

Administration Système et Réseau UNIX

Partie 2

R.488, mars 2013

Présentation du cours

Chapitre 1 – L'environnement réseau

Chapitre 2 – Surveillance et audits du système

Chapitre 3 – Les volumes logiques

Chapitre 4 – Swap et autres systèmes de fichiers

Chapitre 5 – Services réseau

Chapitre 6 – LDAP l'automonteur

Chapitre 7 – Samba

L'environnement réseau

Protocoles utilisés

La plupart des systèmes d'exploitation utilisent actuellement le protocole TCP/IP basé sur des réseaux de type Ethernet (du moins pour les réseaux locaux). Nous allons nous limiter à ce protocole dans ce cours. Un protocole est un ensemble de règles gérant l'échange de données entre deux entités. Les protocoles interviennent à plusieurs niveaux dans une communication.

Protocoles de niveau réseau

- IP (Internet Protocole). Protocole permettant d'émettre et de recevoir des messages au niveau de l'adresse Internet. IPv4 et IPv6 sont supportés par Ubuntu Linux.
- ARP, RARP (Adresse Resolution Protocol, Reverser ARP). Protocoles permettant de convertir des adresses Internet en adresse ethernet (adresses physiques) ou inversement.
- ICMP (Internet Control Message Protocol). Protocole utilisé pour les messages d'erreurs et de diagnostics.

Protocoles de niveau transport

- TCP (Transmission Control Protocol) est un protocole orienté connexion qui fournit un service fiable et à double sens sur lequel beaucoup d'applications s'appuient.
- UDP (User Datagram Protocol) fournit un service de transfert moins fiable, en mode déconnecté.

Protocoles de niveau application (quelques exemples)

- DHCP (Dynamic Host Configuration Protocol) automatise la configuration réseau dans un réseau local.
- DNS (Domain Name System) est une base de données distribuée utilisée pour la correspondance des noms de machines avec leurs adresses Internet (et inversement).
- NFS (Network File System) est une application client/serveur qui permet de partager des systèmes de fichiers disques.
- RPC (Remote Procedure Call) est un protocole permettant d'exécuter des fonctions sur un serveur distant et d'en récupérer le résultat.
- FTP, TFTP (File Transfert Protocol, Trivial FTP) sont des protocoles de transfert de fichiers.
- HTTP (HyperText Transfert Protocol) est utilisé pour le transfert de données adressées par des URL (Uniform Ressource Locator).
- SMTP (Simple Mail Transfert Protocol) fournit le transfert des courriers électroniques.

Il en existe bien sur beaucoup plus, nous en verrons d'autres dans ce cours.

Les cartes réseau

Sous Linux, les périphériques de type cartes réseau sont accessibles par l'intermédiaire d'un fichier de périphérique nommé `/dev/ethX` où X est le numéro de la carte réseau, attribué par le système en fonction de l'ordre de détection de la carte. Par exemple, `/dev/eth0` représente la première carte réseau détectée, `/dev/eth1` la deuxième carte réseau détectée, etc...

Il existe plusieurs façons de voir quels sont les périphériques de type cartes réseau détectés lors du démarrage du système d'exploitation.

La première, la moins pratique, est d'utiliser la commande `dmesg`. Par exemple:

```
# dmesg | grep eth
[  2.463625] e1000 0000:00:03.0: eth0: (PCI:33MHz:32-bit) 08:00:27:34:0f:8a
[  2.463635] e1000 0000:00:03.0: eth0: Intel(R) PRO/1000 Network Connection
[ 10.042488] udevd[279]: renamed network interface eth0 to eth1
[12222.874976] ADDRCONF(NETDEV_UP): eth1: link is not ready
[12222.877231] e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[12222.877803] ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
[32701.791883] e1000: eth1 NIC Link is Down
[32706.781993] e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
```

Ici on voit une carte réseaux de type Intel PRO/1000 nommé par le système `/dev/eth1`.

La deuxième méthode utilise la commande de listage des périphériques PCI (ne fonctionne donc qu'avec les cartes réseau PCI):

```
# lspci | grep Ethernet
00:08.0 Ethernet controller: 3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 74)
00:09.0 Ethernet controller: 3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 74)

lspci -v / -k → driver is use + modprobe
ls /sys/class/net (sysfs)
```

Enfin, la méthode la plus pratique, et qui fonctionne sur Linux mais aussi sur un grand nombre de système UNIX, est l'utilisation de la commande `ifconfig`:

```
# ifconfig -a
eth0      Lien encap:Ethernet  HWaddr 00:04:76:A3:B2:BE
          inet adr:172.16.3.14  Bcast:172.16.255.255  Masque:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10165 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9872 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:3769088 (3.5 Mb)  TX bytes:1616388 (1.5 Mb)
          Interruption:10 Adresse de base:0xec00

eth1      Lien encap:Ethernet  HWaddr 00:04:76:E2:A0:1C
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interruption:11 Adresse de base:0xe800

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10589 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10589 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:722033 (705.1 Kb)  TX bytes:722033 (705.1 Kb)
```

On retrouve dans cet exemple les deux cartes réseau 3COM, avec leur nom de périphérique associé (`/dev/eth0` et `/dev/eth1`), leur configuration actuelle (dans le cas de `eth0`) ainsi que leurs adresses physiques (champ `Hwaddr`).

On note aussi la présence d'une carte virtuelle `/dev/lo0`, qui est la carte de bouclage locale (`localhost`).

Configuration du réseau

Les paramètres réseau les plus souvent modifiés sont:

- Les adresses IP et masques de sous réseau des cartes
- L'adresse de la passerelle par défaut
- Le nom d'hôte de la machine
- La résolution des noms de machines

Configuration manuelle

Tout changement à la configuration fait manuellement est pris en compte immédiatement par le système d'exploitation, sans nécessiter de redémarrage.

Cartes réseau

La commande `ifconfig` permet de stopper ou de démarrer le fonctionnement d'une carte réseau, de voir et de changer la configuration d'une carte.

Exemples :

Voir la configuration courante d'une carte :

```
# ifconfig eth0
eth0      Lien encap:Ethernet  HWaddr 00:04:76:A3:B2:BE
          inet adr:172.16.3.14  Bcast:172.16.255.255  Masque:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:311690 errors:0 dropped:0 overruns:0 frame:0
          TX packets:207906 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:89273137 (85.1 Mb)  TX bytes:37021047 (35.3 Mb)
          Interruption:10 Adresse de base:0xec00
```

Stopper la carte :

```
# ifconfig eth0 down
```

Changer la configuration :

```
# ifconfig eth1 down
# ifconfig eth1 172.16.0.15 netmask 255.255.255.0
# ifconfig eth1 up
```

Passerelle par défaut

La configuration de l'adresse de la passerelle par défaut se fait avec la commande `route`. En effet, définir la passerelle par défaut revient à définir la route par défaut pour tous les paquets non destinés au réseau local.

Exemple :

```
# route add default gw 172.16.0.1
```

Pour voir la configuration actuelle, il faut visualiser les routes avec la commande `netstat` :

```
# netstat -nr
Table de routage IP du noyau
Destination      Passerelle      Genmask         Indic   MSS  Fenêtre  irtt  Iface
172.16.0.0       0.0.0.0         255.255.0.0     U        0  0         0  eth0
169.254.0.0     0.0.0.0         255.255.0.0     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         172.16.0.1     0.0.0.0         UG       0  0         0  eth0
```

Nom d'hôte

Pour manipuler le nom d'hôte, on utilise la commande `hostname`.

Exemple, visualisation du nom d'hôte courant:

```
# hostname
s3p14
```

Exemple, modification du nom d'hôte courant:

```
# hostname monnom
# hostname
monnom
```

Exemple, vérification de la résolution du nom de machine courant:

```
# hostname -s
s3p14.intra.e-cml.org
# hostname monnom
# hostname -s
hostname: Hôte inconnu
```

Le nom d'hôte doit être valide (il doit exister dans le fichier `/etc/hosts` ou dans le DNS), sinon vous pouvez avoir des problèmes de communication réseau ou d'utilisation des applications.

Fichiers de configuration

Les fichiers de configuration sont utilisés par le système d'exploitation au démarrage de celui-ci. Toute modification à ces fichiers ne sera donc prise en compte que lors du prochain démarrage du système.

Cartes réseau

Les paramètres de configuration du système se trouvent (dans le cas de Ubuntu Linux) dans le répertoire `/etc/network`.

En ce qui concerne la configuration des cartes réseau, il y a un fichier de configuration, qui porte le nom `/etc/network/interfaces`.

Toute modification de ce fichier nécessite de redémarrer le réseau :

```
# /etc/init.d/networking restart
```

Exemple, configuration avec deux cartes réseau, une avec une adresse statique et une avec une adresse dynamique :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0 eth1
iface eth0 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    gateway 192.168.0.254

# The secondary network interface
iface eth1 inet dhcp
```

Nom d'hôte

Le nom d'hôte (et le nom de domaine DNS si besoin) par défaut se trouve dans le fichier `/etc/hostname`

Exemple de fichier `/etc/hostname`

```
mamachine.monreseau.example.com
```


Résolution des noms de machines

La résolution des noms de machines peut utiliser deux bases de données pour convertir des adresses IP en nom de machine ou inversement: la base locale et la base distribuée DNS.

La base locale est contenue dans le fichier `/etc/hosts`.

Exemple de fichier `/etc/hosts` :

```
127.0.0.1          localhost.localdomain localhost
172.16.0.1        servnet mail cache servnet.e-cml.org
172.16.0.2        servux servux.e-cml.org
```

L'inconvénient de cette base est qu'elle doit être maintenue manuellement sur chaque machine, et qu'elle peut être perdue en cas de mise à jour ou de réinstallation du système.

Le protocole DNS permet d'utiliser une base de données distribuée (nous verrons cela en détails dans la suite de ce cours). Il est configuré par l'intermédiaire du fichier `/etc/resolv.conf`.

Exemple de fichier `/etc/resolv.conf` :

```
search intra.e-cml.org
nameserver 172.16.0.2
nameserver 172.16.0.22
```

La configuration du DNS peut être testée à l'aide des outils `nslookup` ou `dig`.

Exemples :

```
# nslookup www.e-cml.org
Server:          172.16.0.2
Address:         172.16.0.2#53

Non-authoritative answer:
Name:   www.e-cml.org
Address: 194.167.168.2

# dig www.e-cml.org
...
;; ANSWER SECTION:
www.e-cml.org.          710      IN      A      194.167.168.2
...
```

Si le réseau est configuré à l'aide de DHCP, le fichier `/etc/resolv.conf` est modifié automatiquement par DHCP.

Surveillance du réseau

Ping

La commande `ping` permet de vérifier la présence active d'un autre système et affiche, si celui-ci répond, le temps nécessaire à la réponse.

Exemple :

```
# ping -c 1 servux
PING servux (172.16.0.2) 56(84) bytes of data.
64 bytes from servux (172.16.0.2): icmp_seq=1 ttl=64 time=0.162 ms

--- servux ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.162/0.162/0.162/0.000 ms
```

Cette commande possède aussi un mode "audible" qui permet de la lancer sur une machine et de travailler sur une autre machine tout en surveillant le fonctionnement du réseau:

```
# ping -a s2p01
...
```

nmap

L'outil nmap permet de trouver les services actifs ou filtrés (par un firewall par exemple) d'une machine. Il se connecte aux différents ports pour voir lesquels sont accessibles. Exemple :

```
# nmap servbak
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on servbak.intra.e-cml.org (172.16.0.5):
(The 1598 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
111/tcp   open       sunrpc
873/tcp   open       rsync

Nmap run completed -- 1 IP address (1 host up) scanned in 10 seconds
```

Il existe aussi une interface graphique pour cet outil, Zenmap.

tcpdump

La commande `tcpdump` est très utile pour déterminer les informations transitant entre deux systèmes. On peut visualiser toutes les communications, ou filtrer par nom de machine, par protocole, ...

Exemples:

Visualisation de tout le trafic à destination (ou provenant) de la machine `servux`:

```
# tcpdump host servux
tcpdump: listening on eth0
15:05:37.907418 arp who-has servux tell ismap2.intra.e-cml.org
15:05:37.909431 s3p14.intra.e-cml.org.32778 > servux.domain: 18346+ PTR?
232.0.16.172.in-addr.arpa. (43) (DF)
15:05:37.909696 servux.domain > s3p14.intra.e-cml.org.32778: 18346* 1/1/1 (116)
15:05:37.910308 s3p14.intra.e-cml.org.32778 > servux.domain: 18347+ PTR?
14.3.16.172.in-addr.arpa. (42) (DF)
15:05:37.910496 servux.domain > s3p14.intra.e-cml.org.32778: 18347* 1/1/1 PTR[|domain]
15:05:42.903700 arp who-has servux tell s3p14.intra.e-cml.org
15:05:42.903862 arp reply servux is-at 0:50:4:56:c5:c7
15:05:45.930481 s3p14.intra.e-cml.org > servux: icmp: echo request (DF)
15:05:45.930658 servux > s3p14.intra.e-cml.org: icmp: echo reply (DF)
```

Visualisation du trafic des machines Windows environnantes:

