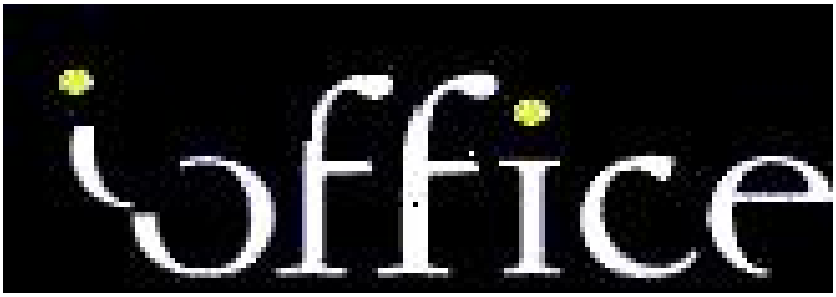


dmd@ioffice.fr

Administration et Sécurisation des Systèmes Linux & BSD[1]

DUPONT Sébastien
dupont_s@epita.fr

19 décembre 2002

[1]Documentation générée par LaTeXet sous licence GNU FDL (Free Documentation License) régulièrement mis a jour et disponible au format PDF en cliquant sur les liens suivants: miroir1, miroir2

TABLE DES MATIÈRES

CHAPITRE 1

Introduction

Le but de cette documentation[1] est de fournir les explications de base afin d'administrer et de sécuriser les systèmes Linux et *BSD nous nous baserons sur les distributions "OpenBSD 3.1", "Mandrake 8.2", "RedHat" et "Debian", mais cela devrait tout de même convenir pour les systèmes GNU/Linux et *BSD en général, les méthodes de configuration restant sensiblement les mêmes.
Pour le reste il est indispensable de considérer que la sécurité d'un système ne se limite pas à une sécurisation de chaques services, mais une sécurisation à chaques niveaux, en effet si le serveur est laissé dans un endroit non sécurisé il sera alors possible en le reboutant de la faire démarrer par un simple CD de boot et ainsi de modifié avec les droits root toutes les données souhaitées.
De la même façon si des comptes utilisateurs utilisent des mots de passe simple voir inexistant il sera possible par l'intermédiaire d'une attaque par dictionnaires "force brut" d'obtenir l'accès au compte. De plus il est indispensable de ne faire qu'une confiance relative au Firewall, qui resteront toujours une rustine. En effet, une récente faille de sécurité sur le client mail "mutt" a bien démontré cela.
De plus il existe certaines failles notamment sur Netfilter (iptables) qui peuvent être exploitées par simple détection de firewall. Pour conclure il ne sera jamais possible d'obtenir une sécurisation totale d'un serveur, néanmoins il sera toujours possible d'éviter un maximum de failles par une bonne administration d'une système et par une politique paranoïaque sur les services autorisés.

[1] Vous pourrez trouver une version de cette documentation régulièrement mise a jour en cliquant sur le lien suivant Administration & et sécurisation des systèmes Linux/BSD

CHAPITRE 2

Les différentes distributions Linux & BSD

FIG. 2.1: **Debian :** Système le plus “stable” des distributions linux, mais relativement difficile à administrer, il bénéficie d’une méthode de mise à jours très réputée (semblable a “urpmi” de Mandrake)

FIG. 2.2: **Mandrake Linux :** La seule distribution Française.... système “user friendly” utilisation la plus simplifiée possible pour l’utilisateur. bénéficie également dans très grand de nombre de paquetages, et d’une mise à jour très rapide (cooker). La Mandrake se base sur la technologie “RPM” développé par RedHat. La Mandrake est particulièrement recommandée si votre poste bénéficie de matériel récent néanmoins il faudra utiliser les outils de configuration avec modération si vous comptez l’utiliser en tant que serveur.

FIG. 2.3: **Suse :** Distribution Allemande. utilise le système Yast pour la configuration très critiqué car non GPL... un comble ! ! !

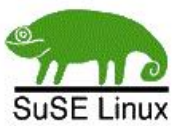
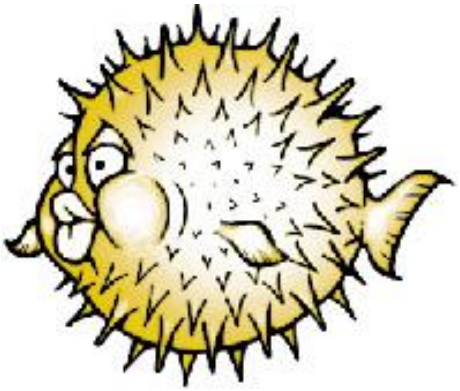


FIG. 2.4: **RedHat Linux :** Compromis stabilité/facilité le plus utilisé pour les serveurs réseaux GNU/Linux, dans les universités par exemple. Néanmoins celle ci ne disposant nativement de programme tel que urpmi de Mandrake ou diskdrake elle reste un peu moins agréable a administrer.

FIG. 2.5: **FreeBSD :** Très utilisée par les serveur car supporte les architectures multiprocesseurs. Relativement simple à administrer. Utilisé par “Yahoo” et par “hotmail” jusque récemment. Possède un grand nombre de paquetages.

FIG. 2.6: **NetBSD :** C’est le système qui supporte le plus grand nombre de plate-forme différentes. Relativement difficile à administrer car aucun outil de configuration Possède également un très grand nombre de paquetages.

FIG. 2.7: **OpenBSD :** Le système le plus sécurisé et le plus stable. Relativement difficile à administrer, possède moins de paquetages (encore que ...) car tous les paquetages subissent une audition complète et bien souvent des correctifs. C’est donc la distribution la plus recommandée si vous souhaitez faire un serveur réseau sécurisé. C’est également OpenBSD qui est le créateur de Openssh. liste des paquetages disponibles : http ://www.openbsd.org/3.1_packages/i386.html



CHAPITRE 3

Première Approche des distributions Linux & BSD

3.1 Manipulation du disque dur

LE disque dur se décompose en deux types de partition, les partitions primaires et les partitions logiques. les partitions primaires sont limitées au nombre de 4. Parmi ces partitions vous pouvez en désigner une en tant que partition étendue à l'intérieur de laquelle vous pouvez mettre des partions logiques (entre 15(SCSI) et 63(IDE)).

Partitions Primaires & Étendues

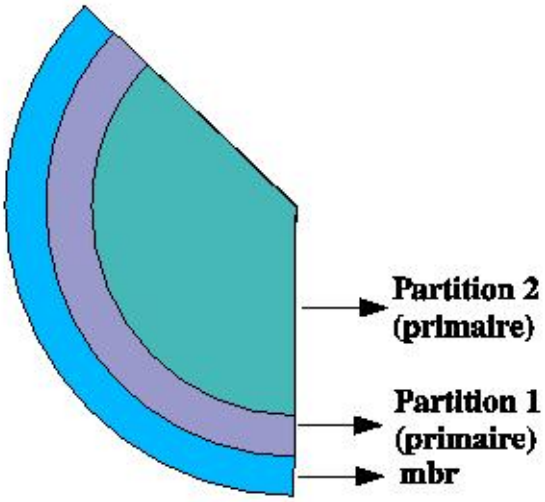
Il n'existe pas de recommandation spéciale par rapport au nombre de partitions primaires, néanmoins il en faudra au minimum une si vous souhaitez installer un système Windows ou *BSD, leur nombre peut donc varier entre 0 et 4. Lors de votre choix de partitionnement il est tout de même préférable d'y inclure une partition étendue afin de ne pas être rapidement limité par le nombre de partition. En ce qui concerne les systèmes Linux il peuvent s'installer aussi bien sur des partitions primaires que sur des partitions logiques, ce qui n'est malheureusement pas le cas des *BSD qui doivent impérativement se situer sur une partition primaire.

Partitions Logiques

Les partitions logiques, qui sont contenues par la partition étendue, sont beaucoup moins restrictives que les partitions primaires, en effet il est possible de créer sur les disques IDE jusqu'à 63 partitions et 15 pour les disques SCSI.

Le MBR (Master Boot Record)

Les premiers secteurs du disque physique sont occupés par le MBR (Master Boot Record). C'est une zone qui se situe avant la première partition physique, et qui est



très petite (512 octets seulement). Cette zone a un rôle vital pour le disque dur. Elle est partagée en deux parties :

La Table des Partitions
Ce sont des données. C'est là que sont enregistrées les informations concernant les quatre partitions primaires : emplacement, taille, propriétés (cachée, amorçable). Si on efface ces données, le PC ne sait plus où sont situées les partitions, et toutes les données sont donc perdues, bien que les données elles même n'aient pas été effacées. C'est de cette table des partitions que vient la limite des 4 partitions primaires possibles.

Le système d'amorçage
C'est un programme exécuté par le BIOS au démarrage du PC qui a pour rôle de lancer le système d'exploitation du PC. Pour cela, le programme stocké dans le MBR exécute en réalité le secteur boot de la partition principale où est installé le système d'exploitation à démarrer.

Parfois, sur cette zone, des programmes autres que celui par défaut sont installés. C'est par exemple le cas de LILO (Linux Loader) et de System Commander. Ces programmes laissent à l'utilisateur le choix du système à démarrer. Il se peut que ces programmes soient trop volumineux pour être entièrement copié dans le MBR, comme c'est le cas pour System Commander. Dans ce cas, le MBR peut démarrer le gestionnaire qui est alors stocké sur une partition primaire sous forme de fichiers. (SYSCOM.SYS pour system commander). Il est alors nécessaire que la partition soit accessible.

Si vous voulez réinstaller le système par défaut dans le MBR, vous devez taper la

commande FDISK /MBR sous DOS. Ceci remettra le système d'amorçage par défaut, sans pour autant effacer la table des partitions. Ceci sert par exemple à désactiver un gestionnaire d'amorçage comme LILO du MBR, où à supprimer un virus qui est situé dans le programme d'amorçage du MBR.

Le système d'amorçage par défaut, celui qui est réinstallé quand on tape FDISK /MBR, a pour rôle d'exécuter le secteur boot de la partition ACTIVE. (Voir propriétés d'une partition).

Structure du secteur de partition principal

Adresse (hexa)	Contenu	Type
000	Programme de partition	Code 446 o max.
1BE	1ère entrée dans la table de partition	16 octets
1CE	2ème entrée dans la table de partition	16 octets
1DE	3ème entrée dans la table de partition	16 octets
1EE	4ème entrée dans la table de partition	16 octets
1FE	AA55 (code d'identification)	2 octets

On constate ici qu'il ne peut pas y avoir plus de 4 partitions primaires.

Structure d'une entrée dans la table de partition

Adresse (hexa)	Contenu	Type
00	État de la partition : - 00 : partition non active - 80 : partition active	1 octet
01	Tête où commence la partition	1 octet
02	Secteur et cylindre où commence la partition	2 octet
04	Type de partition	1 octet
05	Tête où finit la partition	1 octet
06	Secteur et cylindre où finit la partition	2 octets
08	Distance en secteurs entre secteur de partition et secteur de boot de la partition	4 octets
0C	Nombre de secteurs de la partition	4 octets

Une partition étendue se compose d'une nouvelle table de partition, de structure identique à celle de la table principale, permettant ainsi un chaînage vers des partitions supplémentaires "encapsulées" à la suite.

Type de partition

0 Vide	1c Win95 FAT32 cac	65 Novell Netware	bb Boot Wizard hid
1 FAT12	1e Win95 FAT16 cac	70 DiskSecure Mult	c1 DRDOS/sec (FAT-
2 root XENIX	24 NEC DOS	75 PC/IX	c4 DRDOS/sec (FAT-
3 util XENIX	39 Plan 9	80 Old Minix	c6 DRDOS/sec (FAT-
4 FAT16 <32M	3c Reprise Partiti	81 Minix / ancien	c7 Syrinx
5 Étendue	40 Venix 80286	82 Échange Linux	da Non-FS data
6 FAT16	41 Amorce PPC PReP	83 Linux	db CP/M / CTOS / .
7 HPFS/NTFS	42 SFS	84 Lecteur C : cach	de Dell Utility
8 AIX	4d QNX4.x	85 Linux étendu	df BootIt
9 AIX amorcable	4e QNX4.x 2me par	86 Ensemble de vol	e1 Accès DOS
a Gestionnaire d'	4f QNX4.x 3me par	87 Ensemble de vol	e3 R/O DOS
b Win95 FAT32	50 OnTrack DM	8e Linux LVM	e4 SpeedStor
c Win95 FAT32 (LB	51 OnTrack DM6 Aux	93 Amoeba	eb BeOS fs
e Win95 FAT16 (LB	52 CP/M	94 Amoeba BBT	ee EFI GPT
f Win95 Etdue (LB	53 OnTrack DM6 Aux	9f BSD/OS	ef EFI (FAT-12/16/
10 OPUS	54 OnTrackDM6	a0 Hibernation Thi	f0 Linux/PA-RISC b
11 FAT12 caché	55 EZ-Drive	a5 FreeBSD	f1 SpeedStor
12 Diagnostics Com	56 Golden Bow	a6 OpenBSD	f4 SpeedStor
14 FAT16 caché <32	5c Priam Edisk	a7 NeXTSTEP	f2 DOS secondaire
16 FAT16 caché	61 SpeedStor	a9 NetBSD	fd Detection auto
17 HPFS/NTFS caché	63 GNU HURD ou Sys	b7 BSDI fs	fe LANstep
18 AST SmartSleep	64 Novell Netware	b8 Swap BSDI	ff BBT
1b Win95 FAT32 cac			

Secteur boot d'une partition

Chaque partition contient un secteur de boot sur ses premiers secteurs. Le secteur de boot est à la partition, ce que le MBR est au disque dur : Elle contient deux parties :

1. Des données qui indiquent les propriétés de la partition, comme par exemple le nom de volume, ou le système de fichiers.
2. Un programme d'amorçage qui démarre le système d'exploitation. Si la partition contient un système d'exploitation, le programme stocké sur le secteur de boot sert à le démarrer. Le secteur de boot doit être exécuté par le programme du MBR. Si la partition n'est pas amorçable, elle contient généralement un programme qui affiche un message d'erreur comme "Pas de système d'exploitation", au cas où cette partition serait exécutée.

3.2 Système de fichiers.

Le système de fichier *BSD porte le nom de ffs (fast file system), c'est un système de fichier parfaitement stable mais qui fait preuve de lenteur par rapport à ceux utilisés sous Linux (ext2, ext3, reiserfs ....)

Systèmes de Fichiers Journalisés

La plupart des systèmes de fichiers modernes utilisent des techniques de journalisation empruntées au monde des bases de données pour améliorer la reprise sur incident. Les transactions sont écrites séquentiellement sur une partie du disque appelée /journal/ ou /log/ avant de les écrire sur disque à leur place définitive dans le système de fichiers. Les mises en oeuvres varient dans le contenu écrit dans le journal. Certaines n'y mettent que les meta-données, alors que d'autres y enregistrent toutes les écritures.

Maintenant, si un plantage survient avant qu'une entrée de journal ne soit validée, les données originales sont toujours sur disque et vous avez uniquement perdu vos dernières modifications. Si le plantage intervient pendant l'update (i.e. après la validation de l'entrée du journal), l'entrée du journal reflète ce qui aurait dû être fait. Et au réamorçage du système, il n'a qu'a rejouer les transactions du journal, et terminer l'*update* qui a été interrompu.

Dans les deux cas, vous avez des données valides, et non une partition inutilisable. Et comme le temps de reprise sur incident associé à cette approche journalisée, le système est utilisable dans les quelques secondes qui suivent.

Il est aussi important de noter que l'utilisation d'un système de fichiers jounalisé ne rend pas complètement obsolète l'emploi d'un programme de vérification d'intégrité ("fsck"). Des erreurs matérielles ou logicielles peuvent corrompre de façon aléatoire quelques blocs du système de fichiers, interdisant une récupération grâce au journal de transactions.

Les principaux systèmes de fichiers journalisé

ext3fs Aujourd'hui intégré dans les principales distributions Linux.
Avantage : supporte les montages en ext2fs.

Reiserfs Systèmes de fichiers performant.
Inconvénient : impossibilité de convertir une partition existante.

JFS d'IBM Systèmes de fichiers particulièrement adapté aux installations de tailles importantes.
Inconvénient : trop de dépendance envers AIX.

XFS de silicon Graphics Supporte les très gros systèmes de fichiers 18 millions de Tera. Inconvénient : trop de dépendance envers IRIX.

Conclusion

Pour plus de tranquillité de ce point de vue, il est préférable d'utiliser les systèmes de fichiers proposés par défaut. donc ext2fs pour les anciennes versions, et ext3fs depuis la Mandrake8.2 et la RedHat 7.2.

Remarque

Les systèmes de fichiers Windows que sont msdos, fat32 (vfat), et NTFS sont parfaitement supportés sous Linux. néanmoins NTFS reste par défaut accessibles en lecture seul (sauf si recompilation noyau.).

3.3 Installation d'un système *BSD : OpenBSD

Les systèmes *BSD font parti des systèmes les plus complexes à installer particulièrement NetBSD et OpenBSD car d'une part ils s'adressent a des utilisateurs ex-

périmentés et d'autre part les développeurs de ces systèmes préfèrent travailler leur stabilité et leurs fonctionnalitées que leur accessibilité.

De plus en dehors de FreeBSD, lors de l'installation il ne sera possible d'installer qu'un système minimal sans packages supplémentaires.

Il existe des documentations sur l'installation des différents *BSD, que vous pourrez consulter en cliquant sur les liens suivants :

- Documentation sur l'installation d'OpenBSD
- Documentation sur l'installation de NetBSD
- Documentation sur l'installation de FreeBSD

3.4 Utilisations du SHELL

Il existe une dizaine de shells différents, les plus connus sont :

sh	shell de base avec aucune fonctionnalité avancé
ksh	shell plus évolué que sh mais reste restreint
bash	shell le plus utilisé car très rapide et possède des fonctionnalitées tel que la complétions, etc..
zsh	le shell le plus évolué, de plus en plus utilisé car il possède des fonctionnalitées très pratique tel que : la complétions de la suite de chaque commandes par ex : rpm -i apac + [tab] va directement compléter par rpm -i apache....rpm

les shells disposent de "variables d'environnement" tel que le PATH, le PROMPT mais aussi des alias, etc...

voici un exemple du contenu d'un .zshrc :

```
export PROMPT="....."
alias petra='ssh root@petra'
alias e='emacsclient -n'
export PATH=/usr/local/bin :/sbin/ :/usr/X11R6/bin :/bin
```

la variable "PROMPT" permet de personnaliser l'apparence de la ligne de commande.
la variable "PATH" permet d'inclure des répertoires de recherche pour trouver les fichiers exécutables.

exemples :

```
(seb@ramses)[ ]-% sudo vipw
sudo : vipw : command not found
(seb@ramses)[ ]-% export PATH=$PATH :/usr/sbin
(seb@ramses)[ ]-% sudo vipw
root :x :0 :0 :root :/root :/bin/zsh
bin :x :1 :1 :bin :/bin :
daemon :x :2 :2 :daemon :/sbin :
adm :x :3 :4 :adm :/var/adm :
lp :x :4 :7 :lp :/var/spool/lpd :
sync :x :5 :0 :sync :/sbin :/bin/sync
shutdown :x :6 :0 :shutdown :/sbin :/sbin/shutdown
....
```

Cet exemple met en valeur le fait que si la variable PATH venait a être modifiée on aurait de la même façon un problème de sécurité potentiel, en effet si quelqu'un venait á modifier le PATH par PATH=/tmp/bin :$PATH :/usr/sbin n'importe qui disposant des droits suffisant sur /tmp pourrait alors usurper toute les commandes exécuter par l'utilisateur, les conséquences pourrait être désastreuses pour le cas de root.

Remarques

Dans la variable PROMPT le caractère :

^[

est un caractère d'échappement. Il se construit sous Emacs par la suite de touches "(Ctrl+q) + ESC"

3.5 Les Éditeurs vi et Emacs

Éditeur VI

vi est un des éditeurs les plus léger. il est en général un alias de vim sous Linux.

Avantages

vi (vim) permet la coloration syntaxique de nombreux fichiers de configuration. indispensable en environnement limité. extrêmement léger.

Inconvénients

utilisation un peu moins naturelle que sous Emacs.

commandes de base

:w	sauve.
:wq	sauve et quitte.
:q !	quitte sans sauver.
i	mode insertion(édition) .
o	passe une ligne passe en mode insertion.
ESC	quitte le mode insertion.
/	recherche.
$	find de ligne.
u	undo.

Éditeur Emacs

Avantages

personnalisable à volonté. permet des macros extrêmement puissantes. intégrations complète de certaines commandes telles que diff, cvs, jeux, ispell etc .... intégration du reader de mail/news gnus connu comme les plus performant des clients mail.

Inconvénients

Plus lourd et donc plus long à lancer que vi.
configuration par défaut généralement limitée.

commandes de base

Ctrl+x+s	sauve.
Ctrl+x+f	ouvrir.
Ctrl+x+c	quitte.
Ctrl+w	“couper” le texte sélectionné.
Ctrl+y	“coller”
Alt+w	“copier”.
Ctrl+s	“recherche”.
Ctrl+Shift+/	undo.
Alt+Shift+/	complétions.

configuration

les fichiers de configuration de Emacs sont /.emacs et /.Xdefaults.

Ex : /.emacs

(global-font-lock-mode t)	Permet la colorisation syntaxique
(transient-mark-mode t)	Permet la mise en surbrillance de la zone sélectionnée
(global-set-key [f1] ’buffer-menu)	permet d’associer une touche à une action.

Ex : /.Xdefaults

emacs*verticalScrollBars :	off	désactivation de la barre de défilement
emacs*menuBar :	off	désactivation de la barre de menu
emacs*toolBar :	off	désactivation de la barre des options
emacs*foreground :	white	couleur de la police
emacs*background :	black	couleur du fond
emacs*bitmapIcon :	true	
emacs*popup*Font :	7x13	taille de la fonte des popup
emacs*popup*Foreground :	wheat	
emacs*popup*Background :	DarkSlateBlue	
emacs*menubar*Font :	7x13	
emacs*menubar*Foreground :	wheat	
emacs*menubar*Background :	DarkSlateBlue	
emacs*dialog*Foreground :	wheat	
emacs*dialog*Background :	DarkSlateBlue	
emacs*geometry :	80x58+160+50	

3.6 Commandes de bases :

3.6.1 les aides.

man

man est la commande la plus utile du système car elle permet de connaître toutes les options possibles de chaques commandes.
Ex : *man ls*
On notera que les man sont toujours mis jour sous les différents BSD ce qui pas toujours le cas sous linux.

info

Semblable au man mais est aujourd'hui le format officiel de documentation sous Linux.

whatis

Comme son nom l'indique permet d'avoir une description succincte sur un programme.

HOWTO

le Howto est une particularité linux, qui explique comment installer ou configurer certaines programmes, fonctionnalitées ou services.
On trouve généralement ces documentations au format HTML dans */usr/share/doc/HOWTO*

3.6.2 Manipulation des fichiers et des répertoires.

Dans les systèmes de fichier Linux/Unix tous les fichiers sont définis selon trois types de droits :

- les droits attribués au propriétaire du fichier sur le fichier,
- les droits attribués au groupe du fichier sur le fichier,
- et enfin les droits attribués pour tous les autres sur le fichier.

Ensuite pour chacun de ces type on attribuera de la même façon trois types de droits :

- les droits de lecture. (=4 ou r pour read)
- les droits d'écriture. (=2 ou w pour write)
- et enfin les droits d'execution. (=1 ou x pour exécution)

Ainsi lors d'un "ls -l /bin/ls" on pourra voir apparaître la ligne :

```
-rwxr-xr– 1 seb users 15 Aug 15 14 :02 hello_world.pl
```

Nous allons donc décrire la signification de chacun des champs significatifs de cette ligne :

- **rwxr-xr–** Sur ce premier champ apparaissent les droits du fichier que l'on pourra décomposer de la façon suivante :
  - **rwx pour le propriétaire :**
    - r : le propriétaire pourra lire ce fichier.
    - w : le propriétaire pourra modifier ce fichier.
    - x : le propriétaire pourra exécuter ce fichier.
  - **r-x pour le groupe :**
    - r : le groupe pourra lire ce fichier.
    - - : le groupe ne pourra pas modifier ce fichier.
    - x : le groupe exécuter ce fichier.
  - **r– pour les autres :**
    - r : les autres pourront lire ce fichier.
    - - : les autres ne pourront pas modifier ce fichier.
    - - : les autres ne pourront pas exécuter ce fichier.
- **seb** le propriétaire du fichier.
- **users** le groupe du fichier.
- **15 Aug 14 :02** date de création du fichier.
- **hello_world.pl** nom du fichier.

ls

ls permet de visualiser les fichiers contenus dans le répertoire courant. Les options usuelles sont :

-l	affichages de toutes les informations sur le fichier.
-R	affiche également les fichiers des sous répertoires.
-a	affiche les fichiers commençant par un “.” ie : “cachés”
-h	human readable
-F	affiche les nom des répertoires avec “/” aprés leurs noms.
	affiche les nom des fichiers exécutables avec “*” aprés leurs noms.
	affiche les nom des liens avec “&” aprés leurs noms.

cd

cd : “change directory” ie : changer de répertoire
cd ou cd : permet d’accéder au répertoire de l’utilisateur
cd - : permet de retourner au précèdent répertoire.

mv

“move” signifie déplacer ou renommer
mv toto.txt toto2.txt : le fichier toto.txt sera renommer en toto2.txt
mv toto.txt tmp/ : le fichier toto.txt sera déplacé dans le répertoire tmp/

pwd

affiche le répertoire courant.

mkdir

création d’un répertoire.
mkdir répertoire
mkdir -p /home/seb/tmp/devel/test permet de créer les répertoires récursivement.

rmdir

suppression d’un répertoire s’il est vide si celui n’est pas vide il faut utiliser la commande :
rm -rf répertoire

chmod

permet de changer les permissions sur un fichier ou sur répertoire.
Chaque chiffre est interprété de la même façon :

1 pour l’exécution
2 pour l’écriture
3 pour la lecture

On peut cumuler les droits en ajoutant ces chiffres. Ainsi, si le premier chiffre du paramètre est 6, il correspond alors à 4+2 soit la lecture(4) et l’écriture

chmod 777 fichier : droits de lecture, d'écriture, d'exécution pour tout le monde. (dangereux).
chmod +s fichier : positionne le bit SUID a un, c'est a dire que si un utilisateur a suffisamment de droits pour pouvoir exécuter ce programme alors le système considérera que c'est le propriétaire du fichier qui l'aura exécuté.

