

Province de Hainaut
HEPCUT – ISIPH
Catégorie Technique



www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

Administration et sécurité des réseaux.

Jean-François Challe.

I. Gestion des comptes utilisateurs.	1
1. Introduction.	1
2. Les attributs d'un fichier.	1
3. Création de comptes utilisateurs.	2
3.1. Principe de base.	2
3.2. Script de création des comptes.	2
3.3. Options de la commande adduser.	3
3.3.1. Les commentaires.	3
3.3.2. Le répertoire de base.	5
3.3.3. Le shell utilisé.	5
3.3.4. Les groupes.	5
3.3.5. Spécification de l'UID.	6
3.3.6. Date d'expiration.	6
4. Modification d'un compte utilisateur.	6
4.1. La commande usermod.	6
4.2. La commande chage.	7
4.3. La commande passwd.	8
5. Suppression d'un compte utilisateur.	8
6. La gestion des groupes.	8
6.1. Création d'un groupe.	8
6.2. Modification d'un groupe.	9
6.3. Suppressions d'un groupe.	9
7. Arrêt momentané des connexions.	9
8. Message du jour.	9
9. Gestion des quotas.	10
9.1. Introduction.	10
9.2. Processus de mise en place des quotas.	10
9.3. Attribution des quotas.	11
9.3.1. Quotas d'un utilisateur.	11
9.3.2. Quotas d'un groupe.	12
9.3.3. Période de grâce.	12
9.3.4. Attribution des quotas à plusieurs utilisateurs.	12
9.4. Consultation des quotas.	13
II. La séquence de démarrage du système.	15
1. Principe de la séquence de démarrage.	15
2. LILO.	16
2.1. Les arguments du démarrage.	16
2.2. Le fichier /etc/lilo.conf.	17
2.3. Le mot clé password.	17
3. Le processus init.	19
4. Les scripts de démarrage.	20
5. L'arrêt du système.	20
6. Le redémarrage du système.	21

III. Configuration de l'interface réseau.	23
1. La commande ifconfig.	23
2. Structure des fichiers /etc/hosts et /etc/networks.	23
3. Détermination du nom de l'interface.	24
4. Vérification de l'interface avec ifconfig.	26
5. Affectation un masque de sous-réseau.	26
6. Affectation de l'adresse broadcast.	27
7. Autres options.	27
7.1. Activer et désactiver l'interface.	27
7.2. ARP et trailers.	28
7.3. Metric.	28
7.4. MTU.	28
7.5. Mode promiscuous.	29
8. Configuration permanente.	29
9. Linuxconf.	30
IV. Configuration du routage.	31
1. Configuration générale du routage.	31
2. Table de routage minimale.	31
3. Création d'une table de routage statique.	32
3.1. Ajouter des routes statiques.	33
3.2. Ajouter des routes statiques à l'amorçage.	34
4. Les protocoles de routage internes.	35
4.1. Le protocole RIP.	35
4.2. RIP version 2.	38
4.3. OSPF : Open Shortest Path First.	38
5. Les protocoles de routage externe.	40
5.1. EGP : Exterior Gateway Protocol.	40
5.2. BGP : Border Gateway Protocol.	40
5.3. Choisir un protocole de routage.	40
6. gated : Gateway Routing Daemon.	41
7. Configurer gated.	42
7.1. Exemple de fichier /etc/gated.conf.	42
7.1.1. Configuration d'une machine.	42
7.1.2. Configuration d'une passerelle interne.	43
7.1.3. Configuration d'une passerelle externe.	44
V. Configuration du DNS.	47
1. Le fichier /etc/hosts.	47
2. BIND : Berkeley Internet Name Domain.	47
3. Configurer le résolveur.	48
4. Initialisation des données du DNS.	48
4.1. Les fichiers de la base de données.	48

4.1.1. Les enregistrements SOA (Start Of Authority).	49
4.1.2. Les enregistrements NS (Name Server).	49
4.1.3. Les enregistrements d'adresse et d'alias.	49
4.1.4. Les enregistrements PTR (Pointer).	50
4.1.5. Les données de la zone racine.	51
4.2. Le fichier de configuration de BIND.	53
4.3. Les abréviations.	53
4.4. Démarrage d'un serveur-maître primaire.	56
4.5. Démarrage d'un serveur-esclave.	57
4.6. Les champs de l'enregistrement SOA.	57
4.7. Gestion de plusieurs domaines.	58
5. Gestion des sous-domaines.	58
6. La sécurité.	59
6.1. Restriction des requêtes.	59
6.1.1. Restriction sur toutes les requêtes.	59
6.1.2. Restriction des requêtes concernant une zone.	60
6.2. Contrôle des transferts de zones.	60
6.2.1. Limitation globale du transfert.	60
6.2.2. Limitation du transfert à une zone.	61
6.3. Exécution de BIND par un utilisateur sans privilège.	62
7. La commande nslookup.	62
7.1. Recherche de différents types de données.	64
7.2. Réponses faisant autorité.	64
7.3. Changement de serveur.	65
7.4. Visualisation des paquets de requête et de réponse.	66
7.5. Recherche à la manière d'un serveur de noms.	67
7.6. Transferts de zone.	69
VI. Le courrier électronique.	71
1. Introduction.	71
2. La structure d'un message.	71
3. Principe de livraison des mails.	72
4. Le routage du courrier.	72
5. Sendmail.	73
5.1. Les fonctions de sendmail.	73
5.2. Le démon sendmail.	73
5.3. Configuration de sendmail.	73
5.3.1. Configurer le mail local.	74
5.3.2. Configurer le mail distant.	75
5.3.3. Agir comme serveur relais.	75
5.3.4. Options supplémentaires.	77
5.4. Sendmail et les alias.	79
5.5. Utilisation d'un .forward personnel.	80
5.6. Configuration de la sécurité.	80
5.6.1. Les commandes vrfy et expn.	80
5.6.2. Restriction d'accès à la file des messages.	81
5.6.3. Forcer l'identification du client.	81

5.6.4. Message d'invite de sendmail.	81
5.6.5. Contrôler les connexions SMTP avec TCP Wrappers.	82
5.6.6. Lutter contre les attaques de déni de service.	82
5.6.7. Utilisation d'un shell restreint.	82
VII. NFS.	85
1. Introduction.	85
2. Configuration de NFS.	85
3. Exportation d'un système de fichiers.	85
3.1. Règles d'exportation des systèmes de fichiers.	85
3.2. Exportation simple d'un système de fichiers.	85
4. Montage d'un système de fichiers.	86
5. Les options d'exportation.	86
6. Les options de montage.	87
6.1. Les arguments généraux.	87
6.2. Les arguments spécifiques.	88
7. Le montage permanent.	88
8. La résolution des liens symboliques.	88
VIII. NIS.	89
1. Introduction.	89
2. Les notions de maîtres, esclaves et clients.	89
3. Fondements de la gestion NIS.	90
3.1. Installation du serveur NIS maître.	90
3.2. Configuration des clients NIS.	91
3.3. Ajout d'un serveur NIS esclave.	92
4. Les commandes-clients.	93
4.1. ypmatch.	93
4.2. ypcat.	93
4.3. ypwhich.	93
4.4. yppasswd.	94
IX. LPD.	95
1. Introduction.	95
2. Le fichier /etc/printcap.	95
3. Sécurité avec LPD.	96
4. Utilisation de LPD.	97
4.1. Envoyer une requête d'impression.	97
4.2. Gestion du démon d'impression.	97
4.3. Consultation de la file d'attente.	97
4.4. Suppression d'une requête.	98
X. DHCP.	99
1. Introduction.	99
2. Le fichier dhcpd.conf.	99

2.1. Configuration de base.	99
2.2. Lancement du serveur DHCP.	100
2.3. Utilisation des adresses physiques.	101
XI. Samba.	103
1. Introduction.	103
1.1. Partage d'un service disque.	103
1.2. Partage d'une imprimante.	105
1.3. Visualisation des ressources sous UNIX.	106
1.4. Prise en main d'un réseau SMB.	106
1.4.1. Maîtrise de NetBIOS.	106
1.4.2. Attribution d'un nom.	106
1.4.3. Anatomie d'un nom.	107
1.5. Implémentations par Microsoft.	107
1.5.1. Domaines Windows.	107
1.5.2. Exploration.	108
1.5.3. Le service WINS.	108
2. Configuration de Samba.	109
2.1. Fichier de configuration de base pour Samba.	109
2.2. Démarrage du serveur Samba.	110
3. Partages de disque.	110
3.1. Structure du fichier de configuration.	110
3.2. Les sections du fichier de configuration.	110
3.2.1. La section global.	110
3.2.2. La section homes.	111
3.2.3. La section printers.	111
3.3. Configuration du serveur.	111
3.4. Les options réseau.	112
4. Exploration et partages de disques avancés.	113
4.1. L'exploration.	113
4.1.1. Blocage de l'exploration.	113
4.1.2. Election d'explorateurs.	113
4.2. Différences entre systèmes de fichiers.	115
4.2.1 Droits d'accès et fichiers cachés.	115
4.3. Permissions et attributs des fichiers sous MS-DOS et sous UNIX.	116
5. Utilisateur, sécurité et domaines.	116
5.1. Utilisateurs et groupes.	117
5.2. Sécurité des authentifications.	119
5.2.1. Sécurité de niveau partage.	119
5.2.2. Sécurité de niveau utilisateur.	119
5.2.3. Sécurité de niveau serveur.	119
5.2.4. Sécurité de niveau domaine.	120
5.2.5. Ajout d'un serveur Samba à un domaine Windows NT.	121
5.3. Domaine Windows.	121
5.3.1. Introduction.	121
5.3.2. Configuration PDC pour des clients Windows NT.	122
6. Impression.	124
7. Résolution de noms.	124

XII. APACHE.	127
1. Introduction.	127
2. Premier lancement du serveur Web.	127
3. Un véritable site.	128
3.1. Interprétation du code HTML.	128
3.2. Chargement automatique du fichier index.html.	129
3.3. Gestion de plusieurs répertoires.	129
3.4. Utilisation des CGI.	130
3.5. Remarque.	130
4. Gestion de plusieurs sites.	131
XIII. SQUID.	133
1. Introduction.	133
2. Protocoles utilisés.	133
3. Configuration matérielle.	133
4. Installation et mise en route.	134
5. Contourner SQUID.	134
6. Configuration de SQUID.	134
6.1. Ports de communication.	134
6.2. Taille du cache.	134
6.3. Emplacement du cache et de l'historique.	135
6.4. Support de programmes externes.	136
6.5. Timeouts	136
6.6. Contrôles d'accès.	136
6.7. Paramètres administratifs.	138
6.8. Vérification de la connexion.	139
7. Configuration avancée.	139
7.1. Bannir des sites.	139
7.2. Interdire le téléchargement de fichier exécutables.	140
7.3. Authentification des utilisateurs.	140
XIV. IPCHAINS.	143
1. Introduction.	143
2. Choix d'une politique de filtrage.	143
3. Rejeter ou refuser un paquet.	144
4. Filtrage des paquets entrants.	144
4.1. Filtrage d'adresse source distante.	144
4.1.1. Spoofing d'adresses source et adresses illégales.	144
4.1.2. Bloquer les sites problématiques.	145
4.1.3. Limiter les paquets entrants à des hôtes sélectionnés.	145
4.2. Filtrage local d'adresses destination.	145
4.3. Filtrage de port source distant.	145
4.4. Filtrage de port destination local.	146
4.5. Filtrage de l'état de connexion TCP.	146
4.6. Sondes et balayages.	146

4.6.1. Balayages généraux de ports. _____	146
4.6.2. Balayages ciblés de ports. _____	146
4.7. Attaques par déni de service. _____	146
4.7.1. TCP et SYN flooding. _____	147
4.7.2. Ping flooding. _____	147
4.7.3. UDP flooding. _____	147
4.7.4. Bombes ICMP-redirect. _____	147
4.7.5. Autres attaques par déni de service. _____	147
5. Filtrage de paquets sortants. _____	148
5.1. Filtrage d'adresse source locale. _____	148
5.2. Filtrage d'adresse de destination distante. _____	148
5.3. Filtrage de port source local. _____	148
5.4. Filtrage de port destination distant. _____	149
5.5. Filtrage de l'état des connexions TCP sortantes. _____	149
6. Services réseau privés/publics. _____	149
7. Construction d'un firewall. _____	149
7.1. Le programme ipchains. _____	149
7.2. Initialisation du firewall. _____	150
7.2.1. Les constantes symboliques. _____	151
7.2.2. Supprimer les règles existantes. _____	151
7.2.3. Définir la politique par défaut. _____	151
7.2.4. Activer l'interface loopback. _____	152
7.2.5. Le spoofing et les mauvaises adresses source. _____	152
7.3. Filtrage des messages d'état et de contrôle ICMP. _____	154
7.3.1. Messages d'erreur d'état et de contrôle. _____	154
7.3.2. Messages de contrôle ping Echo Request et Echo Reply. _____	155
7.4. Protection des services sur les ports TCP sans privilège. _____	156
7.4.1. Interdire les connexions Open Window. _____	156
7.4.2. Interdire les connexions X Window. _____	156
7.4.3. Interdire les connexions au serveur SOCKS. _____	156
7.5. Protection des services sur les ports UDP sans privilège. _____	156
7.6. Activation des services Internet de base. _____	157
7.6.1. Permettre DNS. _____	157
7.6.2. Filtrer les identifications. _____	158
7.7. Activer des services TCP courants. _____	158
7.7.1. Le courrier électronique. _____	158
7.7.2. Accéder aux services Usenet. _____	161
7.7.3. Le service telnet. _____	161
7.7.4. Le service ssh. _____	162
7.7.5. Le service ftp. _____	162
7.7.6. Les services Web. _____	163
7.7.7. Le service finger. _____	164
7.7.8. Le service whois. _____	164
7.7.9. Le service gopher. _____	165
7.7.10. Le service WAIS. _____	165
7.8. Habilitier les services UDP courants. _____	165
7.8.1. Le programme traceroute. _____	165
7.8.2. Accéder à un serveur DHCP. _____	166
7.8.3. Accéder à un serveur Network Time. _____	166

7.9. Journaliser des paquets entrants refusés. _____	166
7.10. Refuser l'accès aux sites problématiques. _____	166
7.11. Activer l'accès au réseau local. _____	167
7.12. Installer le firewall. _____	167
<i>Bibliographie.</i> _____	168



I. Gestion des comptes utilisateurs.

1. Introduction.

La gestion des comptes utilisateurs est l'un des travaux absolument essentiels dévolu aux administrateurs. Gérer les utilisateurs et les mots de passe sont une tâche de base à laquelle tout administrateur doit attacher une attention particulière car une mauvaise gestion peut gravement affaiblir la sécurité d'un système.

Même dans le cas d'un système offrant des services réseau ne nécessitant pas la gestion de plusieurs utilisateurs, comme le DNS par exemple, il est essentiel de disposer d'au moins un compte autre que celui réservé à l'administrateur du système (root). L'utilisation du compte root doit être limitée à des tâches spécifiques qui ne peuvent pas être réalisées sous l'identité d'un utilisateur normal. Lorsqu'une personne est connectée à un système UNIX sous l'identité root, il a le droit d'effectuer toutes les opérations sans aucune contrainte. Une utilisation abusive de cette identité peut corrompre complètement l'intégrité du système par le biais de fausses manœuvres. Tout administrateur système doit toujours disposer du compte root pour réaliser les tâches administratives critiques mais également d'un compte utilisateur traditionnel afin d'effectuer de petits travaux n'exigeant pas de privilèges particuliers.

Après l'installation d'un système Linux, le compte administrateur est créé et un mot de passe y est associé. Si l'administrateur se connecte à la console, il aura toujours la faculté de s'identifier et de réaliser les travaux nécessaires à la bonne marche de ce système. Néanmoins, dans de nombreux cas, l'administrateur n'utilise que rarement la console. En règle générale l'administrateur se connecte par l'intermédiaire d'un poste distant. Pour des raisons de sécurité, root ne peut se connecter directement qu'à la console. Cette sécurité est établie au travers du fichier de configuration /etc/securitytty. Ce fichier contient la liste des terminaux qui sont considérés comme des points d'entrée sécurisés du système. Ces terminaux n'engendrent pas de connexion réseau, ce qui rend impossible une écoute de la ligne à la recherche du mot de passe du super-utilisateur. Voici un exemple de fichier /etc/securitytty :

```
[root@p200 /etc]# cat securitytty
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
```

Ce fichier décrit que les huit consoles sont des postes de travail sécurisés permettant à l'administrateur de se connecter sans risque. L'effacement de ce fichier peut introduire un trou dans la sécurité. En l'absence du fichier /etc/securitytty, l'administrateur est autorisé à se connecter directement au système à partir de n'importe quel poste de travail.

2. Les attributs d'un fichier.

Une modification ou un effacement du fichier /etc/securitytty peut engendrer une faille dans le système. Théoriquement, seul l'administrateur a les autorisations suffisantes pour effacer ou pour modifier ce fichier. Néanmoins, la sécurité d'un système consiste à placer le plus de barrières possibles de manière à ralentir autant qu'il se peut le processus d'effraction.

Le fichier /etc/securitytty doit donc être protégé contre les risques d'effacement ou de modification. Les commandes UNIX traditionnelles ne permettent pas la sécurisation d'un fichier particulier. De nombreux concepteurs de systèmes UNIX ont apporté leurs extensions de manière à offrir plus de sécurité. Le système Linux ne fait pas exception à cette règle et propose une commande permettant de sécuriser un fichier. Il s'agit de la commande `chattr`. Elle possède de nombreuses options dont une protégeant un fichier contre les modifications et l'effacement. Voici un exemple de sécurisation du fichier /etc/securitytty :

```
[root@p200 /etc]# chattr +i securitytty
```

Comme le montre la session interactive suivante, lorsqu'un fichier possède l'attribut `i`, même l'administrateur système ne peut le supprimer :

```
[root@p200 /etc]# rm securetty
rm: détruire le fichier protégé en écriture `securetty'? y
rm: Ne peut délier `securetty'. : Opération non permise
```

Dans certains cas, l'administrateur doit connaître les attributs associés à un fichier. La commande `lsattr` permet de visualiser tous les attributs affectés à un fichier :

```
[root@p200 /etc]# lsattr securetty
----i--- securetty
```

Toujours pour des raisons de sécurité, seul l'administrateur du système a le droit d'employer la commande `lsattr` pour visualiser les attributs affectés à un fichier. Voici un exemple d'utilisation de la commande `lsattr` par un utilisateur normal :

```
[jfc@p200 /etc]$ lsattr securetty
lsattr: Permission non accordée While reading flags on securetty
```

Les tâches administratives peuvent conduire l'utilisateur `root` à modifier un fichier dont les attributs empêchent son effacement. La commande `chattr` avec l'option `moins` permet d'enlever une protection. Voici un exemple de suppression de l'attribut `i` placé sur le fichier `/etc/securetty` :

```
[root@p200 /etc]# chattr -i securetty
[root@p200 /etc]# rm securetty
rm: détruire `securetty'? y
[root@p200 /etc]# ls securetty
ls: securetty: Aucun fichier ou répertoire de ce type
```

3. Création de comptes utilisateurs.

3.1. Principe de base.

En la qualité de `root`, il est possible d'ajouter des utilisateurs au système par le biais de la commande `adduser`. La version minimale de cette commande consiste à donner en paramètre le nom de l'utilisateur à créer.

```
[root@p200 /root]# adduser jfc
```

Les options de la commande `adduser` permettent de peaufiner le processus de création d'un nouvel utilisateur. La création d'un nouveau compte utilisateur par le biais de la version simplifiée de la commande donne des valeurs par défaut aux options qui ne sont pas spécifiées. C'est ainsi que le compte de l'utilisateur ne peut pas être immédiatement employé aucun mot de passe valide n'y a été associé. L'administrateur du système doit alors employer la commande `passwd` pour donner un mot de passe valide au nouvel utilisateur créé. Voici un exemple d'attribution d'un mot de passe à un compte utilisateur :

```
[root@p200 /root]# passwd jfc
Changing password for user jfc
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

3.2. Script de création des comptes.

Dans un contexte scolaire où il y a de nombreux comptes utilisateurs à créer au même moment, la procédure qui vient d'être mentionnée peut s'avérer très lourde. Non seulement il faut employer la commande `adduser` pour créer un compte utilisateur mais également la commande `passwd` pour attribuer un mot de passe à l'utilisateur créé. Cette modification du mot de passe impose de dactylographier deux fois le mot de passe. Afin de rendre le processus de création d'un utilisateur moins lourd, il est possible de préciser un argument à la commande `passwd` de sorte qu'elle puisse accepter directement un mot de passe.

L'option `--stdin` informe la commande `passwd` de ce qu'elle doit utiliser son entrée standard pour obtenir le mot de passe en clair. L'entrée standard peut, comme dans la majorité des cas des commandes UNIX, obtenir ses informations par l'intermédiaire d'un pipe.

Supposons que le nom d'un utilisateur et son mot de passe soient dérivés de son nom et de son prénom. Le nom de l'utilisateur est formé de la première lettre de son prénom suivi de son nom. Le mot de passe est le nom de l'utilisateur écrit à l'envers. Dans ce cas de figure, il est possible de placer dans un fichier texte, le prénom et le nom de plusieurs utilisateurs qui doivent recevoir un compte. Il suffit alors d'écrire un script UNIX chargé de créer les comptes des utilisateurs. Voici un exemple de script capable de créer automatiquement les comptes utilisateurs à partir d'un fichier de données :

```
#!/bin/bash
read a b
while [ "$a" != "" ]; do
  a=`expr substr "$a" 1 1`
  username=$a$b
  l=`expr length $b`
  pass=""
  while [ "$l" != 0 ]; do
    p=`expr substr $b $l 1`
    l=`expr $l - 1`
    pass=$pass$p
  done
  pass=$pass$a
  adduser $username > /dev/null
  echo $pass | passwd --stdin $username >/dev/null
  read a b
```

3.3. Options de la commande adduser.

3.3.1. Les commentaires.

Lors de la création d'un utilisateur, l'administrateur donne généralement des renseignements complémentaires sur l'identité de la personne qui utilisera le compte. Cette information est généralement constituée du prénom et du nom de la personne utilisatrice. Ce type d'information étant généralement constituée d'au moins deux mots séparés par un espace, il ne faut pas que la commande `adduser` considère ces informations comme étant deux arguments distincts. Pour contrecarrer le principe général de détection des différents arguments sur base de l'espacement, il suffit de placer le prénom et le nom de l'utilisateur entre guillemets. Cette identité doit être précédée de l'option `-c` afin d'informer la commande `adduser` de ce que l'argument suivant est le commentaire à mémoriser. Voici la commande de création d'un utilisateur avec un argument commentaire :

```
[root@p200 /root]# adduser -c "Jean-Francois Challe" jchalle
```

Toute commande `adduser` ajoute les renseignements relatifs au nouvel utilisateur dans les fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`. Voici un extrait de ces trois fichiers :

```
[root@p200 /etc]# cat passwd
jchalle:x:502:502:Jean-Francois Challe:/home/jchalle:/bin/bash
[root@p200 /etc]# cat shadow
jchalle:!!:11642:0:99999:7:::
[root@p200 /etc]# cat group
jchalle:x:502:
```

Le fichier texte `passwd` contient la liste des comptes existants sur le système ainsi que des informations utiles sur ces comptes, comme l'identification de l'utilisateur et de son groupe, la localisation de son répertoire de travail, etc... Souvent, ce fichier contient aussi le mot de passe chiffré de l'utilisateur. Le fichier des mots de passe doit permettre à tout utilisateur de le lire. Par contre, l'écriture dans ce fichier ne peut être effectuée que par le super-utilisateur.

Autrefois, aucun problème de sécurité ne se posait au sujet de ce droit général en lecture. Chacun pouvait consulter les mots de passe cryptés le matériel étant beaucoup trop lent pour déchiffrer les informations. De plus, le principe de base d'UNIX reposait sur une communauté soudée d'utilisateurs sans intentions néfastes. Actuellement, il est de plus en

plus recommandé d'utiliser des systèmes de masquage des mots de passe comme `shadow`. Le fichier `/etc/shadow` contient les mots de passe chiffrés, des utilisateurs. Contrairement au fichier `/etc/passwd`, `/etc/shadow` n'est accessible que par le super-utilisateur du système. Lorsque le masquage des mots de passe est activé, le fichier `/etc/passwd` contient un « x » en lieu et place du mot de passe chiffré.

Une entrée du fichier `/etc/passwd` est de la forme suivante :

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

Chaque champ est séparé du suivant par le caractère « : ». Voici la signification de ces différents champs :

- `account` : ce champ spécifie le nom que l'utilisateur emploiera pour se connecter ;
- `passwd` : dans le cas où les mots de passe masqués ne sont pas activés, ce champ doit contenir la représentation encrytée du mot de passe ;
- `UID` : ce champ contient une valeur entière permettant d'identifier l'utilisateur de manière univoque ;
- `GID` : ce champ contient également une valeur entière permettant d'identifier de manière univoque le groupe auquel l'utilisateur appartient. Ce champ doit correspondre à une entrée du fichier `/etc/group` ;
- `GECOS` : ce champ est optionnel et n'a qu'un rôle informatif. Il contient généralement le nom complet de l'utilisateur. `GECOS` signifie General Electric Comprehensive Operating System. Cette signification est donc un reliquat des anciens systèmes d'exploitation ;
- `directory` : les utilisateurs devant se connecter au système pour y travailler doivent disposer d'un répertoire de travail dans lequel ils stockeront leurs fichiers. L'emplacement de ce répertoire est précisé par ce champ dans la hiérarchie du système de fichiers ;
- `shell` : si un utilisateur doit être à même d'employer des commandes UNIX après sa connexion, il doit disposer d'un interpréteur de commandes par défaut. Le champ `shell` permet de préciser la localisation de l'interpréteur de commandes exploité par l'utilisateur.

Le fichier texte `/etc/shadow` contient les champs suivants :

- le nom de connexion de l'utilisateur ;
- le mot de passe crypté de l'utilisateur ;
- le nombre de jours écoulés entre le premier janvier 1970 et le jour du changement du mot de passe ;
- le nombre minimum de jours avant que l'utilisateur soit obligé de changer de mot de passe ;
- le nombre maximum de jours restant avant que le mot de passe soit changé ;
- le nombre de jours précédant l'expiration d'un mot de passe. Durant cette période, l'utilisateur est averti de ce que son mot de passe doit être changé ;
- le nombre de jours passés depuis l'expiration du mot de passe. Le compte utilisateur est alors désactivé ;
- la date après laquelle l'utilisateur ne pourra plus se connecter. Cette date est exprimée sous la forme du nombre de jours écoulés depuis le premier janvier 1970 ;
- un champ réservé.

Le fichier texte `/etc/group` définit les groupes auxquels appartient chaque utilisateur. Le format de chaque ligne de ce fichier est le suivant :

```
nom du groupe:mot de passe:GID:liste d'utilisateurs
```

Voici la signification des différents champs :

- tout groupe possède un nom. Généralement, tout utilisateur fait partie d'au moins un groupe dont le nom est identique à son nom de compte. Par défaut, l'utilisateur est seul dans son groupe ;
- généralement aucun mot de passe n'est associé à un groupe. Dans ce cas, le champ correspondant est vide ;
- le `GID` est une valeur entière identifiant un groupe de manière univoque ;
- la liste des utilisateurs comporte un ensemble de noms de comptes séparés par une virgule. Cela permet de définir l'ensemble des utilisateurs qui appartiennent à un groupe.

3.3.2. Le répertoire de base.

Lorsque la commande `adduser` est utilisée sans précision du répertoire de travail de l'utilisateur, le système crée un répertoire dans `/home` dont le nom est identique au login. De nombreux administrateurs placent le répertoire de travail des utilisateurs dans un sous-répertoire de `/home`. Dans un environnement réel, il se pourrait que les utilisateurs soient regroupés suivant le type d'activités qu'ils effectuent. Il existerait alors autant de sous-répertoires dans `/home` que de groupes d'utilisateurs (cette notion n'a pas de rapport avec la notion UNIX de groupe).

L'option `-d` de la commande `adduser` permet, comme le montre l'exemple suivant, de spécifier l'emplacement et le nom du répertoire de travail de l'utilisateur.

```
[root@p200 /root]# adduser -d /home/staff/jchalle jchalle
```

Pour que cette commande aboutisse, il faut que le répertoire `/home/staff` soit créé.

3.3.3. Le shell utilisé.

Les systèmes UNIX sont généralement livrés avec plusieurs interpréteurs de commandes, chacun ayant ses caractéristiques propres. Selon l'utilisateur et le travail qui sera réalisé, un type d'interpréteur de commandes peut être choisi au moment de la création du compte. Les systèmes Linux disposent des interpréteurs suivants dans le répertoire `/bin` :

- `ash` ;
- `bash` ;
- `bsh` ;
- `csch` ;
- `ksh` ;
- `sh` ;
- `tcsh` ;
- `zsh` ;

L'option `-s` de la commande `adduser` permet à l'administrateur de spécifier le shell par défaut qui sera employé par l'utilisateur. Voici la syntaxe de la commande `adduser` permettant de configurer un compte pour qu'il utilise le korn shell :

```
[root@p200 /root]# adduser -s /bin/ksh jchalle
```

3.3.4. Les groupes.

Tous les utilisateurs font partie d'au moins un groupe. Par défaut, la commande `adduser` crée un groupe dont le nom sera le même que celui du login. Au moyen de l'option `-g`, il est possible de spécifier un autre groupe par défaut à la condition que ce groupe existe. Voici la commande permettant de créer un nouvel utilisateur dont le groupe par défaut sera celui du super-utilisateur :

```
[root@p200 /root]# adduser -g root jchalle
```

Pour affiner les autorisations d'accès d'un utilisateur, il est possible de le placer dans plusieurs groupes. Cette opération peut être effectuée au moment de la création du compte par l'intermédiaire de l'option `-G` de la commande `adduser`.

```
[root@p200 /root]# adduser -G adm,sys jchalle
```

Lorsque plusieurs utilisateurs forment un groupe de travail penché sur un même projet, il est intéressant qu'ils puissent se partager des informations. Pour cela, chaque utilisateur est placé dans un groupe qui lui est propre mais également dans un groupe où tous les acteurs du projet se trouvent. Une autre utilité de la commande consiste à donner, à certains utilisateurs, des droits correspondants à ceux du super-utilisateur. Dans ce cas, il suffit d'inclure également le nouvel utilisateur dans le groupe `root`.

3.3.5. Spécification de l'UID.

A chaque utilisateur du système est associé un entier unique appelé User IDentifier. Pour les utilisateurs privilégiés du système, cet UID a une valeur comprise entre 0 et 99.

Lorsque la commande `adduser` doit attribuer un UID par défaut, elle donne toujours une valeur supérieure à 99. De plus, si d'autres comptes ont été créés, la commande `adduser` choisit comme valeur d'UID, celle d'une unité supérieure à la plus grande valeur utilisée.

L'administrateur a la possibilité d'imposer une valeur d'UID au moment de la création d'un compte. Cette possibilité permet au gestionnaire du système d'attribuer, par exemple, un UID compris entre 1000 et 2000 aux utilisateurs du service commercial et un UID compris en 3000 et 4000 aux utilisateurs du service recherche et développement.

Voici un exemple de commande `adduser` attribuant l'UID 1000 au nouvel utilisateur créé :

```
[root@p200 /root]# adduser -u 1000 jchalle
```

3.3.6. Date d'expiration.

Dans certains cas, des comptes sont créés pour une durée limitée. C'est, par exemple, le cas lorsqu'une société reçoit un stagiaire. Après une certaine date, le compte ne doit plus être utilisé. L'option `-e` de la commande `adduser` permet de fixer la date à partir de laquelle le compte expire. Cette date peut être révisée à la hausse au moyen de l'option `-f`. L'option `-e` spécifie simplement une date d'expiration alors que l'option `-f` donne le nombre de jours de sursis accordé à l'utilisateur. Voici un exemple de commande créant un compte utilisateur dont l'expiration sera impérativement le 15 janvier 2005 :

```
[root@p200 /root]# adduser -e 2005-01-15 -f 0 jchalle
```

4. Modification d'un compte utilisateur.

4.1. La commande `usermod`.

La commande `usermod` permet de modifier toutes les options fixées par la commande `adduser`. Pour simplifier l'utilisation de cette commande, les options sont identiques à celles de la commande `adduser`. Voici un exemple d'utilisation de la commande `usermod` :

```
[root@p200 /root]# usermod -G sys,adm jchalle
```

Dans certains cas, l'administrateur peut être amené à retirer un utilisateur de tous les groupes dont il fait partie, sauf de son groupe individuel. Voici la syntaxe de la commande à employer :

```
[root@p200 /root]# usermod -G "" jchalle
```

Cela signifie que l'option `-G` remplace totalement les informations. Si l'utilisateur `jchalle` fait partie du groupe `adm` et que, par l'entremise de la commande `usermod`, il est placé dans le groupe `sys`, en réalité, il ne fera plus partie du groupe `adm` mais uniquement du groupe `sys`.

4.2. La commande chage.

La commande chage permet de gérer aisément les contraintes liées à la validité des comptes et des mots de passe. Au moyen de l'option -l, cette commande permet d'afficher les informations de validités associées à un compte :

```
[root@p200 /root]# chage -l jchalle
Minimum :          0
Maximum :          99999
Avertissement :    7
Désactivé :        -1
Dernier changement :      nov 15, 2001
Expiration du mot de passe :    Jamais
Password Inactive:      Jamais
Account Expires:       Jamais
```

La commande chage possède plusieurs options :

- -m : fixe la durée de vie minimale d'un mot de passe. Si une valeur nulle est spécifiée, le mot de passe peut être changé à tout moment ;
- -M : détermine la durée de vie maximale d'un mot de passe.
- -W : spécifie qu'un message d'avertissement doit être envoyé à l'utilisateur un certain nombre de jours avant l'expiration du mot de passe ;
- -d : fixe la date du dernier changement de mot de passe. La date doit être exprimée sous la forme du nombre de jours écoulés depuis le premier janvier 1970 ;
- -E : spécifie la date d'expiration du compte utilisateur. Lorsque cette date est atteinte, l'utilisateur devra contacter l'administrateur du système afin d'obtenir, de nouveau, un accès. Cette date doit être exprimée sous la forme du nombre de jours écoulés depuis le premier janvier 1970 ;
- -I : détermine le nombre de jours d'inactivité permis avant que le profil de l'utilisateur soit bloqué. Une valeur de zéro annihile cette fonctionnalité.

Toutes les dates sont mentionnées sous la forme du nombre de jours écoulés depuis le premier janvier 1970. Parmi les commandes du système, il en existe une retournant le nombre de secondes écoulées depuis le premier janvier 1970 à 1 heure du matin. Il faut donc écrire un script pour qu'il soit aisé de transmettre les dates à la commande chage. Voici un script recevant une date sous la forme JJ MM AAAA et renvoyant le nombre de jours écoulés depuis le premier janvier 1970 :

```
[root@p200 /root]# cat dt
#!/bin/bash
DATE=$2"/"$1"/"$3
ns=`convdate -n $DATE`
ns=`expr $ns + 3600`
nj=`expr $ns / 86400`
echo $nj
```

Voici un exemple d'utilisation de la commande chage :

```
[root@p200 /root]# ./dt 01 12 2010
14944
[root@p200 /root]# chage -m 0 -M 5 -W 3 -E 14944 -I 5 jchalle
```

Dans cet exemple, le mot de passe de l'utilisateur jchalle est limité à 5 jours. Dès que l'utilisateur aura conservé son mot de passe 2 jours (5-3), il sera averti qu'il doit changer son mot de passe. De plus, ce compte expirera le premier décembre 2010. Si le compte n'est pas utilisé durant une période de 5 jours, il sera bloqué.

Voici un exemple de message reçu par l'utilisateur, deux jours après l'exécution de la commande chage :

```
Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on an i586
login: jchalle
Password:
Warning: your password will expire in 5 days
[jchalle@p200 jchalle]$
```

4.3. La commande passwd.

La commande passwd permet de verrouiller et de déverrouiller un compte. L'option `-l` permet de verrouiller le compte alors que l'option `-u` permet de déverrouiller le compte. Voici un exemple d'utilisation de cette commande :

```
[root@p200 /root]# passwd -l jchalle
Changing password for user jchalle
Locking password for user jchalle
passwd: Success
[root@p200 /root]# passwd -u jchalle
Changing password for user jchalle
Unlocking password for user jchalle
passwd: Success
```

5. Suppression d'un compte utilisateur.

La commande `userdel` permet de supprimer un compte utilisateur. Sans argument, cette commande supprime les informations relatives au compte des fichiers `/etc/passwd`, `/etc/group` et `/etc/shadow`. Tous les fichiers de l'utilisateur sont toujours présents dans le répertoire de travail de l'utilisateur. Pour que la suppression du compte puisse être effectuée, le compte utilisateur doit exister et ne doit pas être en cours d'utilisation. Voici un exemple de suppression d'un compte utilisateur :

```
[root@p200 /root]# userdel jchalle
```

Pour que la suppression efface également tous les fichiers que possédait l'utilisateur, il suffit d'ajouter l'option `-r` à la commande `userdel`. Voici un exemple de suppression d'un compte utilisateur ainsi que de tous les fichiers qu'il possédait dans son répertoire de travail :

```
[root@p200 /root]# userdel -r jchalle
```

6. La gestion des groupes.

6.1. Création d'un groupe.

La commande `groupadd` permet d'ajouter de nouveaux groupes au système. La version simplifiée de la commande accepte uniquement le nom du groupe à créer. Dans cette situation, le numéro de groupe associé est toujours une unité supérieure à celle du plus grand numéro de groupe actuellement employé. Voici un exemple simple d'utilisation de la commande `groupadd` :

```
[root@p200 /root]# groupadd students
```

L'administrateur a la possibilité de fixer lui-même le GID qui sera attribué lors de la création d'un groupe. L'option `-g` de la commande `groupadd` doit être suivie du GID à attribuer au groupe à créer. Voici un exemple d'une telle commande :

```
[root@p200 /root]# groupadd -g 1000 students
```

6.2. Modification d'un groupe.

La commande `groupmod` modifie les attributs d'un groupe. L'option `-n` permet de changer le nom d'un groupe sans changer le GID associé. Voici un exemple de modification du nom d'un groupe :

```
[root@p200 /root]# groupmod -n jfchalle jchalle
```

Voici la visualisation de ce changement de nom :

```
[root@p200 /home]# ls -l
total 24
drwxr-xr-x  6 root    root    4096 oct 31 22:48 ftp
drwxr-xr-x  5 root    root    4096 oct 31 22:48 httpd
drwx----- 4 jchalle jfchalle 4096 nov 18 12:15 jchalle
drwx----- 6 jfc     jfc     4096 nov 15 21:04 jfc
drwx----- 4 rc      rc      4096 nov 15 20:45 rc
drwxr-xr-x  2 root    root    4096 nov 17 19:08 staff
```

6.3. Suppressions d'un groupe.

La suppression d'un groupe exige l'emploi de la commande `groupdel` suivie du nom du group à supprimer. Voici un exemple d'une telle commande :

```
[root@p200 /root]# groupdel students
```

7. Arrêt momentané des connexions.

Lorsque l'administrateur doit effectuer certains travaux de maintenance, il ne faut pas que les utilisateurs puissent se connecter au système. Lorsque le fichier `/etc/nologin` existe, seul le super-utilisateur du système a le droit de se connecter. Ce fichier contient généralement un message invitant les utilisateurs à patienter. Voici un exemple d'un tel fichier :

```
[root@p200 /etc]# cat nologin
Systeme indisponible jusqu'à 13:00 (maintenance)
```

Voici un exemple de session interactive lorsqu'un utilisateur tente de se connecter alors que le fichier `/etc/nologin` existe :

```
Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on an i586
login: jfc
Password:
Systeme indisponible jusqu'a 13:00 (maintenance)

Login incorrect
```

Cette méthode ne permet pas d'empêcher les utilisateurs de continuer à travailler mais interdit uniquement les nouvelles connexions.

8. Message du jour.

Lorsque l'administrateur doit communiquer des informations à l'ensemble de la communauté des utilisateurs, il peut le faire au moyen d'un courrier électronique qui sera envoyé à chacun d'entre eux. Ce procédé engendre de multiples copies du même message, ce qui occupe inutilement de la place sur le disque. Pour éviter cela, l'administrateur peut placer, dans le fichier `/etc/motd` (Message Of The Day), le message à délivrer à chaque utilisateur. Voici un exemple de ce fichier :

```
[root@p200 /etc]# cat motd
Un nouveau serveur DNS est maintenant installe.
```

Lorsqu'un utilisateur se connecte au système, le texte du message présent dans le fichier /etc/motd est affiché. Voici un exemple de session interactive lorsqu'un message du jour est configuré :

```
Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on an i586
login: jfc
Password:
Last login: Sun Nov 18 20:59:26 from P2366
Un nouveau serveur DNS est maintenant configuré.
[jfc@p200 jfc]$
```

9. Gestion des quotas.

9.1. Introduction.

Les quotas permettent à l'administrateur de limiter de deux manières, l'espace disque employé par les utilisateurs et par les groupes d'utilisateurs :

- limiter le nombre de fichiers (nombre d'i-nodes) ;
- limiter la place occupée (nombre de blocs de 1kb).

L'arborescence d'un système UNIX étant composé de plusieurs systèmes de fichiers, il est nécessaire de placer un dispositif de quotas sur tous ceux qui peuvent abriter des données en provenance des utilisateurs. Généralement, le répertoire /home contenant les comptes des utilisateurs réside dans un système de fichiers séparé de manière à mieux contrôler l'occupation du disque. Le mécanisme de quotas doit donc être mis en place sur ce système de fichiers.

Sous Linux, les quotas sont gérés selon trois paramètres :

- une limite soft ;
- une limite hard ;
- une période de grâce.

La limite hard est la limite absolue qu'un utilisateur ne peut en aucun cas dépasser. La limite soft peut être dépassée durant une période de temps stipulée par le délais de grâce. Durant cette période, des messages sont envoyés à l'utilisateur pour l'informer de ce qu'il a dépassé la limite autorisée.

9.2. Processus de mise en place des quotas.

Les quotas sont gérés par des fichiers de base de données présents dans la racine des systèmes de fichiers à contrôler. Le fichier quota.user contient les informations relatives aux limites des différents utilisateurs alors que le fichier quota.group contient les limites relatives aux différents groupes du système. Ces deux fichiers doivent être accessibles, en lecture et en écriture, uniquement par le super-utilisateur. Voici un exemple de création des fichiers de base de données de quotas :

```
[root@p200 /]# touch quota.user
[root@p200 /]# touch quota.group
[root@p200 /]# chmod 600 quota.user
[root@p200 /]# chmod 600 quota.group
```

Pour que les quotas soient pris en considération, il faut éditer le fichier /etc/fstab. Tout système de fichiers nécessitant une gestion des quotas soit au niveau des utilisateurs, soit au niveau des groupes ou bien à la fois pour les groupes et pour les utilisateurs doit comporter une indication dans le fichier /etc/fstab. La mention usrquota active les quotas au niveau des utilisateurs et grpquota active les quotas au niveau des groupes.

Voici un exemple d'activation des quotas pour le système de fichiers racine :

```
[root@p200 /etc]# cat fstab
/dev/hda5      /          ext2    defaults,usrquota,grpquota    1 1
/dev/hda1     /boot      ext2    defaults                       1 2
/dev/cdrom    /mnt/cdrom iso9660 noauto,owner,ro              0 0
/dev/fd0      /mnt/floppy auto     noauto,owner                  0 0
none         /proc      proc    defaults                       0 0
none         /dev/pts   devpts  gid=5,mode=620                0 0
/dev/hda6     swap       swap    defaults                       0 0
```

Chaque ligne du fichier `/etc/fstab` décrivant un système de fichiers est constituée de six champs :

- le premier champ est le périphérique bloc à monter ;
- le second champ indique le point de montage du système de fichiers ;
- le troisième champ décrit le type de système de fichiers ;
- le quatrième champ stipule les options de montage du système de fichiers. C'est dans ce champ que l'activation des quotas sera mentionnée ;
- le cinquième champ précise les systèmes de fichiers qui doivent être sauvegardés. Une valeur zéro indique que le système de fichiers ne doit pas être sauvegardé ;
- le sixième champ mentionne l'ordre de vérification des systèmes de fichiers lors du démarrage du système.

Le fichier `/etc/fstab` n'étant lu qu'au démarrage du système, il faut rebooter la machine pour que les modifications rendent la gestion des quotas active.

Au départ, les bases de données `quota.user` et `quota.group` ne sont pas initialisées. Dans un premier temps, il est nécessaire de construire ces bases de données sans imposer de limites. La commande `quotacheck` permet d'effectuer cette opération comme le montre la session interactive suivante :

```
[root@p200 /]# quotacheck -avug
Scanning /dev/hda5 [/] done
Checked 4393 directories and 85060 files
Using quotafile /quota.user
Using quotafile /quota.group
```

La commande `quotacheck` doit construire une base de données pour les utilisateurs (option `-u`) ainsi que pour les groupes (option `-g`). Cette construction est basée sur les informations présentes dans le fichier `/etc/fstab` (option `-a`). Généralement l'administrateur souhaite un rapport d'activité de la commande `quotacheck` (option `-v`).

Bien que les bases de données soient créées, l'administrateur doit enclencher la surveillance des quotas au moyen de la commande `quotaon`. L'arrêt de la surveillance peut être réalisée au moyen de la commande `quotaoff`. Voici un exemple de ces deux commandes :

```
[root@p200 /etc]# quotaon /
[root@p200 /etc]# quotaoff /
```

9.3. Attribution des quotas.

9.3.1. Quotas d'un utilisateur.

La commande `edquota` permet d'éditer les quotas relatifs à un utilisateur. Voici un exemple d'édition des quotas :

```
[root@p200 /etc]# edquota jchalle
Quotas for user jchalle:
/dev/hda5: blocks in use: 184, limits (soft = 190, hard = 200)
          inodes in use: 46, limits (soft = 50, hard = 60)
```

L'administrateur peut modifier les limites placées entre parenthèses. L'absence de quotas est indiquée par des limites nulles.

Dès que l'une des limites soft est atteinte, l'utilisateur voit sa commande échouer et il en est averti par un message. Voici un exemple de session interactive montrant l'échec d'opérations en raison de la limite des quotas :

```
[jchalle@p200 jchalle]$ man ls > m
/: write failed, user disk limit reached.
[jchalle@p200 jchalle]$ touch fichier
/: warning, user file quota exceeded
```

9.3.2. Quotas d'un groupe.

L'option `-g` de la commande `edquota` permet d'éditer les quotas relatifs à un groupe. Voici un exemple d'édition des quotas pour un groupe :

```
[root@p200 /etc]# edquota -g jchalle
Quotas for group jchalle:
/dev/hda5: blocks in use: 148, limits (soft = 0, hard = 0)
          inodes in use: 49, limits (soft = 0, hard = 0)
```

Dès qu'un des membres du groupe dépasse une limite soft, tous les membres du groupe sont dans l'incapacité de continuer leur travail sans avoir au préalable effacé des fichiers afin de répondre aux critères mis en œuvre par le système des quotas.

9.3.3. Période de grâce.

La période de grâce peut être fixée au moyen de l'option `-t` de la commande `edquota`. Voici un exemple fixant les délais à 7 jours :

```
[root@p200 /etc]# edquota -t
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/hda5: block grace period: 7 days, file grace period: 7 days
```

9.3.4. Attribution des quotas à plusieurs utilisateurs.

La gestion des quotas peut être très lourde lorsqu'il faut éditer les limites relatives à chaque utilisateur du système. Dès que les limites sont fixées pour un utilisateur, il est possible de les recopier pour d'autres. L'option `-p` de la commande `edquota` permet de prendre un utilisateur comme base pour fixer les quotas de plusieurs autres. Voici un exemple d'une telle commande :

```
[root@p200 /etc]# edquota -p jchalle jfc jl rc
```

Dans cet exemple, les utilisateurs `jfc`, `jl` et `rc` reçoivent les mêmes limites que l'utilisateur `jchalle`.

9.4. Consultation des quotas.

La commande `repquota` établit un rapport sur l'utilisation des quotas. Voici un exemple de consultation des limites associées aux utilisateurs (option `-u`):

```
[root@p200 /etc]# repquota -u /
                                Block limits
User      used  soft  hard  grace  used  soft  hard  grace
root      -- 1458472  0    0      88539  0    0
daemon   --    8    0    0      3     0    0
news     --  5768  0    0     236   0    0
uucp     -- 1048   0    0     16    0    0
games    --   36   0    0     41    0    0
squid     --    8   0    0      2     0    0
postgres --  6572  0    0     458   0    0
gdm      --    4   0    0      1     0    0
xfs      --    8   0    0      3     0    0
piranha  --    0   0    0      1     0    0
nobody   --    8   0    0      2     0    0
jfc      ++  248  190  200   none   60   50   60   none
rc       --  180  190  200      45   50   60
jchalle  +-  172  190  200     51   50   60  7days
```

Les mêmes vérifications peuvent avoir lieu pour les groupes. Pour cela, il suffit de mentionner l'option `-g`. Voici un exemple de vérification des quotas relatifs aux groupes :

```
[root@p200 /etc]# repquota -g /
                                Block limits
User      used  soft  hard  grace  used  soft  hard  grace
root      -- 1445028  0    0     81528  0    0
bin       --   52   0    0      3     0    0
daemon   --   88   0    0     21    0    0
sys       --    0   0    0     283   0    0
tty       --  932   0    0     655   0    0
disk     --    0   0    0    3753   0    0
lp       --   88   0    0      4     0    0
kmem     --  108   0    0      5     0    0
mail     --  2064  0    0     10    0    0
news     --  5944  0    0     239   0    0
uucp     -- 1244   0    0     461   0    0
man      --  7696  0    0    1791   0    0
floppy   --    0   0    0     20    0    0
games    --   832  0    0     56    0    0
slocate  --   888  0    0      5     0    0
utmp     --   424  0    0      4     0    0
squid     --    8   0    0      2     0    0
postgres --  6572  0    0     458   0    0
gdm      --    4   0    0      1     0    0
xfs      --    8   0    0      3     0    0
ftp      --    4   0    0      1     0    0
nobody   --    4   0    0      2     0    0
jfc      --  236   0    0     56    0    0
rc       --  180   0    0     45    0    0
jchalle  --  148   0    0     49    0    0
```


II. La séquence de démarrage du système.

1. Principe de la séquence de démarrage.

Dès qu'un PC est mis sous tension, le BIOS effectue différents tests pour vérifier que tous les composants sont à même de fonctionner. Cette phase est appelée POST (Power On Self Test). Dès que cette phase de vérification est terminée, le véritable démarrage du système d'exploitation débute par le chargement du secteur de démarrage. Ce secteur de démarrage appelé master boot record est le premier secteur du disque.

Le secteur de boot contient un petit programme dont le rôle consiste à continuer la séquence de lancement du système. Le code du master boot record examine la table des partitions afin d'identifier la partition active. La partition bootable doit contenir dans son secteur de boot un code permettant le lancement du système d'exploitation. Ce secteur appelé le master boot record de la partition est chargé en mémoire et ensuite est exécuté. L'objectif du programme, présent dans le secteur de boot de la partition, réside dans le but de charger entièrement le système d'exploitation en mémoire.

La majorité des distributions de Linux installent un chargeur appelé LILO (Linux LOader) soit dans le MBR du disque, soit dans le MBR de la partition active. Ce chargeur permet à l'utilisateur de choisir le noyau du système d'exploitation qu'il souhaite lancer. Notons que LILO est également capable de lancer d'autres systèmes d'exploitation que Linux.

Lorsque le choix du noyau est effectué, LILO le place en mémoire. L'image du noyau placée en mémoire est une archive compressée contenant en son début le code d'extraction. Consécutivement à cette opération, le noyau vérifie la présence de périphériques (disques durs, lecteurs de disquettes, cartes réseau, ...) et les configure de manière appropriée. Durant cette phase, le noyau affiche des messages sur les périphériques qu'il trouve.

Voici un exemple des messages générés par le noyau au moment du démarrage du système :

```
LILO boot :
Loading linux .....
Linux version 2.2.14-5.0 (root@porkey.devel.redhat.com) (gcc version egcs-2.91.66
 19990314/Linux (egcs-1.1.2 release)) #1 Tue Mar 7 20:53:41 EST 2000
Detected 200456694 Hz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 399.77 BogoMIPS
Memory: 95452k/98304k available (1084k kernel code, 412k reserved, 1292k data, 6
4k init, 0k bigmem)
Dentry hash table entries: 262144 (order 9, 2048k)
Buffer cache hash table entries: 131072 (order 7, 512k)
Page cache hash table entries: 32768 (order 5, 128k)
VFS: Diskquotas version dquot_6.4.0 initialized
CPU: Intel Pentium MMX stepping 03
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
Intel Pentium with F0 0F bug - workaround enabled.
POSIX conformance testing by UNIFIX
PCI: PCI BIOS revision 2.10 entry at 0xf0510
PCI: Using configuration type 1
PCI: Probing PCI hardware
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
TCP: Hash tables configured (ehash 131072 bhash 65536)
Initializing RT netlink socket
Starting kswapd v 1.5
Detected PS/2 Mouse Port.
Serial driver version 4.27 with MANY_PORTS MULTIPORT SHARE_IRQ enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
pty: 256 Unix98 ptys configured
```

```

apm: BIOS version 1.2 Flags 0x0b (Driver version 1.9)
Real Time Clock Driver v1.09
RAM disk driver initialized: 16 RAM disks of 4096K size
PIIX4: IDE controller on PCI bus 00 dev 09
PIIX4: not 100% native mode: will probe irqs later
    ide0: BM-DMA at 0xe000-0xe007, BIOS settings: hda:DMA, hdb:DMA
    ide1: BM-DMA at 0xe008-0xe00f, BIOS settings: hdc:pio, hdd:pio
hda: WDC AC38400L, ATA DISK drive
hdb: ATAPI CDROM 52X, ATAPI CDROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: WDC AC38400L, 8063MB w/256kB Cache, CHS=1027/255/63
hdb: ATAPI 52X CD-ROM drive, 128kB Cache
Uniform CDROM driver Revision: 2.56
Floppy drive(s): fd0 is 1.44M
FDC 0 is a post-1991 82077
md driver 0.90.0 MAX_MD_DEVS=256, MAX_REAL=12
raid5: measuring checksumming speed
raid5: MMX detected, trying high-speed MMX checksum routines
    pII_mmx   :    323.469 MB/sec
    p5_mmx   :    374.523 MB/sec
    8regs    :    216.408 MB/sec
    32regs   :    160.401 MB/sec
using fastest function: p5_mmx (374.523 MB/sec)
scsi : 0 hosts.
scsi : detected total.
md.c: sizeof(mdp_super_t) = 4096
Partition check:
    hda: hda1 hda2 < hda5 hda6 >
autodetecting RAID arrays
autorun ...
... autorun DONE.
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 64k freed
Adding Swap: 136512k swap-space (priority -1)

```

Le noyau essaie ensuite de monter le système de fichiers racine. Si cette opération échoue, le noyau panique et arrête le système. Le système de fichiers racine est normalement monté en lecture seule afin de permettre une vérification de son intégrité.

Le noyau lance ensuite le programme `/sbin/init` en tâche de fond. Ce processus est l'ancêtre de tous les processus du système. C'est à partir de ce programme que toutes les fonctionnalités du système sont démarrées. Lorsque le processus `init` arrive à la fin de l'initialisation de ce système, il lance le programme `getty` pour toutes les consoles afin d'autoriser les utilisateurs à se connecter.

2. LILO.

2.1. Les arguments du démarrage.

Le chargeur Linux est le programme responsable du chargement du noyau Linux une fois que le BIOS a passé le contrôle de l'ordinateur au système d'exploitation. Le chargeur de Linux se manifeste par l'affichage de l'information suivante :

```
LILLO boot:
```

Cette ligne est affichée par le programme LILO avant que le noyau LINUX n'ait démarré. Cette ligne de sortie est l'invite d'amorçage qui permet à l'utilisateur de choisir une image du noyau, à démarrer ou à passer des arguments au noyau.

La capacité de passer des arguments au noyau est utile. Par exemple, en cas de modification non adéquate des bibliothèques du système ou de la perte de données à la suite d'un problème d'alimentation, l'utilisateur peut passer des

arguments au noyau de manière à amorcer le système en mode maintenance. Ce mode de fonctionnement permet à l'administrateur de travailler en mode mono-utilisateur sans être perturbé par les autres utilisateurs. L'activation de ce mode est effectué en transmettant l'argument `single` au noyau. Voici un exemple de démarrage du système en mode maintenance :

```
LILO boot: linux single
```

Le passage d'arguments au noyau peut constituer une faille de sécurité. Le plus dangereux est l'argument `init=` qui permet à l'utilisateur, lors de l'invite de LILO, de spécifier un programme `init` alternatif. Le programme `init` est responsable du démarrage de tous les services, incluant la vérification du nom de compte et du mot de passe. L'argument `init=` permet à l'utilisateur de préciser par exemple l'exécution du shell `/bin/bash` en lieu en place du programme `init` traditionnel. Voici un exemple d'une telle saisie :

```
LILO boot: linux init=/bin/bash
```

Cette simple ligne peut compromettre l'entièreté du système. Lorsque l'argument `init` pointe sur `/bin/bash`, le noyau Linux démarre comme premier processus `/bin/bash`. Cela signifie qu'aucun service ne sera démarré, qu'aucune demande d'authentification ne sera requise pour utiliser le système. De plus, le shell est lancé sous l'identité du super-utilisateur, ce qui permet aux personnes mal intentionnées d'engendrer les pires dégâts.

Des mesures doivent donc être prises pour sécuriser le chargeur LILO.

2.2. Le fichier /etc/lilo.conf.

Le comportement du chargeur est dirigé par le contenu du fichier `/etc/lilo.conf`. Ce fichier est scindé en deux sections principales. La première section contient des paramètres généraux qui s'appliquent à toutes les autres sections. La seconde section du fichier débute par le mot `image=` ou `other=` et s'étend jusqu'à la fin du fichier. Plusieurs images du noyau de Linux peuvent être définies au moyen de plusieurs entrées `image=`. Voici un exemple de fichier `/etc/lilo.conf` :

```
[root@P2450 /etc]# cat lilo.conf
boot=/dev/sda5
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.14-5.0smp
    label=linux
    initrd=/boot/initrd-2.2.14-5.0smp.img
    read-only
    root=/dev/md0

image=/boot/vmlinuz-2.2.14-5.0
    label=linux-up
    initrd=/boot/initrd-2.2.14-5.0.img
    read-only
    root=/dev/md0
```

Ce fichier d'exemple montre deux configurations de démarrage du système. La première permet de charger un noyau pour un système multiprocesseurs et la seconde pour un noyau monoprocesseur.

2.3. Le mot clé password.

La sécurisation du processus de démarrage du système consiste à imposer un mot de passe. Le mot clé `password` permet au système de demander que l'utilisateur entre un mot de passe avant l'amorçage de l'image d'un noyau. Le mot clé `password` peut être placé soit dans la première section, soit dans une partie `image=`.

Si le mot clé password est placé dans la section de configuration globale, toutes tentatives d'amorçage exigeront que l'utilisateur fournisse un mot de passe. Si le mot clé password est placé dans une section image=, seul le lancement de cette image requerra un mot de passe. Voici un exemple de protection globale :

```
[root@P2450 /etc]# cat lilo.conf
boot=/dev/sda5
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
password= "qwerty"
image=/boot/vmlinuz-2.2.14-5.0smp
    label=linux
    initrd=/boot/initrd-2.2.14-5.0smp.img
    read-only
    root=/dev/md0

image=/boot/vmlinuz-2.2.14-5.0
    label=linux-up
    initrd=/boot/initrd-2.2.14-5.0.img
    read-only
    root=/dev/md0
```

Dès que le fichier /etc/lilo.conf est modifié, il faut que le secteur de démarrage reflète cette modification. La commande lilo permet d'écrire un nouveau secteur de démarrage en fonction de la configuration présente dans le fichier /etc/lilo.conf. Voici un exemple d'exécution de cette commande :

```
[root@p2450 /etc]# lilo
Warning: /etc/lilo.conf should be readable only for root if using PASSWORD
Added linux *
```

Dans le cas d'une protection par un mot de passe, le système indique que le fichier /etc/lilo.conf ne puisse être lisible que par le super-utilisateur. Sans cela, le mot de passe ne serait qu'un secret de polichinelle.

Cette méthode de protection exige la présence de l'administrateur lors d'un démarrage du système. En réalité, l'utilité du mot de passe n'a de sens qu'à une condition : que des paramètres soient transmis au travers de l'invite LILO. Le mot clé restricted force ne LILO à demander un mot de passe qu'uniquement dans le cas où des paramètres sont transmis au démarrage. Cela signifie qu'un démarrage normal ne nécessitera pas l'entrée d'un mot de passe.