```
# tcpdump port netbios-ns or port netbios-ssn or port netbios-dgm
tcpdump: listening on eth0
15:09:29.448278 s3p13.intra.e-cml.org.netbios-dgm > 172.16.255.255.netbios-dgm: NBT UDP
PACKET(138)
15:09:29.609338 s3p13.intra.e-cml.org.netbios-dgm > 172.16.255.255.netbios-dgm: NBT UDP
PACKET(138)
15:09:39.422784 s3p14.intra.e-cml.org.53100 > servux.netbios-ssn: S
2835927113:2835927113(0) win 5840 <mss 1460,sackOK,timestamp 10173763 0,nop,wscale 0>
(DF)
15:09:39.422930 servux.netbios-ssn > s3p14.intra.e-cml.org.53100: S
1601797224:1601797224(0) ack 2835927114 win 57344 <mss 1460> (DF)
15:09:39.422967 s3p14.intra.e-cml.org.53100 > servux.netbios-ssn: . ack 1 win 5840 (DF)
15:09:39.675355 s3p14.intra.e-cml.org.53100 > servux.netbios-ssn: P 1:73(72) ack 1 win
5840 NBT Packet (DF)
15:09:39.675961 servux.netbios-ssn > s3p14.intra.e-cml.org.53100: P 1:5(4) ack 73 win
58400 NBT Packet (DF)
15:09:39.676008 s3p14.intra.e-cml.org.53100 > servux.netbios-ssn: . ack 5 win 5840 (DF)
15:09:39.676214 s3p14.intra.e-cml.org.53100 > servux.netbios-ssn: P 73:241(168) ack 5
win 5840 NBT Packet (DF)
15:09:39.676613 servux.netbios-ssn > s3p14.intra.e-cml.org.53100: P 5:95(90) ack 241 win
58400 NBT Packet (DF)
```

Il est possible de faire des règles complexes, comme par exemple:

```
# tcpdump src servux and not port nfs and not host 172.16.255.255
```

Wireshark

Wireshark est un outil permettant d'analyser les paquets récupérés avec `tcpdump` ou tout simplement de remplacer `tcpdump`. Il est utilisable aussi bien en mode commande qu'en mode graphique.

Exemples de capture d'une session FTP:

```
# tshark host servux and port ftp
Capturing on eth0
...
 2.747894 172.16.3.14 -> 172.16.0.2  FTP Request: USER ttoto
 2.792006 172.16.0.2 -> 172.16.3.14  FTP Response: 331 Password required for ttoto.
 2.842161 172.16.3.14 -> 172.16.0.2  TCP 53190 > ftp [ACK] Seq=3274266922
Ack=3279703452 Win=5840 Len=0
 4.257198 172.16.3.14 -> 172.16.0.2  FTP Request: PASS azerty
```

```
4.267257 172.16.0.2 -> 172.16.3.14 FTP Response: 530 Login incorrect.
4.408544 172.16.3.14 -> 172.16.0.2 TCP 53190 > ftp [ACK] Seq=3274266935
Ack=3279703474 Win=5840 Len=0
4.410315 172.16.3.14 -> 172.16.0.2 FTP Request: SYST
4.410480 172.16.0.2 -> 172.16.3.14 FTP Response: 530 Please login with USER and
PASS.
....
```

netstat

Enfin, la commande `netstat` permet de visualiser les informations concernant le fonctionnement du réseau. Cette commande est surtout utile pour résoudre les problèmes. Nous l'avons déjà utilisée pour visualiser les routes, nous pouvons l'utiliser aussi pour:

Voir un résumé du trafic par carte (utile pour voir les collisions, erreurs,) :

```
# netstat -i
Table d'interfaces noyau
Iface      MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500  0 1558248      0      0      0  614852      0      0      0 0 BMRU
lo         16436  0 1143072      0      0      0 1143072      0      0      0 0 LRU
```

Voir les connexions TCP/IP en cours :

```
# netstat --inet
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale      Adresse distante      Etat
tcp      0      0 s3p14.intra.e-cml:36794 fo01.intra.e-cml.or:ssh ESTABLISHED
tcp      0      0 s3p14.intra.e-cml.o:x11 fo01.intra.e-cml.:50768 ESTABLISHED
tcp      0      0 s3p14.intra.e-cml:37250 servsmb.intra.e-cml:ssh ESTABLISHED
tcp      0      0 s3p14.intra.e-cml:48726 servux:ssh            ESTABLISHED
```

Voir un résumé de toutes les communications :

```
# netstat -s | more
Ip:
 1804368 total packets received
  0 forwarded
  0 incoming packets discarded
1765991 incoming packets delivered
1761328 requests sent out
 43971 reassemblies required
14657 packets reassembled ok
...
```

Surveillance et audits

Syslog

Fonctionnement de syslog

Le service `syslog` gère les messages générés par les processus systèmes (qui, par définition, n'interagissent pas avec la console). Ces processus envoient leurs messages au démon `syslogd` qui se charge de les transmettre vers une destination au choix de l'administrateur :

- Dans un fichier journal (ou un périphérique),
- Sur la ou les consoles d'un ou de plusieurs utilisateurs,
- Sur la console système,
- Au démon `syslogd` d'une autre machine.

Les messages sont transmis dans un format standard (tous les `syslog` de tous les systèmes de type UNIX sont compatibles) dont les champs (séparés par des espaces) ont la forme suivante :

```
date et heure nom d'hôte nom du processus[PID]: message
```

Le nom d'hôte correspond à l'hôte sur lequel le processus qui a émis le message fonctionne (et non à l'hôte sur lequel fonctionne `syslogd`), et la date et l'heure correspondent à la date d'émission du message (et non la date de réception par `syslogd`).

Exemple de message :

```
# tail -n 1 /var/log/ftpllog  
Oct 7 15:17:00 servux ftpd[23824]: FTP LOGIN FAILED FROM s3p14
```

Configuration de syslog

On peut configurer la destination des messages transmis par `syslogd` à l'aide du fichier `/etc/syslog.conf`.

☛ Sous Ubuntu Linux, si le logiciel `rsyslog` est installé le fichier est `/etc/rsyslog.d/50-default.conf`

Une entrée dans le fichier de configuration `/etc/syslog.conf` est composée de deux champs séparés par des tabulations. Ces champs sont: *sélecteur* et *action*.

Le champ *sélecteur* permet de créer un filtre sur les messages en fonction de leur *fonction* et de leur *niveau*. Il a la forme suivante: *fonction.niveau*. Plusieurs sélecteurs peuvent être précisés pour une seule destination, il suffit de les séparer par des points virgules.

Le champ *action* permet de préciser la destination des messages correspondants à ce filtre.

Les *fonctions* possibles sont :

- **auth** Messages de sécurité ou d'authentification système.
- **auth_priv** Messages de sécurité ou d'authentification des applications.
- **cron** Messages des démons horaires (`cron` et `at`).
- **ftp** Messages des démons ftp (`ftpd`, `tftpd`, ...).
- **daemon** Messages des démons sans classification particulière.
- **kern** Messages du noyau ou des pilotes de périphériques.
- **local0** à **local7** Messages des applications non systèmes.
- **lpr** Messages du système d'impression/spooler.
- **mail** Messages du système de courrier sortant/entrant.
- **news** Messages du système de news USENET.
- **syslog** Messages internes de `syslogd`.
- **user** Messages utilisateurs génériques.
- **uucp** Messages du système réseau UUCP.
- ***** Messages de toutes provenances.

Les *niveaux* possibles sont (par ordre croissant d'importance) :

- **debug** Messages de mise au point de programme.
- **info** Messages d'informations simples.
- **notice** Conditions qui ne sont pas des conditions d'erreur, mais qui nécessitent un traitement particulier.
- **warning** Messages d'avertissements.
- **err** Messages d'erreurs non critiques.
- **crit** Messages d'erreurs critiques.
- **alert** Conditions d'erreurs graves qui doivent être prises en compte le plus rapidement possible.
- **emerg** Le système est inutilisable ou a planté suite à une erreur logicielle ou à un problème matériel.
- ***** Tous les messages, quelque soit leur importance.
- **none** Tous les messages sauf ceux de la fonction indiquée.

Le champ *action* permet de diriger les messages vers :

- **/fichier** Un fichier (quel que soit son type). Doit être un chemin absolu. Le message est ajouté à la fin du fichier.
- **@hôte** Nom(s) d'hôte(s) vers lequel les messages doivent être transmis. Ces hôtes doivent exécuter syslogd.
- **user1, user2,...** Écrit les messages sur les consoles des utilisateurs listés.
- ***** Écrit les messages sur les consoles de tous les utilisateurs connectés.

Exemples d'entrées de `/etc/syslog.conf` :

```
*.info;mail.none;authpriv.none;cron.none      /var/log/messages
mail.*                                          /var/log/maillog
cron.*                                         /var/log/cron
*.emerg                                        *
```

Toute modification du fichier `/etc/syslog.conf` nécessite une relecture de ce dernier à l'aide du signal `SIGHUP` :

```
# /etc/init.d/syslogd reload
```

Ou, sur Ubuntu Linux :

```
# reload rsyslog
```

Utilisation de `syslog`

Les processus systèmes et les programmes utilisent directement le service `syslog`, et il est possible de l'utiliser dans des scripts ou manuellement (pour tester les modifications faites au fichier `/etc/syslog.conf` par exemple). Pour cela, il faut utiliser la commande `logger`.

Par exemple, la commande suivante :

```
# logger -p "daemon.crit" -t "test" -i "mon message"
```

Envoie le message suivant :

```
oct 7 17:49:40 s3p14 test[8653]: mon message
```

En règle général, `syslogd` est configuré pour envoyer les messages les plus courants dans le fichier `/var/log/messages`. Il est possible de visualiser en temps réel les messages à l'aide de la commande `tail` comme ceci :

```
# tail -f /var/log/messages
```

Utilitaires d'audit

On trouve sur Linux beaucoup d'outils d'audit système, dont certains vous sont déjà familiers, et qui sont utilisés aussi bien par les administrateurs que par les utilisateurs.

who

L'utilitaire `who` peut fournir les informations suivantes:

- Une liste courte du nombre total d'utilisateurs connectés :

```
# who -q
toto toto toto toto
# usager=4
```

- Qui est connecté au système :

```
# who -H
NOM      LIGNE      HEURE      COMMENTAIRE
toto     :0         Oct 10 10:49
toto     pts/0      Oct 10 10:50 (:0.0)
```

- Les caractéristiques de la connexion du terminal courant :

```
# who -m
toto     pts/0      Oct 10 10:50 (:0.0)
# who is god
toto     pts/0      Oct 10 10:50 (:0.0)
```

w

L'utilitaire `w` affiche les utilisateurs connectés et ce qu'ils font :

```
# w -s
11:46:58 up 2:04, 4 users, load average: 0.10, 0.03, 0.15
USER      TTY      FROM          IDLE  WHAT
toto      :0       -             ?    /usr/bin/gnome-session
toto      pts/0    :0.0          0.00s gnome-terminal
toto      pts/1    :0.0          8:58  ssh -l michelon fo01
toto      pts/2    :0.0          1:18  man w
```

last

La commande `last` peut être utilisée pour déterminer toutes les connexions/déconnexions d'un utilisateur :

```
# last ttoto
ttoto          ttyp0      s3p14          Ven 10 oct 11:49  still logged in
ttoto          ttyp0      s2p01          Mer  8 oct 10:02 - 10:02  (00:00)
ttoto          ttyp0      s2p01          Mer  8 oct 09:54 - 09:56  (00:01)
```

Les reboot système peuvent être tracés :

```
# last reboot
reboot        system boot 2.4.20-8      Fri Oct 10 09:43      (02:08)
reboot        system boot 2.4.20-8      Wed Oct  8 11:56      (1+06:42)
reboot        system boot 2.4.20-8      Wed Oct  8 09:25      (02:12)
```

uptime

La commande `uptime` nous renseigne depuis quand le système fonctionne et la charge de celui-ci :

```
# uptime
11:55:48 up 2:13, 4 users, load average: 0.00, 0.00, 0.07
```

ps, pstree, pgrep, top

Voir partie 1, chapitre 8

df, du

Voir partie 1, chapitre 10

netstat

Voir chapitre 1

vmstat

La commande `vmstat` permet de suivre en temps réel l'activité de la mémoire virtuelle, des zones de swap, des processus, des entrées/sorties et des processeurs :

```
# vmstat 1
procs
r  b  w      swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id
2  0  0         0  8340 17272 369460    0    0   11   21  127  424 10   1  89
0  0  0         0  8340 17272 369460    0    0    0    0  212  553 10   0  90
...
```

grep

Les fichiers de logs étant au format texte, la commande `grep` est un des outils les plus utilisés pour l'audit.

Exemple :

```
# grep "su" /var/log/messages
oct 10 10:50:26 s3p14 su(pam_unix)[3589]: session opened for user root by toto(uid=500)
oct 10 12:01:53 s3p14 su(pam_unix)[9938]: authentication failure; logname=toto uid=500
euid=0 tty= ruser=toto rhost= user=root
```

free

La commande `free` affiche la somme totale de mémoire physique et des zones de swap, ainsi que leur utilisation (mémoire libre, partagée, buffers I/O, cache disque, ...):

En kilo octets :

```
# free
              total        used         free       shared    buffers     cached
Mem:           513852      503540         10312           0        10396      381768
-/+ buffers/cache:    111376      402476
Swap:           522072           0         522072
```

En méga octets :

```
# free -m
              total        used         free       shared    buffers     cached
Mem:             501         492           9           0           10         372
-/+ buffers/cache:    109         392
Swap:             509           0         509
```

Remarques

Les distributions ont utilisé différents systèmes en remplacement d'**init** (**sysvinit**) le système de démarrage d'Unix/Linux :

- La distribution **Ubuntu** utilise **Upstart** au lieu de **init** jusqu'à la version 14.10, puis **systemd** à partir de la version **15.04**.
- À partir de sa neuvième version, **Fedora** intègre **Upstart** en lieu et place de **init**. Cependant, **Upstart** est remplacé par **systemd** dans Fedora 15.

Pour connaître le système de démarrage de votre système :