C'est l'une des failles de sécurité les plus exploitées par les crackers, Exemple : si root utilise la commande *chmod +s /bin/cat* : Alors n'importe quel utilisateur pourra afficher Le contenu de n'importe quel fichier du système :
cat /etc/shadow

chown
permet de changer l'utilisateur et le groupe propriétaire d'un fichier :
Ex : *chown seb.users fichier*
"fichier" appartiendra désormais à l'utilisateur seb et au group users. il est également possible de l'appliquer récursivement avec l'option "-R" :
Ex : *chown seb.users rep*

du
La commande "du" permet entre autres de connaître la taille d'un répertoire par blocs d'un Ko. Néanmoins il est possible[1] d'y ajouter l'option "-h"[2] pour rendre le résultat plus explicite.

3.6.3 Les outils de recherche.

find
permet de trouver un fichier a partir d'un répertoire donné :

find / -name "httpd.conf" permet de retrouver le fichier httpd.conf en cherchant dans tous les répertoires depuis la racine.
find / -user root -a (perm -4000 -o -perm -2000) print cette commande permet d'afficher tous les fichiers ayant les droits SUID et appartenant a root du système.

locate
beaucoup plus rapide que find, car il utilise une base de donnée pour trouver un fichier.
Pour mettre a jour cette base de donnée il suffit en tant que root de lancer la commande :

[1] dans la majorité des systèmes
[2] human readable, ie compréhensible par un humain soit en Ko, Mo, Go

updatedb sous Linux
/usr/libexec/locate.updatedb sous bsd

pour chercher un fichier il suffit donc de taper :
locate httpd.conf pour en trouver l'occurrence.
PS : se programme se situe dans le rpm slocate

grep
Permet entre autre de rechercher une chaine de caractère dans un fichier :
grep toto fichier.txt
grep -r toto * pour faire une recherche en incluant les fichiers des sous répertoires.
grep permet aussi de faire une recherche depuis la sortie standart :
ls |grep toto dans cet exemple nous recherchons un ficher dans le répertoire courant dont le nom contient "toto".
Aussi on pourra lui demander de n'afficher que les résultats qui ne correspondent pas la recherche demandée avec l'option "-v" :
ls |grep -v toto dans cet exemple nous recherchons un ficher dans le répertoire courant dont le nom ne contient pas "toto".

3.6.4 Commandes sur les processus.

top
montre les principaux processus du système.
utilisation : *$> top*

ps
utilitaire permettant de voir les processus actifs du système.
utilisation : *$> ps -ax*
Remarque :
Ces commandes permettent de connaître le pid des processus, ce qui permet de les utiliser en complément de "kill".

kill
commande qui permet d'envoyer un signal à un processus :

1	HUP (hang up)	Re-démarrage
2	INT (interrupt)	
3	QUIT (quit)	
6	ABRT (abort)	
9	KILL (non-catchable, non-ignorable kill)	force l'arrêt
14	ALRM (alarm clock)	
15	TERM (software termination signal)	

exemple : *kill -9 pid_processus*

3.6.5 Commandes sur les périphériques.

df

df (display free disk space), permet de connaître l'état d'utilisation d'une partition montée. on utilise généralement df avec l'option "-h" ("Human-readable" output) pour afficher ces statistiques en Byte, Kilobyte, Megabyte, Gigabyte, Terabyte, Petabyte, Exabyte, plutôt qu'en inode moins facile à estimer.

mount/umount

permet de monter et de démonter des partitions :
Ex : *mount -t ext2 /dev/hda7 /home*

-t ext2	cette option permet de spécifier le type de partition pour une partition Linux (ext2), Windows (vfat), dos (msdos), cdrom(iso9660) ....
/dev/hda7	numéro de la partition ou du volume
/home	répertoire dans lequel on va monter la partition c'est à dire le répertoire à partir duquel les fichiers seront accessibles.

Il possible d'ajouter des options supplémentaires tel que :

nosuid :	ne permet l'exécution des fichiers suid dans cette partition ou ce volume.
noexec :	ne permet aucune exécution.
...	

dd

dd - Convertir un fichier en le copiant.

dd copie un fichier (par défaut, depuis l'entrée standard vers la sortie standard) en permettant de sélectionner la taille de bloc, et d'effectuer des conversions.

Il lit son entrée bloc par bloc, en utilisant la taille des blocs d'entrée mentionnée (par défaut 512 octets). Si l'option bs=octets est présente, et si aucune autre conversion que sync, noerror ou notrunc n'est indiquée, il écrit la quantité de données lues (qui peut être plus petite que celle demandée) dans un bloc de sortie indépendant. Ce bloc a exactement la même taille que les données lues, sauf si la conversion sync a été réclamée, auquel cas les données sont complétées avec des octets nuls, ou des espaces (voir plus bas).

Options

if=fichier	Lire les données depuis le fichier indiqué plutôt que depuis l'entrée standard.
of=fichier	Ecrire les données dans le fichier mentionné, et non pas sur la sortie standard. Si conv=notrunc n'est pas indiqué, le fichier est initialement tronqué à la taille spécifiée par seek= (0 octets si seek= n'est pas fourni).
ibs=nombre	Lire le nombre indiqué d'octets en une fois. Par défaut 512.
obs=nombre	Écrire le nombre indiqué d'octets en une fois. Par défaut 512.
bs=nombre	Lire et écrire le nombre indiqué d'octets en une fois. A priorité sur ibs et obs. (et indiquer bs n'est pas équivalent à indiquer la même valeur pour ibs et obs du moins lorsqu'aucune autre conversion que sync, noerror et notrunc n'est indiquée, car cela indique que chaque bloc d'entrée doit être copié dans un bloc de sortie indépendant, sans regrouper les blocs plus courts).
cbs=nombre	Indique la taille des blocs pour les conversion block et unblock.
skip=nombre	Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de la lecture.
seek=nombre	Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de l'écriture.
count=nombre	Copier seulement le nombre indiqué de blocs (dont la taille est fournie par ibs), et non pas tout jusqu'à la fin du fichier.

3.6.6 Commandes de connexion distance.

ssh

commande qui permet de se connecter de façon sécurisé (les données qui transitent seront cryptées entre les deux machines).
Ex : *ssh toto@192.168.0.1*

scp
commande qui permet de copier de façon sécurisé un fichier ou un répertoire sur le compte d'une autre machine.
Ex : *scp fichier toto@192.168.0.1 :.*
scp -r répertoire toto@192.168.0.1 :.

3.6.7 Autres commandes système.

tar et gzip
tar est un utilitaire qui permet de concaténer plusieurs fichiers en un seul, mais ne les compressent pas. C'est pourquoi il est souvent utilisé en parallèle de gzip.

Ainsi pour concaténer plusieurs fichiers en un seul il suffit de taper :
"tar cvf fichier.tar répertoire"
ou "tar cvfz fichier.tar.gz répertoire" si l'on souhaite en plus compresser ces fichiers

De même pour décompacter ce fichier on tapera :
"tar xvf fichier.tar" dans le cas d'un fichier avec une extension tar
"tar xvfz fichier.tar.gz" dans le cas d'un fichier avec une extension tar.gz.

chroot
permet de déplacer la racine d'un système.
Ex : *chroot répertoire*
NOTA : Cette commande est très utilisées pour sécurisé un serveur. En effet lors d'attaque sur un serveur ftp ou http, une personne mal intentionnée n'aura aucun moyen de passer root (ce qui est le pire des cas) sur le système en entier, il sera limité au répertoire chrooté.

vipw
commande utilisé pour modifié les attributs de chaques utilisateurs :

seb :$2a$06$M :1000 :10 : :0 :0 : :/home/seb :/bin/zsh

on trouvera dans l'ordre :
- le login
- le mot de passe crypté
- uid : numéro de l'utilisateur
- gid : numéro du groupe
- ...

- répertoire HOME de l'utilisateur
- et enfin shell par défaut pour l'utilisateur.

CHAPITRE 4

Administration des Système Linux & *BSD

4.1 Les différents répertoires.

Une autre particularité intéressante du monde Unix/Linux est la hiérarchisation des différents répertoires.

/	racine
/bin	commandes de base
/dev	contient tous les liens vers les périphériques
/sbin	contient toutes les commandes réservées a l'administrateur.
/usr	contient tous les programmes
/var	contient les logs/pages web/ mail ...
/tmp	accessible en lecture/écriture pour tous le monde
/etc	contient tous les fichiers de configuration
/lib	contient les librairies de base
/home	contient toutes les données utilisateur
	De plus pour Linux :
/proc	image du noyau et des périphériques installés
/boot	contient le kernel ainsi que la procédure de démarrage

4.2 Commandes d'administration

Par convention il est préférable de ne pas modifier les fichiers /etc/passwd et /etc/master.passwd manuellement, de plus pour ce faire il existe une pléiade d'utilitaires.

4.2.1 utilisateurs et groupes

Utilisateurs

Pour ajouter des utilisateurs il existe deux comandes : “useradd” et “groupadd” Nous utiliserons ici la commande “useradd” (identiques sous Linux ou BSD)

```
useradd   [-c commentaire] [-d rep_perso]
          [-e date_expiration] [-f inactive_time]
          [-g groupe_initial] [-G groupe[,...]]
          [-m [-k rep_squelette] | -M] [-p mot_de_passe]
          [-s shell] [-u uid [ -o]] [-n] [-r] login
```

Il est également possible de changer les variables par défaut avec la commande “useradd -D”

```
useradd -D  [-g groupe_defaut] [-b rep_perso_defaut]
            [-f inactivite] [-e date_expiration_defaut]
            [-s shell_defaut]
```

De même que pour les variables par défaut il est possible de modifier les fichiers squelettes, c’est à dire les fichiers qui sont placés dans le répertoire $HOME durant la création du compte.
Ces fichiers sont placés dans le répertoire “/etc/skel”

Groupes

De la même façon pour ajouter un groupe il existe la commande “groupadd”

4.2.2 Configuration Réseau

La commande la plus importante de configuration réseau est “ifconfig” en effet elle permet :

ifconfig	-a	affiche la liste des périphériques réseaux
ifconfig	eth0 192.168.1.10 netmask 255.255.255.0	configure la carte “eth0” avec l’adresse IP 192.168.1.10 et avec le netmask 255.255.255.0
ifconfig	eth0	affiche la configuration de la carte “eth0”

Remarque :

Sur les systèmes Linux les cartes réseaux sont référencés par eth0,eth1,... , alors que sur les systèmes BSD elles sont référencés par leurs noms de driver, par exemple pour

les cartes 3com le système les désignera par xl0,xl1,... pour les DLink par leur driver via-rhine vr0,vr1,...

4.3 Étapes du démarrage d’un système *BSD

Sur les différents *BSD les étapes de démarrage sont définis dans deux fichiers principaux, il s’agit de /etc/rc.conf et /etc/rc.local.

Typiquement il faudra simplement répondre aux questions posées dans /etc/rc.conf et ajouter des commandes ou des scripts personnels dans /etc/rc.local.

4.4 Étapes du démarrage d’un système Linux

Nous verrons dans cette section quelles sont les différentes étapes de démarrage des *BSD.

Ainsi durant le processus de boot on passera par les étapes suivantes :

1. Après le chargement du “MBR” lilo va donner la main au kernel.
2. Le kernel s’initialise, accède au "initial ram disk" et donne la main au chargeur initrd, qui va permettre de charger des modules supplémentaires avant d’accéder au système (le déroulement de cette étape est décrit dans le fichier linuxrc). Au démarrage du noyau, celui-ci décompresse et monte le contenu du fichier initrd.img sur le disque virtuel /dev/ram0. Pour accéder á ce pseudo système de fichiers on peut procéder de la façon suivante :
   (a) cp /boot/initrd.img /tmp
   (b) cd /tmp
   (c) mv initrd.img initrd.gz
   (d) gunzip initrd.gz
   (e) mkdir img
   (f) mount -t ext2 initrd img -o loop=/dev/loop0
   (g) cd img
   (h) cat linuxrc
3. Après l’initialisation des modules, Linux accède à la partition racine.
   à la limite, un système Linux n’est composé que d’un kernel et d’un interpréteur de commande (/bin/sh) : Un gros DOS ! Un micro système *BSD (un BSD sur une disquette) aura généralement un petit script /etc/rc qui fera un initialisation minimum et ensuite lancera la tâche principale.

4. /sbin/init prend la relève

   Le travail initial du kernel, c'est de préparer l'accès au système de fichier et de donner la main à /sbin/init. Cet utilitaire est présent sur tout les systèmes Unix.

   à partir du fichier /etc/inittab, /sbin/init va démarrer les différents processus nécessaire à l'opération d'un système multi-usagers.

5. /etc/inittab

   Le fichier /etc/inittab a un format relativement simple. Outre des commentaires (ligne débutant par le caractère #), il contient une commande par ligne. Chaque ligne a le format suivant :

   (a) Identificateur C'est généralement une séquence de deux lettres identifiant uniquement cette ligne. /sbin/init utilise cet identificateur lorsqu'il rapporte des erreurs.

   (b) Runlevels

   Chaque commande est associé à un ou plusieurs groupes appelé "run levels". Les "run levels" sont identifiés par un numéro de 0 à 6. La définition exacte de chaque niveau peut varier d'une distribution à l'autre.

   Les 7 niveaux d'exécution classiques :

   0 halte
   1 mode utilisateur unique
   2 mode multi-utilisateurs sans gestion NFS
   3 mode multi-utilisateurs
   4 sans effet
   5 mode X11
   6 Re-démarrage

   (c) Attribut

   Modes d'exécution (liste non exhaustive) :

once	la commande n'est exécutée qu'une fois
off	la commande n'est pas exécutée
wait	attente de l'exécution de la commande
boot	la commande doit s'exécuter durant l'amorçage, le niveau est ignoré
bootwait	idem mais le boot ne se poursuit qu'après la fin de la commande
ctraltdel	la commande est appelée lors de l'appui sur Ctrl+Alt+Del
initdefault	mode d'exécution de Linux

   Remarque :
   Dans le fichier */etc/inittab* on trouvera la ligne suivante :

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
```

```
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
```

C'est donc cette ligne qu'il faudra modifier si l'on veut démarrer en mode graphique(5) ou en mode console(3) comme c'est le cas dans cet exemple.

Ainsi si votre niveau de démarrage est le niveau 3, il vous faudra modifier le répertoire /etc/rc3.d afin d'ajouter ou de supprimer des services :

```
(seb@kset)[~]-% ls /etc/rc3.d
K15httpd       K35vncserver  K65identd   S08ip6tables  S14nfslock   S56rawdevices  S95anacron
K15postgresql  K45arpwatch   K70aep1000  S08ipchains   S17keytable  S56xinetd      S95atd
K16rarpd       K45named      K70bcm5820  S08iptables   S20random    S60lpd         S97rhnsd
K20nfs         K46radvd      K74ntpd     S09isdn       S25netfs     S80sendmail    S98wine
K25squid       K50snmpd      K74ypserv   S10network    S26apmd      S85gpm         S99local
K34yppasswdd   K50snmptrapd  K74ypxfrd   S12syslog     S28autofs    S90crond
K35smb         K50tux        S05kudzu    S13portmap    S55ssh\index{ssh}d      S90xfs
```

Pour le service httpd par exemple on observe que celui commence par un K, qui détermine si le service est activé (S) ou non (K) (Start ou Kill), suivit par un chiffre "15" qui précise le niveau d'ordre d'exécution (ici le service network et activé avant portmap). Pour le reste si le nom ne vous parait pas assez explicite vous pouvez par la commande "ls -l" voir sur quel fichier il pointe, dans le cas "K15httpd" on aura donc le fichier :

```
lrwxrwxrwx 1 root root 15 mai 11 16 :25 /etc/rc3.d/K15httpd -> ../init.d/httpd
```

De même il est possible de visualiser l'ensemble des services disponibles selon le runlevel par la commande "chkconfig –list".

Remarque

on peut très bien faire un mode "rescue" qui se contentera d'ouvrir un shell au démarrage, pour cela il suffira d'ajouter à */etc/lilo.conf* :

```
image=/boot/vmlinuz-2.4.18
label=maintenance
root=/dev/hda5
append="init=/bin/zsh"
read-only
```

Après quoi il suffira d"exécuter la commande "lilo" pour que les changements soient pris en compte au prochain démarrage.

4.5 {Dés}installation de programmes sous Linux/BSD

4.5.1 {Dés}installation de programmes sous Linux

la commande rpm

(Gestionnaire de Paquetages Red Hat) rpm est un puissant gestionnaire de paquetages, qui peut être utilisé pour construire, installer, interroger, vérifier, mettre à jour, et désinstaller des paquetages de logiciels individuels. Un paquetage consiste en une archive de fichiers, et en de l'information sur ce paquetage, incluant le nom, la version et la description du paquetage.

rpm -ivh paquetage.rpm	installation du paquetage.
rpm -Uvh paquetage.rpm	mise à du paquetage.
rpm -ql paquetage	listes des fichiers qui seront installés par le paquetage.
rpm -e paquetage	désinstallation du paquetage.
rpm -qa	liste de tous les paquetages installés.
Options disponibles :	
–force	force l'installation du paquetage.
–nodeps	ne vérifie pas les dépendances du paquetage.

la commande apt-get de Debian

"apt-get install" installe les paquetages en gérant les dépendances. avec la plus solide des réputation en ce qui concerne les "upgrade".

apt-get install paquetage	installation du paquetage avec gestion des dépendances.
apt-get remove paquetage	désinstallation du paquetage avec gestion des dépendances.
apt-cache search fichier	permet de trouver le paquetage d'origine d'un fichier.
apt-get update	mise a jour de la liste des paquetages sur les différentes sources listées dans /etc/apt/sources.list
apt-get upgrade	sélectionne automatiquement tous les paquetages qui doivent être mis à jour selon les paquetages déjà installés et les paquetages listés dans les divers médias enregistrés.

la commande urpmi de Mandrake

urpmi installe les paquetages de la même façon que la commande rpm mais gère en plus les dépendances. On peut comparer son utilisation avec le "apt-get install" de Debian.

urpmi paquetage	installation du paquetage avec gestion des dépendances.
urpme paquetage	désinstallation du paquetage avec gestion des dépendances.
urpmf fichier	urpmf permet de trouver le paquetage d'origine d'un fichier.
Options disponibles :	
–force	force l'installation du paquetage.
–nodeps	ne vérifie pas les dépendances du paquetage.
–auto-select	sélectionne automatiquement tous les paquetages qui doivent être mis à jour selon les paquetages déjà installés et les paquetages listés dans les divers médias enregistrés.

4.5.2 {Dés}installation de programmes sous *BSD

Il existe deux méthodes d'installation sous BSD (NetBSD, FreeBSD, OpenBSD)

Installation des paquetages pré-compilés

Tout comme pour les RPMS, il existe un utilitaire permettant d'installer directement les paquetages :

pkg_add paquetage.tgz	installation du paquetage avec gestion des dépendances.
pkg_delete paquetage	désinstallation du paquetage avec gestion des dépendances.
pkg_info paquetage	permet d'afficher les informations relatives au paquetage concerné.

De la même façon que pour les RPMS pkg_add va rechercher les dépendances dans le répertoire même où lancera l'installation.
A noter pour la désinstallation de paquetage il est préférable d'utiliser le shell **zsh** car il permettra une complétions du nom du paquetage tel qu'il est reconnu par le système. (indispensable s'il l'on ne veut pas passer son temps a rechercher le nom exact du paquetage.)

Installation des paquetages depuis les ports

La deuxième particularité des BSD est de pouvoir installer automatiquement les différents paquetages, dépendance comprises, depuis les sources, celles-ci se situent dans un fichier nommé "ports.tar.gz". NOTA : ce fichier contient à la fois les Makefiles et les patch réalisés par l'équipe des développeurs *BSD, par contre, ce fichier ne contient pas les sources proprement dites du paquetage mais elles seront téléchargées automatiquement. Vous l'aurez compris il sera préférable d'avoir une connexion "haut-débit" si vous souhaitez utiliser cette méthode d'installation...

Ainsi Il faudra dans un premier temps décompresser ce fichier avec la commande :

```
tar xvfz ports.tar.gz -C /usr

Ensuite si vous souhaitez installer le paquetage “zsh” vous devrez procéder
comme suit :

cd /usr/ports/shells/zsh
make
make install
```

NOTA : si vous ne souhaitez pas installer le paquetage mais simplement créer le fichier zsh.tgz (utilisable avec) **pkg_add** il faudra taper à la place de **make install** : “**make package**”
Vous pourrez ainsi récupérer le paquetage dans le répertoires :
/usr/ports/packetages/i386/All
ou encore dans le répertoire :
/usr/ports/packetages/i386/shells

4.6 Les fichiers de Configuration

4.6.1 /etc/rc.d/init.d/ sous Linux

le répertoire */etc/rc.d/init.d/* contient tous les scripts de démarrage des différents services.
Pour démarrer un service on lancera typiquement la commande :
/etc/rc.d/init.d/nom_service start
ou vous êtes déjà dans le répertoire */etc/rc.d/init.d/*
./nom_service start

bien sûr les possibilitées de ces scripts shell d’initialisation ne s’arrêtent pas a start : au minimum vous pourrez trouver start et stop, et en général vous trouverez :
start, stop, status, restart, condrestart, reload.

4.6.2 /etc/inetd.conf sous *BSD ou /etc/xinetd sous Linux

“inetd” (respectivement xinted) démarre les programmes fournissant des services Internet. Au lieu de démarrer ces services au moment de l’initialisation du système, et de les laisser inactifs jusqu’à ce qu’il y ait une demande de connexion, on ne démarre que inetd et celui ci écoute sur tous les ports nécessaires aux services listés dans ses

fichiers de configuration. Lorsqu'une requête arrive, inetd démarre le service correspondant. à cause de la façon dont il fonctionne, inetd (comme xinetd) est aussi appelé le super-serveur.

Les services listés dans les fichiers de configuration de inetd ou xinetd peuvent être séparés en 2 groupes. Les services du premier groupe sont dit multi-threaded et ils nécessitent que le processus père crée un nouveau processus pour chaque nouvelle demande de connexion. Chaque processus gère alors une connexion. Pour de tels services, inetd continue d'écouter pour prendre en charge de nouvelles demandes de connexions, et de lancer de nouveaux processus. D'un autre côté, le deuxième groupe est composé des services pour lesquels un seul démon prend en charge toute les nouvelles requêtes de connexion. De tels services sont appelés single-threaded et inetd ne prendra plus en charge les requêtes destinées à ce serveur tant que le serveur sera en service. Les services de cette catégorie sont habituellement de type datagramme.

Jusqu'à présent, la seule raison de l'existence d'un super-serveur était de préserver les ressources système en évitant de créer de nombreux processus dont la plupart ne seront actifs que très peu de temps. Tout en remplissant ces fonctions, inetd tire parti de l'idée du super-serveur pour ajouter des fonctionnalités telle que le contrôle d'accès et le logging. De plus, inetd ne se restreint pas aux services listés dans le fichier /etc/services. Par conséquent, tous le monde peut utiliser inetd pour démarrer des services personnalisés.

4.6.3 /etc/passwd

Passwd est un fichier de texte qui contient la liste des comptes sur le système, ainsi que des informations utiles sur ces comptes, comme l'identification de l'utilisateur, du groupe, le répertoire personnel, le shell,etc. Actuellement, il est de plus en plus recommandé d'utiliser des systèmes de masquage des mots de passe, comme shadow avec lequel le fichier /etc/passwd contient des "*" à la place des mots de passe, et où ces derniers sont stockés sous forme cryptée dans /etc/shadow qui n'est lisible que parle Super-utilisateur.

Il doit y avoir, dans le fichier des mots de passe, une ligne par utilisateur, avec le format suivant :

account :passwd :UID :GID :GECOS :directory :shell

Les divers champs sont les suivants :

account Le nom que l'utilisateur utilisera pour se connecter, il ne devrait normalement pas contenir de majuscules password La représentation encryptée (optionnelle) du mot de passe.

UID L'ID numérique de l'utilisateur.

GID L'ID numérique du groupe principal de l'utilisateur.

GECOS Ce champ est optionnel et n'a qu'un rôle informatif. Il contient généralement le nom complet de l'utilisateur.

directory Le répertoire de connexion de l'utilisateur (variable d' environnement $HOME).

shell Le programme à exécuter après la phase de connexion (par défaut /bin/sh). Si ce fichier n'existe pas, l'utilisateur ne pourra pas se connecter avec login(1).

4.6.4 /etc/master.passwd ou /etc/shadow

/etc/shadow contient les mots de passe cryptés des utilisateurs ou plus exactement, le résultat du cryptage d'une chaîne de longueur nulle avec le mot de passe comme clé de cryptage ainsi qu'éventuellement des informations sur l'âge des mots de passe :

- Nom de login.
- mot de passe crypté.
- Nombre de jours écoulés depuis le 1er janvier 1970 jusqu'au dernier changement de mot de passe
- Nombre de jours durant lesquels le mot de passe est encore valide Nombre de jours après lesquels le mot de passe doit être changé.
- Nombre de jours avant l'expiration du mot de passe impliquant l'avertissement de l'utilisateur
- Nombre de jours après l'expiration provoquant la désactivation du compte
- Numéro du jour depuis le 1er janvier 1970 á partir duquel le compte a été désactivé
- Champs réservé

Le champs mot de passe doit être rempli. Le mot de passe crypté comprend 13 à 24 caractères pris dans l'alphabet réduit a-z, A-Z, 0-9, . et /. Si le nombre minimum de jours requis est plus grand que le nombre maximum de jours de validé, ce mot de passe ne peut pas être changé par l'utilisateur. Un compte est considéré comme inactif et est désactivé si le mot de passe n'est pas changé dans l'intervalle spécifié après l'expiration du mot de passe. Un compte est également désactivé le jours indiqué quelque soit les autres informations d'expiration. Cette information est prioritaire sur toutes les autres champs présents dans /etc/passwd. Ce fichier ne doit pas être accessible en lecture par les utilisateurs normaux afin de maintenir la sécurité des mots de passe, en particuliers contre les attaques aux dictionnaires.

Remarque

Sur les systèmes *BSD le fichier */etc/shadow* est remplacé par le fichier */etc/master.passwd*

4.6.5 /etc/group

/etc/group est un fichier ASCII qui définit les groupes auxquels appartient chaque utilisateur.

Il y a une ligne par groupe, et chaque ligne a le format :
nom_du_groupe :mot_de_passe :GID :liste_utilisateurs

Les champs sont les suivants :

- **nom_du_group** Le nom du groupe.
- **mot_de_passe** Le mot de passe encrypté du groupe. Si ce champ est vide (presque toujours), aucun mot de passe n'est nécessaire.
- **GID** L'ID numérique du groupe.
- **liste_utilisateurs** Tous les noms des membres du groupe, séparés par des virgules.

4.6.6 /etc/fstab

Le fichier fstab contient des informations sur les différents systèmes de fichiers. fstab est uniquement lus par les programmes, jamais écrit. C'est la responsabilité de l'administrateur de créer et de maintenir ce fichier correctement. Chaque système de fichier est décrit sur une ligne indépendante. Les champs contenus sur chaque ligne sont séparés par des espaces ou des tabulations.

L'ordre des enregistrements au sein de fstab est important car fsck(8), mount(8), et umount(8) utilisent séquentiellement les enregistrements de fstab pour effectuer leurs travaux.

les lignes fstab se compose comme suit :

1. Le premier champ (fs_spec), décrit le périphérique bloc ou le système de fichiers distant a monter.
2. Le second champ (fs_file), indique le point de montage du système de fichier. Pour les partitions de swap ce champ doit être spécifié comme "none" (aucun).
3. Le troisième champ (fs_vfstype), décrit le type de système de fichiers. Le système supporte actuellement les types suivants (et peut-être d'autres, voir /proc/filesystems) :

   ext2 système de fichiers local, noms de fichiers longs, i-noeuds détaillés, et beaucoup d'autres avantages.

   ext3 système de fichiers local, noms de fichiers longs, i-noeuds détaillés, système de fichiers journalisé.

   xiafs système de fichiers local, noms de fichiers longs, i-noeuds détaillés, et d'autres avantages.

msdos système de fichiers local pour les partitions MS-DOS.

vfat système de fichiers local pour les partitions FAT32.

NTFS système de fichiers local pour les partitions NTFS.

HPFS système de fichiers local pour les partitions HPFS.

ISO9660 système de fichiers local pour les CD-ROM.

NFS système de fichiers distant par réseau.

swap partition de disque utilisée pour le swapping.

4. Si vfs_fstype est mentionné comme “ignore”, l’enregistrement n’est pas traité. Ceci permet de visualiser aisément les partitions non utilisées.
5. Le quatrième champ (fs_mntops), indique des options de montage associées au système de fichiers. Il s’agit d’une liste d’options séparées par des virgules. Il contient au moins le type de montage, suivi éventuellement d’options appropriées au type de système de fichiers.

   async Toutes les entrées/sorties sur le système de fichiers seront asynchrone.

   auto Peut être monté par l’argument -a.

   defaults Utilisation des options par défaut : rw, suid, dev, exec, auto, nouser, et async.

   dev Interpréter les fichiers spéciaux de périphériques présents sur le système de fichiers.

   exec Permettre l’exécution de fichiers binaires.

   noauto Ne peut être monté qu’explicitement (l’invocation de mount avec l’argument -a ne montera pas le système de fichiers).

   nodev Ne pas interpréter les fichiers spéciaux de périphériques présents sur le système de fichiers.

   noexec Ne pas permettre l’exécution de fichiers binaires sur le système de fichiers monté. Ceci peut être utile sur un serveur qui contient des fichiers binaires pour des architectures autres que la sienne.

   nosuid Ne pas tenir compte des bits Set-UID ou Set-GID.

   nouser Ne pas autoriser d’utilisateur ordinaire (non root) à monter le système de fichiers. C’est le comportement par défaut.

   remount Remonter un système de fichiers déjà monté. Ceci est utilisé pour changer les attributs de montage, principalement pour autoriser l’écriture sur un système en lecture seule.

   ro Montage du système en lecture seule.

   rw Montage du système en lecture/écriture.

   suid Prendre en compte les bits Set-UID ou Set-GID des fichiers se trouvant sur le système monté.

sync Toutes les entrées/sorties sur le système de fichiers seront synchrones.

user Autoriser les utilisateurs ordinaires (non root) à monter le système de fichiers. Ceci entraîne l'utilisation des options noexec, nosuid, et nodev (à moins qu'elles ne soient explicitement surchargées, comme dans une ligne d'option user,exec,dev,suid).

6. Le cinquième champ (fs_freq), est utilisé par la commande dump(8) pour déterminer quels sont les systèmes de fichiers à sauvegarder. Si le cinquième champ est absent ou vaut zéro, dump supposera qu'il ne faut pas sauvegarder ce système.
7. Le sixième champ (fs_passno), est utilisé par le programme fsck(8) pour déterminer l'ordre de vérification des systèmes de fichiers au moment du démarrage. Le système de fichiers racine doit avoir un champ fs_passno de valeur 1, et les autres un fs_passnode de valeur 2. Les systèmes partageant le même contrôleur seront vérifiés séquentiellement, et ceux utilisant différents contrôleurs seront vérifies simultanément pour utiliser le parallélisme offert par le matériel. Si le sixième champ est absent ou vaut zéro, fsck ne vérifiera pas ce système de fichiers.

4.6.7 /etc/syslog.conf

Le service syslog sert a logger diverse informations dans des fichier. Il est très utile pour la sécurité de votre machine car de nombreuse information sont enregistrer (les tentatives de connexion les services démarrer.), son fonctionnement est configurable au travers du fichier */etc/syslog.conf*. Il contient deux champ ,le sélecteur et l'action. Le sélecteur se compose de deux champs appeler "facility" et "priority". Les valeur de "facility" identifie le sous système qui a émis le message. Les type définis (dans /usr/include/syslog.h.) sont auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security (pareil que auth), syslog, user, uucp et de local0 a local7. Notons que "security" est obsolète et que "mark" est réservée a un usage interne.

L'argument "priority" sert a définir le niveau de détails de se qui doit être logué. Les valeurs possible sont : debug, info , notice, warning, err , crit, alert, emerg.

