers un autre site
</VirtualHost>
<VirtualHost * :80>
    ServerName www.site2.com
    DocumentRoot /var/www/html/site2/
</VirtualHost>
<VirtualHost * :80>
    ServerName private.site1.com
    DocumentRoot /var/www/html/private/
</VirtualHost>
```

Restriction d'accès à un site :

```
<VirtualHost 192.168.0.253>
    ServerName private.site1.com
    DocumentRoot /var/www/html/site1/private
    <Directory /var/www/html/site1/private>
        AuthUserFile /etc/apache/.private
        AuthName "Acces prive"
        AuthType Basic
        <Limit GET POST>
            require valid-user
        </Limit>
    </Directory>
```

Il faut ensuite créer le fichier .private qui contient la liste des utilisateurs autorisés à visualiser le site. Pour cela, utilisons la commande htpasswd :

```
htpasswd -c /etc/apache/.private admin
New password:
Re-type new password:
Adding password for user admin
```

L'option -c permet de créer le fichier .private. Ici l'utilisateur admin vient d'être autorisé à accéder au site private.site1.com.

htpasswd /etc/apache/.private util
New password:
Re-type new password:
Adding password for user util
L'utilisateur util vient d'être ajouté à la liste des utilisateurs autorisé à accéder au site. Il n'est pas nécessaire d'utiliser l'option -c puisque le fichier .private existe déjà.

Les principales directives inclues dans <Directory> :

Options
Spécifie des options associées au répertoire. Les pricipales sont :
None : Aucune option
All : Toutes les options
Indexes : On autorise la visualisation du répertoire dans le cas où la page d'accueil ne serait pas présente.
FollowSymLinks : On autorise le client à suivre les liens symboliques.

AccessFileName
Nom du fichier de configuration qui déclare les utilisateurs autorisés à accéder au site. Par défaut, .htaccess. Chaque répertoire que l'on souhaite sécuriser doit contenir ce fichier.

AllowOverride
Cette directive peut prendre deux valeurs, All et None, qui permettent respectivement d'utiliser ou non le fichier .htaccess.

AuthType
Spécifie le protocole d'identification d'un utilisateur. Le plus utilisé est Basic, car c'est le seul à être supporté. Dans ce cas, les informations passent en clair.

AuthUserFile
Spécifie le nom du fichier qui contient les utilisateurs autorisés à accéder au site. Ce fichier contient le nom des utilisateurs et leur mot de passe. La création de ce fichier et l'ajout d'utilisateur se fait à l'aide de la commande htpasswd.

Require
Lorsque l'on souhaite sécurisé l'accès à un site, cette directive doit être présente car elle impose authentification des utilisateurs.
Par exemple, Require valid-user.

order allow, deny
Ces directives permettent de contrôler l'accès au site.

Exemple:
order deny,allow
allow from 192.168.1
deny from all
On autorise le réseau 192.168.1.0 et on interdit tout le reste (intranet).

Exemple:
order allow,deny
allow from all
Tout le monde est autorisé à accéder au site (site public).

[Retour au menu]

SERVEUR FTP

Il est possible d'installer différents services ftp, les plus connu sont wu-ftpd, vsftpd, pureftpd et proftpd.

VSFTPD (Very Secure FTPD)

Le serveur VSFTPD utilise le fichier **/etc/passwd** pour indiquer les répertoires de connexion des utilisateurs.

Fichiers de configuration :

- **/etc/vsftpd.conf** : fichier principal de configuration du service
- **/etc/vsftpd.ftpusers** : liste des utilisateurs qui ne seront pas autorisés à se connecter au service ftp. Tous les autres sont autorisés

- **/etc/vsftpd.user_list** : Liste des utilisateurs dont le répertoire de connexion est chrooter. Les utilisateurs sont cloisonnés dans leur répertoire, ils ne pourront pas remonter l'arborescence. Leur répertoire est la racine de l'arborescence. Il est souhaitable d'utiliser ce fichier pour augmenter la sécurité de votre serveur
- **/etc/vsftpd.chroot_list** :

PROFTPD

Description du fichier de configuration : **/etc/proftpd.conf**

ServerName "ProFTPD Serveur"
 # Nom du serveur
ServerType standalone
 # Indique si le serveur fonctionne seul ou s'il est lancé par le super-daemon
DefaultServer on
 # Utile si on utilise les VirtualHost
ServerIdent on "Serveur FTP"
 # Permet de masquer la version du serveur en rajoutant une ligne
AccessGrantMsg "Bienvenue %u !"
 # Permet d'afficher un message d'accueil (%u est la variable contenant le nom de l'utilisateur)
DeferWelcome on
 # Permet de ne pas donner d'informations précises sur le serveur
ServerAdmin root@localhost.localdomain
 # Adresse de l'administrateur du service ftp
Port 21
 # Port d'écoute du serveur
Umask 022
 # Les droits d'un fichier ou d'un répertoire créé par un utilisateur sur le serveur, est obtenu en réalisant un ET LOGIQUE entre le répertoire courant et le complément de 022 (755). Cette valeur de 022 est très souvent celle utilisée.
MaxInstances 30
 # Nombre maximum de processus fils que va gérer proftpd (Ne pas dépasser 30 sinon on est vulnérable à des attaques de type DoS)
MaxLoginAttempts 3
 # Nombre de tentative de connexion pour un utilisateur authentifié
AllowStoreRestart on
AllowRetrieveRestart on
 # Autorise la reprise du téléchargement
TransferLog /var/log/proftpd/proftpd.xferlog
SystemLog /var/log/proftpd/proftpd.syslog
ScoreboardFile /var/log/proftpd/proftpd.scoreboard
PidFile /var/log/proftpd/proftpd.pid
 # emplacement des fichiers de log
MasqueradeAddress 80.65.224.232
PassivePorts 50000 50010
 # Utile pour le ftp passif dans le cas ou le routeur ne NATte pas le ftp. Evite que le serveur envoie son adresse IP locale (privée)
DefaultRoot ~
 # Permet de cloisonner l'utilisateur dans son répertoire de connexion (cela évite qu'il puisse remonter l'arborescence)