```
$ ps -p1 | grep systemd && echo systemd || echo upstart
```

Remarque : **systemd** introduit la notion d'unité. Une unité représente un fichier de configuration. Une unité peut être un service (***.service**), un target (***.target**), un montage (***.mount**), un socket (***.socket**), ... L'outil de gestion des services (et des autres unités d'ailleurs) dans **systemd** s'appelle **systemctl**.

Si vous voulez avoir **syslog** en parallèle avec **journald**, il suffit d'installer **syslog-ng**, puis de l'activer :

```
$ sudo systemctl enable syslog-ng.service
```

systemd possède son propre mécanisme de journalisation, **syslog** n'est plus requis par défaut. La commande **journalctl** (pour l'utilisateur *root*) permet d'accéder au *log*.

- Tout le log : **journalctl** (l'option **-f** pour un affichage continu comme pour **tail**)
- Par service : **journalctl -u wicd**
- Par PID : **journalctl _PID=1**
- Par exécutable : **journalctl /usr/sbin/dhccpd**
- Par jour : **journalctl -since="today"**
- Par niveau : **journalctl -p err**

La configuration du journal de **systemd** est réalisée avec le fichier `/etc/systemd/journald.conf`.



cf. <https://doc.ubuntu-fr.org/systemd>

Les volumes logiques

Le RAID

Le RAID (*Redudant Array Of Independent Disks*) permet de combiner plusieurs disques en une seule partition dans le but d'augmenter soit la performance, soit la redondance des informations, soit les deux.

Les différentes méthodes RAID les plus courantes sont nommées par niveau (*level*). Ces techniques sont le *disk striping* (volume segmenté) ou RAID level 0, le *mirroring* (volume en miroir) ou RAID level 1 et le *disk striping with parity* (volume segmenté avec parité) ou RAID 5.

Le principe de base de ces techniques est la distribution des données sur plusieurs disques d'un même ensemble (*disk array*), les disques étant regroupés en un seul **volume logique**. Les données sont découpées en blocs de taille fixe (*chunks*, en général de 32ko ou 64ko), puis ces blocs sont distribués sur les différents disques du volume logique suivant un algorithme déterminé par le niveau RAID.

Il existe deux sortes de RAID :

- Le RAID matériel. Ce sont les contrôleurs de disques (SCSI ou IDE) qui gèrent eux-mêmes le RAID. Le processus est transparent pour le système d'exploitation qui ne voit qu'un seul disque. Il existe des cartes contrôleur qui permettent de créer des ensembles RAID, ainsi que des racks de disques pré configurés avec un niveau de RAID défini.
- Le RAID logiciel. Dans ce cas, un pilote de périphérique est en charge du RAID. Il existe plusieurs solutions pour Linux, dont le pilote MD que nous allons voir.

Les niveaux RAID

Les niveaux RAID sont définis comme suit :

- **Level 0.** Appelé "striping", est utilisé pour améliorer la **performance** en distribuant la charge de lecture/écriture de façon équitable sur tous les disques de l'ensemble. C'est le niveau qui offre le maximum de performances, les temps d'écriture et de lecture étant ainsi (théoriquement) divisés par le nombre de disques. La capacité de stockage de l'ensemble est égale à la somme des tailles des disques. Ce niveau abaisse la sécurité car il rend l'ensemble dépendant de la fiabilité de chaque disque.
- **Level 1.** Appelé "mirroring", est utilisé pour améliorer la **sécurité** en augmentant la redondance, les données étant écrites en plusieurs exemplaires sur plusieurs disques en même temps. C'est le niveau qui offre le maximum de sécurité, mais aussi celui le moins performant à cause du temps nécessaire à l'écriture des données sur chacun des disques. La vitesse de lecture/écriture de l'ensemble à égale à la vitesse du disque le plus lent, et la taille de l'ensemble est égale à la taille du plus petit disque.
- **Level 4.** Ce niveau est très peu utilisé. Il est destiné à améliorer la **sécurité** en utilisant un disque pour le stockage de la parité. Cette parité est utilisée pour recréer un disque defectueux. Si le disque de parité est defectueux, il peut être recréé à partir des disques de données. La capacité de stockage de l'ensemble est égale à la somme des tailles des disques de l'ensemble moins le disque de parité. La performance de l'ensemble est dépendante de la performance du disque de parité qui est sollicité pour toutes les opérations de lecture/écriture, quelque soit le disque sur lequel les données sont écrites.
- **Level 5.** C'est le niveau le plus utilisé. Il offre un maximum de **sécurité** avec une bonne **performance**. Il utilise le principe du striping du Level 0 et la technique de la parité du Level 4, sauf que la parité est répartie sur l'ensemble des disques, éliminant le problème de performance du Level 4 tout en offrant le même niveau de sécurité. La perte de performance vient surtout de l'algorithme utilisé pour le calcul de la parité et de la répartition. La capacité de stockage de l'ensemble est égale à la somme des tailles des disques de l'ensemble moins un (les disques doivent avoir la même taille).
- **Linéaire** ou **concaténé.** C'est un simple groupement de disques en un seul disque logique dans lequel les données sont écrites séquentiellement sans algorithme particulier. Cette solution n'offre aucun gain de performance, pose un problème de sécurité identique au level 0. La capacité de stockage de l'ensemble est égale à la somme des tailles des disques de l'ensemble.

Les ensembles RAID sont souvent constitués (en particuliers pour RAID 1 et RAID 5) de disques *hot swap* qui peuvent être remplacés à chaud, sans arrêt du système d'exploitation ni de l'ensemble RAID. Les disques SCSI et IDE normaux ne sont pas *hot swap*, il faut pour cela des disques spécifiques avec une alimentation électrique spécifique, comme par exemple les disques SCSI SCA ou SATA II.

Une fois le disque changé, il est alors possible de recréer l'intégrité de l'ensemble, soit, pour le RAID 1, de recréer le disque miroir defectueux à partir d'un autre disque, soit, pour le RAID 5, de recréer le disque defectueux à partir des données et des informations de parité des autres disques.

Note : nous ne verrons pas les techniques de récréation de disques.

Configuration

La configuration d'un ensemble RAID à l'aide du pilote MD se fait à l'aide de la commande `mdadm` (installez le paquet correspondant si cette commande n'est pas disponible).

Voici, ci-dessous, quelques exemples de création de volumes logiques avec `mdadm` :

Construction d'un ensemble concaténé à l'aide de deux disques SCSI :

```
# mdadm --create --verbose --level=linear --raid-devices=2 /dev/md0 /dev/sdb6 /dev/sdc5
```

Construction d'un ensemble RAID 0 à l'aide de deux disques SCSI :

```
# mdadm --create --verbose --level=0 --raid-devices=2 /dev/md0 /dev/sdb6 /dev/sdc5
```

Une fois l'ensemble créé et démarré, il faut créer le système de fichiers et le monter, comme pour n'importe quelle partition (`mkfs` et `mount`).

De plus, on peut avoir des informations concernant les ensembles RAID en cours de fonctionnement avec le fichier `/proc/mdstat` :

```
# cat /proc/mdstat
```

Pour stopper un ensemble utilisez la commande `mdadm -stop`. Par exemple :

```
# mdadm --stop /dev/md0
```

Note : les disques doivent être partitionnés avant de constituer l'ensemble. En effet, un ensemble RAID est en fait un ensemble de partitions et non de disques.

Pour que les ensembles soient activés au démarrage il faut créer un fichier de configuration `/etc/mdadm/mdadm.conf`. Ce fichier contient une ligne qui liste les partitions susceptibles d'être utilisées dans un ensemble, par exemple :

```
DEVICE /dev/hda*
```

Les autres lignes décrivent les ensembles et peuvent être générées par la commande `mdadm` :

```
# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Bien entendu, il ne faut pas oublier d'ajouter un ligne au fichier `/etc/fstab` pour que l'ensemble soit monté automatiquement au démarrage.

Cas du md127

Au redémarrage : Si votre RAID est non fonctionnel et reconnu comme `/dev/md127`,
Vous devez réaliser les commandes suivantes :

```
# mdadm --stop /dev/md127

# mdadm --assemble /dev/md0 /dev/sdb6 /dev/sdc5

# mdadm --detail --scan >> /etc/mdadm/mdadm.conf

# update-initramfs -u

# reboot
```



cf. http://doc.ubuntu-fr.org/raid_logiciel#md127

Swap et autres systèmes de fichiers

Gestion de la zone de swap

Concepts

La zone de swap est utilisée lorsque la mémoire physique (RAM) est remplie. Si le système a besoin de plus de ressources mémoires et que la mémoire physique est remplie, les pages de mémoire (page = bloc de taille fixe, en général 4Ko sur l'architecture PC) inactives (non utilisées depuis un certain temps) sont déplacées dans la zone de swap. Bien que la zone de swap peut aider sur des machines possédant peu de mémoire, il vaut mieux faire en sorte qu'elle ne soit **jamais** utilisée en dimensionnant correctement le serveur en fonction de son utilisation. En effet, la zone de swap se trouve sur disque dur, voire sur réseau, qui sont des supports beaucoup plus lents que la mémoire physique.

Pour que l'utilisation de la zone de swap par le système soit la plus efficace possible, il faut que :

- Elle soit stockée dans une partition dédiée et non un fichier normal,
- La partition dédiée à la zone de swap soit sur un disque (voire un contrôleur) différent de la partition système et des données les plus utilisées,
- Que sa taille soit au minimum la taille de la mémoire physique.

Sur Linux, la zone de swap peut être créée dans une partition dédiée, dans un fichier normal, ou sur le réseau.

On désigne mémoire virtuelle l'ensemble de la mémoire utilisée par le système, c'est à dire la mémoire physique plus les zones de swap.

Listage des zones de swap

Les zones de swap en cours d'utilisation par le système peuvent être listées avec la commande `swapon` :

```
# swapon -s
Filename                Type              Size    Used    Priority
/dev/hda5                partition         522072  0       -1
```

Ajout d'une zone de swap

Il est parfois nécessaire d'ajouter une zone de swap. Par exemple, si la taille de la mémoire physique est augmentée, si l'utilisation du serveur devient plus importante, ou pour un besoin temporaire lors du lancement d'un traitement lourd exceptionnel.

Ajout d'une partition de swap

Idéalement, il vaut mieux manipuler les zones de swap, et en particuliers les partitions de swap, lorsque le système est en mode maintenance:

```
# shutdown -t 900 now "Arrêt du serveur pour maintenance"
```

Il faut créer la partition sur un disque existant ou un nouveau disque. Ceci peut se faire avec les outils `parted`, `fdisk` ou autre (voir partie 1, chapitre 10).

Exemple, création d'une partition étendue de 256Mo pour une zone de swap sur un disque existant:

```
# fdisk /dev/hda
...
Commande (m pour aide) : p
...
Commande (m pour aide) : n
Premier cylindre (1358-5005, 1358 par défaut) :
Utilisation de la valeur par défaut 1358
Dernier cylindre ou +size ou +sizeM ou +sizeK (1358-5005, 5005 par défaut) : +256M
...
Commande (m pour aide) : p
...
Commande (m pour aide) : t
Nombre de partitions (1-6): 6
Code hexadécimal (tapez L pour afficher une liste des codes) : 82
Le type de système de la partition 6 a été remplacé par 82 (Echange Linux).
Commande (m pour aide) : w
La table de partition a été modifiée !
...
```

Il faut ensuite créer les structures de la zone de swap dans la partition:

```
# mkswap /dev/hda6
Setting up swapspace version 1, size = 4025155 kB
```

Puis activer cette zone de swap pour qu'elle puisse être utilisée immédiatement:

```
# swapon /dev/hda6
```

On peut vérifier l'activation:

```
# swapon -s
Filename                Type                Size    Used    Priority
/dev/hda5                partition           522072  0       -1
/dev/hda6                partition           262136  0       -2
```

Et enfin on rajoute la zone de swap dans le fichier `/etc/fstab` pour qu'elle soit activée automatiquement au démarrage en ajoutant une ligne comme celle-ci:

```
/dev/hda6  swap          swap          defaults      0 0
```

Ajout d'un fichier de swap

Il peut être pratique d'ajouter un fichier de swap en cas de besoin temporaire.

Pour cela, il faut commencer par créer un fichier vide de la taille de la zone de swap à ajouter (exemple pour l'ajout d'un fichier de swap de 256Mo placé sur la partition système):

```
# dd if=/dev/zero of=/swapfile bs=1M count=256
```

Comme pour une partition de swap, il faut:

Créer les structures de la zone de swap:

```
# mkswap /swapfile
Setting up swapspace version 1, size = 268431 kB
```

Puis activer la zone de swap:

```
swapon -s
Filename                Type                Size    Used    Priority
/dev/hda5                partition           522072  0       -1
/dev/hda6                partition           262136  0       -2
/swapfile                file                262136  0       -3
```

Il est possible de faire en sorte que cette zone de swap soit activée au démarrage, bien que ce ne soit pas une bonne idée d'utiliser un fichier de swap de façon définitive. Il suffit pour cela de rajouter une ligne de cette forme dans le fichier `/etc/fstab`:

```
/swapfile                swap          swap          defaults      0 0
```

Suppression d'une zone de swap

Que ce soit pour une partition ou pour un fichier, la suppression d'une zone de swap est identique:

Il vaut mieux être en mode maintenance:

```
# shutdown -t 900 now "Arrêt du serveur pour maintenance"
```

Stopper l'utilisation de la zone de swap immédiatement:

```
# swapoff /dev/hda6  
# swapoff /swapfile
```

Notez que cette opération peut prendre du temps si ces zones étaient très utilisées: il faut que le système transfère les pages mémoires depuis ces fichiers vers les autres zones de swap, voire vers la mémoire physique le cas échéant. Elle peut donc échouer si il n'y a pas assez de mémoire virtuelle (mémoire physique + zones de swap).