```
# Various entry
auth,authpriv.*                                         /var/log/auth.log
*.*;auth,authpriv.none                                  -/var/log/syslog
user.*                                                  -/var/log/user.log

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;;news.none;authpriv.none               -/var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Mail logging
mail.=debug;mail.=info;mail.=notice                     -/var/log/mail/info
```

```
mail.=warn                                              -/var/log/mail/warnings
mail.err                                                -/var/log/mail/errors

# Kernel logging
kern.=debug;kern.=info;kern.=notice                     -/var/log/kernel/info
kern.=warn                                              -/var/log/kernel/warnings
kern.err                                                /var/log/kernel/errors

# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                                          -/var/log/spooler

# Save boot messages also to boot.log
local7.*                                                -/var/log/boot.log

*.info;mail.none;authpriv.none  /dev/tty7
authpriv.*      /dev/tty7
*.warn;*.err    /dev/tty7
kern.*  /dev/tty7
mail.*  /dev/tty8
```

4.6.8 /etc/hosts

hosts - Correspondances statiques de noms d'hôtes.

Il s'agit d'un fichier de texte simple qui associe les adresses IP avec les noms d'hôtes, une ligne par adresse IP. Pour chaque hôte, une unique ligne doit être présente, avec les informations suivantes :

Adresse_IP Nom_officiel [Alias...]

Les divers champs de la ligne sont séparés par un nombre quelconque d'espaces ou de tabulations. Le texte commençant avec un caractère "#" et s'étendant jusqu'à la fin de la ligne est considéré comme un commentaire, et est donc ignoré. Les noms d'hôtes peuvent contenir n'importe quel caractère imprimable autres que les délimiteurs de champs, les saut de ligne ou le caractère de commentaire. Les alias permettent de disposer de noms différents, d'orthographe simplifiées, de surnoms plus courts, ou de noms d'hôtes générique (comme localhost).

Le système Berkeley Internet Name Domain (BIND) implémente un serveur de noms Internet pour les systèmes Unix. Il remplace le fichier /etc/hosts ou la recherche des noms d'hôtes, et libère un hôte de la nécessité de disposer d'un fichier /etc/hosts complet et à jour.

Dans les systèmes modernes, même si la table des hôtes a été remplacée par DNS, ce mécanisme est encore largement employé pour :

Exemple :

127.0.0.1 localhost
192.168.1.217 foo.nomomaine.org foo
205.230.163.103 www.opensource.org

4.6.9 /etc/resolv.conf

Le fichier /etc/resolv.conf se compose de deux informations importantes que sont : “nameserver” qui définit la machine sur laquelle devront être transmisent les requêtes DNS c’est à dire les machines permettant de transformer les adresses telles que www.google.fr en adresse IP.

On voit ici qu’il est possible d’usurper une adresse par une autre, il sera donc indispensable d’utiliser une machine de confiance.

la deuxième information importante est “search” qui permettra de définir le(s) domaines de recherche des adresses, ainsi lorsque l’on tapera “ssh seb@machine” la recherche va donc se faire de préférence avec l’extension : “ssh seb@machine.domain” “domain” étant le champ suivant la valeur “search”.

4.6.10 /etc/services

services est un fichier de texte ASCII fournissant une correspondance entre un nom intelligible décrivant un service Internet et l’ensemble numéro de port / protocole utilisé.
Chaque programme réseau devrait consulter ce fichier pour obtenir le numéro de port et le protocole sous-jacent au service qu’il fournit. Les fonctions de la bibliothèque C getservent(3), getservbyname(3), getservbyport(3), setservent(3), et endservent(3) permettent d’interroger ce fichier depuis un programme.
Les numéros de ports sont affectés par le IANA (Internet Assigned Numbers Authority), et leur politique actuelle consiste à assigner à la fois les protocoles TCP et UDP à chaque numéro de port. Ainsi la plupart des services auront deux entrées, même si elles n’utilisent que le protocole TCP.
Les numéros de ports en-dessous de 1024 ne peuvent être assignés à une socket que par un programme Super-User (voir bind(2), tcp(7), et udp(7).)
C’est ainsi que les clients se connectant sur ces ports de petits numéros peuvent être sûrs que le service correspondant est une implémentassions standard et non pas le bricolage d’un utilisateur.
La présence d’une ligne indiquant un service dans le fichier services ne signifie pas nécessairement que le service en question est disponible sur la machine. Notez que tous les services réseau ne sont pas obligatoirement lancés par inetd ou xinetd, et donc pas toujours dans xinetd.d ou inetd.conf. En particulier les serveurs de News (NNTP) et de courrier (SMTP) sont souvent initialisés par le système dans les scripts de démarrage.

Chaque ligne décrivant un service est de la forme
service-name port/protocole [alias ...]
voici un exemple simplifié des entrées dans */etc/services* :

```
ftp-data        20/tcp          # default ftp data port
ftp             21/tcp
ssh\index{ssh}              22/tcp
ssh             22/udp
```

```
telnet          23/tcp
# 24 - private
\index{smtp}            25/tcp          mail
# 26 - unassigned
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
bootps          67/tcp          # BOOTP server
bootps          67/udp
bootpc          68/tcp          # BOOTP client
bootpc          68/udp
www             80/tcp          http    # WorldWideWeb HTTP
www             80/udp                  # HyperText Transfer Protocol
pop3            110/tcp         # POP version 3
pop3            110/udp
sftp            115/tcp
nntp            119/tcp         readnews untp   # USENET News Transfer Protocol
imap            143/tcp         imap2           # Internet Message Access Proto
imap            143/udp         imap2           # Internet Message Access Proto
irc             194/tcp                         # Internet Relay Chat
irc             194/udp
imap3           220/tcp                         # Interactive Mail Access
imap3           220/udp                         # Protocol v3

# new for imap ssl
imaps           993/tcp
pop3s           995/tcp

https           443/tcp                         # secure http (SSL)
nfsd            2049/udp        nfs     # NFS server
nfsd            2049/tcp        nfs     # NFS server
#
# Unofficial services
#
rsync           873/tcp                 # rsync server
mysql           3306/tcp                # MySQL

pserver    2401/tcp
```

4.6.11 /etc/sudoers

Sudo permet à un utilisateur d'exécuter une commande en tant que super-utilisateur (ou autre), comme il l'a été spécifié dans le fichier */etc/sudoers*.
Par défaut il sera demander le mot de passe de l'utilisateur, mais il reste possible de le désactiver comme on peut le voir dans l'exemple suivant :

```
# User privilege specification
root ALL=(ALL) ALL
# %wheel ALL=(ALL) ALL
# Same thing without a password
# %wheel ALL=(ALL) NOPASSWD : ALL
seb ALL=(ALL) NOPASSWD : ALL
```

Remarque

sudo est un programme facilitant les commandes "super-utilisateur" néanmoins c'est

un programme SUID dangereux. Qui a fait récemment l'objet d'une faille de sécurité via buffer-overflow.

CHAPITRE 5

Administration Réseaux

5.1 La sécurité du système

UN système d'exploitation n'est jamais fiable à 100%. Ainsi, des failles ont été décelées sous Linux en terme de gestion réseau. Par exemple, avec la commande inetd, il était possible de faire "planter" un serveur Linux qui accepte les connexions TCP, par "flooding" (inondation) de requêtes vers celui-ci. Une cinquantaine de requêtes font que le serveur va refuser les connexions inetd, parce que le serveur aura planté. Cela marche pour tous les services sous inetd, donc ftpd, identd... Bien évidemment, dès que cette faille fut décelée, une nouvelle version corrigée du noyau et des applications avait corrigé ce problème.

Dès que le noyau est modifié, il est nécessaire de le recompiler et de redémarrer la machine pour prendre en compte les nouveaux paramètres : pratiquement, ces mises à jour doivent donc être réalisées en dehors des heures d'utilisation de la machine. Dans le cas d'une machine Linux configurée en routeur , il ne faudra mettre à jour le noyau uniquement si une faille importante mettant en jeu la sécurité du réseau est décelée, la trop grande fréquence des mises à jour pouvant nuire sérieusement aux utilisateurs du réseau.

Pour rajouter des fonctions particulières au noyau sans avoir à recompiler obligatoirement ce dernier, on utilise des modules. Ce sont des morceaux de code chargés dynamiquement, permettant d'ajouter des pilotes de périphériques matériels, des protocoles réseaux... Les modules sont exécutés dans l'espace mémoire du noyau :

- ils possèdent un contrôle total de la machine,
- ils peuvent détourner ou créer un appel système.

Les modules ont à la fois des avantages et des inconvénients :

– ils permettent aux administrateurs d'optimiser la sécurité du système : mettre à jour un pilote, surveillance de l'activité du système (fichiers LOG),
– ils permettent également aux pirates d'attaquer la sécurité du système : appels système néfastes, accès aux fichiers LOG (modifications, suppression), changement des droits, lancement de processus cachés, mise en place de portes dérobées...
Pour plus d'informations sur les modules, consulter le site suivant : http ://dione.ids.pl/ lcamtuf/pliki.html : de nombreuses sources de modules ou de patchs noyaux pour Linux 2.0.3x ou 2.2.x, la majorité en polonais malheureusement.

5.2 IP Aliasing

L'IP aliasing est fonctionnalité très intéressante qui permet d'affecter plusieurs IP à la même carte réseau. Le seul problème viens du fait que l'on ne pourra pas affecter les règles sur ces interfaces au firewall, en effet Netfilter (IPtables) ne le supporte pas. Il sera donc nécessaire de donner des règles sur les adresses IP ce qui affaiblira la fiabilité et la rapidité de celui-ci.

Système	commande
Linux	ifconfig eth0 :1 192.168.1.45 netmask 255.255.255.0
BSD	ifconfig xl0 192.168.1.45 netmask 255.255.255.0 alias

Nota sous Linux il suffira donc d'ajouter " :1", " :2" à la fin de la carte réseau concernée. Sous *BSD il suffira de mettre "alias" à la fin de la commande ifconfig.

5.3 Installation d'un client/serveur SSH

Installation du serveur SSH SSH (Secure Shell, port 22) est un aujourd'hui la façon la plus utilisée pour se connecter sur une autre machine. En effet lors de la connexion le mot de passe ainsi que les données sont cryptées, ce qui évite aux sniffer de pouvoir capturer les mots de passe et les données transitants sur le réseau.

installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh openssh-server-*.rpm	service sshd start
Mandrake	urpmi openssh-server	/etc/rc.d/init.d/sshd start
Debian	apt-get install openssh	/etc/rc.d/init.d/sshd start
OpenBSD	par defaut inclus par defaut.(car créateur !)	/sbin/sshd

configuration

le fichier de configuration se trouve dans */etc/ssh/sshd_config* les options les plus intéressantes sont :

#Port 22	spécifie le port que le serveur doit utiliser.
Protocol 2,1	Protocol utilisé
#LoginGraceTime 600	Temps de connexion maximum
#PermitRootLogin yes	permet ou interdit la connexion “root”
#RSAAuthentication yes	Méthode d’authentification.
#AuthorizedKeysFile .ssh/authorized_keys	fichier utilisé pour “l’autologin”
#PermitEmptyPasswords no	permet ou non les mots de passe vide
X11Forwarding yes	permet ou non d’exporter le DISPLAY

Remarques

Il plus que préférable d’ajouter ces lignes à votre “sshd_config” :

```
ClientAliveInterval 15
ClientAliveCountMax 3
```

afin d’éviter qu’une connexion morte (ex : client débranché) ne fasse l’objet d’une récupération de session.

Connexion SSH sur une autre machine *ssh toto@192.168.0.2*

NOTA : 1. Pour les connections ssh sans donner de mot de passe (ex. scripts de backups) vous trouverez toute la procédure dans la section Astuces/Connexion SSH sans mot de passe.

5.4 Mise en place d’un serveur mail. SMTP

SMTP (Simple Mail Transfer Protocol, port 25), c’est le protocole utilisé pour envoyer des mails. le standart reste sendmail compte tenu de sa stabilité en environnement intensif. Néanmoins nous étudierons d’autres serveurs de mail tels que postfix qui sont inclus par defaut sur certaines distributions, même s’ils ne disposent pas de la solidité de sendmail.

5.4.1 Mise en place de Postix

installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh postfix-*.rpm	service postfix start
Mandrake	urpmi postfix	/etc/init.d/postfix start
Debian	apt-get install exim	/etc/init.d/exim start
OpenBSD	sendmail inclus par defaut.	/usr/sbin/sendmail -L sm-mta -bd -q30m
NetBSD	sendmail inclus par defaut.	/etc/rc.d/sendmail start

Configuration de Postix Afin d'obtenir un serveur de base il vous faudra modifier simplement le fichier */etc/postfix/main.cf* :

1. décommenter la ligne commençant par *#myhostname =*
2. myhostname = nom_de_votre_machine
3. /etc/rc.d/init.d/postfix start

5.4.2 Mise en place de Sendmail

installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh sendmail-*.rpm	service postfix start
Mandrake	urpmi sendmail	/etc/init.d/postfix start
Debian	apt-get install sendmail	/etc/init.d/exim start
OpenBSD	sendmail inclus par defaut.	/usr/sbin/sendmail -L sm-mta -bd -q30m
NetBSD	sendmail inclus par defaut.	/etc/rc.d/sendmail start

Configuration de Sendmail
La configuration de sendmail livrée avec Linux/BSD convient à des sites directement connectés à l'Internet.
Modifier /etc/mail/sendmail.cf à la main est généralement considéré comme une occupation de puriste. La version 8 de sendmail s'accompagne d'une nouvelle manière de générer les fichiers de configuration avec le préprocesseur m4, grâce auquel le travail de configuration à la main se fait à un niveau d'abstraction plus élevé.

Ainsi pour configurer sendmail vous pourrez vous inspirer du fichier sendmail.mc ci-dessous. Cette version a été configurée pour un NetBSD-1.6 il vous faudra donc modifier certains des champs :

ce sera le cas par exemple pour l'emplacement du fichier cf.m4
include('/usr/share/sendmail/m4/cf.m4')

ou encore le champ définissant le type d'OS utilisé :
OSTYPE(bsd4.4)dnl

Vous aurez sûrement remarqué les 'dnl' à la fin des champs, il sont en effet nécessaire si vous voulez que sendmail accepte le fichier généré. On peut s'en passer mais il faudra alors s'assurer qu'il n'existe pas d'espace en fin de ligne.

Enfin une fois ce fichier personnalisé il suffira de tapper la commande :
m4 sendmail.mc > /etc/mail/sendmail.cf pour généré le fichier sendmail.cf utilisé par sendmail.

5.4.3 sendmail.mc personnalisé

```
divert(-1)
dnl les lignes commencent par 'dnl' n'apparaiteront pas
dnl dans le fichier sendmail.cf généré contrairement aux
dnl lignes commençant par un dièse.

divert(0)dnl
#_____________________ Version ____________________________________________
include('/usr/share/sendmail/m4/cf.m4')
VERSIONID('$Id: cours_unix.tex,v 1.71 2002/10/28 23:36:29 seb Exp $')
OSTYPE(bsd4.4)dnl

#_____________________ Options Diverses ___________________________________
#
# ne fait pas de requêtes dns ??
FEATURE('nocanonify')
# Message d'accueil lors de connection a sendmail
# par defaut on a: ($j Sendmail $v/$Z; $b)
# confSMTP_LOGIN_MSG($j Sendmail $Z; $b bienvenue ...)
# !!! DON'T Work !!!
Cw samaro.org
#

#_____________________ Restrictions d'accès _______________________________
#
FEATURE('masquerade_envelope')dnl
#
FEATURE('always_add_domain')dnl
#
FEATURE('redirect')dnl
#
# FEATURE(relay_entire_domain)
#
# accepte de relayer le courrier uniquement
# depuis les hôtes de /etc/hosts
FEATURE(relay_hosts_only)dnl
#
# accepte de relayer les mails des utilisateurs+domaines
# n'appartenant pas au serveur.
FEATURE(accept_unqualified_senders)dnl
#
# Utilisateurs ayant le droit de modifier le champ From:
# lors de l'envoie d'un mail:
FEATURE(use_ct_file)dnl
```

```
define('confCT_FILE','/etc/mail/trusted-users')dnl
#
# gestion liste noire: consulte rbl.maps.vix.com
# avant d'accepter le mail.
# on peut spécifier un autre serveur FEATURE(rbl, 'rbl.hote.org')
FEATURE('dnsbl')
#
# permet d'avoir un accès simplifie aux hôtes, les règles
# sont: OK (accepte), RELAY (accepte de relayer), REJECT (rejette)
# DISCARD (ignore). exemple:
# toto@toto.fr REJECT
# free.fr        OK
# ...
# base générée par makemap hash /etc/mail/access.db < /etc/mail/access
# FEATURE(access_db)



#_____________________ domaines virtuels _____________________________________
#
# permet d'accepter les mails d'autres domaines
FEATURE(use_cw_file)
#
# permet de spécifier les domaines en question
# dans le fichier /etc/virtualnames
# exemple:
# free.fr
# ...
#
define('confCW_FILE','/etc/mail/virtualnames')
# Il est également possible de rediriger tout un domaine virtuel
# vers un seul utilisateur ou plusieurs:
# makemap hash /etc/mail/virtusertable.db < /etc/mail/virtusertable
FEATURE(virtusertable)
#
# La configuration se fait dans le fichier /etc/mail/virtusertable
# exemple:
# @toto.org seb
# test@titi.fr john

#_____________________________________________________________________________
#
MAILER('local')
MAILER('smtp')
```

5.4.4 sendmail.mc compatible avec Cyrus-Imapd

```
divert(-1)
dnl les lignes commencent par 'dnl' n'apparaiteront pas
dnl dans le fichier sendmail.cf généré contrairement aux
dnl lignes commençant par un dièse.

divert(0)dnl
#_____________________ Version ____________________________________________
include('/usr/share/sendmail/m4/cf.m4')
VERSIONID('$Id: cours_unix.tex,v 1.71 2002/10/28 23:36:29 seb Exp $')
OSTYPE(bsd4.4)dnl


#_____________________ ssi cyrus___________________________________________
```

```
# ne laisser ces lignes ssi on utilise cyrus-imapd
define(‘confBIND_OPTS’,‘-DNSRCH -DEFNAMES’)
define(‘confLOCAL_MAILER’, ‘cyrus’)


#_____________________ Options Diverses _____________________________________
#
# ne fait pas de requêtes dns ??
FEATURE(‘nocanonify’)
# Message d’accueil lors de connection a sendmail
# par defaut on a: ($j Sendmail $v/$Z; $b)
# confSMTP_LOGIN_MSG($j Sendmail $Z; $b bienvenue ...)
# !!! DON’T Work !!!
Cw samaro.org
#

#_____________________ Restrictions d’accès _________________________________
#
FEATURE(‘masquerade_envelope’)dnl
#
FEATURE(‘always_add_domain’)dnl
#
FEATURE(‘redirect’)dnl
#
# FEATURE(relay_entire_domain)
#
# accepte de relayer le courrier uniquement
# depuis les hôtes de /etc/hosts
FEATURE(relay_hosts_only)dnl
#
# accepte de relayer les mails des utilisateurs+domaines
# n’appartenant pas au serveur.
FEATURE(accept_unqualified_senders)dnl
#
# Utilisateurs ayant le droit de modifier le champ From:
# lors de l’envoie d’un mail:
FEATURE(use_ct_file)dnl
define(‘confCT_FILE’,‘/etc/mail/trusted-users’)dnl
#
# gestion liste noire: consulte rbl.maps.vix.com
# avant d’accepter le mail.
# on peut spécifier un autre serveur FEATURE(rbl, ’rbl.hote.org’)
FEATURE(‘dnsbl’)
#
# permet d’avoir un accès simplifie aux hôtes, les règles
# sont: OK (accepte), RELAY (accepte de relayer), REJECT (rejette)
# DISCARD (ignore). exemple:
# toto@toto.fr REJECT
# free.fr        OK
# ...
# base générée par makemap hash /etc/mail/access.db < /etc/mail/access
# FEATURE(access_db)



#_____________________ domaines virtuels ____________________________________
#
# permet d’accepter les mails d’autres domaines
FEATURE(use_cw_file)
#
# permet de spécifier les domaines en question
# dans le fichier /etc/virtualnames
# exemple:
# free.fr
# ...
```

```
#
define('confCW_FILE','/etc/mail/virtualnames')
# Il est également possible de rediriger tout un domaine virtuel
# vers un seul utilisateur ou plusieurs:
# makemap hash /etc/mail/virtusertable.db < /etc/mail/virtusertable
FEATURE(virtusertable)
#
# La configuration se fait dans le fichier /etc/mail/virtusertable
# exemple:
# @toto.org seb
# test@titi.fr john

#_____________________________________________________________________________
#
MAILER('local')
MAILER('smtp')
#_____________________________________________________________________________
# a décommenter pour que les mail arrive directement
# dans cyrus si les règles:
# LOCAL_RULE_0
# Rbb + $+ < @ $=w . >  $#cyrusbb $: $1
# ne sont pas acceptées:
#define('CYRUS_MAILER_FLAGS','A5@S')
#define('confLOCAL_MAILER','cyrus')
MAILER('cyrus')

#____________________ ssi cyrus_____________________________________________
LOCAL_RULE_0
Rbb + $+ < @ $=w . >    $#cyrusbb $: $1
```

5.5 installation & configuration d'Apache

Apache est un serveur HTTP (Hyper Transfert Protocol, port 80), il est aujourd'hui le serveur web le plus utilisé sur internet.

installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh apache-*.rpm	service httpd start
Mandrake	urpmi apache	/etc/init.d/httpd start
Debian	apt-get install apache	/etc/init.d/httpd start
OpenBSD	apache inclus par defaut.	httpd

httpd.conf

le fichier de configuration le plus important pour la configuration d'apache est httpd.conf que l'on trouvera sur les différentes distributions :

Système	emplacement de httpd.conf	répertoire du serveur
RedHat		
Mandrake	/etc/httpd/conf/httpd.conf	/var/www/html
Debian		
OpenBSD	/var/www/conf/httpd.conf	/var/www/htdocs

à l'intérieur de ce fichier on trouvera les principales variables que sont :

ServerRoot /etc/httpd	le chemin où trouver les fichiers de configuration.
DocumentRoot /var/www/html	le chemin par défaut pour trouver les fichiers du site.
ErrorLog logs/error_log	l'emplacement ou écrire les logs d'erreur.
ServerAdmin root@localhost	mail de l'administrateur du site.
DirectoryIndex index.html	Définie l'ordre de préférence des fichiers à ouvrir dans chaques répertoires.
LoadModule	permet d'ajouter un module à Apache (ec : php, etc...)
User et Group	Nom et Groupe sous lequel tournera Apache
HostNameLookups (on/off)	indique à Apapche s'il doit rechercher systématiquement les noms des hôtes (peut ralentir les requêtes)
Port	c'est le numéro de port que Apache écoutera. (80)

test de la configuration

Pour tester la configuration d'Apache il existe une option à apachectl : "apachectl configtest"

limitations d'accès

pour générer le fichier .htpasswd il suffit de taper la commande :
htpasswd [-c] .htpasswd username
puis
chmod 644 .htpasswd
ensuite il faut générer le fichier .htaccess en y insérant :

```
AuthUserFile /var/www/html/private   chemin où l'on pourra trouver le fichier htpasswd
AuthGroupFile /dev/null              si l'on utilise pas de group on note /dev/null
AuthName TITRE                       titre de la fenêtre de dialogue
AuthType Basic

<limit GET>
require valid-user
</Limit>
```

puis
chmod 644 .htaccess

5.6 installation & configuration d'Apache sécurisé

Création du certificat DSA du serveur (sous OpenBSD)

```
cd /etc/ssl
openssl dsaparam 1024 -out dsa1024.pem
openssl req -x509 -nodes -newkey dsa :dsa1024.pem -out /etc/ssl/dsacert.pem
-keyout /etc/ssl/private/dsakey.pem
```

Création du certificat RSA pour le serveur web (sous OpenBSD)

```
cd /etc/ssl
openssl genrsa -out /etc/ssl/private/server.key 1024
openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr
```

Enfin on auto-signe notre certificat....