```
<Directory>
    Allowoverride on
</Directory>
```
 # On autorise tout pour les clients authentifiés dans leur répertoire de connexion

```
<Anonymous /home/proftpd>
    User ftp
    Group ftp
    UserAlias anonymous  ftp
    MaxClients 10
    <Limit WRITE>
        DenyAll
    </Limit>
    <Directory upload/*>
        <Limit READ>
            DenyAll
        </Limit>
        <Limit STOR>
            AllowAll
        </Limit>
    </Directory>
</Anonymous>
```
 # Configuration de l'accès anonyme. Seul les utilisateurs ftp et anonymous peuvent se connecter en anonyme. Ces utilisateurs peuvent déposer des fichiers dans le répertoire upload. Le nombre de connexion max simultanée sur le serveur est de 10.

[Retour au menu]

SERVEUR DE MESSAGERIE : SENDMAIL - FETCHMAIL - PROCMAIL

SENDMAIL est le serveur de mails des distributions RedHat (POSTFIX pour les distributions Mandrake).
La commande ntsysv permet de vérifier que SENDMAIL est lancé au démarrage.

Fichiers de configuration : /etc/sendmail.cf aliases , /etc/mail/relay-domains local-host-name virtusertable
Gestion du service: /etc/rc.d/init.d/sendmail start | stop | restart | status

A l'installation de la distribution, SENDMAIL est opérationnel, il peut envoyer du courrier. En effet s'il reçoit une requête SMTP pour envoyer un courrier à toto@free.fr, il interroge le serveur DNS qui lui indique le serveur SMTP à contacter. Ici smtp.free.fr.
Le mail fonctionne aussi pour tous les utilisateurs déclarés sur la machine. Si un utilisateur veut envoyer un mail à toto il peut le faire par exemple en ligne de commande de la façon suivante : mail toto

SENDMAIL serveur de mails accessible depuis un client Outlook, Netscape, Eudora …
Le package imap doit être installé. Vérifiez que inetd ou xinetd lance ipop3 (ouverture du port 110).

Options possibles :

- Le serveur de mail peut être accessible par tout le monde, y compris ceux qui ne font pas partis du réseau local. Certaines personnes mal intentionnées utilisent les serveurs de mails mal sécurisés en tout cas ceux qui autorise le contrôle du relais. Pour éviter que cela se produise, il faut autoriser certaines machines à utiliser le serveur SMTP. Il faut éditer le fichier **/etc/mail/access**.

```
192.168.0. # on autorise le réseau local
80.65.224.232 # on autorise la machine d'adresse IP 80.65.224.232
```

- Le fichier **/etc/mail/local-host-names** (**/etc/sendmail.cw**) permet de spécifier les domaines ou sous-domaines gérer par le serveur.

```
domain.com
srv.domain.com
```

- Le fichier **/etc/mail/virtusertable** permet de diriger le courrier pour les comptes connus du système ou de créer des boites sans qu'il n'existe de compte sur la machine.

```
bruno.simonet@domain.com bruno
bruno-simonet@domain.com bruno
toto@domain.com   toto@free.fr
@domain.com   admin
```

Ici, le courrier envoyé à bruno.simonet@domain.com et bruno-simonet@domain.com sera redirigé vers le compte local bruno. Dans cet exemple, toto n'est pas un compte connu du système mais il est quand même possible d'envoyer du courrier à toto@domain.com. Il sera alors redirigé vers une boite externe. Tout ce qui arrive au domaine est envoyé à l'utilisateur local admin. Pour cette dernière ligne, attention au spam. Utilisez plutôt la ligne :

```
@domain.com   error:nouser No such user here
```

- Le fichier **/etc/aliases** permet de créer des alias de boites aux lettres. Il permet aussi de créer des listes de diffusion.