Si ces zones de swap étaient listées dans le fichier `/etc/fstab`, il ne faut pas oublier de supprimer les lignes correspondantes.

On peut ensuite supprimer physiquement la partition ou le fichier.

Ramfs

Un ramdisk est une zone de mémoire (virtuelle, le système se charge de la répartition en mémoire physique / zone de swap suivant la charge du système) utilisée comme partition. Une fois montée, cette partition est utilisable comme n'importe quelle partie du système de fichiers, qu'il soit sur disque ou sur réseau.

Il y a deux types de ramdisk utilisables dans Linux: l'ancien système, basé sur les périphériques `/dev/ram*`, et le nouveau (à partir de Linux 2.4), basé sur le système de fichiers `ramfs`.

Nous n'allons voir que le système le plus récent.

Pour créer un ramdisk, rien de plus simple: il suffit de le monter.

Exemple :

```
# mkdir /ramdisque
# mount -t ramfs none /ramdisque
```

On peut aussi fixer une taille maximum pour éviter un remplissage de la mémoire. Exemple d'un ramdisk limité à 10Mo :

```
# mkdir /ramdisque
# mount -t ramfs -o maxsize=10M none /ramdisque
```

On peut aussi faire en sorte qu'il soit créé au démarrage en rajoutant une ligne au fichier `/etc/fstab` :

```
none          /ramdisque    ramfs  defaults,maxsize=10M    0 0
```

Services réseau

DHCP

Concepts

DHCP (Dynamic Host Configuration Protocol) permet d'automatiser la configuration TCP/IP des machines du réseau, quel que soit leur système d'exploitation.

L'utilisation de DHCP simplifie l'administration système en regroupant en un seul point la configuration de tout un réseau, et permet de gérer simplement les problèmes de configuration des machines non fixes (portables, PDA, ...) entre les différents sites de l'entreprise.

Le principe de DHCP est le suivant : lorsque une machine démarre, elle émet un message de diffusion (*broadcast*) pour découvrir le serveur DHCP. Elle utilise ensuite les informations du premier serveur DHCP qui répond pour se configurer.

Configuration du service DHCP

Il est nécessaire d'installer le paquet `dhcp3-server` car seuls les paquets clients DHCP sont installés par défaut.

Le serveur DHCP ne pouvant pas se configurer lui-même (il doit connaître son adresse IP pour pouvoir répondre aux requêtes DHCP), il est nécessaire de configurer le serveur avec une adresse statique.

Le serveur DHCP utilise le fichier `/etc/dhcp/dhcpd.conf` comme fichier de configuration et stocke les informations de distribution d'adresses (quelles adresses ont été distribuées à qui) dans le fichier `/var/lib/dhcp/dhcp.leases`.

A chaque modification du fichier `dhcpd.conf`, il faut redémarrer le serveur DHCP :

```
# /etc/init.d/isc-dhcp-server restart
```


Voici un exemple simple de fichier `dhcpd.conf`:

```
option domain-name "mondomaine.com";
deny unknown-clients;
ddns-update-style none;
subnet 10.0.0.0 netmask 255.255.255.0
{
    option broadcast-address 10.0.0.255; # adresse de diffusion
    range 10.0.0.100 10.0.0.250; # plage d'adresses dynamiques
    option domain-name-servers 10.0.0.1, 10.0.0.254; # serveurs DNS
    # adresses semi-statiques
    host machine1
    {
        hardware ethernet 01:01:02:ae:34:c4;
        fixed-address 10.0.0.2;
    }
    # groupe des machines qui ont accès à Internet
    group {
        option routers 10.0.0.254;
        host machine2
        {
            hardware ethernet 02:01:02:ae:34:c4;
            fixed-address 10.0.0.3; # Adresse IP
            # Au cas où on démarre du réseau
            option root-path "/export/tftpboot/macppc";
        }
        host machine3
        {
            hardware ethernet 03:01:02:ae:34:c4;
            fixed-address 10.0.0.4; # Adresse IP
            # c'est une machine avec Windows
            option netbios-name-servers 10.0.0.12;
            option netbios-node-type 8;
        }
    }
}
```

Les machines peuvent être regroupées par groupe pour offrir des paramètres différents suivant certains critères. Par exemple, certaines machines n'ont pas la même passerelle par défaut car elles utilisent un accès à Internet dédié.

Le service DHCP peut aussi servir à renvoyer la configuration des démarrages par réseau (serveur de boot, fichiers de démarrage réseau, ...).

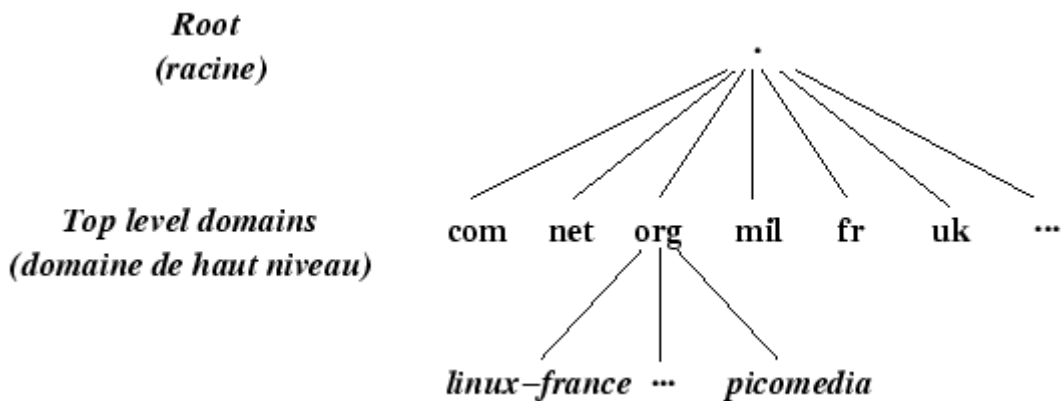
DNS

Concepts

Le service DNS (Domain Name System) est un composant indispensable dans le fonctionnement d'un réseau local ainsi que dans le fonctionnement d'Internet. Nous allons voir comment configurer le serveur de nom fourni avec Ubuntu Linux, BIND (Berkeley Internet Name Domain).

Ce service est utilisé pour associer les adresses IP aux noms complets (FQDN, Fully Qualified Domain Name) des machines (et inversement).

Le DNS est une base de données distribuée, chaque domaine et sous domaine (appelés *zones*) étant gérés par un serveur DNS différent (un serveur peut gérer plusieurs zones). De plus, les serveurs sont organisés entre eux de façon hiérarchique:



Chaque serveur DNS contient, pour chaque zone qu'il gère, les fichiers de la base de données permettant de convertir un nom de machine en adresse IP et inversement, ainsi que les noms et adresses des autres serveurs DNS de la zone et des serveurs de mail.

Une zone est gérée par un et un seul serveur DNS principal, et peut être répliquée sur un ou plusieurs serveurs secondaires.

BIND est composé, entre autre, du démon `/usr/sbin/named` et de la commande `/usr/sbin/rndc`, il lit sa configuration dans le fichier `/etc/bind/named.conf`, et stocke ses informations dans le répertoire `/var/cache/bind/`.

Configuration de BIND

Si BIND n'est pas déjà installé, on peut le faire à l'aide de la commande `apt` (package `bind9`).

Le fichier `named.conf` contient principalement :

- les options du service,
- les déclarations des zones.

Il est organisé par blocs de la forme :

```
motclé paramètre {
    option1;
    option2;
    ...
};
```

Les options les plus utiles sont :

- **directory** : répertoire dans lequel se trouvent les fichiers décrivant le contenu des zones.
- **forwarders** : liste des serveurs vers lesquels diriger les requêtes des machines inconnues.
- **notify** : méthode de notification des serveurs secondaires en cas de changement sur les zones (**yes**, notifier les serveurs secondaires; **no**, ne pas les notifier).

Une déclaration de zone a le format suivant :

```
zone "nom du domaine ou plage d'adresses IP" {
    type type;
    file "fichier";
}
```

où *type* est *master* (pour un serveur primaire) ou *slave* (pour un serveur secondaire), et *fichier* un fichier de zone, stocké dans le répertoire précisé par l'option *directory*.

Exemple de nom de domaine : `mondomaine.com`

Exemple de plage d'adresses IP : `168.192.IN-ADDR.ARPA` (pour le réseau de classe B `192.168.0.0/16`).

A chaque modification de ce fichier, il faut redémarrer le démon `named` :

```
# /etc/init.d/bind9 restart
```

A chaque modification d'un fichier de zone, il faut indiquer au démon `named` de relire les fichiers de zone, de préférence avec la commande `rndc`.

⚠ N'oubliez pas de configurer le service `named` pour qu'il démarre automatiquement au démarrage.

Voici un exemple de fichier `named.conf` :

```
options {
    directory "/var/cache/bind";
    forwarders {
        193.252.19.3;
        193.252.19.4;
    };
};

zone "intra.net" {
    type master;
    file "intra.net.hosts";
};

zone "0.168.192.IN-ADDR.ARPA" {
    type master;
    file "intra.net.rev";
};
```

La commande `rndc`

Cette commande permet de contrôler le démon `named` depuis le serveur ou à distance. Pour pouvoir l'utiliser, il faut configurer correctement le démon `named` et la commande elle-même. En effet, lors de la communication avec le démon `named`, `rndc` utilise un système de clé privée/publique pour l'authentification.

Lors de l'installation du package `bind`, une paire de clé publique/privée a été créée automatiquement et enregistrée dans le fichier `rndc.key`. Il vous suffit donc, pour pouvoir utiliser `rndc`, d'ajouter la ligne suivante au fichier `named.conf`:

```
include "/etc/bind/rndc.key";
```

Les actions les plus utiles de la commande `rndc` sont :

- `halt`, `stop`. Arrêter le démon `named`.
- `reload`. Relire les fichiers de zones.
- `status`. Afficher le statut du démon.
- `stats`. Ecrire des statistiques dans le fichier `/var/named/named.`

Fichiers de zones

Domaines (résolutions de noms)

Les fichiers de zones contiennent les enregistrements constituant les différents noms de machines et serveurs du domaine.

Un fichier de zone peut commencer par des directives comme :

- `$ORIGIN` : Nom de domaine par défaut lorsqu'il n'est pas précisé.
- `$TTL` : *Time To Live*, temps en secondes pendant lequel les enregistrements sont valides dans les caches des serveurs DNS.

Les différents types d'enregistrements les plus courants sont:

- `A` : *Address*, un nom de machine associé à une adresse IP.
- `CNAME` : *Canonical NAME*, un alias sur un nom de machine.
- `MX` : *Mail eXchange*, noms des serveurs de mails du domaine.
- `NS` : *Name Server*, noms des serveurs DNS.
- `SOA` : *Start Of Authority*, informations à propos de cette zone.

Exemple de fichier de zone:

```
$TTL 86400
@           IN           SOA dns1.intra.net. root.intra.net. (
                12345           ; numero de serie, a
                ; augmenter a chaque modification
                ; pour que les serveurs secondaires
                ; rafraichissent leur base de donnees
                21600           ; temps de rafraichissement des
                ; serveurs secondaires
                3600            ; temps d'attente apres une demande
                ; de rafraichissement erronee
                604800          ; temps de vie maximum d'une enregistrement
                ; de cette zone dans les caches des autres
                ; serveurs DNS
                86400           ; temps de vie minimum dans les caches
                )
                IN           NS           dns1.intra.net.
                IN           NS           dns2.intra.net.
                IN           MX           10 mail.intra.net.

dns1         IN           A           192.168.0.3
dns2         IN           A           172.16.0.216
mail         IN           A           192.168.0.2
www          IN           A           192.168.0.5
ftp          IN           CNAME        www
pop          IN           CNAME        mail
smtp         IN           CNAME        mail
```

Plage d'adresses IP (résolutions inverses)

Ces fichiers servent à convertir des adresses IP en noms de machine. Leur contenu est identique aux fichiers de zones des domaines, et le type d'enregistrement le plus utilisé est:

- PTR : *PoinTeR*, une adresse IP associée à un nom de machine.

Exemple de fichier de zone de résolution inverse:

```
$TTL 86400
@           IN           SOA dns1.intra.net. root.intra.net. (
                    12345      ; numero de serie
                              ; a augmenter a chaque modification
                              ; pour que les serveurs secondaires
                              ; rafraichissent leur base de donnees
                    21600     ; temps de rafraichissement des
                              ; serveurs secondaires
                    3600      ; temps d'attente apres une demande
                              ; de rafraichissement erronee
                    604800    ; temps de vie maximum d'une enregistrement
                              ; de cette zone dans les caches des autres
                              ; serveurs DNS
                    86400     ; temps de vie minimum dans les caches
                    )
           IN           NS           dns1.intra.net.
           IN           NS           dns2.intra.net.
1         IN           PTR           dns1.intra.net.
2         IN           PTR           mail.intra.net.
5         IN           PTR           www.intra.net.
```

NFS

Concepts

La plupart des systèmes d'exploitation de type UNIX fournissent une implémentation de NFS dérivée du produit NFS de Sun. NFS (*Network File System*) est un système de partage de fichiers qui utilise les protocoles TCP/IP, RPC et XDR.