```
openssl x509 -req -days 365 -in /etc/ssl/private/server.csr
-signkey /etc/ssl/private/server.key -out /etc/ssl/server.crt
```

Démarrage du serveur web avec support SSL (sous OpenBSD)

Il suffit de replacer, dans le fichier */etc/rc.conf*, la ligne *httpd_flags=""* par *httpd_flags="-DSSL"* ou de lancer en ligne de commande *httpd -DSSL*.

5.7 installation d'un serveur ftp

FTP (File Transfer Protocol, port 21), comme son nom l'indique c'est le protocole utilisé pour le transfert de fichiers. Le serveur FTP le plus répandu aujourd'hui est "wu-ftpd", néanmoins compte tenu des récents gros problèmes de sécurité il est conseillé d'utiliser une version récente.

installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh proftpd-*.rpm	service proftpd start
Mandrake	urpmi proftpd	/etc/init.d/proftpd start
Debian	inclus par defaut	décommentez la ligne ftp dans inetd.conf
OpenBSD	inclus par defaut. néanmoins il faudra ajouter un utilisateur ftp : "useradd -m ftp"	décommentez la ligne ftp dans inetd.conf

Remarque :

Pour les Distributions BSD et Debian il suffira de redémarrer le service inetd pour démarrer le serveur : "kill -HUP 'cat /var/run/inetd.pid'

serveur ftp “normal”

Ensuite si l’on souhaite restreindre l’accès de certains utilisateurs, il suffit d’ajouter un entré au fichier */etc/ftpusers* pour que celui ne puisse plus accéder au serveur ftp.

```
root@ramses /etc # more ftpusers
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

serveur ftp anonyme

anonftp est un path qui permet de se connecter en mode anonyme dans le répertoire */var/ftp*.

Système	commande d’installation
RedHat	rpm -Uvh anonftp-*.rpm
Mandrake	urpmi proftpd-anonymous
Debian	inclus par defaut
OpenBSD	inclus par defaut. il suffit de modifier le fichier /etc/inetd.conf et d’ajouter “ -AUS” la fin de la ligne contenant ftp, le répertoire par defaut sera celui de l’utilisateur “ftp”

Remarque :

Pour se connecter en mode anonyme, il suffit lors de la phase d’authentification d’entrer “anonymous” et comme mot de passe “votre adresse mail”

5.8 installation du proxy Squid

Un serveur proxy permet d’accélérer l’apparition des pages web sur le navigateur. en effet il utilise une partie du disque dur comme cache et évite ainsi d’aller chercher une page identique plusieurs fois sur le serveur distant.
SQUID est le plus populaire des proxy sous Linux/Unix d’une grande stabilité, il supporte les protocoles FTP, HTTP, gopher, SSL. il permet de plus une authentification des clients par login/pass utile si l’on ne souhaite pas utiliser l’IPmasquerade. SQUID permet aussi d’interdire l’accès à certaines pages.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh squid-*.rpm	service squid start
Mandrake	urpmi squid	/etc/init.d/squid start
Debian	apt-get install squid	/etc/init.d/squid start
OpenBSD	pkg_add squid-*.tgz	/usr/local/bin/squid

Configuration

Voici un exemple de configuration de squid : (*/etc/squid/squid.conf*) :

http_port 3128	port utilisé pour le serveur.
hierarchy_stoplist cgi-bin ?	
acl QUERY urlpath_regex cgi-bin	
no_cache deny QUERY	
cache_mem 40 MB	taille maximum allouée dans la mémoire vive !.
cache_dir ufs /var/squid/cache 500 16 256	répertoire ou placer le cache.
cache_access_log /var/squid/logs/access.log	
cache_log /var/squid/logs/cache.log	
acl all src 0.0.0.0/0.0.0.0	définition de la source "all"
acl localnet src 192.168.0.0/255.255.0.0	définition de "localnet"
acl manager proto cache_object	
acl Safe_ports port 80 21 443 1025-65535	
acl localhost src 127.0.0.1/255.255.255.255	définition de "localhost"
acl SSL_ports port 443 563	ports "sécurisés"
acl CONNECT method CONNECT	
http_access allow localnet	permissions pour "localnet"
http_access allow manager localhost	permissions pour "localhost"
http_access deny manager	
http_access deny !Safe_ports	
http_access deny CONNECT !SSL_ports	
http_access deny all	
icp_access allow all	
cache_mgr seb@mail_administrateur.fr	
visible_hostname cache.domain.org	

Proxy transparent.

Un proxy transparent est un proxy que les clients n'auront pas besoin de configurer. En effet le va automatiquement rediriger tous les paquets a destination du port 80 vers l'adresse IP du proxy sur le port du proxy (dans notre cas 3128). Néanmoins il sera indispensable d'ajouter ces lignes au fichier de configuration de squid.

```
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
httpd_accel_host localnet
httpd_accel_host virtual
httpd_accel_host CACHE
```

Navigation anonyme.

Pour éliminer un maximum d'informations sur les clients il est possible de restreindre les balises normalement envoyées aux différents serveurs, pour cela il suffira d'ajouter ces lignes a la fin du fichier /etc/squid/squid.conf.

```
anonymize_headers allow Allow Authorization Cache-Control
anonymize_headers allow Content-Encoding Content-Length
anonymize_headers allow Content-Type Date Expires Host
anonymize_headers allow If-Modified-Since Last-Modified
anonymize_headers allow Location Pragma Accept
anonymize_headers allow Accept-Encoding Accept-Language
anonymize_headers allow Content-Language Mime-Version
anonymize_headers allow Retry-After Title Connection
anonymize_headers allow Proxy-Connection
```

Remarque

Afin de créer la zone de cache il faudra effectuer la commande "squid -z" sur les systèmes *BSD.

5.9 installation d'un serveur samba

Samba est un système de fichier réseaux permettant de partager des fichier entre des systèmes Linux et Windows c'est un portage des fichiers partagés sous Windows. Il est ainsi possible de partager des répertoires ou des imprimantes avec des systèmes Windows.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh samba-*.rpm	service samba start
Mandrake	urpmi samba	/etc/init.d/smbd start /etc/init.d/nmbd start
Debian	apt-get install samba	/etc/init.d/smbd start
OpenBSD	pkg_add samba-*.tgz	/usr/local/libexec/smbd -D /usr/local/libexec/nmbd -D

/etc/smb.conf

```
[global]
server string = Linux Samba Server          # nom du serveur
workgroup = SAMARO
smb passwd file = /etc/samba/smbpasswd
security = user                             # share permet la connexion
                                            # d'utiliateurs qui n'existent
                                            # pas sur le serveur.
allow hosts = 192.168.0. 127.
dns proxy = no
encrypt passwords = yes                     # permet la comptabilité entre
                                            # windows et linux
null passwords = yes
log file = /var/log/samba/log.%m
printing = cups                             # definit le type d'imprimante
load printers = yes                         #partage de toutes les imprimantes
printcap name = lpstat                      # imprimantes configurées
                                            # ici qui seront prises en compte
max log size = 50
_____________________       _____________________

[homes]
comment = Home Directories
browseable = no
writable = yes
_____________________       _____________________

[patage]
comment = documents partages    #description
path = /var/docs
writable = yes                  #permet l'écriture
public = yes                    #aucun mot de passe n'est demandé
browseable = yes                # apparaît dans le réseau du client
_____________________       _____________________

[printers]
comment = Lexmark under linux   #description de l'imprimante
browseable = yes
printable = yes                 # le partage est une imprimante
public = yes
directory = /tmp                #répertoire qui va servir de spool
```

/etc/samba/smbpasswd

pour ajouter un utilisateur on tape simplement en tant que root :

smbpasswd -a utilisateur

on obtiendra ainsi :

```
seb :1000 :90F4B09D49F396ABBA2CF1C :[U ] :LCT-3C909D46
root :0 :C9DA7B438 :CC870177CA0JJHGD :[U ] :LCT-3C9CEB3C
```

5.10 installation d'un serveur pop3{s}

POP (Post Office Protocol, port 110), fournit de base par la majorité de fournisseurs d'accès Internet, POP permet de récupérer son courrier sur un serveur distant. Il moins évolué que IMAP qui permet en plus de stocker et de trier son courrier directement sur le serveur.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh gnu-pop3d-*.rpm	service xinetd restart
Mandrake	urpmi gnu-pop3d	/etc/init.d/xinetd restart
Debian	inclus par defaut.	via /etc/inetd.conf
OpenBSD	inclus par defaut.	via /etc/inetd.conf

utilisation

Pour configurer ce service il suffit d'éditer les fichiers "*pop3*" du répertoire */etc/xined.d*

Remarques

Il est bien sur recommandé d'utiliser pop3s par rapport pop3, en effet afin d'éviter les problèmes de sniffing qui sont extrêmement performants pour les accès non sécurisés.
De plus il ne sera pas nécessaire d'installer un serveur pop3 si vous comptez installer un serveur imap car les serveurs imap possèdent par defaut un serveur pop3.

Création du certificat (sous OpenBSD)

```
cd /etc/ssl
openssl req -new -x509 -nodes -out ipop3d.pem -keyout ipop3d.pem -days 3650
```

5.11 installation d'un serveur imap{s}

IMAP (Internet Message Access Protocol, port 143), IMAP permet de stocker de classer et de trier son courrier directement sur le serveur.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh imap-*.rpm	service imapd start
Mandrake	urpmi imapd	/etc/init.d/imapd start
Debian	apt-get install wu-imapd.	/etc/init.d/imapd start
OpenBSD	pkg_add imap-wu-*.tgz	via /etc/inetd.conf

Remarques

même remarque que pour pop3, préférez imaps à imap autant que possible.

Création du certificat (sous OpenBSD)

```
cd /etc/ssl
openssl req -new -x509 -nodes -out imapd.pem -keyout imapd.pem -days 3650
```

5.12 installation d'un serveur DHCP

configuration dynamique des hôtes et le protocole de démarrage par Internet (BOOTP). DHCP permet à des hôtes appartenant à un réseau TCP/IP de demander et d'obtenir l'affectation d'adresses IP, mais aussi de découvrir les informations relatives au réseau auquel ils sont rattachés.
BOOTP fournit des fonctionnalités similaires, mais avec quelques restrictions. Le protocole DHCP permet à un hôte, qui est inconnu de l'administrateur réseau, d'obtenir l'affectation d'une nouvelle adresse IP parmi un ensemble d'adresses pour ce réseau. Pour ce faire, l'administrateur du réseau alloue un ensemble d'adresses dans chaque sous-réseau et les déclare dans le fichier dhcpd.conf(5).
Au démarrage, dhcpd lit le fichier dhcpd.conf et stocke en mémoire la liste des adresses disponibles dans chaque sous-réseau. Lorsqu'un client demande une adresse en utilisant le protocole DHCP, dhcpd lui en attribue une. Chaque client reçoit son adresse en concession : la concession expire au bout d'une durée choisie par l'administrateur (par défaut, une journée). Les clients qui ont reçu une adresse sont supposés renouveler la concession avant expiration, pour pouvoir continuer à l'utiliser. Une fois que la durée de la concession s'est écoulée, le client titulaire n'est plus autorisé à utiliser l'adresse IP qu'il avait reçue.

Pour garder la trace des concessions accordées en dépit des redémarrages du serveur, dhcpd inscrit la liste des concessions attribuées dans le fichier dhcpd.lease(5). Avant d'accorder une concession à un hôte, dhcpd enregistre l'attribution dans ce fichier et s'assure que le contenu du fichier est recopié sur le disque. Ceci permet d'être sûr que, même dans le cas d'un crash système, dhcpd n'oubliera rien d'une concession qui a été accordée. Au démarrage, après avoir lu le fichier dhcpd.conf, dhcpd lit le fichier dhcpd.leases pour mettre à jour sa mémoire à propos des concessions qui ont été accordées.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh dhcp-server-*.rpm	service dhcpd start
Mandrake	urpmi dhcp-server	/etc/init.d/dhcpd start
Debian	apt-get install dhcpd	/etc/init.d/dhcpd start
OpenBSD	par defaut	dhcpd nom_carte

configuration

/etc/dhcpd.conf

```
server-identifier ns.exemple.fr ;
default-lease-time 36000 ;
max-lease-time 144000 ;
ddns-update-style interim ;
deny unknown-clients ;


subnet 172.253.111.0 netmask 255.255.255.0 {
range 172.253.111.32 172.253.111.127 ;
option domain-name "g11.epita.fr" ;
option domain-name-servers 172.253.111.1 ;
option routers 172.253.111.1 ;
option subnet-mask 255.255.255.0 ;
ddns-updates on ;
ddns-domainname "ns.exemple.fr" ;
# ddns-rev-domainname "in-addr.arpa" ;


host pdc {
hardware ethernet 00 :48 :54 :12 :6c :b1 ;
fixed-address 172.253.111.3 ;
}

host service {
hardware ethernet 00 :48 :54 :12 :72 :83 ;
fixed-address 172.253.111.2 ;
}
}
```

Remarques

l'option "deny unknown-clients" va permettre de refuser toute connexion aux hôtes ayant une adresse MAC différente de "service" et "mail".

5.13 installation de NFS

NFS est un service très utilisé, il permet pour un client d'utiliser un répertoire sur un serveur distant de la même façon que s'il s'agissé d'une partition locale. Les 3 fichiers de configuration principaux sont /etc/exports, /etc/hosts.deny et /etc/hosts.allow.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh nfs-utils-*.rpm	
Mandrake	urpmi nfs-utils	
Debian	apt-get install nfs	
OpenBSD	par defaut	

configuration

Le fichier /etc/exports sert de liste de contrôle d'accès pour les systèmes de fichiers à exporter aux clients NFS. Il est utilisé à la fois par le démon de montage NFS mountd(8) et par le démon serveur NFS nfsd(8).

Le format de ce fichier est similaire à celui de SunOS, exports, avec plusieurs options supplémentaires. Chaque ligne correspond à un point de montage, et à une liste de machines, ou de noms de sous-réseaux autorises à monter le système de fichiers situé en ce point. Une liste éventuelle de paramètres est mise entre parenthèses après le nom de machine. Les lignes blanches sont ignorées, et un # indique un commentaire s'étendant jusqu'à la fin de la ligne.

Options générales :

secure	Cette option réclame que les requêtes proviennent d'un port Internet de numéro inférieur à IPPORT_RESERVED (1024). Elle est en service par défaut. Pour la désactiver, introduire le mot-clé insecure.
ro	N'autorise que des requêtes en lecture-seule sur ce volume NFS. Le comportement par défaut autorise également les requêtes en écriture, ce que l'on peut également mentionner explicitement en utilisant l'option rw.
link_relative	Convertit les liens symboliques absolus (dont le contenu commence par un slash) en lien relatifs en ajoutant le nombre nécessaire de "../" pour aller de l'emplacement du lien à la racine du serveur. Cette option a une sémantique assez subtile, parfois surprenante lorsque la hiérarchie n'est pas montée à la racine.
link_absolute	Laisse les liens symboliques tels qu'ils sont. C'est l'attitude par défaut.

Correspondance des UID

nfsd effectue les contrôles d'accès aux fichiers du serveur en se basant sur les UID et GID fournis avec chaque requête RPC NFS. L'utilisateur s'attend à pouvoir accéder à ses fichiers sur le serveur NFS exactement comme il y accéderait sur un système de fichiers normal.

Ceci nécessite que les mêmes UID et GID soient utilisés sur le serveur et sur la machine cliente. Ce n'est pas toujours vrai, et même parfois indésirable.

Il est souvent très gênant que le Super-User d'une machine cliente soit traité comme le Super-User lorsqu'il accède aux fichiers du serveur. Pour éviter ceci, l'UID 0 (root) est normalement transformé en un autre UID : par exemple en UID anonyme nobody. Ce mode opératoire s'appelle 'root squashing' et est le comportement par défaut. On peut le supprimer avec l'option no_root_squash.

De plus, nfsd vous permet d'indiquer des UIDs ou GIDs arbitraires qui seront transformés en utilisateur anonymes. Enfin, vous pouvez même transformer les requêtes de tous les utilisateurs en UIG et GID anonymes avec l'option all_squash .

Pour être utilisé dans des sites où les UIDs varient suivant les machines, nfsd fournit une méthode de conversion dynamique des UIDs du serveur en UIDs du client et inversement. Ceci est mis en service avec l'option map_daemon et utilise le protocole RPC UGID. Il faut que le démon de conversion ugidd(8) soit actif sur le client.

Voici une liste complète des options de conversion :

root_squash	Transforme les requêtes des UID/GID 0 en UID/GID anonymes. Ceci ne s'applique pas aux autres UID/GID sensibles comme bin.
no_root_squash	Ne pas transformer les UID/GID 0. Cette option est particulièrement utiles pour les stations sans disque.
squash_uids et squash_gids	Cette option précise une liste d'UIDs ou de GIDs qui sont convertis en utilisateurs anonymes. Une liste valide d'UIDs se présente ainsi : squash_uids=0-15,20,25-50 Habituellement la liste à convertir est plus simple, comme dans : squash_uids=0-100
all_squash	Convertit tous les UID/GID en utilisateurs anonymes. Utiles pour exporter avec NFS des répertoires publics de FTP, des répertoires de News, etc... L'option inverse est no_all_squash, qui s'applique par défaut.
map_daemon	Cette option active la conversion dynamique des UID/GID. Chaque UID d'une requête NFS sera transformé en UID équivalent sur le serveur, et l'UID des réponses NFS subira la transformation inverse. Ceci nécessite que le démon rpc.ugidd(8) soit en fonctionnement sur le client. La configuration par défaut est map_identity, qui laisse les UIDs intacts. Les options de 'squash' s'appliquent que la conversion dynamique soit utilisée ou non.
anonuid et anongid	Ces options donne explicitement l'UID et le GID du compte anonyme. Ceci est surtout utiles pour les clients PC/NFS, où l'on désire que toutes les requêtes semblent provenir d'un utilisateur. C'est le cas de l'entrée /home/joe dans l'exemple ci-dessous, qui transforme toutes les requêtes en UID 150 (probablement celle de l'utilisateur joe).

Fichiers exemple :

fichier /etc/exports

```
/ maitre(rw) confiance(rw,no_root_squash)
/projects proj*.local.domain(rw)
/usr *.local.domain(ro)
/home/joe pc001(rw,all_squash,anonuid=150,anongid=100)
/pub (ro,insecure,all_squash)
```

fichier /etc/hosts.deny

On va interdire toutes les machines qui ne sont pas autorisée explicitement dans le /etc/hosts.allow. "ALL : ALL" interdira l'accès a tous les services depuis toutes machines.

fichier /etc/hosts.allow

```
portmap :192.168.1.34
lockd :192.168.1.34
mountd :192.168.1.34
rquotad :192.168.1.34
statd :192.168.1.34
```

La première ligne exporte l'ensemble du système de fichiers vers les machines maître et confiance. En plus des droits d'écriture, toutes conversions d'UID est abandonnée pour l'hôte confiance. La deuxième et la troisième lignes montrent des exemples de noms d'hôtes génériques.

La quatrième ligne montre une entrée pour le client PC/NFS présenté plus haut.

La dernière ligne exporte un répertoire public de FTP, à tous les hôtes dans le monde, en effectuant les requêtes sous le compte anonyme. L'option insecure permet l'accès aux clients dont l'implémentassion NFS n'utilise pas un port réservé.

Sous OpenBSD

La philosophie reste la même néanmoins on va observer quelques petits changements dont voici un exemple :

```
/home -alldirs -mapall=root -network 192.168.0.1 -mask 255.255.255.0
```

avec "-alldirs" pour autoriser les clients a monter n'importe quel sous répertoire. "-mapall=root" pour que toutes les modifications soient considérées comme étant faite par root. "-network 192.168.0.1" pour n'autoriser que "192.168.0.1" comme client. "-mask 255.255.255.0" pour spécifier le masque de réseaux.

utilisation

En tant que client on peut monter un volume nfs par la commande :
mount -t nfs 192.168.1.217 :/home/nfs /nfs
on peut aussi l'inclure directement dans le fichier */etc/fstab* :
server :/usr/local/pub /pub nfs rsize=8192,wsize=8192,timeo=14,intr

sécurisation

Le problème de NFS est que la seule limitation que l'on peut appliquer au niveau du réseau est basé sur l'adresse IP du client, qui peut être usurpée par la technique du spoofing. Il est donc indispensable de renforcer cette sécurité par des règles de afin d'en limiter l'accès aux réseaux dont on peut avoir confiance.

5.14 installation d’un serveur NIS

5.14.1 sous NetBSD

modification du fichier : /var/yp/Makefile.yp

```
DIR=/etc/NIS
Ensuite on remplace la ligne :
all : passwd aliases $AMDMAPS ethers group hosts ipnodes netgroup
networks rpc services protocols netid

par :
all : passwd aliases group hosts netgroup rpc services protocols netid
```

ensuite

```
mkdir /etc/NIS
cd /etc
cp passwd master.passwd hosts group rpc services protocols NIS
```

bien sur il ne faut pas oublier de définir le domaine s’il n’existe pas

```
domainename mondomain.fr
```

```
ypinit -m mondomain.fr

ensuite :
cd /var/yp
make
```

enfin pour /etc/rc.conf, afin que le serveur puisse être démarrer en même temps que le système.

```
rpcbind=YES
ypbind=YES
ypserv=YES
yppasswdd=YES
```

5.15 installation d'un serveur LDAP

5.16 installation d'un serveur CVS

5.17 installation d'un DNS

Le système bind (Berkeley Internet Named Domain) est une implémentassion de DNS (Domain Name Server), qui permet de traduire les noms des machines en adresses IP, par exemple lorsque vous tapez www.wanadoo.fr votre navigateur va faire une recherche DNS pour obtenir :

```
;; ANSWER SECTION:
wanadoo.fr.             55083   IN      A       193.252.19.142
```

Nous verrons ainsi dans cette section comment configurer un tel serveur en prenant pour exemple un cas concret.