```
[root@P2450 /etc]# cat lilo.conf
boot=/dev/sda5
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
password= "qwerty"
restricted
image=/boot/vmlinuz-2.2.14-5.0smp
    label=linux
    initrd=/boot/initrd-2.2.14-5.0smp.img
    read-only
    root=/dev/md0

image=/boot/vmlinuz-2.2.14-5.0
    label=linux-up
    initrd=/boot/initrd-2.2.14-5.0.img
    read-only
    root=/dev/md0
```

3. Le processus init.

Lorsque le noyau a terminé son initialisation, il lance le processus init. Ce processus est chargé de lancer tous les services du système par l'intermédiaire de scripts, dont le principal est /etc/rc.d/rc.

Le comportement du processus init est basé sur le contenu du fichier /etc/inittab. Ce fichier définit la notion de run level. L'idée de ces run levels est que l'administrateur système peut souhaiter booter le système en mode mono utilisateur, en mode multi-utilisateurs ou bien encore provoquer le démarrage de X window.

Tout système UNIX dispose de sept run levels numérotés de 0 à 6. Voici la signification des ces différents niveaux d'exécution :

Run level	Signification
0	Arrêt du système (/etc/rc.d/rc0.d)
1	démarrage du système en mode mono utilisateur (/etc/rc.d/rc1.d)
2	démarrage du système en mode multi-utilisateurs mais sans le support NFS (/etc/rc.d/rc2.d)
3	démarrage du système en mode multi-utilisateurs (/etc/rc.d/rc3.d)
4	non utilisé (/etc/rc.d/rc4.d)
5	démarrage du système en mode multi-utilisateurs mais en lançant X window (/etc/rc.d/rc5.d)
6	redémarrer le système (/etc/rc.d/rc6.d)

Cette convention de numérotation des niveaux d'exécution est parfaitement arbitraire. Rien n'empêche une distribution UNIX d'attacher d'autres significations à ces niveaux d'exécution. La description du travail qui doit être réalisé en fonction du niveau d'exécution est décrite dans le fichier /etc/inittab. Voici un exemple de fichier /etc/inittab :

```
[root@p200 /etc]# cat inittab
id:3:initdefault:

si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6

ud::once:/sbin/update
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

x:5:respawn:/etc/X11/prefdm -nodaemon
```

Chaque ligne de ce fichier est de la forme :

id:runlevel:action:process

Voici la signification des champs de cette ligne :

- id est une séquence d'un maximum de quatre caractères permettant d'identifier la ligne ;
- runlevel est le numéro de niveau d'exécution de cette ligne ;
- action détermine comment le processus doit être lancé ainsi que l'action à prendre lorsqu'il se termine ;
- process spécifie le processus qui doit être exécuté.

Les actions possibles correspondent à des mots clés. Voici la signification de quelques mots clés :

- respawn : dès que le processus est terminé, il est relancé automatiquement. Ce comportement est particulièrement utile pour la gestion des terminaux. Dès qu'un utilisateur clôture une session interactive, cela met un terme au processus de gestion du terminal. Pour qu'une autre personne puisse travailler sur ce terminal il faut que le processus de gestion soit relancé automatiquement ;
- wait : le processus correspondant est lancé une seule fois. Avant de continuer l'initialisation du système, init doit attendre la fin de l'exécution du processus qui vient d'être lancé ;
- once : le processus correspondant est exécuté une seule fois ;
- initdefault : cette entrée détermine le niveau d'exécution par défaut employé par init lors d'une phase de boot normale ;
- sysinit : le processus correspondant sera exécuté durant la séquence de démarrage quel que soit le niveau d'exécution par défaut. En d'autres termes, le champ runlevels est ignoré ;
- powerfail : le processus correspondant est exécuté dès que l'alimentation électrique n'est plus assurée. Pour que cette opération soit possible, il faut que le système informatique soit connecté à un UPS ;
- powerokwait : le processus correspondant est exécuté dès que l'alimentation électrique est restaurée ;
- ctrlaltdel : le processus correspondant est exécuté dès que la combinaison de touches ctrl-alt-del est enfoncée.

A chaque run level correspond l'exécution d'un ensemble de scripts placés dans le répertoire `/etc/rc.d/rc#.d` où le caractère # doit être remplacé par le numéro du niveau d'exécution correspondant.

4. Les scripts de démarrage.

Le répertoire `/etc/rc.d/init.d` contient tous les scripts de démarrage des services du système. Les répertoires `/etc/rc.d/rc#.d` contiennent simplement des liens vers les scripts du répertoire `/etc/rc.d/init.d`. Ces répertoires de liens contiennent des noms de fichiers débutant soit par la lettre S ou bien par la lettre K. En seconde position est codifié un nombre de deux chiffres. Les fichiers débutant par la lettre S lancent les services du système alors que les fichiers débutant par la lettre K les arrêtent. L'ordre d'exécution des scripts dans un répertoire `/etc/rc.d/rc#.d` est déterminé par les deux chiffres suivant la lettre S ou K. Les scripts affectés d'un numéro faible sont exécutés avant ceux portant un numéro plus élevé.

Le répertoire `/etc/rc.d` contient un fichier appelé `rc.local`. Ce fichier est un script dont l'exécution est déclenchée par init lorsque tous les autres scripts d'initialisation du système ont été exécutés.

Le répertoire `/etc/rc.d` contient également un fichier appelé `rc.sysinit`. Ce script est lancé par le processus init avant tout autre script d'initialisation.

5. L'arrêt du système.

L'arrêt d'un système UNIX doit s'effectuer suivant une procédure bien établie. Il est hors de question de couper l'alimentation électrique à n'importe quel moment. Si tel est le cas, les systèmes de fichiers ainsi que les fichiers risquent de se trouver en état d'incohérence. Ce risque est bien réel en raison des buffers employés par le système pour optimiser les accès aux fichiers.

Une autre raison proscrivant l'extinction brutale de la machine est que, dans un système multi-tâches et multi-utilisateurs, il peut y avoir de nombreux travaux exécutés en quasi parallélisme. L'utilisation d'une procédure correcte d'extinction permet de terminer proprement tous les processus.

La commande permettant d'arrêter proprement un système UNIX s'appelle shutdown. Elle est généralement utilisée de deux façons possibles :

- Si le système est employé uniquement par une seule personne, la façon classique d'utiliser la commande shutdown consiste à quitter tous les programmes qui s'exécutent, à se déconnecter de toutes les consoles, à se connecter sous le compte root et à dactylographier la commande suivante :

```
[root@p200 /root]# shutdown -h now
```

L'option `-h` précise que le système doit simplement être arrêté.

- Si le système est employé par plusieurs utilisateurs, il faut leur laisser le temps de terminer les travaux en cours. L'administrateur doit donc lancer la commande en demandant que la procédure d'arrêt soit exécutée de manière différée.

```
[root@p200 /root]# shutdown -h 10 'maintenance'
```

Tous les utilisateurs reçoivent alors le message suivant :

```
Broadcast message from root (pts/0) Mon Nov 19 19:55:02 2001...
maintenance
The system is going DOWN for system halt in 10 minutes !!
```

Lorsque le procédure d'arrêt se lance après le délai précisé, tous les systèmes de fichiers, sauf la racine, sont démontés, les processus utilisateur sont tués, les démons sont arrêtés et enfin le système de fichiers racine est démonté. Lorsque toutes les opérations sont terminées, le processus `init` affiche un message invitant l'administrateur à couper l'alimentation électrique.

6. Le redémarrage du système.

Ceci peut être réalisé en arrêtant d'abord complètement le système en éteignant puis en rallumant la machine. Une façon plus simple consiste à demander à la commande `shutdown` de relancer le système au lieu de simplement l'arrêter. Ceci est réalisé au moyen de l'option `-r` de la commande `shutdown`.

La plupart des systèmes Linux lancent la commande `shutdown -r now` lorsque la combinaison de touche `ctrl-alt-del` est tapée au clavier de la console. Cette possibilité peut constituer une faille de sécurité dans le système. Pour inhiber cette fonctionnalité, il suffit d'éditer le fichier `/etc/inittab` et de supprimer la ligne suivante :

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```


III. Configuration de l'interface réseau.

1. La commande ifconfig.

La commande ifconfig fixe et vérifie la configuration des interfaces réseau. Cette commande affectera les arguments suivants à chacune des interfaces réseau :

- l'adresse IP ;
- le masque de sous-réseau ;
- l'adresse broadcast.

Voici la commande ifconfig employée lors de la configuration d'une interface Ethernet d'un pc fonctionnant sous Linux :

```
ifconfig eth0 212.68.194.203 netmask 255.255.255.240 broadcast 212.68.194.207
```

Un grand nombre d'autres arguments peuvent être utilisés avec cette commande. Toutefois, voici ceux permettant de spécifier les informations de base de TCP/IP pour une interface réseau :

- interface. C'est le nom de l'interface réseau à configurer. L'exemple ci-dessus configure l'interface eth0.
- adresse. Il est permis de donner soit l'adresse IP soit le nom symbolique de la machine. Notre exemple affecte l'adresse 212.68.194.203 à l'interface eth0. Si l'interface réseau est configurée au moyen d'un nom symbolique, il faut que la résolution de l'adresse (transformation du nom symbolique en une adresse IP) soit effectuée au moyen du fichier /etc/hosts. Le serveur DNS étant lancé après la configuration des interfaces réseau, il faut ajouter une entrée dans le fichier /etc/hosts.
- netmask. C'est le masque de sous-réseau pour cette interface. Cet argument peut être omis uniquement dans le cas où le masque est dérivé de la structure de classe traditionnelle. Dans notre exemple, nous avons découpé une adresse de classe C en 16 sous-réseaux (adresse réseau sur 28 bits). Le masque doit, dans cet exemple, posséder ses 28 bits de poids fort à 1. Le masque est donc 255.255.255.240.
- broadcast. C'est l'adresse de diffusion du réseau. La plupart des machines utilisent comme adresse par défaut l'adresse broadcast standard qui correspond à une adresse IP dont tous les bits désignant la machine sont positionnés à 1. L'interface eth0 a donc une adresse broadcast qui est 212.68.194.207. L'adresse broadcast peut également être spécifiée de manière symbolique. Dans ce cas, l'administrateur placera la valeur soit dans le fichier /etc/hosts, soit dans le fichier /etc/networks.

2. Structure des fichiers /etc/hosts et /etc/networks.

Les fichiers /etc/hosts et /etc/networks sont au format texte. Chaque ligne de ces fichiers associe une adresse IP à un ou plusieurs noms symboliques.

Voici un exemple de fichier /etc/hosts :

```
127.0.0.1      localhost localhost.localdomain
212.68.194.203 gateway
```

Voici un exemple de fichier /etc/networks :

```
212.68.194.207 bcasteth0
```

En se basant sur le contenu des fichiers décrits ci-dessus, la commande de configuration de l'interface eth0 peut se rédiger de la manière suivante :

```
ifconfig eth0 gateway netmask 255.255.255.240 broadcast bcasteth0
```

3. Détermination du nom de l'interface.

Un ordinateur peut disposer de plusieurs cartes réseau auxquelles des noms sont associés. En principe, il est possible, de déterminer le nom des interfaces disponibles à partir des messages affichés à la console lors du démarrage du système. Sur bon nombre de systèmes, ces informations peuvent être consultées en utilisant la commande `dmesg`.

L'exemple qui suit montre les messages correspondants à une machine équipée de 6 cartes Ethernet :

```
gateway:~# dmesg | grep "eth[0-9]:*"
eth0: 3Com 3c905B Cyclone 100baseTx at 0x6100, 00:50:04:32:a7:e7, IRQ 11
eth1: 3Com 3c905B Cyclone 100baseTx at 0x6200, 00:50:04:32:98:1b, IRQ 12
eth2: 3Com 3c905B Cyclone 100baseTx at 0x6300, 00:50:04:32:9b:45, IRQ 5
eth3: 3Com 3c905B Cyclone 100baseTx at 0x6400, 00:50:04:32:9b:6b, IRQ 5
eth4: WD80x3 at 0x240, 00 00 C0 08 70 29 WD8013, IRQ 10, shared memory at
0xc8000-0xcbfff.
eth5: NE2000 found at 0x340, using IRQ 9.
```

Il n'est pas toujours aisé de déterminer toutes les interfaces disponibles sur un système en se basant uniquement sur les messages fournis par la commande `dmesg`. Puisqu'elle donne seulement les informations relatives aux interfaces physiques. Un système informatique pouvant être équipé d'interfaces virtuelles, il faut employer les commandes `netstat` et `ifconfig` pour obtenir la liste de toutes les interfaces disponibles. Quant à elle la commande `netstat` permet de visualiser toutes les interfaces configurées.

L'option `-i` de la commande `netstat` permet de visualiser les statistiques relatives aux cartes réseau installées. Néanmoins, si une interface n'est pas active, aucune information n'est fournie par `netstat`. La liste de toutes les interfaces est obtenue par le biais de l'option `-a`. Cette option demande l'affichage des informations concernant toutes les interfaces qu'elles soient actives ou inactives.

Voici un exemple d'utilisation de la commande `netstat` :

```
gateway:~# netstat -ia
Kernel Interface table
Iface  MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500  0  1107745      0      0      0  1039503      0      0      0 BRU
eth1   1500  0   126495      1      0      0   205668      0      0      0 BRU
eth2   1500  0    72856     14      0      0    35432      0      0      0 BRU
eth3   1500  0         0      0      0      0         6      0      0      0 BRU
eth4   1500  0    18739      0      0      0         6      0      0      0 BRU
eth5   1500  0   972754      0      0      0  1019355      0      0      0 BRU
eth5:  1500  0      - no statistics available -      BRU
eth5:  1500  0      - no statistics available -      BRU
eth5:  1500  0      - no statistics available -      BRU
lo     3924  0    65491      0      0      0    65491      0      0      0 LRU
```

Voici la signification des champs affichés par `netstat` :

- **MTU** : cette valeur correspond à la taille en octets de la plus grande trame (paquet) pouvant être transmise par l'interface sans engendrer de fragmentation.
- **Met** : cette valeur numérique fixe le coût pour emprunter une route. Plus cette valeur est faible plus le coût est faible.
- **RX-OK** : informe du nombre de trames reçues sans erreur.
- **RX-ERR** : informe du nombre de trames endommagées qui ont été reçues.
- **RX-DRP** : informe du nombre de trames reçues qui ont été éliminées (mémoire insuffisante).
- **RX-OVR** : informe du nombre de trames qui ont été perdues (arrivée trop rapide des données).
- **TX-OK** : informe du nombre de trames envoyées sans erreur.
- **TX-ERR** : informe du nombre de trames envoyées qui ont été endommagées.
- **TX-DRP** : informe du nombre de trames reçues qui ont été éliminées.
- **TX-OVR** : informe du nombre de trames qui ont été perdues (émission trop rapide des données).
- **Flg** : donne des informations sur l'interface comme le montre le tableau suivant :

flag	signification
B	Une adresse broadcast est attribuée à l'interface
L	L'interface est de type loopback
M	L'interface est configurée en mode promiscuous
O	Le protocole ARP n'est pas utilisé avec cette interface
P	C'est une connexion point à point
R	L'interface est opérationnelle (running)
U	L'interface est en cours d'utilisation (UP)

La commande `ifconfig` peut également être employée pour déterminer les interfaces réseau disponibles. Voici le résultat d'exécution de la commande `ifconfig` sur une machine équipée de 6 cartes Ethernet.

```
gateway:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:04:32:A7:E7
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3066319  errors:0  dropped:0  overruns:0  frame:0
          TX packets:2862439  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x6100

eth1      Link encap:Ethernet  HWaddr 00:50:04:32:98:1B
          inet addr:192.168.23.1  Bcast:192.168.23.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1163838  errors:14  dropped:0  overruns:0  frame:18
          TX packets:1409032  errors:0  dropped:0  overruns:0  carrier:0
          collisions:6765 txqueuelen:100
          Interrupt:12 Base address:0x6200

eth2      Link encap:Ethernet  HWaddr 00:50:04:32:9B:45
          inet addr:192.168.24.1  Bcast:192.168.24.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:806344  errors:1480  dropped:0  overruns:0  frame:2245
          TX packets:814853  errors:0  dropped:0  overruns:0  carrier:0
          collisions:4659 txqueuelen:100
          Interrupt:5 Base address:0x6300

eth3      Link encap:Ethernet  HWaddr 00:50:04:32:9B:6B
          inet addr:192.168.25.1  Bcast:192.168.25.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:14  errors:0  dropped:0  overruns:0  carrier:14
          collisions:0 txqueuelen:100
          Interrupt:5 Base address:0x6400

eth4      Link encap:Ethernet  HWaddr 00:00:C0:08:70:29
          inet addr:192.168.22.1  Bcast:192.168.22.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39374  errors:0  dropped:0  overruns:0  frame:19
          TX packets:17  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0x250 Memory:c8000-cc000

eth5      Link encap:Ethernet  HWaddr 00:00:E8:3A:1B:2E
          inet addr:212.68.194.203  Bcast:212.68.194.207  Mask:255.255.255.240
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1333799  errors:0  dropped:0  overruns:0  frame:0
          TX packets:1350119  errors:0  dropped:0  overruns:0  carrier:0
          collisions:233 txqueuelen:100
          Interrupt:9 Base address:0x340
```

```

eth5:0   Link encap:Ethernet  HWaddr 00:00:E8:3A:1B:2E
         inet addr:212.68.194.202  Bcast:212.68.194.207  Mask:255.255.255.240
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:9 Base address:0x340

eth5:1   Link encap:Ethernet  HWaddr 00:00:E8:3A:1B:2E
         inet addr:212.68.194.201  Bcast:212.68.194.207  Mask:255.255.255.240
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:9 Base address:0x340

eth5:2   Link encap:Ethernet  HWaddr 00:00:E8:3A:1B:2E
         inet addr:212.68.194.200  Bcast:212.68.194.207  Mask:255.255.255.240
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:9 Base address:0x340

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING  MTU:3924  Metric:1
         RX packets:125993 errors:0 dropped:0 overruns:0 frame:0
         TX packets:125993 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0

```

4. Vérification de l'interface avec ifconfig.

Lors de l'installation d'un système UNIX, des valeurs sont attribuées aux interfaces réseau. Toutefois, la configuration peut ne pas correspondre exactement à ce que l'on souhaite. Cependant il est possible d'en modifier, a posteriori, la configuration des interfaces. Avant toute chose, il est préférable de vérifier la configuration d'une interface au moyen de la commande `ifconfig` ayant comme paramètre le nom de l'interface à étudier. Voici le résultat d'exécution de la commande `ifconfig` appliquée à la première carte Ethernet du système :

```

gateway:~# ifconfig eth0
eth0     Link encap:Ethernet  HWaddr 00:50:04:32:A7:E7
         inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3072771 errors:0 dropped:0 overruns:0 frame:0
         TX packets:2868667 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:100
         Interrupt:11 Base address:0x6100

```

Les champs `MTU` et `Metric` montrent les valeurs pour cette interface. Sur un système Linux le champ `Metric` n'est pas exploité, il est uniquement défini pour des raisons de compatibilité. Les lignes `RX` et `TX` fournissent des informations statistiques sur le fonctionnement de l'interface, elles correspondent aux champs retournés par la commande `netstat`. La commande `ifconfig` affiche également des informations sur les caractéristiques physiques du périphérique telles que :

- l'adresse MAC de la carte ;
- l'adresse mémoire utilisée ;
- l'interruption employée.

La seconde ligne d'information stipule l'adresse IP, l'adresse broadcast ainsi que le masque de sous-réseau attribué à cette interface. La troisième ligne contient sous une forme textuelle la valeur des flags relatifs à cette interface (voir commande `netstat`).

5. Affectation un masque de sous-réseau.

Pour fonctionner correctement, chaque interface d'un même réseau doit posséder le même masque de sous-réseau. Supposons que l'on dispose de deux machines interconnectées par l'intermédiaire de l'interface réseau `eth0`. Elles doivent posséder le même masque de sous-réseau, pour leur interface `eth0` respective. Le résultat de l'exécution de la commande `ifconfig` sur les deux machines permet de visualiser l'existence d'un masque réseau identique.

Voici le résultat d'exécution des commandes `ifconfig` sur les deux machines concernées.

```
gateway:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:04:32:A7:E7
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3072771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2868667 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x6100
```

```
[root@linux /root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:60:08:4A:5B:F6
          inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
          IPX/Ethernet 802.3 addr:0060084A5BF6
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:1331052 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1291902 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0xb800
```

6. Affectation de l'adresse broadcast.

Le RFC 919, Broadcasting internet datagrams, définit le format d'une adresse broadcast comme étant une adresse dont tous les bits décrivant la machine sont positionnés à 1. Cette définition de l'adresse broadcast devrait permettre à la commande ifconfig d'en déterminer automatiquement la valeur. Ce n'est malheureusement pas toujours le cas. TCP/IP a été incorporé à UNIX BSD 4.2 avant que le RFC 919 ne soit adopté comme norme. Cette version de UNIX utilise une adresse broadcast dont tous les bits relatifs à la machine sont positionnés à 0. De plus, ce système d'exploitation ne permet pas la modification de l'adresse broadcast durant la configuration du système. Pour cette raison historique, certaines versions d'UNIX mettent, par défaut, l'adresse broadcast au « format 0 » pour être compatibles avec certains anciens systèmes, alors que d'autres versions utilisent la norme et mettent par défaut l'adresse au « format 1 ».

Afin d'éviter toute confusion lors de la définition des adresses broadcast d'un réseau, il est préférable de définir explicitement cette adresse en choisissant l'un des deux formats disponibles. De nos jours, il est vivement conseillé de suivre la norme RFC 919.

7. Autres options.

Jusqu'à présent, la commande ifconfig a été employée principalement pour fixer l'adresse de l'interface, le masque de sous-réseau et l'adresse broadcast. Bien qu'il s'agisse probablement de l'utilité majeure de la commande ifconfig, elle possède d'autres options. Elle peut activer ou désactiver :

- la mise en queue de l'encapsulation des paquets IP ;
- l'utilisation d'ARP ;
- l'interface en elle-même.

La commande ifconfig peut également fixer les valeurs MTU et Metric.

7.1. Activer et désactiver l'interface.

La commande ifconfig possède deux arguments, up et down pour activer et désactiver l'interface réseau. L'argument up active l'interface réseau alors que l'argument down désactive l'interface réseau qui ne pourra plus être utilisée.

Lors de la modification de l'adresse IP associée à une interface, il est impératif de désactiver l'interface avant d'attribuer une nouvelle adresse IP. Le changement d'adresse IP doit être effectué en trois temps :

- désactivation de l'interface réseau ;
- modification de l'adresse IP ;
- activation de l'interface réseau.

7.2. ARP et trailers.

Dans le cas des interfaces Ethernet, la commande `ifconfig` admet les arguments `arp` et `trailers`. L'argument `trailers` active ou désactive la négociation de la mise en queue de l'encapsulation des paquets IP. Pour rappel, les paquets IP sont envoyés à travers plusieurs réseaux physiques en étant encapsulés dans les trames de ces réseaux physiques. La mise en queue est une technique optionnelle qui réduit le nombre de copies mémoire à mémoire que le système récepteur est obligé d'effectuer. Pour activer la mise en queue, il suffit d'ajouter à la commande `ifconfig` l'argument `trailers`. Lorsque cette option est active, cela impose aux autres systèmes d'effectuer une mise en queue lors de la transmission des données. Sur un système Linux, cette option n'existe pas.

Le protocole ARP (Address Resolution Protocol) traduit les adresses IP en mac adresses. Ce protocole peut être activé au moyen de l'option `arp` de la commande `ifconfig`. La désactivation de ce protocole repose sur l'argument `-arp` de la commande `ifconfig`.

Bien que cela ne soit pas recommandé, il est donc possible de désactiver l'utilisation du protocole ARP. Cette décision de désactivation ne devrait être prise qu'au moment de l'utilisation d'un réseau expérimental disposant d'un matériel spécifique. Dans ces circonstances très particulières où des équipements non-standard existent il est important de disposer de la capacité de désactivation du protocole ARP.

7.3. Metric.

Sur certains systèmes, la commande `ifconfig` crée une entrée dans la table de routage pour chaque interface qui possède une adresse IP. Chaque interface est une route vers un réseau. Même si une machine n'est pas une passerelle, son interface est toujours une route vers le réseau local. La commande `ifconfig` détermine la route vers le réseau destination en appliquant le masque de sous-réseau de l'interface à l'adresse IP de l'interface. Par exemple, si une interface réseau possède l'adresse IP 192.168.1.10 et un masque de sous-réseau 255.255.255.0 le réseau de destination est 192.168.1.0.

Le protocole RIP (Routing Information Protocol) est un protocole de routage fréquemment utilisé sous UNIX. Ce protocole a deux utilités :

- diffuser les informations de routage aux autres machines ;
- utiliser les informations de routage entrantes pour construire des tables de routage dynamiques.

Les routes créées par la commande `ifconfig` sont une des sources d'informations de routage diffusées par le protocole RIP. L'argument `metric` de cette commande peut être employé pour contrôler l'utilisation de cette information au niveau du protocole RIP.

Le protocole RIP base sa décision de routage en fonction du coût d'une route par une métrique associée à la route. Cette information ne peut être qu'un nombre. Plus le nombre est petit, plus le coût pour emprunter cette route est faible. Lors de la construction de la table de routage, le protocole RIP favorise les routes à faible coût plutôt que celles plus onéreuses. Un coût faible est généralement associé aux réseaux directement connectés. Par conséquent, le paramètre `metric` par défaut est 0 pour une route passant par une interface directement connectée au réseau. Dans le cas des systèmes Linux, les scripts de démarrage du système donnent la valeur 1 au paramètre `metric`.

En pratique, l'argument `metric` n'est utilisé qu'à la condition de disposer de plusieurs routes ayant une même destination ce qui privilégie l'utilisation d'une route au détriment d'une autre. Lorsqu'un système dispose de plusieurs interfaces vers une même destination, une valeur `metric` faible est associée au réseau de plus haut débit.

7.4. MTU.

Tout réseau possède un MTU (Maximum Transmission Unit), abréviation qui désigne la taille du plus grand paquet pouvant être transmis sans fragmentation sur le réseau physique. Sur un réseau Ethernet, la taille maximale est de 1500 bytes. Le comportement par défaut de la commande `ifconfig` consiste à choisir la valeur la plus grande en fonction du réseau physique employé. Généralement un MTU de grande taille fournit de meilleures performances, toutefois, une valeur plus petite peut être utilisée pour les raisons suivantes :

- éviter la fragmentation. Si les données transitent d'un réseau FDDI (MTU de 4500) vers un réseau Ethernet (MTU de 1500), il est préférable d'employer un MTU de 1500 afin d'éviter une fragmentation des paquets, origine d'une baisse de performance. Naturellement, l'administrateur effectuera cette diminution s'il a de

bonnes raisons de penser que la baisse des performances tire sa source dans la fragmentation des paquets et à cette condition seulement ;

- lors de l'utilisation d'équipements dont la performance est particulièrement faible il n'est pas possible d'utiliser une taille standard de 1006 bytes. Dans ce cas, il est préférable de diminuer la valeur du MTU. Toutefois, une telle solution n'est que temporaire car il faudra songer au remplacement du matériel par un dispositif mieux adapté à l'application employée.

7.5. Mode promiscuous.

Un réseau broadcast permet à une interface de recevoir tous les paquets y compris ceux non destinés à l'hôte propriétaire de l'interface. Ce mode de fonctionnement de l'interface réseau autorise l'utilisation d'outils d'analyse de paquets, technique de mise en évidence des problèmes particulièrement malaisés à détecter.

Ce mode de fonctionnement ouvre également la possibilité d'opérer des actions illicites. L'on peut en effet visualiser le nom et le mot de passe des utilisateurs qui se connectent à un système distant. Lors de la configuration d'un parc informatique, il est important de ne pas configurer toutes les interfaces en mode promiscuous. Afin d'éviter le plus possible les risques de piratage, il est vivement conseillé d'employer des systèmes de cryptage des informations transitant au travers d'un réseau.

8. Configuration permanente.

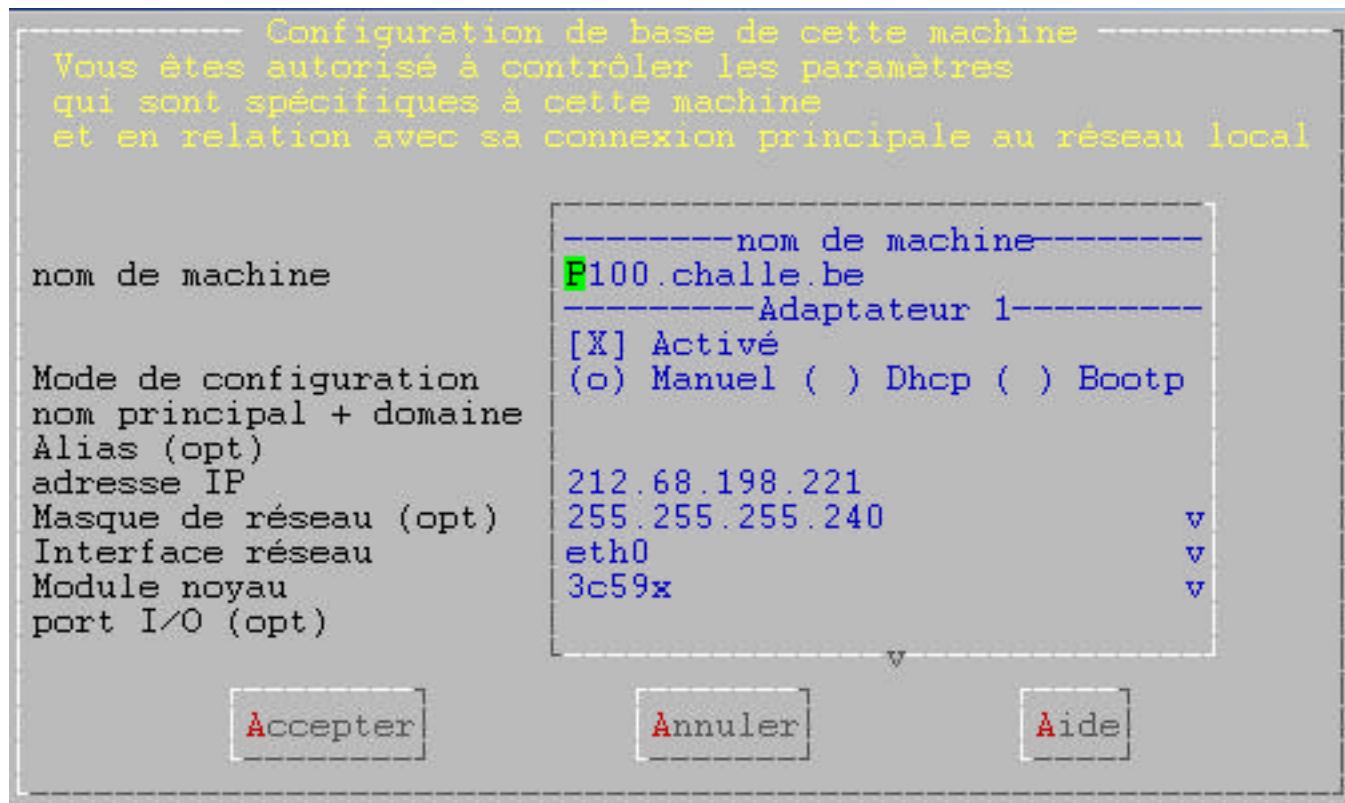
Après chaque démarrage du système, les modifications effectuées, au moyen de la commande `ifconfig`, sont oubliées. Pour les rendre permanentes, il s'indique d'éditer les scripts de démarrage du système. Ces scripts utilisent plusieurs fichiers de configuration qui figurent, dans le cas d'un système Linux RedHat, dans le répertoire `/etc/sysconfig/network-scripts`. Pour chaque interface réseau il existe un fichier `ifcfg-eth` contenant la configuration de l'interface. Voici le résultat d'exécution de la commande `ifconfig` sur une machine ainsi que le contenu du fichier de configuration relatif à cette même interface réseau :

```
[root@P100 network-scripts]# ifconfig eth1
eth1      Lien encap:Ethernet  HWaddr 00:00:C0:36:4B:2E
          inet adr:192.168.1.1  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Paquets Reçus:1783 erreurs:0 jetés:0 débordements:0 trames:0
          Paquets transmis:1479 erreurs:0 jetés:0 débordements:0 carrier:0
          collisions:1 lg file transmission:100
          Interruption:10 Adresse de base:0x230 Mémoire:dc000-e0000
```

```
[root@P100 network-scripts]# cat ifcfg-eth1
DEVICE="eth1"
USERCTL=no
ONBOOT="yes"
BOOTPROTO="none"
BROADCAST=192.168.1.255
NETWORK=192.168.1.0
NETMASK="255.255.255.0"
IPADDR="192.168.1.1"
IPXNETNUM_802_2=" "
IPXPRIMARY_802_2="no"
IPXACTIVE_802_2="no"
IPXNETNUM_802_3=" "
IPXPRIMARY_802_3="no"
IPXACTIVE_802_3="no"
IPXNETNUM_ETHERII=" "
IPXPRIMARY_ETHERII="no"
IPXACTIVE_ETHERII="no"
IPXNETNUM_SNAP=" "
IPXPRIMARY_SNAP="no"
IPXACTIVE_SNAP="no"
```

9. Linuxconf.

La majorité des systèmes UNIX offrent des outils permettant d'éviter la modification manuelle des fichiers de configuration aux administrateurs. Sous Linux, l'utilitaire linuxconf permet, notamment, la configuration des interfaces réseau. Voici un exemple des informations affichées par linuxconf lors de la configuration des interfaces réseau :



Configuration/Réseau/Tâches clientes/Configuration de base de la machine

Après avoir modifié les paramètres, linuxconf se charge :

- de transcrire les renseignements dans les fichiers de configuration du système ;
- d'activer les changements.

IV. Configuration du routage.

Sans le routage, le trafic TCP/IP serait limité à un seul réseau physique. Le routage permet à une machine d'en atteindre d'autres en traversant, si nécessaire, un grand nombre de réseaux intermédiaires.

La plupart des problèmes rencontrés dans la vie quotidienne d'un administrateur de réseaux sont causés par une mauvaise configuration des routeurs et non pas par des erreurs inhérentes aux protocoles de routage.

1. Configuration générale du routage.

Il est impératif de distinguer le routage des protocoles de routage. Tous les systèmes routent des informations mais tous n'exécutent pas des protocoles de routage. Le routage consiste à diffuser des datagrammes en fonction des informations présentes dans la table de routage. Les protocoles de routage sont des programmes qui s'échangent des informations afin de construire des tables de routage.

La configuration de routage d'un réseau ne nécessite pas toujours l'utilisation d'un protocole de routage. Dans le cas où les informations de routage sont immuables, il suffit de créer la table de routage manuellement. Il existe trois configurations possibles :

- le routage minimum. Un réseau totalement isolé de tout autre réseau TCP/IP ne nécessite qu'un routage minimum. Bon nombre de systèmes créent une table de routage minimale lors de l'exécution de la commande `ifconfig`. Cela n'est pas le cas des systèmes Linux où il faut explicitement créer toutes les routes au moyen de la commande `route` ;
- le routage statique. Un réseau possédant un nombre réduit de passerelles vers d'autres réseaux TCP/IP peut être configuré statiquement. Une table de routage statique est construite manuellement en utilisant la commande `route`. Les tables de routage statiques n'évoluent pas en fonction des modifications du réseau. Ce type de configuration est utilisé lorsque les routes ne changent pas ;
- le routage dynamique. Lorsqu'un réseau possède plus d'une route possible vers la même destination il est impératif d'employer un routage dynamique. Une table de routage dynamique est générée à partir des informations échangées par les protocoles de routage. Ces protocoles sont conçus pour échanger des informations d'ajustement du contenu des tables de routage en fonction de l'évolution des réseaux. Les protocoles de routage gèrent des situations complexes de manière sûre et efficace.

En résumé, les routes sont créées :

- automatiquement par la commande `ifconfig` ;
- manuellement par l'administrateur système ;
- dynamiquement par les protocoles de routage.

2. Table de routage minimale.

Examinons le contenu d'une table de routage lorsque les interfaces réseau sont configurées. Sous Linux, supposons également que les routes ont été ajoutées. Comme le montre l'exemple suivant, la commande `netstat` permet de visualiser le contenu de la table de routage

```
[jfc@P100 jfc]$ netstat -nr
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic   MSS Fenêtre  irtt  Iface
192.168.1.1      0.0.0.0         255.255.255.255 UH      0 0        0 eth1
212.68.198.221  0.0.0.0         255.255.255.255 UH      0 0        0 eth0
212.68.198.208  0.0.0.0         255.255.255.240 U       0 0        0 eth0
192.168.1.0      0.0.0.0         255.255.255.0   U       0 0        0 eth1
127.0.0.0        0.0.0.0         255.0.0.0       U       0 0        0 lo
0.0.0.0          212.68.198.209 0.0.0.0         UG      0 0        0 eth0
```

Cette table de routage dénombre trois interfaces. Pour chaque interface, la commande `netstat` montre les routes qui ont été définies. La seconde ligne indique qu'il est possible d'atteindre au travers de l'interface `eth0` l'hôte (indicateur H) dont l'adresse est `212.68.198.221`. La troisième ligne précise que le sous-réseau `212.68.198.208` est joignable par

l'entremise de l'interface eth0. La dernière ligne informe de ce que n'importe quelle destination (0.0.0.0) est atteinte par l'intermédiaire de la passerelle (indicateur G) dont l'adresse IP est 212.68.198.209.

Les capacités de cette table de routage sont aisément vérifiables au moyen de la commande ping. Cette commande émet le datagramme ICMP Echo Request pour forcer la machine distante à émettre un datagramme ICMP Echo Response vers la machine locale. Si les datagrammes peuvent partir de la machine locale vers la machine destination et en revenir, les deux machines peuvent communiquer entre elles. Voici un exemple d'utilisation de la commande ping.

```
[root@P100 /root]# ping 212.68.198.209
PING 212.68.198.209 (212.68.198.209) from 212.68.198.221 : 56(84) bytes
64 bytes from gateway (212.68.198.209): icmp_seq=0 ttl=64 time=56.8 ms
64 bytes from gateway (212.68.198.209): icmp_seq=1 ttl=64 time=78.1 ms
64 bytes from gateway (212.68.198.209): icmp_seq=2 ttl=64 time=53.7 ms
64 bytes from gateway (212.68.198.209): icmp_seq=3 ttl=64 time=26.2 ms
64 bytes from gateway (212.68.198.209): icmp_seq=4 ttl=64 time=53.7 ms
64 bytes from gateway (212.68.198.209): icmp_seq=5 ttl=64 time=50.1 ms
64 bytes from gateway (212.68.198.209): icmp_seq=6 ttl=64 time=55.9 ms
64 bytes from gateway (212.68.198.209): icmp_seq=7 ttl=64 time=87.1 ms
64 bytes from gateway (212.68.198.209): icmp_seq=8 ttl=64 time=56.7 ms
64 bytes from gateway (212.68.198.209): icmp_seq=9 ttl=64 time=50.1 ms
64 bytes from gateway (212.68.198.209): icmp_seq=10 ttl=64 time=25.1 ms
64 bytes from gateway (212.68.198.209): icmp_seq=11 ttl=64 time=53.7 ms

--- 212.68.198.209 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 25.1/53.9/87.1 ms
```

3. Création d'une table de routage statique.

La commande route est utilisée soit pour ajouter soit pour supprimer manuellement des entrées dans la table de routage. Par exemple, pour ajouter une route vers le sous-réseau 212.68.198.208 dans la table de routage d'une machine fonctionnant sous Linux, il suffit de dactylographier :

```
[root@P100 /root]# route add -net 212.68.198.208
```

Le premier argument donné dans cet exemple est l'option add. La commande route prend comme premier argument soit add, soit delete, ce qui lui permet de déterminer si elle doit ajouter une nouvelle route ou détruire une route existante. Il n'y a pas d'opération par défaut. Si aucun argument n'est donné, la commande route affiche le contenu de la table de routage.

La valeur qui suit est l'adresse de destination qui est celle atteinte via cette route.

Le fonctionnement de la commande route peut sembler obscur car rien ne spécifie dans la syntaxe de cette commande, l'interface à laquelle la route doit être appliquée. Le noyau du système compare l'adresse de destination aux adresses réseau de toutes les interfaces configurées. Seule l'interface eth0 a une adresse réseau correspondant à l'adresse destination dans la commande route. Ce principe de fonctionnement justifie l'absence du nom de l'interface réseau dans la syntaxe de la commande route.

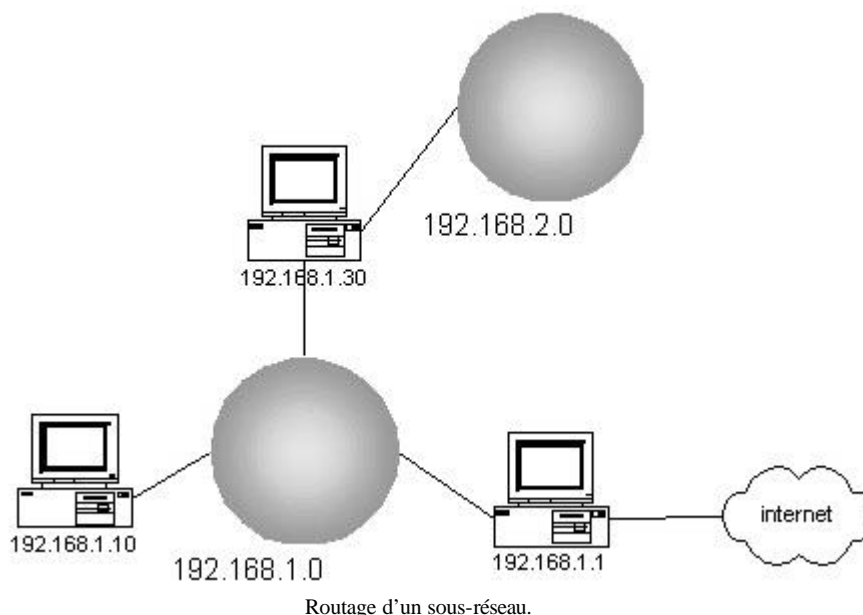
Si l'option default est utilisée comme adresse de destination, la commande route crée une route par défaut (l'adresse associée à une route par défaut est 0.0.0.0). La route par défaut est utilisée chaque fois qu'il n'y a pas de route spécifique vers une destination donnée. Lorsqu'un réseau ne dispose que d'une seule passerelle, une route par défaut est une solution simple pour rediriger l'ensemble du trafic à destination de réseaux externes.

L'option gw permet d'indiquer l'adresse IP de la passerelle externe à travers laquelle les données sont transmises au destinataire. Cette adresse doit être celle d'une passerelle ou bien celle d'un réseau directement connecté. Les routes TCP/IP définissent l'étape suivante à atteindre sur le chemin qui mène les paquets jusqu'à leur destination. Ce prochain relais doit être directement accessible par la machine locale. Il est donc nécessaire qu'il se trouve sur le même réseau.

Le dernier argument de la commande route concerne le paramètre metric du routage. Un grand nombre de systèmes exige la présence de ce paramètre lors de la définition d'une route.

3.1. Ajouter des routes statiques.

Supposons que le réseau d'une entreprise corresponde au schéma suivant :



Imaginons que toutes les machines du réseau soient configurées sauf celle portant l'adresse IP 192.168.1.10. Le réseau d'adresse 192.168.1.0 est constitué de deux passerelles, l'une donnant accès au réseau 192.168.2.0 et une autre permettant de rejoindre Internet. La passerelle 192.168.1.1 sera la passerelle par défaut car utilisée par des milliers de routes. Un nombre plus limité de routes passant par 192.168.1.30 facilite leurs saisies. Le nombre de routes traversant une passerelle est l'élément décisif du choix de la passerelle à adopter comme étant par défaut. Même si la majorité du trafic réseau transite par la passerelle 192.168.1.30 pour rejoindre les autres machines du réseau, celle par défaut devrait être 192.168.1.1.

La mise en service de l'ordinateur portant l'adresse IP 192.168.1.10 nécessite l'exploitation des commandes `ifconfig` et `route` pour configurer respectivement l'interface réseau et le routage. Voici la commande de configuration de l'interface réseau de cet ordinateur :

```
[root@P200 /root]# ifconfig eth0 192.168.1.10 netmask 255.255.255.0
[root@P200 /root]# ifconfig eth0
eth0      Lien encap:Ethernet  HWaddr 00:A0:24:72:4C:A3
          inet adr:192.168.1.10  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Paquets Reçus:9713 erreurs:0 jetés:0 débordements:0 trames:0
          Paquets transmis:8864 erreurs:0 jetés:0 débordements:0 carrier:0
          collisions:2 lg file transmission:100
          Interruption:14 Adresse de base:0xe000
```

Afin de rendre cette interface utilisable, il est nécessaire de définir des routes. Il faut spécifier que l'interface peut :

- rejoindre l'hôte 192.168.1.10 ;
- atteindre le réseau 192.168.1.0 ;
- utiliser la passerelle 192.168.1.1 comme passerelle par défaut ;
- utiliser la passerelle 192.168.1.30 pour rejoindre le sous-réseau 192.168.2.0.

La commande `route` doit être utilisée quatre fois de manière à garnir correctement la table de routage. Voici les quatre utilisations de la commande `route` nécessaires à la configuration du routage :

```
[root@P200 /root]# route add -host 192.168.1.10 eth0
[root@P200 /root]# route add -net 192.168.1.0
[root@P200 /root]# route add default gw 192.168.1.1
[root@P200 /root]# route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.30
```

L'option `-host`, de la commande `route`, permet de spécifier que la destination est un hôte (une machine). L'option `-net` stipule que la destination est un réseau. L'option `default gw` précise que toutes les destinations autres que celles présentes dans la table de routage sont joignables par l'intermédiaire d'une passerelle dont l'adresse IP est 192.168.1.1. L'option `default` est un moyen de spécifier l'adresse 0.0.0.0 désignant n'importe quelle destination possible. L'option `gw` employée seule permet de spécifier la passerelle donnant accès au sous-réseau d'adresse 192.168.2.0

```
[root@P200 /root]# netstat -nr
Table de routage IP du noyau
Destination      Passerelle      Genmask         Indic   MSS  Fenêtre  irtt  Iface
192.168.2.0      192.168.1.30   255.255.255.0  UG      0 0          0  eth0
192.168.1.0      0.0.0.0         255.255.255.0  U       0 0          0  eth0
127.0.0.0        0.0.0.0         255.0.0.0      U       0 0          0  lo
0.0.0.0          192.168.1.1    0.0.0.0        UG      0 0          0  eth0
```

Il est possible de définir le routage de la machine 192.168.1.10 sans déterminer l'accès au réseau 192.168.2.0. Si la passerelle 192.168.1.1 est correctement configurée, sa table de routage définit l'accès à ce sous-réseau. Les paquets provenant de la machine 192.168.1.10 à destination du sous-réseau 192.168.2.0 seraient alors redirigés vers la bonne passerelle. En réalité, la machine d'adresse 192.168.1.10 est informée par la passerelle par défaut de ce que les informations doivent être acheminées vers 192.168.1.30. C'est un message ICMP Redirect qui est envoyé par la passerelle par défaut. Ce message a pour effet de modifier la table de routage de la machine 192.168.1.10 pour tenir compte du nouveau trajet possible.