Même si il souffre de graves défauts (mauvaise gestion des locks, peu de sécurité, ...) il reste le standard pour les partages de fichiers en réseaux hétérogènes (réseaux mélangeant différents systèmes d'exploitation). En effet, les autres systèmes de fichiers réseaux sont soit trop liés à un type de système d'exploitation (SMB, CIFS), soit propriétaires (NCP), soit trop lourd à mettre en oeuvre pour la plupart des réseaux locaux de petites tailles (Coda).

Le principal intérêt de NFS réside donc dans le fait qu'il fonctionne correctement et est performant sur la plupart des systèmes d'exploitation, sans avoir besoin de beaucoup de configuration.

☞ Actuellement la version 4 de NFS résout presque tous les problèmes du protocole mais est encore peu répandue, nous allons mettre en oeuvre la plus courante, la version 3.

Les termes utilisés dans NFS sont:

- *Serveur NFS*. Désigne le système qui possède physiquement les ressources (fichiers, répertoires) et les partages sur le réseau avec d'autres systèmes.
- *Client NFS*. Désigne un système qui monte les ressources partagées sur le réseau. Une fois montées, les ressources apparaissent comme si elles étaient locales.

☞ NFS ne s'occupe pas des droits d'accès aux fichiers et répertoires, il laisse le système d'exploitation s'occuper de la sécurité. Celle-ci est donc gérée de la même façon que pour les fichiers locaux, avec les droits UNIX standards.

Les démons

portmap

Étant donné que NFS utilise le protocole RPC, il est nécessaire que le démon `portmap` (qui gère les communications RPC entre les clients et les serveurs) soit démarré.

Vous pouvez tester la connectivité avec `portmap` comme ceci:

```
# rpcinfo -p serveur
```

où *serveur* est le nom d'hôte du serveur ou du client.

Vous pouvez démarrer manuellement ce service:

```
# /etc/init.d/portmap start
```

Côté serveur

Les différentes opérations du service NFS sont gérées par plusieurs démons:

- `rpc.mountd`. Reçoit et traite les demandes de montages (effectuées avec la commande `mount`). Il utilise le fichier `/etc/exports` pour connaître la liste des ressources partagées et quels sont les clients qui ont le droit de les monter.
- `rpc.nfsd`. (ou `nfsd`) Traite les opérations sur les ressources (lectures, écritures, suppressions, changements des droits, ...).
- `rpc.lockd`. Gère les accès partagés (locks) aux fichiers (optionnel). Remplacé dans les versions récentes de Linux par le démon `lockd` du système, plus performant et plus sûr.
- `rpc.statd`. Gère les redémarrages propres (optionnel). Utilisés après arrêt ou redémarrage d'un serveur alors que des clients étaient connectés.
- `rpc.rquotad`. Permet de gérer les informations sur les quotas disques depuis les clients NFS (optionnel).

Ces services peuvent être démarrés manuellement (le package à installé est `nfs-kernel-server`):

```
# /etc/init.d/nfs-kernel-server start
```

Côté client

Côté client, seuls les démons optionnels (`rpc.lockd`, `rpc.statd` et `rpc.rquotad`) peuvent être utiles si les fonctionnalités correspondantes doivent être fournies.

Configuration du serveur

La configuration du service NFS, côté serveur, se limite à lister les ressources partagées et les droits de montage.

☞ On ne configure que les droits de montage. NFS ne s'occupe pas des droits d'accès aux fichiers et répertoires, il laisse le système d'exploitation s'occuper de la sécurité. Celle-ci est donc gérée de la même façon que pour les fichiers locaux, avec les droits UNIX standards.

Le fichier `/etc/exports` contient la liste des ressources partagées, une ligne par ressource. Le format des lignes est le suivant:

```
répertoire client1(options) client2(options) ...
```

Où:

- *répertoire* est le nom de la ressource à partager. C'est forcément un répertoire local. Le chemin doit être absolu (il doit commencer par '/').
- *client* est un nom d'un ou plusieurs hôtes du réseau sous une des formes suivantes:
 - Nom de machine ou adresse IP, par exemple `machine.intra.net`
 - Sous réseau, par exemple `192.168.0.0/16`
- *options* (optionnel) est une liste d'options séparées par des virgules. Les options les plus utiles sont:
 - `ro` ou `rw` : Read Only (par défaut) ou Read Write
 - `root_squash` : L'utilisateur root local des clients (UID 0, GID 0) est considéré comme un utilisateur anonyme par le serveur (par défaut)
 - `no_root_squash` : L'utilisateur root des clients a les mêmes droits que l'utilisateur root du serveur (dangereux !)
 - `squash_uids` : liste des UID qui seront considérés comme utilisateurs anonymes par le serveur.

Exemple de fichier `/etc/exports`:

```
/export/home 192.168.0.0/16(rw,root_squash) admin.intra.net(rw,no_root_squash)
/usr/local   machin.intra.net(ro) 192.168.0.10(rw)
```

Étant donné que ce fichier est lu par le démon `rpc.mountd` lorsque celui-ci démarre, il faut lui demander de le relire en cas de modification:

```
# /etc/init.d/nfs-kernel-server reload
```

La liste des ressources partagées peut être obtenue à l'aide de la commande `showmount`:

```
# showmount -e
/export/home 192.168.0.0/16,admin.intra.net
/usr/local   machin.intra.net,192.168.0.10
```

Configuration des clients

Aucune configuration particulière n'est nécessaire pour les clients. Les ressources partagées peuvent être listées et montées/démontées manuellement ou automatiquement avec les commandes que nous avons déjà vu.

Exemples:

Listage des ressources partagées d'un serveur depuis un client (la commande `showmount` est installée par le paquet `nfs-common`) :

```
# showmount -e monserveur
/export/home 192.168.0.0/16,admin.intra.net
/export/public *.intra.net
/usr/local   machin.intra.net,192.168.0.10
```

Montage et démontage manuels d'une ressource :

```
# mkdir -p /partage
# mount monserveur:/export/home /partage
# ls /partage/
# umount /partage
```

Pour que cette ressource soit montée automatique au démarrage du client, il suffit de rajouter dans le fichier `/etc/fstab` du client la ligne suivante :

```
monserveur:/export/home /partage nfs rw 0 0
```

Nous verrons dans le chapitre sur l'automonteur comment faire en sorte que les ressources soient montées automatiquement uniquement lorsque l'on en a besoin (et démontées automatiquement lorsque elles sont inutilisées).

Sécurité

Au niveau de la sécurité, NFS ne s'occupe que des droits de montage (quels clients ont le droit de monter les ressources partagées). En ce qui concerne les droits d'accès aux fichiers et répertoires, NFS laisse le système d'exploitation s'en occuper. Les droits UNIX classiques (`user/group/other`) s'appliquent donc, à l'aide d'une correspondance entre les UID/GID des utilisateurs des machines clients et les UID/GID des utilisateurs du serveur (à l'exception de l'utilisateur `root`, voir les options `root_squash` et `no_root_squash` du fichier `/etc/exports`).

Il est donc nécessaire d'avoir les mêmes utilisateurs des deux côtés, voire la même base d'utilisateurs. NFS n'est donc utilisable pleinement qu'avec une base d'utilisateurs unique pour tout le réseau (voir le chapitre sur LDAP).

Remarques DHCP

Il est nécessaire d'installer le paquet `isc-dhcp-server` car seuls les paquets clients DHCP sont installés par défaut.

Il faut éditer le fichier de configuration `/etc/dhcp/dhcpd.conf`.

Configuration d'adresses semi-statiques :

```
deny unknown-clients; # voir remarque

subnet 10.0.0.0 netmask 255.0.0.0
{
    option broadcast-address 10.255.255.255; # adresse de diffusion
    range 10.0.0.100 10.0.0.250; # plage d'adresses dynamiques
    option routers 10.0.0.1; # passerelle par défaut
    option domain-name-servers 10.0.0.1, 10.0.0.6; # serveurs DNS
    # adresses semi-statiques
    host machine1
    {
        hardware ethernet 01:01:02:ae:34:c4;
        fixed-address 10.0.0.2;
    }
    # ...
}
```

Remarque : Seule la machine1 d'adresse MAC 01:01:02:ae:34:c4 obtiendra une adresse IP (10.0.0.2 d'après la configuration). Cela provient de l'option `deny unknown-clients` qui rejettera tous les autres clients car ils seront inconnus. Sinon, il faut enlever cette option !

Remarques DNS côté client

Le fichier `/etc/resolv.conf` permet d'indiquer le ou les domaines de recherche et les différents serveurs DNS à utiliser.

Exemple de fichier `/etc/resolv.conf` :

```
nameserver 192.168.0.3 # adresse du serveur DNS du réseau local
nameserver 10.0.0.1 # le serveur suivant à utiliser en cas de défaillance du serveur
précédent
search intra.net # nom du domaine géré par le serveur DNS local
```



Le fichier `/etc/resolv.conf` est maintenant généré automatiquement par le service `networking`. Vous pouvez éditer les fichiers `/etc/resolvconf/resolv.conf.d/head` et `/etc/resolvconf/resolv.conf.d/tail` qui produisent un contenu qui sera ajouté avant (*head*) et après (*tail*) dans le fichier `resolv.conf`.

Le configuration peut aussi être réalisée :

- par DHCP et l'option `domain-name-servers`
- directement dans le fichier `/etc/network/interfaces` avec les options `dns-nameservers` et `dns-search` à partir de certaines versions.

Exemple pour `eth0` dans le fichier `/etc/network/interfaces` :

```
iface eth0 inet static # ou ... dhcp
dns-nameservers 192.168.0.3 10.0.0.1
dns-search intra.net
```

LDAP et l'automoteur

Principes d'un annuaire

Concepts

Un annuaire, ou service de noms, est une application client-serveur dont le rôle est de convertir les requêtes de nommage, comme les noms de machines, les noms d'utilisateurs, répertoires personnels... en leur identifiant ou localisation associés.

L'annuaire centralise l'information d'un réseau d'une entreprise. Cette centralisation apporte les avantages suivants:

- Un point unique d'administration.
- Une information consistante.
- Une vue uniforme du réseau.
- La propagation immédiate des changements à tous les clients.
- L'assurance que les clients ne perdront pas l'information.
- Des systèmes de sauvegarde, par exemple avec des serveurs secondaires.
- Des systèmes de répartition de charge (toujours avec des serveurs secondaires).

Le but est de fournir toutes les informations utiles au fonctionnement du réseau à partir d'une seule et unique source.

Services d'annuaires

Nous avons déjà vu un service de noms, le DNS, qui est spécialisé dans les noms de machines.

Les autres services de type annuaire les plus souvent rencontrés sont:

- NIS (*Network Information Service*). Annuaire plat utilisé en environnement UNIX. Il permet de centraliser les informations normalement contenues dans les fichiers situés dans le répertoire `/etc` des clients. De moins en moins utilisé.
- Lan Manager et dérivés (NT). Annuaire plats utilisés en environnement MS-DOS, IBM OS/2 et Microsoft Windows (domaines NT). Permet de gérer les utilisateurs et groupes seulement. De moins en moins utilisé.
- NIS+. Annuaire hiérarchisé et distribué disponible pour l'environnement Sun Solaris et UNIX. Peu répandu.
- Active Directory. Annuaire hiérarchisé et distribué disponible pour l'environnement Microsoft Windows.
- NDS (*Novell Directory Server*). Annuaire hiérarchisé et distribué pour l'environnement Novell Netware.
- LDAP (*Lightweight Directory Access Protocol*). Annuaire hiérarchisé et protocole de consultation d'annuaires (le protocole seul peut être utilisé avec d'autres annuaires comme Active Directory, Novell ou des bases de données).

Le commutateur de service de noms

On peut utiliser les services d'annuaires suivants avec Linux :

- DNS,
- NIS,
- NIS+,
- LDAP.

Ils s'utilisent en plus des informations locales (fichiers `/etc/hosts`, `/etc/passwd`, `/etc/shadow`, `/etc/group`, ...) pour l'authentification et les informations des groupes, des utilisateurs (noms, uids, répertoires, shells, ...) , des machines et des services réseaux.

Pour chaque type d'information, on peut fixer l'ordre de recherche de l'information à l'aide du fichier `/etc/nsswitch.conf` à l'aide d'une ligne de ce format:

```
information: service1 [condition] service2 [condition] service3 ...
```

Les types d'informations sont les suivants:

- **aliases**. Alias d'adresses électroniques pour le MTA.
- **ether**. Correspondances entre les adresses IP et les adresses physiques (Ethernet).
- **group**. Liste des groupes.
- **hosts**. Correspondances entre les adresses IP et les noms de machines.
- **network**. Liste des masques des réseaux.
- **passwd**. Liste des utilisateurs.
- **protocols**. Liste des protocoles réseaux.
- **rpc**. Liste des services RPC.
- **services**. Liste des services TCP/IP et de leurs numéros de ports.
- **shadow**. Mots de passe des utilisateurs.

Ces types d'informations correspondent aux contenus des fichiers locaux de mêmes noms (`/etc/aliases`, `/etc/ether`, `/etc/passwd`, `/etc/group`, ...).

Les services peuvent être:

- **files**. Utilise les fichiers locaux situés dans `/etc`.
- **dns**. Utilise le service DNS (uniquement pour l'entrée **hosts**).
- **nis** ou **yp**. Utilise le service NIS.
- **nis+** ou **nisplus**. Utilise le service NIS.
- **ldap**. Utilise le protocole LDAP.

Les conditions permettent d'effectuer des actions suivant le résultat de la consultation du service (précédent la condition). Les états de retour des consultations sont:

- **success**. L'information a été trouvée.
- **notfound**. Le service a répondu mais ne contient pas l'information cherchée.
- **unavail**. Le service n'a pas répondu.
- **tryagain**. Le service répond mais est temporairement indisponible.