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh bind-*.rpm	servine named start
Mandrake	urpmi bind	/etc/init.d/named start
Mandrake	apt-get install bind9	/etc/init.d/bind9 start
OpenBSD	pkg_add bind-9*.tgz	/usr/local/sbin/named -u named -t /var/namedb

Remarque :
OpenBSD : bind-4 est installé par defaut néanmoins il est préférable d'installer la version 9.2.

configuration

named.conf

```
options {
directory "/namedb" ;
} ;

zone "." in {
type hint ;
file "root.cache" ;
} ;

zone "127.IN-ADDR.ARPA" {
type master ;
notify no ;
file "127.rev" ;
} ;

zone "wanadoo.fr" {
type master ;
notify no ;
file "wanadoo.fr" ;
} ;

zone "0.168.192.in-addr.arpa" {
type master ;
notify no ;
file "0.168.192.rev" ;
} ;
```

wanadoo.fr

```
$TTL 3600
@ IN SOA wanadoo.fr. hostmaster.wanadoo.fr. (
200202049 ; serial, todays date + todays serial
8H ; refresh, seconds
2H ; retry, seconds
1W ; expire, seconds
1D ) ; minimum, seconds
;
NS wanadoo.fr.
MX 10 mail.wanadoo.fr.
TXT "wanadoo.fr... "

localhost A 127.0.0.1

petra A 192.168.0.1

wanadoo.fr. A 212.198.192.215

ns CNAME wanadoo.fr.
www CNAME wanadoo.fr.
imap CNAME wanadoo.fr.
pop CNAME wanadoo.fr.
smtp CNAME wanadoo.fr.
```

0.168.192.rev

```
$TTL 86400 ; 1 day
@ IN SOA petra.wanadoo.fr. root.wanadoo.fr. (
200202044 ; Serial
8H ; Refresh
2H ; Retry
1W ; Expire
1D ) ; Minimum
IN NS petra.wanadoo.fr.
1 IN PTR petra.wanadoo.fr.
```

5.18 Mise en place d'un VPN

Pour se faire nous utiliserons un programme très utilisé, stable et sécurisé(128 bits) nommé vtun. Pour de amples informations rendez-vous sur http ://vtun.sourceforge.net

Système	commande d'installation	démarrage du programme
RedHat	rpm -Uvh vtun-*.rpm	service vtund start
Mandrake	urpmi vtun	/etc/init.d/vtund start
Debian	apt-get install vtun	/etc/init.d/vtund start
OpenBSD	pkg_add vtun.*.tgz	

5.19 les fichiers hosts.allow et hosts.deny

Comme dans toute politique de défense correcte, il est préférable de tout interdire puis d'autoriser service par service.

```
(root@petra)[/etc]-# cat hosts.deny
ALL: ALLl
```

ALL : 192.168.0.3	On autorise tous les services pour les postes
ALL : 192.168.0.35	192.168.0.3 et 192.168.0.35
smtp : ALL	accès depuis tous les postes.
sshd : ALL	
http : ALL	
sendmail : ALL	
domain : ALL	
portmap : 192.168.0.0/255.255.255.0	depuis tous les postes du réseau 192.168.0

Pour tester ces règle on pourra utiliser */usr/sbin/tcpdmatch*

```
$> tcpdmatch sshd www.wanadoo.fr
  ...
  access: granted

ou encore

$> tcpdmatch in.telnetd www.wanadoo.fr
  ...
  access: denied
```

5.20 installation d'un Firewall

Avant de commencer il faudra différencier plusieurs cas :

- le premier est la configuration du firewall sous kernel 2.2* auquel cas il faudra utiliser “ipchains”.
- le second sous kernel 2.4* auquel cas il faudra utiliser “netfilter” avec “iptables”.
- enfin sous OpenBSD3.0 avec l’utilisation de pf (packet filter and NAT subsystems).

Dans le cas d’une version de Mandrake > Mandrake8.0, on utilisera netfilter (iptables). Le firewall agit comme un goulot d’étranglement par lequel toutes les machines d’un réseau doivent passer pour communiquer avec l’extérieur.

Il existe deux grandes familles de firewalls :

les firewalls IP ou filtrants ils fonctionnent au niveau paquet (niveau 3 du modèle OSI). Ils sont conçus pour contrôler le flux de paquets en fonction de l’adresse IP d’origine, l’adresse IP de destination, ainsi que le port et l’information de type de paquet contenue dans chacun de ceux-ci. Ils permettent une bonne sécurité face aux différentes attaques.

les firewalls dits “proxy” ils fonctionnement au niveau applicatif, et font passer les requêtes d’un réseau à l’autre : ils permettent ainsi l’accès indirect a internet. L’inconvénient ici est que chaque utilisateur doit configurer manuellement son navigateur web ou son client FTP pour communiquer avec l’extérieur via le proxy.

Au contraire, un firewall filtrant permet un accès transparent pour l’utilisateur, tout en assurant une bonne sécurité, puisque l’on peut totalement contrôler le flux qui le traverse. Nous nous concentrerons essentiellement sur le premier type de firewall. Les proxys seront présentés dans la dernière partie de l’exposé traitant des autres développements applicatifs possibles sous Linux.

Configuration

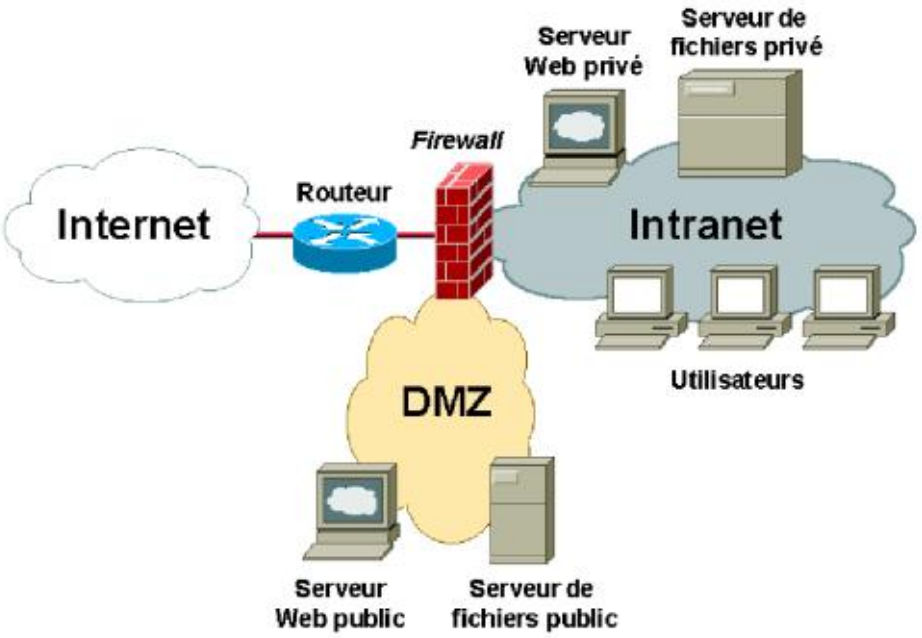
Architecture La figure précédente représente une architecture classique de réseau d’entreprise derrière un firewall.

On y distingue :

1. Internet, auquel l’entreprise est reliée par le routeur du fournisseur d’accès,
2. une DMZ (DeMilitarised Zone ou zone démilitarisée) qui contient les serveurs publics de l’entreprise : serveur web internez, serveur mail, serveur de fichiers public etc.,
3. l’intranet de l’entreprise, zone privée et protégée de l’entreprise.

Physiquement, la machine sur laquelle tourne le firewall comporte trois cartes Ethernet, connectées au routeur, au LAN de la DMZ, et au LAN privé de l’entreprise.

nous supposerons pour le reste de cette section que le réseau a pour paramètres :



- address : 192.168.1.1
- netmask : 255.255.255.0
- network : 192.168.1.0
- broadcoast : 192.168.1.255

5.20.1 Configuration de Netfilter (IPtables)

Installation

Système	commande d'installation	démarrage du service
RedHat	rpm -Uvh iptables-*.rpm	/etc/init.d/iptables start
Mandrake	urpmi iptables	/etc/init.d/iptables start
Debian	il faudra utiliser un noyau 2.4	
OpenBSD	iptables n'existe pas sous OpenBSD	

Le problème est que le firewall par defaut sur Mandrake et RedHat est pour des raisons de compatibilité ipchains. Il va donc être nécessaire d'arrêter "ipchains" avant de pouvoir utiliser "iptables", pour se faire :

```
/etc/init.d/ipchains stop
rmmod ipchains
```

Afin de vérifier que le noyau supporte a été compilé avec l'option Netfilter, il faut vérifier que ip_conntrack & ip_tables apparaissent bien avec la commande "dmesg".
dmesg | grep ip_conntrack

dmesg | grep ip_tables

sinon :

```
# modprobe ip_tables
# modprobe ip_nat_ftp
# modprobe ip_nat_irc
# modprobe iptable_filter
# modprobe iptable_mangle
# modprobe iptable_nat
```

Si on a besoin de pouvoir forwarder les paquets IP (dans la plupart des cas), il sera nécessaire d'exécuter cette commande :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Utilisation

Dans la configuration de base les paquets passent par trois chaînes de règles :

- INPUT : par laquelle passent tous les paquets entrant par une interface.
- FORWARD : par laquelle passent tous les paquets qui sont transmis d'une interface à une autre.
- OUTPUT : par laquelle passent tous les paquets avant de sortir d'une interface.

Les quatres comportements les plus courants sont :

- ACCEPT : On laisse passer le paquet.
- REJECT : On rejette le paquet et on envoie le paquet d'erreur associé.
- LOG : Enregistre une notification du paquet dans syslog.
- DROP : Le paquet est ignoré aucune réponse n'est envoyé.

Voici les principales options de iptables :

- N : Création d'une nouvelle chaîne.
- X : Suppression d'une chaîne vide.
- P : Changement de la politique par défaut d'une chaîne.
- L : liste les règles d'une chaîne.
- F : Élimine toutes les règles d'une chaîne.

Enfin pour les modifications de chaînes :

- A : ajoute une règle à la fin d'une chaîne.
- I : Insère une nouvelle règle a une position donnée.
- R : Remplace une règle donnée dans une chaîne.
- D : Efface une règle dans une chaîne (numéro d'ordre ou règle)

Exemples

Pour nater sur l'interface eth0 :

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Pour créer un proxy transparent :
iptables -t nat -A PREROUTING -s ! 192.168.1.1
-i eth0 -p tcp –dport 80 -j REDIRECT –to-port 3128

Pour supprimer les règles actives :
iptables -F
iptables -F -t nat
iptables -X

Afin d'empêcher les ping des autres machines :
iptables -A INPUT -s 192.168.1.217 -p icmp –icmp-type echo-reply -j DROP

Afin de supprimer la première règle :
iptables -D INPUT 1

Afin de refuser tout trafic TCP sauf ce qui provient de l'adresse IP 192.168.1.217 sera traduite par la commande suivante :
iptables -A INPUT -p tcp –source ! 192.168.1.217 -j DENY

Afin de spécifier un protocole : tcp, udp, icmp, all (tous) -> -p –protocol
iptables -A INPUT -p icmp -j DENY

Afin de spécifier la source : -s –source
iptables -A INPUT -p tcp -s 192.168.42.42 -j ACCEPT

Afin de spécifier une adresse destination : -d –destination
iptables -A FORWARD -p tcp -d 10.1.0.1 -j ACCEPT

Afin de spécifier une interface d'entrée : -i –in-interface
iptables -A INPUT -p icmp -i eth0 -j DENY

Afin de spécifier une interface de sortie : -o –out-interface
iptables -A OUTPUT -p icmp -o eth0 -j DENY

Afin de spécifier le port source ou une plage de ports : –sport –source-port :
iptables -A INPUT -p tcp –sport 80 -j ACCEPT
iptables -A INPUT -p udp –sport 80 -j DROP
iptables -A OUTPUT -p tcp -m multiport –sport 3128,21,1000 -j DROP
iptables -A OUTPUT -p tcp –sport 1024 :2042 -j ACCEPT

Afin de spécifier le port destination : –dport –destination-port

iptables -A INPUT -p tcp –dport 110 -j DENY
iptables -A INPUT -p udp –dport 110 -j DENY
iptables -A INPUT -p tcp -m multiport –dport 110,4242,119 -j DROP
iptables -A INPUT -p tcp –sport 4925 :4633 -j ACCEPT

Pour rediriger sur la DMZ :
iptables -t nat -A PREROUTING -d 42.42.42.42
-p tcp –dport 80 -j DNAT –to-destination 192.168.1.2 :80

Règle SSH :
iptables -A INPUT -i eth1 -s 192.168.2.42 -m state
–state NEW,ESTABLISHED -p tcp –dport 22 -j LOG_ACCEPT
iptables -A OUTPUT -o eth1 -d 192.168.2.42 -m state
–state ESTABLISHED -p tc –sport 22 -j LOG_ACCEPT

```
################ exemple de conf pour iptables ######################
# !/bin/sh

 ## Information sur le réseau.
INTERNALIF="eth1"  # carte donnant sur le réseau interne.
INTERNALNET="192.168.1.0/24"  # IP Réseau Interne.
INTERNALBCAST="192.168.1.255"  # IP Broadcast.
EXTERNALIF="eth0"  #carte donnant sur l'extérieur.
EXTERNALIP="1.2.3.4" # Adresse IP externe nécessaire en cas de DNAT

 ## Suppression de toutes les règles :
iptables -F INPUT # règles sur les paquets entrants
iptables -F OUTPUT # règles sur les paquets sortants
iptables -F FORWARD # règles sur le Forwarding/masquerading
iptables -t nat -F # règles sur le Nat

 ##Modification des règles tcp/ip du noyau.
 #Disabling IP Spoofing attacks.
echo 2 > /proc/sys/net/ipv4/conf/all/rp_filter
 # Ne pas répondre aux pings broadcast
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
 # autorisation du forwarding
echo 1 >/proc/sys/net/ipv4/ip_forward
 # Blockage routage
echo 0 >/proc/sys/net/ipv4/conf/all/accept_source_route
```

```
 # Suppression des timestamps.
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
 # autorisation des SYN Cookies
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
 # empêche les redirections
echo 0 >/proc/sys/net/ipv4/conf/all/accept_redirects
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
 # autorise les adresses IP dynamiques
echo "1" > /proc/sys/net/ipv4/ip_dynaddr
# Log pacquets avec des adresses impossibles.
echo 1 >/proc/sys/net/ipv4/conf/all/log_martians
#Set out local port range
echo "32768 61000" >/proc/sys/net/ipv4/ip_local_port_range


# Réduit les DoS en réduisant les timeouts
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 1800 > /proc/sys/net/ipv4/tcp_keepalive_time
echo 1 > /proc/sys/net/ipv4/tcp_window_scaling
echo 0 > /proc/sys/net/ipv4/tcp_sack
echo 1280 > /proc/sys/net/ipv4/tcp_max_syn_backlog

## Mise en place des règles de base.
# rejet des mauvais paquets : trop courts, les paquets
# TCP et UDP ayant zéro comme source ou comme destination,
# taille nulle ou trop grande, paquets fragmentés
# pour plus dinfo http ://www.linux-mag.com/2000-01/bestdefense_02.html

iptables -A INPUT -m unclean -j DROP
iptables -A FORWARD -m unclean -j DROP
iptables -A INPUT -m state –state INVALID -j DROP
iptables -A FORWARD -m state –state INVALID -j DROP

# autorise toutes les connections sur l'interface interne
iptables -A INPUT -i lo -j ACCEPT

# Refuse les connections depuis l'interface interne vers l'extérieur.
iptables -A INPUT -d 127.0.0.0/8 -j REJECT

# traffic illimité depuis le réseau interne.
iptables -A INPUT -i $INTERNALIF -s $INTERNALNET -j ACCEPT

# autorise tunnel IPV6.
```

```
#iptables -A INPUT -p ipv6 -j ACCEPT

# autorise tunnel IPSEC.
#iptables -A INPUT -p 50 -j ACCEPT
# autorise tous paquets depuis le serveur ipsec vers le réseau interne.
#iptables -A FORWARD -i ipsec0 -o $INTERNALIF -j ACCEPT

# Refuse tous paquets depuis l'extérieur prétendant être du réseau interne.
iptables -A INPUT -i $EXTERNALIF -s $INTERNALNET -j REJECT

## ICMP
# ne pas faire suivre les pings de l'extérieur vers le réseau interne.
iptables -A FORWARD -p icmp –icmp-type echo-request -o $INTERNALIF -j REJECT

#protection contre le ping flood.
iptables -A INPUT -p icmp –icmp-type echo-request -m limit –limit 1/s -j ACCEPT
iptables -A INPUT -p icmp –icmp-type echo-request -j DROP


# Refuse l'icmp vers une adresse de broadcast.
iptables -A INPUT -p icmp -d $INTERNALBCAST -j DROP

# autorise tout autre icmp.
iptables -A INPUT -p icmp -j ACCEPT

# autorise les connections établies.
iptables -A INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

# Refuse de rediriger les paquets samba
iptables -A FORWARD -o $EXTERNALIF -p tcp –dport 137 -j REJECT
iptables -A FORWARD -o $EXTERNALIF -p tcp –dport 138 -j REJECT
iptables -A FORWARD -o $EXTERNALIF -p tcp –dport 139 -j REJECT
iptables -A FORWARD -o $EXTERNALIF -p udp –dport 137 -j REJECT
iptables -A FORWARD -o $EXTERNALIF -p udp –dport 138 -j REJECT
iptables -A FORWARD -o $EXTERNALIF -p udp –dport 139 -j REJECT
iptables -A INPUT -i $EXTERNALIF -p udp –dport 137 -j REJECT

# Autorise tous les autres paquets a être forwardé
iptables -A FORWARD -o $EXTERNALIF -i $INTERNALIF -j ACCEPT

iptables -A FORWARD -i $EXTERNALIF -m state
–state ESTABLISHED,RELATED -j ACCEPT
```

```
# autorise les requêtes DHCP en tant que client.
#iptables -A INPUT -p udp -d 255.255.255.255 –dport 68 -j ACCEPT

# autorise les requêtes DHCP en tant que serveur.
#iptables -A INPUT -i $INTERNALIF -p tcp –sport 68 –dport 67 -j ACCEPT
#iptables -A INPUT -i $INTERNALIF -p udp –sport 68 –dport 67 -j ACCEPT

# autorise les requêtes vers les serveur DNS contenus
# dans /etc/resolv.conf :
#cat /etc/resolv.conf |
 #awk '/ñameserver/ {print $2}' |
#xargs -n1 iptables -A INPUT -p udp –sport 53 -j ACCEPT -s

# autorise les paquets entrant pour les différents services listés :
iptables -A INPUT -p tcp –dport 20 -j ACCEPT # ftp-data
iptables -A INPUT -p tcp –dport 21 -j ACCEPT # ftp
iptables -A INPUT -p tcp –dport 22 -j ACCEPT # ssh
#iptables -A INPUT -p tcp –dport 23 -j ACCEPT #telnet

# autorise les paquets à destination du serveur de mail mais les limitent
# à 1 par seconde pour éviter les attaques de type DoS.
iptables -A INPUT -p tcp –dport 25 –syn -m limit –limit 1/s
–limit-burst 10 -j ACCEPT
iptables -A INPUT -p tcp –dport 25 –syn -j DROP
iptables -A INPUT -p tcp –dport 25 -j ACCEPT
# DNS
iptables -A INPUT -p tcp –dport 53 -j ACCEPT
iptables -A INPUT -p udp –dport 53 -j ACCEPT
# http
iptables -A INPUT -p tcp –dport 80 -j ACCEPT
# POP3
#iptables -A INPUT -p tcp –dport 110 -j ACCEPT
# Imaps
iptables -A INPUT -p tcp –dport 993 -j ACCEPT
# identd
#iptables -A INPUT -p tcp –dport 113 -j ACCEPT
# https
iptables -A INPUT -p tcp –dport 443 -j ACCEPT
#pserser
iptables -A INPUT -p tcp –dport 2401 -j ACCEPT
# autorise les paquets à destination du serveur VNC ce qui n'est pas forcement
# une bonne idée.
#iptables -A INPUT -p tcp –dport 5801 -j ACCEPT
```

```
#iptables -A INPUT -p tcp –dport 5901 -j ACCEPT
#iptables -A INPUT -p tcp –dport 6001 -j ACCEPT

## DNAT
#iptables -A PREROUTING -t nat -i $EXTERNALIF -p tcp
# -d $EXTERNALIP –dport 80 -j DNAT –to 192.168.0.10 :80
#iptables -A FORWARD -i $EXTERNALIF -p tcp -d 192.168.0.10 –dport 80 -j ACCEPT
#iptables -A PREROUTING -t nat -i $EXTERNALIF -p tcp
# -d $EXTERNALIP –dport 25 -j DNAT –to 192.168.0.10 :25
#iptables -A FORWARD -i $EXTERNALIF -p tcp -d 192.168.0.10 –dport 25 -j ACCEPT

# Certains services doivent être loggés et refusés
#iptables -A INPUT -p tcp –dport 1433 -m limit -j LOG
# –log-prefix "Firewalled packet : MSSQL "
#iptables -A INPUT -p tcp –dport 1433 -j DROP
#iptables -A INPUT -p tcp –dport 6670 -m limit -j LOG
# –log-prefix "Firewalled packet : Deepthrt "
#iptables -A INPUT -p tcp –dport 6670 -j DROP
#iptables -A INPUT -p tcp –dport 6711 -m limit -j LOG
# –log-prefix "Firewalled packet : Sub7 "
#iptables -A INPUT -p tcp –dport 6711 -j DROP
# –log-prefix "Firewalled packet : BO "
#iptables -A INPUT -p tcp –dport 31337 -j DROP
iptables -A INPUT -p tcp –dport 6000 -m limit -j LOG
–log-prefix "Firewalled packet : XWin "
iptables -A INPUT -p tcp –dport 6000 -j DROP

# traceroutes
iptables -A INPUT -p udp –dport 33434 :33523 -j DROP

iptables -A INPUT -p tcp –dport 113 -j REJECT

# Ne pas logger les paquets igmp
iptables -A INPUT -p igmp -j REJECT

# Ne pas logger les requêtes http{s}
iptables -A INPUT -p tcp –dport 80 -j REJECT
iptables -A INPUT -p tcp –dport 443 -j REJECT

# Si les paquets ne correspondent a aucunes des règles on les
# log et on les rejettes
iptables -A INPUT -p tcp –syn -m limit –limit 5/minute -j LOG
–log-prefix "Firewalled packet :"
```

```
iptables -A FORWARD -p tcp –syn -m limit –limit 5/minute -j LOG
–log-prefix "Firewalled packet :"
# Rejet
iptables -A INPUT -p tcp -j REJECT –reject-with tcp-reset
iptables -A INPUT -p all -j DROP
iptables -A FORWARD -p tcp -j REJECT –reject-with tcp-reset
iptables -A FORWARD -p all -j DROP

# autorise tout de même si elles se dirigent vers l'extérieur
iptables -A OUTPUT -j ACCEPT

# Masquerage des connections internes vers l'extérieur
iptables -A POSTROUTING -t nat -o $EXTERNALIF -j MASQUERADE

exit 0
```

Exemple de script pour iptables

5.20.2 Configuration de pf (packet filter)

Pour toute information complémentaire vous pourrez consulter le OpenBSD Packet Filter HOWTO à l'adresse : http ://www.inebriated.demon.nl/pf-howto/html

commandes de base

Pour activer vos règles aprés avoir complété le fichier de configuration */etc/pf.conf*

activation pfctl -f /etc/pf.conf

désactivation pfctl -F all

```
################ exemple d'un fichier /etc/pf.conf ######################
ext_if="xl0"
int_if="vr0"
unroutable="{ 127.0.0.0/8, 10.0.0.0/8, 172.16.0.0/12,
 192.168.0.0/16, 255.255.255.255/32 }"
services_tcp="{ smtp, submission, domain, auth, www, https, > 1024 }"
services_udp="{ domain }"
ipv6_net="{ 2001 :470 :1f00 :ffff : :245, 2001 :470 :1f00 :390 :0 :0 :0 :0/64 }"

# See pf.conf(5) for syntax and examples
# My external interface is kue0 (62.65.145.30, my only routable address) and
# the private network is 10.0.0.0/8, for which i'm doing NAT. There's an
```

```
# IPv6 tunnel, too.

# normalize all packets
scrub out all
scrub in all

# nat private network to single routable address
nat on $ext_if inet from 192.168.1.0/24 to any -> $ext_if

# redirect https connections from work to sshd
#rdr on $ext_if inet proto tcp from 0.0.0.0
# to $ext_if port 443 -> $ext_if port 22
#rdr on $int_if inet proto tcp from 0.0.0.0
# to $int_if port 443 -> $ext_if port 22

# block and log everything by default
block out log all
block in log all
block return-rst out log inet proto tcp all
block return-rst in log inet proto tcp all
block return-icmp out log inet proto udp all
block return-icmp in log inet proto udp all

# unfiltered interfaces
pass out quick on { lo0, enc0, $int_if } all
pass in quick on { lo0, enc0, $int_if } all

# ================================================
# common rules for all filtered interfaces
# ================================================

# silently drop noise
block return-rst in quick proto tcp from any to any
port { 111, 6000, 6667 }
block return-icmp in quick proto udp from any to any
port { 137 }

# silently drop TCP non-SYN packets (only SYNs create state)
block out quick proto tcp all flags /S
block in quick proto tcp all flags /S

# ================================================
# external interface (all external IPv4 traffic)
```

```
# ============================================

# block and log outgoing packets that don't have my address as source, they are
# either spoofed or something is misconfigured (NAT disabled, for instance),
# we want to be nice and don't send out garbage.
block out log quick on $ext_if inet from !$ext_if to any

# silently drop broadcasts (ADSL noise)
block in quick on $ext_if inet from any to { 255.255.255.255, 62.65.145.31 }

# block and log incoming packets from reserved address space and invalid
# addresses, they are either spoofed or misconfigured, we can't reply to
# them anyway (hence, no return-rst).
block in log quick on $ext_if inet from $unroutable to any

# ICMP
pass out on $ext_if inet proto icmp from $ext_if to any
 icmp-type 8 code 0 keep state
pass in on $ext_if inet proto icmp from any to $ext_if
 icmp-type 8 code 0 keep state

# UDP
pass out on $ext_if inet proto udp from $ext_if to any
 keep state
pass in on $ext_if inet proto udp from any to $ext_if
 port $services_udp keep state

# TCP
pass out on $ext_if inet proto tcp from $ext_if to any
 flags S/SA keep state
pass in on $ext_if inet proto tcp from any to $ext_if
 port $services_tcp flags S/SA keep state

# other protocols (IPv6 tunnel)
pass out on $ext_if inet proto ipv6 from $ext_if to 64.71.128.82 keep state
pass in on $ext_if inet proto ipv6 from 64.71.128.82 to $ext_if keep state

# ============================================
# tunnel interface (all external IPv6 traffic)
# ============================================

# ICMP
pass out on gif0 inet6 proto ipv6-icmp from $ipv6_net to any
```

```
 ipv6-icmp-type echoreq keep state
pass in on gif0 inet6 proto ipv6-icmp from any to $ipv6_net
 ipv6-icmp-type echoreq keep state

# UDP
pass out on gif0 inet6 proto udp from $ipv6_net to any keep state
pass in on gif0 inet6 proto udp from any to $ipv6_net
 port $services_udp keep state

# TCP
pass out on gif0 inet6 proto tcp from $ipv6_net to any flags S/SA keep state
pass in on gif0 inet6 proto tcp from any to $ipv6_net
port $services_tcp flags S/SA keep state
```

5.20.3 Configuration du filtrage IP (IPchains)

Installation

RedHat	rpm -Uvh ipchains-*.rpm
Mandrake	urpmi ipchains

Les ipchains Linux sont une réécriture du code pare-feu IPv4 de Linux (en grande partie inspiré de BSD) ainsi que de ipfwadm qui était lui-même une réécriture du ipfw de BSD, je crois. Il est indispensable d'administrer les filtres de paquets IP dans les noyaux Linux versions 2.1.102 et au-delà.

On travaille avec les chaînes par noms. Il existe au début trois chaînes prédéfinies input, output et forward, qu'il est impossible de supprimer. On peut créer des chaînes personnelles. Des règles peuvent ensuite être ajoutées et supprimées de ces ensembles de règles.