```
# alias
postmaster: bruno
# liste de diffusion
profs: bruno, william, gerard, issam
```

Commandes utiles :

mailq ou **sendmail -bp** permet de vérifier le courrier en attente.
sendmail -q force la livraison du courrier en attente.
sendmail -q1h permet de dire à sendmail de traiter la file d'attente toutes les heures.

FETCHMAIL permet de récupérer le courrier. C'est un serveur POP. Pour indiquer les comptes de messagerie à interroger il faut éditer le fichier /root/.fetchmailrc. Ce fichier est structuré de la manière suivante :

```
poll pop.free.fr protocol POP3
    username toto password toto is toto option keep
    username tata password tata is tata option keep
poll pop.libertysurf.fr protocol POP3
    username titi password titi is titi option keep
```

Ici, il existe un utilisateur toto sur le serveur de messagerie. FETCHMAIL contacte le serveur POP de free, le protocole utilisé est POP3, le nom du compte est toto et le mot de passe est toto. keep permet de laisser les messages sur le serveur.

Pour récupérer le courrier il faut lancer la commande fetchmail. fetchmail -k permet de laisser une copie sur le serveur dans le cas ou l'option keep n'est pas spécifiée dans le fichier .fetchmailrc.

Si l'on souhaite automatiser la récupération du courrier il faut éditer la crontab à l'aide de la commande crontab -e. L'édition du fichier se fait à l'aide de l'éditeur vi.

Procédure : (utilisation de vi)

[i] , pour passer en mode édition,
Editez la crontab comme vous le souhaitez (cf. exemple ci-dessous),
[ESC] pour passer en mode commande,
[:wq] pour enregistrer le fichier puis sortir.

Structure de la crontab : 1 2 3 4 5 6
1 : minute de 00 à 59
2 : heure de 00 à 23
3 : jours du mois de 1 à 31
4 : mois de l'année de 1 à 12
5 : jours de la semaine de 0 à 7 (0 étant le dimanche)
6 : commande telle qu'elle serait tapée dans le shell

Exemple :
Nous souhaitons récupérer le courrier toutes les heures de 7h00 à 18h00 du lundi au vendredi :
0 7-18 * * 1-5 fetchmail
Nous souhaitons récupérer le courrier à 7h00, 13h00 et 18h00 du lundi au vendredi :
0 7,13,18 * * 1-5 fetchmail

PROCMAIL permet de filtrer le courrier. Il est possible de rediriger le courrier en destination d'une même boite vers d'autre compte en fonction de plusieurs critères (sujet, présence d'un mot particulier dans le corps du message, expéditeur…).

[Retour au menu]

SERVEUR DE FICHIERS : SAMBA

Le serveur SAMBA permet depuis un serveur Linux de partager des ressources pour des clients Windows 95, 98, Me, NT, 2000, Mac…
SAMBA est capable d'offrir les services réseaux suivants :

- partages de fichiers et de répertoires,
- partages d'imprimantes,
- gestion des comptes utilisateurs,
- gestion des permissions d'accès,
- exécution de scripts de connexion personnalisés.

Répertoire de configuration : **/etc/samba**
Fichier de configuration de SAMBA :
smb.conf : fichier de configuration
smbusers : fichier contenant les utilisateurs SAMBA
smbpasswd : fichier des mots de passe
Gestion du service: **/etc/init.d/smb start** | **stop** | **restart** | **status**
La commande **testparm** permet de vérifier la syntaxe du fichier **smb.conf**

Exemple de configuration (fichier smb.conf) :

[global]
workgroup = MRIM
 # Nom du groupe de travail : MRIM
netbios name = SAMBA
 # Nom du serveur : SAMBA
server string = Serveur SAMBA
 # Description du serveur visible dans le voisinage réseau : Serveur SAMBA
log /var/log/samba/%m.log
 # Emplacement des fichiers de log
encrypt passwords = yes
 # Permet de crypter les mots de passe (compatible Windows)
security = share / user
 # Sécurité au niveau partage / utilisateur

[homes]
comment = Repertoire perso
 # Description du partage visible dans le voisinage réseau
writable = yes
 # Ecriture autorisée
browseable = no
 # Non visible dans le voisinage réseau
create mode = 0700
 # Droits imposés aux fichiers créés dans cette ressource
directory mode = 0700
 # Droits imposés aux répertoires créés dans cette ressource

```
[public]
   comment = Repertoire public
   path = /home/public
   public = yes
   browseable = yes
   writable = yes
   # Création d'un répertoire public accessible par tous en lecture/écriture

[outils]
   comment = Repertoire outils
   path = /home/outils
   public = yes
   browseable = yes
   write list = @profs
   create mode = 0744
   directory mode = 0744
   force group = prof
   # Création d'un répertoire outils accessible en lecture par tous
   # Seuls les membres du groupe profs peuvent écrire dans ce répertoire

[travail]
    comment = Repertoire de travail
    path = /home/outils
    public = no
    browseable = yes
    writable = yes
    valid users = @eleves @profs
    # Création d'un répertoire outils accessible uniquement par
    # les membres du groupe profs et élèves
```

Commandes utiles :

- **smbadduser toto:uid** : permet d'ajouter un utilisateur SAMBA
- **smbpasswd toto** : permet de changer le mot de passe de l'utilisateur SAMBA.
- **smbpasswd -a toto** : permet d'ajouter un utilisateur SAMBA
- **smbpasswd -d toto** : permet de supprimer un utilisateur SAMBA

SAMBA : Contrôleur Principal de Domaine