Et les actions possibles sont:

- **return**. Arrêter la recherche.
- **continue**. Continuer la recherche avec le service suivant.

Exemple de fichier `/etc/nsswitch.conf`:

```
passwd:      files nis ldap
group:       files nis ldap
shadow:      files nis ldap

hosts:       dns [!UNAVAIL=return] files
networks:    nis [NOTFOUND=return] files
ethers:      nis [NOTFOUND=return] files
protocols:   nis files
rpc:         nis files
services:    nis files
```

LDAP

Principes

LDAP (Lightweight Directory Access Protocol) est un système d'annuaire basé sur X500 est apporté, entre autre :

- Un architecture arborescente.
- Un protocole de communication utilisable sur Internet, notamment à travers les firewalls.
- Un protocole de communication crypté.

LDAP n'est pas qu'un système d'annuaire, c'est aussi un protocole qui permet d'interroger des systèmes d'annuaires X500 comme par exemple Microsoft Active Directory.

Le standard X500 définit comment un annuaire doit être organisé. Les annuaires X500 sont organisés sous forme d'arbre avec une racine et différents niveaux pour chaque catégorie d'informations (par exemple: par villes, par société, par service, ...).

Un annuaire simple

Il est possible de centraliser les informations des utilisateurs et des groupes d'un réseau composé de machines UNIX et Windows avec un annuaire LDAP extrêmement simple. Voici un exemple d'arborescence classique pour un réseau de taille moyenne :

- Une racine portant le nom du domaine de la société, par exemple : mondomaine.com. Le nom X500 (appelé DN : Distinguished Name) du domaine sera :

```
dn: dc=mondomaine, dc=com
```

(dc veut dire Domain Component, qui identifie un racine de l'arbre)

- Une branche pour stocker les informations des utilisateurs, People :

```
dn: ou=People, dc=mondomaine, dc=com
```

(ou : Organizational Unit, qui identifie un branche de l'arbre)

- Une branche pour stocker les informations des groupes, Group :

```
dn: ou=Group, dc=mondomaine, dc=com
```

- Un utilisateur LDAP qui fait office d'administrateur, admin:

```
dn: cn=admin, dc=mondomaine, dc=com
```

(cn : Common Name, qui identifie feuille de l'arbre)

Cette arborescence est suffisante pour les comptes UNIX. Pour les comptes Windows, il suffira de rajouter deux branches pour stocker les informations sur les domaines et les machines des domaines.

Configuration d'un serveur LDAP

Le service LDAP n'est pas installé par défaut lors de l'installation de Ubuntu Linux. Il faut donc l'installer à l'aide de la commande `apt-get` (paquets `slapd` et `ldap-utils`).

Fichiers de configurations

Une fois le package installé, il faut modifier la configuration à l'aide de la commande `dpkg-reconfigure slapd` pour déclarer le domaine:

```
# sudo dpkg-reconfigure slapd
```

Modifiez aussi la ligne « BASE » du fichier `/etc/ldap/ldap.conf`. Ce fichier est utilisé pour les outils LDAP en ligne de commande (cela nous servira pour tester le bon fonctionnement du service).

La commande `dpkg-reconfigure` crée le répertoire `/etc/ldap/slapd.d` qui contient la configuration du serveur, ainsi qu'une base de données vide dans le répertoire `/var/lib/ldap`.

Une fois les répertoires créés il faut ajouter les classes d'objets que l'on veut stocker dans l'annuaire. Ces classes sont déclarées dans des fichiers nommés schémas :

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Enfin on initialise la base de données avec un fichier de type LDIF contenant la configuration :

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f ldapconfig.ldif
```

Démarrage du service

Une fois le fichier de configuration prêt, démarrez le service :

```
# /etc/init.d/slapd start
```

Outils LDAP

Il est nécessaire d'installer le paquet `ldap-utils` pour bénéficier des commandes de gestion LDAP : `ldapadd`, `ldapsearch`, ...

Création de la base de données

Maintenant que le serveur LDAP est installé et le domaine initialisé, il faut y mettre des informations, c'est à dire au minimum :

- Le domaine.
- L'utilisateur admin.
- Les branches.
- Les groupes.
- Les utilisateurs.

Sans outil d'administration graphique on peut communiquer avec notre serveur LDAP à l'aide de fichiers textes au format LDIF (LDAP Data Interchange Format). Nous allons utiliser ces fichiers pour remplir notre base à partir des groupes et des utilisateurs existants du système.

Pour commencer, il faut créer un fichier texte que nous nommerons `domain.ldif` contenant la description du domaine et de l'arborescence :

```
dn: dc=mondomaine,dc=com
objectClass: top
objectClass: dcObject
objectclass: organization
o: Example Organization
dc: mondomaine

dn: cn=admin,dc=mondomaine,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
userPassword: secret

dn: ou=people, dc=mondomaine ,dc=com
ou: people
objectClass: organizationalUnit

dn: ou=group, dc=mondomaine ,dc=com
ou: group
objectClass: organizationalUnit
```

Un fois ce fichier créé, il faut insérer ces données dans la base à l'aide de la commande `ldapadd` :

```
# ldapadd -D "cn=admin,dc=mondomain,dc=com" -W -x -f domain.ldif
```

Il est possible de voir le contenu de la base de données à l'aide de la commande `slapcat`.

Pour les groupes, nous allons utiliser les groupes existants.

Avec OpenLDAP est fourni un programme (disponible dans le paquet `migrationtools`) qui transforme un fichier `/etc/group` en fichier LDIF. Cet outils utilise la variable d'environnement `LDAP_BASEDN` pour connaître le nom du domaine :

```
# export LDAP_BASEDN="dc=mondomaine,dc=com"
# /usr/share/migrationtools/migrate_group.pl /etc/group groups.ldif
```

La même manipulation est à effectuer pour les utilisateurs :

```
# /usr/share/migrationtools/migrate_passwd.pl /etc/passwd passwd.ldif
```

Une fois les fichiers générés et édités, la commande `ldapadd` permet d'envoyer les données au serveur LDAP:`slapd`

```
# ldapadd -D "cn=admin,dc=mondomain,dc=com" -W -x < groups.ldif
# ldapadd -D "cn=admin,dc=mondomain,dc=com" -W -x < passwd.ldif
```

On peut vérifier que les données sont bien dans la base avec la commande `ldapsearch` :

```
# ldapsearch -x
```

Configuration des clients

Lors de l'installation des paquets `ldap-auth-client` et `libnss-ldap` la commande `dpkg-reconfigure` affiche une série de questions sur :

- L'URL du serveur LDAP (exemple : `ldap://monserveur/`)
- Le base DN utilisé (exemple : `dc=mondomaine,dc=com`)
- Le compte utilisé si l'accès anonyme est désactivé (évitiez d'utiliser le compte admin...)

Cet outil se charge ensuite de modifier pour vous les fichiers `/etc/ldap.conf` et `/etc/ldap.secret`.

Si cela n'a pas marché il est possible d'exécuter à nouveau la configuration :

```
# dpkg-reconfigure ldap-auth-config
```

Il est cependant obligatoire de modifier le fichier `/etc/ldap.conf` si la structure de votre annuaire n'est pas une structure « classique ».

Il vous faudra aussi modifier le fichier `/etc/nsswitch.conf` pour que le système utilise LDAP. Pour cela, la commande `auth-client-config` est utilisée :

```
# auth-client-config -t nss -p lac_ldap
# pam-auth-update
```

A partir de ce moment, il est possible d'ouvrir une session avec n'importe quel compte du domaine à partir de n'importe quel client, à condition que les répertoires des utilisateurs soient accessibles depuis toutes les machines clientes.

Gestion des utilisateurs

Il existe des outils graphiques ou web (par exemple `phpLdapAdmin`) pour l'administration de l'annuaire LDAP.

Cependant il est toujours pratique de pouvoir gérer les utilisateurs et groupes depuis la ligne de commande, ne serait-ce que pour automatiser ces tâches à l'aide de scripts. Les outils standard LDAP (`ldapadd`, `ldapmodify`, etc...) ne sont pas très pratiques car ils nécessitent la manipulation de fichiers au format LDIF.

Il existe un paquet nommé `ldapscripts` qui contient des outils pratiques pour la gestion de l'annuaire, par exemple :

- `ldapadduser`, `ldapdeleteuser`, `ldapmodifyuser`
- `ldapsetpasswd`
- `ldapaddgroup`, `ldapdeletegroup`, `ldapmodifygroup`
- `ldapaddusertogroup`, `ldapdeleteuserfromgroup`

Pour voir tous les scripts fournis par le paquet :

```
# dpkg -L ldapscripts | grep bin
```

Pour être utilisables ces outils doivent être configurés (fichier `/etc/ldapscripts/ldapscripts.conf`) et nécessite le mot de passe de l'utilisateur `admin` (fichier `/etc/ldapscripts/ldapscripts.passwd`).

L'automonteur

Nécessite de l'automonteur

Pour que les répertoires de tous les utilisateurs soient accessibles depuis tous les clients, il faut :

- Exporter le répertoire `/home` depuis le serveur :

Ajoute de la ligne suivante dans `/etc/exports`:

```
/home * (rw,async)
```

Et forcer la relecture de ce fichier:

```
/etc/init.d/nfs-kernel-server reload
```

- Monter les répertoires `/home` depuis tous les clients en ajoutant la ligne suivante au fichier `/etc/fstab` :

```
mon_serveur:/home /home nfs defaults 2 2
```

Mais cette technique a des inconvénients :

- Si les répertoires des utilisateurs sont sur des serveurs ou dans des répertoires différents, il faut une entrée pour chaque serveur/répertoire dans le fichier `/etc/exports` et dans les fichiers `/etc/fstab` de chaque client. Cela devient rapidement une contrainte dans une grosse entreprise.
- La modification de la configuration des clients nécessite de passer sur chaque poste client pour modifier `/etc/fstab`. Même problème que précédemment.
- Tous les répertoires des utilisateurs sont montés sur toutes les machines clients. Cela génère un trafic réseau inutile et surcharge les serveurs pour rien.

Concepts

L'automonteur est un système qui permet de monter les ressources locales (disques, CDRom, clés USB, ...) ou distantes (partages NFS) automatiquement et de façon transparente à l'utilisateur.

En effet, on a vu qu'il existe deux moyens de monter des ressources:

- Manuellement avec la commande `mount`. Celle-ci ne pouvant être utilisée que par l'utilisateur `root`, il est impossible pour un utilisateur normal de monter, par exemple, des partages NFS.
- Automatiquement au démarrage du système à l'aide du fichier `/etc/fstab`. L'inconvénient de cette solution est que les partages réseaux sont montés tout le temps sur tous les clients, même si ils ne sont pas utilisés. Ceci a pour effet de charger le réseau et les serveurs inutilement. De plus, les répertoires des utilisateurs ne sont pas forcément stockés sur le même serveur.

Avantages

Les avantages de l'utilisation de l'automonteur sont les suivants:

- Les ressources sont montées à la demande, c'est à dire uniquement lorsque on les utilise.
- Les ressources sont démontées automatiquement. A partir d'un certain temps d'inutilisation, l'automonteur démonte les partages.
- L'utilisation de l'automonteur n'a aucune incidence sur les serveurs NFS, la configuration est nécessaire uniquement côté client. De plus, la configuration de l'automonteur peut être centralisée grâce à NIS ou LDAP.

Fonctionnement

L'automonteur est constitué de deux composants qui fonctionnent de manière simultanée pour accomplir un montage automatique:

- le programme `automount`.

Ce démon monte les répertoires à la demande et les démonte en cas de non utilisation. Il utilise des *tables de montage* pour connaître les répertoires à monter et les options de montage correspondantes.

- Le système de fichiers `autofs`.

Ce système de fichiers a pour fonction d'intercepter les requêtes d'accès aux répertoires (et à leur contenu) gérés par l'automonteur. A chaque accès, `autofs` envoie au démon `automount` une requête, qui va monter le répertoire ou simplement réinitialiser le temps de non utilisation.

Il utilise aussi les *tables de montage* pour connaître les répertoires à surveiller.

☞ L'automonteur peut servir à monter autre chose que des partages NFS. On peut aussi l'utiliser pour des partages SMB/CIFS (Serveurs Microsoft Windows) et aussi pour les CDROM, clés USB, ...

Configuration locale (sur les clients)

La configuration manuelle (à l'aide de fichiers de configuration) de l'automonteur consiste à remplir (uniquement sur le client) les tables de montages. Dans le jargon d'automont une table est un fichier de configuration qui lie un répertoire local sur le client avec un répertoire sur le serveur.

La table master

Cette table référence l'ensemble des tables. Elle contient une ligne par table, et chaque ligne lie un répertoire à une table.

Cette table est stockée dans le fichier `/etc/auto.master`.

Exemple de table `/etc/auto.master`:

```
/home    /etc/auto.home
/mnt     /etc/auto.mnt
```

Les autres tables

Les tables directes associent un sous répertoire à une source et des options de montage (optionnelles). Elles contiennent une ligne par sous répertoire et sont stockées dans les fichiers listés dans la table master.


Exemples de lignes possibles:

```
toto      serveur:/home/staff/toto
cantona   serveur:/home/guests/cantona
public    grosserveur.intra.net:/export/public
openoffice -ro,soft  srvoo:/opt/OpenOffice2.0
solitaire -fstype=smbfs  ://MACHINE1/C
windoz    -fstype=vfat  :/dev/hda1
cdrom     -fstype=iso9660  :/dev/hdd
usb       -fstype=auto  :/dev/sda1
```

Démarrage et utilisation

Pour démarrer le service `autofs` :

```
# service autofs start
```

 N'oubliez pas de configurer le service `autofs` pour qu'il démarre automatiquement au démarrage.

Il suffit ensuite de se déplacer dans les répertoires listés dans les tables pour que les montages s'effectuent. Une inactivité de l'utilisateur provoque un démontage.

L'automonteur et LDAP

L'automonteur s'utilise souvent avec LDAP, notamment pour les répertoires des utilisateurs, ce qui permet de centraliser les tables de montages pour simplifier l'administration.

L'utilisation de l'automonteur avec LDAP implique de placer les tables de l'automonteur dans la base de données LDAP sur le serveur.

Pour installer l'automonteur et son extension LDAP, installez le paquet `autofs-ldap`.

Sur le serveur LDAP : schéma

Pour ajouter le schéma `autofs` à la configure du serveur LDAP :

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/autofs.ldif
```

Note : ce fichier peut ne pas exister suivant la version du paquet, voir dans le répertoire du TP pour le trouver.