Les opérations nécessaires pour travailler sur les chaînes sont les suivantes :

1. création d'une nouvelle chaîne (-N) ;
2. suppression d'une chaîne vide (-X) ;
3. modification de la spécification pour une chaîne prédéfinie (-P)
4. liste des règles d'une chaîne (-F) ;
5. remise à zéro des compteurs de paquets et d'octets de toutes les règles d'une chaîne (-Z).

Il existe différentes méthodes pour manipuler les règles à l'intérieur des chaînes :

1. ajout d'une nouvelle règle dans une chaîne (-A) ;
2. insertion d'une nouvelle règle à une position donnée d'une chaîne (-I) ;
3. remplacement d'une règle à une position donnée d'une chaîne (-R) ;
4. suppression d'une règle à une position donnée d'une chaîne (-D) ;

5. suppression de la première règle correspondante dans une chaîne (-D).

Il existe quelques opérations pour le masquage qui se trouvent dans ipchains dans l'attente d'un bon emplacement pour les placer :

1. liste des connexions actuellement masquées (-M -L) ;
2. positionnement du temps d'expiration du masquage (-M -S).

Il y a quelques détails de chronologie qui interviennent dans la modification des règles de pare-feu. Si l'on n'est pas suffisamment prudent, il est possible de laisser passer quelques paquets pendant la mise en place des modifications. Une approche simpliste est la suivante :

```
# ipchains -I input 1 -j DENY
# ipchains -I output 1 -j DENY
# ipchains -I forward 1 -j DENY

... réalisation des modification ...

# ipchains -D input 1
# ipchains -D output 1
# ipchains -D forward 1
```

Cela interdit tout passage de paquet durant les modifications.
Voici une copie des règles de pare-feu qui précèdent dans IPchains :

```
# !/bin/sh
#
# rc.firewall
#
## Tout vider et repartir du debut
/sbin/ipchains -F input
/sbin/ipchains -F output
/sbin/ipchains -F forward

## Redirection pour le mandatement transparent de HTTP
#$IPCHAINS -A input -p tcp -s 192.1.2.0/24 -d 0/0 80 -j REDIRECT 8080

## Création de nos propres chaînes
/sbin/ipchains -N ma-chaine
# On autorise le courriel entrant vers le serveur
/sbin/ipchains -A ma-chaine -s 0.0.0.0/0 smtp -d 192.1.2.10 1024 :-j ACCEPT
# On autorise les connexions courriel vers l'extérieur
/sbin/ipchains -A ma-chaine -s 192.1.2.10 -d 0.0.0.0/0 smtp -j ACCEPT
# On autorise les connexions web entrantes vers le serveur
/sbin/ipchains -A ma-chaine -s 0.0.0.0/0 www -d 192.1.2.11 1024 : -j ACCEPT
```

```
# On autorise les connexions web vers l'extérieur
/sbin/ipchains -A ma-chaine -s 192.1.2.0/24 1024 : -d 0.0.0.0/0 www -j ACCEPT
# On autorise le trafic DNS
/sbin/ipchains -A ma-chaine -p UDP -s 0.0.0.0/0 dns -d 192.1.2.0/24 -j ACCEPT

## Si on utilise le masquage
# On ne masque pas le trafic interne
/sbin/ipchains -A forward -s 192.1.2.0/24 -d 192.1.2.0/24 -j ACCEPT
# On ne masque pas l'interface externe directe
/sbin/ipchains -A forward -s 199.1.2.0/24 -d 0/0 -j ACCEPT
# On masque tout paquet interne qui sort
/sbin/ipchains -A forward -s 192.1.2.0/24 -d 0/0 -j MASQ

## On interdit tout le reste
/sbin/ipchains -P my-chains input DENY
```

Il ne faut pas s'arrêter là. Ce n'est pas un pare-feu très puissant et je suis sûr qu'il y a d'autres services que vous souhaiteriez fournir. à nouveau, lisez le IPCHAINS-HOWTO.

5.21 la translation d'adresses : NAT ou IPmasquerade

La fonction de translation d'adresses est très utile pour rendre invisibles de l'extérieur les machines d'un réseau local ayant un plan d'adressage privé. Cette fonction est généralement implémentée sur un routeur ou éventuellement un firewall dans le cas d'un système fédéré. Ce dernier apparaît comme étant alors le seul système utilisant la connexion internet. Ceci permet d'optimiser l'utilisation du plan d'adressage, avec un coût moindre, car il suffit d'avoir une seule connexion à internet pour plusieurs machines.

La différence entre la NAT et un proxy applicatif réside dans le fait qu'un proxy agit comme un intermédiaire entre Internet et les machines connectées au LAN. C'est très bien, cependant chaque application que vous voudrez exécuter sur votre machine et qui devra se connecter à Internet à travers le serveur proxy DEVRA savoir comment utiliser un proxy. Toutes les applications ne savent pas faire ça (et spécialement les jeux). De plus, il n'y a pas des proxies applicatifs pour tous les services Internet actuels. La NAT fait correspondre de manière transparente votre réseau interne à une adresse donnée pour qu'il puisse se connecter à Internet. Le seul avantage en termes de sécurité à utiliser un proxy par rapport à la NAT est que le proxy peut posséder des fonctionnalités de sécurité, et peut filtrer le trafic d'après le contenu, pour protéger votre machine Windows contre un virus macro, ou pour protéger votre logiciel client contre les débordements de tampon...etc. La maintenance de ce type de filtres est sou-

vent lourde.

En terme de sécurité réseau, le protocole NAT :

1. est transparent pour les ordinateurs du domaine,
2. cache l'ensemble des machines du réseau derrière la passerelle, qui apparaît comme étant le seul système connecté à Internet,
3. permet une sécurité accrue, car il n'existe aucun moyen pour quelqu'un de l'extérieur, d'accéder aux machines internes. En effet l'extérieur ne voit qu'une seule machine : la passerelle sur laquelle est implémentée NAT.

La machine Linux sur laquelle sera installé le protocole NAT devra réaliser les fonctionnalités suivantes :

1. la table NAT doit avoir une entrée pour chaque association statique, afin que chaque poste puisse avoir accès à Internet,
2. Une double traduction est effectuée par le routeur NAT :
3. dans le sens interne vers externe : adresse locale interne vers adresse globale interne adresse globale externe vers adresse locale externe
4. dans le sens externe vers interne : adresse locale externe vers adresse globale externe adresse globale interne vers adresse locale interne
5. l'administrateur spécifie le pool d'adresses locales internes à traduire et le pool d'adresses globales internes,
6. chaque fois qu'un paquet avec une adresse IP locale franchit le routeur NAT, il reçoit une adresse prise dans le pool des adresses globales,
7. cette traduction reste effective tant que la table NAT n'est pas nettoyée ou que la période de traduction n'est pas expirée. Dans ce cas, un paquet provenant du même hôte doit obtenir une nouvelle adresse du pool d'adresses globales.

Avantages et inconvénients de NAT

Du point de vue de la sécurité, NAT cache l'identité réelle des machines. Il n'est pas possible d'identifier l'identité d'un paquet dont l'adresse IP a été traduite de façon dynamique. Cet avantage est intéressant pour les sites désirant se protéger de l'extérieur.

Par contre, NAT "casse" le modèle IP de bout en bout : la perte de traçabilité de bout en bout rend la recherche impossible en cas de piratage provoquée par une machine interne.

De manière générale, toute application spécifiant l'adresse IP dans la partie des données (et non pas dans l'en-tête) ne fonctionne pas avec NAT.

NAT réduit les possibilités de communications sécurisées : impossible de traduire des données cryptées. Il faut dans ce cas utiliser l'adressage statique avec une adresse globale.

Enfin, le multicast n'est actuellement pas supporté par NAT : il est possible de recevoir du trafic multicast mais une source interne ne peut pas émettre. Une parade consiste à contourner le NAT, avec l'utilisation du mode "sparse" du protocole PIM (Protocol Independent Multicast) : identification de la source (interne) à un point de rendez-vous ("Core") situé à l'extérieur du réseau local protégé par NAT. PIM "Sparse Mode" est analogue au protocole CBT (Core-Based Tree).

Configuration

On supposera que votre réseau local est en 192.168.1.X

Première étape Éditer le fichier /etc/sysconfig/network

```
NETWORKING=yes
FORWARD_IPV4=true
HOSTNAME=tavel.kervao.fr
DOMAINNAME=kervao.fr
GATEWAY=
GATEWAYDEV=
```

noyau 2.2.x

1. éditer/créez le fichier /etc/rc.d/rc.firewall (ou /etc/rc.d/rc.local si diffèrent de Mandrake) et ajouter à la fin, les lignes suivantes :

   ```
   ipchains -P forward DENY
   ipchains -A forward -s 192.168.13.0/24 -j MASQ
   ```

2. Reboutez votre machine
3. Pour voir si l'IP Masquerade est bien pris en compte, tapez :

   ipchains -L forward

   Vous devriez avoir :

   ```
   Chain forward (policy DENY):
   target     prot opt    source               destination          ports
   MASQ       all  ------  192.168.13.0/24      anywhere             n/a
   ```

Par ailleurs le fichier */proc/sys/net/ipv4/ip_forward* doit contenir un 1.

noyau 2.4.x

Pour les noyaux 2.4.X il vous faudra installer le package iptables : (urpmi iptables)

Créer un fichier /etc/rc.d/rc.firewall contenant :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
iptables -A FORWARD -i eth2 -o eth1 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp -d 0.0.0.0/0 --dport 80 -j DNAT --to 192.168.2.178:3128
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

ou

#!/bin/sh
# Load the NAT module (this pulls in all the others).
modprobe iptable_nat

# Turn on IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# In the NAT table (-t nat), Append a rule (-A) after routing (POSTROUTING)
# which says to MASQUERADE the connection (-j MASQUERADE).
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE

# Allows forwarding specifically to our LAN
iptables -A FORWARD -s 192.168.1.0/24 -j ACCEPT
```

Pour prendre en compte les modifs vous pouvez exécuter le fichier qui a des droits d'exécution (700).

/etc/rc.d/rc.firewall

Remarque :

Il existe un outil graphique pour réaliser l'IPmasquerade, c'est drakgw, mais il est fortement conseillé de ne pas l'utiliser (attribution d'une adresse IP arbitraire, installation d'un serveur DNS, NIS, et DHCP ! !).

5.22 Outils de configuration

5.22.1 installation de webmin

Webmin (Administration par Web, port 10000) est un utilitaire puissant de configuration système et Réseaux par interface web.

Installation

RedHat	rpm -Uvh webmin-*.rpm
Mandrake	urpmi webmin
OpenBSD	Pour l'installer sous openbsd il va falloir utiliser les ports soit "cd /usr/ports/sysutils/webmin && make install"

utilisation

pour accéder à cette interface il suffit de tapper dans votre navigateur préféré :

```
http ://localhost :10000
```

Remarques

il est préférable de désactiver cet outils si l'on ne l'utilise pas. En effet même s'il reste possible d'en limiter l'accès et de le protéger par un firewall. la moindre faille correspondrait à un accès racine. De plus très récemment un grave problème de sécurité a été découvert.

CHAPITRE 6

Les outils des Hackers

6.1 Les différents types d'attaque

6.1.1 Cartographie des vulnérabilités

Cette opération consiste à déterminer le système d'exploitation, les services ouverts ainsi que leur version afin de pouvoir déterminer les éventuelles failles et les exploiter.
Un des outils les populaire et les plus puissant est "nmap". Il permet entre autres le half-scan (sans établir de connexion) *nmap -sS IP_du_serveur*
ou encore, le fingerprinting (détection d'OS)
nmap -sS -O IP_du_serveur

6.1.2 SUID/SGID

S-bit positionné pour l'utilisateur

Les processus sous Linux s'exécute avec l'identifiant de l'utilisateur (user-ID). Ceci leur donne accès aux ressources (fichier etc ...) auxquelles l'utilisateur a accès. L'identifiant effectif est celui qui détermine les accès aux fichiers.