```
[root@P200 /root]# netstat -nr
Table de routage IP du noyau
Destination      Passerelle      Genmask         Indic   MSS  Fenêtre  irtt  Iface
192.168.2.0      192.168.1.30   255.255.255.0  UGD     0 0          0  eth0
192.168.1.0      0.0.0.0         255.255.255.0  U       0 0          0  eth0
127.0.0.0        0.0.0.0         255.0.0.0      U       0 0          0  lo
0.0.0.0          192.168.1.1    0.0.0.0        UG      0 0          0  eth0
```

Certains administrateurs réseau exploitent ces redirections lors de la conception du réseau. Toutes les machines sont configurées avec une route par défaut, même celles connectées à un réseau possédant plus d'une passerelle. Ces passerelles échangent des informations de routage via les protocoles de routage et redirigent les machines vers la meilleure passerelle à utiliser pour une route donnée. Ce type de routage, dépendant des redirections ICMP, est devenu très populaire grâce à l'utilisation des ordinateurs personnels. Beaucoup de PC ne peuvent pas exécuter de protocole de routage. Certains le peuvent mais ne possèdent pas la commande `route` et sont limités à une seule route par défaut. De façon évidente, ce type de routage est aisé à mettre en place et est bien adapté à une configuration via un serveur de configuration, puisque la même route par défaut est utilisée sur toutes les machines. Pour ces raisons, certains administrateurs réseau encouragent la redirection de messages ICMP.

Par contre d'autres administrateurs préfèrent éviter ces redirections et maintiennent directement le contenu des tables de routage. Au moyen de la commande `route`, et pour éviter ces redirections, des routes spécifiques doivent être installées pour chaque sous-réseau.

3.2. Ajouter des routes statiques à l'amorçage.

Le choix d'un routage statique impose la modification des scripts de démarrage afin d'y ajouter les routes statiques souhaitées.

Sous un système Linux RedHat, le fichier `/etc/sysconfig/static-routes` contient la définition des routes statiques. Voici un exemple de contenu de ce fichier :

```
[root@gateway2 sysconfig]# cat static-routes
eth0 net 192.168.22.0 netmask 255.255.255.0 gw 192.168.1.10
eth0 net 192.168.23.0 netmask 255.255.255.0 gw 192.168.1.10
eth0 net 192.168.24.0 netmask 255.255.255.0 gw 192.168.1.10
eth0 net 192.168.25.0 netmask 255.255.255.0 gw 192.168.1.10
eth0 net 192.168.213.0 netmask 255.255.255.0 gw 192.168.1.30
eth0 net 192.168.220.0 netmask 255.255.255.0 gw 192.168.1.30
eth0 net 192.168.217.0 netmask 255.255.255.0 gw 192.168.1.30
eth0 net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.3
```

Le système RedHat ne nécessite donc pas la modification des scripts de démarrage du système mais l'édition d'un fichier de configuration.

4. Les protocoles de routage internes.

Les protocoles de routage sont divisés en deux groupes :

- les protocoles internes ;
- les protocoles externes.

Un protocole interne est utilisé à l'intérieur de systèmes réseau indépendants. En utilisant la terminologie TCP/IP, ces systèmes réseau indépendants sont appelés des systèmes autonomes. Un système autonome est un ensemble de réseaux et de passerelles avec son propre mécanisme interne de traitement des informations de routage et de transmission de ces informations. A l'intérieur d'un système autonome, les informations de routage sont donc échangées en utilisant un protocole interne choisi par l'administrateur du système.

Tous les protocoles de routage internes réalisent les mêmes opérations de base à savoir :

- déterminer la meilleure route à suivre pour atteindre une destination donnée ;
- diffuser les informations de routage aux autres systèmes du réseau.

Il existe plusieurs protocoles de routage internes, ils sont :

- RIP (Routing Information Protocol) est le protocole de routage interne le plus répandu sur les plates-formes UNIX. Ce protocole, destiné aux réseaux locaux fait partie intégrante des systèmes UNIX. RIP choisit la meilleure route en utilisant le nombre le plus faible d'étapes. Il s'agit du nombre de passerelles à travers lesquelles les informations doivent transiter avant d'atteindre leur destination. Le protocole RIP suppose la meilleure route comme étant celle qui utilise le moins de passerelles. Cette approche est appelée algorithme à vecteur de distance.
- Hello est un protocole utilisateur du temps comme facteur de décision lorsqu'il est amené à choisir une route. Le principe est d'exploiter la durée mise par un datagramme pour effectuer un trajet aller-retour entre la source et la destination. Un paquet Hello contient l'heure d'expédition. Lorsque le paquet arrive à destination, le système récepteur détermine le temps mis par le paquet pour atteindre la destination. Hello est un protocole de moins en moins utilisé. A l'origine, il était employé sur la dorsale NSFNET à 56kps.
- IS-IS (Intermediate System to Intermediate System) est un protocole interne de routage provenant du modèle OSI. Il s'agit d'un protocole par state-link (état des liaisons) de type SPF (Shortest Path First). Ce protocole est utilisé sur les lignes T1 dont la capacité est de 1.544 Mb/s.
- OSPF (Open Shortest Path First) est un autre protocole par état des liaisons développé pour TCP/IP. Il est particulièrement bien adapté aux réseaux importants.

Les protocoles RIP et OSPF seront les seuls à être développés dans la suite de cet exposé car ils sont les plus fréquemment employés.

4.1. Le protocole RIP.

Le protocole RIP est lancé par le démon de routage routed. Dès son lancement, ce démon effectue une requête en vue d'obtenir les mises à jour en matière de routage ensuite il attend les réponses à ses requêtes. Lorsqu'un routeur est configuré pour répondre à ce type de requêtes, il répond par un paquet de mise à jour constitué des informations de sa

table de routage. Ce paquet contient les adresses de destination de la table de routage et les métriques de routage associées à chaque destination. Ces paquets sont émis non seulement pour répondre à des requêtes mais aussi de temps à autre pour que les informations de routage restent valables.

Le démon `routed` utilise les informations contenues dans les paquets de mise à jour pour construire sa table de routage. Si ces informations contiennent une route dirigée vers une destination qui n'existe pas dans la table de routage locale, une nouvelle route est créée. Si dans les informations de mise à jour figure une route dont la destination est déjà présente dans la table de routage locale, cette nouvelle route n'est utilisée que si elle est de moindre coût que celle déjà mémorisée. Le coût de la nouvelle route est déterminé en additionnant au coût pour atteindre la passerelle émettrice du paquet de mise à jour le coût spécifié dans le paquet. Si la somme obtenue est inférieure à la métrique actuelle de la route, la nouvelle route est utilisée.

Le protocole RIP détruit également des routes de la table de routage. Cette opération de destruction est effectuée dans deux cas, si :

- la passerelle vers une destination indique que le coût de la route est supérieur à 15 ;
- RIP présume qu'une passerelle qui n'envoie pas régulièrement d'informations de mise à jour est hors service. Toutes les routes passant par cette passerelle sont détruites si aucune mise à jour n'est reçue après un certain temps. En règle générale, des mises à jour sont transmises toutes les 30 secondes. Si une passerelle ne diffuse pas d'information après 180 secondes, toutes les routes passant par cette passerelle sont supprimées de la table de routage.

L'activation du protocole RIP nécessite le lancement du démon `routed` par l'intermédiaire de la commande `routed`. Le démon `routed` lit le fichier `/etc/gateways` au moment de son démarrage. Les informations contenues dans ce fichier sont ajoutées à la table de routage. Le protocole RIP est capable de créer une table de routage fonctionnelle en utilisant les mises à jour reçues des diffuseurs RIP. Il est parfois nécessaire de compléter cette information par une route initiale par défaut ou par des informations concernant une passerelle qui n'annonce pas ses routes. Dans ce cas, le fichier `/etc/gateways` offre les informations de routage supplémentaires. Ce fichier sert le plus souvent à définir une route par défaut comme le montre l'exemple suivant :

```
[root@P200 /etc]# cat gateways
net 0.0.0.0 gateway 192.168.1.1 metric 1 passive
```

Cette entrée débute par le mot-clé `net`. Toutes les entrées commencent soit par `net`, soit par `host` pour indiquer que l'adresse qui suit est soit celle d'un réseau, soit celle d'une machine. L'adresse de destination `0.0.0.0` est l'adresse utilisée pour la route par défaut. Ensuite, le mot-clé `gateway` est suivi de l'adresse de la passerelle qu'il faut contacter pour rejoindre la destination spécifiée. Après cela, le mot-clé `metric` est suivi d'une valeur numérique. Cette valeur désigne le coût de la route. Cette métrique représente le nombre de passerelles à franchir pour rejoindre la destination.

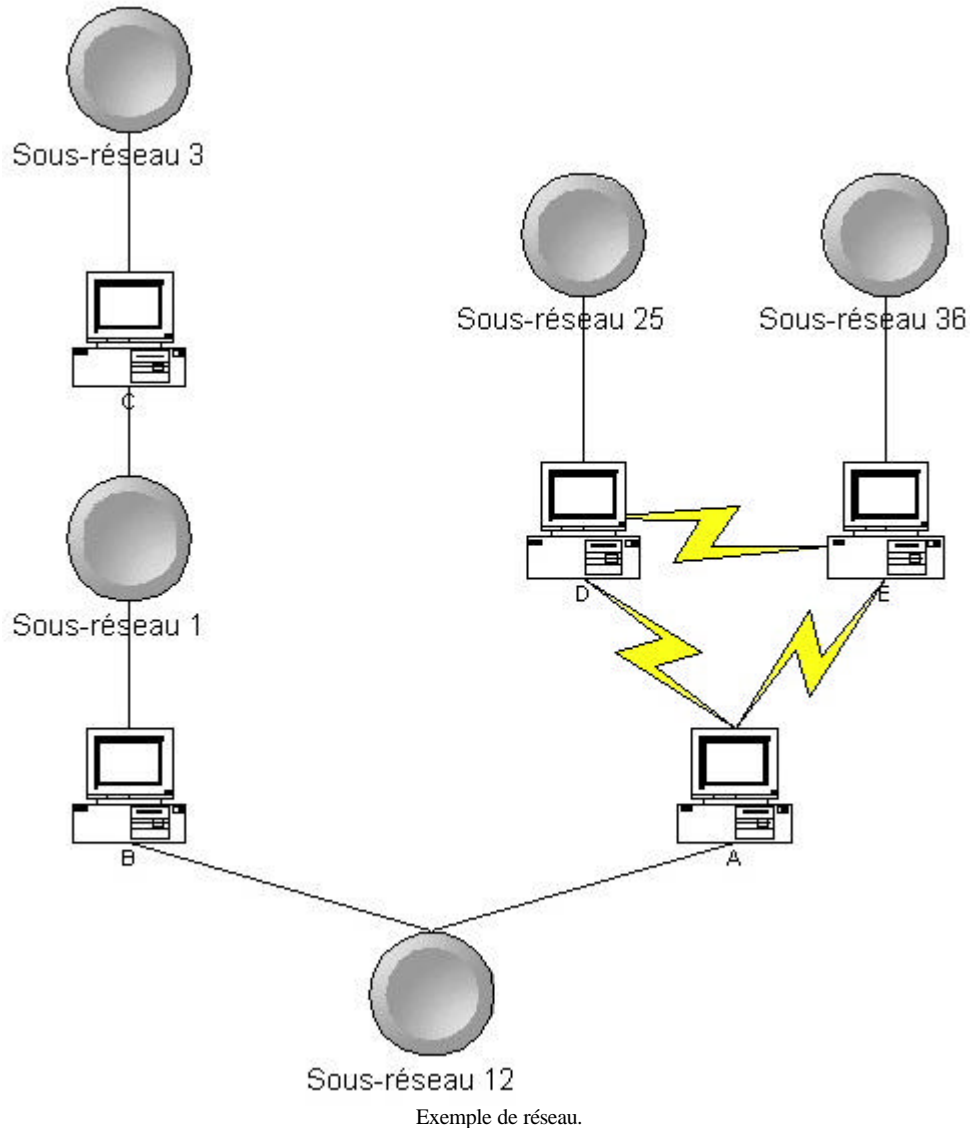
Toutes les entrées du fichier `/etc/gateways` se terminent soit par `passive`, soit par `active`. Le mot-clé `passive` signifie que la passerelle donnée dans l'entrée n'est pas à prendre en considération dans l'échange des informations de routage. Cela peut s'avérer utile lorsque la passerelle ne diffuse pas d'informations de routage. Dans ce cas, le mot-clé `passive` permet d'éviter la destruction de la route en raison de la non réception d'informations de mises à jour. L'utilisation de l'option `passive` est donc une autre manière de définir une route statique sous le contrôle du démon `routed`. Par contre, le mot-clé `active` crée une route qui peut être mise à jour par RIP. Une passerelle active est sensée produire régulièrement des informations de mises à jour.

Le protocole RIP semble aisé à implanter et à configurer. Néanmoins, RIP possède trois défauts :

- Envergure de réseau limitée. La plus longue route gérée par RIP est constituée de 15 relais. Un routeur RIP ne peut maintenir la table de routage d'un réseau constitué de plus de 15 passerelles en cascade.
- Convergence lente. La destruction d'une route nécessite l'échange d'informations de mises à jour jusqu'à ce que le coût atteigne 16 (comptage jusqu'à l'infini). Cela peut se produire lorsqu'un changement dans la structure du réseau éloigne une destination. De plus, RIP doit attendre 180 secondes avant de détruire une route incorrecte. En clair il faut attendre un délai relativement important avant que la table de routage reflète l'état réel du réseau.
- Routage sur des classes complètes. Pour le protocole RIP, toutes les adresses font partie des classes A, B ou C. Cela rend RIP incapable de gérer un réseau sans classe.

Il ne sera pas possible de changer l'envergure du réseau de manière significative car le protocole doit être capable d'atteindre une métrique dont la valeur est considérée comme infinie. Le travail réalisé sur le protocole RIP est concentré sur la convergence lente et sur le routage de classes complètes.

Des fonctionnalités ont été ajoutées au protocole RIP afin de résoudre le problème de la lenteur de convergence. Afin de comprendre les améliorations apportées, il faut maîtriser la notion de comptage jusqu'à l'infini.



L'exemple ci-dessus montre que A atteint le sous-réseau 3 via les passerelles B et C. Le sous-réseau 3 est à deux relais de A et à un relais de B. Donc, B annonce un coût de 1 pour le sous-réseau 3 et A annonce un coût de 2 pour ce même sous-réseau. Le trafic est donc routé par B pour atteindre le sous-réseau 3. Ce fonctionnement persiste jusqu'au moment où la passerelle C s'arrête. Dans ce cas, B attend une information de mise à jour émanant de C durant 180 secondes. Pendant cette période, B continue d'envoyer à A des informations de mises à jour stipulant que B se trouve toujours à 1 relais du sous-réseau 3. Lorsque les 180 secondes sont écoulées, B enlève toutes les routes qui passaient par C de sa table de routage. Néanmoins, A dispose toujours, dans sa table de routage, d'une information stipulant qu'il se trouve à 2 relais du sous-réseau 3. Une information de mise à jour est envoyée de A vers B ce qui provoque la mise à jour de la table de routage de B qui considère être maintenant à 3 relais du sous-réseau 3. La machine B envoie à la machine A une information de mise à jour stipulant qu'elle est à présent à 3 relais du sous-réseau 3. La machine A met sa table de routage à jour en considérant être à 4 relais du sous-réseau 3. Ce processus d'échange et d'augmentation du coût de la route vers le sous-réseau 3 continue jusqu'au moment où un coût de 16 est atteint sur les deux machines.

Le split horizon (horizon partagé) et le poison reverse (retour empoisonné) sont deux fonctionnalités qui tentent d'empêcher le comptage jusqu'à l'infini.

- La fonctionnalité split horizon permet à une passerelle de ne pas diffuser les informations de mises à jour vers les sous-réseaux qui ont permis la création de cette information. Cette règle interdit à la passerelle A de diffuser à la passerelle B des informations de routage concernant le sous-réseau 3 car les informations ont été obtenues à partir de la passerelle B.
- La fonctionnalité poison reverse est une amélioration du split horizon. L'interdiction de diffusion est la même que pour le split horizon. Avec cette fonctionnalité, la passerelle A annonce aux systèmes connectés par l'intermédiaire du sous-réseau 12 que le sous-réseau 3 a maintenant une métrique de 16.

En supposant que les fonctionnalités split horizon et poison reverse soient implantées, que se passe-t-il si la passerelle A cesse son activité ? Avec le split horizon, les passerelles D et E n'annoncent pas à A la route vers le sous-réseau 12 car elles ont appris la route de A. Toutefois, elles annoncent la route vers le sous-réseau 12 à tous les autres sous-réseaux. Lorsque la passerelle A s'arrête, les passerelles D et E effectuent leur propre comptage jusqu'à l'infini avant de supprimer la route vers le sous-réseau 12. Dès la fin du comptage à l'infini, les passerelles D et E envoient un trigger update (mise à jour déclenchée) aux autres sous-réseaux afin de limiter le comptage à l'infini et de minimiser la bande passante employée. Ainsi, lorsque les passerelles D et E ont terminé le comptage à l'infini, une mise à jour déclenchée est envoyée aux sous-réseaux 25 et 36 pour qu'ils suppriment la route vers le sous-réseau 12.

4.2. RIP version 2.

Le protocole RIP-2 défini par le RFC 1723 est une nouvelle version de RIP capable de prendre en charge les réseaux sans classe. Pour cela, le masque de sous-réseau fait partie intégrante des paquets échangés. Par rapport à RIP, les mises à jour sont transmises par l'intermédiaire de l'adresse multicast 224.0.0.9.

Les passerelles qui utilisent toujours le protocole RIP peuvent exploiter les informations du protocole RIP-2. Il suffit à ces routeurs d'extraire les données dont ils ont besoin des paquets RIP-2.

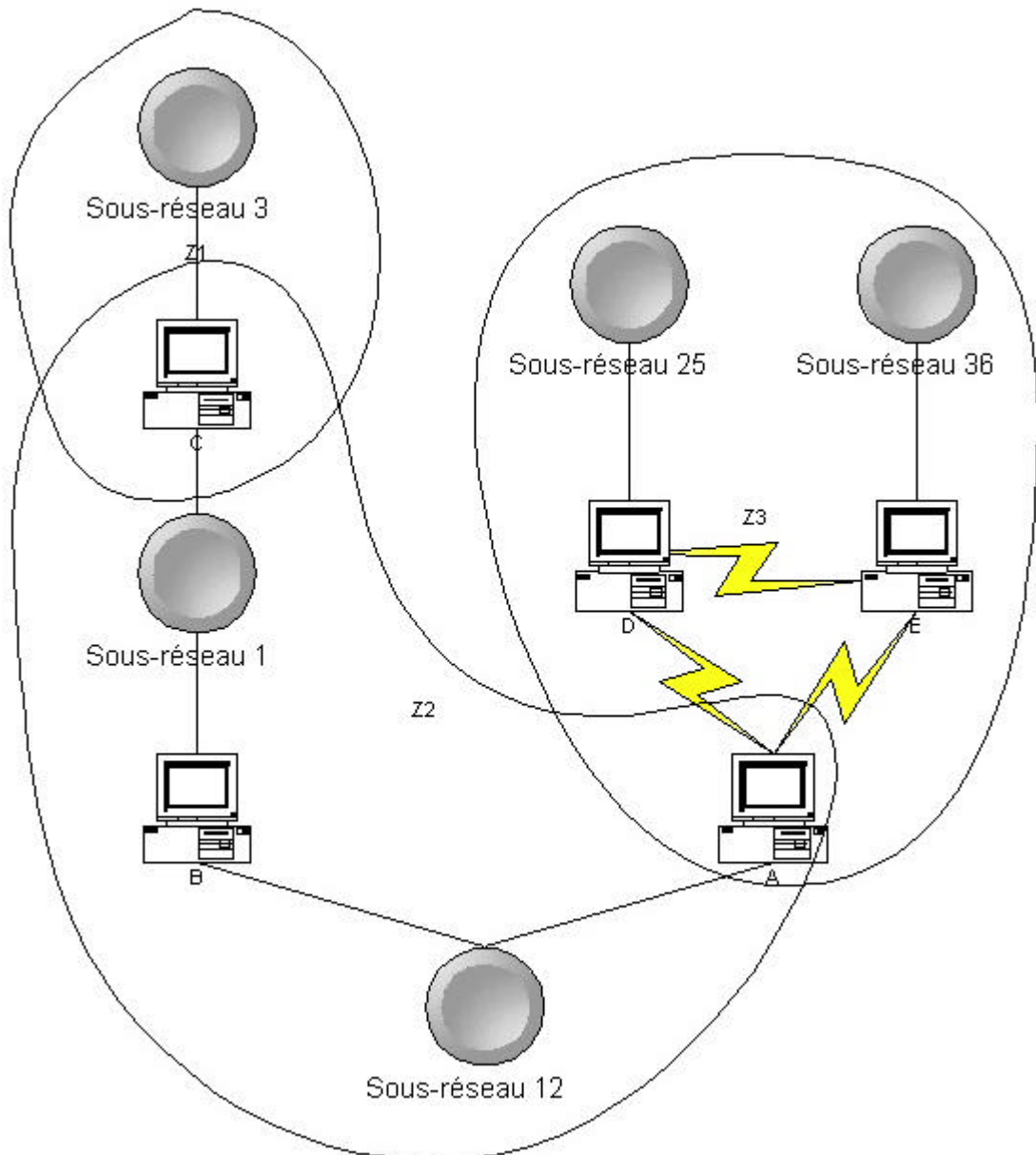
4.3. OSPF : Open Shortest Path First.

OSPF, défini dans le RFC 2178, est un protocole par état de liaison. Il est très différent de RIP. Un routeur utilisateur de RIP partage les informations concernant le réseau tout entier (au moins un système unique autonome) avec ses voisins. Inversement, un routeur qui utilise OSPF partage ses informations concernant ses voisins avec le réseau tout entier. OSPF définit quelques notions :

- Zone (area). Une zone est un ensemble arbitraire de réseaux, de machines, de routeurs interconnectés. Les zones échangent des informations de routage avec d'autres zones par l'intermédiaire de routeurs interzones (area border routers)
- Dorsale (backbone). Une dorsale est une zone particulière qui interconnecte toutes les zones à l'intérieur d'un système autonome. Chaque zone doit être connectée à la dorsale car elle est chargée de diffuser les informations de routage entre les zones.
- Zone terminale (stub area). Une zone terminale ne possède qu'un seul routeur interzone. En clair, il n'y a qu'une seule route vers l'extérieur de cette zone. Le routeur doit donc simplement s'annoncer comme étant la route par défaut.

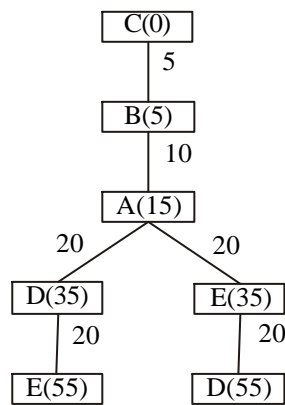
Un système autonome de grande taille doit être divisé en zones. Le réseau utilisé dans l'exemple ci-dessous est petit mais rien n'interdit de le diviser en zones. Supposons qu'il soit divisé en trois zones : la zone 1 contient le sous-réseau 3, la zone 2 contient les sous-réseaux 1 et 12 et la zone 3 contient les sous-réseaux 25, 36 et les liaisons PPP. De plus, on pourrait définir la zone 1 comme étant une zone terminale car C est dans cette zone le seul routeur interzone. Nous pourrions également définir la zone 2 comme étant une dorsale car elle interconnecte les deux autres zones et toutes les informations de routage entre les zones 1 et 3 doivent être diffusées par la zone 2. La zone 2 contient les routeurs interzones A et C et un routeur interne, B. La zone 3 contient trois routeurs A, D et E.

Le protocole OSPF offre une grande souplesse de découpe des systèmes autonomes.



Découpe du réseau en zones.

Tout routeur OSPF construit un graphe orienté modélisant le réseau tout entier. Ce graphe est une carte du réseau du point de vue du routeur. La racine du graphe est le routeur.



Graphe du routeur C.

Le graphe est construit à partir de la base de données d'états de liaison. Elle inclut des informations sur les routeurs du réseau ainsi que sur tous leurs voisins. Pour le système autonome qui nous occupe il y a 5 routeurs et 10 voisins. Un coût est défini pour chaque routeur et chaque liaison. Cela permet au système de calculer le plus court chemin dans un graphe ce qui détermine la route à emprunter pour atteindre une destination.

Pour découvrir ses voisins, un routeur envoie un paquet hello et attend de recevoir de ses voisins des paquets hello. Le paquet hello contient le routeur émetteur et la liste des routeurs adjacents.

5. Les protocoles de routage externe.

Les protocoles de routage externe sont utilisés pour échanger des informations entre les systèmes autonomes. Les informations de routage échangées entre les systèmes autonomes sont appelées informations de joignabilité (reachability information). Ces informations de joignabilité sont simplement des données spécifiant quel réseau peut être rejoint en passant par un système autonome.

5.1. EGP : Exterior Gateway Protocol.

Une passerelle qui utilise EGP annonce qu'elle peut atteindre les réseaux faisant partie de son système autonome. Elle n'annonce pas qu'elle peut atteindre des réseaux situés à l'extérieur de son système autonome.

Dans un premier temps, les systèmes s'échangent des messages afin d'établir un dialogue entre les passerelles EGP. Les participants à une communication EGP sont appelés des voisins. L'échange d'informations entre les passerelles est appelé acquisition d'un voisin.

Dès que les voisins sont acquis, les informations de routage sont demandées. Le voisin répond en envoyant un paquet d'accessibilité appelé mise à jour. La passerelle réceptrice utilise ces informations pour garnir sa table de routage. Si un voisin n'est pas à même de répondre à trois demandes successives d'informations de routage, la passerelle demanderesse supprime les routes qui passent par ce voisin de sa table de routage.

Actuellement, les passerelles centrales d'Internet ne sont pas à même de faire face à l'évolution rapide du réseau. Aussi, le protocole EGP qui était employé dans ces passerelles a-t-il été remplacé par un protocole distribué s'adaptant à l'évolution rapide d'Internet.

5.2. BGP : Border Gateway Protocol.

BGP est le protocole de routage le plus utilisé sur Internet. Il est basé sur une spécification de protocole du modèle OSI. BGP gère le routage sur base de raisons non techniques (accords de peering)

Les politiques de routage ne font pas partie du protocole BGP. Elles sont définies de manière externe au moment de la configuration du système. Les fournisseurs d'accès à Internet ont la faculté de développer des politiques de routage basées sur des accords bilatéraux avec d'autres fournisseurs d'accès.

BGP peut être configuré de manière à contrôler les routes annoncées ainsi que les routes acceptées des autres fournisseurs.

Le protocole BGP est implanté au-dessus de TCP/IP, ce qui permet d'utiliser un service de transmission fiable. BGP utilise le port 179. Il contacte ses voisins en utilisant le système de négociation TCP, en trois étapes.

5.3. Choisir un protocole de routage.

Il existe un nombre important de protocoles de routage. Néanmoins, le choix d'un protocole de routage est aisé. Les protocoles ont été établis pour répondre à des problèmes inhérents aux réseaux de grande taille. Dans ce cas, OSPF est un choix judicieux.

Dans le cas où un protocole de routage externe doit être mis en œuvre, le choix à opérer dépendra de celui effectué par le voisin. Pour que deux systèmes autonomes puissent s'échanger des informations de routage, il faut qu'ils exploitent le même protocole externe. Dans la majorité des cas, le protocole BGP est adopté.

Le type d'équipement influence également le choix du protocole. Les routeurs sont généralement des systèmes dédiés gérant un grand nombre de protocoles. Les ordinateurs ne font généralement pas tourner des protocoles de routage. Permettre à des ordinateurs de participer au routage dynamique pourrait réduire le choix de protocoles. Par exemple, les systèmes UNIX sont généralement livrés avec uniquement le protocole RIP. Toutefois, le démon gated des systèmes UNIX permet d'exploiter un grand nombre de protocoles.

Même si les performances des systèmes dédiés sont supérieures à celles des systèmes UNIX, gated permet d'exploiter une machine UNIX comme routeur.

6. gated : Gateway Routing Daemon.

Le développement de programmes de routage pour les systèmes UNIX est relativement limité. La majorité des machines UNIX effectuent des tâches de routage simple pour lesquelles RIP est suffisant. Les applications lourdes et complexes de routage demandant l'exploitation de protocoles avancés sont gérées par des routeurs spécialisés. Le démon gated offre aux systèmes UNIX la plupart des protocoles de routage avancés. Ce démon combine plusieurs protocoles de routage en un seul programme.

Le démon gated présente l'avantage d'offrir des fonctionnalités en principe uniquement disponibles sur des routeurs dédiés, elles sont celles ci :

- les informations de plusieurs protocoles sont combinées par gated de manière à choisir les meilleures routes ;
- des routes apprises par un protocole interne peuvent être diffusées en utilisant un protocole externe. Cela permet d'adapter les annonces externes aux changements internes ;
- les politiques de routage peuvent être implantées pour contrôler quelles routes sont acceptées et quelles routes sont diffusées ;
- tous les protocoles sont configurés au moyen du fichier /etc/gated.conf ;
- le démon gated est mis à jour régulièrement.

Dans toute implantation d'un protocole de routage, il existe un aspect externe et un aspect interne. Le côté externe permet l'échange d'informations avec les systèmes distants. Le côté interne quant à lui utilise les informations reçues des systèmes distants pour mettre à jour la table de routage. Dans le cas de gated, les fonctions du protocole externe sont les mêmes que celles des autres implantations. Par contre, la partie interne, est propre au système UNIX.

Chaque protocole possède sa propre métrique qui détermine la meilleure route à prendre. Aussi, gated doit-il jongler avec ces différentes métriques pour opérer un choix de routage. Pour cela, gated utilise une valeur permettant de pondérer les protocoles. Cette valeur est appelée préférence. Ces préférences permettent à gated de combiner en une seule table de routage les informations de routage émanant de différentes sources. Le tableau suivant donne la liste des sources d'informations potentielles reçues par gated ainsi que les préférences associées. Ces valeurs de préférences s'échelonnent de 0 à 255. La valeur la plus petite indique la route prioritaire.

Type de route	Préférence par défaut
Route directe	0
OSPF	10
Route par défaut générée de manière interne	20
Redirection ICMP	30
Route statique	60
Protocole Hello	90
RIP	100
Route OSPF ASE	150
BGP	170
EGP	200

Préférences par défaut.

Les préférences peuvent être modifiées lors de la configuration du démon gated. Il est aussi possible d'affecter des préférences à des routes provenant :

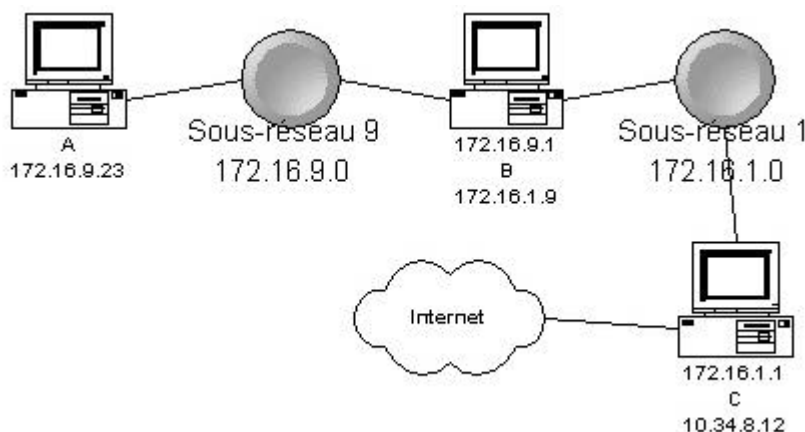
- d'une interface réseau spécifique ;
- d'un protocole spécifique ;
- d'une passerelle spécifique.

7. Configurer gated.

Le démon gated lit sa configuration à partir du fichier `/etc/gated.conf`. Les commandes qu'il contient ressemblent à du code C. Toutes les instructions sont terminées par un point-virgule. Les accolades permettent de rassembler les instructions.

7.1. Exemple de fichier `/etc/gated.conf`.

Considérons le réseau dont voici la vue schématique.



Exemple de topologie de routage.

La passerelle B interconnecte le sous-réseau 172.16.9.0 et le sous-réseau 172.16.1.0. Pour les machines du réseau 9, la passerelle se présente comme une passerelle par défaut vers le monde extérieur. Elle utilise RIP-2 et diffuse les routes sur le sous-réseau 9. Sur le sous-réseau 1, la passerelle B se présente comme étant la passerelle donnant accès au sous-réseau 9 en utilisant OSPF. La passerelle C fournit l'accès à Internet au sous-réseau 1. Puisque la passerelle C fournit cet accès, elle se présente comme étant celle par défaut aux autres systèmes du sous-réseau 1 en utilisant OSPF. Vu de l'extérieur, la passerelle C offre un accès au système autonome, elle utilise BGP pour se présenter comme étant le chemin pour atteindre les réseaux qu'elle détecte grâce à OSPF.

7.1.1. Configuration d'une machine.

La configuration de la machine A est très simple. Voici un exemple de configuration :

```
rip yes {
  nobroadcast;
  interface 172.16.9.23
    version 2
    multicast
    authentication simple "password";
};
```

L'instruction `rip yes` active RIP. Le mot-clé `nobroadcast` évite à la machine de diffuser ses mises à jour par un paquet broadcast. Cette option est activée par défaut lorsque la machine ne possède qu'une seule interface réseau.

Le membre `interface` définit les paramètres de l'interface pour RIP. Ce membre indique que les mises à jour RIP-2 seront reçues sous la forme de multicasts sur l'interface dont l'adresse IP est 172.16.9.23. De plus, les mises à jour contiendront une authentification sous la forme d'un mot de passe en clair qui sera dans notre cas `password`. Ce mot de passe permet d'accepter des mises à jour d'un diffuseur RIP mais uniquement dans le cas où le mot de passe fourni est correct.

7.1.2. Configuration d'une passerelle interne.

La configuration des passerelles est plus complexe que celle d'une banale machine. Les passerelles possèdent plusieurs interfaces et peuvent utiliser plusieurs protocoles de routage. La passerelle B utilise RIP-2 pour la gestion du sous-réseau 9 afin d'annoncer les routes aux machines UNIX. Mais elle utilise OSPF sur le sous-réseau 1 pour échanger les routes avec d'autres passerelles. Voici le contenu du fichier /etc/gated.conf de la passerelle B.

```
interfaces {
  interface 172.16.9.1 passive;
};
routerid 172.16.1.9;
rip yes {
  broadcast;
  defaultmetric 5;
  interface 172.16.9.1
    version 2
    multicast
    authentication simple "password";
};
ospf yes {
  backbone {
    authtype simple;
    interface 172.16.1.9 {
      priority 5
      authkey "pw";
    };
  };
};
};
```

L'instruction `interfaces` définit les caractéristiques du routage des interfaces réseau. Le mot clé `passive` du membre `interface` est utilisé pour créer une route statique qui ne sera pas retirée de la table de routage. Dans ce cas, la route permanente passe par une interface réseau directement reliée à la passerelle. En principe, lorsque `gated` ne reçoit plus de mises à jour de routage par une interface, il considère qu'elle fonctionne mal. En conséquence, son coût est augmenté pour ne plus router de trafic via cette interface. Dans cet exemple, la passerelle B est le seul moyen d'accès au réseau 9. Il n'y aura pas d'informations de mise à jour en provenance de cette interface. Il faut dès lors éviter que `gated` ne supprime la route vers le réseau à suite du manque de réception d'informations de mises à jour. En qualifiant la route de `passive`, rien ne permettra à `gated` d'éliminer la route.

L'instruction `routerid` définit l'identification du routeur pour le protocole OSPF, processus nécessaire car, par défaut, `gated` utilise l'adresse de la première interface. Cette instruction permet de spécifier l'interface qui utilise véritablement OSPF.

Par rapport à l'exemple précédent, l'instruction `rip yes` contient la définition d'une métrique. Le membre `defaultmetric` définit la métrique RIP à utiliser pour annoncer les routes apprises d'autres protocoles de routage. Cette passerelle utilise les protocoles RIP et OSPF. Il est nécessaire de diffuser les routes apprises par OSPF aux clients RIP. Sans la précision d'une métrique par défaut, les clients RIP ne seraient pas informés des routes transmises par OSPF.

L'instruction `ospf yes` active le protocole OSPF. La passerelle B fait partie de la zone dorsale. Pour refléter cette situation, le membre `backbone` est spécifié.

Le membre `authtype simple` indique qu'une authentification simple basée sur des mots de passe est utilisée dans la zone dorsale. Deux choix d'authentification sont possibles : `simple` ou `none`. Le paramètre `none` indique qu'aucune identification n'est effectuée. Le paramètre `simple` précise qu'un mot de passe doit être utilisé. Ce mot de passe est défini par `authkey`.

L'interface qui relie ce routeur à la zone dorsale est définie par le membre `interface`.

7.1.3. Configuration d'une passerelle externe.

La configuration de la passerelle C est plus complexe car elle utilise à la fois OSPF et BGP. Voici son fichier de configuration :

```
autonomoussystem 249;
routerid 172.16.1.1;
rip no;
bgp yes {
    preference 50 ;
    group type external peeras 164 {
        peer 10.6.0.103 ;
        peer 10.20.0.72 ;
    };
};
ospf yes {
    backbone {
        authtype simple ;
        interface 172.16.1.1 {
            priority 10 ;
            authkey "pw" ;
        } ;
    };
};
export proto bgp as 164 {
    proto direct ;
    proto ospf ;
};
export proto ospfase type 2 {
    proto bgp as 164 {
        all ;
    };
};
```

Cette configuration active à la fois le protocole BGP et le protocole OSPF. Chaque système autonome porte un identificateur sous forme d'un numéro. Le protocole BGP doit connaître cet identificateur qui est par exemple 249. Le protocole OSPF doit connaître l'identification du routeur. Cette information est fournie par la définition routerid. Notons que autonomoussystem et routerid sont des définitions et non pas des instructions. Comme dans la majorité des langages, les définitions doivent précéder les instructions.

Par défaut, le protocole RIP est lancé par gated. Dans cette configuration, RIP n'est pas utilisé, il faut donc le désactiver.

Le protocole BGP est activé au moyen de l'instruction bgp yes. Le membre preference 50 fixe la préférence des routes reçues par BGP à 50. Ce membre permet donc de modifier les préférences par défaut. Pour rappel, la préférence par défaut pour le protocole BGP est de 170.

Le membre groupe définit que la passerelle C se connecte à un système autonome portant l'identification 164. Les accords de peering permettent de spécifier à partir de quelles passerelles les informations de mises à jour seront acceptées. Dans l'exemple, nous souhaitons accepter les mises à jours en provenance des passerelles 10.6.0.103 et 10.20.0.72. Dans le cas où toutes les mises à jour provenant de n'importe quel système sont acceptées, il suffit de remplacer le mot peer par allow.

La configuration du protocole OSPF est la même que celle de l'exemple précédent sauf en ce qui concerne la priorité qui est de 10 au lieu de 5. La passerelle C doit gérer un trafic particulièrement important, une priorité moindre lui est attribuée de manière à favoriser l'utilisation de la passerelle B.

L'instruction export contrôle les routes que gated indique aux autres routeurs. La première instruction export demande à gated d'utiliser BGP (proto bgp) pour diffuser les informations de routage au système autonome 164. Les informations qui seront diffusées proviennent d'informations directes (proto direct) ou de routes obtenues à partir du

protocole OSPF (proto ospf). L'instruction export spécifie donc vers qui les informations sont transmises alors que le membre proto définit les informations qui seront diffusées.

La seconde instruction export a la signification suivante : les routes obtenues du système autonome 164 par le protocole BGP sont diffusées via OSPF. Puisque ces routes proviennent d'un système autonome externe elles sont annoncées ASE. Le protocole de diffusion est donc OSPFASE.

Les routes reçues d'un système autonome externe peuvent avoir ou non une métrique comparable au protocole OSPF. Dans notre exemple, les métriques ne sont pas comparables c'est la raison pour laquelle le paramètre type 2 est ajouté. Si les métriques étaient comparables le type 1 serait spécifié.

La seconde instruction export précise également que la source des routes provient du protocole BGP (proto bgp) en connexion avec le système autonome 164.

V. Configuration du DNS.

Le service des noms n'est pas nécessaire aux ordinateurs pour communiquer entre eux. Il s'agit en fait d'un service dont le but est de rendre le réseau plus convivial. Les logiciels peuvent se contenter des adresses IP mais les utilisateurs préfèrent utiliser des noms de machines.

1. Le fichier /etc/hosts.

Les petites infrastructures réseau ne nécessitent pas toujours la configuration d'une architecture client/serveur pour transformer les noms symboliques en adresses IP. Les systèmes UNIX disposent du fichier /etc/hosts. Ce fichier associe un ou plusieurs noms à une adresse IP. Lorsqu'un nom symbolique est employé, le système consulte, d'abord le contenu de ce fichier dans un but de recherche d'une adresse IP correspondant au nom spécifié. Voici un exemple de fichier /etc/hosts :

```
[root@P100 /etc]# cat hosts
127.0.0.1      P100.challe.be  P100      localhost.localdomain  localhost
212.68.198.209 gateway
212.68.194.200 isec1
212.68.194.201 isec2
212.68.194.202 isec3
212.68.194.203 isec4
212.68.245.203 jmb
```

Cette configuration n'est exploitable que sur la machine où réside le fichier. Cela ne constitue donc pas une base de données distribuée, exploitable par d'autres hôtes du réseau. A petite échelle, il est envisageable de configurer plusieurs ordinateurs de cette manière. A plus grande échelle, cette méthode présente le désavantage d'imposer la gestion de multiples copies rigoureusement identiques du même fichier.

2. BIND : Berkeley Internet Name Domain.

Sous UNIX, le DNS est implanté par le programme BIND. BIND est un système client/serveur. La partie client est appelée le résolveur. Elle génère des requêtes pour obtenir des informations relatives à un nom et les envoie au serveur. Le serveur DNS répond aux requêtes des clients. La partie serveur de BIND est le démon named. Le RFC 1033, Domain Administrators Operations Guide, définit la syntaxe des fichiers de configuration du serveur de noms.

BIND supporte quatre niveaux de services qui peuvent être définis dans les fichiers de configuration :

- Les systèmes résolveurs. Le résolveur est le programme qui demande des informations concernant un domaine aux serveurs de noms. La majorité des ordinateurs, appelés systèmes résolveurs, n'exploitent que le résolveur ; ils n'exécutent pas de serveur de noms. Ces systèmes nécessitent uniquement la modification du fichier /etc/resolv.conf.
- Les serveurs primaires. Le serveur primaire de noms est la source officielle de toutes les informations concernant un domaine spécifique. Les informations relatives au domaine figurent dans un fichier de zone. La configuration d'un serveur primaire implique la création d'un ensemble complexe de fichiers.
- Les serveurs secondaires. Un serveur secondaire transfère un ensemble complet d'informations sur le domaine à partir du serveur primaire. Ce transfert est appelé un transfert de fichier de zone. Un serveur secondaire conserve une copie complète de toutes les informations du domaine. Cela lui permet de répondre aux requêtes émanant des clients comme s'il était le serveur primaire. La seule différence entre un serveur primaire et un serveur secondaire est l'inexistence de fichiers de description des zones au niveau du serveur secondaire. Un serveur secondaire est également source officielle d'informations concernant un domaine spécifique. Les domaines sont généralement gérés par un serveur primaire et un ou plusieurs serveurs secondaires. De cette manière, en cas de panne du serveur primaire, les informations du domaine seront toujours disponibles. Cela contribue à la robustesse du système.
- Les serveurs caches. Un serveur de cache utilise le programme de serveur de noms mais il ne possède aucun fichier de configuration de la base de données distribuée. Un serveur cache se contente d'apprendre des informations provenant d'autres serveurs de noms. Lorsqu'un serveur cache dispose de l'information demandée par un client, aucune requête vers d'autres serveurs de noms n'est lancée. La copie locale de l'information recherchée est envoyée au client. Ce type de serveur n'est donc pas la source officielle des informations ces données étant de seconde main. L'exploitation d'un serveur cache limite le nombre de

requêtes formulées aux serveurs qui constituent la source officielle des informations. En corollaire, cette configuration diminue le trafic réseau et, par le fait même, économise la bande passante.

Un serveur peut utiliser n'importe laquelle de ces configurations ou, comme cela est souvent le cas, il peut associer plusieurs éléments de plusieurs configurations.

3. Configurer le résolveur.

La configuration du résolveur figure dans le fichier `/etc/resolv.conf` et ce fichier est lu dès l'instant où un processus utilisant le résolveur est lancé. En l'absence de ce fichier, le résolveur tente de se connecter au serveur `named` fonctionnant sur la machine. Bien que cela puisse fonctionner, cette technique n'est pas recommandable. En effet, cela configure le résolveur avec des paramètres par défaut dont les valeurs peuvent varier d'un système à l'autre.

Voici un exemple de fichier `/etc/resolv.conf` :

```
[root@P200 /etc]# cat resolv.conf
domain challe.be
search brutele.be challe.yi.org
nameserver 192.168.1.1
nameserver 192.168.1.50
```

Il existe certaines variations dépendant du système utilisé mais voici la signification des entrées les plus couramment acceptées par la plupart des systèmes :

- `nameserver`. Les entrées `nameserver` indiquent les adresses IP des serveurs auxquelles le résolveur envoie ses requêtes pour obtenir des informations. Les serveurs de noms sont interrogés dans l'ordre de leur apparition dans le fichier. Si aucune réponse n'est reçue d'un serveur, le serveur suivant dans la liste est alors consulté jusqu'à ce que le nombre maximum de serveurs soit essayé. Dans la plupart des configurations, `BIND` est capable de gérer trois serveurs de noms. Si aucune entrée `nameserver` ne se trouve dans le fichier `/etc/resolv.conf` ou si ce fichier n'existe pas, toutes les requêtes sont envoyées au serveur de la machine. Toutefois, si le fichier `/etc/resolv.conf` contient une entrée `nameserver`, le serveur local n'est pas interrogé sauf si une entrée `nameserver` pointe sur elle. Dans le cas où le serveur de nom est implanté sur la machine où le résolveur est configuré, il est préférable de ne pas employer l'adresse loopback de la machine cela pouvant poser des problèmes sur certaines versions d'UNIX.
- `domain`. L'entrée `domain` définit le nom de domaine par défaut. Le résolveur ajoute le nom de domaine par défaut à n'importe quel nom de machine qui ne contient pas de point. Le résolveur utilise ensuite le nom expansé dans la requête qu'il envoie au serveur de noms. Dans l'exemple ci-dessus, une requête `P100` sera complétée en `P100.challe.be`.
- `search`. L'entrée `search` définit une liste de domaines recherchés lorsqu'un nom ne contient pas de point. Cela signifie que les entrées `domain` et `search` sont redondantes. En réalité, lorsqu'un nom ne contient pas de point, l'entrée `domain` est en premier lieu utilisée pour compléter le nom. Si la recherche échoue, ce sont les entrées `search` qui sont exploitées dans l'ordre de leurs apparitions pour tenter de compléter le nom. Il n'est pas recommandé d'utiliser à la fois `domain` et `search` dans une même configuration.

4. Initialisation des données du DNS.

La version de la table `/etc/hosts` pour le DNS est composée de deux fichiers. L'un d'eux relie les noms d'hôtes à leur adresse, un autre relie les adresses à leur nom d'hôte. La correspondance adresse-nom est appelée correspondance inverse. Ces deux fichiers constituent la base de données du DNS. Pour relier entre eux tous les fichiers de base de données, un serveur de noms doit posséder un fichier d'initialisation qui est généralement `/etc/named.conf`

4.1. Les fichiers de la base de données.

Les recherches dans la base de données du DNS ignorent la casse des caractères. Les informations peuvent être saisies en majuscules, en minuscules ou dans n'importe quel mixage des deux.

4.1.1. Les enregistrements SOA (Start Of Authority).

Le premier enregistrement de chacun des fichiers est l'enregistrement SOA. Il indique que le serveur de noms est la meilleure source d'informations pour les données de cette zone. Considérons que le serveur de noms a l'autorité sur la zone challe.be

Voici l'enregistrement SOA du fichier de zone challe.be :

```
[root@P200 named]# cat challe.be
challe.be.                IN          SOA      P200.challe.be.        root.challe.be. (
                           1998120701; Serial
                           28800       ; Refresh
                           14400       ; Retry
                           3600000    ; Expire
                           86400    ) ; Minimum
```

Le nom challe.be. doit commencer à la première colonne du fichier. L'administrateur doit s'assurer de ce que le nom se termine bien par un point de manière à préciser un nom complet depuis la racine de l'arbre du DNS.

Le mot-clé IN signale que l'enregistrement est dans la classe internet. Il existe d'autres classes, mais aucune n'est actuellement très utilisée. Le champ classe est facultatif ; s'il est omis le système supposera qu'il s'agit de la classe IN.

Le premier nom après SOA (challe.be) est le nom du serveur-maitre primaire. Le second nom est l'adresse email de l'administrateur du domaine. Pour lire correctement cette adresse mail il faut remplacer le premier point par le symbole @. Les serveurs de noms n'utilisent jamais cette adresse email ; elle est là à titre indicatif.

Les parenthèses permettent de présenter l'enregistrement SOA sur plusieurs lignes. La plupart des champs à l'intérieur de l'enregistrement SOA sont destinés aux serveurs secondaires.

Le même enregistrement SOA figure au début des fichiers de résolution inverse des noms.

4.1.2. Les enregistrements NS (Name Server).

Les enregistrements NS permettent de définir les serveurs de noms. Dans notre exemple, il y a un serveur de noms implanté dans l'hôte P100.

Voici un extrait du fichier de base de données contenant un enregistrement NS.

```
[root@P200 named]# cat challe.be
challe.be.                IN          SOA      P200.challe.be.        root.challe.be. (
                           1998120701; Serial
                           28800       ; Refresh
                           14400       ; Retry
                           3600000    ; Expire
                           86400    ) ; Minimum
challe.be.                IN          NS       P200.challe.be.
```

Les fichiers de résolution inverse contiennent également un enregistrement NS.

4.1.3. Les enregistrements d'adresse et d'alias.

Il faut établir la correspondance entre les noms et les adresses IP. Il faut ajouter les correspondances au fichier challe.be.

Voici un extrait du fichier challe.be :

```
[root@P200 named]# cat challe.be
challe.be.                IN      SOA    P200.challe.be.      root.challe.be. (
                            1998120701; Serial
                            28800    ; Refresh
                            14400    ; Retry
                            3600000  ; Expire
                            86400 ) ; Minimum

challe.be.                IN      NS     P200.challe.be.

P100.challe.be.          IN      A      192.168.1.1
macii.challe.be.         IN      A      192.168.1.2
powerpc.challe.be.       IN      A      192.168.1.3
fuji.challe.be.          IN      A      192.168.1.4
p166.challe.be.          IN      A      192.168.1.5
P200.challe.be.          IN      A      192.168.1.10
P133.challe.be.          IN      A      192.168.1.20
P166PLUS.challe.be.     IN      A      192.168.1.30
P2450.challe.be.         IN      A      192.168.1.40
P2300.challe.be.         IN      A      192.168.1.50
se30.challe.be.          IN      A      192.168.1.60
P3450.challe.be.         IN      A      192.168.1.70
P2366.challe.be.         IN      A      192.168.1.80
modem.challe.be.         IN      A      192.168.1.100
HP4.challe.be.           IN      A      192.168.1.192

www.challe.be.           IN      CNAME  P100.challe.be.
ftp.challe.be.           IN      CNAME  P100.challe.be.
irc.challe.be.           IN      CNAME  P100.challe.be.
mail.challe.be.          IN      CNAME  P100.challe.be.
pop.challe.be.           IN      CNAME  P100.challe.be.
news.challe.be.          IN      CNAME  P100.challe.be.
```

Le premier bloc de lignes fait correspondre une adresse IP à chaque nom (le A signifie adresse). Un même nom peut apparaître dans plusieurs enregistrements et par le fait même être associé à plus d'une adresse IP. A la différence de la recherche dans la table /etc/hosts, une recherche dans le DNS peut conduire au renvoi de plusieurs adresses par nom. Si le client et le serveur de noms sont sur le même réseau, certains serveurs indiquent en premier lieu l'adresse « la plus proche » du client.

Le dernier bloc de lignes est la table d'alias. Ces enregistrements sont définis par la clause CNAME (Canonical Name). Lorsqu'un serveur recherche un nom et trouve un enregistrement CNAME, il remplace le nom par le nom canonique et procède à une nouvelle recherche. Par exemple : si lors de la recherche de www, le serveur trouve un enregistrement CNAME qui désigne P100, il recherche alors P100 et renvoie l'adresse correspondante.

4.1.4. Les enregistrements PTR (Pointer).

Le serveur DNS doit également contenir une correspondance entre les adresses IP et les noms. Il s'agit de la résolution inverse. Voici un exemple de table inverse pour le réseau 192.168.1 :

```
[root@P200 named]# cat 1.168.192
1.168.192.in-addr.arpa.  IN      SOA    P200.challe.be.      root.challe.be
(
                            1998120701 ; Serial
                            28800    ; Refresh
                            14400    ; Retry
                            3600000  ; Expire
                            86400 ) ; Minimum

1.168.192.in-addr.arpa.  NS     P200.challe.be.

1.1.168.192.in-addr.arpa. IN     PTR    P100.challe.be.
```

```

2.1.168.192.in-addr.arpa.      IN      PTR      macii.challe.be.
3.1.168.192.in-addr.arpa.      IN      PTR      powerpc.challe.be.
4.1.168.192.in-addr.arpa.      IN      PTR      fuji.challe.be.
5.1.168.192.in-addr.arpa.      IN      PTR      P166.challe.be.
10.1.168.192.in-addr.arpa.     IN      PTR      P200.challe.be.
20.1.168.192.in-addr.arpa.     IN      PTR      P133.challe.be.
30.1.168.192.in-addr.arpa.     IN      PTR      P166PLUS.challe.be.
40.1.168.192.in-addr.arpa.     IN      PTR      P2450.challe.be.
50.1.168.192.in-addr.arpa.     IN      PTR      P2300.challe.be.
60.1.168.192.in-addr.arpa.     IN      PTR      se30.challe.be.
70.1.168.192.in-addr.arpa.     IN      PTR      P3450.challe.be.
80.1.168.192.in-addr.arpa.     IN      PTR      P2366.challe.be.
100.1.168.192.in-addr.arpa.    IN      PTR      modem.challe.be.
192.1.168.192.in-addr.arpa.    IN      PTR      HP4.challe.be.

```

Le serveur de noms a besoin d'un autre fichier de correspondance inverse pour le réseau loopback. Ce réseau est utilisé par un hôte pour communiquer avec lui-même. Le numéro de ce réseau est toujours 127.0.0 et l'adresse de l'hôte est toujours 127.0.0.1. Voici le fichier de résolution inverse pour l'adresse loopback :

```

[root@P200 named]# cat 0.0.127
0.0.127.in-addr.arpa.          IN      SOA      P200.challe.be.      root.challe.be (
                                1998120701 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000   ; Expire
                                86400    ) ; Minimum

0.0.127.in-addr.arpa.          NS      localhost.

```

Ce fichier, en apparence anodin, est indispensable aux serveurs de noms. Aucun serveur n'a reçu la délégation pour le réseau 127 et malgré son utilisation réelle, chaque serveur doit en être responsable pour lui-même. L'omission de ce fichier permettra au serveur de fonctionner mais la recherche de 127.0.0.1 échouera car le serveur racine ne peut réaliser la correspondance 127.0.0.1 à l'hôte local.

4.1.5. Les données de la zone racine.

Le serveur DNS a aussi besoin de connaître l'emplacement des serveurs de noms de la racine. L'objectif du fichier `named.ca` consiste à définir l'emplacement des serveurs de la racine. Généralement, les configurations de serveurs de noms possèdent déjà un tel fichier. Voici un exemple de ce fichier :

```

[root@P100 named]# cat named.ca
;      This file holds the information on root name servers needed to
;      initialize cache of Internet domain name servers
;      (e.g. reference this file in the "cache . <file>"
;      configuration file of BIND domain name servers).
;
;      This file is made available by InterNIC registration services
;      under anonymous FTP as
;      file          /domain/named.root
;      on server     FTP.RS.INTERNIC.NET
;      -OR- under Gopher at  RS.INTERNIC.NET
;      under menu    InterNIC Registration Services (NSI)
;      submenu       InterNIC Registration Archives
;      file          named.root
;
;      last update:   Aug 22, 1997
;      related version of root zone:  1997082200
;
;
; formerly NS.INTERNIC.NET
;

```

```

.                3600000  IN  NS  A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.  3600000  A    198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                3600000  NS   B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.  3600000  A    128.9.0.107
;
; formerly C.PSI.NET
;
.                3600000  NS   C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.  3600000  A    192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                3600000  NS   D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.  3600000  A    128.8.10.90
;
; formerly NS.NASA.GOV
;
.                3600000  NS   E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.  3600000  A    192.203.230.10
;
; formerly NS.ISC.ORG
;
.                3600000  NS   F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.  3600000  A    192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                3600000  NS   G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.  3600000  A    192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                3600000  NS   H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.  3600000  A    128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                3600000  NS   I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.  3600000  A    192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                3600000  NS   J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.  3600000  A    198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                3600000  NS   K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.  3600000  A    193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                3600000  NS   L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.  3600000  A    198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                3600000  NS   M.ROOT-SERVERS.NET.

```

```
M.ROOT-SERVERS.NET.      3600000      A      202.12.27.33
; End of File
```

Le nom de domaine « . » est celui du domaine racine. Cette liste de serveur étant susceptible d'évoluer, il est conseillé de télécharger régulièrement la dernière version de ce fichier. Certains serveurs de noms sont capables de télécharger automatiquement les mises à jour de ce fichier. Néanmoins, cette fonctionnalité ne s'est pas montrée à la hauteur des exigences.

4.2. Le fichier de configuration de BIND.

Le fichier de configuration est exploité afin de stipuler à BIND quels fichiers de base de données il doit utiliser. Les fichiers de base de données contiennent des informations dont le format est décrit dans les standards du DNS. Par contre, le fichier de configuration est spécifique à BIND et n'est pas défini dans les RFC. Actuellement, deux versions de BIND existent : la version 4 et la version 8. Des problèmes de sécurité ont rendu la version 4 de BIND dépassée. Il est fortement recommandé d'installer et d'exploiter la dernière version du serveur de noms.

Voici un exemple de fichier de configuration de la version 8 de BIND :

```
[root@P200 /etc]# cat named.conf
options {
    directory "/var/named";
};

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

zone "localhost" IN {
    type master;
    file "localhost";
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "0.0.127";
};

zone "challe.be" IN {
    type master;
    file "challe.be";
};

zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "1.168.192";
};
```

Habituellement, les fichiers de configuration contiennent une ligne indiquant le répertoire dans lequel sont situés les fichiers de données (directory "/var/named"). Le serveur de noms l'utilise comme répertoire de travail, ce qui permet d'utiliser des chemins relatifs pour accéder aux fichiers de données.

Sur un serveur-maître primaire, le fichier de configuration contient un enregistrement zone pour chaque fichier de données à lire. Le mot zone est suivi du nom du domaine à gérer. Le mot IN indique que la classe est internet. Le type master permet d'informer le démon BIND de ce que le serveur est primaire. Le dernier paramètre indique l'emplacement du fichier de données au moyen d'un chemin relatif.

4.3. Les abréviations.

Jusqu'à présent, les fichiers de données du serveur de noms ont été pleinement qualifiés car aucune forme raccourcie pouvant prêter à confusion n'a été employée.

L'information placée entre guillemets dans l'enregistrement zone indique le nom de domaine. Ce nom est la clé de la plupart des abréviations utilisables. Ce domaine est l'origine de toutes les données du fichier de base de données. Cette

origine est ajoutée à tous les noms ne se terminant pas par un point. Cela permet de modifier les fichier challe.be et 1.168.192 de la manière suivante :

```
[root@P200 named]# cat challe.be
challe.be.                IN      SOA     P200.challe.be.      root.challe.be. (
                           1998120701; Serial
                           28800      ; Refresh
                           14400      ; Retry
                           3600000    ; Expire
                           86400 )    ; Minimum

                           IN      NS      P200

P100                       IN      A       192.168.1.1
macii                      IN      A       192.168.1.2
powerpc                    IN      A       192.168.1.3
fuji                       IN      A       192.168.1.4
p166                       IN      A       192.168.1.5
P200                       IN      A       192.168.1.10
P133                       IN      A       192.168.1.20
P166PLUS                   IN      A       192.168.1.30
P2450                      IN      A       192.168.1.40
P2300                      IN      A       192.168.1.50
se30                       IN      A       192.168.1.60
P3450                      IN      A       192.168.1.70
P2366                      IN      A       192.168.1.80
modem                      IN      A       192.168.1.100
HP4                        IN      A       192.168.1.192

www                        IN      CNAME   P100
ftp                        IN      CNAME   P100
irc                        IN      CNAME   P100
mail                       IN      CNAME   P100
pop                        IN      CNAME   P100
news                       IN      CNAME   P100
[root@P200 named]# cat 1.168.192
1.168.192.in-addr.arpa.   IN      SOA     P200.challe.be.      root.challe.be
(
                           1998120701 ; Serial
                           28800      ; Refresh
                           14400      ; Retry
                           3600000    ; Expire
                           86400 )    ; Minimum

                           NS      P200.challe.be.

1      IN      PTR     P100.challe.be.
2      IN      PTR     macii.challe.be.
3      IN      PTR     powerpc.challe.be.
4      IN      PTR     fuji.challe.be.
5      IN      PTR     P166.challe.be.
10     IN      PTR     P200.challe.be.
20     IN      PTR     P133.challe.be.
30     IN      PTR     P166PLUS.challe.be.
40     IN      PTR     P2450.challe.be.
50     IN      PTR     P2300.challe.be.
60     IN      PTR     se30.challe.be.
70     IN      PTR     P3450.challe.be.
80     IN      PTR     P2366.challe.be.
100    IN      PTR     modem.challe.be.
192    IN      PTR     HP4.challe.be.
```


Si le nom de domaine est le même que celui de l'origine précisée dans le fichier named.conf, le nom peut être remplacé par le caractère @. Ceci est généralement le cas dans le l'enregistrement SOA. Notons que dans le cas de la clause NS, le caractère @ est implicite.

Voici la nouvelle version des fichiers de données :

```
[root@P200 named]# cat challe.be
@           IN           SOA    P200.challe.be.      root.challe.be. (
                                1998120701; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

           IN           NS      P200

P100       IN           A       192.168.1.1
macii      IN           A       192.168.1.2
powerpc    IN           A       192.168.1.3
fuji       IN           A       192.168.1.4
p166       IN           A       192.168.1.5
P200       IN           A       192.168.1.10
P133       IN           A       192.168.1.20
P166PLUS   IN           A       192.168.1.30
P2450      IN           A       192.168.1.40
P2300      IN           A       192.168.1.50
se30       IN           A       192.168.1.60
P3450      IN           A       192.168.1.70
P2366      IN           A       192.168.1.80
modem      IN           A       192.168.1.100
HP4        IN           A       192.168.1.192
www        IN           CNAME  P100
ftp        IN           CNAME  P100
irc        IN           CNAME  P100
mail       IN           CNAME  P100
pop        IN           CNAME  P100
news       IN           CNAME  P100
[root@P200 named]# cat 1.168.192
@           IN           SOA    P200.challe.be.      root.challe.be (
                                1998120701 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

           NS           P200.challe.be.