Sur le serveur LDAP : données

Il faut tout d'abords ajouter trois branches à notre arborescence LDAP pour stocker les informations de l'automonteur, soit l'équivalent des fichiers `auto.master` et `auto.home` :

```
dn: ou=Automount,dc=mondomaine,dc=com
ou: Automount
objectClass: organizationalUnit

dn: ou=auto.master,ou=Automount,dc=mondomaine,dc=com
ou: auto.master
objectClass: top
objectClass: automountMap

dn: ou=auto.home,ou=Automount,dc=mondomaine,dc=com
ou: auto.home
objectClass: top
objectClass: automountMap
```

Une fois ces branches ajoutées, on ajoute une sous branche pour le montage automatique du répertoire `/home`, dans la branche `auto.master`, qui référence la branche `auto.home` :

```
dn: cn=/home,ou=auto.master,ou=Automount,dc=mondomaine,dc=com
cn: /home
objectClass: automount
automountInformation: ldap:monserveur:ou=auto.home,ou=Automount,dc=mondomaine ,dc=com
```


Ensuite, pour chaque entrée de la table auto.home, il faut ajouter l'enregistrement correspondant dans l'annuaire. Voici une entrée type pour un répertoire utilisateur :

```
dn: cn=cantona,ou=auto.home,ou=Automount,dc=mondomaine,dc=com
cn: cantona
objectClass: automount
automountInformation: -fstype=nfs,hard,intr,nodev,nosuid serveur:/home/guests/cantona
```

Et voici une entrée qui permet de prendre en charge tous les utilisateurs dont les répertoires sont sur le même serveur dans le même répertoire :

```
dn: cn=/,ou=auto.home,ou=Automount,dc=mondomaine,dc=com
cn: /
objectClass: top
objectClass: automount
automountInformation: -fstype=nfs,hard,intr,nodev,nosuid serveur:/home/&
```

Sur les clients

Si le client est déjà configuré pour l'authentification LDAP, il suffit que l'automonteur soit démarré (et qu'il est été configuré pour démarrer automatiquement) et qu'il y est la ligne suivante dans le fichier `/etc/nsswitch.conf` :

```
automount: ldap
```

Notez qu'il est nécessaire de redémarrer l'automonteur après la modification du fichier `/etc/nsswitch.conf` pour que l'automonteur lise les nouvelles tables depuis le serveur LDAP.

☛* Pour éviter toute confusion, il est conseillé d'effacer toutes les tables existantes sur les clients.

Remarques LDAP

Étape n°1 : authentification LDAP



cf. <http://www.itzgeek.com/how-tos/linux/debian/install-and-configure-openldap-on-ubuntu-16-04-debian-8.html>

Côté serveur :

```
# apt-get install slapd ldap-utils migrationtools

# dpkg -l | grep ^ii | grep -E "ldap|slapd"
ii  ldap-utils                2.4.31-1+nmu2ubuntu8.2      amd64      OpenLDAP utilities
ii  libldap-2.4-2:amd64      2.4.31-1+nmu2ubuntu8.2      amd64      OpenLDAP libraries
ii  slapd                    2.4.31-1+nmu2ubuntu8.2      amd64      OpenLDAP server (slapd
)

# dpkg-reconfigure slapd

...
Choose the backend format for LDAP: HDB
...
```

Puis :

```
# export LDAP_BASEDN="dc=xxx,dc=esimed"

# vim /etc/ldap/ldap.conf
BASE    dc=xxx,dc=esimed
```

```
# ldapsearch -x
# ldapsearch -x -D "cn=admin,dc=xxx,dc=esimed" -W
...
dn: dc=xxx,dc=esimed
objectClass: top
objectClass: dcObject
objectClass: organization
o: xxx
dc: xxx

dn: cn=admin,dc=xxx,dc=esimed
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
...

# slapcat
...
dn: dc=xxx,dc=esimed
objectClass: top
objectClass: dcObject
objectClass: organization
o: xxx
dc: xxx
structuralObjectClass: organization
entryUUID: b546785c-6454-1035-9d75-f518921c4641
creatorsName: cn=admin,dc=xxx,dc=esimed
createTimestamp: 20160210151422Z
```

```

entryCSN: 20160210151422.198955Z#000000#000#000000
modifiersName: cn=admin,dc=xxx,dc=esimed
modifyTimestamp: 20160210151422Z

dn: cn=admin,dc=xxx,dc=esimed
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9ZkxvMzE2UWF4N3doNGplSk85MFh4ZUhnWU1xbjBjQ2I=
structuralObjectClass: organizationalRole
entryUUID: b54ae414-6454-1035-9d76-f518921c4641
creatorsName: cn=admin,dc=xxx,dc=esimed
createTimestamp: 20160210151422Z
entryCSN: 20160210151422.227923Z#000000#000#000000
modifiersName: cn=admin,dc=xxx,dc=esimed
modifyTimestamp: 20160210151422Z

```

Vérification et initialisation (si nécessaire) :

```

// La liste des schémas disponibles :
# ls /etc/ldap/schema/
collective.ldif  core.ldif      duaconf.ldif  inetorgperson.ldif  ldapns.schema  nis.schema
pmi.schema      collective.schema  core.schema  duaconf.schema  inetorgperson.schema  misc.ldif
openldap.ldif  ppolicy.ldif  corba.ldif   cosine.ldif     dyngroup.ldif   java.ldif
misc.schema    openldap.schema  ppolicy.schema  corba.schema  cosine.schema   dyngroup.schema
java.schema    nis.ldif       pmi.ldif     README

// La liste des schémas déjà ajoutés dans l'annuaire :
# ls -l /etc/ldap/slapd.d/cn\=config/cn\=schema
-rw----- 1 openldap openldap 15527 févr. 10 16:14 cn={0}core.ldif
-rw----- 1 openldap openldap 11361 févr. 10 16:14 cn={1}cosine.ldif
-rw----- 1 openldap openldap 6491 févr. 10 16:14 cn={2}nis.ldif
-rw----- 1 openldap openldap 2855 févr. 10 16:14 cn={3}inetorgperson.ldif

// Sinon ajout des classes d'objets :
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif

// Définition de la base de données :
# cat /etc/ldap/slapd.d/cn\=config/cn\=module\{0\}.ldif
...
dn: cn=module{0}
objectClass: olcModuleList
cn: module{0}
olcModulePath: /usr/lib/ldap
olcModuleLoad: {0}back_hdb
structuralObjectClass: olcModuleList
entryUUID: b541424c-6454-1035-87e8-8185b263d2ac
creatorsName: cn=config
createTimestamp: 20160210151422Z
entryCSN: 20160210151422.164810Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20160210151422Z

# cat /etc/ldap/slapd.d/cn\=config/olcDatabase\=\{1\}hdb.ldif
...
dn: olcDatabase={1}hdb
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig

```

```
olcDatabase: {1}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=xxx,dc=esimed
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous
  auth by dn="cn=admin,dc=xxx,dc=esimed" write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by dn="cn=admin,dc=xxx,dc=esimed" write by * read
olcLastMod: TRUE
olcRootDN: cn=admin,dc=xxx,dc=esimed
olcRootPW:: e1NTSEF9ZkxvMzE2UWF4N3doNGplSk85MFh4ZUhnWULxbjBjQ2I=
olcDbCheckpoint: 512 30
olcDbConfig: {0}set_cachesize 0 2097152 0
olcDbConfig: {1}set_lk_max_objects 1500
olcDbConfig: {2}set_lk_max_locks 1500
olcDbConfig: {3}set_lk_max_lockers 1500
olcDbIndex: objectClass eq
structuralObjectClass: olcHdbConfig
entryUUID: b5415e8a-6454-1035-87ea-8185b263d2ac
creatorsName: cn=config
createTimestamp: 20160210151422Z
entryCSN: 20160210151422.165534Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20160210151422Z

// Sinon initialisation :
# vim ldapconfig.ldif
```

Load dynamic backend modules

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb
```

Database settings

```
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=local,dc=net
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=xxx,dc=esimed
olcRootPW: xxxxxxxx
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=xxx,dc=esimed" write by anonymous
  auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=xxx,dc=esimed" write by * read
```

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f ldapconfig.ldif
```

```
// Description du domaine :
# slapcat | grep "dc=xxx,dc=esimed"
dn: dc=xxx,dc=esimed
creatorsName: cn=admin,dc=xxx,dc=esimed
modifiersName: cn=admin,dc=xxx,dc=esimed
dn: cn=admin,dc=xxx,dc=esimed
creatorsName: cn=admin,dc=xxx,dc=esimed
modifiersName: cn=admin,dc=xxx,dc=esimed

// Sinon initialisation :
# vim domain.ldif
```

```
dn: dc=xxx,dc=esimed
objectClass: top
objectClass: dcObject
objectclass: organization
o: Example Organization
dc: xxx

dn: cn=admin,dc=xxx,dc=esimed
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
userPassword: secret
```

```
// éventuellement :
# /etc/init.d/slapd start
```

Maintenant, les utilisateurs et les groupes :

```
// Branches pour les utilisateurs (people) et les groupes (group) :
# slapcat | grep "ou=people"
# slapcat | grep "ou=group"

// Sinon ajout :
# vim xxx.esimed.ldif
```

```
dn: ou=people, dc=xxx, dc=esimed
ou: people
objectClass: organizationalUnit

dn: ou=group, dc=xxx, dc=esimed
ou: group
objectClass: organizationalUnit
```

```
# ldapadd -D "cn=admin,dc=xxx,dc=esimed" -W -x < xxx.esimed.ldif
adding new entry "ou=people, dc=xxx, dc=esimed"
adding new entry "ou=group, dc=xxx, dc=esimed"

# /usr/share/migrationtools/migrate_passwd.pl /etc/passwd passwd.ldif
# /usr/share/migrationtools/migrate_group.pl /etc/group groups.ldif

# ldapadd -D "cn=admin,dc=xxx,dc=esimed" -W -x < groups.ldif
# ldapadd -D "cn=admin,dc=xxx,dc=esimed" -W -x < passwd.ldif

# ldapsearch -x
```

Côté client :

```
# apt-get install ldap-auth-client ldap-auth-config
# auth-client-config -t nss -p lac_ldap

# dpkg -l | grep -E "ldap"
ii ldap-auth-client          0.5.3          all          meta-package for LDAP
   authentication
ii ldap-auth-config         0.5.3          all          Config package for
   LDAP authentication
ii libldap-2.4-2:amd64     2.4.31-1+nmu2ubuntu8  amd64       OpenLDAP libraries
ii libnss-ldap:amd64      264-2.2ubuntu4.14.04.1  amd64       NSS module for using
   LDAP as a naming service
ii libpam-ldap:amd64      184-8.5ubuntu3  amd64       Pluggable
   Authentication Module for LDAP

# cat /etc/ldap.conf | grep '^[^#]'
base dc=xxx,dc=esimed
uri ldap://192.168.0.1/
ldap_version 3
pam_password md5

# cat /etc/nsswitch.conf | grep '^[^#]'
passwd: files ldap
group: files ldap
shadow: files ldap
hosts:      files dns
networks:   files
protocols:  db files
services:   db files
ethers:     db files
rpc:        db files
netgroup:  nis

// Vérification :
# getent passwd
...
user1*:1001:1001:user1:/home/user1:/bin/bash

# getent group
...
user1*:1001:

# su --login user1
Mot de passe :
No directory, logging in with HOME=/

// Sinon :
# dpkg-reconfigure ldap-auth-config
```

Conclusion : il est maintenant possible de s'authentifier à partir d'un compte présent dans l'annuaire LDAP mais il y a un problème concernant le répertoire personnel de l'utilisateur. Il serait possible d'exporter en NFS les répertoires personnels des utilisateurs concernés.

Étape n°2 : automonteur et LDAP

Rappels :

- Le système de fichiers `autofs` a pour fonction d'intercepter les requêtes d'accès aux répertoires (et à leur contenu) gérés par l'automonteur. A chaque accès, `autofs` envoie au démon `automount` une requête, qui va monter le répertoire ou simplement réinitialiser le temps de non utilisation.
- L'automonteur s'utilise souvent avec LDAP, notamment pour les répertoires des utilisateurs, ce qui permet de centraliser les tables de montages pour simplifier l'administration. L'utilisation de l'automonteur avec LDAP implique de placer les tables de l'automonteur dans la base de données LDAP sur le serveur.