```
find / -user root -a \( -perm -4000 -o -perm -2000 \) -print
```

6.1.3 Le crackage par mot de passe

La manière la plus classique par laquelle un hacker va essayer d'obtenir un mot de passe est l'attaque avec un dictionnaire. Dans ce genre d'attaque, le hacker utilise un dictionnaire de mots et de noms propres, et il les essaie un à un pour vérifier si le

mot de passe est valide. Ces attaques se font avec des programmes qui peuvent deviner des milliers de mots de passe à la seconde, même quand ceux-ci sont “hachés”. Ce procédé est d’autant plus facile qu’il lui permet de tester des variations sur les mots : mots écrits à l’envers, majuscules et minuscules, ajout de chiffres à la fin du mot...

Cet article en anglais décrit le fonctionnement d’un logiciel de crackage par mot de passe, et fournit les noms de quelques logiciels les plus utilisés par les Hackers.

http ://www.xphys.tuwien.ac.at/ mike/security/network-security.html

6.1.4 Le sniffing des mots de passe et des paquets

Si un hacker ne peut pas deviner un mot de passe, il a d’autres outils pour l’obtenir. Une façon qui est devenue assez populaire est le sniffing. La plupart des réseaux utilisent la technologie de broadcast (comme Ethernet). En pratique, tous les ordinateurs sauf le destinataire du message vont s’apercevoir que le message ne leur est pas destiné et vont donc l’ignorer. Mais par contre, beaucoup d’ordinateurs peuvent être programmés pour regarder chaque message qui traverse le réseau (mode promiscuité).

Il existe des programmes qui utilisent ce procédé et qui capturent tous les messages qui circulent sur le réseau en repérant les mots de passe. Si quelqu’un se connecte à un ordinateur à travers un réseau (telnet, rlogin, ftp...), alors cette personne risque contre son gré de donner son mot de passe. C’est pourquoi il existe une menace sérieuse pour les personnes qui se connectent sur des ordinateurs distants, où les mots de passe apparaissent en clair dans la trame.

Les programmes de sniffing les plus connus sont Esniff et TCPDump.

Mais un sniffer peut tout aussi bien être bénéfique à l’administrateur réseau, puisqu’il permettrait de déceler avant les Hackers les failles de sécurité de son réseau.

Ethereal v0.8.12 sous Linux permet de journaliser les événements définis par l’administrateur. Il est en outre compatible avec les journaux de LOG des routeurs Cisco (Cisco Secure IDS iplog files). Ethereal est téléchargeable à l’adresse suivante :
http ://www.ethereal.com/

Voici ci-dessous une liste d’autres sniffers disponibles dans le commerce :

1. ATM Sniffer Network Analyzer : http ://www.networkassociates.com décode plus de 250 protocoles.
2. Shomiti Systems Century LAN Analyzer : http ://www.shomiti.com supporte le standard Ethernet et fonctionne sous Windows 95/98 et NT.
3. PacketView de Klos Technologies : ftp.klos.com/demo/pvdemo.zip ce sniffer est basé sur DOS, idéal pour les environnements Ethernet.
4. Network Probe 8000 : http ://www.netcommcorp.com fait une analyse d’environ 13 protocoles dont TCP/IP, Microsoft, NFS, Novell.
5. LANWatch http ://www.guesswork.com marche sous DOS, Windows 9x et NT.
6. EtherPeek : http ://www.aggroup.com pour Windows et plates-formes Macintosh.

7. Ethload : sniffer qui permet de surveiller les sessions rlogin et telnet téléchargeable sur : http ://www.computercraft.com/noprogs/ethld104.zip
8. Linux sniffer : sniffer de mots de passe uniquement, en langage C

La meilleure défense contre l'attaque de sniffers est l'utilisation d'un protocole de chiffrement comme SSL (Secure Socket Layer).

6.1.5 L'IP spoofing

L'adresse IP d'un ordinateur est l'adresse qui est utilisée pour reconnaître un ordinateur sur internet. Un des principaux problèmes est qu'en utilisant le routage source d'IP, l'ordinateur du hacker peut se faire passer pour un ordinateur connu. Le routage source d'IP est une option qui peut être utilisée pour spécifier une route directe à une destination et renvoyer le chemin de retour à l'expéditeur. La route peut inclure l'utilisation d'autres routeurs ou de serveurs qui n'auraient normalement pas été utilisés pour faire suivre les paquets à la destination finale. Voici un exemple qui montre comment ceci peut être utilisé de façon à ce que l'ordinateur de l'intrus apparaisse comme étant l'ordinateur certifié par le serveur :

1. l'agresseur change l'adresse IP de son ordinateur pour faire croire qu'il est un client certifié par le serveur,
2. il va ensuite construire une route source jusqu'au serveur qui spécifiera le chemin de retour direct que les paquets IP devront prendre pour aller au serveur et qu'ils devront prendre pour retourner à l'ordinateur de l'agresseur en utilisant le client certifié comme dernière étape dans la route vers le serveur,
3. l'agresseur envoie une requête client au serveur en utilisant la route source,
4. le serveur accepte la requête du client comme si elle provenait directement du client certifié et retourne une réponse au client,
5. le client, utilisant la route source, faire suivre le paquet à l'ordinateur de l'agresseur.

Beaucoup de machines Unix acceptent les paquets de route source et les redirigent comme la route source l'indique. Beaucoup de routeurs acceptent également les paquets de route source bien que certains d'entre eux peuvent être configurés pour bloquer ces paquets.

Le routeur, pour des raisons de sécurité, ne devra pas accepter le routage source.

Une autre manière encore plus simple pour spoofer un client est d'attendre que le système client ait éteint sa machine et de se faire passer ensuite pour ce dernier. Les entreprises utilisent souvent des PC et le protocole TCP/IP et NFS pour se connecter à des serveurs Unix et obtenir un accès aux répertoires et aux fichiers du serveur. Comme NFS utilise uniquement les adresses IP pour authentifier les clients, un intrus pourrait configurer un PC avec le même nom et la même adresse IP qu'un autre ordinateur, et alors essayer de lancer des connexions au serveur Unix comme s'il était le vrai client. Ceci est très simple à réaliser et ressemblerait à une attaque de l'intérieur.

Le routeur devra donc refuser les connexions d'une machine ayant la même adresse IP qu'une machine interne, mais se trouvant à l'extérieur du réseau local. Les e-mails sont particulièrement sujets au spoofing car ils sont faciles à réaliser. Les courriers électroniques sans l'ajout d'une signature électronique ne peuvent pas être d'origine fiable. Il est facile par Telnet de se connecter directement au port SMTP du système (port 25). Le serveur recevant ces commandes fait confiance à cette personne si elle s'identifie. D'où le fait que le courrier électronique peut lui aussi être spoofé facilement en entrant une adresse d'expéditeur différente de l'adresse réelle. On peut donc sans aucun privilège falsifier ou spoofer le courrier électronique.

D'autres services comme le DNS peuvent aussi être spoofés mais avec toutefois plus de difficultés que le courrier électronique. Ces services représentent une crainte qui mérite d'être considérée quand on les utilise.

Le routeur firewall devra tenir régulièrement à jour ses fichiers LOG afin de contrôler toute tentative de piratage. De plus, ces fichiers LOG devront être sécurisés pour éviter toute modification malveillante.

6.1.6 Les scanners

Un scanner est un programme qui permet de savoir quels ports sont ouverts sur une machine donnée. Les Hackers utilisent les scanners pour savoir comment ils vont procéder pour attaquer une machine. Leur utilisation n'est heureusement pas seulement malsaine, car les scanners peuvent aussi permettre de prévenir une attaque.

Le plus connu des scanners réseau est WS_Ping ProPack, que l'on peut trouver sur http ://www.ipswitch.com/french/wsping.html

Les fichiers LOG générés par les scanners ne doivent pas être modifiables par un pirate.

6.1.7 Les chevaux de Troie

Un cheval de Troie est un programme qui se cache lui-même dans un autre programme apparemment au-dessus de tout soupçon. Quand la victime (l'utilisateur normal) lance ce programme, elle lance par là même le cheval de Troie caché. Actuellement, les chevaux de Troie les plus utilisés sont : Back Orifice 2000, Backdoor, Netbus, Subseven, Socket de Troie.

La méthode la plus efficace pour se protéger de ces programmes néfastes est d'utiliser un bon antivirus comme Norton 2000 ou Network Associates. Des programmes spécifiques permettent également de scruter toute tentative de connexion sur les ports scrutés. Lockdown 2000 est le plus connu d'entre eux : une fois une tentative de connexion détectée, il fait un traceroute sur l'IP qui a tenté la connexion. La version 4 possède en bibliothèque 488 signatures de "Troyans".

La machine Linux devra être équipée d'un antivirus permettant de repérer non seulement les virus, mais également les chevaux de Troie.

6.1.8 Les vers

Un ver est un programme capable de se propager et de s'auto-reproduire sans l'utilisation d'un programme quelconque ni d'une action par une personne. Sur chaque ordinateur où il agit, le ver crée une nouvelle liste de machines distantes cibles.

En parallèle, le ver :

1. essaie de trouver les mots de passe des comptes utilisateurs,
2. essaie d'entrer dans chaque machine cible en se faisant passer pour un utilisateur de la

machine "attaquante" (après avoir cracké le mot de passe utilisateur), et en utilisant un ancien bug dans le protocole finger, qui permet de savoir quels sont les usagers connectés sur une machine distante ou sur quelle machine est connecté un utilisateur donné.

Les attaques de vers sont toutefois très rares parce que les serveurs sur internet sont de plus en plus performants (Windows NT Server ou Apache), mais c'est toujours une méthode utilisée par les hackers quand un nouveau bug est découvert dans un système d'exploitation. Les vers permettent aux agresseurs d'attaquer un maximum de sites en peu de temps.

Le routeur firewall ne doit pas s'attarder à filtrer les vers : c'est la qualité du système d'exploitation qui doit permettre d'enrayer toute attaque de vers.

6.1.9 Les trappes

Une trappe est un point d'entrée dans un système informatique qui passe au-dessus des mesures de sécurité normales. C'est généralement un programme caché ou un composant électronique rendant le système de protection inefficace. De plus, la trappe est souvent activée par un événement ou une action normale (exemple : trappe dans les premières versions de Internet Explorer 5).

Pareillement au type d'attaque précédent, les trappes sont des programmes qui ne peuvent pas être détectés au niveau IP, mais au niveau application (signature). C'est donc le rôle de l'antivirus et du système d'exploitation de détruire les trappes.

6.1.10 Les bombes logiques

Ce sont des dispositifs programmés dont le déclenchement s'effectue à un moment déterminé en exploitant la date du système, le lancement d'une commande, ou n'importe quel appel au système.

Les bombes logiques doivent être repérées au niveau applicatif, par un antivirus performant.

6.1.11 Le TCP-SYN flooding

Quand un client essaie d'établir une connexion TCP sur un serveur, le client et le serveur échangent une séquence de messages. Cette connexion technique s'applique à toutes les connexions TCP/IP (Telnet, web, e-mails...).

Le système client commence par envoyer un message SYN (pour synchronisation) au serveur. Le serveur renvoie alors un accusé de réception du SYN : SYN-ACK au client. Le client finit alors par établir la connexion en répondant par un ACK. La connexion (au niveau 4 du modèle OSI) entre le client et le serveur est donc ouverte et le service d'échange de données peut s'exécuter. La faille vient du fait qu'au moment où le serveur a renvoyé un accusé de réception du SYN (SYN- ACK), le serveur mais n'a pas encore reçu le ACK du client. C'est alors une connexion dite semi- ouverte. Le serveur construit dans sa mémoire système une structure de données décrivant toutes les connexions courantes. Cette structure de données est de taille finie, ce qui veut dire qu'il peut se créer un dépassement de capacité (overflow) en créant intentionnellement trop de connexions partiellement ouvertes.

Le fait de créer ces semi-connexions sans se faire repérer est facilement réalisable avec l'IP spoofing. L'ordinateur de l'agresseur envoie des messages SYN au serveur victime ; ceux-ci paraissent provenir d'un ordinateur bien défini mais font référence à un système client qui n'est pas capable de répondre au message SYN-ACK. Ce qui veut dire que le message ACK final ne sera jamais envoyé au serveur victime.

Ces semi-connexions dans la structure de données du serveur victime vont éventuellement créer un débordement dans cette structure et le serveur sera incapable d'accepter d'autres connexions tant que la table ne sera pas vidée. Normalement, il y a un système de time-out associé à chaque connexion ouverte, donc les semi-connexions devraient expirer et le serveur victime récupérer de la place libre dans sa mémoire pour d'autres connexions. Toutefois, le système agresseur peut simplement continuer à envoyer des paquets dont l'IP est spoofée plus vite que le serveur victime puisse expirer les semi-connexions.

Dans la plupart des cas, la victime d'une telle attaque aura des difficultés à accepter toute nouvelle connexion. Dans ces cas, l'attaque n'affectera pas les connexions déjà existantes ou la capacité à créer des connexions de l'intérieur vers l'extérieur. Par contre, dans certains cas, le serveur aura épuisé toutes ses ressources mémoires, et pourra "planter" et donc être rendu inopérant.

La localisation de l'attaque est très souvent obscure parce que les adresses IP des paquets SYN envoyés sont rarement plausibles. Quand le paquet arrive au serveur victime, il n'y a aucun moyen de déterminer sa véritable source. Comme internet fait suivre les paquets sur une adresse de destination, le seul moyen de valider la source d'un paquet est d'utiliser le filtrage.

Avec la technologie actuelle du protocole IP, il est impossible d'éliminer tous les paquets spoofés. Mais il existe quelques solutions pour réduire le nombre de paquets spoofés et sortant du réseau.

Le routeur devra limiter les entrées à l'interface externe, en n'accordant pas le droit

d'entrée à un paquet qui a une adresse source du réseau interne.

Il peut aussi être bon de filtrer les paquets sortants qui ont une adresse source différente du réseau interne afin de prévenir une attaque d'IP spoofing provenant du réseau interne.

La combinaison de ces deux types de filtrage doit empêcher les agresseurs extérieurs d'envoyer des paquets prétendant provenir du réseau interne. Cela doit également empêcher les paquets provenant du réseau interne de prétendre venir de l'extérieur du réseau.

L'IP spoofing et le SYN-flooding sont actuellement les deux principaux problèmes de la sécurité sur Internet.

6.1.12 Le Nuke

Les Nukes sont des plantages du système d'exploitation dûs à des pirates qui connaissent votre adresse IP, et qui utilisent un bug du système. Les Nukes font généralement "planter" le système et il ne reste plus qu'à rebouter.

Avec Linux, et sa particularité d'être en "Open Source", les failles du système se résolvent d'elles- mêmes, puisque ce sont les utilisateurs qui détectent et corrigent les failles. Le temps de latence entre la détection de la faille et sa réparation est donc plus court qu'avec les autres systèmes d'exploitation, ce qui réduit potentiellement les possibilités d'attaques.

Il faudra donc réaliser une veille technologique du système Linux, et télécharger les patchs de correction et de mise à jour.

6.1.13 Le Flood

Le Flood consiste à envoyer très rapidement de gros paquets d'informations à la machine routeur (à condition d'avoir un PING très court, comme c'est le cas pour un RLE), ce qui risque de faire "planter" la machine ("Ping Of Death" : message ICMP de 64 ko qui faisait planter Windows).

Une solution consiste à avoir une bonne gestion de la fenêtre d'anticipation TCP, et d'envoyer autant que faire se peut des messages ICMP de réduction de fenêtre.

6.1.14 Le Spamming

Le Spamming consiste à envoyer plusieurs milliers de messages identiques à une boîte aux lettres pour la faire saturer. En effet, les mails ne sont pas directs, ainsi lorsque le courrier sera relevé, celui-ci mettra beaucoup trop de temps et la boîte aux lettres sera alors inutilisable.

Le routeur firewall pourra cependant détecter les tentatives de Spamming, en sniffant les paquets IP arrivants, et en remarquant une trop grande fréquence dans la réception d'un même message (taille du paquet, adresses source et destination identiques).

6.1.15 Les virus

Les constructeurs de firewalls tendent maintenant à fournir avec leurs produits une solution antivirus complète, qui permet de filtrer les attaques logicielles comme les chevaux de Troie, les vers, les trappes et les bombes logiques.

Les éléments actifs du réseau sont désormais de véritables remparts contre une pléthore d'attaques, qu'elles soient au niveau réseau ou au niveau applicatif. Cela rend la tâche des administrateurs réseau plus simple, car toutes les fonctions de sécurité sont fédérées sur un seul et même équipement, plus robuste et entièrement administrable.

Les éléments composant le réseau ne sont pas les seuls remparts aux attaques. Le système d'exploitation garantit un niveau de sécurité supplémentaire vis-à-vis des attaques de type virus, nuke ou trappes. C'est pourquoi nous nous intéressons maintenant à l'aspect sécurité de Linux.

6.1.16 Attaque du Display

Comment protéger sa session X ?

Dans le fichier .xsession (si vous en avez un dans la racine), mettre les lignes suivantes :
Inhibe toute connexion
xhost -
Utiliser xauth pour modifier le fichier d'autorisation .Xauthority.
Exemple :
% xauth list
xauth : creating new authority file /home/user/.Xauthority
% xauth
Using authority file /home/user/.Xauthority
xauth> ?
Commands :
add exit extract help info list
merge nextract nlist nmerge quit remove
source ?
xauth> list
the-pilgrim :0 MIT-MAGIC-COOKIE-1 383242.....
xauth> remove the-pilgrim :0
1 entries removed
xauth> list
xauth> quit
Le fichier .Xauthority doit avoir les droits rw- — —
Une fois connectés, il ne faut pas laisser votre terminal sans surveillance. Quelqu'un peut changer les droits de vos comptes/fichiers, charger des images, etc.

CHAPITRE 7

Les solutions de défense

7.1 Les détecteurs d'intrusion IDS

7.1.1 snort

Snort est un système de détection d'intrusion réseau en sources ouvertes, capable d'effectuer l'analyse du trafic en temps réel et de la journalisation de paquets sur des réseaux IP. Il peut effectuer de l'analyse de protocoles, de la recherche / correspondance de contenu et peut être utilisé pour détecter une variété d'attaques et de scans, tels que des débordements de tampons, des scans de ports furtifs, des attaques CGI, des scans SMB, des tentatives d'identification d'OS, et bien plus. Snort utilise un lan gage de règles flexible pour décrire le trafic qu'il devrait collecter ou laisser passer, ainsi qu'un moteur de détection qui utilise une architecture modulaire de plu gin. Snort possède aussi des capacités modulaires d'alertes temps réel, incorporant des plugins d'alerte et de journalisation pour syslog, de fichiers textes en ASCII, de sockets UNIX, de messages WinPopup à des clients Windows en utilisant smbclient de Samba, de base de données (Mysql/PostgreSQL/Oracle/ODBC) ou XML.

Snort a trois utilisations principales. Il peut être utilisé comme un simple renifleur de paquets comme tcp dump(1), un enregistreur de paquets (utile pour déboguer le trafic réseau, etc), ou comme un complet système de détection d'intrusion réseau.

Snort journalise les paquets dans le format binaire tcp dump(1) , vers une base de données ou dans le format ASCII décodé de Snort vers une "hiérarchie" de répertoires de journalisation qui sont nommés d'après l'adresse IP du système "étranger".

Système	commande d'installation	démarrage du programme
RedHat	rpm -Uvh snort-*.rpm	service snortd start
Mandrake	urpmi snort	/etc/init.d/snortd start
Debian	apt-get install snort	/etc/init.d/snortd start
OpenBSD	pkg_add snort.*.tgz	snort -A full -D

Sous OpenBSD Vous pourrez trouver la liste des règles a utiliser dans le répertoire /usr/local/share/examples/snort

ensuite il faudra :

1. copier le fichier /usr/local/share/examples/snort/snort.conf dans /etc
2. modifier le fichier /etc/snort.conf et remplacer la ligne :
   var RULE_PATH ./ par
   var RULE_PATH /usr/local/share/examples/snort
3. ensuite : mkdir /var/log/snort

4. enfin pour lancer snort il suffira de tapper la commande :
   /usr/local/bin/snort -A full -D -c /etc/snort.conf et directement l'inclure dans le fichier */etc/rc.local* pour qu'il soit lancé a chaque démarrage.

7.1.2 aide

Aide est un très puissant détecteur d'intrusion système en effet il va tout d'abord cartographier (faire une prise d'empreinte) tous les fichiers sensibles du système et ensuite vérifier la façon dont il ont été modifiés, ainsi si celle-ci ne semble pas correcte alors "aide" va signaler toute les modifications effectuées.

Système	commande d'installation	démarrage du programme
RedHat	rpm -Uvh aide-*.rpm	service aide start
Mandrake	urpmi aide	/etc/init.d/aide start
Debian	apt-get install aide	/etc/init.d/aide start
OpenBSD	pkg_add aide.*.tgz	aide –check

Pour se faire :

1. il va être nécessaire de créer une base de donnée sur le système existant avec la commande : *aide –init*.
2. ensuite la base de donnée utilisée étant /var/db/aide.db il sera nécessaire de copier /var/db/aide.db.new générée par *aide –init* :
   cp /var/db/aide.db.new /var/db/aide.db
3. enfin *aide –check* pour lancer la vérification qui donnera la liste des fichiers modifiés.

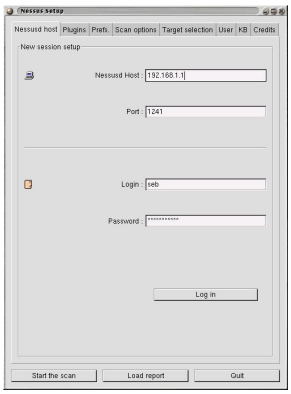
7.2 Les outils de diagnostique

7.2.1 Nessus

Nessus est l'outil indispensable pour un adminstrateur réseau, en effet celui-ci vous permettra d'effectuer un audit complet de votre système en recensant par exemple les problème liés à certaines versions de vos différents serveurs.

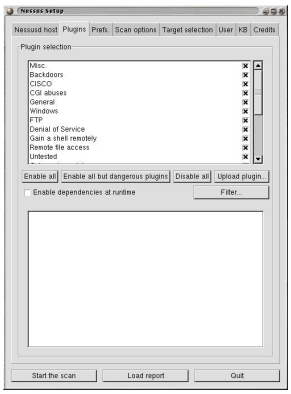
Ainsi à la fin du scan un rapport clair sera généré. Pour se faire il faudra procédé comme suit :

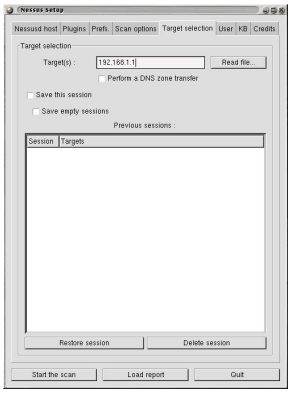
1. nessus-adduser
2. nessus-mkcert
3. lancer le serveur : nessusd sous BSD ou /etc/init.d/nessusd start sous linux
4. ensuite il faudra vous connecter en tant que client avec la commande nessus
5. sélectionner les plugins souhaités pour le scan et enfin choisir la machine a scanner en cliquant sur l'onglet "target selection".



Connexion avec le login créé
par la commande "nessus-adduser"

Selection des plugins a
utiliser durant le scan





Selection de la cible a scanner

7.2.2 Dsniff

dsniff est une collection de programmes permettant d'auditer un réseau et d'effectuer des tests d'intrusion ainsi dans cette collection on pourra noter la présence de :

1. dsniff,
2. filesnarf,
3. mailsnarf,
4. urlsnarf,
5. et webspy

qui permettrons d'écouter passivement le réseau afin d'en extraire les données les plus intéressantes on verra ainsi apparaître tous les logins/passe, les contenus des mails qui transitent sur le réseau ! !. dans le cas ou tout serait relié par un hub par exemple.
Il existe une deuxième collection de programmes : arpspoof, dnsspoof, et macoff qui facilitent l'interception du trafic réseau normalement non accessible à un attaquant.
Enfin sshmitm et webmitm qui permeteront de rediriger les sessions SSH et HTTPS en se basant sur certaines anciennes failles.

7.3 les outils de cryptage et d'authentification

7.3.1 IPsec

IPsec vise à prévenir les diverses attaques rendues possibles par le protocole IP, notamment empêcher un adversaire d'espionner les données circulant sur le réseau ou de se faire passer pour autrui afin d'accéder à des ressources ou données protégées.

Dans ce but, IPsec peut fournir, suivant les options sélectionnées, tout ou partie des services de sécurité suivants :

Confidentialité des données et protection partielle contre l'analyse du trafic.

Les données transportées ne peuvent être lues par un adversaire espionnant les communications. En particulier, aucun mot de passe, aucune information confidentielle ne circule en clair sur le réseau. Il est même possible, dans certains cas, de chiffrer les en-têtes des paquets IP et ainsi masquer, par exemple, les adresses source et destination réelles. On parle alors de protection contre l'analyse du trafic.

Authenticité des données et contrôle l'accès continu.

L'authenticité est composée de deux services, généralement fournis conjointement par un même mécanisme : l'authentification de l'origine des données et l'intégrité. L'authentification de l'origine des données garantit que les données reçues proviennent de l'expéditeur déclaré. L'intégrité garantit qu'elles n'ont pas été modifiées durant leur transfert.

La garantie de l'authenticité de chaque paquet reçu permet de mettre en oeuvre un contrôle l'accès fort tout au long d'une communication, contrairement à un contrôle d'accès simple à l'ouverture de la connexion, qui n'empêche pas un adversaire de récupérer une communication à son compte. Ce service permet en particulier de protéger l'accès à des ressources ou données privées.

Protection contre le rejeu

La protection contre le rejeu permet de détecter une tentative d'attaque consistant à envoyer de nouveau un paquet valide intercepté précédemment sur le réseau.

Ces services sont basés sur des mécanismes cryptographiques modernes qui leur confèrent un niveau de sécurité élevé lorsqu'ils sont utilisés avec des algorithmes forts.

7.3.2 Utilisation de Kerberos

Système de sécurité et d'authentification des utilisateurs du réseau local du MIT. Il ne semble pas avoir été piraté depuis sa mise en service, il y a une dizaine d'années.

Le protocole Kerberos permet un chiffrement de l'ensemble des transactions entre applications adaptés pour l'utiliser. On trouve des versions dites kerberisées des outils telnet, ftp, fetchmail...

Ce protocole fonctionne sur un contrôle de l'identité des clients, qui peuvent obtenir par la suite des tickets pour s'authentifier auprès des différents services présents

sur le réseau.

Toutes les transactions passant par Kerberos sont chiffrées, ce qui résout un ensemble de problèmes de sécurité. En revanche, le système impose sur les utilisateurs une certaine lourdeur, et, en dépit de sa fiabilité, ne peut être recommandé partout.

Ce module détaille le fonctionnement du protocole Kerberos, et son implémentassion sous Linux. Il décrit les modalités d'utilisation de Kerberos et de sa configuration, ainsi que les inconvénients qu'il occasionne.

Ce module est susceptible d'intéresser les administrateurs de réseaux nécessitant une haute sécurité, et ceux désireux de découvrir la conception d'un protocole sécurisé et les parades possibles aux problèmes de sécurité.

7.3.3 Signature sécurisé de mail avec GnuPG

Introduction a GnuPG

Pourquoi utiliser GPG ?

Si l'intérêt de crypter ces fichiers et ces mails ne présente que peu d'attrait pour le grand public ne désirant pas a jouer a James Bond, la cryptographie offre d'autres avantages plus tangibles comme par exemple :

* Possibilité de signer un mail : en effet GPG permet simplement a quelqu'un disposant de votre clef publique, (nous reviendrons plus tard sur les concepts de clés privée et publique), de vérifier que vous êtes vraiment la personne l'ayant envoyé. La cryptographie dans ce cas-ci remplit la fonction de signature numérique * Possibilité de certifier un mail : il est aussi possible par le même mécanisme que le précèdent de certifier que le contenu du mail n'a pas été modifie depuis son envoi. Il s'agit donc ici d'un scelle virtuel.

La signature se présentera sous la forme d'un attachement a votre mail et sera donc interprète par votre client mail s'il le supporte et ignore dans le cas contraire. Votre mail restera lisible comme par avant.

GPG permet aussi bien sur de crypter des mails et des fichiers.

Comment ça marche ?

GPG s'appuie sur des algorithmes a base de mécanismes de clef prive et publique. Sans rentrer dans trop de détails qui ne sont pas dans l'optique de ce tip, signalons juste que des messages crypte avec votre clef prive sera décrypter avec votre clef publique (Ceci est une grosse conceptualisation pour une explication plus en profondeur et correcte se rendre ici). Vous ne donnerez donc que votre clef publique a des tierces personnes, votre clef prive ne sera jamais divulgue d'où un énorme gain de sécurité par rapport a un système avec une seule clef.

Ceci est pour la partie technique, mais le système repose sur un concept tout aussi important, les réseaux de ?Trusted keys ?. En effet, même si vous pouvez vérifier la signature d'un mail rien ne vous garantie que la personne qui vous la envoyer existe

vraiment et que son identité correspond. Pour cela, par exemple pendant les réunions de LUG ou autre, les personnes s'échangent leurs fingerprint tout en vérifiant leurs papiers d'identités. Puis ils ?signent ? les clefs qu'ils ont vérifier. Quand on récupère une clef d'un serveur, on peut voire la liste des personnes ayant garantie l'identité de la personne. donc si par exemple, la clef a été signe par quelqu'un en qui vous avez confiance, vous pouvez considérer la clef comme valide. Il se crée donc un réseau de confiance.

Installation de GPG

Configuration initiale de GPG :

Tout d'abord il vous faudra générer votre clef :
exécuter GPG une première fois pour que les fichiers de configuration soit crée,

```
$> gpg
gpg : /home/seb/.gnupg : directory created
gpg : /home/seb/.gnupg/options : new options file created
gpg : you have to start GnuPG again, so it can read the new options file
```

Puis une seconde fois pour générer les trousseaux de clefs prives et publiques.

```
$> gpg
gpg : /home/seb/.gnupg/secring.gpg : keyring created
gpg : /home/seb/.gnupg/pubring.gpg : keyring created
```

Enfin exécutez :

```
$> gpg –gen-key
Please select what kind of key you want :
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) ElGamal (sign and encrypt)
Your selection ?

Tapez juste [entrer] a ce prompt

DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
minimum keysize is 768 bits
default keysize is 1024 bits
highest suggested keysize is 2048 bits
What keysize do you want ? (1024)

Pareil, restez avec le choix par défaut.
```

```
Requested keysize is 1024 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for ? (0)

Idem, votre clef sera définitive par défaut.

Key does not expire at all
Is this correct (y/n) ?

Taper [entrer] après avoir vérifier si les options sont correctes.

You need a User-ID to identify your key ; the software constructs the
user id
from Real Name, Comment and Email Address in this form :
Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de> ?
Real name :
Il suffit ici de taper son prénom et nom.
mail address : |
Assez explicite
Comment :

C'est la place idéale pour y mettre votre nickname
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit |

Appuyer sur (O) si vous êtes satisfait.

You need a Passphrase to protect your secret key.

Enter passphrase :

Et voila enfin le moment tant attendu ou vous entrerez votre mot de
passe, vu que celui ci vous servira a protéger votre identité, veuillez
choisir un mot de passe assez facile a retenir pour que vous n'ayez pas
a la noter quelque part et assez complexe pour éviter que quelqu'un ne
la devine, une mot de passe compose de caractères alphanumériques en
mixed case est parfait dans ces cas la.

| We need to generate a lot of random bytes. It is a good idea to perform
```

```
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation ; this gives the random number
generator a better chance to gain enough entropy.
.++++++++++++++..++++++++++.++++++++++++
++.++++++++++++++++++++.++++++++++.+++++++
++++++++......<+++++>..+++++.................
.................+++++
public and secret key created and signed.
```

Et voila vos clefs ont été crées.

Maintenant éditez /.gnupg/options et décommentez la ligne :

#keyserver wwwkeys.nl.pgp.net

Vous pouvez évidement mettre un autre serveur si vous disposez de son adresse.

Il s'agit maintenant d'envoyer votre clef sur le serveur pour que les personnes voulant récupérer votre clef publique n'aura besoin que de votre ID.

Voyons tout d'abord plus en détail ces différents termes :

tapez :

```
$> gpg –list-keys

Normalement il devrait y avoir une seule clef, la votre.

/home/seb/.gnupg/pubring.gpg
————————————————
pub 1024D/19E27067 2001-10-15 Seb (#####) <##########>
sub 1024g/B0418883 2001-10-15


1024D = la longueur de la clef
19E27067 = votre User ID.
```

et maintenant vous pouvez faire un

```
$> gpg –fingerprint 19E27067*
pub 1024D/19E27067 2001-10-15 Seb (#####) <#########<>
Key fingerprint = 767C 7E76 7142 FF87 .... 8788 7F47 .... 19E2 7067
sub 1024g/B0418883 2001-10-15
```

Vous avez sûrement remarqué que les 8 derniers caractères de votre fingerprint contient votre ID.

Donc, si vous voulez donner votre clef a quelqu'un il faudra lui donner votre fin-

gerprint entier mais il n'aurait qu'a entrer votre ID et vérifier le reste du fingerprint.

pour cela il faut évidement que votre clef soit sur un serveur, les serveurs sont relies entre eux a la manière des serveurs dns, il suffit donc de l'envoyer sur un des serveur.

```
$> gpg –send-key 19E27067*
```

Ajout de clef :

Vous avez récupérer le fingerprint d'un ami, après avoir vérifier ses papiers d'identités (cette vérification est obligatoire pour un fonctionnent du réseaux des ?trusted key ?).

Une fois devant votre ordinateur faites :

```
$> gpg –recv-key son_id
```

(rappel : son ID est les 8 derniers caractères de son fingerprint pour une clef 1024D) Puis :

```
$> gpg –fingerprint son_id
```

et vérifier si le fingerprint correspond bien a celui qu'il vous a donne.
Maintenant que vous pouvez confirmer que son identité correspond bien a sa clef vous pouvez la signer

```
$> gpg –sign-key son_id
```

Et finalement, il faut faire savoir au serveur que vous avez signer la clef

```
$> gpg –send-key son_id
```

Ajout d'un UID :

Il arrive souvent qu'une même personne ait plusieurs comptes mails par exemple, un pour le travail, un personnel etc ...

Dans ce cas gpg permet d'associer plusieurs UID a la même personne. Pour cela il suffit de faire :

```
$> gpg –edit-key votre_id
```

la, tapez : adduid

A ce moment la, les mêmes questions que lors de la création de la clef seront pose a la différence qu'une nouvelle clef ne sera pas génère.
il suffit de taper : save
pour quitter et sauvegarder les changements.

Vous pouvez voir les changement avec :

```
$> gpg –list-key
```

Et il faudra évidement mettre a jour la clef dans le serveur :

```
$> gpg –send-key votre_id
```

A noter que les nouveaux ids devront êtres signe aussi.

Intégration dans les clients mails :

* Dans Sylpheed :

Il suffit de d'activer le support gpg dans l'onglet "privacy" des options.

Révoquer une clef :

Si votre clef est corrompue, plus utilise, ou compromise, il vous la révoquer, mais pour cela il faut créer un certificat de révocation.