1          IN           PTR    P100.challe.be.
2          IN           PTR    macii.challe.be.
3          IN           PTR    powerpc.challe.be.
4          IN           PTR    fuji.challe.be.
5          IN           PTR    P166.challe.be.
10         IN           PTR    P200.challe.be.
20         IN           PTR    P133.challe.be.
30         IN           PTR    P166PLUS.challe.be.
40         IN           PTR    P2450.challe.be.
50         IN           PTR    P2300.challe.be.
60         IN           PTR    se30.challe.be.
70         IN           PTR    P3450.challe.be.
80         IN           PTR    P2366.challe.be.
100        IN           PTR    modem.challe.be.
192        IN           PTR    HP4.challe.be.
```

4.4. Démarrage d'un serveur-maître primaire.

Le démarrage du serveur de noms nécessite l'utilisation du compte root car BIND exploite un port privilégié. Le lancement du serveur nécessite la dactylographie de la commande suivante :

```
[root@P200 named]# /etc/rc.d/init.d/named start
```

Cette commande est uniquement valable sur Linux RedHat.

Dès que le serveur de noms est lancé, il est possible de lui envoyer des requêtes au moyen de la commande nslookup. Par exemple, la recherche de l'adresse IP de l'hôte P2300 est réalisée au moyen de la commande suivante :

```
[root@P200 named]# nslookup p2300
Server: P200.challe.be
Address: 192.168.1.10

Name: p2300.challe.be
Address: 192.168.1.50
```

Si la recherche échoue, la commande nslookup se termine avec un message d'erreur, comme le montre l'illustration suivante :

```
[root@P200 named]# nslookup p1597
Server: P200.challe.be
Address: 192.168.1.10

*** P200.challe.be can't find p1597: Non-existent host/domain
```

Lorsqu'une adresse IP est fournie, la commande nslookup sait qu'elle doit rechercher un enregistrement PTR dans la table de résolution inverse. Voici un exemple d'interrogation du domaine in-addr.arpa :

```
[root@P200 named]# nslookup 192.168.1.192
Server: P200.challe.be
Address: 192.168.1.10

Name: HP4.challe.be
Address: 192.168.1.192
```

Le serveur de noms actuellement configuré est capable d'interroger les serveurs racine lorsqu'une requête concerne un autre domaine. Voici un exemple d'une telle interrogation :

```
[root@P200 named]# nslookup www.redhat.com
Server: P200.challe.be
Address: 192.168.1.10

Name: www.redhat.com
Addresses: 216.148.218.197, 216.148.218.195
```

Le serveur de noms peut être interrogé à partir d'une autre machine. Pour cela, la commande nslookup doit préciser le nom de l'hôte dont l'adresse est recherchée ainsi que le nom du serveur de noms à interroger. Voici un exemple de cette situation :

```
[jfc@P100 jfc]$ nslookup hp4.challe.be p200
Server: P200.challe.be
Address: 192.168.1.10

Name: hp4.challe.be
Address: 192.168.1.192
```

4.5. Démarrage d'un serveur-esclave.

Pour la robustesse du système, il est nécessaire d'installer un second serveur de noms. Si un domaine est géré par un seul serveur de noms et qu'il vienne à s'arrêter, plus personne ne pourra effectuer de recherche sur le domaine. Un second serveur de noms partage la charge avec le premier ou la reçoit en totalité en cas de défaillance du premier.

Un serveur-esclave peut être la réplique exacte d'un serveur-maître. Cela oblige l'administrateur à gérer deux bases de données qui doivent être rigoureusement identiques. Une autre méthode de configuration consiste à stipuler que le serveur-esclave obtiendra ses données automatiquement du serveur-maître. Pour effectuer cette configuration, il suffit d'indiquer dans le fichier /etc/named.conf que le serveur est de type slave pour une certaine zone. En plus de cela, il faut indiquer l'adresse IP du serveur-maître qui constitue la source d'information.

Bien qu'un serveur-esclave n'aie pas nécessairement besoin de fichiers pour mémoriser les bases de données, il est préférable de stipuler un nom de fichier dans le champ file. De cette manière, lors de son premier lancement le serveur-esclave placera une copie des informations provenant du maître dans ces fichiers. En cas de défaillance du maître, l'esclave possèdera toujours une copie des informations.

Voici un exemple de configuration d'un serveur-esclave :

```
[root@P2450 /etc]# cat named.conf
options {
    directory "/var/named/slave";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "challe.be." {
    type slave;
    file "challe.be";
    masters { 192.168.1.1; };
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "1.168.192.in-addr.arpa" {
    type slave;
    file "1.168.192";
    masters { 192.168.1.1; };
};
```

4.6. Les champs de l'enregistrement SOA.

Voici un enregistrement SOA :

```
[root@P100 named]# cat challe.be
@      IN      SOA      challe.be. root.localhost. (
                                1998120701 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                604800     ; Expire
                                86400     ) ; TTL
```

Jusqu'à présent, la signification des valeurs placées entre parenthèses n'a pas encore été donnée.

Le numéro de série s'applique à toutes les données de la zone. De nombreux administrateurs utilisent la date du jour suivi d'un numéro de séquence. Le format est donc AAAAMMJJSS. Chaque fois qu'une mise à jour du fichier est réalisée, il est important d'incrémenter le numéro de série.

Lorsqu'un esclave demande les données de la zone à un serveur maître, il fournit d'abord son numéro de série. Si celui de l'esclave est inférieur à celui du maître, les données de l'esclave sont obsolètes. Dans ce cas, l'esclave télécharge une nouvelle copie de la zone. Quand un serveur-esclave démarre et qu'il ne possède pas de fichier de sauvegarde, il télécharge toujours la zone.

Les quatre champs suivants indiquent des intervalles de temps en seconde :

- **Rafraîchissement (refresh).** L'intervalle de rafraîchissement indique la périodicité de test de validité de ses données à l'esclave qui, lui, demande l'enregistrement SOA de la zone à chaque intervalle de rafraîchissement. Dans l'exemple, cet intervalle est de 8 heures. En fonction du niveau de service souhaité et de la fréquence de modification des données, cette valeur peut-être augmentée ou diminuée.
- **Nouvel essai (retry).** Si le serveur-esclave n'arrive pas à contacter le serveur-maître au bout de la période de rafraîchissement, il essaie de le contacter selon la périodicité de nouvel essai. Cette périodicité est généralement plus courte que la périodicité de rafraîchissement mais ce n'est pas une obligation à respecter. Dans l'exemple, cet intervalle est de 4 heures.
- **Obsolescence (expire).** Si l'esclave n'arrive pas à contacter le serveur-maître avant l'obsolescence, l'esclave arrête sa fonction de serveur en cessant de répondre aux requêtes, les données étant considérées comme trop anciennes pour être valables. Une obsolescence d'une semaine est une valeur courante. Cette périodicité doit être impérativement supérieure à celles de rafraîchissement et de nouvel essai.
- **TTL (Time To Live).** La TTL est la durée de validité des données. Le serveur de noms renvoie cette valeur avec ses réponses aux demandes de résolution des noms. Cette valeur impose aux autres serveurs de limiter la conservation des données dans la mémoire-cache. Si les données évoluent peu, une valeur de plusieurs jours est permise. Par contre, si les données évoluent rapidement, une valeur de moins d'une heure peut être employée. Néanmoins, une si faible valeur n'est pas recommandée car elle engendre du trafic réseau.

En général, des durées plus longues diminuent la charge du serveur de noms et augmentent le délai de propagation des mises à jour. Inversement, des durées plus courtes augmentent la charge du serveur de noms et diminuent le délai de propagation. La RFC 1537 recommande les valeurs suivantes :

86400	Rafraîchissement après 24 heures
7200	Nouvel essai après 2 heures
2592000	Obsolescence après 30 jours
345600	TTL minimale de 4 jours

4.7. Gestion de plusieurs domaines.

Un serveur de noms est capable de gérer plusieurs zones simultanément. Il suffit pour cela d'ajouter autant d'enregistrements de type zone que de zones à gérer. Il reste ensuite à configurer les fichiers de la base de données en conséquence.

5. Gestion des sous-domaines.

La délégation d'un sous-domaine consiste à créer un serveur de noms pour le sous-domaine et à placer, un lien vers le domaine enfant au niveau du domaine parent. La configuration du sous-domaine suit exactement la procédure qui a été décrite dans les paragraphes précédents de ce chapitre. Il faut donc se concentrer sur la délégation d'un sous-domaine au niveau du domaine parent. Le serveur de noms BIND permet d'effectuer très simplement cette opération. Il suffit d'ajouter les lignes suivantes au fichier /etc/named.conf du domaine parent :

```
zone "bureau1.challe.be" {
    type stub;
    file "db.bureau1";
    masters { 192.168.1.40; };
};
```

Le type stub informe le serveur de noms du domaine parent de ce qu'une délégation a lieu pour la zone bureau1.challe.be. Il faut alors spécifier l'adresse IP du serveur-enfant au moyen du champ masters.

La délégation de la résolution inverse des noms est beaucoup plus complexe. En substance, il faut découper le réseau en sous-réseaux (au niveau des adresses IP) et déléguer chaque sous-réseau.

6. La sécurité.

L'un des moyens permettant d'augmenter la sécurité des serveurs de noms consiste à installer la version de BIND la plus récente. La version 8 de BIND est plus robuste que la version 4 en raison des nouvelles fonctions de sécurité qu'elle apporte. Néanmoins, cela ne suffit pas car de nouvelles attaques apparaissent tous les jours.

6.1. Restriction des requêtes.

Jusqu'à la version 4 de BIND, un administrateur ne pouvait pas effectuer de contrôle d'accès à son serveur. Cette approche correspond à la philosophie d'Internet consistant à rendre les informations aisément accessibles. L'évolution du comportement des utilisateurs oblige les administrateurs à protéger leurs systèmes en masquant certaines parties de l'espace de nommage tout en rendant d'autres parties disponibles.

L'option `allow-query` permet de configurer un contrôle d'accès basé sur l'adresse IP du client à l'origine de la requête. Ce contrôle peut concerner soit une zone particulière, soit l'ensemble des requêtes reçues. La liste d'accès stipule quelles adresses IP sont habilitées à envoyer des requêtes au serveur.

6.1.1. Restriction sur toutes les requêtes.

Pour indiquer à un serveur de noms de ne répondre qu'aux requêtes provenant du réseau 192.168.1.0 ainsi qu'à la machine dont l'adresse IP est 212.68.194.200, il suffit d'ajouter ces informations à la liste globale des permissions. Voici un exemple de permissions :

```
[root@P100 /etc]# cat named.conf
options {
    directory "/var/named";
    allow-query {192.168.1/24; 212.68.194.200; };
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "challe.be" {
    type master;
    file "challe.be";
};
zone "challe.yi.org" {
    type master;
    file "challe.yi.org";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192";
};
```

6.1.2. Restriction des requêtes concernant une zone.

Le serveur de noms BIND permet également l'application d'une liste d'accès à une zone particulière. Dans ce cas, il suffit d'utiliser la clause `allow-query` dans la définition de la structure de zone à protéger.

```
[root@P100 /etc]# cat named.conf
options {
    directory "/var/named";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "challe.be" {
    type master;
    file "challe.be";
    allow-query {192.168.1/24; };
};
zone "challe.yi.org" {
    type master;
    file "challe.yi.org";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192";
    allow-query {192.168.1/24; };
};
```

Tout serveur, qu'il soit maître ou esclave, peut appliquer un contrôle d'accès à une zone. Les listes spécifiques à une zone sont prépondérantes sur la liste globale. En l'absence d'une liste spécifique à une zone, c'est la liste globale qui est utilisée.

6.2. Contrôle des transferts de zones.

Il est très important de vérifier que seuls les réels serveurs -esclaves d'une zone peuvent demander un transfert de zone. Les utilisateurs d'hôtes distants qui peuvent interroger un serveur ne peuvent rechercher que des données concernant des hôtes dont ils connaissent le nom. Ils ne peuvent opérer que sur une seule à la fois. Les utilisateurs qui peuvent demander un transfert de zone peuvent obtenir la liste complète des hôtes d'un domaine.

La clause `allow-transfer` permet aux administrateurs d'appliquer une liste d'accès aux transferts de zones. Cette clause est apte à limiter les transferts d'une zone spécifique ou bien limiter les transferts de toutes les zones.

6.2.1. Limitation globale du transfert.

La limitation globale du transfert de zone est effectuée en plaçant la clause `allow-transfer` dans la structure option du fichier `/etc/named.conf`. Voici la structure de ce fichier lorsque l'administrateur accepte de transférer toutes les zones au serveur-esclave dont l'adresse IP est 192.168.1.40 :

```
[root@P100 /etc]# cat named.conf
options {
    directory "/var/named";
    allow-transfer { 192.168.1.40; };
};
zone "." {
    type hint;
    file "named.ca";
};
```

```
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "challe.be" {
    type master;
    file "challe.be";
};
zone "challe.yi.org" {
    type master;
    file "challe.yi.org";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192";
};
```

6.2.2. Limitation du transfert à une zone.

Afin de rendre plus fin le contrôle de transfert de zone, la clause `allow-transfer` sera placée dans une structure de zone. Dans l'exemple qui suit, le transfert de la zone `challe.be` sera acceptée uniquement dans le cas où le serveur-esclave a pour adresse IP `192.168.1.40`.

```
[root@P100 /etc]# cat named.conf
options {
    directory "/var/named";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "challe.be" {
    type master;
    file "challe.be";
    allow-transfer { 192.168.1.40; };
};
zone "challe.yi.org" {
    type master;
    file "challe.yi.org";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192";
};
```

Les transferts de zones peuvent également s'effectuer au départ d'un serveur-esclave. Il ne faut donc pas oublier de sécuriser également ces serveurs.

Dans le cas où une liste globale de permissions de transfert de zone est spécifiée ainsi que des listes locales, la prépondérance est donnée aux listes locales.

Il est concevable qu'un administrateur configure un serveur de noms de manière telle qu'il accepte les transferts de zone vers toutes les machines du sous-réseau `192.168.1` mais que seul l'hôte `192.168.1.40` puisse demander un transfert de la zone `challe.be`. Cette configuration nécessite deux listes de permissions :

- une globale ;
- une locale.

La liste globale permet d'accepter les transferts vers les machines du sous-réseau 192.16.1 alors que la liste locale permet uniquement à la machine 192.168.1.40 d'effectuer un transfert de la zone challe.be. Voici un exemple d'une telle configuration :

```
[root@P100 /etc]# cat named.conf
options {
    directory "/var/named";
    allow-transfer { 192.168.1/24; };
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "challe.be" {
    type master;
    file "challe.be";
    allow-transfer { 192.168.1.40; };
};
zone "challe.yi.org" {
    type master;
    file "challe.yi.org";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192";
};
```

6.3. Exécution de BIND par un utilisateur sans privilège.

L'exécution de BIND par un utilisateur privilégié tel que root pourrait être dangereuse et c'est ce qui se passe par défaut. Si un pirate trouve une faille dans le serveur de noms lui permettant de lire ou d'écrire des fichiers, il obtiendrait un accès privilégié au système de fichiers. S'il peut exploiter une imperfection lui permettant d'exécuter des commandes, il le fera sous l'identité de l'utilisateur privilégié.

Le serveur BIND a été conçu de manière à changer l'identité de l'utilisateur et du groupe pour le compte de qui il s'exécute. Le serveur de noms peut donc être exécuté avec moins de privilèges. De cette manière, si un pirate utilise BIND pour s'introduire dans le système, il n'obtiendra pas les droits de root.

Voici quelques options à employer lors du lancement du serveur BIND :

- -u : impose l'identité de l'utilisateur qui exécute BIND à l'issue de son démarrage (named -u named).
- -g : impose l'identité du groupe qui exécute BIND à l'issue de son démarrage (named -g other). Si l'option -u n'est pas utilisée conjointement avec l'option -g, le serveur utilise le groupe primaire de l'utilisateur spécifié.
- -t : indique la racine du système de fichiers vu par BIND.

La meilleure manière de protéger le système consiste à créer un nouvel utilisateur sans privilège, comme par exemple, named et d'exécuter BIND sous cette identité. Puisque le serveur de noms lit le fichier named.conf avant d'abandonner les privilèges de root, il n'est pas nécessaire de changer les autorisations d'accès sur ce fichier. Par contre, il faut modifier les permissions et les propriétés des fichiers de données afin que l'utilisateur sans privilège puisse les lire.

7. La commande nslookup.

L'outil nslookup permet de vérifier de manière efficace la configuration des serveurs de noms. La commande nslookup génère des requêtes soit comme un resolver, soit comme un serveur de noms. Cependant, nslookup n'utilise pas les

fonctions du resolver mais ses propres procédures pour interroger les serveurs de noms. Aussi son comportement bien que très similaire à celui d'un resolver diffère-t-il légèrement.

La commande nslookup peut être utilisée soit de manière interactive, soit de manière non interactive. Le lancement d'une session interactive exige simplement le lancement sans paramètre de la commande nslookup. Par contre l'exécution de nslookup en mode non interactif impose d'exploiter nslookup avec des paramètres.

Voici un exemple de session interactive :

```
[root@P100 /root]# nslookup
Default Server:  P100.challe.be
Address:  192.168.1.1

>
```

Pour obtenir de l'aide, il suffit de taper help ou « ? ». Pour quitter nslookup, il faut taper ctrl-D.

En mode non interactif, il faut donner un nom à rechercher sur la ligne de commande :

```
[root@P100 /root]# nslookup p2300.challe.be
Server:  P100.challe.be
Address:  192.168.1.1

Name:    p2300.challe.be
Address:  192.168.1.50
```

En mode interactif dispose d'un ensemble d'options qui peuvent être visualisées au moyen de la commande set all :

```
[root@P100 /root]# nslookup
Default Server:  P100.challe.be
Address:  192.168.1.1

> set all
Default Server:  P100.challe.be
Address:  192.168.1.1

Set options:
  nodebug          defname          search           recurse
  nod2             novc             noignoretc      port=53
  querytype=A     class=IN        timeout=5       retry=2
  root=a.root-servers.net.
  domain=challe.be
  srchlist=challe.be

>
```

La commande nslookup informe l'utilisateur de ce que le serveur par défaut est l'hôte p100.challe.be dont l'adresse IP est 192.168.1.1. Cela signifie que toutes les requêtes générées par nslookup seront envoyées à ce serveur.

Les options se répartissent en deux catégories :

- les options booléennes ;
- les options à valeur.

Les options qui ne sont pas suivies d'un signe égal sont booléennes ; elles sont soit inhibées, soit activées. Pour inhiber une option booléenne, il suffit d'ajouter le préfixe « no » au nom de l'option. Dans l'exemple ci-dessus, nodebug signifie que le débogage est inhibé.

La modification d'une option de nslookup varie selon que nslookup est utilisé en mode interactif ou non. Dans une session interactive, la commande set permet de modifier une option : set debug. En ligne de commande, le mot set est remplacé par le caractère tiret : nslookup -debug brutele.be.

7.1. Recherche de différents types de données.

Par défaut, nslookup recherche l'adresse correspondant à un nom ou le nom correspondant à une adresse. D'autres types de recherche peuvent être effectuée en employant l'option « q ».

```
[root@P100 /root]# nslookup
Default Server:  P100.challe.be
Address:  192.168.1.1

> P200
Server:  P100.challe.be
Address:  192.168.1.1

Name:    P200.challe.be
Address:  192.168.1.10

> set q=any
> P200
Server:  P100.challe.be
Address:  192.168.1.1

P200.challe.be  internet address = 192.168.1.10
challe.be      nameserver = P100.challe.be
P100.challe.be internet address = 192.168.1.1
> set q=ptr
> P100
Server:  P100.challe.be
Address:  192.168.1.1

challe.be
    origin = challe.be
    mail addr = root.localhost
    serial = 1998120701
    refresh = 28800 (8H)
    retry = 14400 (4H)
    expire = 3600000 (5w6d16h)
    minimum ttl = 86400 (1D)
> set q=a
> p200
Server:  P100.challe.be
Address:  192.168.1.1

Name:    p200.challe.be
Address:  192.168.1.10

>
```

7.2. Réponses faisant autorité.

La première fois que nslookup recherche un nom distant, la réponse fait autorité, alors qu'elle ne le fait plus la seconde fois.

La première fois que le serveur local recherche www.isiph.be, il contacte un serveur de isiph.be dont la réponse fait autorité. Le serveur local retransmet cette réponse à nslookup et la place également dans sa mémoire-cache. Lors de la seconde recherche de www.isiph.be, le serveur local utilise l'information stockée dans sa mémoire-cache, ce qui produit une réponse ne faisant pas autorité.

Dans l'exemple qui suit, les noms recherchés sont totalement qualifiés (point terminal) mais le résultat serait le même sans le point terminal. L'utilisation du point terminal n'est en effet pas toujours indispensable. Au lieu d'hésiter sur la

mise ou non d'un point terminal, il est préférable d'en mettre un systématiquement dès que le nom est totalement qualifié.

```
[root@P100 /root]# nslookup
Default Server: P100.challe.be
Address: 192.168.1.1

> www.isiph.be.
Server: P100.challe.be
Address: 192.168.1.1

Name: email.isiph.be
Address: 212.68.213.100
Aliases: www.isiph.be

> www.isiph.be.
Server: P100.challe.be
Address: 192.168.1.1

Non-authoritative answer:
Name: email.isiph.be
Address: 212.68.213.100
Aliases: www.isiph.be

>
```

7.3. Changement de serveur.

La commande nslookup peut interroger directement un serveur spécifique. Pour cela, il suffit de définir le nom du serveur à interroger par l'intermédiaire de l'option server. Pour retourner au serveur local, il suffit d'employer l'option lserver.

```
[root@P100 /root]# nslookup
Default Server: P100.challe.be
Address: 192.168.1.1
```

Le serveur utilisé lors du démarrage de la session interactive est le lserver P100.challe.be.

```
> server email.isiph.be.
Default Server: email.isiph.be
Address: 212.68.213.100
```

```
> www.isiph.be
Server: email.isiph.be
Address: 212.68.213.100
```

```
*** email.isiph.be can't find www.isiph.be: No response from server
```

Puisque le nouveau serveur ne répond pas, il est possible de revenir au serveur d'origine en utilisant l'option server.

```
> server p100.challe.be
*** Can't find address for server p100.challe.be: No response from server
> lserver p100.challe.be
Default Server: p100.challe.be
Address: 192.168.1.1
```

Le serveur email.isiph.be n'est pas capable de retrouver l'adresse du serveur p100.challe.be. Aussi, la commande lserver permet-elle d'interroger le serveur local afin d'effectuer cette opération.

```
> www.isiph.be.
Server: p100.challe.be
```

```
Address: 192.168.1.1

Non-authoritative answer:
Name:     email.isiph.be
Address:  212.68.213.100
Aliases:  www.isiph.be

>
```

Le nom du serveur à interroger peut être placé comme second argument de la requête. L'interrogation du serveur yi.org au sujet de endor.yi.org donne la réponse suivante :

```
[root@P100 /root]# nslookup
Default Server:  P100.challe.be
Address: 192.168.1.1

> endor.yi.org yi.org
Server: yi.org
Address: 139.142.244.62

Non-authoritative answer:
Name:     endor.yi.org
Address:  212.68.199.229

>
```

Ce type d'opération peut également être réalisée de manière non interactive :

```
[root@P100 /root]# nslookup endor.yi.org yi.org
Server: yi.org
Address: 139.142.244.62

Non-authoritative answer:
Name:     endor.yi.org
Address:  212.68.199.229
```

7.4. Visualisation des paquets de requête et de réponse.

En validant l'option debug, nslookup peut visualiser les requêtes expédiées et les réponses reçues.

Un paquet DNS est composé de cinq sections :

- section d'en-tête ;
- section de la question ;
- section des réponses ;
- section des serveurs faisant autorité ;
- section des enregistrements complémentaires.

La section d'en-tête est présente dans chaque question et dans chaque réponse. Le code de l'opération est toujours égal à QUERY. L'identificateur id sert à associer une réponse à une requête ainsi qu'à détecter les duplications. Le champ des drapeaux (header flags), différencie la requête de la réponse. La chaîne want recursion indique le souhait de l'expéditeur de la requête de voir le serveur exécuter la totalité du travail. La chaîne auth answer indique que cette réponse fait autorité, c'est-à-dire que la réponse provient d'un serveur faisant autorité et non d'une mémoire cache.

Il y a toujours une section question dans un paquet DNS. Cette section contient le nom, le type et la classe de l'information recherchée. En raison de la structure du paquet DNS, il n'est pas possible de poser simultanément plusieurs questions.

La section des réponses contient les enregistrements qui répondent à la question posée.

La section des serveurs faisant autorité contient les enregistrements NS du serveur.

La section des enregistrements complémentaires est destinée à fournir des informations complémentaires à celles d'autres sections. Par exemple, si un serveur est mentionné dans la section des serveurs faisant autorité, l'adresse de ce serveur apparaît dans cette section.

Voici un exemple de session interactive ayant activé l'option debug :

```
[root@P100 /root]# nslookup
Default Server:  P100.challe.be
Address:  192.168.1.1

> set debug
> P200
Server:  P100.challe.be
Address:  192.168.1.1

;; res_nmkquery(QUERY, P200.challe.be, IN, A)
-----
Got answer:
  HEADER:
    opcode = QUERY, id = 4992, rcode = NOERROR
    header flags:  response, auth. answer, want recursion, recursion avail.
    questions = 1,  answers = 1,  authority records = 1,  additional = 1

  QUESTIONS:
    P200.challe.be, type = A, class = IN
  ANSWERS:
-> P200.challe.be
    internet address = 192.168.1.10
    ttl = 86400 (1D)
  AUTHORITY RECORDS:
-> challe.be
    nameserver = P100.challe.be
    ttl = 86400 (1D)
  ADDITIONAL RECORDS:
-> P100.challe.be
    internet address = 192.168.1.1
    ttl = 86400 (1D)

-----
Name:      P200.challe.be
Address:   192.168.1.10

>
```

7.5. Recherche à la manière d'un serveur de noms.

La commande nslookup peut envoyer des requêtes à la manière d'un serveur de noms. Pour cela, il n'est pas nécessaire d'effectuer une recherche récursive. De plus, il ne faut pas utiliser la liste de recherche du resolver. Il s'indique dès lors, de désactiver les fonctions récursive et de recherche. La désactivation de la consultation de la liste de recherche n'est pas nécessaire si les noms sont entièrement qualifiés.

Lorsque nslookup fonctionne en mode récursif, le serveur interrogé doit de lui-même interroger d'autres serveurs distants pour répondre à la question. Par contre, en mode non récursif, le serveur se contente de donner la réponse la plus proche.

Dans la recherche de www.isiph.be, la meilleure réponse que puisse fournir un serveur de noms est la liste des serveurs de be.

```
[root@P100 /root]# nslookup
Default Server: P100.challe.be
Address: 192.168.1.1

> set norec
> set nosearch
> www.isiph.be.
Server: P100.challe.be
Address: 192.168.1.1

Name: www.isiph.be
Served by:
- AUTH02.NS.UU.NET
  198.6.1.82
  be
- NS.EU.NET
  192.16.202.11
  be
- NS.BELNET.be
  193.190.198.10, 193.190.198.2
  be
- DNS.CS.KULEUVEN.AC.be
  134.58.40.4
  be
- SECDNS.EUNET.be
  193.74.208.139
  be
- SUSIC.SUNET.SE
  192.36.125.2
  be
- MASTER.DNS.be
  194.7.171.243
  be
```

Il faut à présent interroger les serveurs de be de manière à affiner la recherche.

```
> server auth02.ns.UU.net
Default Server: auth02.ns.UU.net
Address: 198.6.1.82

> www.isiph.be.
Server: auth02.ns.UU.net
Address: 198.6.1.82

Name: www.isiph.be
Served by:
- ns.isiph.be
  212.68.213.98
  isiph.be
- brutele01.brutele.be
  212.68.193.32
  isiph.be
```

Le serveur auth02.ns.UU.net spécifie la liste des serveurs isiph.be. Il suffit à présent d'interroger l'un de ces serveurs pour obtenir l'information recherchée.

```
> server ns.isiph.be
Default Server: ns.isiph.be
Address: 212.68.213.98
```

```
> www.isiph.be.  
Server: ns.isiph.be  
Address: 212.68.213.98  
  
Name: email.isiph.be  
Address: 212.68.213.100  
Aliases: www.isiph.be  
  
>
```

Cet exemple montre le mécanisme de recherche utilisé par un serveur de noms. Tout au long de ces manipulations, la question était : quelle est l'adresse de `www.isiph.be` ? Si un serveur de `be` avait eu la réponse en mémoire-cache, il aurait envoyé la réponse au lieu de fournir des renseignements sur les serveurs de `isiph.be`.

7.6. Transferts de zone.

Il est possible à la commande `nslookup` de transférer la totalité d'une zone au moyen de l'option `ls`. Comme il est possible de rediriger la visualisation vers un fichier. Soit pour des raisons de sécurité, soit pour limiter leur charge certains serveurs peuvent refuser de transférer leur zone.

Voici un exemple de transfert de zone :

```
[root@P100 /root]# nslookup  
Default Server: P100.challe.be  
Address: 192.168.1.1  
  
> ls challe.be.  
[P100.challe.be]  
$ORIGIN challe.be.  
P3450 1D IN A 192.168.1.70  
P2366 1D IN A 192.168.1.80  
macii 1D IN A 192.168.1.2  
powerpc 1D IN A 192.168.1.3  
P100 1D IN A 192.168.1.1  
se30 1D IN A 192.168.1.60  
P2450 1D IN A 192.168.1.40  
P166PLUS 1D IN A 192.168.1.30  
P2300 1D IN A 192.168.1.50  
P133 1D IN A 192.168.1.20  
P2450 1D IN A 192.168.1.40  
HP4 1D IN A 192.168.1.192  
fuji 1D IN A 192.168.1.4  
P166 1D IN A 192.168.1.5  
modem 1D IN A 192.168.1.100  
P200 1D IN A 192.168.1.10  
  
>
```



VI. Le courrier électronique.

1. Introduction.

Depuis la création des premiers réseaux, le courrier électronique est l'application la plus employée. Au début, ce service consistait en la simple copie d'un fichier d'une machine à une autre. Avec l'émergence d'Internet, le concept de transfert d'informations est resté identique bien qu'un routage complexe et une charge de travail accrue ayant donné naissance à un schéma plus élaboré.

Un certain nombre de programmes de transport de mails (Mail Transport Agent) ont été implémentés sur les systèmes UNIX. Le MTA le plus connu est sendmail. Cette application nécessite la création d'un fichier de configuration afin d'y imposer des règles définissant le comportement du programme de transport de mails.

2. La structure d'un message.

Un mail est généralement composé :

- du corps du message ;
- de données d'administration.

Le corps du message est uniquement composé du texte du message alors que les données d'administration sont composées d'informations concernant les destinataires du message, le moyen de transport utilisé, etc ...

Les informations administratives peuvent être rassemblées en deux catégories. La première est constituée de toutes les informations relatives au transport de l'information comme les adresses de l'expéditeur et du destinataire. Cette catégorie est appelée l'enveloppe. Ces informations peuvent être modifiées par les logiciels de transport. La seconde catégorie est constituée des informations permettant de manipuler le message. Cette catégorie regroupe donc des informations qui ne sont pas spécifiques au mécanisme de transport des informations. Dans cette catégorie figurent le sujet du message, la liste de tous les destinataires ainsi que la date d'expédition du dit message. Ces informations forment l'en-tête du mail. La structure de l'en-tête est définie par la RFC-822.

L'évolution dans l'utilisation des mails a nécessité des changements en vue d'une adaptation au chiffrement des informations, à l'utilisation d'un jeu de caractères international, ... Ces changements dans l'utilisation du courrier électronique ont conduit à la création d'autres standards. Néanmoins, dans tous ces standards, l'en-tête est constitué de plusieurs lignes séparées les unes des autres par un caractère de fin de ligne. Une ligne est composée d'un nom de champ suivi du caractère deux points et d'un espace. A la suite de cette description figure la valeur du champ.

Voici un exemple d'en-tête :

```
From jfc@isec.be Sat Oct 27 11:48:11 2001
Return-Path: <jfc@isec.be>
Received: from gateway1.isec.net (root@isec4 [212.68.194.203])
        by mail.challe.yi.org (8.9.3/8.9.3) with ESMTTP id LAA02458;
        Sat, 27 Oct 2001 11:48:10 +0200
From: jfc@isec.be
Received-Date: Sat, 27 Oct 2001 11:48:10 +0200
Received: (from jfc@localhost)
        by gateway1.isec.net (8.9.3/8.9.3/Debian 8.9.3-21) id LAA28363;
        Sat, 27 Oct 2001 11:56:07 +0200
Date: Sat, 27 Oct 2001 11:56:07 +0200
Message-Id: <200110270956.LAA28363@gateway1.isec.net>
To: jfc@challe.yi.org
Subject: tst
Cc: root@challe.yi.org
```

Généralement, tous les en-têtes sont générés par le programme qui est utilisé pour rédiger les mails. Voici la liste des en-têtes les plus employés ainsi que leurs significations :

- From. Ce champ contient l'adresse mail de l'expéditeur.
- To. Cette partie de l'en-tête donne la liste de tous les destinataires du message.
- Cc. Ce champ (Carbon copies) détermine la liste des destinataires qui recevront une copie du message.
- Subject. Chaque mail contient un champ sujet permettant de décrire en quelques mots l'objet du courrier.
- Date. Tout mail possède une date d'envoi.
- Reply-TO. Un mail peut spécifier l'adresse à laquelle les réponses doivent être expédiées.
- Message-ID. La valeur de ce champ est une chaîne générée par le programme de transport du système de départ permettant d'identifier le message de manière univoque.
- Received. Tous les hôtes par lesquels le courrier transite ajoutent un tel champ à l'en-tête. Les valeurs présentes dans ce champ sont l'identification du site, un identificateur de message, le moment de la réception du message, le site de provenance du message ainsi que le nom du logiciel de transport utilisé. Ces informations permettent de retracer le chemin emprunté par un mail.

3. Principe de livraison des mails.

Un mail est composé en utilisant des programmes tels que netscape messenger, outlook express, pine etc ... Ces programmes sont appelés des MUA (Mail User Agent). Lorsqu'un utilisateur expédie un mail en utilisant l'interface d'un MUA, le message est transmis à un MTA chargé de l'expédition. Sur la plupart des systèmes le logiciel MTA est chargé à la fois de l'acheminement des mails locaux et distants.

La livraison d'un courrier local est plus qu'un simple ajout d'un nouveau message dans une boîte de réception. Les MTA peuvent traiter des alias ainsi que le forwarding. De plus, si un message ne peut être délivré en raison de la non existence du destinataire, par exemple, le MTA doit être capable d'envoyer automatiquement un message d'erreur à l'expéditeur.

La livraison de courriers distants est généralement effectuée au moyen du protocole SMTP (Simple Mail Transfer Protocol). Ce protocole est conçu pour délivrer directement le courrier à la machine du destinataire. En réalité, une négociation a lieu entre les démons SMTP des machines source et destination.

4. Le routage du courrier.

Le processus consistant à diriger les mails vers un hôte est appelé le routage. A partir du moment où l'adresse IP du destinataire est connue, la majorité du travail de routage est réalisé par la couche IP.

L'adresse d'un destinataire est constituée d'un nom d'utilisateur et d'un nom de domaine. Lorsqu'un courrier doit être délivré à un certain domaine, il est nécessaire de savoir quelle machine est capable de gérer l'arrivée de mails. Pour effectuer ce travail il faut faire appel au DNS. Les fichiers de configuration du DNS peuvent contenir un enregistrement MX (Mail eXchanger).

Les enregistrements MX désignent un hôte appelé échangeur de messages qui tantôt traite le courrier, et tantôt le retransmet. Le traitement du courrier consiste en la livraison du courrier à l'adresse indiquée. La retransmission du courrier concerne son envoi vers sa destination finale ou vers un autre échangeur de messages proche de la destination.

Chaque enregistrement MX possède une valeur de préférence codée sur 16 bits. Cette valeur indique, au routeur, l'ordre de priorité à utiliser lors du choix d'un échangeur de messages.

Voici une partie du fichier /var/named/challe.be :

```
[root@P100 named]# cat challe.be | more
@      IN      SOA      challe.be. root.localhost. (
                                1998120701 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000   ; Expire
                                86400     ) ; Minimum

      NS      P100.challe.be.
      MX      10      P100.challe.be.
```

Dans cet exemple, la machine P100.challe.be est l'échangeur de messages pour le domaine challe.be. Cela signifie que si un mail stipule une adresse dans le domaine challe.be, c'est l'hôte P100.challe.be qui est à même de recevoir l'information.

5. Sendmail.

La diversité des programmes et des protocoles utilisés en matière de courrier électronique complique la configuration de sendmail. Le protocole SMTP envoie le courrier électronique à travers les réseaux TCP/IP. Un autre programme envoie du courrier électronique entre les utilisateurs situés sur la même machine. Chacun de ces systèmes de courrier électronique possède son propre programme de distribution et son propre système d'adressage.

5.1. Les fonctions de sendmail.

L'application sendmail supprime la plus grande partie des problèmes engendrés par l'utilisation de plusieurs programmes de livraison de courrier électronique. Le programme sendmail route le courrier électronique vers le programme de livraison correct en se basant sur l'adresse email. Il accepte du courrier provenant d'un programme de courrier électronique de niveau utilisateur (MUA), interprète l'adresse, la réécrit sous une forme appropriée pour le programme de livraison et enfin route le courrier vers le bon programme de livraison. De plus, pour tout courrier arrivant, sendmail interprète l'adresse et, soit délivre le courrier, soit le renvoie à un autre système.

En plus de router le courrier entre les programmes utilisateurs et les programmes de livraison, sendmail joue deux rôles supplémentaires :

- recevoir et livrer les courriers électroniques SMTP (Internet) ;
- fournir un système d'alias de courrier électronique permettant la mise en place de listes de discussion.

5.2. Le démon sendmail.

Par défaut, sendmail est capable de recevoir un courrier SMTP en provenance du réseau. Pour effectuer cela, il suffit de lancer sendmail au démarrage du système. Dès son lancement, sendmail scrute le port TCP numéro 25. Il traite ainsi le courrier électronique entrant. Le fichier de démarrage automatique de sendmail contient généralement deux options comme le montre un extrait du fichier /etc/rc.d/init.d/sendmail :

```
daemon /usr/sbin/sendmail -bd -q 15m
```

La seconde option de lancement de sendmail (-q) spécifie le rythme de traitement des courriers électroniques de la file d'attente. Dans l'exemple ci-dessus, la file d'attente est traitée toutes les 15 minutes. Lorsque temporairement un message ne peut pas être délivré, il est placé dans une file d'attente. Cette mise en attente peut être provoquée par une indisponibilité temporaire de la machine réceptrice du message. Lorsque le message aura été délivré, il sera retiré de la file d'attente.

La première option (-bd) concerne directement la réception du courrier électronique par SMTP. Cette option ordonne à sendmail de fonctionner en tant que démon et de scruter le port TCP 25 pour réceptionner le courrier électronique. L'emploi de cette option est impératif dans le cas où le système doit être en mesure de recevoir du courrier électronique via TCP/IP.

5.3. Configuration de sendmail.

Le programme sendmail est le serveur de mails. Conçu en 1982, souple et puissant, il est présent et utilisé sur la plupart des serveurs. Ce produit est en évolution constante depuis plus de 20 ans. Ce MTA a mauvaise réputation. Pendant de nombreuses années, il n'existait pas d'alternative aussi puissante, et l'écriture du fichier de configuration était un cauchemar pour bon nombre d'administrateurs systèmes. Aujourd'hui, les macros m4 ont rendu le programme plus simple à configurer. Ce système de mails peut gérer un nombre important de connexions simultanées. C'est une des raisons qui a conduit la plupart des administrateurs de gros systèmes à l'employer.

Le programme sendmail est gourmand en ressources CPU. Si un administrateur est confronté à la configuration d'un petit site ne recevant pas beaucoup de courriers ou à un serveur modeste, il est préférable de choisir un autre MTA. De plus, bien que des améliorations dans le processus de configuration exista, sendmail reste un des serveurs de courriers électroniques les plus difficiles à configurer.

5.3.1. Configurer le mail local.

L'objectif est de configurer un serveur de courriers électroniques pour une utilisation interne. Les différents utilisateurs d'un système UNIX doivent avoir la possibilité de s'échanger des messages sans pour autant envoyer des mails vers l'extérieur.

Voici un exemple de fichier de macros m4 destinées à la configuration du courrier local :

```
[root@P100 sendmail] cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
FEATURE(local_procmail)
MAILER(procmail)
```

Tout fichier de macros doit contenir la directive OSTYPE afin de spécifier le système d'exploitation utilisé. Le paramètre est le nom d'un fichier contenant des informations sur :

- l'emplacement des fichiers de configuration ;
- les chemins et les arguments employés par les mailers ;
- les répertoires utilisés par sendmail pour stocker les message ;
- ...

Cette option est donc obligatoire et devrait apparaître au début de tout fichier de configuration.

La macro MAILER permet de spécifier comment le courrier doit être distribué. Dans cette configuration, les mails doivent uniquement être échangés entre les utilisateurs du système. Sous Linux, l'agent de distribution local est le programme procmail. Il faut donc spécifier l'argument procmail au niveau de la macro MAILER.

Le programme procmail réside à un endroit précis de la hiérarchie du système de fichier. Cet emplacement est défini au moyen de la macro FEATURE. Le paramètre local_procmail de cette macro stipule notamment quel est le chemin d'accès au programme procmail.

La première ligne du fichier est destinée à inclure des définitions servant à une bonne génération du fichier sendmail.cf. Dès que le fichier sendmail.mc est encodé, il faut générer le fichier sendmail.cf au moyen de la commande suivante :

```
[root@P100 sendmail] m4 sendmail.mc > /etc/sendmail.cf
```

Cette commande demande au macro processeur m4 de lire le fichier sendmail.mc et de générer un fichier sendmail.cf qui sera placé dans le répertoire /etc. Le serveur de courriers électroniques doit à présent être redémarré afin qu'il applique la nouvelle configuration.

```
[root@P100 sendmail] /etc/rc.d/init.d/sendmail restart
```

A présent, les utilisateurs peuvent s'échanger du courrier localement en utilisant la commande mail. Voici un exemple de message envoyé par l'utilisateur root à l'utilisateur jfc.

```
[root@P100 sendmail]# mail jfc
Subject: Nouveau serveur de mails
Monsieur,

Le serveur de mails est des maintenant disponible sur ce systeme.
.
Cc:
```

L'utilisateur jfc reçoit un message lui indiquant qu'un courrier est arrivé. Il suffit à jfc d'utiliser la commande mail sans paramètre pour activer le mode interactif de ce programme afin de relever le courrier. La commande p lui permet de visualiser le texte du message. La commande d permet d'effacer un message alors que la commande q permet de quitter le logiciel mail.

```

You have new mail.
[jfc@P100 jfc]$ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/jfc": 1 message 1 new
>N 1 root          Sun Nov  4 20:21  15/405  "Nouveau serveur de ma"
& p
Message 1:
From root Sun Nov  4 20:21:27 2001
Date: Sun, 4 Nov 2001 20:21:26 +0100
From: root <root>
To: jfc
Subject: Nouveau serveur de mails

Monsieur,

Le serveur de mails est des maintenant disponible sur ce systeme.

& d
& q
[jfc@P100 jfc]$

```

5.3.2. Configurer le mail distant.

Dans cette configuration, il faut que le serveur de courriers électroniques soit à la fois capable d'envoyer du courrier vers un destinataire connu d'Internet mais également d'en recevoir en provenance d'Internet.

Sur Internet, les courriers sont échangés au moyen du protocole SMTP. Il faut donc que sendmail soit capable de gérer ce type de connexion. Pour cela, il faut ajouter MAILER(smtp) à la configuration précédente. De cette manière, l'agent de transport SMTP est connu de sendmail.

Lorsque les mails arrivent, ils contiennent le nom complet du destinataire, nom complet qui doit être accepté en local. Pour cela, il faut stipuler, dans le fichier sendmail.cw, le nom de domaine qui dans ce cas est challe.yi.org. Ce fichier doit contenir la liste de tous les domaines pour lesquels sendmail doit accepter les messages. Voici le contenu des fichiers sendmail.mc et sendmail.cw :

```

[root@P100 sendmail] cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
[root@P100 sendmail]# cd /etc
[root@P100 /etc]# cat sendmail.cw
# sendmail.cw - include all aliases for your machine here.
challe.yi.org

```

Après avoir activé ces modifications, sendmail permet d'envoyer du courrier vers des utilisateurs distants ainsi que d'en recevoir en provenance d'utilisateurs distants.

5.3.3. Agir comme serveur relais.

Un grand nombre de configurations sont établies de telle sorte que des utilisateurs sous Windows puissent envoyer du courrier en utilisant le serveur SMTP à présent opérationnel. Dans l'état actuel des choses, cette opération se soldera par un échec car pour des raisons de sécurité le serveur de mail refuse d'être serveur relais. Pour cela, il faut alors définir dans la base de données des accès que certaines machines peuvent exploiter le serveur de mails comme serveur relais. De plus, la configuration de sendmail doit être modifiée afin qu'il puisse consulter cette base de données. Il faut pour cela ajouter la macro FEATURE(`access_db').

Voici la nouvelle configuration présente dans le fichier sendmail.mc :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

Supposons que le serveur de mails soit un routeur entre un réseau local et internet. Supposons également que le réseau local soit dans le domaine challe.be. Il faut alors éditer le fichier `/etc/mail/access` afin d'autoriser les machines du domaine challe.be à utiliser le serveur en tant que serveur relais.

Voici le contenu du fichier `/etc/mail/access` :

```
[root@P100 mail]# cat access
# Check the /usr/doc/sendmail-8.9.3/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/doc/sendmail-8.9.3/README.cf is part of the sendmail-doc
# package.
#
# by default we allow relaying from localhost...
localhost.localdomain      RELAY
localhost                  RELAY
challe.be                  RELAY
```

Le fichier `/etc/mail/access` n'est qu'une version lisible du véritable fichier de base de données qui est `/etc/mail/access.db`. Afin de rendre actives, les modifications présentes dans le fichier texte, il faut générer la base de données au moyen du `makefile` présent dans le répertoire `/etc/mail`. Cette création nécessite simplement l'utilisation de la commande `make`.

Le fichier `/etc/mail/access` peut contenir d'autres spécifications que `RELAY`. Voici la liste des spécifications les plus courantes :

Valeur	Action
OK	Accepter le courrier même si d'autres règles opposent un refus.
RELAY	Accepter le message même s'il n'est pas destiné à l'hôte exécutant sendmail
REJECT	Rejeter le message en envoyant un message de refus

Dans le même ordre d'idée, les utilisateurs d'Internet relèvent leur courrier à distance. Pour cela, ils configurent leur programme de gestion du courrier électronique pour qu'il se connecte à un serveur POP (Post Office Protocol) afin d'y télécharger le courrier. Cette fonctionnalité, qui ne fait pas partie de sendmail, constitue un outil annexe. Pour qu'un système UNIX agisse comme serveur POP, il suffit qu'il exécute le démon `pop`. Le lancement de ce démon est effectué par le super démon `inetd`. Le fichier de configuration de `inetd` est `/etc/inetd.conf`. Il faut que les lignes concernant POP ne soient pas placées en commentaire pour que le serveur POP puisse fonctionner.

Voici extrait du fichier `inetd.conf` relatif au serveur POP :

```
#
# Pop and imap mail services et al
#
pop-2  stream  tcp    nowait  root    /usr/sbin/tcpd  ipop2d
pop-3  stream  tcp    nowait  root    /usr/sbin/tcpd  ipop3d
imap   stream  tcp    nowait  root    /usr/sbin/tcpd  imapd
#
```

Ces entrées supposent que l'exécutable se trouve dans le répertoire `/etc/sbin`. Les ports de communication utilisés par ces services sont définis dans le fichier `/etc/services`. Voici un extrait du fichier `/etc/services` :

```

pop2          109/tcp          pop-2  postoffice      # POP version 2
pop2          109/udp          pop-2
pop3          110/tcp          pop-3          # POP version 3
pop3          110/udp          pop-3
imap2         143/tcp          imap          # Interim Mail Access Proto v2
imap2         143/udp          imap
imap3         220/tcp          # Interactive Mail Access
imap3         220/udp          # Protocol v3

```

5.3.4. Options supplémentaires.

Lorsque des courriers locaux sont échangés, le nom de la machine n'est pas indiqué après le nom de l'utilisateur. Pour remédier à cette situation, il faut explicitement demander à sendmail d'ajouter le nom de domaine au moyen de FEATURE (always_add_domain). Voici le nouveau fichier de configuration de sendmail permettant cette fonctionnalité.

```

[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')

```

L'envoi d'un courrier ne nécessite pas l'utilisation d'un nom complet car sendmail modifiera lui même l'adresse du destinataire. Il est donc permis d'envoyer un mail local de la manière suivante :

```

[root@P100 sendmail]# mail jfc
Subject: message
bonjour
.
Cc:

```

L'utilisateur remarque que le mail provient de root@ns.challe.yi.org et non pas simplement de root.

```

[jfc@P100 jfc]$ mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/jfc": 1 messages 1 new
> N 1 root@ns.challe.yi.or Sun Nov 4 20:56 13/353 "message"
& p 2
Message 1:
From root Sun Nov 4 20:56:43 2001
Date: Sun, 4 Nov 2001 20:56:43 +0100
From: root <root@ns.challe.yi.org>
To: jfc@ns.challe.yi.org
Subject: message

bonjour

&

```

Dans le cadre d'une utilisation normale d'Internet, des gens souscrivent à des listes de discussion, publient des articles du UseNet, ... Ces opérations diffusent l'adresse de courrier électronique. Certains font la chasse aux adresses mails pour les revendre à des publicistes vantant leurs produits via Internet. Des personnes reçoivent ainsi des courriers non sollicités appelés spam.

Fort heureusement, sendmail permet d'éliminer les spams. Pour cela, il faut se référer à The Real-time blackhole List, organisation qui maintient une base de données d'adresses mails source de spams. Le programme sendmail peut exploiter cette base de données et refuser des mails provenant d'adresses qui figurent sur cette liste noire. Pour activer

cette fonctionnalité, il suffit d'ajouter FEATURE(rbl) au fichier de configuration de sendmail. Voici la nouvelle version du fichier sendmail.mc :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

Lorsque des messages sont envoyés, sendmail ajoute automatiquement à l'adresse de l'expéditeur le nom de domaine pleinement qualifié. Avec la configuration actuelle, tous les mails envoyés ont une adresse d'expédition de la forme utilisateur@ns.challe.yi.org. Pour éviter que le nom de la machine soit ajouté au nom de domaine, il suffit de définir ce dernier à la main au moyen d'une macro define comme le montre l'exemple suivant :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confDOMAIN_NAME', `challe.yi.org')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

L'exemple suivant montre que le nom de domaine n'est plus pleinement qualifié.

```
[root@P100 /root]# mail jfc
Subject: Bonjour
Bonjour Monsieur
.
Cc:

[jfc@P100 jfc]$ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/jfc": 1 message 1 new
>N 1 root@challe.yi.org  Tue Nov  6 13:32  13/350  "Bonjour"
& p
Message 1:
From root  Tue Nov  6 13:32:42 2001
Date: Tue, 6 Nov 2001 13:32:42 +0100
From: root <root@challe.yi.org>
To: jfc@challe.yi.org
Subject: Bonjour

Bonjour Monsieur

&
```


5.4. Sendmail et les alias.

Les alias doivent fournir :

- d'autres noms (surnoms) pour les utilisateurs ;
- la possibilité de renvoyer du courrier à d'autres machines ;
- les listes de discussion.

Ces alias sont définis dans le fichier `/etc/aliases`. Le format des entrées de ce fichier est :

```
alias: destinataire {,destinataire, ... }
```

Le mot « alias » est le nom de l'utilisateur à qui le courrier électronique est adressé. Les mots « destinataire » spécifient les noms à qui le courrier est véritablement destiné. Le destinataire peut être le nom d'un utilisateur du système, un autre alias ou bien une adresse électronique complète contenant à la fois le nom d'un utilisateur et celui d'une machine. Un courrier envoyé à alias sera envoyé à tous les destinataires spécifiés dans la liste. Cette méthode permet la création d'une liste de discussion.

Voici un exemple de fichier `/etc/aliases` :

```
[root@P100 /etc]# cat aliases
MAILER-DAEMON:  postmaster
postmaster:    root
bin:           root
daemon:        root
games:         root
ingres:        root
nobody:        root
system:        root
toor:          root
uucp:          root
manager:       root
dumper:        root
operator:      root
decode:        root
jfchalle:     jfc
jf.challe:     jfc
mailinglist:   jfc, rc, root, jfc@p200.challe.be
```

Les 14 premières lignes de ce fichier sont les alias par défaut du système. La quinzième ligne du fichier indique que tous les mails envoyés à jfchalle sont redirigés vers l'utilisateur jfc. La dernière ligne du fichier crée une liste de discussion. Cela signifie que tous les mails envoyés à mailinglist seront envoyés aux utilisateurs jfc, rc et root ainsi qu'à l'utilisateur jfc de la machine p200.challe.be.

Le programme sendmail n'exploite pas directement le fichier `/etc/aliases`. Ce fichier doit être traité de manière à générer une base de données qui sera exploitée par sendmail. Pour transformer le fichier `/etc/aliases` en une base de données, il faut utiliser la commande `newaliases` ou bien la commande `sendmail` avec l'option `-bi`.

Chaque fois que le fichier des alias est modifié, il faut générer une nouvelle base de données afin que les modifications soient prises en compte par sendmail. Voici les deux exemples de création de la base de données des alias :

```
[root@P100 /etc]# sendmail -bi
/etc/aliases: 18 aliases, longest 34 bytes, 243 bytes total
[root@P100 /etc]# newaliases
/etc/aliases: 18 aliases, longest 34 bytes, 243 bytes total
```

5.5. Utilisation d'un .forward personnel.

En plus du mécanisme de renvoi de courrier électronique par les alias, sendmail permet à un utilisateur de définir lui-même son propre renvoi de courrier. Il lui suffit de créer un fichier .forward à la racine de son compte utilisateur. Le programme sendmail consulte ce fichier après avoir utilisé le fichier des alias et avant d'effectuer la diffusion finale du courrier à l'utilisateur. Si le fichier .forward existe, sendmail diffuse le message en fonction du contenu de ce fichier.

Voici un exemple de fichier .forward ou l'utilisateur jfc redirige lui-même son courrier vers jfc@p200.challe.be :

```
[jfc@P100 jfc]$ cat .forward
jfc@P200.challe.be
```

5.6. Configuration de la sécurité.

5.6.1. Les commandes vrfy et expn.

Les commandes SMTP vrfy et expn sont généralement interdites car elles fournissent des informations sur les adresses de messageries disponibles sur le serveur. Voici une session telnet lancée sur le port SMTP à partir de la machine exécutant le serveur sendmail :

```
jfc@gateway:~$ telnet challe.yi.org 25
Trying 212.68.198.221...
Connected to challe.yi.org.
Escape character is '^]'.
220 challe.yi.org ESMTP Sendmail 8.9.3/8.9.3; Wed, 7 Nov 2001 11:32:01 +0100
vrfy jfc
250 Jean-Francois Challe <jfc@challe.yi.org>
expn jfc
250 Jean-Francois Challe <jfc@challe.yi.org>
```

Pour interdire l'utilisation de ces commandes, il faut définir confPRIVACY_FLAGS comme le montre l'exemple suivant :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confPRIVACY_FLAGS',novrfy noexpn)
define(`confDOMAIN_NAME',`challe.yi.org')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

Les drapeaux novrfy et noexpn permettent d'interdire respectivement les commandes vrfy et expn. Voici un exemple de session interactive lorsque les commandes vrfy et expn sont interdites :

```
jfc@gateway:~$ telnet challe.yi.org 25
Trying 212.68.198.221...
Connected to challe.yi.org.
Escape character is '^]'.
220 challe.yi.org ESMTP Sendmail 8.9.3/8.9.3; Wed, 7 Nov 2001 11:38:35 +0100
vrfy
252 Cannot VRFY user; try RCPT to attempt delivery (or try finger)
expn
502 Sorry, we do not allow this operation
```

5.6.2. Restriction d'accès à la file des messages.

Pour des raisons de sécurité, il est nécessaire d'empêcher l'accès à la file des messages par des personnes ne disposant pas des autorisations nécessaires. Cette option est activée en positionnant le drapeau restrictmailq.

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confPRIVACY_FLAGS',authwarnings novrfy noexpn restrictmailq)
define(`confDOMAIN_NAME',`challe.yi.org')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

5.6.3. Forcer l'identification du client.

Les utilisateurs souhaitant utiliser le serveur SMTP peuvent également être invités à s'identifier avant de pouvoir envoyer un message. Pour cela, il suffit d'imposer le drapeau needmailhelo. Voici le fichier sendmail.mc modifié :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confPRIVACY_FLAGS',authwarnings novrfy noexpn restrictmailq needmailhelo)
define(`confDOMAIN_NAME',`challe.yi.org')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

5.6.4. Message d'invite de sendmail.

Par défaut, sendmail envoie un certain nombre d'informations lorsqu'un client se connecte au serveur de messagerie. Ces informations permettent de découvrir sans effort, la version de sendmail utilisée ainsi que la version du fichier sendmail.cf. Le drapeau confSMTP_LOGIN_MSG permet de définir un nouveau message d'accueil. Voici la nouvelle version du fichier sendmail.mc

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confPRIVACY_FLAGS',authwarnings novrfy noexpn restrictmailq needmailhelo)
define(`confSMTP_LOGIN_MSG',Bonjour)
define(`confDOMAIN_NAME',`challe.yi.org')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```

Lorsqu'un client se connecte au serveur, il obtient le message suivant :

```
jfc@gateway:~$ telnet challe.yi.org 25
Trying 212.68.198.221...
Connected to challe.yi.org.
Escape character is '^]'.
220 Bonjour ESMTP
```

5.6.5. Contrôler les connexions SMTP avec TCP Wrappers.

Le paquetage TCP Wrappers permet d'effectuer un contrôle et un filtrage au niveau des connexions réseau faites sur le système. Il est possible de contrôler les connexions par adresse IP, nom de machine ou nom de domaine. Il est ainsi possible d'utiliser les fichiers /etc/hosts.allow et /etc/hosts.deny pour autoriser ou non les connexions au serveur de messagerie.

Pour des raisons de sécurité, le fichier /etc/hosts.deny devrait toujours contenir ALL:ALL. Cela signifie que tout ce qui n'est pas explicitement autorisé dans /etc/hosts.allow est interdit.

Si l'administrateur souhaite autoriser tout serveur à communiquer avec sendmail à l'exception la machine mail.isiph.be, il faut placer les informations suivantes dans les fichiers /etc/hosts.allow et /etc/hosts.deny :

```
[root@P100 etc]# cat hosts.allow
sendmail:ALL
[root@P100 etc]# cat hosts.deny
sendmail:mail.isiph.be
ALL:ALL
```

5.6.6. Lutter contre les attaques de déni de service.

De nombreux paramètres peuvent être configurés dans le fichier sendmail.mc afin d'offrir une résistance plus importante aux attaques visant à saturer les ressources du système. Voici une liste des variables intéressantes :

Variable	Signification
define(`confMin_FREE_BLOCKS',`100')	Définition du nombre minimum de blocs libres dans la queue pour accepter les nouveaux messages.
define(`confMAX_MESSAGE_SIZE',`5000000')	Détermine la taille maximale des messages qui sont acceptés (taille exprimée en bits).
define(`confAUTO_REBUIL',`FALSE')	Interdiction de la recompilation automatique des alias.
define(`confQUEUE_LA',`8')	Cette valeur indique une charge (nombre de messages à traiter simultanément) à partir de laquelle les messages ne sont plus délivrés mais placés en file d'attente. Il est conseillé de donner une valeur correspondant à 8 fois le nombre de processeurs disponibles dans la machine.
define(`confREFUSE_LA',`12')	Lorsque la charge moyenne exprimée en terme de nombre de messages à traiter au cours de la dernière minute dépasse la valeur indiquée, sendmail refuse les messages. Il est conseillé de donner une valeur correspondant à 12 fois le nombre de processeurs disponibles dans la machine.

5.6.7. Utilisation d'un shell restreint.

Bien que la programmation de sendmail soit soignée, il est possible de le détourner de son but premier en l'obligeant à exécuter certaines commandes au niveau du shell. Pour protéger le système contre ce type d'attaque, il suffit de limiter les commandes qui peuvent être exécutées par sendmail. Pour cela, il s'indique d'imposer à sendmail l'utilisation d'un shell restreint lorsqu'il doit exécuter des commandes. Ce shell est smrsh (SendMail Restricted Shell).

Voici le fichier sendmail.mc modifié pour imposer à sendmail d'employer un shell restreint :

```
[root@P100 sendmail]# cat sendmail.mc
include(`/usr/lib/sendmail-cf/m4/cf.m4')
OSTYPE(`linux')
define(`confPRIVACY_FLAGS',authwarnings novrfy noexpn restrictmailq needmailhelo)
define(`confSMTP_LOGIN_MSG',Bonjour)
define(`confDOMAIN_NAME',`challe.yi.org')
FEATURE(`smrsh',`/usr/sbin/smrsh')
FEATURE(rbl)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
```


VII. NFS.

1. Introduction.

Le système de fichiers partagés (Network File System) est un système de fichiers distribués fournissant un mécanisme transparent d'accès à des systèmes de fichiers distants. Le système NFS permet de centraliser l'administration des disques plutôt que de dupliquer des répertoires comme /usr/local sur tous les systèmes. Le système de fichiers distribués partage un ou plusieurs répertoires de sorte qu'il n'existe qu'une seule copie des informations sur lesquelles des systèmes distants peuvent opérer. Par le biais de NFS, les utilisateurs ne doivent pas se connecter au moyen d'une session telnet pour accéder aux informations. Dans le même ordre d'idée, il n'est pas nécessaire d'utiliser une disquette ou une bande magnétique pour transporter les données d'un système à l'autre. Dès que NFS est correctement configuré, les utilisateurs peuvent exploiter les fichiers distants comme s'ils résidaient sur la machine locale.

2. Configuration de NFS.

La mise en œuvre de NFS sur le serveur et sur les postes clients nécessite le lancement de démons qui implantent le protocole NFS. De plus, des démons auxiliaires sont lancés afin d'assurer des services tels que le verrouillage des fichiers, l'exportation des systèmes de fichiers et le montage des partages. Ces démons sont lancés par les scripts de démarrage du système.

3. Exportation d'un système de fichiers.

Lorsqu'un serveur partage des ressources au moyen de NFS, il conserve une liste des systèmes de fichiers exportés dans un fichier. Ce fichier associe des restrictions d'accès à chaque système de fichiers exporté. Lorsqu'un client souhaite monter un système de fichiers partagés, le serveur compare la demande du client à la liste des systèmes de fichiers qu'il exporte afin de déterminer si le client a le droit d'accéder aux informations.

La table des systèmes de fichiers exportés figure dans le fichier /etc/exports. En plus des fichiers initialement exportés, le super-utilisateur a à tout moment la possibilité d'exporter d'autres systèmes de fichiers au moyen de la commande exportfs. Lors du démarrage du système, le fichier /etc/exports est lu. Pour chacune des entrées de ce fichier, le système exécute la commande exportfs correspondante de manière à rendre les systèmes de fichiers exportés disponibles pour les clients. Lorsqu'un client monte un système de fichiers exporté, le serveur conserve la trace de l'utilisation de la ressource dans le fichier /var/lib/nfs/xtab.

3.1. Règles d'exportation des systèmes de fichiers.

Pour rendre un système de fichiers accessible par l'intermédiaire de NFS, il est impératif suivre les règles suivantes :

- Tout système de fichiers ou tout sous-répertoire d'un système de fichiers peut être exporté par un serveur. Cela permet d'exporter un nombre très limité de fichiers. Cette méthode est fortement employée par les stations de travail qui ne possèdent pas de disque.
- Il est permis d'exporter n'importe quel sous-répertoire d'un système de fichiers exporté à la condition qu'il réside sur des disques différents. Par exemple, si le répertoire /home réside sur le disque /dev/hda1 et que le répertoire /home/jfc réside sur le disque /dev/dha2 il est permis d'exporter ces deux systèmes de fichiers.
- Il est interdit d'exporter un répertoire parent d'un système de fichier exporté sauf s'il réside sur des disques différents. Cette règle est identique à la précédente sauf que les permissions sont exprimées en terme de répertoire parent au lieu de répertoire enfant.
- Seuls les systèmes de fichiers locaux peuvent être exportés.

3.2. Exportation simple d'un système de fichiers.

Le fichier /etc/exports contient la liste des systèmes de fichiers que le serveur exporte ainsi que les restrictions et les options correspondantes. Chaque ligne de ce fichier correspond à un point de montage et à une liste de machines ou de sous-réseaux autorisés à monter l'élément correspondant. Voici un exemple de fichier /etc/exports :

```
[root@P100 /etc]# cat exports
/mnt/cdrom
/mnt/floppy
```

Dans cet exemple, les répertoires /mnt/cdrom et /mnt/floppy de l'hôte P100 sont accessibles en lecture seule à partir de tous les hôtes du réseau.

La modification du fichier /etc/exports n'a pas de répercussion immédiate sur les permissions d'accès aux systèmes de fichiers. En effet, le fichier /etc/exports est lu au moment du démarrage du système. De ce fait, si le super-utilisateur décide de modifier le fichier des exportations, il doit, après avoir effectué les changements, les faire prendre en considération par le système. Pour cela, il lui suffit d'employer la commande `exportfs` en demandant une relecture du fichier /etc/exports. Voici la syntaxe de la commande `exportfs` forçant la relecture du fichier des exportations :

```
[root@P100 /etc]# exportfs -r
```

4. Montage d'un système de fichiers.

Les machines clientes peuvent monter un système de fichiers exporté au moyen de la commande `mount`. Voici un exemple de montage d'un système de fichiers.