```
[global]
   security = user
   domain master = yes
   domain logons = yes
```

Remarque : Le nom donné à la rubrique **workgroup** est le nom choisi pour le domaine.

[Retour au menu]

SERVEUR DHCP : DHCPD

dhcp3-server est le serveur DHCP sous linux. Il dépend du package **dhcp3-common**

Fichier de configuration : **/etc/dhcp3/dhcpd.conf**
Fichier des options : **/etc/default/dhcp3-server**
Fichier de log : **/var/lib/dhcp3/dhcpd.leases**
Gestion du service: **/etc/init.d/dhcpd start** | **stop** | **restart** | **status**

Description du fichier de configuration dhcpd.conf :

```
ddns-update-style none / interim / ad-hoc;
# Spécifie le mode de mise à jour du DNS
Subnet 192.168.1.0 netmask 255.255.255.0 {
    # Déclaration du réseau et du masque
    Range 192.168.1.100 192.168.1.199;
    # Plage d'adresses disponibles pour les clients
    Option domain-name "mrim.net";
    # Nom du domaine
    Option domain-name-servers 192.168.1.253, 192.168.1.252;
    # Adresse du serveur DNS
    Option broadcast-address 192.168.1.255;
```

 # Adresse utilisée pour la diffusion
Option subnet-mask 255.255.255.0;
 # Valeur du masque de sous-réseau
Option routers 192.168.1.254;
 # Adresse de la passerelle
Option ntp-servers 192.168.1.251;
 # Adresse du serveur de temps
Default-lease-time43200;
 # Durée du bail (12 heures)
Max-lease-time 86400;
 # Durée maximale du bail (24 heures)
}

Il est possible d'affecter toujours la même adresse à la même machine. L'affectation est fonction de l'adresse MAC.

Host station {
 # Déclaration de la machine.
Hardware ethernet 00:00:88:88:aa:aa;
 # Adresse MAC de la machine.
Fixed-address 192.168.1.10;
 # Adresse IP affectée à cette machine.
}

Une fois le fichier dhcpd.conf modifié on peut lancer le service. A tout moment la visualisation du fichier dhcpd.leases permettra de connaître les machines ayant demandées un bail.

[Retour au menu]

SERVEUR DNS : BIND

Pour installer BIND il faut utiliser le package bind9 et caching-nameserver. Pour tester la configuration DNS il est possible d'installer aussi le packet bind-util.

Répertoire de configuration : **/etc/bind/**

Fichiers de configuration :

- **named.conf** : paramètres généraux
- **named.ca** : indique les serveurs DNS racines
- **localhost.zone** : résolution des adresses locales
- **named.loca**l : résolution inverse des adresses locales
- **domain.com.hosts** : correspondance nom de machine > @ IP
- **0.168.192.hosts** : correspondance @ IP > nom de machine

Gestion du service: **/etc/init.d/bind9 start** | **stop** | **restart** | **status**

La configuration de BIND consiste à configurer 4 fichiers. En effet le fichier named.ca ne doit pas être touché. C'est le fichier qui contient les adresses des serveurs DNS racines appelé aussi serveur DNS root. Nous souhaitons ici configurer un domaine (domain.com) avec comme adresse de réseau 192.168.0.0. Le serveur DNS (srv) a pour adresse IP 192.168.0.253 et le serveur de mail (mail) a pour adresse IP 192.168.0.252.

Nous allons voir la syntaxe du fichier de configuration du service named (named.conf) puis nous verrons comment écrire un fichier de zone.
Ce fichier est composé de 2 parties :
- La 1° partie concerne les options du serveur. Dans notre cas, le serveur essaye de répondre en premier aux requêtes DNS qui lui sont envoyées (forward first). S'il ne réussi pas à les résoudre, il interrogera les serveurs spécifiés dans la rubrique forwarders, ici les serveurs DNS de notre fournisseur d'accès.
- La 2° partie permet de créer les zones que le serveur devra gérer.

zone permet de créer une zone. C'est en fait un domaine DNS.
type permet de préciser le rôle du serveur :
type master : serveur maître
type slave : serveur secondaire
type hint : serveur cache
file permet de préciser le fichier qui décrit le domaine.

 # Exemple d'une zone où le serveur joue le rôle d'un serveur maître :
zone "lycée.com" {
type master;
file "file.lycee.com";
};

 # Exemple d'une zone pour un serveur secondaire :
zone "lycee.com" {

```
   type slave;
   file "file.lycee.com";
   masters {
      192.168.0.250;
   }
};
```

Description du fichier **named.conf** :