```
$> gpg –gen-revoke user_id

Create a révocation certificate for this key ?
Please select the reason for the revocation :
1 = Key has been compromised
2 = Key is superseded
3 = Key is no longer used
0 = Cancel
(Probably you want to select 1 here)
Your decision ?
```

CHAPITRE 8

La recette magique d'un système sécurisé...

Malheureusement et contrairement au titre de ce chapitre la sécurité d'un système ne viendra jamais d'un logiciel miracle mais bel et bien d'une administration aussi paranoïaque que possible. Néanmoins il est possible de donner une liste de règles simples a suivre :

8.1 fonctionnalitées de la machine.

Déterminez de façon précise ce que seront les fonctionnalitées nécessaires et suffisantes de chaque machines :

- comment sera-t-elle connectée au réseau.
- quels utilisateurs y auront accès.
- quels programmes nécessaires et suffisants

8.2 Le disque Dur

Éviter que le puisse remplir le disque dur

En effet il est a considérer qu'il est possible de freezer un système simplement en remplissant les répertoires /tmp, /var/log ou encore les répertoires des logs des différents IDS (détection d'intrusion). C'est pourquoi il sera indispensable de réserver des partitions pour ces différents répertoires et mettre en place des quotas pour chaques utilisateurs.

Restreindre les options de montage des différentes partitions

Dans les options de montage des différentes partitions il est possible d'y interdire toute exécution de programmes, ou l'utilisation de fichier a permissions SUID/GUID

avec respectivement les options noexec et nosuid. Ce qui pourrait être le cas par exemple des répertoires /home/ftp/pub, /mnt/cdrom, etc.

8.3 les Programmes

limiter le nombre de programmes du système

En effet il faut considérer que chaque programme va représenter une faille de sécurité potentiel supplémentaire. Il sera donc indispensable pour ce faire d'installer un système minimal par défaut puis d'ajouter un a un les programmes nécessaires en identifiant ceux donnant accès a des fichiers SUID

la mise a jour

Il faut considérer que chaque programme possède une faille de sécurité potentiel et que donc elle finira par être découverte c'est pourquoi il faudra se tenir informé des mises a jour de sécurité.

8.4 les droits utilisateurs

L'utilisateur root ayant un accès illimité a tous les fichiers du système il va donc falloir restreindre voir supprimer les logins de root. Par exemple le supprimer son accès via ssh en ajoutant la ligne "PermitRootLogin no" au fichier /etc/ssh/sshd_config. Ensuite il est important de vérifier l'umask par defaut donné par le fichier /etc/profile.

8.5 les services

Chaque service étant une porte ouverte sur votre machine

CHAPITRE 9

Cryptographie

9.1 Sécurité relative de la Cryptographie

Attaquant	Budget	Outil	Casser clef de 40 bits		Casser clef de 56 bits	
			Temps	Coût	Temps	Coût
Hacker	40$	Temps machine	1 semaine	0,08 $	38 ans	5 000 $
PME	10K$	circuit prédiffusé	12 min.	0,08 $	18 mois	5 000 $
Grande Ent.	300K$	circuit prédiffusé	24 sec.	0,08 $	19 jours	5 000 $
	300K$	ASIC	0,18 sec.	0,001 $	3 heures	38 $
Multinationale	10 M$	ASIC	0.005 sec.	0,001 $	6 min.	38 $
Etat (CIA,DST)	300 M$	ASIC	0.0002 sec.	0,001 $	12 sec.	38 $

TAB. 9.1: Source : Matthew Blaze, rapport présenté au sénat américain juin 96

9.2 Performance des différents algorithmes

Rapidité de algorithmes symétrique.

Algorithme	Kbyte/s	Mbit/s
des-cbc	11920	95
3des-cbc	4867	39
cast128-cbc	9622	77
blowfish-cbc	15530	124
twofish-cbc	17481	140
rc5-12-cbc	11970	96
rc5-16-cbc	9411	75

rc5-20-cbc	7586	61
rc6-cbc	8937	71
mars-cbc	5545	44
saferk64-cbc	6304	50
saferk128-cbc	4179	34
safersk64-cbc	5037	40
safersk128-cbc	4179	33
skipjack-cbc	1957	16
idea-cbc	4798	38

Rapidité de algorithmes des tables de hash.

HMAC	Kbyte/s	Mbit/s
hmac-md5	72178	577
hmac-sha1	18366	147
hmac-ripemd160	23080	185

Rapidité de algorithmes asymétrique.

RSA	1024 bit keygen	0.35 seconds/key
RSA	1024 bit encryption	1786 times/second
RSA	1024 bit decryption	75 times/second
RSA	1024 bit signing	74 times/second
RSA	1024 bit verification	1991 times/second
RSA	2048 bit keygen	2.83 seconds/key
RSA	2048 bit encryption	672 times/second
RSA	2048 bit decryption	12 times/second
RSA	2048 bit signing	12 times/second
RSA	2048 bit verification	715 times/second
DSA	keygen	1.23 seconds/key
DSA	signing	137 times/second
DSA	verification	70 times/second
EC-MODP	keygen	0.20 seconds/key
EC-MODP	ElGamal encrypt	45 times/second
EC-MODP	ElGamal decrypt	90 times/second
EC-MODP	DSA sign	87 times/second
EC-MODP	DSA verify	44 times/second
EC-GF2N	keygen	0.10 seconds/key
EC-GF2N	ElGamal encrypt	31 times/second
EC-GF2N	ElGamal decrypt	61 times/second
EC-GF2N	DSA sign	60 times/second
EC-GF2N	DSA verify	30 times/second

Tous ces tests ont été réalisé avec un PIII 600Mhz

9.3 Différents Types de Cryptographie

Algorithmes Cryptographiques (restreint)
les données sont cryptées/décryptées par un algorithme.

Algorithmes à clef secrète
les données sont cryptées/décryptées par une clef unique.

Algorithmes à clef Publique
les données sont cryptées/décryptées par deux clefs différentes .

9.3.1 L'algorithme RSA

9.4 Protocoles sécurisés. OpenSSL.

9.4.1 Fonctionnement d'un protocol sécurisé.

SSL utilise les fonctionnalités offertes par TCP/IP pour permettre aux couches supérieures d'accéder à un mode d'accès sécurisé. Dans les plus courants utilisant ce mode, on va trouver bien sûr HTTP mais aussi LDAP, SMTP, NNTP ou encore IMAP.

Les trois fonctionnalités de SSL sont :

Authentification du serveur cela permet à un utilisateur d'avoir une confirmation de l'identité d'un serveur. En effet un programme client SSL utilise des méthodes de chiffrement à clé publique pour vérifier si le certificat et l'identité publique fournis par le serveur sont valides et ont été fournis par un fournisseur de certificat présent dans la liste de ceux connus du client. Cette fonctionnalité est importante dans la mesure où le client doit envoyer des données confidentielles comme son numéro de carte bleue.

Authentification du client la technique est ici exactement la même que pour l'authentification du serveur. Cela peut servir si le serveur envoie des informations importantes à un client, qui doit, dans ce cas être authentifier.

Chiffrement des données toutes les données issues de l'entité émettrice sont chiffrées et déchiffrées par l'entité réceptrice, ce qui permet de garantir la confidentialité des données. Un mécanisme permet également de garantir l'intégrité des données.

Le protocole SSL peut être divisé en 2 sous protocoles : l'encodage (record) et la négociation (handshake). Le premier définit tout ce qui touche à l'envoi des données. Le second est utilisé pendant toute la phase de négociation entre le client et le serveur jusqu'à ce que tous les paramètres soient validés par l'un et l'autre.

9.4.2 L'encodage dans SSL

SSL est capable d'utiliser différents mécanismes de chiffrement créés pour l'authentification, l'envoi de certificats ou l'établissement des clés. Le choix des mécanismes de sécurité mis en oeuvre dépend de plusieurs paramètres : la politique de sécurité de l'entreprise possédant le serveur, la version du protocole SSL ou encore les lois gouvernementales. Le but du sous-protocole de négociation est en outre d'obtenir un accord entre le client et le serveur pour le chiffrement utilisé.

Lors de la phase de négociation, le client et le serveur vont se mettre d'accord sur le meilleur algorithme de chiffrement utilisable entre les parties. La table ci-dessous rend compte des différents algorithmes pouvant être utilisés :

MEILLEURES GARANTIES

3DES qui supporte un chiffrement à 168 bits couplé avec SHA-1 pour l'intégrité. Ce mécanisme n'est autorisé qu'à l'intérieur des USA et est approprié aux banques car 3DES est nettement moins rapide que RC4. Supporté par SSL 2.0 et 3.0

BONNES GARANTIES

Ce chiffrement est suffisamment fort pour garantir la plupart des transactions électroniques

RC4 avec un chiffrement 128 bits couplé à MD5 pour l'intégrité. RC4 est le plus rapide des modes de chiffrement offert. Supporté par SSL 2.0 et 3.0.

RC2 avec un chiffrement 128 bits couplé à MD5 pour l'intégrité. RC2 est plus lent que RC4 et n'est plus supporté que par SSL 2.0.

DES qui permet un chiffrement sur 56 bits couplé avec SHA-1. Ce chiffrement reste moins performant que RC4 ou RC2. Il est supporté par SSl 2.0 et 3.0 à la différence que SSL 2.0 utilise MD5 pour authentifier les messages.

CHIFFREMENT DÉDIE A L'EXPORTATION

C'est le chiffrement qui procure la plus haute sécurité pour une exportation internationale

RC4 avec un chiffrement 40 bits et MD5. Supporté par SSL 2.0 et 3.0

RC2 avec un chiffrement 40 bits et MD5. Supporté par SSL 2.0 et 3.0

FAIBLES GARANTIES

Ce mécanisme garantit l'intégrité des données, mais les données qui circulent ne sont pas chiffrées.

MD5 Authentification des messages avec MD5 sans chiffrement. Cette méthode permet seulement de garantir l'intégrité des données échangées. Elle est typiquement utilisée dans le cas où le serveur et le client n'ont aucun chiffrement en commun.

9.4.3 La négociation dans SSL

Le SSL combine simultanément l'utilisation de clés publiques et de clés symétriques. Les clés publiques privées ou clés asymétriques procurent en effet une très bonne méthode pour l'authentification mais son utilisation est coûteuse en terme de bande passante. A l'opposé, le mécanisme de clé symétrique (identique pour chiffrer et déchiffrer) est extrêmement rapide mais pas réellement adapté à l'authentification d'un tiers.

Ainsi SSL va utiliser son protocole de négociation qui va être apte à partir des clés publiques et privées du client et du serveur d'établir une communication entre les deux entités avec une clé secrète (symétrique) de taille nettement inférieure à celle rencontrée pour des clés publiques (128 bits contre 1024 ou plus).

Mécanisme

1 Le client envoie au serveur sa version du protocole SSL, ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le serveur a besoin.

2 Le serveur renvoie sa version de SSL, ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le client a besoin. Le serveur envoie également son propre certificat, et si le client demande une information nécessitant un certificat, il demande également un certificat client.

3 Le client utilise les informations envoyées par le serveur pour l'authentifier. Si le serveur ne peut pas être authentifié, la connexion n'a pas lieu.

4 Avec les données préalablement échangées, le client est en mesure d'envoyer au serveur une pré clé secrète, qu'il chiffre avec la clé publique du serveur. Si le serveur a requis une authentification du client, celui ci (le client) renvoie également au serveur un bloc de données signé ainsi que son certificat.

5 Si le serveur a requis une authentification, il authentifie le client. Le serveur utilise alors sa clé privée de façon à pouvoir déchiffrer la pré clé secrète. Le serveur effectue alors une suite d'actions (également effectuées par le client) pour obtenir une clé secrète à partir de la pré clé secrète.

6 Le client et le serveur utilisent la clé secrète pour générer des clés de session qui seront les clés symétriques utilisées pour le chiffrement, déchiffrement des données et l'intégrité.

7 Le client envoie alors un avertissement au serveur le prévenant que les prochains messages seront chiffrés avec la clé de session. Puis il envoie un message chiffré indiquant que la phase de négociation est terminée.

8 Le serveur envoie alors un avertissement au client comme quoi les prochains messages seront chiffrés avec la clé de session. Puis il envoie un message (chiffré cette fois) indiquant que la phase de négociation est terminée.

9 La phase de négociation est alors terminée.

Authentification

Dans le cas du SSL il est pour le moment assez rare de rencontrer une authentification du client. En effet, la plupart des applications utilisées sur Internet ne requièrent pas un tel niveau de sécurité. De plus, l'authentification du client ressemble très fortement à une authentification du serveur. L'authentification serveur a lieu comme suit :

[pic]

1 Vérification de la date de validité du certificat du serveur.

2 Est-ce que l'autorité de certification est une autorité de confiance ? Pour vérifier cela, chaque client maintien une liste de noms de domaines. Si le nom spécifié (nom émetteur) correspond à un nom rentré dans la liste, le client passe à l'étape suivante.

3 Vérification de la clé publique à partir de la signature. Le client vérifie la validité de la signature (chiffrée avec la clé privée) fournie dans le certificat grâce à la clé publique qui a été fournie elle aussi dans le certificat. A partir de ce point, le certificat du serveur est considéré comme valable.

4 Vérification du nom de domaine du serveur. Cette étape permet de vérifier que le nom de domaine du serveur défini dans le certificat correspond bien à la même adresse Internet. Cette étape n'est pas obligatoire et dépend de l'implémentassion du client SSL. Elle permet cependant d'éviter qu'un usurpateur vienne jouer les intermédiaires entre le client et le serveur et se fasse passer pour l'un et l'autre auprès des deux entités. Vérifier la validité du nom de domaine est le seul moyen d'éviter ce genre de faille qui permet, dans ce cas, à la personne malveillante d'intercepter les informations transitant pendant la négociation et donc ultérieurement de prendre la place du client une fois cette phase passée. Pour usurper l'identité du serveur, le pirate devra également se procurer la clé privée du serveur.

5 Le serveur est maintenant authentifié, la phase de négociation se poursuit.

CHAPITRE 10

Astuces

10.1 Recuperation du passe Root.

10.1.1 Recuperation du passe Root sous Linux.

Sous Linux il vous faudra simplement démarrer le système depuis le cdrom d'installation et selectionner le mode rescue (generalement disponible en tappant f3 puis rescue ou linux-resuce).

Après cette étape il suffit de monter votre partition racine :
 # mount /dev/hda5 /mnt ou mount -t ext2 /dev/hda5 /mnt

et de la chrooter.(*chrooter = déplacer la racine, ie : "/mnt/disque5 = /"*)
 # chroot /mnt

une fois cette opération effectuée vous êtes désormais root sur votre linux, il vous suffira donc de tapper la commande "passwd" pour changer le mot de passe. redémarrer et c'est repartit :).

10.1.2 Recuperation du passe Root sous BSD

Sous les différents *BSD c'est encore plus simple, il suffit de tapper sur la barre espace durant le boot.
Tapper "boot -s" *(boot en single user)*.
Tapper "mount -o rw /" *(accès en ecriture au système de fichier)*.
et enfin Tapper "passwd"

10.2 Recompilation du Noyau

La recompilation peut représenter un élément non négligeable de l'administration Système, en effet le noyau par défaut comporte des options qu'il ne sont pas forcement nécessaires et ou il serait possible d'économiser un peu de mémoire.

Pour se faire il vous faudra récupérer la dernière version stable du noyau linux (2.4.19) sur www.kernel.org et de procéder comme suit :

```
tar xvfz linux-2.4.19.tar.gz -C /usr/src
cd /usr/src/linux
make xconfig (interface graphique de configuration)
make ()
make bzImage (création du noyau)
make modules (compilation des modules)
make modules_install (installation des modules dans /lib/modules)
cd /usr/src/linux/arch/i386/boot/
cp bzImage /boot/vmlinuz-new
```

bien sur il est possible et même plus que probable que vous ayez certains messages d'erreurs au moment de la compilation, auquel cas il faudra tout reprendre depuis l'étape "make xconfig" afin de (de)sélectionner certaines options.

Une fois cette étape terminée il vous faudra modifier le fichier /etc/lilo.conf afin de lui indiquer comment démarrer avec votre nouveau noyau

/etc/lilo.conf

```
...
image=/boot/vmlinuz-new
label=linuxnew
root=/dev/hda5
initrd=/boot/initrd.img
append="devfs=mount"
read-only
```

Bien sur vous devez remplacer "/dev/hda5" par votre partition racine. Pour finir n'oubliez pas de tapper la commande "lilo" qui va réellement modifier le message de boot.

10.3 export DISPLAY

Le mode graphique sur Linux/BSD utilisant Xfree86 étant beaucoup plus puissant que sur Windows, il permet des manipulations telles que l' "export DISPLAY". En effet il est possible lorsque l'on se connecte sur un serveur distant de lui spécifier le "DISPLAY" qu'il doit utiliser.
Par exemple supposons que souhaitiez utiliser "netscape" qui existe sur votre serveur mais pas sur votre machine.

1. Autorisation depuis votre poste de manipuler le DISPLAY par le serveur :
   xhost +IP_du_serveur
   Ex :

```
(seb@ramses)[~]-% xhost +192.168.1.217
192.168.1.217 being added to access control list
```

2. Connexion sur le serveur :
   ssh login@serveur
3. Exportation du DISPLAY :
   export DISPLAY=IP_du_client :0.0
4. exécuter "netscape" :
   netscape
5. Si tout a été configuré correctement netscape apparaîtra sur votre poste client.

Remarque :
l'une des commandes les plus dangereuse aurait été de taper *xhost +* auquel cas vous auriez autorisé tout poste client a prendre possession de votre display, et donc de pouvoir exécuter toute les commandes possibles avec vos droit.

10.4 Serveur RPMS pour Mandrake

Le but de cette astuce est de permettre l'installation en réseau d'entreprise ou personnel via FTP des paquetages RPMS sans pour autant avoir a donner les différents CD

Coté Serveur

Prérequis serveur FTP

Copie des RPMS le choix de l'emplacement où copier les fichiers est sans réelle importance nous conviendrons de copier tous les fichiers rpm dans /var/ftp/pub/RPMS :
cp /mnt/cdrom/Mandrake/RPMS/ /var/ftp/pub/RPMS*

cp /mnt/cdrom/Mandrake/RPMS2/ /var/ftp/pub/RPMS*
etc...

Génération du hdlist lors de cette étape nous allons générer le fichier hdlist que nous inclurons par la suite dans le répertoire /var/ftp/pub/RPMS :
urpmi.addmedia all file ://var/ftp/pub/RPMS
à partir de quoi urpmi va créer le fichier */var/lib/urpmi/hdlist.all.cz*
cp /var/lib/urpmi/hdlist.all.cz /var/ftp/pub/RPMS

Coté Client

suppression des anciens média :
urpmi.removemedia cdrom1
urpmi.removemedia cdrom2
etc...

ajout du nouveau média :
urpmi.addmedia all ftp ://serveur/pub/RPMS with hdlist.all.cz

Ainsi, le poste client n'aura plus qu'à taper la commande "*urpmi paquetage*" pour que tous les fichiers de dépendance soient téléchargées et installés automatiquement.

10.5 Mail 100% anonymes

telnet smtp.wanadoo.fr 25
HELO esmtp
MAIL FROM : j.chirac@elysee.fr
RCPT TO : mail@dequi.fr
DATA
subject : coucou
.
QUIT

10.6 Connexion SSH sans mot de passe

Avertissement : cette manoeuvre qui a pour but de simplifier les connections peut se révéler extrêmement dangereuse si quelqu'un peut avoir un accès complet a votre compte, ex : admin. syst., etc ...
En effet, si vous laissez la passphrase vide alors cette autre personne pourra se connecter partout où vous avez copié votre clef publique il lui faudra simplement récupérer la clef /.ssh/id_dsa

Ccl : Toute votre sécurité résidera sur la sécurité de votre compte.

le but de la manoeuvre est de copier la clef générée par ssh dans les fichiers /.ssh/authorized_keys /.ssh/authorized_keys2 sur le poste serveur. Lors de la génération de clef vous devez taper :

1. Génération des clefs coté client.
   (a) *ssh-kengen* pour générer les clefs /.ssh/identity et /.ssh/identity.pub pour le protocol ssh1

   (b) *ssh-kengen -t dsa* pour générer les clefs /.ssh/id_dsa et /.ssh/id_dsa.pub pour le protocol ssh2

2. ensuite il faut copier ces clefs sur le serveur :
   (a) Dans le cas ou aucune clef n'existe sur le serveur :
      i. Si le répertoire ".ssh" n'existe pas sur le serveur il faudra préalablement le créer par *mkdir .ssh*
      ii. *scp .ssh/identity.pub login@serveur :.ssh/authorized_keys*
      iii. *scp .ssh/id_dsa.pub login@serveur :.ssh/authorized_keys2*
   (b) Dans le cas ou une clef existe déjà sur le serveur :
      i. *scp .ssh/identity.pub login@serveur :.ssh/keys1*
      ii. *scp .ssh/id_dsa.pub login@serveur :.ssh/keys2*
      iii. *ssh login@serveur*
      iv. *cd .ssh*
      v. *cat keys1 » authorized_keys*
      vi. *cat keys2 » authorized_keys2*
      vii. *rm keys1 keys2*

10.7 mirroring OpenBSD-current Mandrake-cooker

10.8 Manipulation des images ISO & Gravure

le but cette astuce et de pouvoir créer, manipuler, graver une image iso. Pour se faire Linux (si cela a été spécifié lors de compilation du noyau) permet de pouvoir monter un périphérique en loopback.

Création d'une image ISO

la création d'une image iso sous Linux/*BSD se fait par le programme mkisofs, qui permet toutes les manipulations possibles autour de l'image.

par exemple pour créer une image non bootage :
mkisofs -J -R -o OpenBSD.iso OpenBSD/

de même pour créer une image bootage :
mkisofs -b 3.1/i386/cdrom31.fs -J -R -o OpenBSD.iso OpenBSD/
-b : fichier de démarrage (chemin relatif).
-J : extension Joilet (windows) pour les fichier de 64 caractères.
-R : extention Rock Ridge.
-o : nom de l'image créée.
OpenBSD/ : répertoire contenant les fichiers a inclure.

Options supplémentaires :
-c boot.catalog : création du catalogue de boot.
-l : permet les noms longs (32 caractères).
-r : mets les permissions sur les fichiers.
-L : permet les noms de fichier commençant par un ".".

Manipulation d'une image ISO

Sous Linux	
mount -t iso9660 image.iso /iso -o loop=loop0	pour monter l'image dans le répertoire /iso.
et *umount /iso*	pour démonter cette image.
Sous *BSD	
vnconfig -c vnd0 image.iso	
mount -t cd9660 /dev/vnd0c /mnt	pour monter l'image dans le répertoire /mnt
et *umount /mnt*	
vnconfig -u vnd	pour démonter l'image
Sous Solaris	
mount -F fbk -o ro,type=hsfs	
/dev/fbk0 :output.iso /mnt	pour monter l'image
et *umount /mnt*	pour démonter l'image

Gravure d'une image ISO sur cdrom

Pour se faire il faudra commencer par exécuter la commande "cdrecord –scanbus" si vous n'obtenez pas une liste contenant votre graveur, il vous faudra alors, dans le cas d'un graveur IDE, charger le module ide-scsi par la commande "modprobe ide-scsi".

il vous faudra alors noter le numero du périphérique qui sera nécessaire pour l'étape suivante.

cdrecord -dev=1,0,0 -speed=12 -v image.iso
-dev=1,0,0 : numero du device.
-speed=12 : spécifiant la vitesse de gravure.
-v : mode affichant tous les messages.
-eject : éjection du cdrom aprés gravure.
image.iso l'image iso que vous voulez graver.

10.9 Signature et vérification d'intégrité des fichiers via MD5/GnuPG

10.9.1 via MD5

Pour créer une empreinte il suffit de tapper la commande :
md5sum image.iso
bc94f333a9090cc298ab

ou pour plusieurs fichiers
md5sum *.iso
bc94f333a9090cc298ab
fc94sdfsfd300cc298ab

il est possible également de générer le md5 checksum et de le stocker dans un fichier par :
md5sum *.iso > fichier.md5

ensuite pour vérifier :
md5sum -c fichier.md5

10.9.2 via GnuPG

Les signatures PGP reposent sur une méthode appelée chiffrage à clef publique. Cette méthode permet à un développeur ou un groupe de développeurs de signer de manière électronique un fichier ou un message en utilisant une clef secrète ou clef privée. La clef privée ne devra et ne sera jamais divulguée publiquement. Pour vérifier une signature, nous utiliserons la petite soeur de la clef privée appelée clef publique. Cette dernière est disponible pour tous et habituellement répertoriée dans un serveur de clefs.

Dans notre exemple, nous avons besoin de la clef publique correspondant au projet de Kernel Linux. Les développeurs du kernel étant des personnes très consciencieuses, une page spécifique a été mise en place signalant le numéro de clef à utiliser (http ://www.kernel.org/signature.html) : 0x517D0F0E.

Pour poursuivre, il vous faudra utiliser le logiciel GnuPG. C'est cet utilitaire qui vous permettra de récupérer la clef publique et de vous en servir pour vérifier la signature de l'archive /linux-2.4.4.tar.gz/. Commençons donc par récupérer la clef auprès du serveur de clefs :

```
$ gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
gpg: requête de la clé 517D0F0E de wwwkeys.pgp.net...
gpg: clé 517D0F0E: clé publique importée
gpg:        Quantité totale traitée: 1
gpg:                       importée: 1

Ceci fait, nous pouvons vérifier la validité de l'archive en combinant
toutes les informations (clef public, archive et fichier signature) :

$ gpg --verify linux-2.4.4.tar.gz.sign linux-2.4.4.tar.gz
gpg: Signature faite sam 28 avr 2001 03:48:04 CEST avec une clé DSA ID 517D0F0E
gpg: Bonne signature de "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>"
...
```

"Bonne signature...", c'est parfait, notre archive a été vérifiée avec succès. Nous pouvons faire confiance en son contenu. En réalité, ceci n'est pas tout à fait exact. Bien qu'il soit extrêmement difficile de "casser" une telle clef, il est toujours possible que quelqu'un ait piraté le serveur *www.kernel.org* pour y glisser un faux numéro de clef ou encore *wwwkeys.pgp.net* pour fournir une fausse clef publique. Bien sûr, ceci ne passerait pas inaperçu : pas plus que quelques heures et des alertes seraient lancées de toute part. Sachez seulement que cela reste toujours possible et qu'il n'existe pas de sécurité absolue.

Notez qu'il existe des interfaces utilisateur très conviviales pour GnuPG ainsi que des modules vous permettant d'automatiser les vérifications depuis votre client ftp et/ou mail. Les serveurs de clefs sont nombreux sur Internet et théoriquement tous reliés entre eux. Si vous désirez récupérer une clef publique afin de vérifier la signature de l'auteur, il vous suffira de faire une simple recherche avec son nom. Certes, dans la plupart des cas, l'auteur spécifie son numéro de clef sur sa homepage ...

10.10 Réinstallation de lilo sans disquette

Pour ce faire il faudra au préalable se procurer un CD de démarrage d'une version Linux proposant un mode rescue comme Mandrake ou Red Hat.
Durant cette réinstallation de lilo, nous supposerons que votre disque dur se trouve en tant que "master" sur la première nappe de votre carte mère et donc sur le disque */dev/hda*, bien sûr il peut en être autrement :

“master” (maître) sur la première nappe	/dev/hda
“slave” (esclave) sur la première nappe	/dev/hdb
“master” sur la première nappe	/dev/hdc
“salve” sur la première nappe	/dev/hdd
“premier disque dur SCSI”	/dev/sda
“second disque dur SCSI”	/dev/sdb
etc	

1. démarrage en boutant avec le CD-ROM.
2. tapez “F1” puis “rescue” au prompt lilo.
3. à partir de ce moment le système va se charger depuis le cdrom et créer un système de fichier virtuel directement dans la mémoire.
4. à cette étape il vous faudra connaître la partition utilisée par Linux (souvent /dev/hda5) ou procéder comme suit :
   (a) Pour connaître l’emplacement de votre partition Linux vous allez taper *fdisk -l /dev/hda*
   (b) Si vous obtenez quelque chose ressemblant à :

```
root@kset_mdk /home/seb # fdisk -l /dev/hda

Disque /dev/hda : 255 têtes, 63 secteurs, 4865 cylindres
Unites = cylindres sur 16065 * 512 octets

Peripherique Amorce    Debut       Fin    Blocs   Id  Systeme
/dev/hda1              1       952   7646908+  a9  NetBSD
/dev/hda2            953      1576   5012280   a6  OpenBSD
/dev/hda3           1577      2959  11108947+   c  Win95 FAT32 (LBA)
/dev/hda4           2960      4866  15314481    5  Etendue
/dev/hda5           2962      3637   5429938+  83  Linux
/dev/hda6           3638      3691    433723+  82  Echange Linux
```

   Il y a de très fortes chances pour que votre partition Linux se situe sur */dev/hda5* vous pourrez donc directement passer à la section “5”.
   (c) Par contre si vous obtenez quelque chose ressemblant à :

```
root@kset_mdk /home/seb # fdisk -l /dev/hda

Disque /dev/hda : 255 tetes, 63 secteurs, 4865 cylindres
Unites = cylindres sur 16065 * 512 octets

Peripherique Amorce    Debut       Fin    Blocs   Id  Systeme
/dev/hda1              1       952   7646908+  a9  NetBSD
/dev/hda2            953      1576   5012280   a6  OpenBSD
/dev/hda3           1577      2959  11108947+   c  Win95 FAT32 (LBA)
/dev/hda4           2960      4866  15314481    5  Etendue
/dev/hda5           2962      3637   5429938+  83  Linux
/dev/hda6           3638      3691    433723+  82  Echange Linux
/dev/hda8           3692      4318   5036314+  83  Linux
/dev/hda9           4319      4866   4398282   83  Linux
```

   Même s’il y a de très fortes chances pour que votre partition Linux se situe sur */dev/hda5* vous devrez monter chacunes de partitions pour vous en assurer, en procédant par exemple de la façon suivante :

 # mkdir /mnt/disque5
 # mkdir /mnt/disque8
 # mkdir /mnt/disque9
 # mount /dev/hda5 /mnt/disque5 ou mount -t ext2 /dev/hda5 /mnt/disque5
 # mount /dev/hda8 /mnt/disque8
 # mount /dev/hda9 /mnt/disque9

et ensuite vous allez vous rendre dans chaques partitions */mnt/disque5, /mnt/disque8, /mnt/disque9* afin de savoir quelle partition correspond à celle qui contient le lilo que vous voulez restaurer. Une fois celle-ci identifiée vous pourrez vous rendre à l'étape "6".

5. Dans cette on montera la partition qui contient Linux (si cela n'est pas déjà fait) par :
   # mount /dev/hda5 /mnt/disque5 ou mount -t ext2 /dev/hda5 /mnt/disque5

6. Ensuite il faudra chrooter la partition qui contient Linux (*chrooter = déplacer la racine, ie : "/mnt/disque5 = /"*)
7. Modifier si cela doit l'être le fichier lilo.conf par : *vi /etc/lilo.conf*
8. et enfin exécuter le programme lilo pour qu'il se re-installe sur le "mbr" ou l' "endoit" spécifier dans "boot=/dev/hda" par :
   # lilo
9. redémarrage de la machine :
   # reboot
10. à partir de ce moment si tout a été exécuter correctement vous pourrez redémarrer Linux sans problèmes.

Remarque

Mandrake 8.2

Le CD-ROM de démarrage de la Mandrake8.2, permet de restaurer directement lilo dans le menu "rescue", par contre si vous vous souhaitez faire des modifications sur */etc/lilo.conf* ou sur tout autre fichier il vous faudra refaire les étapes précédentes.

CHAPITRE 11

Historique des changements

11.1 DONE

fin septembre 2002 :	Astuces	Intégrité signatures,MD5.
11 octobre 2002 :	Administration Réseaux	IP aliasing.
11 octobre 2002 :	Administration Réseaux	Sendmail normal/cyrus.

11.2 TODO

Système	Exemple fstab
Firewall	Iptables, ipf.conf NetBSD & FreeBSD
Admin. Reseaux	Mise en place de NIS,NYS. Mirroring, apt-get mirroring. Cyrus. Webmail. VPN. IPsec.
Crypto	Protocole RSA.
Astuces	aspell.

CHAPITRE 12

A propos

Pour toutes remarques ou rectifications sur cette documentation merci de m'envoyer un mail a l'adresse dupont_s@epita.fr avec pour sujet [Cours Administration].

Cette Documentation a été réalisée sous LaTeX.

INDEX