```
[root@p200 /root]# mount -t nfs P100:/mnt/cdrom /mnt
```

Cette commande permet de monter le système de fichiers /mnt/cdrom résidents sur l'hôte P100 dans le répertoire /mnt de la machine P200. Dès que cette commande est exécutée, il est possible d'exploiter les fichiers placés dans le répertoire /mnt comme s'ils faisaient partie du disque local du client.

Les différents systèmes de fichiers montés peuvent être visualisés au moyen de la commande `df`. Voici un exemple d'exécution de cette commande :

```
[root@p200 /root]# df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/hda5        7961648    1440976   6116240  19% /
/dev/hda1         23302       2482     19617  11% /boot
P100:/mnt/cdrom 13156032 11587248   900496  93% /mnt
```

5. Les options d'exportation.

Plusieurs options peuvent modifier la manière dont un système de fichiers est exporté au travers du réseau :

- `rw` : permet aux clients NFS de lire et d'écrire sur le système de fichiers exporté
- `ro` : empêche les clients NFS d'écrire sur le système de fichiers.

Voici un exemple d'utilisation des options `ro` et `rw` :

```
[root@P100 /etc]# cat exports
/mnt/cdrom      192.168.1.0/255.255.255.0(rw)
/mnt/floppy     (ro)
/home           P200(rw)
```

La première ligne du fichier indique que le répertoire /mnt/cdrom est exporté en lecture et en écriture vers tous les hôtes du sous réseau 192.168.1.0. La seconde ligne exporte le répertoire /mnt/floppy en lecture seule vers tous les hôtes. La dernière ligne du fichier /etc/exports permet d'exporter le répertoire /home en lecture et en écriture uniquement vers l'hôte p200.

D'après la description qui vient d'être faite, tout utilisateur de l'hôte p200 devrait avoir accès en lecture et en écriture à l'ensemble de la hiérarchie des fichiers accessibles à partir de /home. Voici un exemple de session interactive lancée à partir de l'hôte P200 :

```
[root@p200 /root]# mount -t nfs P100:/home /mnt/floppy
[root@p200 /root]# cd /mnt/floppy/
[root@p200 floppy]# touch ess
touch: ess: Permission non accordée
```


Bien que ce soit le super-utilisateur de la machine P200 qui tente de créer un fichier dans le répertoire /home de l'hôte P100, l'opération échoue. Lorsqu'un utilisateur d'UID 0 (root) accède à un système de fichiers exporté, son UID est transformé en l'UID de l'utilisateur nobody ou bien en la valeur 65534 si l'utilisateur nobody n'existe pas sur le système client. Cette technique constituant le comportement par défaut porte le nom de *not squashing*. Ce comportement peut être inhibé au moyen de l'option `no_root_squash`. Voici le fichier `/etc/exports` modifié :

```
[root@P100 /etc]# cat exports
/mnt/cdrom      192.168.1.0/255.255.255.0(rw)
/mnt/floppy     (ro)
/home          P200(rw,no_root_squash)
```

Une session interactive sur l'hôte P200 montre que l'utilisateur root de la machine P200 est considéré comme étant le super-utilisateur de la machine P100.

```
[root@p200 /root]# mount -t nfs P100:/home /mnt/floppy
[root@p200 /root]# cd /mnt/floppy
[root@p200 floppy]# touch ess
[root@p200 floppy]# ls -l
total 28
-rw-r--r--    1 root    root          0 nov 10 19:48 ess
drwxr-xr-x    8 root    root        4096 oct 25 18:56 ftp
drwxr-xr-x    5 root    root        4096 jui 26 2000 httpd
drwx-----  42 jfc     jfc        4096 nov 10 17:53 jfc
drwx-----   4 504     504        4096 avr  9 2001 jl
drwx-----  10 502     502        4096 aoû 14 2000 multi
drwx-----   4 503     503        4096 nov 16 2000 pppuser
drwx-----  11 501     501        4096 oct 29 11:59 rc
```

Voici la liste des options les plus utilisées :

- `root_squash` : transforme les UID/GID 0 en UID/GID anonyme.
- `no_root_squash` : ne transforme pas les UID/GID 0 en UID/GID anonyme.
- `all_squash` : transforme les UID/GID de tous les utilisateurs en UID/GID anonyme. L'option inverse est `no_all_squash`.
- `squash_uid` et `squash_gid` : ces options précisent une liste d'UID et de GID qui sont convertis en utilisateur anonyme. Il est ainsi possible d'écrire `squash_uid=0-50,70-100`. De cette manière les UID de 0 à 50 et de 70 à 100 sont convertis en utilisateur anonyme.
- `anonuid` et `anongid` : ces options permettent de stipuler l'UID et le GID d'un utilisateur. Ces options sont particulièrement utiles lorsque des systèmes de fichiers sont exportés vers des clients Windows qui implémentent le protocole NFS. Dans ce cas, un PC est généralement dédié à un utilisateur. Il est alors nécessaire que cet utilisateur puisse manipuler ses fichiers en exploitant son identité UNIX. C'est ainsi que `anonuid=501` et `anongid=501` permettent de fixer l'identité de l'utilisateur.

6. Les options de montage.

La commande `mount` accepte un certain nombre d'arguments généraux ainsi que des arguments spécifiques au protocole NFS.

6.1. Les arguments généraux.

Les options de la commande `mount` sont préfixés par `-o`. Deux options peuvent s'avérer utiles : `ro` et `rw`. Elles permettent respectivement de monter un système de fichiers soit en lecture seule ou bien alors en lecture et en écriture. Bien que les indications d'accès aux systèmes de fichiers NFS soient définies par le contenu du fichier `/etc/exports`, un client a toujours la possibilité de restreindre ses accès.

Voici un exemple de session interactive consistant à monter en lecture seule le répertoire /home de la machine P100 bien que d'après le fichier `/etc/exports`, P200 y ait accès en lecture et en écriture :

```
[root@p200 /root]# mount -t nfs -o ro P100:/home /mnt/floppy
[root@p200 /root]# cd /mnt/floppy
[root@p200 floppy]# touch ess
touch: ess: Système de fichiers accessible en lecture seulement
```

6.2. Les arguments spécifiques.

Les informations échangées entre le serveur et les clients sont organisées en bloc dont la taille est un facteur déterminant de performance. Il est possible de contrôler indépendamment la taille des blocs lus par le client et la taille des blocs écrits par le client. Les options `rsize` et `wsize` contrôlent cet aspect de la communication. Par l'intermédiaire de ces options, le client et le serveur se mettent d'accord sur la taille des unités d'échange des informations. Ces tailles peuvent être d'au maximum 8192 ou 32768 bytes suivant les versions de NFS utilisées. Généralement une valeur de 8192 bytes est spécifiée.

Voici un exemple de la syntaxe utilisée pour spécifier ces options :

```
[root@p200 /root]# mount -t nfs -o rsize=8192,wsize=8192 P100:/home /mnt/floppy
```

Lorsqu'un serveur n'est pas opérationnel, le noyau des clients essaye indéfiniment de monter le système de fichiers spécifié. Afin d'empêcher ce comportement, il est permis d'utiliser l'option `soft`. Au bout d'un certain temps précisé par l'option `timeo`, la tentative de montage prend fin.

Voici un exemple de ce type de montage :

```
[root@p200 /root]# mount -t nfs -o rsize=8192,wsize=8192,soft,timeo=10
P100:/home /mnt/floppy
```

7. Le montage permanent.

Lors du démarrage du système, le fichier `/etc/fstab` est lu afin de monter les systèmes de fichiers qui y sont spécifiés. Il est permis de placer dans ce fichier, une entrée stipulant le montage d'un système de fichiers NFS lors du démarrage du système. Voici un exemple de fichier `fstab` :

```
[root@p200 /etc]# cat fstab
/dev/hda5          /                ext2    defaults    1 1
/dev/hda1         /boot            ext2    defaults    1 2
/dev/cdrom        /mnt/cdrom       iso9660 noauto,owner,ro 0 0
/dev/fd0          /mnt/floppy      auto    noauto,owner 0 0
none             /proc           proc    defaults    0 0
none             /dev/pts        devpts  gid=5,mode=620 0 0
/dev/hda6        swap            swap    defaults    0 0
192.168.1.1:/home /mnt/floppy      nfs     rw,rsize=8192,wsize=8192,soft 0 0
```

8. La résolution des liens symboliques.

L'utilisation de liens symboliques peut engendrer des problèmes lorsqu'un système de fichiers est exporté par NFS. Lorsqu'un processus référence un lien symbolique, un appel système est généré pour trouver l'emplacement du répertoire original. Dans le cas d'une exportation NFS, ce même comportement a lieu. Lorsque le client rencontre un lien symbolique, il demande au serveur le chemin d'accès vers le répertoire d'origine. L'information retournée par le serveur est alors interprétée par le client qui recherche le répertoire dans sa hiérarchie locale. Il est possible que le lien n'existe que sur le serveur et pas sur le client. Dans ce cas, la résolution du lien symbolique posera un problème car le fichier n'existe pas.

L'utilisation conjointe de NFS et des liens symboliques doit être étudiée avec attention. Il faut impérativement qu'un lien symbolique défini au niveau du serveur ait un sens au niveau des clients. Il faut pour cela, soit créer les liens symboliques au niveau de tous les clients, soit utiliser un schéma d'exportation ne brisant pas les liens symboliques.

VIII. NIS.

1. Introduction.

Le protocole NIS (Network Information Service) est un exemple de service d'annuaire. Pour comprendre d'utilité d'un tel service, prenons l'exemple du service des renseignements téléphoniques. Lorsque un abonné au téléphone souhaite connaître le numéro d'appel d'un correspondant, il peut exploiter le service des renseignements. L'appelant est un client du service d'annuaire alors que l'opérateur, qui se trouve à l'autre bout de la ligne, est considéré comme étant le serveur. L'opérateur consulte une base de données afin de répondre à la demande du client. Dans un système informatique, l'opérateur est remplacé par un programme qui consulte une base de données. La partie client est un ensemble de fonctions de bibliothèque pouvant contacter l'application distante ayant accès à la base de données. Le DNS qui a été étudié précédemment est un exemple de service d'annuaire.

Le protocole NIS a été développé au milieu des années 80 par la société Sun Microsystems désireuse de résoudre le problème de gestion d'un grand nombre de stations sous UNIX. Lorsque les fonctionnalités de TCP/IP ont été ajoutées aux systèmes UNIX, les administrateurs ont éprouvé des difficultés de gestion des fichiers `/etc/hosts` et `/etc/passwd`. Ajouter un système dans un réseau comptant plus de 100 hôtes et gérer les utilisateurs correspondants devenait problématique. Pour le fichier `/etc/hosts`, le DNS a été inventé et pour la gestion centralisée des informations relatives aux utilisateurs, NIS a été développé. A l'origine NIS était également appelé le service des Yellow Pages (YP)

Un problème majeur dans un environnement distribué comportant plusieurs systèmes UNIX est de maintenir de multiples copies des fichiers de mots de passe et de groupes. Dans une architecture distribuée, un utilisateur doit avoir la possibilité d'exploiter n'importe quelle machine du réseau pour effectuer son travail. Cela impose un partage à la fois des informations relatives à l'identification des utilisateurs mais aussi des données manipulées par les utilisateurs. Le partage des données des utilisateurs entre les différentes machines du réseau est réalisé au moyen de NFS. Le partage des informations d'authentification des utilisateurs est opéré au moyen du protocole NIS.

Le protocole NIS distribue une base de données qui remplace les informations locales concernant les utilisateurs. Au lieu de maintenir de multiples copies des fichiers `/etc/passwd` et `/etc/group`, l'administrateur doit uniquement maintenir, dans un état cohérent, une base de données localisée sur un serveur NIS. Les machines ayant besoin d'informations relatives aux utilisateurs doivent simplement se connecter au serveur NIS pour obtenir les renseignements souhaités. Le couple de services NIS et NFS permet de partager à la fois le répertoire `/home` et les données d'authentification entre les hôtes du réseau.

2. Les notions de maîtres, esclaves et clients.

Le service NIS est construit sur la base d'un modèle client-serveur. Un serveur NIS est un hôte qui contient les fichiers NIS appelés maps. Les clients sont des hôtes qui requièrent des informations de ces maps. Les serveurs sont en plus subdivisés en serveurs maîtres et serveurs esclaves. Un serveur maître est l'unique propriétaire des maps alors que les serveurs esclaves gèrent les requêtes en provenance des clients mais sans pour autant avoir la possibilité de modifier les maps. Le serveur maître est responsable de la maintenance et de la distribution des maps vers les esclaves. Cela signifie que la modification d'une map au niveau d'un serveur maître à une répercussion au niveau des esclaves.

Un client doit être configuré de manière à spécifier la source d'information à employer lors d'une requête. Cette configuration est effectuée par l'intermédiaire du fichier `/etc/nsswitch.conf`. Voici un exemple d'un tel fichier :

```
[root@P100 /etc]# cat nsswitch.conf
# Legal entries are:
#
#       nisplus or nis+      Use NIS+ (NIS version 3)
#       nis or yp           Use NIS (NIS version 2), also called YP
#       dns                 Use DNS (Domain Name Service)
#       files               Use the local files
#       db                  Use the local database (.db) files
#       compat              Use NIS on compat mode
#       hesiod               Use Hesiod for user lookups
#       [NOTFOUND=return]   Stop searching if not found so far
#
```

```
passwd: files nisplus nis
shadow: files nisplus nis
group: files nisplus nis
hosts: files nisplus nis dns
bootparams: nisplus [NOTFOUND=return] files
ethers: files
netmasks: files
networks: files
protocols: files
rpc: files
services: files
netgroup: nisplus
publickey: nisplus
automount: files nisplus
aliases: files nisplus
```

L'entrée hosts du fichier `/etc/nsswitch.conf` contient les valeurs files, nisplus, nis et dns. Lorsqu'une requête de conversion d'un nom d'hôte en adresse IP est requise, le fichier `/etc/hosts` est tout d'abord consulté. Si le fichier ne contient pas l'information recherchée, la recherche se poursuit au moyen des services NIS+ et NIS. En d'autres termes, un serveur NIS est contacté afin de pouvoir exploiter les informations présentes dans le fichier `/etc/hosts` du serveur. Si le serveur NIS ne possède pas l'information recherchée, c'est au serveur DNS d'intervenir. Une entrée du fichier `/etc/nsswitch.conf` stipule dans quel ordre les services de résolution de noms doivent être employés.

Le réseau d'une entreprise peut être composé de deux groupes de machines. Il peut en exister un groupe utilisé par les services administratifs de l'entreprise et un autre groupe par le service recherche et développement. Bien que ces deux groupes de machines soient interconnectés par le biais du même réseau, les politiques de partage d'informations entre les hôtes diffèrent suivant qu'une machine appartient à l'un ou à l'autre groupe. Pour refléter une différence potentielle de gestion des groupes de machines, NIS propose la notion de domaine. Un domaine est un ensemble de machines qui partagent un même ensemble de maps NIS.

3. Fondements de la gestion NIS.

La gestion NIS inclut la mise en place d'un serveur et la configuration des clients. La configuration d'un serveur NIS comporte les actions suivantes :

- installation d'un nouvel environnement NIS sur les serveurs maître et esclave ;
- lancement du démon ypserv permettant à un hôte d'agir en tant que serveur ;
- ajout de nouveaux serveurs esclaves en fonction de la croissance du réseau et des performances souhaitées.

La mise en œuvre des clients nécessite les tâches suivantes :

- modification des fichiers de configuration du système afin de permettre au client de tirer parti des avantages de NIS ;
- lancement du démon ypbind permettant à un client d'effectuer des requêtes NIS.

3.1. Installation du serveur NIS maître.

Pour qu'une machine puisse agir en tant que serveur, elle doit impérativement lancer les démons NIS au démarrage du système. Dans le cas d'un serveur NIS, les services à démarrer sont `/etc/rc.d/init.d/ypserv` et `/etc/rc.d/init.d/yppasswdd`.

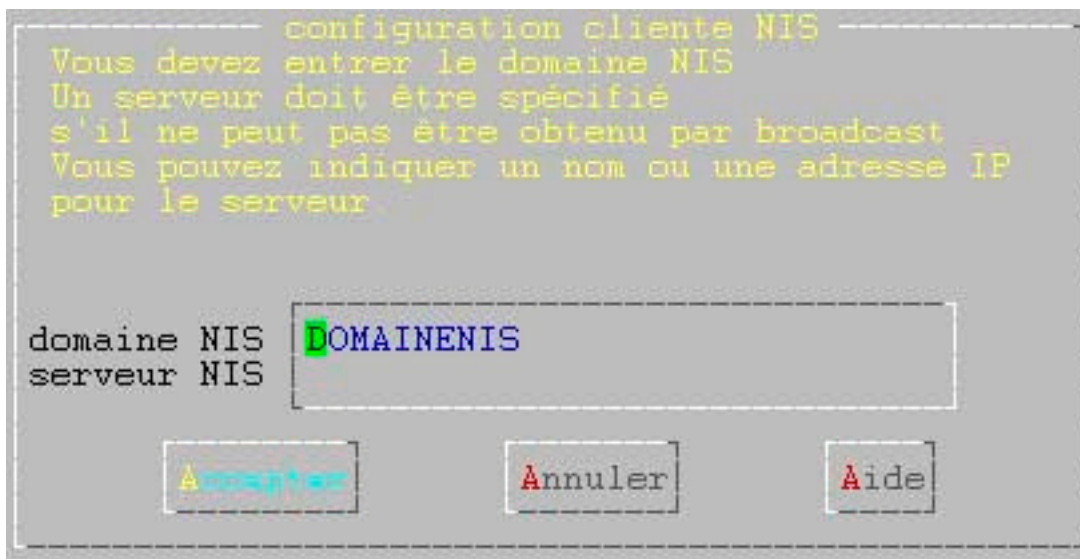
Tout serveur NIS doit gérer un domaine. Le nom de ce domaine peut être fixé au moyen de la commande `domainname`. Voici un exemple d'utilisation permettant de fixer le nom de domaine NIS à la valeur `domainenis` :

```
[root@P100 /root]# domainname DOMAINENIS
```

La commande `domainname` permet de fixer le nom de domaine tant que le système n'est pas redémarré. Une mémorisation permanente du nom de domaine nécessite la modification du fichier `/etc/sysconfig/network`. Il faut ajouter une entrée `NISDOMAIN` dans ce fichier comme le montre l'exemple suivant :

```
NETWORKING=yes
HOSTNAME="P100.challe.yi.org"
GATEWAY=212.68.198.209
FORWARD_IPV4=yes
GATEWAYDEV=eth0
NISDOMAIN=DOMAINENIS
```

Cette mémorisation du nom de domaine NIS peut également être réalisée au moyen de l'utilitaire linuxconf. Il faut alors choisir Configuration/Réseau/Tâches clientes/NIS. Après avoir effectué ce choix, une fenêtre permet d'encoder le nom de domaine NIS.



Mémorisation du nom de domaine NIS.

A présent, les maps peuvent être créées de la manière suivante :

```
[root@P100 sysconfig]# /usr/lib/yp/ypinit -m
```

L'option `-m` de la commande `ypinit` indique que l'administrateur souhaite configurer un serveur maître (master).

Si les fichiers ayant donné naissance aux maps sont modifiés, il faut recréer les maps. Pour cela, il suffit d'aller dans le répertoire `/var/yp` qui contient les maps. Ce répertoire contient également un fichier `Makefile` permettant de générer à nouveau les maps. Voici un exemple de recréation des maps :

```
[root@P100 yp]# make
```

3.2. Configuration des clients NIS.

Dès qu'un serveur NIS est configuré, il est permis de mettre en place des clients NIS qui interrogeront le serveur. Il ne faut jamais configurer des clients NIS aussi longtemps qu'il n'existe pas au moins un serveur NIS. En effet, la configuration d'un client sans serveur pourrait conduire à un plantage des machines clientes.

Aussi bien dans le cas du client que dans celui du serveur, il faut fixer le nom de domaine NIS à utiliser. Cette opération peut être réalisée soit :

- en utilisant la commande `domainname` ;
- en éditant le fichier `/etc/sysconfig/network` ;
- en utilisant `linuxconf`.

Dès que cette opération est réalisée, il faut lancer le client NIS par l'intermédiaire de la commande :

```
[root@P100 root]# /etc/rc.d/init.d/ybind start
```

Dès que le client NIS est opérationnel, les références aux fichiers d'administration sont gérées de deux manières selon la configuration du fichier `/etc/nsswitch.conf` :

- la base de données NIS remplace les copies locales de certains fichiers ;
- certains fichiers sont enrichis des informations provenant du serveur NIS.

Pour que l'impression d'enrichissement soit possible, il faut que le fichier `/etc/nsswitch.conf` précise que les copies locales des fichiers doivent être consultées avant d'exploiter la base de données NIS. C'est par exemple le cas pour les fichiers :

- `passwd` ;
- `shadow` ;
- `group` ;
- `hosts`.

Le partage par NIS des fichiers `passwd`, `shadow` et `group` permet à un utilisateur n'ayant pas de compte sur une machine de se connecter avec le nom d'utilisateur et le mot de passe qu'il possède au niveau du serveur NIS. En utilisant NIS conjointement avec le service NFS, des utilisateurs pourront se connecter à n'importe quelle machine et exploiter les données qu'ils possèdent dans leur répertoire.

Le partage du fichier `hosts` permet de remplacer, dans une certaine mesure, la notion de DNS. Les projets NIS et DNS ont été développés en parallèle. Cela signifie qu'au début de la conception de NIS, le service DNS n'existait pas. Le service NIS a dès lors été chargé de distribuer les informations du fichier `hosts` de sorte qu'une seule copie des informations soit nécessaire.

Le service NIS ne peut cependant pas remplacer complètement les fichiers locaux. En effet, le service NIS n'est pas lancé immédiatement au démarrage du système. Il faut donc que les copies locales des fichiers puissent fournir les renseignements nécessaires au démarrage du système avant que le service NIS soit lancé. Si le serveur NIS tombe en panne, il est nécessaire que les clients puissent toujours assurer un service minimum. L'existence des fichiers locaux permet de ne pas interrompre complètement les services de ces machines.

3.3. Ajout d'un serveur NIS esclave.

La configuration d'un serveur esclave consiste à mettre en place une machine qui obtient ses informations à partir d'un serveur et qui les fournit à ses clients. Un serveur esclave est donc à la fois un client et un serveur. Il faut donc configurer le serveur esclave comme une machine cliente.

Pour que le serveur esclave puisse distribuer des informations NIS, il faut qu'il possède une copie des bases de données présentes au niveau du serveur maître. La création des maps n'est donc pas basée sur les fichiers locaux mais sur les maps existants au niveau du serveur. La création des maps sur un serveur esclave nécessite la commande suivante :

```
[root@P133 root]# /usr/lib/yp/ypinit -s P100.challe.be
```

L'option `-s` de la commande `ypinit` indique qu'il faut créer des maps pour un serveur esclave.

Maintenant que les bases de données sont créées au niveau du serveur esclave, celui-ci doit être en mesure de répondre à des requêtes en provenance des clients. Il faut donc lancer les services serveurs de NIS comme dans le cas d'un serveur maître.

Lorsqu'un client NIS se présente sur le réseau, il obtient la liste des serveurs pouvant répondre à ses questions. A ce moment, le serveur maître ne diffuse pas l'information indiquant la présence d'une nouvelle machine serveur. Il faut ajouter à cette liste le nom du nouveau serveur en éditant le fichier `/var/yp/ypservers` du serveur maître comme le montre la session interactive suivante :

```
[root@P100 yp]# cat ypservers
P100.challe.be
P133.challe.be
[root@P100 yp]# make
```

La liste des serveurs disponibles fait partie de la base de données NIS. Une modification du fichier ypservers doit donc être suivie d'une reconstruction des maps au moyen de la commande make.

4. Les commandes-clients.

Les utilisateurs de systèmes fondés sur NIS ont la possibilité d'interagir avec la base de données au moyen de différentes commandes.

4.1. ypmatch.

La commande ypmatch est une commande grep pour les maps NIS. Cette commande recherche une chaîne dans une map et imprime les données correspondantes.

Le premier paramètre de la commande ypmatch est la chaîne à rechercher et le second paramètre est le nom de la map dans laquelle il faut effectuer la recherche. Voici un exemple d'utilisation de la commande ypmatch :

```
[jfc@p200 jfc]$ ypmatch jfc passwd
jfc:$1$RTbs4X0mVLTKq2A0CQnysr0:500:500:Jean-Francois Challe:/home/jfc:/bin/bash
[jfc@p200 jfc]$ ypmatch isec1 hosts
212.68.194.200 isec1
[jfc@p200 jfc]$
```

4.2. ypcat.

La commande ypcat est l'équivalent de la commande cat. Cette commande permet de visualiser le contenu d'une map. Le paramètre de la commande est donc le nom d'une map. Voici un exemple d'exécution de la commande ypcat :

```
[jfc@p200 jfc]$ ypcat hosts
127.0.0.1      P100.challe.be  P100      localhost.localdomain  localhost
212.68.194.201 isec2
127.0.0.1      P100.challe.be  P100      localhost.localdomain  localhost
127.0.0.1      P100.challe.be  P100      localhost.localdomain  localhost
212.68.194.203 isec4
212.68.245.203 jmb
212.68.194.200 isec1
127.0.0.1      P100.challe.be  P100      localhost.localdomain  localhost
212.68.198.209 gateway
212.68.194.202 isec3
```

4.3. ypwhich.

La commande ypwhich permet de connaître l'adresse IP du serveur NIS actuellement employé par un client. Voici un exemple d'utilisation de cette commande :

```
[jfc@p200 jfc]$ ypwhich
192.168.1.1
```

La commande ypwhich avec l'option -x permet de connaître l'ensemble des maps disponibles :

```
[jfc@p200 jfc]$ ypwhich -x
Use "ethers"      for map "ethers.byname"
Use "aliases"     for map "mail.aliases"
Use "services"   for map "services.byname"
Use "protocols"  for map "protocols.bynumber"
Use "hosts"       for map "hosts.byname"
Use "networks"   for map "networks.byaddr"
Use "group"       for map "group.byname"
Use "passwd"      for map "passwd.byname"
```

4.4. yppasswd.

Lorsqu'un utilisateur souhaite changer de mot de passe, il ne doit plus utiliser la commande passwd mais la commande yppasswd pour que les modifications soient effectuées au niveau du serveur.

IX. LPD.

1. Introduction.

Les imprimantes connectées à un système Linux sont généralement branchées par l'intermédiaire du port parallèle. Chaque port parallèle du système correspond à un device dans le répertoire /dev. Le premier port parallèle est /dev/lp0, le second /dev/lp1 et le troisième /dev/lp2.

Si une imprimante est branchée sur le premier port parallèle (/dev/lp0), il est possible de déclencher l'impression en envoyant directement des informations vers le périphérique. Voici un exemple d'une telle opération :

```
[root@P100 /root]# cat ess.c > /dev/lp0
```

Procéder de la sorte pour imprimer un document est de loin la plus mauvaise méthode. Durant tout le processus d'impression, l'utilisateur n'a pas la possibilité d'effectuer d'autres travaux. De plus, d'autres utilisateurs n'auront pas la possibilité d'imprimer tant que le port est en cours d'utilisation par un autre processus.

Comme la majorité des systèmes d'exploitation dignes de ce nom, Linux possède la notion de spool d'impression. Cela permet aux utilisateurs de soumettre leurs travaux au spooler qui se chargera d'imprimer le document correctement. Cette technique permet un partage de la ressource d'impression sans y imposer un accès exclusif.

Sous UNIX, le démon LPD (Line Printer Daemon) est le serveur d'impression pour les utilisateurs locaux et distants. Il gère les différentes imprimantes ainsi que leurs files d'attente respectives.

2. Le fichier /etc/printcap.

Lorsque LPD est lancé, il lit le fichier /etc/printcap pour découvrir les imprimantes existantes. Ce fichier définit les imprimantes et leurs caractéristiques. Configurer ce fichier est la partie la plus délicate du travail car l'analyseur qui exploite ce fichier est très tatillon et la syntaxe des paramètres du fichier est particulièrement obscure. La plupart des problèmes de syntaxe et de lecture du fichier peuvent être évités en suivant quelques règles :

- commencer chaque entrée par un nom d'imprimante à la première colonne du fichier. Aucun espace ne doit précéder le premier nom d'une imprimante. Plusieurs noms d'imprimante peuvent être utilisés à la condition qu'ils soient séparés par le caractère |. L'une des entrées doit absolument porter le nom lp. Si plusieurs imprimantes sont connectées, ce nom doit être affecté à l'imprimante par défaut ;
- les entrées du fichier printcap peuvent être rédigées sur plusieurs lignes à la condition que soit utilisé le caractère d'échappement \ à la fin de chaque ligne et en commençant toute nouvelle ligne par une tabulation. Aucun espace ne doit se trouver avant le caractère \. De plus, le caractère suivant \ doit obligatoirement être un retour de chariot ;
- tout champ, autre que le nom de l'imprimante, commence et se termine par le caractère deux-points. Le caractère précédent \ ou suivant une tabulation doit être le caractère deux-points.

Les paramètres de configuration utilisés dans le fichier /etc/printcap décrivent les caractéristiques de l'imprimante. Le démon LPD a besoin de connaître ces caractéristiques afin de communiquer avec l'imprimante. Les paramètres, codés sur deux caractères, peuvent généralement être affectés d'une valeur. La syntaxe des paramètres varie légèrement en fonction du type de la valeur affectée. Les paramètres sont de trois types :

- booléen : toutes les valeurs de ce type sont par défaut à faux. Spécifier une valeur booléenne active sa fonction. Ces paramètres sont spécifiés en utilisant leur nom. Par exemple, :rc: active les mesure de sécurité pour les utilisateurs distants ;
- numérique : il est permis d'affecter une valeur numérique à certains paramètres. Par exemple, :br#9600 fixe la vitesse de communication avec une imprimante série ;
- chaîne de caractères : certains paramètres peuvent utiliser une valeur sous la forme d'une chaîne de caractères. Par exemple, :rp=laserjet: définit le nom d'une imprimante distante.

Les serveurs d'impression n'ont en général qu'une ou deux imprimantes directement connectées. Toutes les autres définies dans le fichier /etc/printcap sont probablement des imprimantes distantes.

Voici un exemple de fichier `/etc/printcap` :

```
[root@P100 /etc]# cat printcap
seikosha|lp:\
    :sd=/var/spool/lpd/seikosha:\
    :mx#0:\
    :sh:\
    :lp=/dev/lp0:
hp4:\
    :sd=/var/spool/lpd/hp4:\
    :mx#0:\
    :sh:\
    :rm=192.168.1.192:\
    :rp=:
```

Dans cet exemple, l'imprimante par défaut seikosha est connectée au port lpt1 de l'ordinateur alors que l'imprimante hp4 est distante. Le paramètre `sd` définit l'emplacement du répertoire destiné à la mémorisation des travaux d'impression. Ce répertoire doit appartenir au super-utilisateur du système. Le paramètre `mx` fixe la taille maximale des fichiers qui peuvent être imprimés. La valeur 0 indique qu'aucune limite n'est fixée. Dans le cas où une valeur différente de zéro est indiquée, elle est exprimée en kilo bytes. Le paramètre `sh` indique que les pages d'en-tête ne doivent pas être imprimées. En l'absence de ce paramètre booléen, chaque impression est précédée d'une page d'en-tête contenant le nom de l'utilisateur qui a lancé l'impression du document. Dans le cas d'une imprimante locale, il faut spécifier le périphérique à employer. Le paramètre `lp` prend pour valeur une chaîne de caractères indiquant le chemin d'accès à la ressource d'impression. Dans le cas d'une imprimante distante, il faut préciser le nom ou l'adresse IP de la machine qui possède l'imprimante. Dans cet exemple, l'imprimante HP laserjet 4 Mplus dispose d'un kit jet direct émulant un grand nombre de protocoles dont LPD. Le démon LPD de la machine UNIX est alors capable d'envoyer les travaux d'impression au démon LPD de l'imprimante. Le paramètre `rp` permet d'indiquer le nom de l'imprimante distante à utiliser.

3. Sécurité avec LPD.

Lorsqu'un hôte ne dispose pas d'imprimante, il peut être intéressant de le configurer pour qu'il puisse exploiter le service d'impression d'une autre machine. Le fichier `/etc/printcap` du client doit refléter l'utilisation d'une imprimante distante. En supposant que l'hôte P200.challe.be soit un client du service d'impression de l'hôte P100.challe.be, voici le fichier `/etc/printcap` de l'hôte P200.challe.be :

```
[root@p200 /etc]# cat printcap
lp:\
    :sd=/var/spool/lpd/lp:\
    :mx#0:\
    :sh:\
    :rm=P100.challe.be:\
    :rp=seikosha:
```

Le paramètre `rm` précise que le serveur d'impression est P100.challe.be alors que le paramètre `rp` définit le nom de l'imprimante à employer au niveau du serveur. Bien que l'hôte p200.challe.be utilise la file d'impression d'un serveur, il faut néanmoins y définir un répertoire pour la file d'attente des travaux d'impression. Cette configuration permet au client d'accepter des demandes d'impression même lorsque le serveur n'est pas disponible. Il faut donc créer le répertoire `/var/spool/lpd/lp` au niveau du client au moyen de la commande suivante :

```
[root@p200 /etc]# mkdir /var/spool/lpd/lp
```

Si un travail d'impression est soumis au serveur, il refusera d'effectuer l'opération car le démon LPD utilise le modèle de sécurité de machine fiable. Il est courant de permettre le partage d'une imprimante sans accorder aucun privilège au serveur d'impression. Pour cela, LPD utilise le fichier `/etc/hosts.lpd`. Une machine digne de confiance, déclarée dans ce fichier ne peut accéder qu'aux imprimantes. Voici le contenu du fichier `/etc/hosts.lpd` du serveur d'impression :

```
[root@P100 /etc]# cat hosts.lpd
P200.challe.be
```

4. Utilisation de LPD.

4.1. Envoyer une requête d'impression.

Les documents à imprimer sont envoyés au démon d'impression via la commande `lpr` (Line Printer Remote). La commande `lpr` possède un grand nombre d'arguments, mais en général, on donne juste le nom de l'imprimante ainsi que le fichier à imprimer. Voici un exemple d'envoi d'une requête d'impression vers l'imprimante `hp4` :

```
[root@P100 /etc]# lpr -P hp4 printcap
```

Sans le paramètre `-P`, la commande `lpr` envoie la requête d'impression vers l'imprimante par défaut portant le nom `lp`.

4.2. Gestion du démon d'impression.

La commande `lpc` (Line Printer Control Program) est utilisée par l'administrateur pour contrôler le fonctionnement du système d'impression. Pour chaque imprimante configurée dans le fichier `/etc/printcap`, la commande `lpc` peut être employée pour :

- activer ou désactiver une imprimante ;
- activer ou désactiver une file d'impression ;
- changer l'ordre des requêtes dans la file d'attente ;
- déterminer le statut des imprimantes.

Sans argument précisé, la commande `lpc` entre en mode interactif. Par contre, si des arguments sont spécifiés, la commande `lpc` interprète le premier comme étant une commande et les autres comme étant des paramètres. Voici la liste des commandes reconnues par `lpc` :

- `abort {all | printer }` : arrête le démon d'impression immédiatement ;
- `clean {all | printer }` : efface les fichiers temporaires qui ne peuvent pas être imprimés ;
- `disable {all | printer }` : désactive une file d'attente. A ce moment, plus aucuns travaux ne peuvent être soumis à cette file via la commande `lpr` ;
- `down {all | printer } message` : désactive une file d'attente, désactive l'impression et place le message dans le fichier `status` correspondant. Cela permet d'informer les utilisateurs des raisons pour lesquelles l'impression est désactivée ;
- `enable {all | printer }` : active la file d'attente. Cela permet à la commande `lpr` de soumettre des requêtes ;
- `exit` : termine une session interactive ;
- `restart {all | printer }` : essaie de relancer le démon d'impression ;
- `start {all | printer }` : active l'impression et relance la file d'attente ;
- `status {all | printer }` : affiche le statut des démons et des files d'attente ;
- `stop {all | printer }` : désactive un démon après que la requête en cours d'impression soit terminée ;
- `topq printer [jobnum ...] [user ...]` : place une requête en tête de la file d'attente ;
- `up {all | printer }` : relance tout.

4.3. Consultation de la file d'attente.

La commande `lpq` (Line Printer Queue) examine le répertoire contenant la file d'attente et affiche l'état des requêtes spécifiées ou de tous les travaux associés à un utilisateur. Si `lpq` est appelé sans argument, elle retourne l'état de tous les travaux dans la file d'attente.

Cette commande accepte plusieurs options :

- `-P` : indique une imprimante particulière. Sans cette options, l'imprimante par défaut est consultée. Tous les autres arguments sont interprétés comme des noms d'utilisateurs ou des numéros de travaux ;
- `-l` : toute l'information disponible sur un travail est affichée.

Voici un exemple d'utilisation de la commande `lpq` :

```
[root@P100 /root]# lpq
lp is ready and printing
Rank  Owner      Job  Files          Total Size
1st   jfc         29  ex.doc        23552 bytes
1st   jfc         28  ess.c         53 bytes
```

4.4. Suppression d'un requête.

La commande lprm (Line Printer ReMove) supprime des travaux de la file d'attente d'une imprimante. La commande lprm accepte plusieurs options :

- -P :indique quelle imprimante est concernée. Sans cela, c'est l'imprimante par défaut qui est employée ;
- - : un seul signe moins permet de supprimer toutes les requêtes appartenant à un utilisateur. Si l'utilisateur est root, tous les travaux sont supprimés ;
- utilisateur : la commande lprm tente de supprimer tous les travaux de l'utilisateur ou des utilisateurs spécifiés. Cette option n'est exploitable que par le super-utilisateur ;
- job # : un utilisateur peut supprimer des travaux de manière individuelle en précisant leur numéro.

Voici un exemple d'utilisation de cette commande :

```
[root@P100 /root]# lprm -
dfa028P100.challe.be dequeued
cfa028P100.challe.be dequeued
dfa029P100.challe.be dequeued
cfa029P100.challe.be dequeued
```

X. DHCP.

1. Introduction.

Le protocole DHCP (Dynamic Host Configuration Protocol) permet de configurer dynamiquement des clients. Lorsqu'un hôte doit être connecté à un réseau, il faut configurer :

- l'adresse IP ;
- le masque de sous-réseau ;
- l'adresse de diffusion ;
- l'adresse de la passerelle ;
- l'adresse du serveur DNS ;
- le nom de la machine.

Le protocole DHCP permet d'affecter une adresse de trois manières :

- allocation manuelle. L'administrateur garde un contrôle total sur les adresses en les affectant de manière spécifique aux clients ;
- allocation automatique. Le serveur DHCP affecte de façon permanente une adresse à partir d'un ensemble d'adresses disponibles. L'administrateur n'intervient pas de façon précise dans l'affectation d'une adresse à un client ;
- allocation dynamique. Le serveur affecte une adresse à un client DHCP pour une durée limitée. Cette durée de vie limitée de l'adresse est appelée un bail. Le client peut renvoyer l'adresse au serveur à n'importe quel moment. Le serveur réclame automatiquement l'adresse après l'expiration du bail.

L'allocation dynamique est utile dans les réseaux de grande taille où un grand nombre de machines sont ajoutées et supprimées. Les adresses inutilisées retrouvent l'ensemble des adresses libres sans que les utilisateurs ou l'administrateur n'aient à intervenir. L'allocation dynamique permet à un réseau d'avoir une utilisation maximale d'un ensemble réduit d'adresses. Cela est particulièrement recommandé dans le cas de systèmes mobiles qui passent d'un sous-réseau à l'autre et qui doivent donc se voir réaffecter une nouvelle adresse à chaque endroit où ils sont connectés.

L'allocation dynamique d'adresses ne fonctionne pas sur tous les systèmes (les serveurs de noms, de courrier électronique, etc ...). En d'autres termes, tous les systèmes accessibles au moyen d'une résolution de nom ne peuvent recevoir une adresse dynamiquement. Ces systèmes doivent donc être configurés manuellement.

L'affectation d'adresse dynamique a principalement une répercussion sur le DNS. Le DNS est nécessaire pour faire correspondre un nom de machine à une adresse IP. Il ne peut pas jouer son rôle si les adresses IP changent sans arrêt sans qu'il ne soit tenu au courant de ces modifications. Pour faire en sorte que l'affectation d'adresses dynamiques fonctionne avec tous les systèmes, il est nécessaire d'avoir un nouveau type de DNS qui puisse être mis à jour dynamiquement par le serveur DHCP. Une norme de DNS dynamique est actuellement à l'étude.

2. Le fichier dhcpd.conf.

La configuration du serveur DHCP figure dans le fichier /etc/dhcpd.conf. Ce fichier de configuration contient les instructions indiquant au serveur quels sous-réseaux et quelles machines il sert ainsi que les informations de configuration qu'il leur fournit.

2.1. Configuration de base.

Le fichier /etc/dhcpd.conf est constitué de deux parties, une définissant les paramètres globaux et l'autre définissant des paramètres spécifiques à un sous-réseau.

Voici un exemple de fichier /etc/dhcpd.conf :

```
default-lease-time 86400;
get-lease-hostnames true;
option subnet-mask 255.255.255.0;
option domain-name "challe.be";
option domain-name-servers 192.168.1.1;
option lpr-servers 192.168.1.192;
option interface-mtu 1500;
subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    range 192.168.1.200 192.168.1.240;
}
```

L'option `default-lease-time` indique au serveur, la durée en seconde d'un bail d'adresse. Lorsque le nombre de secondes précisé expirera, l'adresse sera récupérée par le serveur. L'option `get-lease-hostnames` demande au serveur DHCP de fournir un nom de machine chaque fois qu'un client reçoit une adresse dynamique. De plus, le nom de machine est obtenu via le DNS. Ce paramètre est un booléen. S'il est positionné à faux, ce qui est la valeur par défaut, le client reçoit une adresse mais pas de nom. Rechercher le nom de machine pour toute adresse dynamique possible allonge sensiblement le temps de réponse à une demande d'attribution d'une adresse.

Les cinq lignes suivantes du fichier `/etc/dhcpd.conf` définissent des valeurs qui seront utilisées par les clients pour mener à bien leur processus de configuration. La signification de ces options est clairement indiquée par leur nom. Dans cet exemple, les clients recevront :

- le masque de sous-réseau 255.255.255.0 ;
- le nom de domaine challe.be ;
- l'adresse du serveur DNS 192.168.1.1
- l'adresse du serveur d'impression 192.168.1.192 ;
- le MTU de 1500.

L'instruction `subnet` définit le réseau que le serveur DHCP va servir. L'identité de chaque réseau est déterminée par l'adresse et le masque du réseau. Ces deux éléments sont requis par cette instruction. Les options et les paramètres définis dans une instruction `subnet` ne s'appliquent qu'au sous-réseau et à ses clients. Les options utilisées dans cet exemple indiquent aux clients, l'adresse du router et l'adresse de diffusion à utiliser.

Le paramètre `range` définit l'intervalle des adresses disponibles pour l'allocation dynamique d'adresses. Ce `range` se trouve toujours associé à une instruction `subnet`. L'intervalle de ce paramètre est défini par les adresses de début et de fin. La première est l'adresse la plus petite qui puisse être automatiquement affectée et la seconde est l'adresse la plus grande pouvant être automatiquement affectée.

Une instruction `subnet` peut contenir plusieurs clauses `range`. Cela permet d'adapter la configuration du serveur DHCP à la structure d'un réseau déjà existant. Dans un grand nombre de sites, un serveur DHCP est ajouté à une configuration existante. Dans ce cas, les adresses fixes ont déjà été attribuées. Le protocole DHCP doit donc s'adapter à l'existant.

2.2. Lancement du serveur DHCP.

Le script `/etc/rc.d/init.d/dhcpd` permet de lancer automatiquement le serveur DHCP lors du démarrage du système. Ce script contient simplement la commande de lancement du démon `dhcpd`. Par défaut, ce programme tente de servir tous les réseaux auxquels la machine est connectée. Dans le cas d'un serveur équipé de plusieurs interfaces de connexion à un réseau, cela peut avoir des répercussions désastreuses. Supposons qu'une machine soit équipée de deux cartes Ethernet, l'une connectée à Internet et l'autre à un réseau local. Le démon DHCPD va donc tenter de servir à la fois le réseau local mais également Internet. Avec la configuration de l'exemple précédent, le lancement du démon DHCPD échouera car aucune instruction `subnet` ne définit la gestion de la partie internet. Une solution simple au problème consiste à ajouter une nouvelle clause `subnet` pour gérer la partie Internet. Une telle configuration peut rendre la connexion Internet inutilisable. En effet, le serveur DHCP se substitue au serveur DHCP du fournisseur d'accès à Internet, ce qui peut naturellement perturber un certain nombre de clients. En réaction, le fournisseur d'accès risque de désactiver la connexion Internet de tout client faisant tourner un serveur DHCP. Afin d'activer le serveur DHCP pour le réseau local, il suffit de modifier le script `/etc/rc.d/init.d/dhcpd` en spécifiant l'interface réseau qui doit être gérée. Voici un exemple de modification du fichier `/etc/rc.d/init.d/dhcpd` :

```
[root@P100 init.d]# cat dhcpd
. /etc/rc.d/init.d/functions
. /etc/sysconfig/network
[ ${NETWORKING} = "no" ] && exit 0
[ -f /usr/sbin/dhcpd ] || exit 0
[ -f /etc/dhcpd.conf ] || exit 0
RETVAL=0
case "$1" in
  start)
    echo -n "Starting dhcpd: "
    daemon /usr/sbin/dhcpd eth1
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dhcpd
    ;;
  stop)
    echo -n "Shutting down dhcpd: "
    killproc dhcpd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/dhcpd
    ;;
  restart|reload)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
  status)
    status dhcpd
    RETVAL=$?
    ;;
  *)
    echo "Usage: dhcpd {start|stop|restart|status}"
    exit 1
esac
exit $RETVAL
```

Dans cet exemple, la commande de lancement du serveur DHCP possède le paramètre eth1 afin de préciser au démon qu'il ne doit gérer que cette interface.

2.3. Utilisation des adresses physiques.

Un serveur DHCP peut être configuré pour attribuer une adresse IP spécifique à un hôte précis. Chaque interface réseau possède une adresse physique univoque appelée adresse MAC. Sur base de cette adresse fixe, le serveur DHCP peut retourner une adresse IP qui sera toujours la même. De cette manière, un client DHCP peut obtenir une adresse IP fixe. Voici un exemple de configuration d'une adresse IP fixe :

```
[root@P100 jfc]# cat dhcpd.conf
get-lease-hostnames true;
option subnet-mask 255.255.255.0;
option domain-name "challe.be";
option domain-name-servers 192.168.1.1;
option lpr-servers 192.168.1.192;
option interface-mtu 1500;
host P2366 {
    hardware ethernet 00:60:97:8E:8A:A6;
    fixed-address 192.168.1.80;
    option routers 192.168.1.1;
}
```

Le problème posé par ce type de configuration consiste en la connaissance des adresses MAC. La commande arp permet de connaître la liste de toutes les adresses MAC. Voici un exemple d'exécution de cette commande :

```
[root@P100 /etc]# arp
Adresse          TypeMap AdresseMat      Indicateurs      Iface
powerpc.challe.be ether    00:05:02:B4:5F:5B C                  eth1
P3450.challe.be ether    00:00:C0:F8:B2:A4 C                  eth1
P2450.challe.be ether    00:10:4B:DC:A2:8E C                  eth1
P200.challe.be ether    00:80:C8:C1:3A:78 C                  eth1
P133.challe.be ether    00:00:C0:64:21:67 C                  eth1
P2300.challe.be ether    00:00:C0:E1:20:67 C                  eth1
gateway          ether    00:00:77:95:5D:54 C                  eth0
P166PLUS.challe.be ether    00:00:C0:BC:F1:A3 C                  eth1
P2366.challe.be ether    00:60:97:8E:8A:A6 C                  eth1
HP4.challe.be ether    00:60:B0:13:6F:D5 C                  eth1
```


XI. Samba.

1. Introduction.

Samba est une suite d'applications UNIX utilisant le protocole SMB (Server Message Block). La plupart des systèmes d'exploitation, comme Windows et OS/2, prennent en charge la communication réseau à l'aide de SMB. Grâce à ce protocole, Samba permet à des serveurs Unix de communiquer avec des produits Microsoft Windows. Ainsi, une machine UNIX dotée de Samba peut elle être vue comme un serveur de réseau Microsoft, en proposant les services suivants :

- le partage d'un ou de plusieurs systèmes de fichier;
- le partage d'imprimantes installées sur le serveur et sur ses clients ;
- l'assistance des clients dans l'exploration à l'aide du voisinage réseau ;
- la prise en charge de l'authentification de clients se connectant à un domaine Windows ;
- la prise en charge de la résolution de noms WINS.

La suite Samba s'articule autour de deux démons UNIX fournissant des ressources partagées à des clients SMB du réseau. Ces démons s'appellent :

- `smbd` : qui permet le partage de fichiers et d'imprimantes sur un réseau SMB et prend en charge l'authentification et les droits d'accès des clients SMB ;
- `nmbd` : qui est associé au service WINS et prend en charge la résolution de noms.

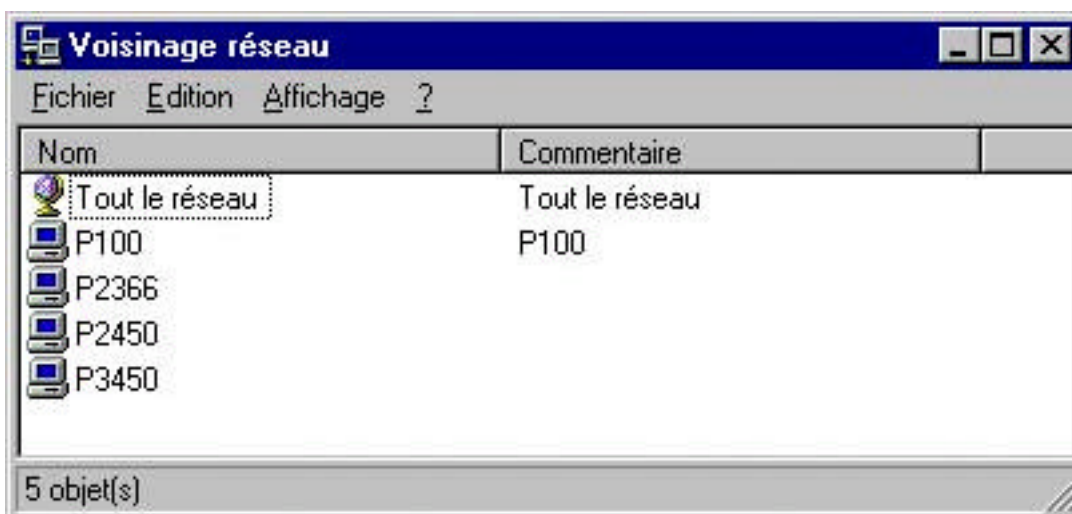
Plusieurs raisons peuvent conduire à la mise en œuvre d'un serveur Samba, elles sont :

- la recherche des fonctionnalités offertes par un serveur Windows NT sans être dans l'obligation d'en acheter un ;
- le souhait de créer un espace commun pour les données des utilisateurs afin de remplacer un serveur Windows par un serveur UNIX ou inversement ;
- le souhait de partager des imprimantes entre des stations de travail sous Windows et sous UNIX ;
- la nécessité d'accéder à des fichiers résidant sous Windows depuis un serveur UNIX.

Supposons dans la suite de cette section que l'on dispose d'un réseau où la machine de nom P100 est un serveur Samba faisant partie du groupe PRIVE.

1.1. Partage d'un service disque.

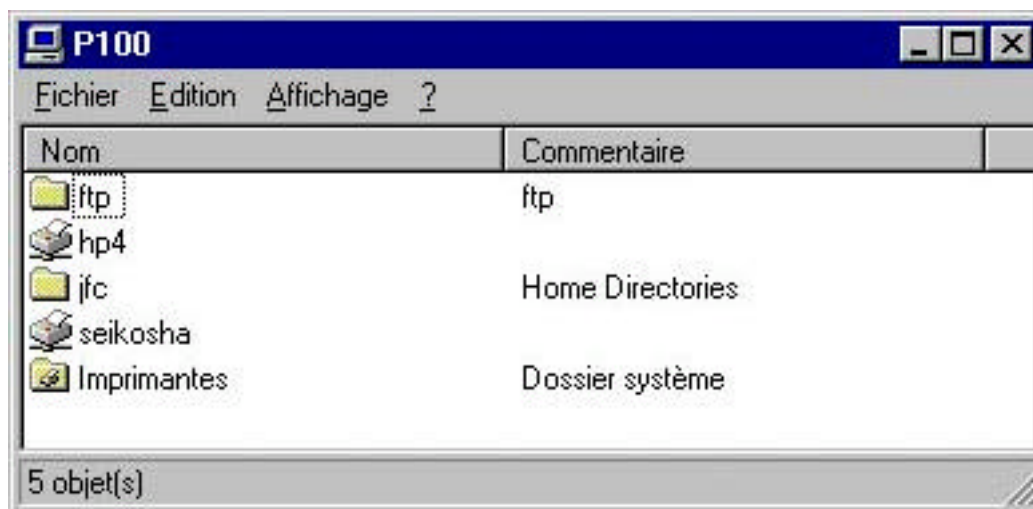
Si tout est parfaitement configuré, le serveur Samba, P100, est visible dans le voisinage réseau d'une machine cliente. L'outil voisinage réseau affiche les noms de tous les ordinateurs appartenant au groupe PRIVE.



Liste des ordinateurs du groupe PRIVE.

Sous Windows, les noms d'hôtes combinent des majuscules et des minuscules. La casse n'a pas d'importance pour les noms d'hôtes : dans les écrans d'affichage et les résultats des commandes, P100 et p100 par exemple, font référence au même ordinateur.

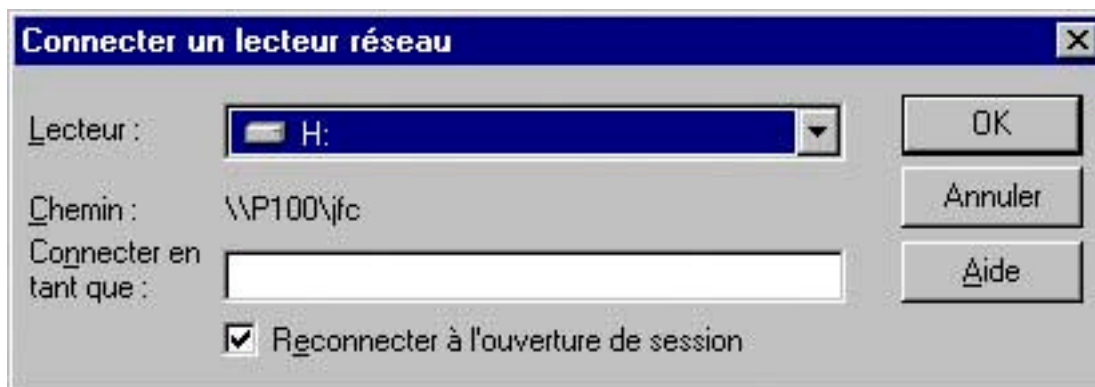
En cliquant deux fois sur l'icône P100, Windows affiche les ressources partagées appartenant à ce serveur. Cette action établit un partage avec le serveur et lui demande la liste des partages proposés, c'est-à-dire les ressources de fichiers et les ressources d'imprimantes. Dans le cas présent, le serveur compte les imprimantes hp4 et seikosha ainsi que les partages de disques ftp et jfc.



Ressources disponibles sur le serveur P100.

Cet exemple montre que grâce à Samba, un client Windows considère le serveur UNIX comme étant un serveur SMB. Il est présenté sous la forme d'une ressource accessible comme n'importe quelle autre.

L'option connecter un lecteur réseau de l'explorateur Windows permet d'attribuer une lettre de lecteur à un répertoire réseau recensé. La lettre de lecteur devient un raccourci pratique auquel se réfèrent les applications pour accéder au dossier réseau concerné. Il est permis d'y stocker des données, d'y installer et d'y lancer des programmes.



Connexion du lecteur réseau jfc du serveur P100.

Le paramètre Chemin de la boîte de dialogue «connecter un lecteur réseau» est une autre manière de représenter un répertoire d'une machine réseau. Dans le monde Windows, cette notation obéit à la convention UNC (Universal Naming Convention).

Une fois le lecteur réseau défini, Windows et ses logiciels se comportent comme si le répertoire était un disque dur local. Cette unité peut alors recevoir les applications acceptant un fonctionnement multi-utilisateurs. L'illustration suivante montre l'unité réseau comme si elle faisait partie des périphériques de stockage appartenant au client. L'icône particulière du lecteur H : indique qu'il s'agit d'une unité réseau et non pas d'une unité locale.



Répertoire réseau associé à la lettre de lecteur H.

1.2. Partage d'une imprimante.

On peut remarquer que l'imprimante seikosha apparaît sous les partages disponibles du serveur P100. Cela signifie que le serveur UNIX possède une imprimante pouvant être partagée par les différents clients SMB du groupe de travail. Les requêtes qu'envoient les clients à l'imprimante sont dirigées dans une file d'attente d'impression du serveur UNIX et sont traitées dans l'ordre de réception.

Du côté Windows, la configuration d'une imprimante gérée par Samba est plus aisée que celle d'un partage de disque. Pour installer le pilote d'une imprimante sur le client réseau, il suffit de cliquer deux fois sur l'icône de l'imprimante puis d'identifier le constructeur et le modèle appropriés. Windows est alors capable de formater correctement toute information envoyée à l'imprimante réseau, puis d'accéder à cette dernière comme s'il s'agissait d'une imprimante locale. L'imprimante réseau est identifiée par une icône particulière dans la fenêtre Imprimantes de Windows.



Imprimantes disponibles sur un client Windows.

1.3. Visualisation des ressources sous UNIX.

Samba est considéré par UNIX comme un ensemble de démons. Les messages qu'ils génèrent peuvent être consultés dans les fichiers de log. Le serveur SMB est configuré à partir du fichier /etc/smb.conf. Pour connaître l'action des démons, il suffit d'employer la commande smbstatus :

```
[root@P100 /root]# smbstatus

Samba version 2.0.6
Service      uid      gid      pid      machine
-----
jfc          jfc      jfc      1760     p2366    (192.168.1.80) Wed Oct 24 15:17:18 2001
ftp          jfc      jfc      1760     p2366    (192.168.1.80) Wed Oct 24 15:18:22 2001
jfc          jfc      jfc      2001     p2450    (192.168.1.40) Wed Oct 24 15:53:41 2001

No locked files

Share mode memory usage (bytes):
  1048464(99%) free + 56(0%) used + 56(0%) overhead = 1048576(100%) total
```

1.4. Prise en main d'un réseau SMB.

1.4.1. Maîtrise de NetBIOS.

En 1984, IBM développa une interface de programmation simple pour la gestion de ses ordinateurs, appelée NetBIOS (Network Basic Input/Output). L'interface NetBIOS garantissait aux applications, une structure rudimentaire de connexion et de partage des données avec d'autres ordinateurs.

L'interface de programmation NetBIOS peut être considérée comme un ensemble d'extensions réseau au BIOS standard. Avec le BIOS, tous les appels de bas niveau s'appliquent directement aux couches matérielles de l'ordinateur. Cependant, NetBIOS devait initialement échanger des instructions avec des ordinateurs en réseaux. L'acheminement des demandes entre ordinateurs nécessitait donc un protocole de transport de bas niveau.

En 1985, IBM définissait ce protocole, intégré à l'interface de programmation NetBIOS et baptisé NetBEUI (NetBIOS Extended User Interface). NetBEUI est destiné aux réseaux locaux de petite taille et permet d'attribuer un nom unique d'une longueur maximale de 15 caractères à chaque ordinateur. Par réseau de petite taille, on entend réseau de moins de 255 machines, considéré en 1985, comme une limite absolue.

Le protocole NetBEUI était très répandu parmi les applications réseau, il en était de même pour celles s'exécutant sous Windows for Workgroups. Une version NetBIOS sur protocole IPX fut ensuite proposée. Par la suite, les protocoles choisis par la communauté Internet étant TCP/IP, NetBIOS fut porté sur la pile TCP/IP et est appelé NBT (NetBIOS over TCP/IP).

1.4.2. Attribution d'un nom.

Dans le monde NetBIOS, chaque ordinateur qui se connecte au réseau annonce son nom. Ce processus s'appelle l'enregistrement de noms. Deux machines ne peuvent utiliser le même nom, afin d'éviter tout problème de communication avec les autres machines. Pour cela, deux approches sont possibles :

- utiliser un serveur de noms NetBIOS ;
- permettre à chaque machine du réseau de défendre son nom ;

Par ailleurs, le réseau doit avoir le moyen de résoudre un nom NetBIOS en une adresse IP. Cette fois encore, deux approches sont possibles :

- chaque machine transmet son adresse IP lorsqu'elle entend une demande de diffusion de son nom NetBIOS ;
- le serveur de noms NetBIOS transforme le nom en adresse IP.

1.4.3. Anatomie d'un nom.

Les noms NetBIOS existent dans un espace de noms plat, ce qui signifie qu'ils ne sont suivis d'aucun élément. Un nom unique identifie chaque ordinateur.

Il n'est pas surprenant que tous les noms DNS valides soient également des noms NetBIOS valides. En fait, le nom DNS d'un serveur Samba est souvent repris comme nom NetBIOS. Si, par exemple, une machine s'appelle P100.challe.be, son nom NetBIOS a de fortes chances d'être P100.

Avec NetBIOS, une machine avertit non seulement les autres de sa présence mais signale également les types de services qu'elle propose. Par exemple, P100 indique qu'elle n'est pas uniquement un poste de travail mais aussi un serveur de fichiers. Pour cela, un seizième octet, appelé type de ressource, est adjoint au nom de la machine.

Le type de ressource d'une longueur d'un octet, indique qu'un service unique est fourni par une machine. Dans le résultat des commandes Windows, il figure souvent entre chevrons (<>) après le nom NetBIOS.

Pour connaître les noms enregistrés par une machine NBT spécifique, il suffit de lancer la commande Windows NBTSTAT.