```
options {
   directory "/var/named";
   forward first;
   forwarders {
      62.4.16.70;
      62.4.16.80;
      62.4.17.109;
      };
   };

zone "." IN {
   type hint;
   file "named.ca";
   };

zone "localhost.zone" IN {
   type master ;
   file "localhost.zone"
   notify no ;
   };

zone "0.0.127.in-addr.arpa" IN {
   type master;
   file "named.local"
   notify no;
   };

zone "domain.com" IN {
   type master;
   file "domain.com.hosts";
   };

zone "0.168.192.in-addr.arpa" IN {
   type master;
   file "0.168.192.hosts";
   };
```

Voyons maintenant la configuration d'un fichier de zone :

[nom] [durée de vie] [classe] [type] [donnée]
nom : c'est le nom d'un objet, par exemple un nom DNS. Si le nom n'est pas présent, l'enregistrement référence l'objet précédent.
durée de vie : c'est la durée de vie en secondes de l'objet dans le cache. S'il n'est pas précisé, la durée de vie sera celle précisée dans le champ SOA (valeur minimum).
classe : la classe indique le réseau de transport utilisé. Les réseaux TCP/IP sont définis par la classe IN (INternet).
type : c'est le type de l'enregistrement. Voici les principaux :
SOA (Start Of Authority) : Débute un enregistrement général d'un fichier de zone.
NS (Name Server") : Identifie un serveur de nom.
A (Adress) : Permet la conversion d'un nom DNS en adresse IP.
PTR (Pointer) : Permet la conversion d'une adresse IP en nom DNS.
MX (Mailer eXchange): Identifie un serveur de messagerie.
CNAME (Canonical NAME) : Donne un alias à une machine.
HINFO (Host INFOrmation) : Décrit l'architecture matérielle et le système d'exploitation d'un hôte.
WKS (Well Know Services) : Décrit les services disponibles.
donnée : c'est la donnée de l'enregistrement, par exemple l'adresse IP pour un enregistrement de type A.

Syntaxe d'un fichier de zone :

```
@   ttl   IN   SOA   nom_du_servuer_principal   email_de_l'administrateur (
                        numéro de série
                        rafraichissement
                        nouvel essai
                        expire
                        minimum )
```

```
IN    NS    nom_d'un_serveur_maître
```

Description du fichier **domain.com.hosts** :

```
$TTL 86400
@ IN SOA srv.domain.com. root.srv.domain.com. (
                    2001051500 ;
                    28800 ;
                    14400 ;
                    3600000 ;
                    86400 ) ;
   IN   NS    srv.domain.com.
   IN   MX  10  mail.domain.com.

;@ IP des machines
srv    IN    A    192.168.0.253
mail    IN    A    192.168.0.252

;alias
www   IN    CNAME    srv
smtp    IN    CNAME    mail
pop   IN    CNAME    mail
```

Description du fichier **0.168.192.hosts** :

```
$TTL 86400
@ IN SOA srv.domain.com. root.srv.domain.com. (
                    2001051500 ;
                    28800 ;
                    14400 ;
                    3600000 ;
                    86400 ) ;
    IN    NS   srv.domain.com.