```
C:\>NBTSTAT -a P100

Connexion au réseau local:
Adresse IP du noeud : [192.168.1.80] ID d'étendue : []

    Table de noms NetBIOS des ordinateurs distants

      Nom                Type                État
      -----
P100                <00>  UNIQUE                Inscrit
P100                <03>  UNIQUE                Inscrit
P100                <20>  UNIQUE                Inscrit
PRIVE                <00>  GROUP                 Inscrit
PRIVE                <1B>  UNIQUE                Inscrit
PRIVE                <1E>  GROUP                 Inscrit

Adresse MAC = 00-00-00-00-00-00
```

Le protocole SMB exploite également le concept de groupe permettant aux machines de s'enregistrer sur le réseau. L'exemple de NBTSTAT montre que le serveur Samba P100 est également membre du groupe PRIVE

1.5. Implémentations par Microsoft.

1.5.1. Domaines Windows.

Un groupe de travail est un ensemble d'ordinateurs SMB résidant dans le même sous-réseau et qui sont rattachés au même groupe SMB. Un domaine Windows en est une extension. Il s'agit d'un groupe de travail de machines SMB ayant pour particularité d'être géré par un serveur jouant le rôle de contrôleur de domaine. Un domaine Windows dépourvu de contrôleur de domaine n'est qu'un groupe de travail.

Un contrôleur de domaine (serveur de connexions) utilise deux protocoles distincts : l'un pour les communications avec les ordinateurs Windows 95/98/Me, l'autre pour les communications avec les ordinateurs Windows NT/2000. Samba implémente le premier protocole mais ne prend pas encore en charge le protocole des machines Windows NT sauf dans la version 2.1.

Le contrôleur de domaine est le centre nerveux d'un domaine Windows. Les contrôleurs de domaine sont chargés de l'authentification des utilisateurs. L'authentification est un processus consistant à accorder ou à refuser l'accès à une ressource partagée sur une autre machine à un utilisateur.

Chaque contrôleur de domaine confie l'administration des authentifications à la SAM (Security Account Manager) qui gère une base de données centralisée de mots de passe.

Dans un domaine Windows, lorsqu'un client non authentifié demande l'accès aux partages d'un serveur, ce dernier interroge le contrôleur de domaine pour savoir si l'utilisateur est authentifié ou non. Dans l'affirmative, le serveur établit une connexion de session porteuse des droits d'accès enregistrés pour le service et l'utilisateur. Dans la négative, la connexion est refusée. Lorsque le contrôleur de domaine authentifie un utilisateur, un jeton spécial est renvoyé au client de sorte que l'utilisateur n'ait pas à s'identifier à nouveau pour accéder aux autres ressources du domaine. A cette étape, l'utilisateur est connecté au domaine.

La redondance est un concept important des domaines Windows. Le contrôleur actif d'un domaine s'appelle le PDC (Primary Domain Controller). Un domaine peut comporter un ou plusieurs BDC (Backup Domain Controller). Un BDC prend le relais d'un PDC en cas de dysfonctionnement ou de défaillance de celui-ci. Les BDC synchronisent régulièrement leurs données SAM avec le PDC.

Certains aspects sont communs à un groupe de travail Windows et à un domaine Windows puisque le concept de domaine Windows n'a pas évolué depuis Windows NT 3.51 et que les domaines Windows doivent être compatibles avec les groupes de travail Windows for Workgroups 3.1. En fait, un domaine Windows est un groupe de travail Windows comprenant un ou plusieurs contrôleurs de domaine.

A partir de la version 2.1, Samba peut être configuré comme PDC. Par contre, il est incapable de remplir la fonction de BDC étant donné le caractère fermé du protocole Microsoft employé pour la synchronisation des données SAM. L'évolution du produit Samba provient de la volonté d'une équipe d'effectuer du reverse engineering de manière à découvrir les secrets de fabrication de Microsoft.

1.5.2. Exploration.

L'exploration est la réponse à la question formulée par les utilisateurs : « Quelles sont les machines connectées au réseau ? ». Les utilisateurs doivent connaître le nom de l'ordinateur auquel ils souhaitent se connecter, puis doivent entrer le chemin d'accès à la ressource avant d'explorer,.

Il existe donc deux types d'exploration :

- l'exploration d'une liste de machines ;
- l'exploration des ressources partagées d'une machine spécifique.

Pour que l'exploration de premier type soit possible, un ordinateur doit tenir à jour une liste des machines accessibles sur le réseau. L'ordinateur détenant cette liste est appelé l'explorateur principal local et la liste est appelée la liste d'exploration. Lorsqu'un ordinateur doit explorer le réseau, il demande la liste d'exploration à l'explorateur principal local.

Pour explorer les ressources d'une machine, l'utilisateur doit s'y connecter. Si l'authentification réussit, la machine sollicitée renvoie la liste des ressources auxquelles l'utilisateur a le droit d'accéder.

Dans un groupe de travail Windows, chaque serveur doit annoncer non seulement sa présence à l'explorateur principal local après l'enregistrement de son nom NetBIOS, mais aussi son départ lorsqu'il est mis hors tension. Toute machine Windows (2000, NT, 95, 98, Me) peut jouer le rôle d'explorateur principal local. Comme le contrôleur de domaine, l'explorateur principal local peut, en cas de panne, être assisté d'un ou de plusieurs explorateurs secondaires. Pour assurer une fluidité de fonctionnement, les explorateurs secondaires synchronisent régulièrement leur liste d'exploration avec l'explorateur principal.

Lorsque l'explorateur principal s'arrête, il convient d'élire un nouvel explorateur principal. L'algorithme d'élection permet à plusieurs ordinateurs de choisir lequel d'entre eux devient explorateur principal, les autres étant alors explorateurs secondaires. Une élection peut être provoquée à n'importe quel moment. Lors de la mise en route d'un ordinateur, il signale sa présence et provoque une élection au cours de laquelle l'explorateur principal peut être remis en cause.

1.5.3. Le service WINS.

Le service WINS est l'implémentation par Microsoft d'un serveur de noms NetBIOS. De plus, WINS est dynamique : lors de sa première connexion, un client doit fournir ses noms d'hôte, son adresse et son groupe au serveur WINS local.

Ce dernier mémorise ces informations tant que le client met périodiquement à jour son enregistrement WINS, signifiant ainsi qu'il est toujours connecté au réseau.

Bien qu'il soit possible de configurer plusieurs serveurs WINS qui doivent coopérer entre eux, ce mécanisme devient très rapidement lourd et peu efficace. L'idéal est de codifier au niveau de chaque client, l'adresse d'un serveur WINS qui sera le même pour tous.

2. Configuration de Samba.

2.1. Fichier de configuration de base pour Samba.

La configuration de Samba exige la modification du fichier `/etc/samba/smb.conf`. Ce fichier est l'unique source de renseignements nécessaires à la configuration de Samba.

Le logiciel Samba peut être configuré de manière telle que le serveur fasse partie d'un groupe de travail ou d'un domaine. Voici un exemple de configuration de Samba pour un groupe de travail :

```
[root@p200 samba]# cat smb.conf
[global]
    workgroup = PRIVE
[test]
    comment = Test et mise au point de Samba
    path = /home/tst
    read only = no
    guest ok = yes
```

Ce fichier de configuration appelle, test le partage du serveur Samba. Ce dernier devient également membre du groupe PRIVE, auquel doit appartenir chaque client. Pour que les clients puissent utiliser le partage test, il faut qu'il corresponde à un répertoire du système UNIX. Dans cet exemple, le répertoire UNIX correspondant est `/home/tst`. L'administrateur doit donc créer ce répertoire veillant, par exemple, à permettre à tous le monde d'y accéder pour y lire et y écrire des données. Voici les commandes à exécuter pour le répertoire créé et pour lui donner les permissions d'accès suffisantes :

```
[root@p200 /root]# mkdir /home/tst
[root@p200 /root]# chmod 777 /home/tst
```

La clause `read only = no` indique à Samba de laisser un accès en lecture et en écriture aux données du partage. Si la valeur de cette clause avait été fixée à `yes`, personne n'aurait pu écrire d'informations dans le partage.

Sous Windows 98 ainsi qu'à partir de Windows NT 4.0 Service Pack 3, les mots de passe qui transitent sur le réseau sont cryptés. Par défaut, Samba s'attend à recevoir les mots de passe en clair. Dans ce type d'environnement, il faut que Samba soit configuré pour recevoir des informations chiffrées. Voici le fichier `/etc/samba/smb.conf` modifié pour accepter les mots de passe cryptés :

```
[root@p200 samba]# cat smb.conf
[global]
    workgroup = PRIVE
    encrypt passwords = yes
[test]
    comment = Test et mise au point de Samba
    path = /home/tst
    read only = no
    guest ok = yes
```

Dans ce cas de figure, le mot de passe ne transite plus en clair sur le réseau. En réalité, le client chiffre l'information et la transmet au serveur qui la comparera avec sa version cryptée. Si les deux informations sont identiques, le client aura accès à la ressource. Cela signifie que le serveur Samba doit disposer d'une version cryptée des mots de passe. La commande `smbpasswd` permet d'ajouter un utilisateur à la base de données des noms/mots de passe. Cette base de données sera consultée par le serveur pour déterminer si un client peut accéder aux informations. Voici un exemple de commande qui ajoute un utilisateur UNIX existant à la base de données :

```
[root@p200 /root]# smbpasswd -a jfc
New SMB passwd:
Retype new SMB password:
Startsmbfilepwent_internal: unable to open file /etc/samba/smbpasswd. Error was
Aucun fichier ou répertoire de ce type
Added user jfc.
```

Le message d'erreur généré par le système n'est pas important : la première fois qu'un utilisateur est ajouté à la base de données, cela provoque la création du fichier des mots de passe qui n'existait pas.

Une autre technique permet à des clients Windows NT et Windows 2000 de se connecter à un serveur Samba sans que ce dernier soit dans l'obligation de chiffrer les mots de passe. La base de registre de Windows contient une entrée imposant le chiffrement des mots de passe. Il suffit de modifier cette entrée pour indiquer que les mots de passe doivent transiter en clair sur le réseau. L'entrée de la base de registre à modifier dans le cas de Windows NT est :

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword :00000001
```

Dans le cas de Windows 2000, l'entrée de la base de registre à modifier est :

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation
\Parameters]"EnablePlainTextPassword"=dword :00000001
```

2.2. Démarrage du serveur Samba.

Dans le cas de la distribution Linux RedHat, le démon Samba est lancé par le script /etc/rc.d/init.d/smb. L'administrateur doit donc simplement configurer la machine Linux pour qu'elle lance le serveur Samba au démarrage. Il peut aussi employer la commande suivante :

```
[root@p200 /root]# /etc/rc.d/init.d/smb start
```

3. Partages de disque.

3.1. Structure du fichier de configuration.

Le fichier /etc/samba/smb.conf est constitué de rubriques qui, à l'exception de la rubrique [global], désignent chacune un partage de disque ou d'imprimante accessibles aux clients ouvrant une session avec le serveur Samba. Les autres lignes sont des options propres au partage considéré. Le contenu d'une rubrique est compris entre le titre entre crochets et le titre suivant ou, s'il s'agit de la dernière rubrique, la fin du fichier de configuration. Chaque option de configuration se présente sous la forme d'un nom, suivi du signe d'égalité, lui-même suivi de la valeur de l'option.

Le fichier de configuration peut être modifié lors de l'exécution du démon Samba. Par défaut, Samba vérifie le fichier de configuration toutes les 60 secondes. S'il y trouve des modifications, celles-ci prennent immédiatement effet. Certaines modifications ne sont pas directement prises en compte par les clients. Par exemple, des modifications apportées à un partage en cours d'utilisation ne le seront que lorsque le client s'en déconnecte et s'y reconnecte. Les paramètres propres au serveur comme le nom du groupe de travail n'entrent pas en vigueur immédiatement. Cela évite que les clients actifs soient déconnectés brusquement ou rencontrent des problèmes d'accès lorsqu'une session est ouverte.

3.2. Les sections du fichier de configuration.

3.2.1. La section global.

Bien que facultative, la section global figure virtuellement dans chaque fichier de configuration de Samba. Les options définies dans cette section s'appliquent à tous les autres partages comme si le contenu de la section y était copié. Cependant, une option définie dans la section «global» peut généralement être reprise et modifiée dans une autre section. En d'autres termes, ce comportement est identique au masquage d'une variable globale par une variable locale de même nom.

Prenons l'exemple de l'option `read only`. Si elle est définie à la fois dans la section `global` et dans la section `test`, c'est la déclaration de la section `test` qui prévaut.

```
[root@p200 samba]# cat smb.conf
[global]
    workgroup = PRIVE
    encrypt passwords = yes
    read only = yes
[test]
    comment = Test et mise au point de Samba
    path = /home/tst
    read only = no
```

3.2.2. La section `homes`.

Si un client tente de se connecter à un partage ne figurant pas dans le fichier `smb.conf`, Samba recherche un partage `homes` dans le fichier de configuration. S'il en trouve un, le nom de partage non identifié est assimilé à un nom d'utilisateur UNIX qui est alors recherché dans la base des mots de passe du serveur Samba. Si le nom d'utilisateur existe, le serveur considère que le client est un utilisateur UNIX tentant de se connecter à son répertoire personnel. Cela signifie que Samba crée dynamiquement des partages qui ne figurent pas dans le fichier de configuration.

3.2.3. La section `printers`.

Si un client souhaite accéder à un partage qui n'est pas défini dans le fichier de configuration et que ce nom ne correspond pas à un utilisateur UNIX, Samba vérifie s'il s'agit d'une imprimante. Pour ce faire, il recherche le nom du partage dans le fichier `/etc/printcap`.

Dans le cas où un partage `homes` est déclaré dans le fichier de configuration, il est inutile d'associer un partage à chaque imprimante du système. Au besoin, Samba consulte la déclaration des imprimantes UNIX pour les mettre à la disposition des clients.

3.3. Configuration du serveur.

Voici un exemple de fichier de configuration à analyser :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
```

Ce fichier de configuration est simple : il enregistre sur le réseau NBT, le serveur Samba sous le nom NetBIOS P200. La machine appartient au groupe de travail PRIVE et propose aux clients une description du serveur sous la forme de la phrase : « serveur Samba de l'hote P200 ».

L'option `netbios name` a pour valeur par défaut le nom de l'hôte du serveur Samba, c'est-à-dire la première partie du nom DNS complet de la machine. Par exemple, une machine dont le nom DNS est P200.challe.be aura P200 pour nom NetBIOS par défaut. Cette option permet de redéfinir le nom NetBIOS de la machine dans le fichier de configuration.

Le paramètre `server string` définit le commentaire figurant à côté du nom du serveur dans le voisinage réseau et dans le gestionnaire d'impression Microsoft Windows.

Le paramètre `workgroup` définit le groupe de travail dans lequel le serveur Samba se présente. Les clients souhaitant accéder aux partages du serveur Samba doivent appartenir au même groupe de travail NetBIOS.

Le partage test est associé au répertoire /home/tst et possède un libellé descriptif ainsi qu'un nom de volume. Le partage est accessible en écriture, ce qui permet aux utilisateurs d'y inscrire des données. Par défaut, un partage est défini en lecture seule. Pour modifier cette propriété, il faut spécifier `writeable = yes`. Les options `read only` et `writeable` produisent le même effet. Définir l'option `read only = yes` revient à définir `writeable = no`. Ces options sont donc des alternatives possibles pouvant être considérées comme des synonymes inversés.

L'option `path` indique le chemin d'accès à la racine du partage de fichiers ou celui de l'impression. Il est permis de choisir n'importe quel répertoire du serveur Samba à la condition que le propriétaire du processus dispose d'un accès en lecture et en écriture à ce répertoire. S'il concerne un partage d'impression, le chemin d'accès doit pointer vers un répertoire temporaire dans lequel les fichiers transitent avant d'être dirigés vers le spool de l'imprimante

3.4. Les options réseau.

Lorsque Samba est exécuté sur une machine connectée à plusieurs réseaux, des options de sécurité en fonction du réseau client peuvent être mises en place. Voici un exemple de fichier `smb.conf` :

```
[root@p100 etc]# cat smb.conf
[global]
  netbios name = P100
  server string = serveur Samba de l'hote P100
  workgroup = PRIVE
  encrypt passwords = yes
  hosts allow = 192.168.1.
  hosts deny = 192.168.1.50
  interfaces = 192.168.1.1/255.255.255.0
  bind interfaces only = yes
[test]
  comment = Disque de donnees
  volume = exemple-de-partage
  path = /home/tst
  writeable = yes
```

Les options `hosts allow` et `hosts deny` fournissent un moyen de sécurité pour autoriser ou pour refuser les connexions des hôtes en fonction de leurs adresses IP. Dans cet exemple, la valeur `192.168.1.` est attribuée à l'option `hosts allow`. Le point terminal signifie que l'adresse n'est pas complète mais qu'elle englobe tous les hôtes du sous-réseau `192.168.1.0`. Cependant, l'accès à l'hôte `192.168.1.50` est interdit par l'option `hosts deny`. Voici les règles d'application des options `hosts allow` et `hosts deny` :

- si aucune option `allow` ou `deny` ne figure dans le fichier de configuration, Samba autorise les connexions à partir de toutes les machines autorisées par le système lui-même ;
- les options `hosts allow` et `hosts deny` figurant dans la section `global` s'appliquent à tous les partages même si des options de substitution sont définies pour certains d'entre eux ;
- si seule l'option `hosts allow` est associée à un partage, seuls les hôtes spécifiés ont accès à celui-ci. Tous les autres sont exclus ;
- si seule une option `hosts deny` est associée à un partage, celui-ci est accessible par toutes les machines ne figurant pas dans la liste ;
- si une option `hosts allow` et une option `hosts deny` sont définies, l'hôte doit figurer dans la liste des hôtes autorisés et pas dans la liste des hôtes exclus pour pouvoir accéder au partage. Autrement, il n'a pas accès.

En standard, Samba n'envoie des données que par l'intermédiaire de l'interface réseau principale. Pour transmettre des données via plusieurs interfaces, il faut indiquer la liste complète de interfaces. Si Samba est connecté aux sous-réseaux `192.168.1.0` et `212.68.198.208`, il faut indiquer, en paramètre de l'option `interfaces`, la liste de tous les réseaux qui doivent être desservis par Samba. Le réseau `212.68.198.208` étant une connexion vers Internet, il est préférable de limiter l'accès à Samba au réseau interne. Cette configuration est nécessaire car rien ne dit que Samba considère l'interface desservant le réseau interne comme interface principale.

L'option `bind interfaces only` demande à Samba de n'accepter des messages broadcast qu'en provenance des réseaux définis dans la clause `interfaces`. L'effet de cette option est différent de celui des options `hosts allow` et `hosts deny` qui empêchent des machines de se connecter aux services mais pas de recevoir des messages de diffusion. L'option `bind`

interfaces only est un bon moyen d'empêcher le serveur Samba de recevoir des datagrammes émanant d'autres sous-réseaux et de réduire le nombre de réseaux desservis par Samba.

4. Exploration et partages de disques avancés.

4.1. L'exploration.

L'exploration désigne la fonction et l'opération permettant d'examiner les serveurs et les partages disponibles sur le réseau. Sur des clients Windows, le résultat de la recherche de ressources partagées s'affiche dans le dossier voisinage réseau. Pour afficher les ressources d'imprimante ou de partage disque, l'utilisateur n'a plus qu'à cliquer sur l'icône représentant le serveur.

Pour afficher la liste des serveurs disponibles sur le réseau, l'utilisateur peut aussi exécuter la commande `net view` à l'invite Windows. Voici un exemple d'exécution de cette commande :

```
C:\>net view
Nom de serveur          Remarque
-----
\\P100                  P100
\\P200                  serveur Samba de l'hote P200
\\P2366
\\P3450
La commande s'est terminée correctement.
```

4.1.1. Blocage de l'exploration.

L'option booléenne `browseable` permet de ne pas afficher un partage dans le voisinage réseau. Bien qu'un partage standard de disque soit le plus souvent affiché, l'option `browseable` est utile au cas où le partage est homes. Ce partage, regroupant les répertoires personnels des utilisateurs, est souvent protégé en exploration de sorte que ses ressources ne soient pas consultables, ce qui n'empêche pas chaque utilisateur d'accéder à son répertoire personnel à chaque fois qu'il se connecte au partage. Voici un exemple de limitation de l'exploration :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

4.1.2. Election d'explorateurs.

Une machine d'un réseau conserve la liste de toutes les machines actives. Cette liste est appelée liste d'exploration et le serveur qui la gère est appelé explorateur local principal. Ce dernier met continuellement la liste à jour et la diffuse à la demande, sur le réseau.

Un ordinateur devient explorateur local principal à l'issue d'une élection. Cette élection peut avoir lieu à tout moment. Le serveur Samba peut provoquer une élection pour de multiples raisons et notamment pour faire en sorte d'être toujours l'explorateur principal ou, au contraire, de ne jamais le devenir.

Toute machine candidate à une élection doit diffuser des informations concernant :

- la version du protocole d'élection ;
- le système d'exploitation de la machine ;
- la durée de la session du client ;
- le nom d'hôte du client.

Pour le moment, toutes les machines utilisent la même version du protocole d'élection. Ce paramètre n'entre donc pas en ligne de compte pour déterminer l'élu. A chaque version du système d'exploitation correspond une valeur entière comme le montre le tableau suivant :

Système	Valeur
Windows NT Server 4.0	33
Windows NT Server 3.51	32
Windows NT Workstation 4.0	17
Windows NT Workstation 3.51	16
Windows 98	2
Windows 95	1
Windows 3.1 for Workgroups	1

En fonction des informations diffusées, chaque ordinateur du réseau reçoit une valeur qui détermine son rôle :

Rôle	Valeur
Contrôleur principal de domaine	128
Client WINS	32
Explorateur principal favori	8
Explorateur principal actif	4
Explorateur en attente	2
Explorateur secondaire actif	1

En conclusion, si l'administrateur du serveur UNIX souhaite que Samba joue le rôle d'explorateur local principal, il suffit de modifier le paramètre `os level` et de lui donner la valeur 34.

Un serveur Windows NT jouant le rôle de contrôleur principal de domaine possède une valeur cachée lui permettant, dans certaines conditions, de jouer le rôle d'explorateur local principal. Cette valeur est appelée bit de l'explorateur principal favori. Sur un serveur Samba, ce bit peut être positionné à l'aide de l'option `preferred master`. Voici un exemple de fichier de configuration forçant Samba à être un explorateur local principal :

```
[root@p200 samba]# cat smb.conf
[global]
  netbios name = P200
  server string = serveur Samba de l'hote P200
  workgroup = PRIVE
  encrypt passwords = yes
  os level = 33
  preferred master = yes
[test]
  comment = Disque de donnees
  volume = exemple-de-partage
  path = /home/tst
  writeable = yes
  browseable = yes
[homes]
  comment = Home directory
  read only = no
  browseable = no
```

4.2. Différences entre systèmes de fichiers.

La compatibilité entre les systèmes de fichiers UNIX et non UNIX constitue l'une des principales difficultés auxquelles Samba est confronté. Le serveur Samba doit gérer :

- les liens symboliques ;
- les fichiers cachés ;
- les permissions sur les fichiers.

4.2.1 Droits d'accès et fichiers cachés.

Dans certains cas, un fichier doit absolument être caché ou inaccessible à l'utilisateur. Dans d'autres cas, le fichier ne devra pas être visible lors de la consultation d'un répertoire mais son accès devra être possible. Sous Windows, les attributs associés à un fichier définissent les propriétés d'accès et d'affichage. Sous UNIX, le masquage d'un fichier dans la liste d'un répertoire se fait par l'ajout d'un point devant son nom. Les fichiers de configuration ou de paramètres ne sont pas visibles lors de l'exécution d'un `ls` ordinaire. L'interdiction d'accès nécessite, quant à elle, la gestion des permissions sur les fichiers et/ou les répertoires.

L'option `hide dot file` permet, selon sa valeur, de cacher des fichiers dont le nom commence par un point. En réalité, ces fichiers seront considérés par Windows comme affectés de l'attribut caché. Cela n'empêche pas un utilisateur Windows de configurer son explorateur de fichiers pour qu'il affiche tous les fichiers y compris ceux affectés de l'attribut caché. Voici un exemple de configuration permettant de masquer les fichiers débutant par le caractère point :

```
[root@p200 samba]# cat smb.conf
[global]
  netbios name = P200
  server string = serveur Samba de l'hote P200
  workgroup = PRIVE
  encrypt passwords = yes
  os level = 33
  preferred master = yes
[test]
  comment = Disque de donnees
  volume = exemple-de-partage
  path = /home/tst
  writeable = yes
  browseable = yes
[homes]
  comment = Home directory
  read only = no
  browseable = no
  hide dot files = yes
```

L'option `hide files` permet d'affiner le comportement en définissant un masque de noms de fichiers à cacher. Par exemple, le fichier `mbox` contenant les courriers électroniques reçus par l'utilisateur peut être caché. En guise d'exemple il serait également possible de masquer tous les fichiers dont le nom débute par la lettre `D` (majuscule ou minuscule). Voici le contenu du fichier de configuration permettant d'effectuer cette opération :

```
[root@p200 samba]# cat smb.conf
[global]
  netbios name = P200
  server string = serveur Samba de l'hote P200
  workgroup = PRIVE
  encrypt passwords = yes
  os level = 33
  preferred master = yes
[test]
  comment = Disque de donnees
  volume = exemple-de-partage
  path = /home/tst
```

```

writeable = yes
browseable = yes
[homes]
comment = Home directory
read only = no
browseable = no
hide dot files = yes
hide files = /mbox/D*/

```

Pour empêcher l'affichage des fichiers, quelle que soit l'option de visualisation choisie dans l'explorateur de fichiers, il faut utiliser l'option veto files. De même syntaxe que l'option hide files, elle permet de définir la liste des fichiers qui seront toujours masqués à l'utilisateur. Par exemple, les fichiers débutant par le caractère point devraient toujours être masqués quelle que soit l'option d'affichage des répertoires. Voici une configuration permettant ce comportement :

```

[root@p200 samba]# cat smb.conf
[global]
netbios name = P200
server string = serveur Samba de l'hote P200
workgroup = PRIVE
encrypt passwords = yes
os level = 33
preferred master = yes
[test]
comment = Disque de donnees
volume = exemple-de-partage
path = /home/tst
writeable = yes
browseable = yes
[homes]
comment = Home directory
read only = no
browseable = no
veto files = /.*/

```

Les fichiers masqués par l'option veto files deviennent purement et simplement inaccessibles par les clients Windows.

4.3. Permissions et attributs des fichiers sous MS-DOS et sous UNIX.

Sous Windows, les attributs d'un fichier sont :

- lecture seule ;
- système ;
- caché ;
- archive ;

Théoriquement, Samba est capable de traiter les attributs Windows en affectant une signification particulières aux droits sur les fichiers UNIX. Expérimentalement, la version 2.2.2 de Samba n'est capable de gérer que les attributs lecture seule et archive. L'attribut lecture seule consiste à retirer le droit d'écriture au propriétaire du fichier alors que l'attribut archive consiste à ajouter le droit d'exécution au propriétaire du fichier

5. Utilisateur, sécurité et domaines.

La configuration des utilisateurs avec le serveur Samba soulève un certain nombre de problèmes. Les administrateurs éprouvent souvent des difficultés d'authentification. Avant tout, il convient de préciser que si les clients de Samba exécutent Windows 98 ou Windows NT, il faut configurer la prise en charge des mots de passe cryptés sur le serveur Samba. Sans cela, il sera impossible d'établir des connexions entre les clients et le serveur.

5.1. Utilisateurs et groupes.

Le moyen le plus simple de configurer un utilisateur client consiste à lui créer un compte UNIX ainsi qu'un répertoire personnel sur le serveur et à informer Samba de l'existence de cet utilisateur. Cette démarche est celle appliquée jusqu'à présent dans ce chapitre.

L'utilisation de ressources informatiques, par un groupe de personnes, conduit au partage des données. Plusieurs utilisateurs souhaitent se partager des données. L'utilisation d'un partage commun à plusieurs personnes est une réponse à ce problème. L'option `valid users` peut recevoir une liste de noms d'utilisateurs autorisés à accéder aux informations. Voici l'application de ce concept au partage test :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = jfc rc
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

L'exploitation de cette méthode pour partager des informations entre un grand nombre d'utilisateurs a ses limites. Il devient rapidement malaisé d'administrer la liste des utilisateurs autorisés. Le système UNIX offre la notion de groupe d'utilisateurs. Lorsque l'administrateur désire autoriser l'accès à une ressource par plusieurs utilisateurs, il lui suffit de créer un groupe dans lequel seront placés tous les utilisateurs devant se partager des données et de déclarer au niveau de la configuration de Samba qu'un partage peut être utilisé par les membres de ce groupe. Pour rappel, voici la séquence de commandes aptes à créer un groupe et à y placer des utilisateurs :

```
[root@p200 samba]# groupadd staff
[root@p200 samba]# usermod -G staff jfc
[root@p200 samba]# usermod -G staff rc
```

Le fichier de configuration de Samba doit contenir, au niveau de l'option `valid users`, le nom du groupe d'utilisateurs autorisés à accéder aux données. Voici un exemple de cette configuration :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = @staff
```

Certains utilisateurs ou groupes d'utilisateurs devraient ne pas être autorisés à se connecter via Samba. En conséquence, l'option globale `invalid users` permet d'interdire l'utilisation des services de Samba à certains utilisateurs. Voici un exemple permettant d'interdire à l'utilisateur `root` d'exploiter Samba :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
    invalid users = root
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = @staff
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

Des partages peuvent être déclarés accessibles uniquement sous l'identité d'un utilisateur par défaut. La configuration d'origine de Samba fixe l'utilisateur anonyme à `nobody` alors que l'option `guest account` permet de spécifier une autre identité.

Supposons que l'administrateur souhaite laisser un accès libre au site ftp de la machine UNIX. Tous les fichiers présents dans le répertoire `/home/ftp` doivent appartenir à l'utilisateur `ftp`. De plus, seul cet utilisateur a les autorisations suffisantes pour tout effectuer dans ce répertoire. Voici un exemple de configuration d'un partage sous l'identité d'un utilisateur par défaut :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
    invalid users = root
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = @staff
[homes]
    comment = Home directory
    read only = no
    browseable = no
[ftp]
    path=/home/ftp
    comment = site ftp de l'hote P200
    writeable = yes
    guest ok = yes
    guest account = ftp
    guest only = yes
```


L'option `guest account` fixe l'identité de l'utilisateur invité alors que les options `guest ok` et `guest only` permettent respectivement d'accepter un accès au partage par un utilisateur invité et uniquement par celui-ci. L'idée de partage total du répertoire ftp n'est pas nécessairement un exemple à mettre en œuvre mais elle peut être appliquée à d'autres partages moins sensibles.

5.2. Sécurité des authentifications.

Afin de permettre aux utilisateurs d'accéder aux partages, Samba les authentifie au moyen d'un nom d'utilisateur et d'un mot de passe. La façon dont Samba traite le mot de passe, et par conséquent la stratégie qu'il adopte pour l'authentification des utilisateurs, est déterminée par l'option de configuration `security`. Le serveur Samba prend en charge les quatre niveaux de sécurité suivants :

- sécurité de niveau partage. Chaque partage d'un groupe de travail dispose d'un ou de plusieurs mots de passe. Quiconque connaît l'un d'eux peut se connecter au partage ;
- sécurité de niveau utilisateur. Chaque partage d'un groupe de travail est configuré pour autoriser l'accès de certains utilisateurs. A chaque nouvelle connexion, le serveur Samba vérifie si les utilisateurs et leurs mots de passe sont répertoriés. Dans l'affirmative, il autorise l'accès au partage ;
- sécurité de niveau serveur. Dans cette stratégie de sécurité semblable à la précédente, le serveur Samba valide les utilisateurs et leurs mots de passe à l'aide d'un serveur SMB séparé avant d'accorder l'accès au partage ;
- sécurité de niveau domaine. Samba authentifie les utilisateurs auprès du PDC dont il est membre. Après authentification, l'utilisateur reçoit un jeton spécial qui lui donne accès à n'importe quel partage du domaine, à la condition de posséder les permissions appropriées. Grâce au jeton, le PDC n'a pas à valider le mot de passe chaque fois que l'utilisateur tente d'accéder à un autre partage du domaine.

5.2.1. Sécurité de niveau partage.

Dans ce modèle, un ou plusieurs mots de passe sont associés à chaque partage. Ne limitant pas à des utilisateurs spécifiques les accès aux partages, ce modèle diffère radicalement des autres modes de sécurité. Pour accéder à un partage, il suffit de posséder le mot de passe approprié. Certains partages en ont plusieurs, l'un accordant un accès uniquement en lecture, l'autre autorisant un accès en lecture et en écriture.

Ce modèle de sécurité correspond à l'option `security = share`. Il est uniquement employé sous Windows 95/98. Bien que Samba puisse être configuré de cette manière, cela constitue de nos jours, une méthode désuète peu sécurisée. C'est pour cette raison que l'étude de cette technique ne sera pas poussée plus loin.

5.2.2. Sécurité de niveau utilisateur.

La sécurité de niveau utilisateur est le mode conseillé par les développeurs de Samba. Dans ce mode, chaque partage est manipulé par des utilisateurs distincts, dont les mots de passe figurent dans la base de données des mots de passe du serveur Samba. Pour restreindre l'accès d'un partage à des utilisateurs spécifiques, il est permis d'employer l'option `valid users` qui a été étudiée précédemment. Les utilisateurs répertoriés au niveau de cette option peuvent se connecter au partage à la condition de fournir le mot de passe conforme à celui figurant dans la base de données. Après l'identification initiale, l'utilisateur n'a plus à fournir à nouveau son mot de passe pour accéder au partage. Ce type de configuration (`security = user`) correspond au mode de fonctionnement par défaut de Samba.

5.2.3. Sécurité de niveau serveur.

Elle est similaire à la sécurité au niveau utilisateur. Dans ce mode (`security = server`), Samba délègue l'authentification à un serveur de mots de passe SMB, qui est généralement un autre serveur Samba ou un serveur Windows NT jouant le rôle de PDC. Lorsqu'un client demande à se connecter à un partage, le serveur doit vérifier que l'utilisateur est autorisé à le faire. Pour cela, il tente de valider le mot de passe en soumettant les informations d'authentification au serveur de mots de passe SMB via un protocole spécifique. Si le mot de passe est accepté, une session est établie avec le client.

La configuration d'une sécurité serveur exige de préciser l'emplacement du ou des serveurs de mots de passe SMB dans le fichier de configuration de Samba. L'option `password server` accepte une liste de serveurs qui seront exploités successivement pour valider l'accès d'un utilisateur. Le nom du serveur de mots de passe doit correspondre au nom NetBIOS et pas au nom DNS. Si le premier serveur disponible refuse le mot de passe, la connexion échoue car Samba ne soumet pas la requête à un autre serveur. Voici un exemple d'une telle configuration :

```
[root@p200 samba]# cat smb.conf
[global]
    security = server
    password server = P166PLUS
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
    invalid users = root
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = @staff
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

5.2.4. Sécurité de niveau domaine.

Elle est identique à la sécurité de niveau serveur, hormis le fait que le serveur Samba soit membre d'un domaine Windows. Chaque domaine comporte un contrôleur, généralement un serveur Windows NT chargé de l'authentification des mots de passe. Un contrôleur de domaine fournit les services d'un serveur de mots de passe. Il répertorie les noms des utilisateurs et les mots de passe dans le module d'authentification de sécurité plus communément appelé module SAM (Security Authentication Module). Voici un exemple de sécurité de niveau domaine (security = domain):

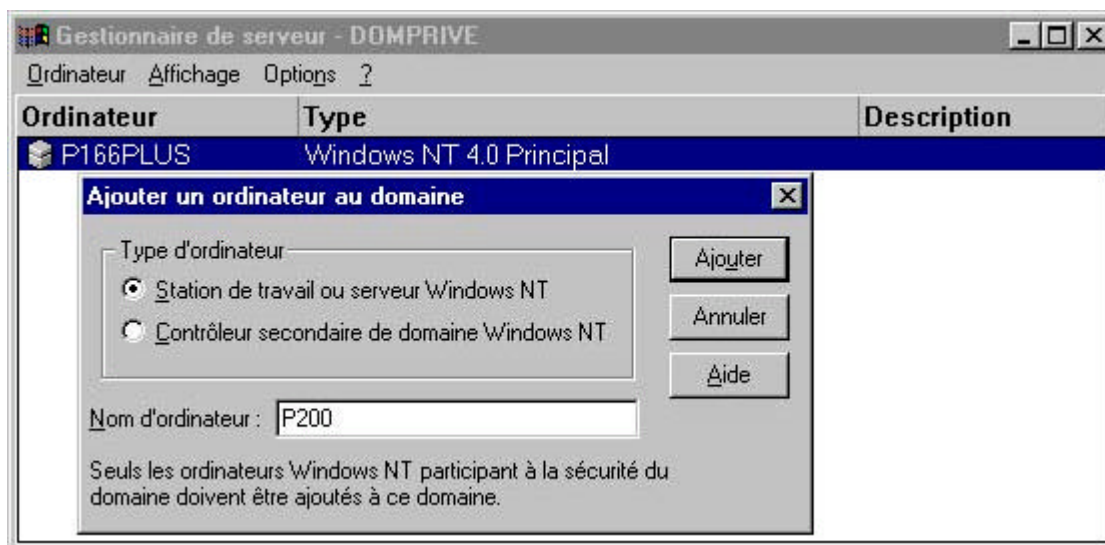
```
[root@p200 samba]# cat smb.conf
[global]
    security = domain
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = PRIVE
    encrypt passwords = yes
    os level = 34
    preferred master = yes
    invalid users = root
[test]
    comment = Disque de donnees
    volume = exemple-de-partage
    path = /home/tst
    writeable = yes
    browseable = yes
    valid users = @staff
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

La sécurité de niveau domaine permet d'employer le mécanisme Windows NT natif, ce qui offre de nombreux avantages, à savoir :

- simplicité de la configuration ;
- meilleure intégration des serveurs NT dans le réseau Samba ;

5.2.5. Ajout d'un serveur Samba à un domaine Windows NT.

Cette opération est aisément réalisable si le domaine NT existe déjà. Il faut en premier lieu arrêter le serveur Samba, puis, à partir du contrôleur principal il faut ajouter le serveur au domaine NT à l'aide de l'outil Gestionnaire de serveur comme le montre l'illustration suivante :



Ajout d'un ordinateur au domaine.

A l'aide de la commande `smbpasswd`, il faut ajouter la machine P200 au domaine NT dont le PDC est P166PLUS. Voici la syntaxe de la commande à employer :

```
[root@p200 samba]# smbpasswd -j DOMPRIVE -r P166PLUS
```

Afin que le serveur Samba soit serveur membre du domaine NT, il faut modifier le fichier de configuration de la manière suivante :

```
[root@p200 samba]# cat smb.conf
[global]
    netbios name = P200
    server string = serveur Samba de l'hote P200
    workgroup = DOMPRIVE
    security = domain
    password server = P166PLUS
    encrypt passwords = yes
    os level = 34
    preferred master = yes
    invalid users = root
[homes]
    comment = Home directory
    read only = no
    browseable = no
```

Le serveur Samba est maintenant configuré pour utiliser la sécurité de niveau domaine.

5.3. Domaine Windows.

5.3.1. Introduction.

A partir de la version 2.1 de Samba, il est possible de configurer une machine UNIX en un PDC. Pour se connecter au domaine, l'utilisateur doit présenter un nom et un mot de passe valides au démarrage. Ces informations sont comparées à celles figurant dans la base de données du contrôleur principal de domaine. Si le couple d'informations saisi n'est pas valide, l'utilisateur en est immédiatement averti et il ne peut se connecter au domaine. Après une connexion à un domaine, tout utilisateur peut accéder à n'importe quel partage pour lequel il dispose des droits suffisants. Cet accès

aux partages n'exige pas une nouvelle authentification de la part de l'utilisateur. En effet, le PDC revoie un jeton à l'ordinateur client au moment de l'acceptation de l'ouverture d'une session. Cela évite de multiplier les consultations de la base de données du PDC lors de chaque connexion à une ressource. Ce mécanisme transparent pour l'utilisateur permet de diminuer considérablement le trafic réseau.

5.3.2. Configuration PDC pour des clients Windows NT.

Pour être contrôleur principal de domaine, le démon Samba doit employer la sécurité de niveau utilisateur ainsi que les mots de passe chiffrés. De plus, l'authentification des utilisateurs doit être effectuée au moment de la demande de connexion au domaine. L'option booléenne `domain logons` permet d'activer cette fonctionnalité.

Les différents clients Windows doivent avoir un accès à une ressource appelée `netlogon`. Cette ressource privée doit être protégée contre l'écriture.

Voici un exemple de fichier de configuration de Samba pour qu'il devienne PDC :

```
[root@p200 samba]# cat smb.conf
[global]
  netbios name = P200
  server string = serveur Samba de l'hote P200
  workgroup = DOMPRIVE
  security = user
  domain logons = yes
  encrypt passwords = yes
  os level = 34
  preferred master = yes
  domain master = yes
  local master = yes
[netlogon]
  comment = Service de connexion au domaine
  path = /home/logon
  public = no
  writeable = no
  browseable = no
[homes]
  comment = Home directory
  read only = no
  browseable = no
```

Tous les clients Windows NT se connectant à un PDC font usage de comptes approuvés. Ces comptes permettent à un ordinateur de se connecter au PDC, ce qui permet à ce dernier d'approuver toutes les connexions. A tout point de vue, les comptes d'approbation traités par le serveur Samba sont des comptes d'utilisateurs UNIX standard. Le nom de connexion d'un compte d'approbation d'une machine est le nom de cette dernière auquel un signe dollar est ajouté. Si la machine Windows NT s'appelle P2450, le nom de connexion sera P2450\$. Le mot de passe initial du compte est le nom de la machine en minuscule. Pour créer un compte d'approbation sur le serveur Samba, il faut créer un compte UNIX avec le nom de la machine et un mot de passe chiffré dans la base de données `smbpasswd`.

Le fichier `/etc/passwd` doit être modifié pour y ajouter un nouvel utilisateur. Il n'est pas nécessaire d'employer la procédure standard de création des profils utilisateurs, l'utilisateur ne devant pas avoir de répertoire personnel ni de shell associé. Voici un exemple d'entrée à ajouter dans le fichier `/etc/passwd` :

```
p2450$:*:1000:900:Compte approbation:/dev/null:/dev/null
```

L'astérisque signifie que le champ mot de passe est désactivé. Le démon Samba stockera le mot de passe dans le fichier `smbpasswd` ; de plus aucun utilisateur ne pourra se connecter à la machine via telnet en utilisant ce profil. Hormis le nom de compte, la seule valeur significative est son UID (1000) utilisé également dans la base de mots de passe chiffrés.

Il faut ensuite ajouter le mot de passe chiffré à l'aide de la commande `smbpasswd` comme le montre l'exemple suivant :

```
[root@p200 /root]# smbpasswd -a -m p2450
Added user p2450$.
```

L'option `-m` indique que le compte d'approbation doit être généré. La commande `smbpasswd` génère automatiquement le mot de passe chiffré à partir du nom NetBIOS de la machine.

Les clients Windows NT doivent être configurés pour se connecter au domaine. C'est boîte de dialogue réseau qui permet de réaliser cette opération. Il suffit de cliquer sur le bouton modifier de manière à changer de domaine. Voici un exemple de cette manipulation :



Configuration du domaine de connexion d'un client Windows NT.

Il ne faut surtout pas cocher la case « Créer un compte d'ordinateur dans le domaine » car Samba ne gère pas encore cette fonctionnalité.

6. Impression.

Une imprimante raccordée à un serveur Samba apparaît dans la liste des partages affichés dans le voisinage réseau. Si le client possède le pilote de périphérique approprié, il peut aisément envoyer des travaux à une imprimante connectée à un serveur Samba.

Si le partage printers est défini dans le fichier `/etc/samba/smb.conf`, le démon Samba lit automatiquement le fichier de définition des imprimantes et crée les partages d'impression pour chacune des imprimantes accessibles sous UNIX. Voici un exemple de configuration :

```
[root@P100 /etc]# cat smb.conf
[global]
    workgroup = PRIVE
    server string = P100
    encrypt passwords = yes
[homes]
    comment = Home Directories
    read only = No
    browseable = No
[printers]
    comment = All Printers
    path = /var/spool/samba
    print ok = Yes
    browseable = No
```

7. Résolution de noms.

Avant l'apparition de serveurs de noms NetBIOS, la résolution de noms s'effectuait exclusivement par diffusion. Dans ce mode, pour connaître l'adresse d'une machine, il suffisait de diffuser son nom sur le réseau et la machine concernée renvoyait, en principe, l'information. Cette technique n'est pas appropriée que ce soit pour l'exploration ou pour l'enregistrement et la résolution des noms car elle ne traverse pas aisément un inter-réseau, de plus, elle a tendance à surcharger les réseaux. Pour y remédier, Microsoft propose le service WINS, véritable serveur de noms NetBIOS. L'administrateur peut installer un serveur WINS sur une machine et fournir son adresse à tous les clients du réseau. En conséquence, à la différence des diffusions les demandes d'enregistrement et de résolution de noms peuvent être adressées à une seule et même machine, depuis n'importe quel point du réseau.

Le serveur Samba peut être configuré pour être un client WINS ou pour être un serveur WINS.

La configuration d'un serveur Samba comme client WINS exige l'emploi de l'option `wins server`. Cette option doit avoir pour valeur l'adresse IP de la machine agissant comme serveur WINS.

```
[root@P100 /etc]# cat smb.conf
[global]
    workgroup = PRIVE
    server string = P100
    encrypt passwords = yes
    wins server = 192.168.1.80
[homes]
    comment = Home Directories
    read only = No
    browseable = No
[printers]
    comment = All Printers
    path = /var/spool/samba
    print ok = Yes
    browseable = No
```

Pour que Samba soit serveur WINS, il faut définir deux options dans le fichier de configuration. L'option `wins support` suffit à rendre Samba serveur WINS. Lorsque Samba est serveur WINS, les administrateurs utilisent souvent l'option

name resolver order. Cette option indique l'ordre des méthodes de résolution de noms NetBIOS. Quatre méthodes de résolution des noms sont possibles :

- lmhosts : le fichier LMHOSTS de type LAN Manager est utilisé. Ce fichier est semblable au fichier /etc/hosts si ce n'est que les noms sont de type NetBIOS ;
- hosts : cet argument indique une méthode de résolution standard sous UNIX. Cette résolution stipule qu'il faut utiliser le fichier /etc/hosts ou le DNS ou le protocole NIS ou encore une combinaison des trois ;
- wins : cet argument impose l'emploi du serveur WINS ;
- bcast : utilise une méthode de diffusion.

Voici un exemple de configuration de Samba en tant que serveur WINS :

```
[root@P100 /etc]# cat smb.conf
[global]
  wins support = yes
  name resolver order = wins lmhosts hosts bcast
  workgroup = PRIVE
  server string = P100
  encrypt passwords = yes
  wins server = 192.168.1.80
[homes]
  comment = Home Directories
  read only = No
  browseable = No
[printers]
  comment = All Printers
  path = /var/spool/samba
  print ok = Yes
  browseable = No
```

L'ordre d'apparition des valeurs au niveau de l'option name resolver order indique l'ordre successif des méthodes qui sont employées pour résoudre un nom.

XII. APACHE.

1. Introduction.

Lors de l'accès à un site Web, par exemple `http://www.challe.be/doc`, un ensemble de données sont envoyées via Internet, à la machine disposant de cette adresse IP. Une telle requête appelée URL (Universal Resource Locator) est constituée de trois parties :

- une méthode qui dans notre exemple est `http`. Cela signifie l'utilisation du protocole HTTP (HyperText Transfer Protocol) ;
- un nom d'hôte qui est dans ce cas `www.challe.be`;
- un nom de répertoire qui est ici `doc`.

En utilisant un navigateur, cette requête parvient au port 80 de l'hôte `www.challe.be`. Le serveur a deux possibilités de traduction de l'URL :

- soit par un nom de fichier et en ce cas il doit renvoyer son contenu au client ;
- soit sous un nom de programme et renvoyer ses sorties au client.

Dans tous les cas, le serveur doit exécuter un serveur Web qui est à l'écoute du réseau et qui réagit aux requêtes qui lui sont soumises en renvoyant des messages. Le programme Apache est un serveur Web pouvant fonctionner sur un grand nombre de systèmes d'exploitation multitâches. Cette caractéristique fait d'Apache le serveur Web le plus utilisé.

Au repos, le serveur Apache ne fait rien d'autre qu'écouter les ports TCP des adresses IP indiquées dans son fichier de configuration. Lorsqu'une requête `http` se présente sur un port valide, Apache la reçoit et en analyse les en-têtes. Il applique alors les règles qu'il trouve dans le fichier de configuration et agit en conséquence.

Le principal contrôle exercé sur Apache, par l'administrateur, passe par le fichier de configuration. L'administrateur dispose d'environ 150 directives influant sur le comportement d'Apache.

2. Premier lancement du serveur Web.

Le serveur Apache présume que le fichier de configuration est `/etc/httpd/conf/httpd.conf`. Généralement, le site Web se trouve dans le répertoire `/home/httpd`. Ce répertoire est composé de deux autres répertoires : `/home/httpd/html` et `/home/httpd/cgi-bin`. Le premier répertoire est destiné à contenir les fichiers `html` du site Web alors que le second accueille les programmes de type CGI (Common Gateway Interface).

Le fichier de configuration du serveur APACHE doit contenir l'option `DocumentRoot` afin de pouvoir indiquer l'emplacement des fichiers du site. Voici un exemple de configuration :

```
[root@p200 conf]# cat httpd.conf
User nobody
Group nobody
ServerName p200.challe.be
DocumentRoot /home/httpd/html
AccessConfig /dev/null
ResourceConfig /dev/null
```

Les options `User` et `Group` forcent le serveur à s'exécuter sous l'identité d'un utilisateur particulier. Pour des raisons de sécurité, cet utilisateur ne doit pas avoir de privilège, sans quoi la totalité du système de fichiers pourrait être modifié par les clients du site Web.

L'option `ServerName` permet d'informer APACHE du nom du site qu'il doit servir. Ce nom est utilisé lors de communications avec les clients en vue de les informer du nom du site.