;@ IP inverses
253    IN    PTR    srv.domain.com.
252    IN    PTR    mail.domain.com.
```

Tester le fonctionnement du DNS :

named-checkconf
cette commande permet de tester le fichier de configuration **/etc/named.conf**.
named-checkzone
cette commande permet de tester les zones.
named-checkzone domain.com /var/named/domain.com.hosts

Commandes utiles :

nslookup
outils permettant de tester le DNS.
[root@srv /root]# nslookup
> srv
permet d'obtenir des informations sur la machine srv
> set all
affichage de la configuration DNS du serveur
> exit
pour quitter nslookup

dig
commande permettant d'obtenir des informations sur une machine ou sur un domaine.
dig srv.domain.com
dig domain.com

host
commande permettant d'obtenir des informations sur une machine ou sur un domaine.
host -al domain.com
host srv.domain.com

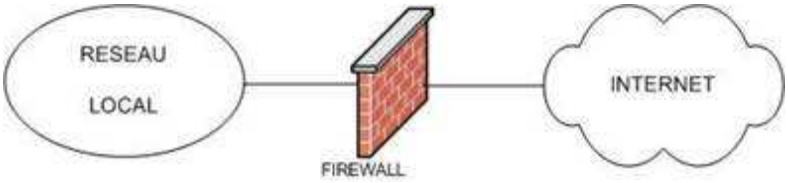
[Retour au menu]

Configurer un pare-feu avec NETFILTER

Netfilter est le firewall des distributions linux récentes pris en charge par les noyaux 2.4. Il est le remplaçant de ipchains. La configuration se fait en grande partie par la commande iptables dont nous verrons la syntaxe par la suite.

Avant toute chose :

Le "firewall" ou "Pare-feu" est un système, logiciel ou matériel, permettant de contrôler et éventuellement bloquer la circulation des paquets. Il se place entre 2 voir plusieurs réseaux. En règle générale, on le trouve entre un réseau local et Internet. Un firewall dispose d'au moins 2 interfaces.



Il permet :

- d'autoriser ou refuser l'accès au réseau local depuis l'extérieur,
- d'autoriser ou refuser l'accès vers l'extérieur (Internet) depuis le réseau local,
- d'établir des redirections permettant de rendre accessible depuis l'extérieur un serveur appartenant au réseau local, ...

Il analyse donc les paquets qui entrent, qui sortent et qui circulent par ces interfaces. Il sait travailler sur les couches 3 et 4 du modèle OSI. Il est aussi capable de faire des restrictions en fonction des adresses MAC ou encore de faire du partage de connexion Internet en laissant passer des protocoles comme FTP ou IRC, par conséquent, il peut travailler sur les couches 2 et 7.

La mise en place d'un firewall est une tâche délicate. Cela nécessite d'avoir de bonne connaissance sur le protocole réseaux IP et sur les protocoles de transport TCP, UDP. Il vous faudra avoir une parfaite connaissance de ce qu'est un port de communication.

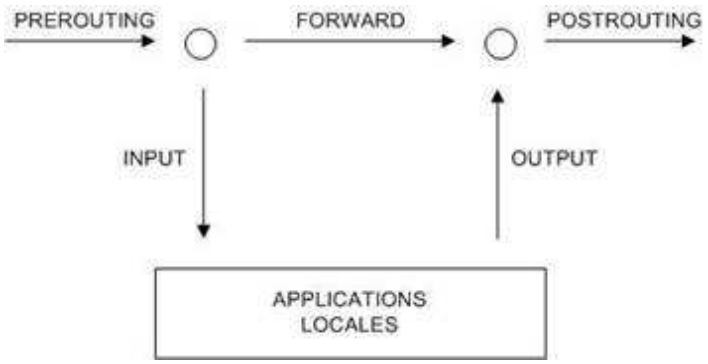
Fonctionnement :

- Les paquets sont examinés, le firewall regarde s'ils correspondent aux règles dans l'ordre croissant de numérotation.
- Si le paquet correspond à la 1° règle, il prend la décision correspondante (accepte, refuse, ...) et passe à l'examen du paquet suivant.
- Si le paquet ne correspond à la 1° règle, il passe à la règle suivante jusqu'à ce qu'une décision soit prise. Si aucune règle ne correspond, le paquet est soumis à la règle par défaut.

Définitions :

- Table : lorsqu'un paquet arrive, Netfilter le prend en charge et le passe à son système d'évaluation de règles que sont les tables. Une table est constituée de chaînes.
- Chaîne : la chaîne est une suite de règles constituées de motifs (pattern) et de cible (target).
- Motif : le motif permet de reconnaître les paquets selon un ou plusieurs critères.
- Cible : la cible est la décision prise lorsque le paquet est reconnu selon les critères

Description des tables :



FILTER	NAT	MANGLE
Chaînes : INPUT OUTPUT	Chaînes : PREROUTING POSTROUTING	Chaînes : Toutes

FORWARD Règles : ACCEPT DENY DROP LOG Modules : iptables_filter.o	OUTPUT Règles : SNAT DNAT MASQUERADE REDIRECT Modules : iptables_nat.o	Règles : MARK TTL TOS TCPMSS IPV4OPTSSTRIP Modules : iptables_mangle.o

Construction d'une chaîne :

	TABLE	CHAINE	MOTIF DE RECONNAISSANCE	CIBLE
iptables	-t filter	-A INPUT	-p, -s, -d, --dport, --sport, -m, ...	-j ACCEPT