Les options `AccessConfig` et `ResourceConfig` informent APACHE de l'emplacement des fichiers de configuration supplémentaires. Les fichiers de configuration `/etc/httpd/conf/access.conf` et `/etc/httpd/conf/srm.conf` sont traités après le fichier général de configuration. Actuellement, les concepteurs d'APACHE recommandent de placer la totalité de la

configuration dans le fichier `/etc/httpd/conf/httpd.conf`. Il faut donc informer le démon APACHE que les deux autres fichiers ne doivent pas être employés. Les options `AccessConfig` et `ResourceConfig` indiquent l'emplacement des fichiers supplémentaires de configuration. Dans l'exemple précédent, ces fichiers de configuration ne sont pas utilisés. Cela est indiqué par le nom de fichier `/dev/null`.

3. Un véritable site.

3.1. Interprétation du code HTML.

L'exemple précédent ne donne pas des résultats satisfaisants avec tous les browsers. La navigation vers l'URL `http://p200.challe.be/index.html` n'affiche une page correcte que par l'utilisation de Microsoft Internet Explorer. L'emploi de Netscape Navigator ou de lynx fait apparaître le code HTML de la page et non pas son interprétation. En effet, ces navigateurs exploitent l'extension du fichier pour déterminer la manière dont il doit être interprété. Aussi, un fichier de type `.html` doit-il être interprété comme contenant du code HTML. Cette interprétation est rendue possible par l'intermédiaire du fichier `/etc/mime.types`. Ce fichier contient, pour chaque extension, la réaction appropriée du navigateur. L'utilisation du fichier `/etc/mime.types` est précisée par l'option `TypesConfig` suivie du nom du fichier `mime`.

Toutes les versions d'APACHE ne sont pas compilées de manière à prendre en compte la gestion mime. Dans certains cas, l'utilisation de mime dépend du chargement d'un module spécifique contenant des extensions d'APACHE. Ces extensions peuvent être chargées au moyen des options `LoadModule` et `AddModule`. Le programme APACHE a été conçu pour être exécuté sur un grand nombre de systèmes dont les interfaces de programmation diffèrent. Les différentes fonctionnalités d'APACHE étant parfois dépendantes du système d'exploitation sous-jacent, il est plus commode de rédiger le programme en le découpant en plusieurs modules. Les modules spécifiques aux différents systèmes seront ainsi réécrits plusieurs fois. Cette approche modulaire présente également l'avantage de réduire au strict minimum la taille du code exécutable. En effet, toutes les implémentations d'APACHE ne nécessitent pas toujours les mêmes fonctionnalités. De ce fait, APACHE a été découpé en plusieurs modules chargeables à la demande.

Pour qu'APACHE puisse charger dynamiquement des modules, il faut lui adjoindre le code de gestion de ces modules. L'option `AddModule` permet de charger le module `mod_so.c` autorisant le chargement dynamique de modules. Le principe de chargement d'un module se passe en deux étapes, il y a d'une part son chargement et d'autre part son ajout à la liste des modules. L'option `LoadModule` permet le chargement alors que l'option `AddModule` permet l'ajout.

Dans le cas qui nous occupe actuellement, il faut charger le module `mime`. Voici le contenu du fichier de configuration d'APACHE :

```
[root@p200 conf]# cat httpd.conf
ServerType standalone
ServerRoot "/etc/httpd"
LoadModule mime_module          modules/mod_mime.so
ClearModuleList
AddModule mod_mime.c
AddModule mod_so.c
User nobody
Group nobody
ServerAdmin root@p200.challe.be
DocumentRoot "/home/httpd/html"
TypesConfig /etc/mime.types
```

L'option `ServerType` détermine comment le serveur Web est lancé. Il peut être activé de manière autonome ou par l'intermédiaire du démon `inetd`. Généralement, APACHE est configuré en `standalone` bien que la sécurité soit meilleure s'il est exécuté par `inetd`. Les administrateurs préfèrent ignorer la sécurité car un lancement par `inetd` implique que lors de chaque demande de connexion, une nouvelle instance du serveur Web est créée. Dès que la connexion est terminée, le serveur est arrêté. Ce comportement peut dégrader les performances du système.

Pour qu'APACHE soit en mesure de trouver les modules qu'il doit charger, il doit localiser le répertoire contenant les informations de configuration. Le répertoire `/etc/httpd` est la racine d'un sous-répertoire contenant les fichiers de configuration, les fichiers d'enregistrement des erreurs ainsi que les fichiers correspondant aux différents modules. L'option `ServerRoot` fixe l'emplacement de la racine du système de fichiers contenant les informations précitées.

Le serveur est fourni avec une liste de modules actifs. Cette liste peut être vidée au moyen de l'option `ClearModuleList`. Cette option présume que la liste sera reconstruite en utilisant l'option `AddModule`.

3.2. Chargement automatique du fichier `index.html`.

Lorsqu'un client souhaite accéder à un site Web, il ne précise que le nom de la machine sur laquelle réside le site. Cependant la configuration actuelle ne permet pas ce genre de requête. En effet, sans nom de fichier, le serveur n'a pas suffisamment d'informations pour déterminer le fichier par défaut qu'il faut envoyer au client. Une telle requête se soldera donc par un échec.

Dans la majorité des cas, le nom du fichier à charger par défaut est `index.html`. L'option `DirectoryIndex` permet de fixer le nom du fichier qui doit être envoyé au client lors d'une requête ne contenant que le nom de la machine. Pour que cette option soit reconnue, il faut charger le module `mod_dir`. Voici la nouvelle configuration du serveur APACHE :

```
[root@p200 conf]# cat httpd.conf
ServerType standalone
ServerRoot "/etc/httpd"
LoadModule mime_module          modules/mod_mime.so
LoadModule dir_module           modules/mod_dir.so
ClearModuleList
AddModule mod_mime.c
AddModule mod_dir.c
AddModule mod_so.c
Port 80
User nobody
Group nobody
ServerAdmin root@p200.challe.be
DocumentRoot "/home/httpd/html"
TypesConfig /etc/mime.types
DirectoryIndex index.html
```

3.3. Gestion de plusieurs répertoires.

Le site d'un serveur APACHE est, entre autres, constitué du répertoire `/home/httpd/html` et du répertoire `/home/httpd/icons`. Le dernier répertoire contient des icônes pouvant être employées dans les différentes pages HTML du site. Le code HTML fait référence à ces objets comme s'ils figuraient dans le répertoire `/home/httpd/html/icons`. Dans l'état actuel des choses, une telle référence se soldera par un échec car le répertoire `/home/httpd/html/icons` n'existe pas. Le fichier de configuration d'APACHE doit être modifié pour qu'une référence au répertoire `icons` soit considérée comme une tentative d'accès au répertoire `/home/httpd/icons`. L'option `Alias` détermine l'emplacement réel d'un répertoire correspondant au dossier demandé. Pour que cette option soit utilisable, il faut charger le module `mod_alias`. Voici un exemple de configuration du serveur APACHE prenant en charge les alias :

```
[root@p200 conf]# cat httpd.conf
ServerType standalone
ServerRoot "/etc/httpd"
LoadModule mime_module          modules/mod_mime.so
LoadModule dir_module           modules/mod_dir.so
LoadModule alias_module         modules/mod_alias.so
ClearModuleList
AddModule mod_mime.c
AddModule mod_dir.c
AddModule mod_alias.c
AddModule mod_so.c
Port 80
User nobody
Group nobody
ServerAdmin root@p200.challe.be
DocumentRoot "/home/httpd/html"
TypesConfig /etc/mime.types
DirectoryIndex index.html
Alias /icons/ "/home/httpd/icons/"
```

3.4. Utilisation des CGI.

Les sites Web peuvent être configurés de manière à ce qu'un programme soit exécuté par le serveur. Ces programmes sont appelés des scripts CGI (Common Gateway Interface). Ces scripts peuvent être écrits en langage C, en bash ou en perl. Voici un exemple de script écrit au moyen de l'interpréteur de commandes bash :

```
[root@p200 cgi-bin]# cat script
#!/bin/bash
echo "content-type:text/html"
echo
echo "<html>"
echo "  Bonjour Monsieur"
echo "</html>"
```

Voici le même exemple de CGI rédigé en langage C :

```
[root@P100 cgi-bin]# cat ess.c
#include <stdio.h>
void main () {
    printf("content-type:text/html\n\n");
    printf("<html>\n");
    printf("bonjour monsieur\n");
    printf("</html>\n");
}
```

Pour que les scripts soient exécutables, il faut non seulement inclure le module CGI dans la configuration d'APACHE mais également définir le répertoire /home/httpd/cgi-bin comme étant un répertoire accessible à partir des clients. Voici la nouvelle configuration du serveur :

```
[root@p200 conf]# cat httpd.conf
ServerType standalone
ServerRoot "/etc/httpd"
LoadModule mime_module          modules/mod_mime.so
LoadModule dir_module           modules/mod_dir.so
LoadModule cgi_module           modules/mod_cgi.so
LoadModule alias_module         modules/mod_alias.so
ClearModuleList
AddModule mod_dir.c
AddModule mod_cgi.c
AddModule mod_alias.c
AddModule mod_so.c
Port 80
User nobody
Group nobody
ServerAdmin root@p200.challe.be
DocumentRoot "/home/httpd/html"
TypesConfig /etc/mime.types
DirectoryIndex index.html
Alias /icons/ "/home/httpd/icons/"
ScriptAlias /cgi-bin/ "/home/httpd/cgi-bin/"
```

3.5. Remarque.

Les distributions d'APACHE sont fournies avec un fichier /etc/httpd/conf/httpd.conf préétabli contenant de nombreuses options et proposant beaucoup de fonctionnalités. Généralement, l'administrateur considère ce fichier comme une base à la configuration du serveur Web. A partir de maintenant, les modifications seront directement appliquées au fichier de base fourni avec APACHE.

4. Gestion de plusieurs sites.

Lorsque l'on dispose d'une connexion Internet permanente, il est permis d'héberger plusieurs sites Web sur le même ordinateur. Le serveur APACHE est capable d'effectuer cette gestion en se basant soit sur des adresses IP différentes, soit sur des URL différentes.

La méthode basée sur l'adresse IP exige la configuration d'une carte réseau ayant plusieurs adresses IP. Bien que cette technique de résolution donne de bons résultats, elle est à présent désuète et la méthode basée sur le nom lui est préférée. Le fonctionnement de la méthode basée sur le nom repose sur la capacité des navigateurs actuels à envoyer le nom du site à atteindre en même temps que la requête qu'ils formulent.

Cette méthode exige d'enregistrer plusieurs noms DNS pour une même adresse IP. En supposant que des clients accèdent aux sites p200.challe.be, www.challe.be et wwwwww.challe.be, il faut modifier la configuration du serveur DNS en conséquence, comme le montre l'illustration suivante :

```
[root@P100 named]# cat challe.be
@      IN      SOA      challe.be. root.localhost. (
                                1998120701 ; Serial
                                28800    ; Refresh
                                14400    ; Retry
                                3600000  ; Expire
                                86400   ) ; Minimum

      NS      P100.challe.be.
      MX      10      P100.challe.be.

P100      A      192.168.1.1
www       CNAME   P100
ftp       CNAME   P100
irc       CNAME   P100
mail     CNAME   P100
pop      CNAME   P100
news     CNAME   P100

P200      A      192.168.1.10
wwwwww    CNAME   P200
wwwwww    CNAME   P200
P2450    A      192.168.1.40
P133     A      192.168.1.20
macii    A      192.168.1.2
powerpc  A      192.168.1.3
fuji     A      192.168.1.4
P166     A      192.168.1.5
P166PLUS A      192.168.1.30
P2300    A      192.168.1.50
se30     A      192.168.1.60
HP4      A      192.168.1.192
PHASER560 A      192.168.1.193
P3450    A      192.168.1.70
P2366    A      192.168.1.80
modem    A      192.168.1.100
```

La configuration des hôtes virtuels exige la modification du fichier de configuration fourni avec APACHE en y ajoutant les lignes suivantes :

```
NameVirtualHost 192.168.1.10
<VirtualHost 192.168.1.10>
    ServerAdmin root@challe.be
    DocumentRoot /home/httpd/html/
    ServerName p200.challe.be
</VirtualHost>
```

```
<VirtualHost 192.168.1.10>
  ServerAdmin root@challe.be
  DocumentRoot /home/httpd/html/wwwwww
  ServerName wwwwww.challe.be
</VirtualHost>
<VirtualHost 192.168.1.10>
  ServerAdmin root@challe.be
  DocumentRoot /home/httpd/html/wwwwww
  ServerName www.challe.be
</VirtualHost>
```

L'option NameVirtualHost indique à APACHE que les requêtes vers cette adresse IP seront subdivisées par noms. Il peut sembler que les options ServerName jouent un rôle primordial. En fait elles ne font que de fournir un nom qu'APACHE devra envoyer aux clients.

XIII. SQUID.

1. Introduction.

Le logiciel SQUID est un serveur mandataire (proxy server) dont l'objectif est double :

- mandater l'accès aux serveurs Web. En fonction du mandat qui lui est attribué, SQUID va participer à la protection des réseaux internes connectés à Internet. La conséquence importante de ce mandat est que l'adresse IP source d'une requête n'est plus celle du client mais celle du serveur mandataire ;
- garder en mémoire cache les pages HTML et d'autres objets (images, animations...) téléchargés sur le Web pour ainsi améliorer la rapidité de la connexion Internet. Les objets sont périodiquement éliminés ou rafraîchis.

Le serveur SQUID est formé de deux démons : squid et unlinkd. Le démon unlinkd aide squid dans la gestion de la mémoire cache placée sur le disque.

2. Protocoles utilisés.

Les clients qui s'adressent à SQUID doivent employer le protocole HTTP sur TCP. Un client FTP ne pourra donc pas se servir de SQUID, au même titre que les clients bâtis sur UDP, ce qui exclut un grand nombre de programmes multimédias. Le serveur SQUID est toutefois capable de gérer les accès et d'aller chercher des informations sur les serveurs suivants :

- HTTP (HyperText Transfer Protocol)
- FTP (File Transfer Protocol)
- WAIS (Wide Area Information System)
- Gopher
- SSL (Secure Socket Layer)

Pour la gestion de sa mémoire cache et des communications entre les serveurs mandataires, SQUID utilise les protocoles suivants :

- HTTP servant à retirer les objets des mémoires cache, des autres serveurs mandataires ;
- ICP (Internet Cache Protocol), sur UDP. Le protocole ICP est la base des communications inter-mandataires ;
- Cache digests, particulièrement utile pour améliorer les communications inter-mandataires. Chaque serveur proxy crée, toutes les dix minutes, une petite table qui reprend tous les objets mémorisés. Cette table, appelée cache digest, est transférable vers les serveurs voisins en utilisant le protocole HTTP. Cette façon de procéder réduit considérablement le trafic réseau et le temps de latence nécessaire à la découverte d'un objet. Dans 5 % des cas, un objet présent dans la liste n'est plus sur le disque dur, le contenu du disque évoluant vite. Dans ce cas, SQUID effectue un accès direct au Web ;
- SNMP (Simple Network Management Protocol) ;
- CARP (Cache Array Routing Protocol) ;
- HTCP (Hyper Text Caching Protocol).

3. Configuration matérielle.

Le serveur SQUID utilise une grande quantité de mémoire vive comme mémoire cache car elle est d'accès bien plus rapide qu'un disque dur. Pour chaque objet conservé en mémoire cache disque, il y a 56 octets situés en mémoire vive : les meta données (metadata). C'est ce qu'on appelle la StoreEntry data structure. Pour chaque StoreEntry, il y a en plus en mémoire vive 16 octets destinés à une somme de contrôle. S'il y a un million d'objets sur le disque dur, il faut 70 MO de mémoire vive rien que pour les meta données.

A cela s'ajoutent la mémoire vive utilisée par le démon SQUID lui-même, par les objets téléchargés non encore placés sur disque, par les tables des caches IP, par des informations sur l'état des requêtes en cours, par différentes statistiques mises en place par le démon SQUID... Tant et si bien que si l'administrateur souhaite calculer la quantité de mémoire vive consommée par SQUID, il doit multiplier par deux ou trois la valeur définie à la ligne cache_mem du fichier

/etc/squid/squid.conf. Un ralentissement progressif de SQUID traduit la plupart du temps une quantité de mémoire vive insuffisante.

Les facteurs matériels les plus importants pour la performance d'un démon SQUID sont, par ordre décroissant d'importance, la rapidité du disque dur, la quantité de mémoire vive et la puissance du processeur.

4. Installation et mise en route.

Le démon SQUID est l'exécutable /usr/sbin/squid qui appartient à l'utilisateur root du groupe root. L'arrêt et le redémarrage de SQUID est commandé par le script de démarrage du système /etc/rc.d/init.d/squid.

5. Contourner SQUID.

Lorsque la configuration du navigateur Web est terminée, il est possible utiliser SQUID pour se connecter à Internet. Dans le cas de Netscape Navigator, l'utilisation du bouton "Recharger" (Reload) permet d'aller chercher la page Web directement sur le site d'origine, sans passer par la mémoire cache.

Si, pour des raisons d'authentification (site bancaire...), l'adresse IP source des paquets doit être celle du client Web, il faut indiquer dans la configuration du client l'obligation de ne pas passer par le serveur mandataire pour atteindre telle ou telle adresse IP.

6. Configuration de SQUID.

La configuration de SQUID se fait à l'aide du fichier de configuration /etc/squid.conf. Les lignes qui suivent montrent une configuration de base, apte à fonctionner sur la plupart des machines.

6.1. Ports de communication.

```
http_port 8080
icp_port 3130
```

Le serveur SQUID écoute les requêtes HTTP sur le port 3128, c'est-à-dire sur un port non privilégié, au delà de 1023. Un autre numéro de port souvent utilisé est le 8080 devenu un pseudo-standard des serveurs mandataires. Le démon SQUID peut écouter plusieurs ports HTTP (http_port 3128 8080).

Le port 3130 sert aux requêtes ICP.

6.2. Taille du cache.

```
cache_mem 32 MB
cache_swap_low 90
cache_swap_high 95
```

La taille de mémoire vive que SQUID utilise comme mémoire cache est de 32 MB. Lorsque les 32 MB sont remplis à 95%, les objets les plus anciens et ou les moins utilisés sont détruits jusqu'à ce que le remplissage de la mémoire cache atteigne 90%.

```
maximum_object_size 4096 KB
```

La taille au-delà de laquelle un objet ne sera pas mis en mémoire cache sur le disque est de 4 KB.

```
ipcache_size 1024
ipcache_low 90
ipcache_high 95
```

La mémoire vive destinée au cache IP retient au maximum 1024 entrées. Le cache IP est une table qui conserve les noms d'hôte et les adresses IP correspondantes, ce qui permet, lors de la requête d'un client, d'éviter le recours aux services DNS. La structure d'une entrée du cache IP est celle-ci :


```
nom_d'hôte flags lstref TTL N adresse_IP
```

La signification des différents champs est la suivante :

- nom_d'hôte est le nom de la machine dont l'adresse IP est gardée en mémoire ;
- la variable flags peut prendre les valeurs C (cached) ou L (locked) ;
- lstref (last referenced) est une variable définie en secondes.
- TTL signifie time to live.
- N est le nombre d'adresses IP.
- adresse_IP est l'adresse qui correspond au nom d'hôte.

```
fqdn_cache_size 1024
```

La mémoire vive destinée au cache fqdn est apte à retenir un maximum de 1024 entrées. Les entrées de ce cache retiennent les concordances entre les adresses IP et les noms d'hôte correspondants. Il s'agit donc d'une correspondance inverse au sens du DNS.

La structure d'une entrée du cache fqdn est la suivante :

```
adresse_IP flags lstref TTL N nom_d'hôte
```

La signification de ces champs est identique à celle correspondant à l'option ipcache_size.

6.3. Emplacement du cache et de l'historique.

```
Cache_dir ufs /var/spool/squid/cache 100 16 256
```

Cette option sert à préciser l'endroit du disque dur où va se situer la mémoire cache, en l'occurrence le répertoire /var/squid/cache. Il est permis de spécifier plusieurs lignes cache_dir pour répartir la mémoire cache sur plusieurs disques durs, ce qui augmente fortement les performances de SQUID, surtout lorsqu'un grand nombre d'utilisateurs ont recours à ses services. Le ou les répertoires doivent exister et être accessibles en écriture par le processus SQUID. Par défaut, le répertoire /var/squid/cache est créé lors de l'installation du paquetage et appartient à l'utilisateur squid.

La façon dont SQUID gère la mémoire cache obéit à la norme ufs. Mais le système asyncufs existe aussi. Ce dernier type est plus performant mais aussi plus récent et moins fiable. Il diminue le temps de latence des requêtes adressées à SQUID.

La valeur 100 est le nombre de MB réservé au répertoire /var/squid/cache. Le serveur SQUID est d'autant plus efficace qu'il a de la place. Si l'on réserve un disque dur d'une dizaine de giga octets pour SQUID, il est en général conseillé de donner une taille de 6 à 7 GO au répertoire /var/squid/cache.

Le chiffre 16 représente le nombre de répertoires de premier niveau qui sont créés sous /var/squid/cache. Le chiffre 256 est le nombre de répertoires de second niveau. Tous ces répertoires et sous-répertoires ne contiennent que quelques objets, ils permettent à SQUID de trouver plus rapidement l'information dont il a besoin.

A partir de la version 2, il est possible d'ajouter une ligne cache_dir sans effacer ni perturber la mémoire cache existante. Pour cela, SQUID doit être arrêté afin de réaliser les modifications dans le fichier /etc/squid/squid.conf. Après exécution de ces modifications, le serveur peut être relancé.

```
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log
cache_swap_log /var/log/squid/swap.log
```

Les quatre lignes ci-dessus définissent les fichiers d'historique qui seront utilisés et leur place respective au sein de l'arborescence du système de fichiers. Si un fichier ne doit pas être créé, il suffit de dactylographier le mot none en lieu et place du nom du fichier.

```
debug_options ALL,1
```

Le niveau de débogage est fixé à 1 pour toutes les sections. Le niveau maximal est 9 mais il donne un flux d'informations très important. Les messages de débogage sont stockés dans les fichiers d'historique.

6.4. Support de programmes externes.

```
ftp_user jfc@challe.yi.org
```

Le serveur SQUID est capable d'adresser des requêtes à un serveur FTP. Pour avoir accès aux services FTP, il faut donner un nom d'utilisateur et un mot de passe. L'accès public utilise anonymous comme nom d'utilisateur et une adresse électronique comme mot de passe. Cette adresse est souvent fictive, alors que de rares serveurs FTP demandent une adresse fonctionnelle. Si on veut en utiliser une qui permette en plus de recevoir les éventuels commentaires des administrateurs des services FTP, on ajoutera cette ligne au fichier squid.conf.

```
authenticate_program /usr/sbin/pam_auth /etc/passwd
```

La ligne ci-dessus définit le module d'authentification externe que SQUID va employer si l'administrateur désire mettre en place un contrôle d'accès aux services mandataires via le nom d'utilisateur et son mot de passe. Les modules externes qui sont les plus employés sont NSCA et PAM. Pour les noms et mots de passe, la solution de facilité consiste à d'indiquer à SQUID le fichier /etc/passwd comme référence. Si non, il est possible de créer un fichier indépendant auquel on ajoute les noms et mots de passe avec l'exécutable htpasswd.

Deux remarques sont à émettre. Si plusieurs serveurs mandataires sont en rapport, seul celui qui est en contact direct avec le client est capable de l'authentifier. L'authentification doit donc être désactivée dans les communications inter-mandataires. L'authentification ne fonctionne pas avec les serveurs mandataires transparents.

6.5. Timeouts

```
connect_timeout 120 seconds
```

Certains systèmes d'exploitation, dont Linux, ne stoppent pas les requêtes de connexion aux serveurs Web de façon suffisamment performante. C'est la raison pour laquelle SQUID peut être configuré pour stopper d'autorité les requêtes qui ne seraient pas honorées par exemple après deux minutes.

6.6. Contrôles d'accès.

La restriction d'accès au serveur mandataire est nécessaire car certains utilisateurs malveillants peuvent vouloir utiliser un serveur pour masquer leur identité ou profiter de lui pour améliorer la rapidité de leur propre connexion Internet.

Lors du contrôle d'accès, SQUID vérifie la section ligne par ligne, de haut en bas du fichier. Lorsqu'une ligne convient, SQUID ne va pas plus loin et c'est la première règle valide rencontrée qui est utilisée. Si une requête de connexion à SQUID ne rencontre aucune des règles définies, l'accès par défaut entre en vigueur. Cet accès par défaut est le contraire de ce qu'indique la dernière ligne : si elle montre un deny, l'accès sera autorisé pour tous et vice-versa. Cette règle du contraire a été bâtie en pensant que beaucoup d'administrateurs oublient de placer une règle deny all après avoir autorisé tel ou tel groupe à accéder à SQUID. Si aucune règle n'existe, tout le monde peut utiliser SQUID. Le contrôle d'accès effectué par SQUID se fait sur base du protocole de la requête : ce sont des opérateurs différents qui seront vérifiés selon que la requête utilise le protocole HTTP (opérateur http_access) ou l'ICP (opérateur icp_access). Voici une configuration de base :

```
acl all src 0/0
acl local src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl connexion method CONNECT
acl manager proto cache_object
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
http_access allow local localhost
http_access deny manager
icp_access allow all
http_access deny !Safe_ports
http_access deny connexion !SSL_ports
```

```
http_access deny all
miss_access allow all
```

Les sept premières lignes de la configuration de base commencent par l'option `acl` (access control list) qui définit une classe, c'est-à-dire un ensemble ou groupe, le plus souvent d'utilisateurs. L'option `acl` est suivie du nom donné à la classe, par exemple `all`, `local`, `localhost`... Il est permis de choisir n'importe quel nom à la condition qu'il précise au mieux la classe créée.

La distinction des membres d'une classe repose sur un filtre particulier. Parmi les filtres SQUID propose :

- l'adresse IP source, c'est-à-dire l'adresse IP de la machine qui adresse une requête à SQUID ou bien l'adresse IP destination qui est l'adresse IP du serveur Web sollicité. Les trois premières lignes de la configuration de base définissent trois classes sur base des adresses IP source complétées par leur masque de sous-réseau. Pour utiliser les adresses IP destination, on remplace `src` par `dst`. Il y a moyen de définir plusieurs groupes d'adresses IP qui forment une classe, par exemple `acl local src 192.168.0.0/255.255.255.0`.
- un nom de domaine source ou destination. Si l'administrateur désire limiter l'accès à SQUID aux personnes qui appartiennent au domaine `challe.be`, il suffit de créer une classe basée sur le filtre `srcdomain challe.be`. Pour le domaine de destination, il faut remplacer `srcdomain` pour `dstdomain`.
- certains mots au sein de l'URL demandée. Dans l'exemple qui suit, SQUID refusera de se connecter sur un site qui contient le mot `sexe` :

```
acl urlporno url_regex -i sexe
```

Le mot `regex` signifie `regular expression` et le drapeau `-i` sert à la prise en compte des majuscules aussi bien que des minuscules.

- la date et l'heure courantes. Afin de permettre les connexions à tel ou tel moment de la journée. La syntaxe est :

```
acl utilisateurs time jour heure_début:minute heure_fin:minute
```

où `jour` est `S` (dimanche), `M` (lundi), `T` (mardi), `W` (mercredi), `H` (jeudi), `F` (vendredi) ou `A` (samedi). Il est permis de définir par exemple la classe :

```
acl nuit time 19:00 6:00
```

- le numéro de port destination. Les deux dernières lignes `acl` définissent deux groupes de ports destination. Les ports `443` et `563` sont les ports d'écoute standards des serveurs HTTP et NNTP sécurisés par SSL. Les ports `70` et `210` sont les ports par défaut utilisés par les serveurs Gopher et WAIS.
- le protocole utilisé par SQUID pour satisfaire une requête (FTP, HTTP, SSL...). Dans la configuration de base présentée plus haut, une ligne `acl manager proto cache_object` est présente. Le serveur SQUID utilise le protocole `cache_object` pour donner des informations telles que les statistiques d'utilisation, de performance et d'autres qui ne sont pas destinées à n'importe qui. Cette ligne existe par défaut pour permettre au gestionnaire de cache de gérer et d'obtenir des informations sur la mémoire cache de SQUID. Prenons un autre exemple :

```
acl ftp proto FTP
http_access deny ftp
```

Ces option font en sorte qu'aucune machine du réseau ne pourra adresser une requête à un serveur FTP via SQUID.

- la méthode que le protocole HTTP utilise pour aller chercher les objets. Le serveur SQUID en reconnaît plusieurs. En voici quelques-unes : `GET` (recherches simples et rapatriement d'objets), `HEAD` (rapatriement de meta données), `POST` (pour soumettre des objets à un programme), `PUT` (pour envoyer des objets à un site (uploader)), `DELETE` (pour supprimer des objets), `TRACE`, `OPTIONS`, `CONNECT` (pour les connexions via tunnel SSL), `ICP_QUERY`, `PURGE` (pour enlever les objets du cache), `PROPFIND` (pour connaître les propriétés d'un objet), `PROPATCH`, `MKCOL`, `MOVE`, `COPY`, `LOCK` (pour bloquer les modifications d'un objet), `UNLOCK`... C'est grâce à la requête que lui envoie le client que SQUID sait quelle méthode il doit utiliser.
- le nom d'utilisateur et le mot de passe. Le filtre à mettre en place est `proxy_auth`. Voici un exemple :

```
acl all src 0/0
acl auth proxy_auth REQUIRED
http_access allow auth
http_access deny all
```

Les deux premières lignes de l'exemple ci-dessus définissent les classes all et auth. Les deux dernières lignes règlent les accès à SQUID. Le mot REQUIRED signifie que n'importe quel utilisateur en passe d'être authentifié correctement va faire partie de la classe utilisateurs. Ce mot peut être remplacé par une série de noms d'utilisateurs (acl utilisateurs proxy_auth nathalie laura christine isabelle).

Les lignes de la configuration de base qui suivent celles commençant par acl utilisent ce qu'on appelle des opérateurs (http_access, icp_access, snmp_access...). Les opérateurs définissent le type de fonction de SQUID auquel une classe peut avoir accès. L'opérateur est suivi des mots allow ou deny en fonction de ce que l'administrateur cherche à permettre ou à interdire. Voici les principaux opérateurs :

- http_access, icp_access. L'opérateur http_access est le plus répandu, il sert à ouvrir ou au contraire à limiter l'accès des clients à SQUID.
- no_cache. Cet opérateur sert à prévenir la mise en mémoire cache de certaines pages, les pages locales du réseau par exemple. De cette façon, la mémoire cache est réservée de façon beaucoup plus efficace pour les pages recherchées sur Internet.
- ident_lookup_access. Cet opérateur demande à SQUID de vérifier l'identité des machines qui font appel à ses services et de l'enregistrer dans ses fichiers d'historique. Il ne s'agit pas ici à proprement parler d'une restriction d'accès mais plutôt d'un stockage d'information dans un but de protection contre les attaques extérieures notamment. Pour exemple, les deux règles qui suivent demandent à SQUID de vérifier les identités des machines qui ne font pas partie du réseau local :

```
ident_lookup_access deny local
ident_lookup_access allow all
```

- Puisque cette vérification prend du temps et retarde SQUID, il faut qu'elle soit utilisée à bon escient.
- miss_access. Cet opérateur, suivi de deny, force SQUID à donner uniquement à ses clients des objets déjà présents sur le disque dur. En d'autres termes, il est interdit d'aller chercher un objet sur Internet pour répondre à une requête. Cette option est surtout utile pour régler les relations inter-mandataires : entre serveurs SQUID pairs, seules les données déjà présentes sur le disque dur sont partagées. Un serveur mandataire ne doit pas réaliser une demande d'objets via un autre serveur pair, sauf en cas de tentative de piratage. En créant une classe regroupant les serveurs SQUID pairs et en utilisant cet opérateur suivi de deny, seuls les objets sur le disque dur seront échangés.
- delay_classes. Cet opérateur permet de diminuer la bande passante attribuée à certains sites. On peut imaginer, dans une institution scolaire, de diminuer la bande passante attribuée à des serveurs de jeux.

Il est souvent nécessaire de combiner plusieurs classes par opérateur. Les classes utilisées dans les exemples qui suivent ont été reprises des exemples précédents :

http_access deny local nuit va interdire au membres de la classe local l'accès au serveur SQUID pendant la nuit.

http_access deny all !local va interdire l'accès au serveur SQUID à toutes les adresses IP source à l'exception de celles du réseau local.

http_access allow local auth va autoriser l'accès aux seuls membres du groupe local et cela après authentification.

Les combinaisons de classes rendent la configuration de SQUID très souple.

6.7. Paramètres administratifs.

```
cache_mgr root@challe.be
cache_effective_user squid
cache_effective_group nogroup
```

En cas de problèmes, un courrier électronique sera envoyé à root. C'est au lancement de l'exécutable /usr/sbin/squid appartenant à l'utilisateur root et au groupe du même nom que pour des raisons de sécurité le démon va subir le changement d'identité indiqué ici.

6.8. Vérification de la connexion.

```
dns_testnames netscape.com internic.net nlanr.net microsoft.com
```

Cette ligne définit les noms de domaines que SQUID va tester lors de son lancement afin de déterminer si la connexion Internet est opérationnelle.

7. Configuration avancée.

7.1. Bannir des sites.

Certaines sociétés ne souhaitent pas que leurs employés puissent visualiser le contenu de sites sensibles. Pour cela, la configuration de SQUID peut être adaptée en créant des règles d'interdiction de consultation de sites. Voici un exemple d'une telle configuration :

```
http_port 8080
ftp_user jfc@challe.yi.org
acl all src 0.0.0.0/0.0.0.0
acl local src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl bannirsites url_regex -i "/etc/squid/bannir"
http_access deny bannirsites
http_access allow local
http_access allow localhost
http_access deny all
```

Les options acl définissent quatre groupes :

- toutes les adresses IP possibles (all) ;
- les adresses IP du réseau local (local) ;
- l'adresse loopback de la machine hébergeant le serveur (localhost) ;
- les sites à bannir (bannirsites).

Contrairement aux options http_access l'ordre de définition des groupes n'a aucune importance. Lorsque SQUID doit déterminer si un accès est autorisé ou interdit, il parcourt la liste des options http_access dans l'ordre de leur apparition. Dès qu'une option http_access est valable, l'exploration des règles d'accès prend fin. Cela signifie, par exemple, qu'il ne faut pas placer la règle http_access deny all en premier lieu. En effet, cette règle sera toujours valable, ce qui interdira toute utilisation du serveur mandataire. L'algorithme de détermination si un client a le droit d'employer le serveur mandataire est le suivant :

```
si demande d'accès à un site interdit alors
  refuser l'accès
sinon
  si le client fait partie du réseau local alors
    autoriser l'accès
  sinon
    si le client est la machine serveur alors
      autoriser l'accès
    sinon
      interdit l'accès
  fsi
fsi
```

Cette imbrication de vérification peut amener l'administrateur à autoriser un client à tout effectuer. Supposons que l'utilisateur de la machine 192.168.1.80 soit autorisé à accéder aux sites bannis. Cette configuration exige l'ajout d'une classe correspondant à la machine qui doit être autorisée à tout faire, l'ajout aussi d'une règle d'accès placée au début de la liste des options http_access autorisant toujours cette machine à employer les services de SQUID. Voici un exemple de ce type de configuration :

```
http_port 8080
ftp_user jfc@challe.yi.org
acl all src 0.0.0.0/0.0.0.0
acl local src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl bannirsites url_regex -i "/etc/squid/bannir"
acl P2366 src 192.168.1.80/255.255.255.255
http_access allow P2366
http_access deny bannirsites
http_access allow local
http_access allow localhost
http_access deny all
```

Dans les deux exemples qui précèdent, l'option `acl bannirsites` se fonde sur une vérification d'URL présentes dans le fichier `/etc/squid/bannir`. Ce fichier contient la liste des sites qui doivent être inaccessibles. Voici un exemple d'un tel fichier :

```
[root@p200 squid]# cat bannir
http://www.msn.com
http://www.microsoft.com
```

Les sites dont l'URL débute soit par `http://www.msn.com`, soit par `http://www.microsoft.com` ne peuvent être rejoints par les clients du serveur mandataire. Notons que l'ensemble de ces sites est banni car les différentes lignes du fichier représentent des expressions régulières.

Si l'administrateur effectue des modifications dans le fichier des sites bannis, il faut impérativement relancer le serveur SQUID, les expressions régulières présentes dans ce fichier n'étant lues que lors du lancement du serveur mandataire.

7.2. Interdire le téléchargement de fichier exécutables.

L'accès à Internet peut amener les utilisateurs à rechercher des programmes pour les installer sur leur machine. Ce type de comportement est de nature à corrompre l'intégrité des systèmes et à obliger le service informatique à réinstaller régulièrement les PC des utilisateurs. Le serveur SQUID peut être configuré de manière à interdire le téléchargement de fichiers exécutables sous Windows et donc l'installation illicite de programmes.

Le téléchargement d'un exécutable consiste à demander une URL dont le nom se termine par `.exe`. L'expression régulière `.exe$` stipule toutes les URL se terminant par `.exe`. L'interdiction de téléchargement de ces fichiers consiste simplement en l'ajout d'une règle au niveau du fichier des sites bannis comme le montre l'exemple suivant :

```
[root@p200 squid]# cat bannir
http://www.msn.com
http://www.microsoft.com
.exe$
```

7.3. Authentification des utilisateurs.

L'authentification des utilisateurs consiste à demander un nom et un mot de passe lors l'ouverture d'une session. Ce couple d'informations est transmis à un programme d'authentification qui retourne la chaîne OK si l'utilisateur a le droit de se connecter et la chaîne ERR s'il n'a pas l'autorisation d'employer le serveur mandataire. Cette authentification peut être réalisée par un programme que l'administrateur peut écrire lui-même.

Pour simplifier l'écriture de ce programme, le nom de l'utilisateur ainsi que son mot de passe sont codés directement dans le programme. Naturellement, une utilisation appropriée de cette fonctionnalité consiste à créer une base de données des noms des utilisateurs et des mots de passe associés qui sera exploitée par le programme d'authentification.

Lorsqu'un navigateur envoie l'identification d'un utilisateur, il le fait sur la base d'une chaîne de caractères contenant dans cet ordre, le nom de l'utilisateur, un espace et le mot de passe. Ces deux informations sont terminées par le caractère de retour de chariot (`\n`).

Le programme d'authentification doit effectuer une boucle jusqu'au moment où la fin du fichier est rencontrée au niveau de l'entrée standard.

Voici un exemple de programme permettant de gérer les authentifications :

```
[root@p200 squid]# cat auth.c
#include <stdio.h>
#include <string.h>
#define BUFSIZE 512
void main () {
    char buffer[BUFSIZE],*p;
    int continuer=1;
    if (setvbuf(stdout,NULL,_IOLBF,0)==0)
        while (continuer) {
            if (fgets(buffer,BUFSIZE,stdin)==NULL)
                continuer=0;
            else {
                if ((p=strchr(buffer,'\n'))!=NULL)
                    *p='\0';
                if ((p=strchr(buffer,' '))==NULL)
                    printf("ERR\n");
                else {
                    *p++='\0';
                    if ((strcmp(buffer,"jfc")==0)&&(strcmp(p,"azerty")==0))
                        printf("OK\n");
                    else
                        printf("ERR\n");
                }
            }
        }
}
```

L'activation du processus d'authentification nécessite la modification du fichier de configuration de SQUID. L'option `authenticate_program` détermine l'emplacement du programme d'authentification alors que l'option `acl passwd proxy_auth REQUIRED` exige que tout utilisateur correctement identifié fasse partie de cette classe. Pour autoriser l'accès, il faut ajouter une option `http_access` comme le montre l'exemple suivant :

```
[root@p200 squid]# cat squid.conf
http_port 8080
ftp_user jfc@challe.yi.org
authenticate_program /etc/squid/auth
acl all src 0.0.0.0/0.0.0.0
acl local src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl bannirsites url_regex -i "/etc/squid/bannir"
acl passwd proxy_auth REQUIRED
http_access deny bannirsites
http_access deny !local !localhost
http_access allow passwd
http_access deny all
```


XIV. IPCHAINS.

1. Introduction.

Un firewall est un routeur filtre de paquets implanté dans le système d'exploitation. Son champ d'opération couvre le réseau IP et les couches du protocole de transport. Il protège le système en prenant des décisions de routage après avoir filtré les paquets en se basant sur l'information contenue dans l'en-tête d'un paquet IP.

Un firewall filtre de paquets consiste en une liste de règles d'acceptation et de refus. Ces règles définissent explicitement les paquets qui seront et ne seront pas autorisés à traverser l'interface réseau. Les règles utilisent les champs d'en-tête du paquet pour décider si un paquet doit être :

- routé vers sa destination ;
- mis de côté de façon non apparente ;
- bloqué et retourné, avec un message d'erreur, à la machine émettrice.

Ces règles sont fondées sur :

- la carte d'interface réseau ;
- l'adresse IP de l'hôte ;
- l'adresse IP source ;
- l'adresse IP destination ;
- le numéro de port du service UDP ou TCP ;
- les indicateurs de connexion TCP ;
- les types de messages ICMP ;
- l'entrée ou la sortie du paquet.

Un firewall filtre de paquets fonctionne sur les couches réseau (IP et ICMP) et transport (UDP et TCP) du modèle de référence TCP/IP.

L'idée générale suppose le contrôler avec attention de ce qui passe entre la machine et Internet auquel elle y est directement connectée. Il faut filtrer individuellement aussi exactement que possible ce qui vient de l'extérieur et ce qui sort de la machine. Le firewall filtre, indépendamment ce qui entre et ce qui sort à travers l'interface. Le filtrage d'entrée et le filtrage de sortie peuvent reposer sur des règles complètement différentes sont appelées chaînes, les listes des règles définissant ce qui peut entrer et ce qui peut sortir sont appelées chaînes. Elles sont ainsi appelées car un paquet est comparé individuellement à chaque règle de la liste jusqu'à ce qu'une correspondance soit trouvée ou que la liste soit épuisée.

Cette configuration est relativement puissante mais n'est pas un mécanisme de sécurité très fiable. Elle constitue seulement une partie de la stratégie de sécurité qui doit être mise en place. Ce type de sécurité est d'un niveau trop bas pour permettre une authentification et un contrôle d'accès raffinés. Ces services de sécurité doivent être fournis à des niveaux plus élevés. Le protocole IP ne possède pas la capacité de vérifier l'identité de l'émetteur. La seule information d'identification porte sur l'adresse IP de l'émetteur du paquet. Ni le réseau, ni la couche transport ne peuvent vérifier que l'adresse IP est exacte. Néanmoins, par rapport à ce qui peut être mis en œuvre à des niveaux plus élevés, le niveau paquet permet un contrôle étendu de l'accès direct aux ports.

2. Choix d'une politique de filtrage.

Chaque chaîne du firewall comprend une politique par défaut et une série d'actions à entreprendre en réponse à des types spécifiques de messages. Chaque paquet est vérifié à la lumière de chaque règle de la liste jusqu'à ce qu'une correspondance soit trouvée. Si le paquet ne correspond à aucune règle, il n'aboutit pas et la politique par défaut est appliquée au paquet. Deux approches possible s'appliquent à un firewall :

- tout ce qui n'est pas explicitement autorisé est interdit ;
- tout ce qui n'est pas explicitement interdit est autorisé.

La politique du tout interdire est l'approche recommandée. Cette approche permet plus aisément la configuration d'un firewall sécuritaire, mais chaque service souhaité doit être explicitement activé. Cela signifie qu'une bonne

compréhension du protocole de communication à activer est nécessaire. Cette approche requiert plus de travail avant de rendre disponible un accès à Internet.

La politique du tout accepter permet une mise en œuvre aisée et immédiate, mais elle oblige l'administrateur à anticiper chaque type d'accès concevable à désactiver. Le danger consiste à ne pas anticiper un type d'accès dangereux avant qu'il ne soit trop tard ou ultérieurement à rendre disponible un service peu sûr. Développer un firewall, selon cette politique, présente beaucoup plus de travail et de difficultés et augmente considérablement le risque d'erreurs.

3. Rejeter ou refuser un paquet.

Le mécanisme du firewall offre un choix : soit rejeter (reject) soit refuser (deny) les paquets. Lorsqu'un paquet est rejeté, il est mis de côté et un message d'erreur ICMP est renvoyé à l'émetteur. Lorsqu'un paquet est refusé, il est simplement mis de côté sans aucune notification à l'émetteur.

Le refus est souvent le meilleur choix. Il y a trois raisons à cela :

- l'envoi d'une réponse d'erreur augmente le trafic sur le réseau. La majorité des paquets ignorés le sont parce qu'ils sont malveillants et non parce qu'ils représentent une tentative innocente d'accéder à un service qui ne se trouve pas offert ;
- tout paquet ayant donné lieu à une réponse peut être utilisé pour une attaque par déni de service ;
- toute réponse, même un message d'erreur, procure à un cracker potentiel de l'information utile.

4. Filtrage des paquets entrants.

Le côté par lequel entre l'information sur l'interface externe est le plus intéressant pour sécuriser un site. Un filtre de paquets peut être basé sur :

- l'adresse source ;
- l'adresse de destination ;
- le port source ;
- le port destination ;
- l'identificateur de l'état TCP.

4.1. Filtrage d'adresse source distante.

Au niveau du paquet, le seul moyen d'identifier l'émetteur est l'adresse IP présente dans l'en-tête du paquet. Cela ouvre la porte au spoofing d'adresses source là où l'émetteur indique dans le champ source une adresse erronée, en lieu et place de sa propre adresse. L'adresse indiquée peut, être inexistante ou être celle de quelqu'un d'autre. Cela peut permettre à des paquets louches de s'introduire dans le système, usurpant l'identité de la machine visée afin d'attaquer d'autres sites.

4.1.1. Spoofing d'adresses source et adresses illégales.

Il existe six classes majeures d'adresses source devant être refusées sur une interface externe. Il s'agit des paquets entrants se réclamant ainsi :

- adresse IP du firewall. Aucun paquet entrant ne peut légalement afficher l'identité du firewall. Puisque l'adresse source est la seule information disponible et qu'elle peut être modifiée, il s'agit d'une forme de spoofing détectable au niveau du filtrage de paquets ;
- adresses IP privées de classe A, B et C. Les séries d'adresses dans chacun des intervalles des classes A, B et C sont réservées à une utilisation au sein d'un réseau local. Elles ne sont pas destinées à un usage sur Internet. Comme telles, ces adresses peuvent être utilisées par n'importe quel Intranet sans qu'il lui soit nécessaire d'acquérir des adresses IP enregistrées en bonne et due forme. Les adresses source privées sont souvent rencontrées sur les sous-réseaux locaux des fournisseurs d'accès à Internet en raison de systèmes mal configurés aussi bien que sur des sites ayant fait l'objet d'une attaque par spoofing. Les adresses privées de classe A se voient assigner les intervalles de 10.0.0.0 à 10.255.255.255. Les adresses privées de classe B se voient assigner les intervalles de 172.16.0.0 à 172.31.255.255. Les adresses privées de classe C se voient assigner les intervalles de 192.168.0.0 à 192.168.255.255 ;

- Adresses IP multicast de classe D. Les adresses IP, dans l'intervalle de la classe D, sont mises à part pour une utilisation en tant qu'adresses destination lorsqu'elles participent à un broadcast réseau en mode multicast. Elle se répartissent entre 224.0.0.0 et 239.255.255.255 ;
- adresses IP réservées de classe E. Les adresses IP dans l'intervalle de la classe E sont mises de côté pour une utilisation future à titre expérimental et ne sont pas assignées au public. Elles s'étendent de 240.0.0.0 à 247.255.255.255 ;
- adresses d'interface loopback. L'interface loopback est une interface privée utilisée par les systèmes UNIX pour les services locaux. Plutôt que d'envoyer le trafic local à travers le pilote de l'interface réseau et pour une meilleure performance le système d'exploitation emprunte un raccourci à travers l'interface loopback. Par définition, le trafic loopback est destiné à être généré par le système. Il ne sort pas du réseau. L'adresse loopback va de 127.0.0.0 à 127.255.255.255. Généralement les systèmes s'y réfèrent sous la forme 127.0.0.1, localhost ou interface loopback, lo ;
- adresses broadcast mal formées. Les adresses broadcast sont des adresses spéciales qui s'appliquent à toutes les machines sur un réseau. L'adresse source de broadcast sera soit 0.0.0.0, soit une adresse IP normale. Seuls les clients DHCP verront des paquets broadcast entrants provenant de l'adresse source 0.0.0.0. Il ne semble pas que l'adresse 0.0.0.0 soit rencontrée dans d'autres situations. Il ne s'agit pas d'une adresse point à point valable.

4.1.2. Bloquer les sites problématiques.

Une autre façon de mettre en place un plan de filtrage d'adresses source est de bloquer tous les accès à une machine sélectionnée ou à un bloc d'adresses IP. C'est de cette manière que la communauté Internet tend à réagir envers les sites problématiques. Si un site se bâtit une mauvaise réputation sur Internet, les autres sites essaient de le bloquer d'emblée. En d'autres termes, un site de mauvaise réputation se verra refusé l'accès aux services offerts par la machine firewall.

4.1.3. Limiter les paquets entrants à des hôtes sélectionnés.

Il se pourrait que l'administrateur ne souhaite accepter que certains genres de paquets entrants provenant uniquement de sites externes spécifiques. Dans ces cas, les règles du firewall définiront soit des adresses IP spécifiques, soit un intervalle limité d'adresses source dont les paquets seront acceptés.

La première classe de paquets entrants vient de serveurs distants répondant aux requêtes du firewall. Si certains services, tels que les services Web ou FTP, peuvent provenir de partout, d'autres services viendront de façon légitime uniquement du fournisseur d'accès à Internet ou d'hôtes de confiance spécialement choisis. Les exemples sont :

- le service POP ;
- l'assignation dynamique d'adresses IP ;
- les réponses DNS.

La seconde classe de paquets entrants vient de clients distants accédant aux services offerts par le site mis en place. Là encore, alors que certaines connexions de services peuvent provenir de partout (Web), d'autres services locaux ne seront offerts qu'à un petit nombre d'utilisateurs distants, de confiance. Les exemples de services locaux restreints pourraient être telnet, ssh, finger.

4.2. Filtrage local d'adresses destination.

Filtrer les paquets en se basant sur l'adresse destination est de moindre importance. La carte d'interface réseau ignore les paquets qui ne lui sont pas adressés. Les paquets broadcast font exception car ils seront retransmis à tous les hôtes du réseau.

4.3. Filtrage de port source distant.

Le port source des paquets entrants identifie le programme de l'hôte distant qui envoie le message. Les requêtes et les connexions entrantes, provenant de clients distants et allant vers les services offerts par le site à sécuriser, auront un port source dans le rang des ports sans privilège. Par exemple, l'hôte de serveur Web sera sollicité par des ports sources compris entre 1024 et 65535. Les réponses entrantes provenant de serveurs distants qui ont été contactés auront un port source correspondant au service particulier appelé. Lors de la connexion à un site Web distant, tous les messages entrants venant du serveur distant auront le port source 80 (le numéro de port du service http).

4.4. Filtrage de port destination local.

Le port destination des paquets entrants identifie le programme ou le service auquel le paquet est destiné. Les requêtes entrantes allant vers les services locaux auront comme port destination le numéro de service assigné au service particulier appelé. Un paquet entrant destiné au serveur Web aura le numéro de port destination 80.

4.5. Filtrage de l'état de connexion TCP.

Les règles d'acceptation des paquets TCP peuvent faire usage des indicateurs d'état de la connexion. Toutes les connexions TCP disposent d'un champ état de connexion en raison du protocole d'authentification en trois phases. Les paquets TCP entrants issus de clients distants auront l'indicateur SYN, positionné uniquement dans le premier paquet reçu. Tous les paquets entrant après la première requête de connexion, auront l'indicateur ACK positionné. Les règles du firewall autoriseront les paquets à destination des services offerts quel que soit l'état des indicateurs. Par contre, les paquets entrants provenant de serveurs distants seront toujours des réponses à la requête initiale de connexion lancée par le programme client s'exécutant localement. Chaque paquet reçu d'un serveur distant aura l'indicateur ACK positionné. Les règles du firewall impliqueront que tous les paquets entrants provenant de serveurs distants soient des réponses à des requêtes (indicateur ACK positionné).

4.6. Sondes et balayages.

Une sonde est une tentative de connexion à un port. Un balayage est un ensemble de sondes vers des ports différents.

Intrinsèquement, les sondes et les balayages sont inoffensifs. Sur Internet, la seule façon de découvrir si un site offre un service particulier est de sonder le port correspondant. Malheureusement, les sondes et les balayages deviennent de moins en moins innocents. Ils sont davantage une phase initiale de recherche des vulnérabilités avant une attaque en règle.

4.6.1. Balayages généraux de ports.

Des balayages généraux de ports sont des sondes aveugles s'étendant sur un large ensemble de ports de services. Ces balayages sont probablement générés par d'anciens outils de maintenance réseau. Ces outils sont devenus moins fréquents en raison de la disponibilité d'outils plus sophistiqués.

4.6.2. Balayages ciblés de ports.

Les balayages ciblés de ports visent des vulnérabilités spécifiques. Les outils récents tentent d'identifier :

- le matériel;
- le système d'exploitation ;
- les versions de logiciels.

Les cibles sont souvent sondées en vue de la recherche des vulnérabilités spécifiques, tel un serveur de mails non sécurisé. Chaque jour, les journaux des firewalls font état de tentatives de connexions ratées de toutes sortes. Il s'agit généralement de tentatives de piratages avortées ne compromettant pas l'intégrité du système car les ports sont bloqués, le firewall remplissant bien son office

4.7. Attaques par déni de service.

Les attaques par déni de service (Denial Of Service) consistent à inonder un système de paquets de telle sorte que la connexion Internet sera interrompue ou sérieusement dégradée, poussant les serveurs locaux à leurs extrêmes limites et faisant en sorte que les requêtes réelles soient impossibles à recevoir ou dans le pire des cas, que le système plante complètement.

Il est impossible de protéger complètement un système contre les attaques par déni de service car le nombre de formes possibles dépendent uniquement de l'imagination du pirate. Toutefois, ces attaques empruntent habituellement une des nombreuses formes classiques :

- TCP ou SYN flooding ;
- ping flooding ;
- UDP flooding ;
- Les bombes ICMP-redirect.

4.7.1. TCP et SYN flooding.

Une attaque de type TCP flooding ou SYN flooding consomme les ressources du système jusqu'à ce qu'aucune nouvelle connexion TCP ne soit possible. L'attaque se sert du protocole d'authentification en trois étapes conjointement avec le spoofing de l'adresse source.

L'attaquant déguise son adresse source et initie une connexion vers un des services TCP. Tout comme un client qui tente une connexion TCP, le meneur de l'attaque envoie un message SYN. Le serveur y répond en envoyant un accusé de réception SYN-ACK. Toutefois, dans ce cas, l'adresse à laquelle le serveur répond n'est pas celle d'où provient l'attaque car il s'agit d'une adresse inexistante. L'étape finale de l'établissement de la connexion TCP qui est de recevoir un ACK en réponse, n'aura jamais lieu. Par conséquent, les ressources de connexion réseau, de taille finie, sont absorbées. Les connexions demeurent demi-ouvertes jusqu'à ce que chaque tentative de connexion ait atteint le temps limite imparti. Le pirate inonde le port de requêtes de connexion plus vite que le temps d'expiration imparti pour pouvoir libérer des ressources. Si l'attaque persiste, toutes les ressources seront en cours d'utilisation et aucune nouvelle requête de connexion ne pourra être acceptée. Si la cible est le port smtp, il sera impossible de recevoir du courrier électronique. Si le port http est visé, personne ne pourra se connecter au site Web.

Plusieurs solutions sont disponibles. L'une des plus connue est le filtrage des adresses source en bloquant les adresses spoofées les plus communément utilisées.

4.7.2. Ping flooding.

Tout message qui provoque une réponse du serveur peut être utilisé pour dégrader la connexion réseau en forçant le système à dépenser une grande partie de son temps en réponses. Le message ICMP echo-request envoyé par un ping est souvent accusé dans ce cas. De plus, un vieil exploit appelé ping de la mort (Ping of Death) impliquait l'envoi de paquets ping de taille élevée. Le résultat en est que les systèmes vulnérables peuvent planter. Plusieurs systèmes UNIX dont Linux sont invulnérables à ce ping de la mort. Puisque historiquement, ping a été impliqué dans des attaques par déni de service, plusieurs sites ne répondent plus aux requêtes ping extérieures.

4.7.3. UDP flooding.

Le protocole UDP est particulièrement utile en tant qu'outil de déni de service. Contrairement à TCP, UDP est sans état. Les mécanismes de contrôle de flux ne sont pas inclus. Il n'y a pas d'indicateur d'état de connexion. Les numéros de séquence des datagrammes ne sont pas utilisés. Aucune information n'est conservée sur la nature du prochain paquet attendu. Il est relativement aisé de faire en sorte qu'un système soit si occupé à répondre aux sondes UDP qu'aucune bande passante ne soit laissée au trafic réseau légitime.

Parce que, de façon inhérente, les services UDP sont moins sécurisés que les services TCP, plusieurs sites désactivent tous les ports UDP qui ne sont pas absolument nécessaires.

4.7.4. Bombes ICMP-redirect.

Un message ICMP-redirect indique au système cible qu'il doit changer ses tables de routage pour favoriser un chemin plus court. Si le serveur exécute les démons gated et routed et honore les messages de redirection, il est possible à un pirate de tromper le système cible de façon à ce qu'il prenne la machine de l'attaquant pour une machine du réseau local ou une machine du fournisseur d'accès à Internet.

4.7.5. Autres attaques par déni de service.

La connectivité réseau n'est pas la seule concernée par les attaques par déni de service. Voici quelques exemples :

- un système de fichiers peut être saturé par l'enregistrement d'une grande quantité :
 - de messages dans les fichiers journaux ;
 - de courriers électroniques de grande taille.

Il pourrait être judicieux de configurer les limites des ressources et de prévoir une partition séparée pour les systèmes de fichiers qui croissent rapidement ;

- les serveurs peuvent être bloqués si un grand nombre de données leur sont envoyées et que les buffers débordent. Plusieurs faiblesses connues sont les buffers overflows.