	nat	OUTPUT		DENY
	mangle	FORWARD		DROP
		PREROUTING		MASQUERADE
		POSTROUTING		REDIRECT
				DNAT
				SNAT

Utilisation de la commande iptables :

-L Liste les règles d'une table :
iptables -L : on liste les règles de la table par défaut (FILTER)

-t Permet de spécifier la table. Si on utilise pas ce switch on s'adresse à la table par défaut (FILTER) :
iptables -t nat -L : on liste les règles de la table NAT

-P Fixe la police par défaut :
iptables -P INPUT DROP : par défaut, tout ce qui entre est rejeté

-F Vide les chaînes d'une table :
iptables -F : vide les règles de la table FILTER

-N Crée une nouvelle chaîne :
iptables -N LOG_DROP : ajoute la chaîne LOG_DROP

-X Supprime une chaîne :
iptables -X LOG_DROP : supprime la chaîne LOG_DROP

-E Renomme une chaîne

-Z RAZ des compteurs d'une chaîne

-A Ajoute une règle :
iptables -A INPUT -i lo -j ACCEPT : ajoute une règle dans la chaîne INPUT de la table FILTER. Ici, tout ce qui entre par l'interface lo est accepté.

-D Supprime une règle :
iptables -D INPUT -i lo -j ACCEPT : supprime la règle spécifiée. Ici, on supprime la règle que l'on a créée au-dessus.

-R Remplace une règle

-I Insère une règle

-p Spécifie le protocole : tcp, udp, icmp, all
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT : autorise l'accès au port 80 par l'interface eth0.
iptables -A OUTPUT -p tcp -o eth0 --sport 1024: --dport 80 -j ACCEPT : autorisation des requêtes clientes depuis eth0 vers un serveur web.

-s Spécifie l'adresse source :
iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -j ACCEPT : autorise l'accès au service SSH depuis le réseau local.

-d Spécifie l'adresse destination :
iptables -A INPUT -p tcp -d 192.168.1.1/24 --dport 21 -j ACCEPT : autorise l'accés au service FTP sur la machine 192.168.1.1.

--sport Spécifie le port source :
iptables -A INPUT -p udp -o eth0 --sport 53 -j ACCEPT : autorise les requêtes DNS vers l'extérieur.

--dport Spécifie le port destination :
iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 25 -j DROP : interdit les requêtes SMTP depuis n'importe quelle machine.

-m Permet de spécifier plusieurs port en même temps
iptables -A INPUT -p tcp -s 0.0.0.0 -m multiport --dport 21,25,80,110,143,443 -j ACCEPT : on autorise l'accès aux ports 21, 25, 80, 110, 143 et 443 sur la machine venant de n'importe où.

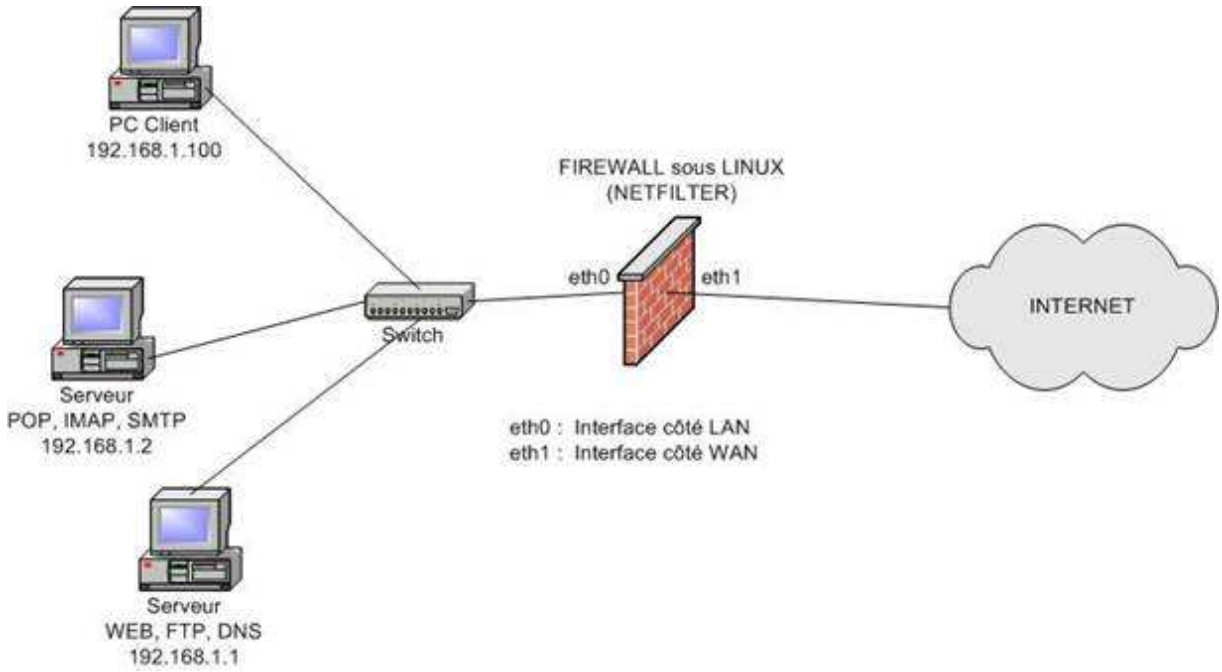
-i Spécifie l'interface d'entrée

-o Spécifie l'interface de sortie

-j Indique ce que l'on désire faire du paquet :
ACCEPT : accepte le paquet
DENY : refuse le paquet
DROP : rejète le paquet
MASQUERADE : masque l'adresse source du paquet (utile pour le partage de connexion)
REDIRECT : redirige le paquet en interne vers un autre port
DNAT : modifie l'adresse de destination du paquet
SNAT : modifie l'adresse source du paquet

Configuration :

Voici le schéma du réseau sur lequel nous allons travailler. Le firewall permet l'accès à Internet depuis le réseau local (passerelle). Il permet aussi depuis Internet d'accéder aux services installés sur les 2 serveurs qui sont: WEB, FTP, POP et SMTP. Les services DNS et IMAP sont accessibles uniquement depuis le réseau local.



Nous allons voir une configuration possible pour le firewall. Pour faciliter la tâche d'administration, nous allons éditer un fichier (script), dans lequel nous allons entrer l'ensemble des règles. Il sera alors facile de modifier la configuration en ajoutant, retirant ou déplaçant les règles. Le fichier que nous allons créer s'appelle fw.

 # Chargement des modules afin de pouvoir utiliser les tables Filter et Nat. Le module ip_nat_ftp permet de NATté le protocole ftp, les clients du réseau local pourront faire des requêtes FTP vers l'extérieur.
modprobe iptable_filter
modprobe iptable_nat
modprobe ip_nat_ftp

 # Mise en place de la politique par défaut. Ici, on interdit tout !
iptables -P INPUT DROP