5. Filtrage de paquets sortants.

Si l'environnement est sûr, le filtrage des paquets sortants n'est pas aussi critique que le filtrage des paquets entrants. Le système ne répondra pas à des messages entrants auxquels le firewall fera opposition. Mais le filtrage symétrique est encore plus sécurisé car il protège aussi les autres hôtes du réseau des erreurs pouvant survenir au niveau de la machine firewall.

Le pire des scénarios serait qu'un pirate réussisse à obtenir un accès au firewall. Par réaction, le filtrage de paquets sortants fournit alors un peu plus de protection jusqu'à ce que le pirate obtienne un accès root et désactive le firewall.

Le filtrage des messages sortants permet d'exécuter des services sur le réseau local sans diffuser de paquets vers Internet. Ce type de filtre est particulièrement utile dans le cas où un serveur DHCP est présent, alors les paquets ne doivent pas être envoyés vers Internet.

5.1. Filtrage d'adresse source locale.

Le filtrage des paquets sortants basé sur l'adresse source est aisé. L'adresse source est toujours l'adresse IP de l'ordinateur. Il n'y a aucune raison de permettre à des paquets sortants d'avoir une autre adresse source.

5.2. Filtrage d'adresse de destination distante.

Tout comme pour les paquets entrants, l'administrateur peut permettre à certains types de paquets sortants d'être adressés uniquement à des réseaux distants spécifiques ou à des machines individuelles.

La première classe de paquets sortants à filtrer selon l'adresse de destination sont les paquets destinés à des serveurs distants qui ont été contactés par la machine firewall. Les paquets destinés à des serveurs Web ou à des serveurs FTP doivent théoriquement atteindre toute zone d'Internet. Par contre, des services comme POP ne sont accessibles qu'à travers certains serveurs où les utilisateurs ont souscrit un abonnement. C'est également le cas des serveurs de groupes de discussions.

La seconde classe de paquets sortants à filtrer selon l'adresse de destination sont les paquets destinés à des clients distants qui accèdent à un service offert par le site en cours de sécurisation. Par exemple, les réponses du serveur Web ne seront acceptées qu'à la condition que le destinataire soit un site ami. Des exemples de services locaux à restreindre seraient telnet, ssh et finger. Non seulement les règles du firewall refusent les connexions entrantes vers ces services mais les règles ne devraient pas davantage permettre des réponses sortantes pour ces services. C'est donc une question de symétrie.

Certains éditeurs de logiciels ont tendance à forcer les machines clientes à leur envoyer des informations à l'insu des utilisateurs. L'administrateur peut dans ce cas interdire la sortie de paquets vers ces destinations.

5.3. Filtrage de port source local.

Définir explicitement les ports qui pourront être employés pour des connexions sortantes dépend :

- des programmes clients ;
- des programmes serveurs ;

Les connexions émanant des programmes clients proviennent la plupart du temps d'un port source sans privilège. Le firewall doit donc imposer aux clients l'emploi de ports sans privilège.

Les paquets sortants émanant des programmes serveurs proviennent toujours d'un port source déterminé. Les serveurs doivent donc être limités, par une règle du firewall, à un port déterminé. Il est par exemple exclu qu'un serveur Web présente un port source autre que 80.

5.4. Filtrage de port destination distant.

Les programmes clients locaux sont conçus pour être connectés à des serveurs réseau offrant leurs services à partir de ports spécifiques. Dans cette perspective, il s'agit de limiter les clients locaux à une connexion, seulement au port qui leur est associé. Cela permet au firewall d'interdire les balayages de ports.

5.5. Filtrage de l'état des connexions TCP sortantes.

Les règles d'acceptation des paquets TCP sortants peuvent faire usage des indicateurs d'état de connexion associés aux connexions TCP.

Les paquets TCP sortants provenant de clients locaux portent l'indicateur SYN, placé dans le premier paquet envoyé en tant que partie de la procédure d'authentification en trois phases. La requête de connexion initiale ne présente pas l'indicateur ACK. Par la suite, tous les paquets sortants ne porteront que l'indicateur ACK. Les règles du firewall doivent autoriser les clients locaux à émettre des paquets sortants soit avec l'indicateur SYN, soit avec l'indicateur ACK.

Les paquets sortants provenant des serveurs locaux sont toujours des réponses à la requête initiale de connexion initiée à partir d'un programme client distant. Chaque paquet envoyé à partir d'un serveur local doit donc porter l'indicateur ACK. Les règles du firewall doivent ainsi imposer, aux paquets sortants provenant des serveurs locaux, à porter l'indicateur ACK.

6. Services réseau privés/publics.

Une des manières les plus aisées de faciliter les intrusions est de permettre un accès extérieur aux services locaux qui sont désignés uniquement pour un usage local. Certains services, s'ils sont offerts localement, ne devraient jamais franchir la frontière entre le réseau local et Internet. Certains de ces services ennuient le voisinage (serveur DHCP), d'autres fournissent de l'information qu'il serait préférable de réserver à un usage privé.

La façon la plus simple de protéger un site est de ne pas offrir de service. Néanmoins si un service doit être utilisé localement, cette règle ne peut être appliquée. Les services comme NFS, basé sur les RPC sont reconnus comme étant malaisés à sécuriser au niveau du filtrage de paquets. Une manière de préserver l'ordinateur est de ne pas héberger de services réseau qui ne sont pas destinés à un usage public sur la machine firewall. Si le service n'est pas disponible, rien ne permet à un client distant de s'y connecter.

7. Construction d'un firewall.

Le firewall qui sera construit dans cette section est fondé sur la directive par défaut du tout refuser. Donc, tout le trafic est bloqué par défaut. Les services sont permis seulement par exception à cette règle.

7.1. Le programme ipchains.

Le programme d'administration de firewall ipchains crée les règles individuelles du filtre de paquets pour les chaînes d'input et d'output. Un des aspects le plus important de la définition d'un firewall est l'ordre dans lequel les règles sont définies.

Les règles du filtrage de paquets sont sauvegardées à l'intérieur de chaînes d'input, d'output ou de forward des tables du noyau, dans le même ordre qu'elles ont été définies. Les règles individuelles sont insérées au début de la chaîne ou ajoutées à la fin. L'ordre de définition des règles est celui dans lequel elles seront ajoutées aux tables du noyau et par le fait même, l'ordre dans lequel chaque règle sera comparée à chaque paquet est identique.

Chaque fois qu'un paquet extérieur arrive à l'interface réseau, les champs de son en-tête sont comparés à chacune des règles de la chaîne d'input jusqu'à ce qu'une correspondance soit trouvée. A l'inverse, les champs d'en-tête de chaque paquet acheminé à l'interface réseau provenant de l'intérieur sont comparés à chacune des règles de la chaîne d'output jusqu'à ce qu'une correspondance soit trouvée. Dans chaque direction, lorsqu'une correspondance est trouvée, la comparaison s'arrête et la directive de la règle s'applique, soit accepter, rejeter ou refuser. Si le paquet ne satisfait à aucune règle de la chaîne, alors la directive par défaut s'applique.

Voici le format de la commande ipchains ainsi que la signification des différentes options :

```
ipchains -A|-I [chain] [-i interface] [-p protocol] [[!]y] [-s address
[port[:port]]] [-d address [port[:port]]] -j policy [-l]
```

Option	Description
-A [chain]	Ajoute une règle à la fin d'une chaîne. Les chaînes utilisables sont input, output et forward. Si une chaîne n'est pas spécifiée, la règle s'applique alors à toutes les chaînes.
-I [chain]	Insère une règle au début d'une chaîne. Les chaînes utilisables sont input, output et forward. Si une chaîne n'est pas spécifiée, la règle s'applique alors à toutes les chaînes.
-i <interface>	Spécifie l'interface réseau à laquelle la règle s'applique. Si l'interface n'est pas spécifiée, la règle s'applique à toutes les interfaces.
-p <protocol>	Spécifie le protocole IP auquel la règle s'applique. Si l'option -p n'est pas employée, la règle s'applique à tous les protocoles. Les noms de protocoles supportés sont tcp, udp, icmp et all. Les noms ou les numéros des protocoles de /etc/protocols sont également permis.
-y	L'indicateur SYN doit être inscrit et l'indicateur ACK doit être enlevé du message TCP, indiquant ainsi une requête pour l'établissement d'une connexion. Si l'option -y n'apparaît pas, les bits indicateurs TCP ne sont pas vérifiés.
!-y	L'indicateur ACK doit être inscrit dans le message TCP, indiquant une réponse initiale à une requête de connexion ou une continuation d'une connexion établie. Si !-y n'apparaît pas comme argument, les bit indicateurs TCP ne sont pas vérifiés.
-s <address> [<port>]	Spécifie l'adresse source du paquet. Si une adresse source n'est pas spécifiée, toutes les adresses unicast sources sont sous-entendues. Si un port ou une série de ports est donné, la règle ne s'applique qu'à ces ports. Sans la spécification d'un port, la règle s'applique à tous les ports source. Une série de ports est définie par des nombres identifiant les ports de début et de fin, séparés par deux-points. Si un port est inscrit, une adresse doit être donnée.
-d <address> [<port>]	Spécifie l'adresse de destination d'un paquet. Si une adresse de destination n'est pas spécifiée, toutes les adresses unicast de destination sont sous-entendues. Si un port ou une série de ports est donné, la règle ne s'applique qu'à ces ports. Sans la spécification d'un port, la règle s'applique à tous les ports de destination. Une série de ports est définie par des nombres indiquant les ports de début et de fin, séparés par deux-points. Si un port est inscrit, une adresse doit être donnée.
-j <policy>	Spécifie la directive d'acheminement des paquets, soit ACCEPT, DENY ou REJECT. La chaîne forward peut également prendre la directive MASQ.
-l	Ecrit un message informatif du noyau dans le journal du système, par défaut, chaque fois qu'un message satisfait une règle.

7.2. Initialisation du firewall.

Une firewall se construit à l'aide d'une série de règles constituant un filtre de paquets et définies par des options de la ligne de commande de ipchains. La commande ipchains est exécutée lors de chaque définition d'une règle. Sachant qu'un script de firewall peut comporter plusieurs centaines de règles, cela dénombre les exécutions de la commande ipchains.

L'utilisation de la commande ipchains devrait toujours se faire à l'aide d'un script et non pas en ligne de commande. Il est impératif d'exécuter le script de firewall dans sa totalité car sans cela, le firewall pourrait accepter ou refuser des paquets de manière inappropriée. Lorsque les chaînes sont initialisées et que la politique de tout refuser par défaut est appliquée, tous les services du réseau sont bloqués jusqu'à ce que l'accès à l'interface soit explicitement réactivé. Dans le même ordre d'idée, l'exécution du script de firewall doit toujours être réalisé depuis la console. Il ne faut jamais exécuter le script depuis une machine distante car le trafic distant sera bloqué.

Les règles du firewall étant exécutées dans l'ordre de leur apparition dans la chaîne, il est impératif de les ranger de la plus spécifique à la plus générale.

L'initialisation du firewall couvre une grande quantité de paramètres incluant :

- la définition de constantes utilisées dans le script ;
- la suppression des règles existantes dans les chaînes du firewall ;
- l'établissement d'une politique par défaut pour les chaînes d'input et d'output ;

- le rétablissement de l'interface loopback pour une opération normale du système ;
- le refus d'accès aux hôtes spécifiques ou aux réseaux indésirables ;
- la définition de règles de base pour la protection contre de mauvaises adresses ;
- la protection de certains services fonctionnant sur des ports sans privilège.

7.2.1. Les constantes symboliques.

Un script de firewall est plus aisé à lire et à entretenir si des constantes symboliques sont employées pour les noms et les adresses qui reviennent souvent.

```
#!/bin/bash
INTRANETADDR=192.168.1.1
INTRANETNETWORK=192.168.1.0/24
INTERNETADDR=212.68.198.221
NETMASK="255.255.255.240"
NETWORK="212.68.198.208"
LOOPBACK="127.0.0.0/8"
CLASSEA="10.0.0.0/8"
CLASSEB="172.16.0.0/12"
CLASSEC="192.168.0.0/16"
CLASSED="224.0.0.0/4"
CLASSEE="240.0.0.0/5"
BROADCASTSRC="255.255.255.255"
BROADCASTDEST="0.0.0.0"
INTRANETETH=eth1
INTERNETETH=eth0
FAIPOPSEVER="mail.brutele.be"
FAINEWSSERVER="news.brutele.be"
```

7.2.2. Supprimer les règles existantes.

La première étape de la définition d'une série de règles de filtrage est la suppression des règles existantes dans les chaînes. Si tel n'était pas le cas, chaque nouvelle règle définie serait ajoutée à la suite des règles déjà en place. Les paquets pourraient donc satisfaire à une règle préexistante avant d'atteindre le point de la chaîne nouvellement ajouté.

La commande suivante permet de vider toutes les chaînes.

```
ipchains -F
```

Les chaînes étant vides, le système est dans l'état par défaut où il accepte tout.

7.2.3. Définir la politique par défaut.

L'effet secondaire de l'opération consistant à vider les chaînes est que le système retourne à son état par défaut, incluant la politique de tout accepter. Dans le cas de la construction d'un firewall sécuritaire, il faut établir la politique par défaut du tout refuser.

```
ipchains -P input DENY
ipchains -P output REJECT
ipchains -P forward REJECT
```

Les paquets arrivant au firewall sont refusés sans envoi de message d'avertissement à l'expéditeur alors que les paquets sortants sont rejetés avec émission d'un message d'erreur ICMP vers l'expéditeur interne. La différence pour l'utilisateur final réside, par exemple, dans le fait que si une personne d'un site distant tente de se connecter au site Web local, le navigateur de cette personne restera sans réponse jusqu'à ce que son système affiche un message de délai TCP expiré. Cet utilisateur n'obtiendra aucune indication lui permettant de déterminer si le site ou le serveur Web existe réellement. Par contre, si l'utilisateur local tente de rejoindre un site WEB à distance, le navigateur recevra un message d'erreur indiquant, par exemple, que l'opération n'est pas permise.

A ce point, tout le trafic du réseau est bloqué.

7.2.4. Activer l'interface loopback.

Le firewall doit activer, sans restriction, le trafic loopback. En effet, certaines fonctionnalités du système exploitent l'interface loopback pour offrir des services. C'est, par exemple, le cas de l'interface X window. Lorsque tout est permis, les règles sont simples. Il suffit seulement d'annuler les effets de la politique du tout refuser par défaut en acceptant tout sur l'interface loopback.

```
ipchains -A input -i lo -j ACCEPT
ipchains -A output -i lo -j ACCEPT
```

7.2.5. Le spoofing et les mauvaises adresses source.

Au niveau du filtrage de paquets, une des fausses adresses identifiable avec certitude est l'imitation de l'adresse IP du firewall. Il faut donc établir une règle refusant tous les paquets entrants prétendant provenir du firewall.

```
ipchains -A input -i $INTERNETETH -s $INTERNETADDR -j DENY -1
```

Il n'est pas nécessaire de bloquer les paquets sortants qui sont destinés au firewall. En effet, ces paquets passent toujours par l'interface loopback.

Les adresses IP réservées aux réseaux privés ne peuvent en aucun cas être reçues par le firewall. De même, le firewall ne doit jamais diffuser les adresses privées vers Internet. Voici une série de règles empêchant la réception et la distribution de ces adresses :

```
ipchains -A input -i $INTERNETETH -s $CLASSEA -j DENY
ipchains -A input -i $INTERNETETH -d $CLASSEA -j DENY
ipchains -A output -i $INTERNETETH -s $CLASSEA -j DENY -1
ipchains -A output -i $INTERNETETH -d $CLASSEA -j DENY -1
ipchains -A input -i $INTERNETETH -s $CLASSEB -j DENY
ipchains -A input -i $INTERNETETH -d $CLASSEB -j DENY
ipchains -A output -i $INTERNETETH -s $CLASSEB -j DENY -1
ipchains -A output -i $INTERNETETH -d $CLASSEB -j DENY -1
ipchains -A input -i $INTERNETETH -s $CLASSEB -j DENY
ipchains -A input -i $INTERNETETH -d $CLASSEB -j DENY
ipchains -A output -i $INTERNETETH -s $CLASSEB -j DENY -1
ipchains -A output -i $INTERNETETH -d $CLASSEB -j DENY -1
ipchains -A input -i $INTERNETETH -s $CLASSEB -j DENY
ipchains -A input -i $INTERNETETH -d $CLASSEB -j DENY
ipchains -A output -i $INTERNETETH -s $CLASSEB -j DENY -1
ipchains -A output -i $INTERNETETH -d $CLASSEB -j DENY -1
```

Il faut également refuser les paquets possédant une adresse source réservée à l'interface loopback :

```
ipchains -A input -i $INTERNETETH -s $LOOPBACK -j DENY
ipchains -A output -i $INTERNETETH -s $LOOPBACK -j DENY -1
```

Les paquets broadcast peuvent avoir une adresse source ou une adresse destination illégale. Un paquet entrant possédant une adresse source 255.255.255.255 est certainement illégal car l'adresse source ne peut être que 0.0.0.0. Dans le même ordre d'idée, un paquet entrant possédant une adresse destination 0.0.0.0 est certainement illégal car l'adresse destination ne peut être que 255.255.255.255. En résumé, les paquets entrants ayant une adresse source correspondant à une adresse broadcast de destination doivent être éliminés de même que les paquets entrants ayant une adresse de destination correspondant à une adresse broadcast source. Cette inversion dans l'utilisation des adresses broadcast n'est pas une erreur mais une sonde employée pour identifier les machines utilisant UNIX.

```
ipchains -A input -i $INTERNETETH -s $BROADCASTDEST -j DENY
ipchains -A input -i $INTERNETETH -d $BROADCASTSRC -j DENY -1
```

Les adresses multicast ne sont légales qu'à la condition d'être une adresse de destination. Cela signifie que les paquets possédant une adresse source de type multicast sont des attaques par spoofing.

```
ipchains -A input -i $INTERNETETH -s $CLASSED -j DENY -1
ipchains -A output -i $INTERNETETH -s $CLASSED -j REJECT -1
```

De plus, le firewall ne doit pas envoyer des paquets multicast vers Internet. La règle suivante bloque l'envoi des paquets multicast :

```
ipchains -A output -i $INTERNETETH -d $CLASSED -j REJECT -l
```

Si la machine firewall n'est pas inscrite à un service broadcast, il est permis au moyen de la règle suivante, de refuser tous les paquets multicast entrants :

```
ipchains -A input -i $INTERNETETH -d $CLASSED -j REJECT -l
```

Les adresses de la classe E sont réservées à un usage expérimental et ne sont, par conséquent, pas employées sur Internet. Il faut donc refuser tous les paquets entrants prétendant provenir d'un réseau de classe E.

```
ipchains -A input -i $INTERNETETH -s $CLASSEE -j DENY -l
```

L'Internet Assigned Numbers Authority (IANA) administre la distribution et l'enregistrement des espaces d'adresses IP du monde entier. Quelques blocs d'adresses sont identifiés comme étant réservés à l'IANA. Ces adresses ne devraient pas apparaître dans le réseau public Internet. Voici les plages d'adresses réservées à l'IANA :

Début	Fin	Début	Fin	Début	Fin
1.0.0.0	1.255.255.255	69.0.0.0	69.255.255.255	116.0.0.0	116.255.255.255
2.0.0.0	2.255.255.255	70.0.0.0	70.255.255.255	117.0.0.0	117.255.255.255
5.0.0.0	5.255.255.255	71.0.0.0	71.255.255.255	118.0.0.0	118.255.255.255
7.0.0.0	7.255.255.255	72.0.0.0	72.255.255.255	119.0.0.0	119.255.255.255
23.0.0.0	23.255.255.255	73.0.0.0	73.255.255.255	120.0.0.0	120.255.255.255
27.0.0.0	27.255.255.255	74.0.0.0	74.255.255.255	121.0.0.0	121.255.255.255
31.0.0.0	31.255.255.255	75.0.0.0	75.255.255.255	122.0.0.0	122.255.255.255
37.0.0.0	37.255.255.255	76.0.0.0	76.255.255.255	123.0.0.0	123.255.255.255
39.0.0.0	39.255.255.255	77.0.0.0	77.255.255.255	124.0.0.0	124.255.255.255
41.0.0.0	41.255.255.255	78.0.0.0	78.255.255.255	125.0.0.0	125.255.255.255
42.0.0.0	42.255.255.255	79.0.0.0	79.255.255.255	126.0.0.0	126.255.255.255
58.0.0.0	58.255.255.255	80.0.0.0	95.255.255.255	217.0.0.0	217.255.255.255
60.0.0.0	60.255.255.255	96.0.0.0	111.255.255.255	218.0.0.0	218.255.255.255
65.0.0.0	65.255.255.255	112.0.0.0	112.255.255.255	219.0.0.0	219.255.255.255
66.0.0.0	66.255.255.255	113.0.0.0	113.255.255.255	220.0.0.0	223.255.255.255
67.0.0.0	67.255.255.255	114.0.0.0	114.255.255.255		
68.0.0.0	68.255.255.255	115.0.0.0	115.255.255.255		

Voici les règles du firewall refusant les paquets provenant soit-disant de l'IANA :

```
ipchains -A input -i $INTERNETETH -s 1.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 2.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 5.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 7.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 23.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 27.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 31.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 37.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 39.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 41.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 42.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 58.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 60.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 65.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 66.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 67.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 68.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 69.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 70.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 71.0.0.0/8 -j DENY -l
```

```

ipchains -A input -i $INTERNETETH -s 72.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 73.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 74.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 75.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 76.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 77.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 78.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 79.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 80.0.0.0/4 -j DENY -l
ipchains -A input -i $INTERNETETH -s 96.0.0.0/4 -j DENY -l
ipchains -A input -i $INTERNETETH -s 112.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 113.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 114.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 115.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 116.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 117.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 118.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 119.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 120.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 121.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 122.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 123.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 124.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 125.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 126.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 217.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 218.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 219.0.0.0/8 -j DENY -l
ipchains -A input -i $INTERNETETH -s 220.0.0.0/6 -j DENY -l

```

7.3. Filtrage des messages d'état et de contrôle ICMP.

Les messages de contrôle ICMP apparaissent à la suite d'un certain nombre d'erreurs, ils sont produits par un programme d'analyse de réseau tel que ping et traceroute. Voici les types de messages ICMP les plus courants :

Code	Nom symbolique	Description
0	echo-reply	Réponse à un ping.
3	destination-unreachable	Message d'erreur général ; un routeur le long du trajet est incapable de transmettre le paquet à la destination suivante (utilisé par traceroute).
4	source-quench	Contrôle de flux de niveau IP entre deux routeurs ou entre un routeur et un hôte.
5	redirect	Message d'acheminement renvoyé à l'expéditeur lorsque le routeur détermine qu'il existe un chemin plus court.
8	echo-request	Requête ping.
11	time-exceeded	Message d'acheminement renvoyé lorsque le nombre maximal de hops est dépassé (utilisé par traceroute).
12	parameter-problem	Valeur non attendue se trouvant dans l'en-tête du paquet.

7.3.1. Messages d'erreur d'état et de contrôle.

Les messages ICMP Source Quench sont envoyés lorsqu'une connexion source, habituellement un routeur, expédie des données à la destination suivante, plus vite que celle-ci ne peut l'accepter. Ce message ICMP est donc utilisé comme une forme primitive de contrôle de flux. Ces messages doivent donc être acceptés par le firewall :

```

ipchains -A input -i $INTERNETETH -p icmp -s any/0 4 -d $INTERNETADDR -j ACCEPT
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 4 -d any/0 -j ACCEPT

```

Un message ICMP Parameter Problem est envoyé lorsqu'un paquet reçu contient des données illégales ou inattendues, dans son en-tête. Il en est de même, lorsque le checksum de l'en-tête ne correspond pas au checksum calculé par la machine réceptrice : un message ICMP Parameter Problem est envoyé. Le firewall doit donc laisser passer ce message ICMP :

```
ipchains -A input -i $INTERNETETH -p icmp -s any/0 12 -d $INTERNETADDR -j ACCEPT
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 12 -d any/0
-j ACCEPT
```

Un message ICMP Destination Unreachable est un message général d'erreur que doit laisser passer le firewall si l'administrateur accepte que la machine réponde à un traceroute :

```
ipchains -A input -i $INTERNETETH -p icmp -s any/0 3 -d $INTERNETADDR -j ACCEPT
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 3 -d any/0 -j ACCEPT
```

Un message ICMP Time Exceeded indique que le temps imparti est écoulé ou plus spécifiquement que le nombre maximal de hops d'un paquet a été dépassé. Sur les réseaux actuels, un message Time Exceeded entrant est la plupart du temps une réponse ICMP à une requête traceroute sortante :

```
ipchains -A input -i $INTERNETETH -p icmp -s any/0 11 -d $INTERNETADDR -j ACCEPT
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 11 -d any/0
-j ACCEPT
```

7.3.2. Messages de contrôle ping Echo Request et Echo Reply.

Le firewall doit être capable d'envoyer des requêtes ping (Echo Request) vers tous les hôtes distants. Cela signifie que le firewall doit être en mesure d'envoyer des requêtes Echo Request et de recevoir des Echo Reply. Voici les règles d'acceptation du firewall :

```
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 8 -d any/0 -j ACCEPT
ipchains -A input -i $INTERNETETH -p icmp -s any/0 0 -d $INTERNETADDR -j ACCEPT
```

Les pings provenant d'hôtes distants doivent être acceptés à la condition qu'il soient émis par des hôtes de confiance. Voici un exemple où toutes les demandes provenant de la machine 212.68.198.209 sont acceptées par le firewall :

```
ipchains -A input -i $INTERNETETH -p icmp -s 212.68.198.209/32 8
-d $INTERNETADDR -j ACCEPT
ipchains -A output -i $INTERNETETH -p icmp -s $INTERNETADDR 0
-d 212.68.198.209/32 -j ACCEPT
```

Les attaques smurf ont utilisé des paquets ping, diffusant, en continuité, des messages Echo Request aux hôtes du réseau, messages porteurs d'une fausse adresse IP source, soit celle de la victime. Le résultat en est que chaque machine du réseau bombarde constamment la machine de la victime de messages Echo Reply, engorgeant ainsi complètement la bande passante.

Bien que la politique du tout refuser empêche les attaques smurf, il est intéressant de configurer le firewall pour qu'il enregistre une attaque en cours dans le fichier des événements. Pour enregistrer ces attaques, il faut savoir que l'adresse de destination des paquets peut être :

- l'adresse broadcast de destination (0.0.0.0) ;
- l'adresse réseau du firewall ;
- le masque réseau du firewall ;

Voici une série de règles mémorisant les tentatives d'attaques smurf :

```
ipchains -A input -i $INTERNETETH -p icmp -d $BROADCASTDEST -j DENY -l
ipchains -A output -i $INETNETETH -p icmp -d $BROADCASTDEST -j REJECT -l
ipchains -A input -i $INTERNETETH -p icmp -d $NETMASK -j DENY -l
ipchains -A output -i $INETNETETH -p icmp -d $NETMASK -j REJECT -l
ipchains -A input -i $INTERNETETH -p icmp -d $NETWORK -j DENY -l
ipchains -A output -i $INETNETETH -p icmp -d $NETWORK -j REJECT -l
```

7.4. Protection des services sur les ports TCP sans privilège.

7.4.1. Interdire les connexions Open Window.

Les connexions sortantes dirigées vers un gestionnaire Open Window distant devraient être interdites. L'administrateur doit donc inscrire une règle dans le firewall interdisant l'initiation d'une connexion vers un gestionnaire Open Window. Cela signifie que les paquets sortants dotés de l'indicateur SYN doivent être refusés.

Le gestionnaire Open Window utilise le port 2000 pour accepter les connexions. Voici la règle bloquant l'initiation de ce type de requête :

```
ipchains -A output -i $INTERNETETH -p tcp -y -s $INTERNETADDR -d any/0 2000
-j REJECT
```

Il n'est pas nécessaire de bloquer explicitement les demandes de connexion entrantes vers le port 2000 car le système Linux est dépourvu de gestionnaire Open Window.

7.4.2. Interdire les connexions X Window.

Les connexions aux serveurs X Window distants devraient être interdites. Afin d'autoriser plusieurs connexions concurrentes au serveur X Window, les ports allant de 6000 à 6063 sont utilisés. L'administrateur doit donc bloquer les initiations de connexions vers ces ports. De même les demandes de connexions au serveur X Window local doivent être refusées :

```
ipchains -A output -i $INTERNETETH -p tcp -y -s $INTERNETADDR -d any/0 6000:6063
-j REJECT
ipchains -A input -i *INTERNETETH -p tcp -y -d $INTERNETADDR 6000:6063
-j DENY -l
```

7.4.3. Interdire les connexions au serveur SOCKS.

Le serveur mandataire SOCKS exploite le port 1080. Ce type de service est exploité pour rendre des connexions anonymes. Il faut donc interdire, non seulement, l'utilisation du serveur SOCK local par des clients distants mais également l'exploitation de serveurs SOCK distants depuis le firewall.

```
ipchains -A output -i $INTERNETETH -p tcp -y -s $INTERNETADDR -d any/0 1080
-j REJECT -l
ipchains -A input -i $INTERNETETH -p tcp -y -d $INTERNETADDR 1080 -j DENY -l
```

7.5. Protection des services sur les ports UDP sans privilège.

Contrairement au protocole TCP, le protocole UDP ne possède pas d'état de connexion. La seule possibilité offerte à l'administrateur est tout simplement de bloquer l'accès UDP.

Dans la catégorie des services UDP, le protocole NFS est le principal service concerné par le filtrage. Le service NFS opère sur le port 2049. Il faut donc bloquer toutes les connexions entrantes vers le port 2049. Dans le cas où un serveur NFS n'est pas exécuté sur la machine firewall, la règle suivante n'est pas nécessaire :

```
ipchains -A input -i $INTERNETETH -p udp -d $INTERNETADDR 2049 -j DENY -l
```

Dans de rares cas, le protocole NFS est compilé pour travailler au moyen du protocole TCP. Dans cette situation, il faut bloquer les connexions entrantes et sortantes :

```
ipchains -A input -i $INTERNETETH -p tcp -y -d $INTERNETADDR 2049 -j DENY -l
ipchains -A output -i $INTERNETETH -p tcp -y -d any/0 2049 -j DENY -l
```

7.6. Activation des services Internet de base.

Deux services sont absolument requis, ils sont :

- DNS ;
- ident.

Le DNS effectue la traduction entre les noms d'hôtes et leur adresse IP associée. Le service identd, quant à lui, fournit le nom de l'utilisateur, ou l'ID, associé à une connexion. Cette traduction est couramment demandée par un serveur de courrier distant lorsqu'un email est expédié. Il n'est pas nécessaire d'offrir le service identd. Par contre pour éviter des délais d'expiration trop longs, il faut répondre à une demande de connexion entrante d'une manière ou d'une autre.

7.6.1. Permettre DNS.

Le service DNS est un protocole de communication qui s'appuie à la fois sur UDP et sur TCP. Dans les deux cas, le port 53 est utilisé.

Les requêtes d'équivalence (nslookup) sont normalement faites sous UDP, que ce soit des requêtes d'un client vers le serveur ou entre des serveurs. La communication UDP peut faillir si l'information renvoyée est trop grande pour être contenue dans un paquet DNS UDP. Le serveur place alors, dans l'en-tête du message, un bit indicateur de données tronquées. Dans ce cas, le protocole permet un autre essai sous TCP. En pratique, TCP n'est pas utile lors d'une requête DNS sauf dans le cas d'un transfert de zones entre les serveurs primaire et secondaire.

Le résolveur DNS incorporé au code de la bibliothèque réseau peut requérir la conversion d'un nom symbolique en une adresse. Cette requête est, par exemple, envoyée au serveur DNS du fournisseur d'accès à Internet.

En supposant que le serveur DNS du fournisseur d'accès à Internet soit 212.68.193.32, voici les règles du firewall autorisant les demandes de résolution de noms :

```
ipchains -A output -i $INTERNETETH -p udp -s $INTERNETADDR 1024:65535
-d 212.68.193.32 53 -j ACCEPT
ipchains -A input -i $INTERNETETH -p udp -s 212.68.193.32 53
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Si la taille des informations est trop importante, il est impossible de les inclure dans un datagramme UDP. Dans ce cas, le DNS essaye de résoudre le problème en utilisant le protocole TCP. Voici les règles permettant la communication DNS en TCP :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.32 53 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.32 53
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Dans le cas des DNS, lorsqu'un serveur local ne possède pas l'information demandée par un client, il contacte un serveur distant et lui transfère la requête. Ce type de configuration est couramment employée afin de mettre dans la mémoire cache du serveur de noms un certain nombre de résolutions de noms. A la différence des échanges entre un client et un serveur, les échanges entre les serveurs passent toujours par le port 53 d'un côté comme de l'autre. De plus, les requêtes d'équivalence sont toujours faites au moyen du protocole UDP. Voici les règles permettant les échanges entre les serveurs :

```
ipchains -A output -I $INTERNETETH -p udp -s $INTERNETADDR 53 -d any/0 53
-j ACCEPT
ipchains -A input -I $INTERNETETH -p udp -s any/0 53 -d $INTERNETADDR 53
-j ACCEPT
```

Si le site est constitué de plusieurs machines, il y a de fortes chances pour qu'un serveur DNS soit présent. Ce serveur doit être joignable de manière distante afin de résoudre les noms. Non seulement le serveur doit être consultable par des clients mais également par des serveurs :

```
ipchains -A input -i $INTERNETETH -p udp -s any/0 1024:65535 -d $INTERNETADDR 53
-j ACCEPT
ipchains -A output -i $INTERNETETH -p udp -s $INTERNETETH 53 -d any/0 1024:65535
-j ACCEPT
ipchains -A input -i $INTERNETETH -p udp -s any/0 53 -d $INTERNETADDR 53
-j ACCEPT
ipchains -A output -i $INTERNETETH -p udp -s $INTERNETETH 53 -d any/0 53
-j ACCEPT
```

Lorsqu'un client distant envoie une requête au serveur DNS présent sur la machine firewall, il se peut que les données soient tronquées et que le protocole TCP soit exploité. De même, si un serveur DNS secondaire est présent, il faut que le transfert de zone soit possible. Pour autoriser cela, il faut que le firewall laisse passer les requêtes TCP :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 53 1024:65535
-d $INTERNETADDR 53 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 53
-d any/0 1024:65535 -j ACCEPT
```

7.6.2. Filtrer les identifications.

Le service d'identification des utilisateurs identd est utilisé lors de l'envoi de courrier ou lors de la publication d'un article Usenet. Quelques sites FTP sont également configurés pour demander une identification. Pour contacter un serveur d'identification, il faut employer le port 113.

En tant que client, il n'y a aucune raison d'interdire les requêtes d'identification sortantes :

```
ipchains -A output $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535 -d any/0 113
-j ACCEPT
ipchains -A input $INTERNETETH -p tcp ! -y -s any/0 113
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Si un serveur identd est présent sur la machine firewall, il faut accepter les requêtes entrantes sur le port 113.

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 113 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 113
-d any/0 1024:65535 -j ACCEPT
```

7.7. Activer des services TCP courants.

L'administrateur désire probablement activer un ou plusieurs services courants basés sur le protocole TCP. Voici les services les plus couramment utilisés de nos jours sur Internet :

- le courrier électronique ;
- les groupes de discussions ;
- telnet ;
- ssh ;
- ftp ;
- http ;
- finger ;
- whois ;
- gopher ;
- WAIS.

7.7.1. Le courrier électronique.

La majorité des utilisateurs d'Internet souhaitent exploiter le service du courrier électronique. La configuration à appliquer au niveau du firewall dépend du fournisseur d'accès à Internet. Le courrier électronique est envoyé au travers

du réseau au moyen du protocole SMTP assigné au port 25. La réception locale de messages s'effectue communément à l'aide d'un des trois protocoles suivants :

- SMTP (port 25) ;
- POP (port 110) ;
- IMAP (port 143).

Lorsque du courrier sortant est relayé, au travers d'un serveur SMTP externe, le programme de courrier envoie tous vers le serveur du fournisseur d'accès à Internet. Les deux règles suivantes autorisent le firewall à relayer le courrier au travers de la passerelle SMTP du fournisseur d'accès à Internet dont l'adresse est, par exemple, 212.68.193.7 :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s $212.68.193.7 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Si la machine locale est équipée d'un serveur de courrier électronique, les clients peuvent être configurés de manière à envoyer le courrier au serveur local plutôt qu'au serveur du fournisseur d'accès à Internet. Dans ce cas, c'est le serveur local qui est chargé de contacter les serveurs destinataires du courrier. Voici deux règles permettant l'envoi du courrier au moyen d'un serveur local :

```
ipchains -A output -i $INTERNETETH -p tcp -s INTERNETADDR 1024:65535
-d any/0 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

La réception du courrier peut s'opérer de différentes manières :

- recevoir du courrier au moyen du serveur SMTP local ;
- retirer du courrier en utilisant le serveur POP du fournisseur d'accès à Internet ;
- retirer du courrier en utilisant le serveur IMAP du fournisseur d'accès à Internet.

Pour recevoir le courrier directement sur le serveur SMTP local, il faut configurer le firewall pour qu'il accepte les courriers entrants provenant de tous les hôtes d'Internet. Voici les deux règles à appliquer dans ce cas :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 25 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 25
-d any/0 1024:65535 -j ACCEPT
```

Pour retirer le courrier à partir du serveur POP du fournisseur d'accès à Internet, il faut que le firewall accepte de contacter ce service distant sur le port 110. En supposant que l'adresse IP du serveur POP du fournisseur d'accès à Internet soit 212.68.193.8, voici les règles imposant au firewall de laisser passer les paquets :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.8 110 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.8 110
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Pour retirer le courrier à partir du serveur IMAP du fournisseur d'accès à Internet, il faut que le firewall accepte de contacter ce service distant sur le port 143. En supposant que l'adresse IP du serveur POP du fournisseur d'accès à Internet soit 212.68.193.8, voici les règles imposant au firewall de laisser passer les paquets :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.8 143 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.8 143
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Différentes combinaisons d'envoi et de réception du courrier sont possibles :

- expédier le courrier en tant que client SMPT et le recevoir en tant que client POP ;
- expédier le courrier en tant que client SMPT et le recevoir en tant que client IMAP ;
- expédier le courrier en tant que client SMTP et le recevoir en tant que serveur SMTP ;
- expédier le courrier en tant que serveur SMTP et le recevoir en tant que serveur SMTP ;

Sachant que le serveur SMTP porte l'adresse 212.68.193.7 et que le serveur POP porte l'adresse 212.68.193.8, voici la configuration à appliquer dans le cas où le courrier est expédié en tant que client SMTP et reçu en tant que client POP :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.7 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 110 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.7 110
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Sachant que le serveur SMTP porte l'adresse 212.68.193.7 et que le serveur IMAP porte l'adresse 212.68.193.8, voici la configuration à appliquer dans le cas où le courrier est expédié en tant que client SMTP et reçu en tant que client IMAP :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.7 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 143 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.7 143
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Sachant que le serveur SMTP du fournisseur d'accès à Internet porte l'adresse 212.68.193.7, voici la configuration à appliquer dans le cas où le courrier est expédié en tant que client SMTP et reçu en tant que serveur SMTP :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.7 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.7 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535 -d $INTERNETADDR 25
-j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 25
-d any/0 1024:65535 -j ACCEPT
```

Voici la configuration à appliquer pour envoyer et recevoir du courrier en tant que serveur SMTP :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 25 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 25
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535 -d $INTERNETADDR 25
-j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 25
-d any/0 1024:65535 -j ACCEPT
```

Les utilisateurs du courrier électronique exploitent les services POP et IMAP pour retirer des messages. Si les clients font partie d'un réseau local situé derrière le firewall, aucune règle ne doit lui être ajoutée pour autoriser ce service. Par contre, si l'administrateur souhaite que des clients distants puissent retirer du courrier, il faut modifier la configuration du firewall. Il est cependant important de savoir que des attaques fructueuses peuvent être menées à partir de ces services. Dans les exemples qui suivent, tout client distant aura la possibilité de retirer du courrier à partir des services POP et SMTP bien que dans la pratique, il soit souhaitable d'offrir ces services à seulement un nombre limité de clients.

Voici la configuration du firewall pour autoriser des clients distants à employer le serveur POP sur le port 110 pour retirer du courrier :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 110 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 110
-d any/0 1024:65535 -j ACCEPT
```

Dans le même ordre d'idée, voici une configuration du firewall autorisant tous les clients distants à retirer du courrier à partir du serveur IMAP local :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 143 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 143
-d any/0 1024:65535 -j ACCEPT
```

7.7.2. Accéder aux services Usenet.

Le port 119 offre l'accès au service NNTP. Si l'administrateur accepte que les utilisateurs lisent des messages en provenance du serveur de news du fournisseur d'accès à Internet, il faut configurer le firewall de la manière suivante en supposant que le serveur NNTP porte l'adresse 212.68.193.222 :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.222 119 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.222 119
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Si un serveur NNTP existe localement, il faut que le firewall permette à des clients distants de l'exploiter. Voici les deux règles ouvrant le firewall pour cette utilisation :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 119 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 119
-d any/0 1024:65535 -j ACCEPT
```

Les fournisseurs d'accès à Internet disposent localement de leur propre serveur NNTP. Ce serveur est en relation avec un autre serveur NNTP de manière à rendre possible l'échange des messages. Ce serveur distant accepte d'alimenter automatiquement le serveur local et le serveur local peut également alimenter le serveur distant. En supposant que le serveur NNTP du fournisseur d'accès à Internet accepte les relations d'égal à égal avec le serveur local, voici les deux règles à ajouter au firewall :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d 212.68.193.222 119 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s 212.68.193.222 119
-d $INTERNETADDR 104:65535 -j ACCEPT
```

7.7.3. Le service telnet.

Durant plusieurs années, telnet a été le moyen standard de se connecter à Internet. A présent, ce standard est vu de plus en plus comme un moyen de communication non sécurisé. Le serveur telnet est actif sur le port 23.

Voici les règles à ajouter au firewall si l'administrateur accepte que des utilisateurs locaux accèdent à des serveurs telnet distants :

```
ipchains -A ouput -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 23 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 23
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Si le service telnet doit être offert à l'extérieur, il faut ouvrir l'accès au port 23. Dans la pratique, il ne faut pas accepter les demandes de connexion émanant de n'importe où. Il est préférable d'accepter uniquement une demande en provenance d'hôtes de confiance.

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 23 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 23
-d any/0 1024:65535 -j ACCEPT
```

7.7.4. Le service ssh.

Le shell sécurisé SSH est de loin préférable à l'emploi de telnet car les communications entre les machines sont cryptées. Par défaut, les connexions sont initiées entre le port non privilégié du client et le port 22 du serveur. Durant l'initialisation de la connexion, le port du client est réassigné à un port privilégié compris entre 1023 et 513. Le numéro de port attribué au client est le premier port libre en partant de 1023 et en décroissant vers le port 513.

```
ipchains -A ouput -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 22 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 22
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A ouput -i $INTERNETETH -p tcp -s $INTERNETADDR 513:1023
-d any/0 22 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 22
-d $INTERNETADDR 513:1023 -j ACCEPT
```

Voici les règles à appliquer pour permettre aux clients distants d'accéder au serveur SSH local :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 22 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 22
-d any/0 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp -s any/0 513:1023
-d $INTERNETADDR 22 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 22
-d any/0 513:1023 -j ACCEPT
```

7.7.5. Le service ftp.

Le protocole FTP demeure le moyen le plus courant de transférer des fichiers entre deux machines en réseau. Ce protocole utilise deux ports privilégiés, l'un pour envoyer les commandes et l'autre pour envoyer les données. Le port 21 est employé lors de l'établissement initial de la connexion au serveur et par la suite pour transmettre les commandes à l'utilisateur. Le port 20 est utilisé pour l'établissement d'un canal servant au transport des données.

Le protocole FTP possède deux modes d'échange des données entre le client et le serveur :

- le mode normal (utilisation du port 20) ;
- le mode passif (utilisation d'un port non privilégié).

Le mode normal est utilisé par défaut par les clients FTP lors de la connexion à un site distant. Le mode passif est un mécanisme plus récent qui est utilisé par défaut lors de l'établissement d'une connexion au moyen d'un navigateur Internet.

Afin d'autoriser les connexions à des serveurs FTP distants, il faut non seulement permettre l'utilisation du port 21 mais également du port 20. Si en plus de cela, l'administrateur autorise le mode passif, il faut accepter les connexions entre des ports non privilégiés. Dans le cas d'une connexion en mode normal, c'est le serveur distant qui rappelle le client. Par contre, en mode passif, c'est le client qui initie les connexions à la fois pour les données et les commandes. Voici les règles ouvrant le firewall aussi bien en mode normal qu'en mode passif :

```

ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 21 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 21
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp -s any/0 20
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 1024:65535
-d any/0 20 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 1024:65535
-d $INTERNETADDR 1024:65535 -j ACCEPT

```

La décision d'offrir les services FTP au reste du monde est toujours controversée. Bien que les sites FTP abondent sur Internet, il existe plusieurs exploits de pirates contre la sécurité de ce service. Si l'objectif consiste à fournir un accès uniquement en lecture à quelques fichiers, il est préférable d'en envisager la publication par l'entremise d'un serveur Web. Si le serveur FTP doit permettre la mémorisation de fichiers clients, il est impératif de le refuser sous l'identité de l'utilisateur anonyme. Voici les règles du firewall ouvrant un site FTP à la fois en mode normal et en mode passif :

```

ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 21 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 21
-d any/0 1024:65535 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 20
-d any/0 1024:65535 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 1024:65535
-d $INTERNETADDR 20 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 1024:65535 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 1024:65535
-d any/0 1024:65535 -j ACCEPT

```

7.7.6. Les services Web.

Les services Web sont disponibles sur le port 80. Voici les règles permettant un accès à des sites distants :

```

ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 80 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 80
-d $INTERNETADDR 1024:65535 -j ACCEPT

```

Si un serveur Web local est disponible, il faut en permettre l'accès par l'intermédiaire des règles suivantes :

```

ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 80 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 80
-d any/0 1024:65535 -j ACCEPT

```

Les sites commerciaux utilisent une connexion Web sécurisée afin de permettre aux clients d'introduire, en toute sécurité, les données relatives à leur carte bancaire. Ce service est offert par l'intermédiaire du port 443. Voici deux règles autorisant les connexions à des sites Web sécurisés :

```

ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 443 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 443
-d $INTERNETADDR 1024:65535 -j ACCEPT

```

Dans le cas où le site local héberge une forme de commerce électronique, il faut permettre aux clients distants d'accéder au service sécurisé du serveur Web. Voici deux règles autorisant cet accès :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 443 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 443
-d any/0 1024:65535 -j ACCEPT
```

L'accès public à un serveur mandataire est très courant. En tant que client, le navigateur doit être configuré afin d'exploiter un proxy serveur distant. A la condition que ce service soit offert sur le port 8080, voici deux règles autorisant l'accès aux serveurs mandataires distants :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 8080 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 8080
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Bien qu'il soit possible d'offrir un service mandataire à la communauté Internet, cela ne sera pas présenté ici pour des raisons d'éthiques.

7.7.7. Le service finger.

Le service finger fournit des renseignements sur le compte de l'utilisateur, incluant par exemple le nom de login, le nom réel, etc... En raison du droit à la vie privée, ce genre de service n'est généralement pas offert. Néanmoins voici la méthode à utiliser pour permettre l'accès à des serveurs finger distants sachant qu'ils utilisent le port 79 :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 79 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 79
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

Bien que cela soit à présent déconseillé, voici les règles permettant d'ouvrir localement un serveur finger :

```
ipchains -A input -i $INTERNETETH -p tcp -s any/0 1024:65535
-d $INTERNETADDR 79 -j ACCEPT
ipchains -A output -i $INTERNETETH -p tcp ! -y -s $INTERNETADDR 79
-d any/0 1024:65535 -j ACCEPT
```

7.7.8. Le service whois.

Comme le montre l'exemple suivant, le service whois permet d'obtenir des informations sur un domaine.

```
[root@P100 /root]# whois mit.edu
[whois.crsnic.net]
```

```
Whois Server Version 1.3
```

```
Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.
```

```
Domain Name: MIT.EDU
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: http://www.networksolutions.com
Name Server: BITSY.MIT.EDU
Name Server: STRAWB.MIT.EDU
Name Server: W20NS.MIT.EDU
Updated Date: 05-nov-2001
```

```
>>> Last update of whois database: Thu, 20 Dec 2001 05:17:22 EST <<<
```

```
The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

Ce protocole utilise le port 43 afin de retirer les données souhaitées. Voici un exemple de configuration du firewall autorisant la consultation des informations offertes par les serveurs whois distants :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 43 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 43
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

7.7.9. Le service gopher.

Le service d'information gopher est encore disponible bien que son emploi ait été totalement remplacé par les moteurs de recherche du Web. Voici les règles donnant un accès aux services gopher sur le port 70 :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 70 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 70
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

7.7.10. Le service WAIS.

Les serveurs WAIS sont à présent connus comme des moteurs de recherche. Les navigateurs Web sont des façades graphiques type à ce genre de serveur. Sachant que le port 210 est utilisé par les serveurs WAIS, voici les deux règles de configuration du firewall donnant accès distant à ces services :

```
ipchains -A output -i $INTERNETETH -p tcp -s $INTERNETADDR 1024:65535
-d any/0 210 -j ACCEPT
ipchains -A input -i $INTERNETETH -p tcp ! -y -s any/0 210
-d $INTERNETADDR 1024:65535 -j ACCEPT
```

7.8. Habiliter les services UDP courants.

Le protocole UDP est sans état et il est par définition moins sécurisé que le protocole TCP basé sur la notion de connexion. De ce fait, plusieurs sites conscients de la sécurité interdisent ou limitent autant que possible, l'accès aux services UDP. Naturellement, les échanges DNS basés sur UDP sont nécessaires, mais le serveur de noms distant peut être explicitement désigné dans les règles du firewall. Voici trois services basés sur UDP :

- traceroute;
- DHCP (Dynamic Host Configuration Protocol) ;
- NTP (Network Time Protocol).

7.8.1. Le programme traceroute.

Le programme traceroute est un service UDP qui demande aux systèmes intermédiaires de générer des messages ICMP Time Exceeded afin de rassembler des informations sur le nombre de sauts effectués. Ce programme demande aussi au système cible de retourner un message de port non trouvé (Destination Unreachable), indiquant au client l'extrémité de la route.

Par défaut, le firewall construit dans ce chapitre, bloque les paquets traceroute entrants. Donc, les réponses ICMP aux requêtes traceroute ne seront pas envoyées.

Le service traceroute peut être configuré afin de prendre n'importe quel port ou série de ports. Il est donc malaisé de bloquer tous les paquets traceroute entrants en spécifiant une série de ports. Toutefois, ce service emploie traditionnellement la série de ports source compris entre 32769 et 65535, et la série de ports destination compris entre 33434 et 35523.

Si l'un des utilisateurs du système à l'intention d'employer le programme traceroute, il faut habilitier les ports client.

```
ipchains -A output -i $INTERNETETH -p udp -s $INTERNETADDR 32769:65535
-d any/0 33434:35523 -j ACCEPT
```

Les requêtes UDP étant de par leur nature peu sécurisées, l'acceptation des requêtes traceroute doit être limitée à quelques hôtes de confiance.

```
ipchains -A input -i $INTERNETETH -p udp -s any/0 32769:65535 -d $INTERNETADDR
33434:35523 -j ACCEPT
```

7.8.2. Accéder à un serveur DHCP.

Bien qu'il soit possible de configurer un firewall lorsque des adresses IP sont attribuées dynamiquement, la matière sortant du cadre de ce cours, ne sera pas abordée. En effet, il n'existe pas de véritable site fonctionnant au moyen d'une adresse IP qui change perpétuellement.

7.8.3. Accéder à un serveur Network Time.

Le service NTP permet l'accès à un ou plusieurs fournisseurs Internet d'horloge. Ce genre d'outil est très pratique pour maintenir l'horloge interne du système constamment à jour. Seul la partie cliente sera envisagée dans ce cours car peu de sites sont connectés à une horloge atomique. Sachant qu'un serveur NTP utilise le port 123, voici les deux règles du firewall autorisant l'exploitation d'un client NTP :

```
ipchains -A output -i $INTERNETETH -p udp -s $INTERNETADDR 1024:65535 -d
195.13.23.5 123 -j ACCEPT
ipchains -A input $ INTERNETETH -p udp -s 195.13.23.5 123 -d $INTERNETADDR
1024:65535 -j ACCEPT
```

Dans cet exemple, l'adresse 195.13.23.5 est celle du serveur NTP de la Belgique.

7.9. Journaliser des paquets entrants refusés.

Tous les paquets se conformant à une règle peuvent être placés dans un journal grâce à l'option `-l` de la commande `ipchains`. Quelques-unes des règles qui ont été présentées avaient cette option de journalisation activée. Le spoofing des adresses IP en est un exemple.

Des règles peuvent être définies dans le but de créer une entrée dans un journal. Les paquets intéressants sont ceux qui semblent suspects, ceux, par exemple indiquant une sorte de balayage ou de test. Tous les paquets étant refusés par défaut, il faut créer une règle spécifique pour susciter une mémorisation.

L'identité des paquets à mettre dans un journal est un choix individuel. Des personnes veulent enregistrer tous les paquets refusés. Pour d'autres, l'enregistrement systématique pourrait rapidement créer un engorgement du fichier journal. Certaines aussi se sentent à l'aise du fait que les paquets aient été refusés et elles ne se préoccupent pas de leur nature. D'autres encore sont très intéressées par les balayages de ports ou par un certain type de paquets bien précis.

Pour placer tous les paquets rejetés dans le journal, il suffit d'ajouter la règle suivante :

```
ipchains -A input -i $INTERNETETH -j DENY -l
```

Pour certains, cette règle produirait trop d'entrées au journal, ou trop d'entrées sans intérêt. Généralement les administrateurs préfèrent mémoriser uniquement les accès aux ports privilégiés sur lesquels aucun service n'est offert.

7.10. Refuser l'accès aux sites problématiques.

Si un site particulier à une mauvaise réputation ou a de mauvaises habitudes, une règle permet de lui refuser tout accès. En supposant que le site 212.68.194.200 n'ait pas un comportement correct, la règle suivante lui interdit tout accès :

```
ipchains -I input -i $INTERNETETH -s 212.68.194.200/32 -j DENY
```

Cette règle étant la première, tous les paquets provenant de cette adresse source seront refusés, peu importe le type de protocole du message, de son port source ou de son port destination.

A ce moment, les règles du firewall sont définies, la machine peut être connectée à Internet.

7.11. Activer l'accès au réseau local.

Si la machine firewall est située entre Internet et un réseau local, les machines du réseau local n'ont ni accès à l'interface réseau interne de la machine firewall, ni à Internet. Afin de supporter un réseau local derrière un firewall, quelques règles supplémentaires sont nécessaires pour permettre l'accès à l'interface réseau interne de la machine firewall et pour faire circuler le trafic vers Internet. Lorsque la machine firewall fonctionne de cette manière, avec au moins deux interfaces réseau, ce système est appelé firewall bastion ou zone démilitarisée (DMZ).

Voici une paire de règles qui permettent d'ouvrir la communication entre la machine firewall et le réseau local :

```
ipchains -A input -i $INTRANETETH -s $INTRANETNETWORK -j ACCEPT
ipchains -A output -i $INTRANETETH -d $INTRANETNETWORK -j ACCEPT
```

Ces règles permettent l'accès de toutes les machines du réseau local à la machine firewall mais pas à Internet au travers du firewall. Par définition, une machine firewall ne route pas le trafic dynamiquement ou automatiquement. Des règles supplémentaires sont donc nécessaires pour router le trafic local plus loin.

La communication entre le réseau local et Internet doit être à la fois transférée et masquée. Le transfert IP est un service du noyau permettant à une machine Linux d'agir comme un routeur entre deux réseaux, transférant ainsi le trafic d'un réseau à l'autre.

Voici une commande à inclure dans les scripts de démarrage du système permettant d'effectuer le routage :

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

Une autre manière de résoudre le problème consiste à introduire une règle dans le firewall de façon à autoriser le transfert et le masquage qui permet à la machine firewall d'agir comme un serveur mandataire. Voici cette règle :

```
ipchains -A forward -i $INTERNETETH -s $INTRANETNETWORK -j MASQ
```

7.12. Installer le firewall.

L'installation du firewall nécessite l'exécution du script qui a été créé dans ce chapitre. Pour que les modifications soient permanentes, il suffit de lancer la commande suivante :

```
/etc/rc.d/init.d/ipchains save
```

Bibliographie.

- [1] Cricket Liu & Paul Albitz ; *DNS and BIND* ; O'Reilly ; Sebastopol ; 1998.
- [2] Craig Hunt ; *TCP/IP Network Administration* ; O'Reilly ; Sebastopol ; 1997.
- [3] Bryan Costales & Eric Allman ; *Sendmail* ; O'Reilly ; Sebastopol ; 1997.
- [4] D.Brent Chapman & Elizabeth D. Zwicky ; *Building Internet Firewalls* ; O'Reilly ; Sebastopol ; 1995.
- [5] Robert L. Ziegler ; *Linux firewalls* ; Macmillan Technical Publishing ; Indianapolis ; 2000
- [6] Douglas E. Comer ; *Internetworking with TCP/IP volume I principles, protocols, and architecture* ; Prentice-Hall International ; London ; 1991.
- [7] Douglas E. Comer & David L. Stevens ; *Internetworking with TCP/IP volume II Design, Implementation, and internals* ; Prentice-Hall International ; London ; 1991.
- [8] Hal Stern, Mike Eisler & Ricardo Labiaga ; *Managing NFS and NIS* ; O'Reilly ; Sebastopol ; 2001
- [9] Robert Eckstein, David Collier-Brown, Peter Kelly ; *Using samba* ; O'Reilly ; Sebastopol ; 1999
- [10] Ben Laurie & Peter Laurie ; *Apache : The definitive guide* ; O'Reilly ; Sebastopol ; 1999
- [11] David HM Spector ; *Building Linux clusters* ; O'Reilly ; Sebastopol ; 2000
- [12] Gisèle Cizault ; *Ipv6 théorie et pratique* ; O'Reilly ; Paris ; 1999
- [13] Aron Hsiao ; *Sams teach yourself Linux security in 24 hours* ; Addison Wesley Longman ; Massachussetts ; 2001
- [14] Kamran Husain & Timothy Parker ; *Linux Unleashed* ; Computer Publishing ; Indianapolis ; 1996
- [15] Anne H. Carasik ; *Linux system administration* ; IDG Book Worldwide ; London ; 1999