```
  iptables -P OUTPUT DROP
  iptables -P FORWARD DROP

# On vide les chaînes Filter et Nat (réinitialisation du firewall)
  iptables -F
  iptables -F -t nat

# Interdit les requêtes ping sur le firewall
  echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

# Interdit les broadcasts
  echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Vérifie la source des paquets. Utile contre le spoofing d'IP
 for Filter in /proc/sys/net/ipv4/conf/*/rp_filter; do
   echo 1 > $Filter
 done

# Protège contre le 'SYN-flood', demande de connexion répétée.
  echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# On autorise les paquets en destination ou en provenance de l'interface loopback
  iptables -A INPUT -i lo -j ACCEPT
  iptables -A INPUT -i lo -j ACCEPT

# On autorise les clients du réseau local à faire des requêtes sur internet (http et ftp)
  iptables -A FORWARD -p tcp -i eth0 -o eth1 --dport 80 -j ACCEPT
  iptables -A FORWARD -p tcp -i eth1 -o eth0 --sport 80 -j ACCEPT
  iptables -A FORWARD -p tcp -i eth0 -o eth1 --dport 21 -j ACCEPT
  iptables -A FORWARD -p tcp -i eth1 -o eth0 --sport 21 -j ACCEPT

# On autorise le serveur DNS de faire des requêtes DNS vers l'extérieur (vers les serveur DNS du FAI par exemple)
  iptables -A FORWARD -p tcp -i eth0 -o eth1 -d 192.168.1.1 --dport 53 -j ACCEPT
  iptables -A FORWARD -p tcp -i eth1 -o eth0 -s 192.168.1.1 --sport 53 -j ACCEPT

# Autoriser le forwarding. On autorise le firewall à se passer les paquets d'une interface à une autre. Le fichier ip_forward est à 1 pour activer le forwarding à 0 sinon.
  echo 1 > /proc/sys/net/ipv4/ip_forward

# Activation du NAT
  iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
 ou
  iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j MASQUERADE

# Redirection des requêtes POP et SMTP venant d'Internet vers le serveur de mail
  iptables -t nat -A PREROUTING -p tcp --dport 100 -i ppp0 -j DNAT --to 192.168.1.2
  iptables -t nat -A PREROUTING -p tcp --dport 25 -i ppp0 -j DNAT --to 192.168.1.2

# Redirection des requêtes HTTP et FTP venant d'Internet vers le serveur web
  iptables -t nat -A PREROUTING -p tcp --dport 80 -i ppp0 -j DNAT --to 192.168.1.1
  iptables -t nat -A PREROUTING -p tcp --dport 21 -i ppp0 -j DNAT --to 192.168.1.1

# Mise en place d'un proxy transparent. Prenont le cas où le service Proxy est installé sur la machine qui fait office de firewall.
  iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 80 -j REDIRECT --to-port 3128

# Si on souhaite loguer certaines règles. Ici, on logue toutes requêtes SSH depuis Internet.
  iptables -N LOG_DROP
  iptables -A LOG_DROP -j LOG --log-tcp-options --log-ip-options --log-prefix '[IPT DROP] : '
  iptables -A LOG_DROP -j DROP
  iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 22 -j LOG_DROP

# Autoriser les connexions SSH dans la limite d'une connexion par minute. Les connexions refusées seront loguées
 iptables -N DROP_SYN_SSH
 iptables -A DROP_SYN_SSH -j LOG --log-prefix '[DROP_SYN_SSH]:'
 iptables -A DROP_SYN_SSH -j DROP
 iptables -A INPUT -p tcp --dport 22 --syn -m state --state NEW -m limit --limit 1/m --limit-burst 1 -j ACCEPT
 iptables -A INPUT -p tcp --dport 22 --syn -m state --state NEW -j DROP_SYN_SSH
```

Le suivi de connexion (Conntrack) :
Netfilter permet de réaliser un suivi de connexion, c'est-à-dire qu'il est capable de savoir si un paquet est nouveau ou a un lien avec une connexion déjà établie.

NEW : nouvelle connexion

ESTABLISHED : connexion déjà établie
RELATED : la connexion a un lien avec une connexion déjà établie. Utile lors d'une connexion FTP à l'ouverture d'une nouvelle session pour le transfert des données.
INVALID : connexion qui n'a aucun lien avec les trois précédente.

Ce script est un exemple, il n'est bien sur pas optimisé. Il a seulement pour but de montrer les quelques opérations de base que l'on peut effectuer avec Netfilter qui possède beaucoup d'autres possibilités. Il suffit maintenant de lancer le script par la commande ./fw. N'oubliez pas de lancer ce script au démarrage de la machine pour être sur que le firewall est configuré après un redémarrage. Il est judicieux d'utiliser des outils comme NMAP, NESSUS, IDS SNORT pour vérifier la bonne configuration de votre firewall.

[Retour au menu]