Côté serveur :

```
# apt-get install autofs-ldap

# ls /etc/ldap/schema/a*
/etc/ldap/schema/autofs.schema

// pas de fichier ldif donc : soit convertir le schema en ldif avec slapcat soit le télécharger
# wget https://launchpadlibrarian.net/55451730/autofs.ldif

# ldapadd -Y EXTERNAL -H ldapi:/// -f autofs.ldif

// pour tous les répertoires des utilisateurs dans /home :
# vim automount.ldif
```

```
dn: ou=Automount,dc=xxx,dc=esimed
ou: Automount
objectClass: organizationalUnit

dn: ou=auto.master,ou=Automount,dc=xxx,dc=esimed
ou: auto.master
objectClass: top
objectClass: automountMap

dn: ou=auto.home,ou=Automount,dc=xxx,dc=esimed
ou: auto.home
objectClass: top
objectClass: automountMap

dn: cn=/home,ou=auto.master,ou=Automount,dc=xxx,dc=esimed
cn: /home
objectClass: automount
automountInformation: ldap:192.168.0.1:ou=auto.home,ou=Automount,dc=xxx,dc=esimed

dn: cn=/,ou=auto.home,ou=Automount,dc=xxx,dc=esimed
cn: /
objectClass: top
objectClass: automount
automountInformation: -fstype=nfs,hard,intr,nodev,nosuid 192.168.0.1:/home/&

dn: cn=user1,ou=auto.home,ou=Automount,dc=xxx,dc=esimed
cn: user1
objectClass: automount
automountInformation: -fstype=nfs,hard,intr,nodev,nosuid 192.168.0.1:/home/user1
```

```
# ldapadd -D "cn=admin,dc=xxx,dc=esimed" -W -x -f automount.ldif

# cat /etc/exports
/home *(rw,async)

# service nfs-kernel-server restart
```

Côté client(s) :

```
# apt-get install autofs-ldap

# cat /etc/nsswitch.conf
...
automount: ldap

# cat /etc/default/autofs
MASTER_MAP_NAME="ou=auto.master,ou=automount,dc=xxx,dc=esimed"
LDAP_URI="ldap://192.168.0.1/"
SEARCH_BASE="ou=automount,dc=xxx,dc=esimed"
MAP_OBJECT_CLASS="automountMap"
ENTRY_OBJECT_CLASS="automount"
MAP_ATTRIBUTE="ou"
ENTRY_ATTRIBUTE="cn"
VALUE_ATTRIBUTE="automountInformation"

# service autofs restart

// Vérification :
# su --login user1
Mot de passe :
user1@client:~$ pwd
/home/user1
```

Remarque : sur les clients, 3 fichiers sont concernés

- /etc/default/autofs
- /etc/autofs_ldap_auth.conf
- /etc/nsswitch.conf



cf. <https://help.ubuntu.com/community/AutofsLDAP>

Concepts

Le logiciel Samba est utilisé pour le partage de fichiers et d'imprimantes à l'aide des protocoles SMB et CIFS. Ces protocoles étant ceux utilisés pour les systèmes d'exploitation Microsoft, l'installation de Samba sur une machine équipée de Linux permet d'intégrer celle-ci dans le "réseau Microsoft" de l'entreprise, voire de prendre la place d'un serveur Microsoft Windows.

L'utilisation combinée de LDAP, NFS et Samba sur un même serveur permet le fonctionnement d'un réseau hétérogène composé de machines Linux, Microsoft Windows et MacOS X. Les utilisateurs peuvent ainsi utiliser aussi bien un système que l'autre tout en gardant leurs identifications (nom d'utilisateur, mot de passe) et en ayant leurs données disponibles (répertoires personnels stockés sur le serveur) sur les deux systèmes.

Configuration

Après avoir installé le paquet `samba`, on peut configurer le service à l'aide du fichier `/etc/samba/smb.conf`.

Une fois configuré, le service peut être arrêté et redémarré à l'aide du script de démarrage :

```
# service smb restart
# service nmb restart
```

Le fichier `/etc/samba/smb.conf` est composé de deux parties :

- Une partie globale, qui permet de configurer le fonctionnement du service.
- Une partie partages, où sont listés les partages de répertoires et d'imprimantes et leurs paramètres.

Ce fichier est pré-configuré sous Ubuntu, il suffit de l'enlever les commentaires sur les lignes suivantes pour partager le répertoire `/home` pour tout les utilisateurs :

```
[homes]
  comment = Home Directories
  browseable = no
```

Pour voir si samba fonctionne et quelles sont les connections en cours :

```
# smbstatus
```

Pour voir les partages (avec le paquet `smbclient`) :

```
# smbclient -L nomduserveur
```

Gestion des utilisateurs

Dans sa configuration par défaut Samba utilise sa propre base d'utilisateurs. En effet, tous les utilisateurs du serveur ne sont pas forcément des utilisateurs de machines Microsoft Windows.

De plus la base des utilisateurs ne contient pas forcément toutes les informations nécessaires aux clients Microsoft.

Si vous voulez qu'un utilisateur puisse se connecter au serveur Samba depuis une machine Microsoft Windows, vous devez l'ajouter manuellement, soit en utilisant SWAT (onglet "Password"), soit avec la commande `smbpasswd` :

```
# smbpasswd -a utilisateur
```

Il faut bien sûr que l'utilisateur existe en temps qu'utilisateur UNIX.

Intégration à un domaine

Pour intégrer un serveur Samba à un domaine NT existant et géré par un serveur Microsoft Windows NT, il faut utiliser la commande `smbpasswd` :

```
# smbpasswd -r NOM_PDC -U Administrateur -J DOMAINE
```

Gestion d'un domaine

Il est possible de configurer Samba comme contrôleur de domaine NT (PDC, *Primary Domain Controller*). Ceci permet aux utilisateurs de clients Microsoft Windows d'ouvrir une session comme si ils étaient connectés à un serveur Microsoft Windows NT et de bénéficier d'un environnement réseau complet (répertoire personnel, script d'ouverture de session, authentification, ...) et identique à leur environnement réseau sous Linux (si vous avez aussi configuré NIS ou LDAP et l'automonteur).

Pour activer la gestion des domaines il faut ajouter les lignes suivantes dans le fichier de configuration `/etc/samba/smb.conf` :

```
encrypt passwords = yes
domain master = yes
domain logons = yes
wins support = Yes
```

La gestion des utilisateurs est la même, mais il faut en plus gérer les machines qui seront membres du domaine (si besoin, de plus en plus rare avec les ordinateurs portables). Dans un domaine NT, une machine a un compte utilisateur, il faut donc le créer. Cela se passe comme pour les utilisateurs, sauf que les noms des comptes machines doivent se terminer par un caractère `$`.

Exemple d'ajout d'un compte machine dans la base des utilisateur UNIX:

```
# adduser -d /tmp -s /bin/false machine\$\
```

Pour ajouter dans la base des utilisateurs de Samba un compte machine avec la commande `smbpasswd`, il faut utiliser l'option `-m`:

```
# smbpasswd -a -m machine
```

Samba et LDAP (Gestion d'un domaine NT)

Installation du schema dans LDAP

Copiez le fichier `samba.ldif` contenant le schema pour les objets utilisés par Samba puis importez le dans l'annuaire :

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f samba.ldif
```

Pour que Samba utilise la base de données LDAP il vous faut modifier le fichier de configuration `/etc/samba/smb.conf` en ajoutant les lignes suivantes :

```
ldap suffix = dc=mondomaine,dc=com
ldap ssl = off
ldap admin dn = cn=admin,dc=mondomaine ,dc=com
ldap group suffix = ou=Group
ldap user suffix = ou=People
ldap machine suffix = ou=Computers
passdb backend = ldapsam:ldap://localhost
```

(note : supprimer la ligne 'passdb backend' existante).

Puis enregistrer le mot de passe de l'utilisateur « `cn=admin,dc=mondomaine ,dc=com` » pour que Samba puisse s'authentifier auprès du serveur LDAP :

```
$ smbpasswd -w monmotdepasse
```

Rajoutez dans l'arborescence LDAP la branche qui contiendra tous les domaines :

```
dn: ou=Domains,dc=mondomaine,dc=com
ou: Domains
objectClass: organizationalUnit
```

La branche qui contiendra les comptes des machines :

```
dn: ou=Computers,dc=mondomaine,dc=com
ou: Machines
objectClass: organizationalUnit
```

Ainsi que l'entrée correspondant à votre domaine :

```
dn: sambaDomainName=mondomaine ,ou=Domains,dc=mondomaine,dc=com
objectClass: sambaDomain
sambaDomainName: mondomaine
sambaSID: S-1-5-21-3532407026-2732750505-3074995650
sambaAlgorithmicRidBase: 1000
```

La valeur de l'attribut `sambaSID` (Identifiant de Sécurité Windows) est obtenue avec la commande suivante :

```
# net getlocalsid
```

Gestion des utilisateurs et des machines

Pour chaque compte qui doit pouvoir se connecter à l'aide d'une machine Windows il faut ajouter le compte UNIX dans la base de données LDAP puis exécuter la commande `smbpasswd` pour ajouter ou modifier les attributs spécifiques à Samba dans la base LDAP, par exemple :

```
# smbpasswd -a utilisateur
```

A partir de ce moment l'utilisateur peut accéder à son répertoire personnel à travers l'URI :

```
\\MONSERVEUR\UTILISATEUR
```

Vous pouvez vérifier quels sont les utilisateurs qui ont le droit de se connecter par Samba à l'aide d'une recherche dans l'annuaire :

```
# ldapsearch -x "(objectclass=sambaSamAccount) "
```

De même, pour chaque machine Windows membre du domaine il faut créer un compte UNIX (désactivé) puis modifier ce compte pour Samba, par exemple pour une machine Windows nommée POSTE1 :

```
# ldapaddgroup computers
# ldapadduser poste1\$\ computers
# smbpasswd -m -a poste1\$\
```

Vous pouvez maintenant intégrer POSTE1 au domain.

Notez que les noms des comptes machines dans LDAP se terminent par le caractère « \$ ».

Lorsqu'une machine doit être intégrée au domaine, utilisez le compte root. Par défaut, seul ce compte a le droit de modifier le SID des machines dans la base LDAP lors d'une intégration au domaine.

Note :

Si vous votre base de données LDAP contient déjà des comptes UNIX il convient de les modifier à l'aide de la commande `smbpasswd` de la même façon que ci-dessus (y compris le compte root).

Remarques SAMBA

Samba et LDAP (Gestion d'un domaine NT)



cf. <https://help.ubuntu.com/lts/serverguide/samba-ldap.html>

Installer le paquet `smbldap-tools` :

```
# apt-get install smbldap-tools
```

Ce paquet fournit les commandes suivantes :

```
/usr/sbin/smbldap-groupadd
/usr/sbin/smbldap-groupdel
/usr/sbin/smbldap-grouplist
/usr/sbin/smbldap-groupmod
/usr/sbin/smbldap-groupshow
/usr/sbin/smbldap-passwd
/usr/sbin/smbldap-populate
/usr/sbin/smbldap-useradd
/usr/sbin/smbldap-userdel
/usr/sbin/smbldap-userinfo
/usr/sbin/smbldap-userlist
/usr/sbin/smbldap-usermod
/usr/sbin/smbldap-usershow
```

Pour configurer manuellement le paquet, vous devez créer puis éditer les fichiers `/etc/smbldap-tools/smbldap.conf` et `/etc/smbldap-tools/smbldap_bind.conf` :

`/etc/smbldap-tools/smbldap.conf`

```
SID="S-1-5-21-2406945009-1442160664-939686157"
masterLDAP="127.0.0.1"
masterPort="389"
slaveLDAP="127.0.0.1"
ldapTLS="0"
verify="require"
suffix="dc=mondomaine,dc=com"
usersdn="ou=Users,${suffix}"
computersdn="ou=Computers,${suffix}"
groupsdn="ou=Groups,${suffix}"
idmapdn="ou=ldmap,${suffix}"
scope="sub"
userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
userGecos="User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"
defaultMaxPasswordAge="3650"
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
```

La valeur du SID (Identifiant de Sécurité Windows) est obtenue avec la commande suivante :

```
# net getlocalsid
```

```
/etc/smbldap-tools/smbldap_bind.conf
```

```
masterDN="cn=admin,dc=mondomaine,dc=com"
masterPw="secret"
```

Le script `smbldap-populate` va ensuite ajouter les objets LDAP nécessaires pour Samba.

```
// sauvegarde de l'annuaire
# slapcat -l backup.ldif

# smbldap-populate

// export au format LDIF
# smbldap-populate -e samba-populate.ldif
```

Votre répertoire LDAP a maintenant les informations nécessaires pour authentifier les utilisateurs Samba.

Il est possible de consulter l'état des groupes et utilisateurs :

```
# smbldap-grouplist
gid |cn          |
512 |Domain Admins |
513 |Domain Users  |
514 |Domain Guests |
515 |Domain Computers |
544 |Administrators |
...

# smbldap-groupshow "Domain Admins"
dn: cn=Domain Admins,ou=Groups,dc=esimed,dc=com
objectClass: top,posixGroup,sambaGroupMapping
cn: Domain Admins
gidNumber: 512
memberUid: root,user1
description: Netbios Domain Administrators
sambaSID: S-1-5-21-2406945009-1442160664-939686157-512
sambaGroupType: 2
displayName: Domain Admins

# smbldap-userlist
uid |username
...
1001 |user1          |
10000 |machine1$      |
0 |root           |
65534 |nobody         |
```

Manipulations sur les clients Windows

Windows 7

Les clés de registres windows 7 à créer et/ou modifier (avec `regedit`) :

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanWorkstation\Parameters]
"DomainCompatibilityMode"=dword:00000001
"DNSNameResolutionRequired"=dword:00000000
```

Windows XP

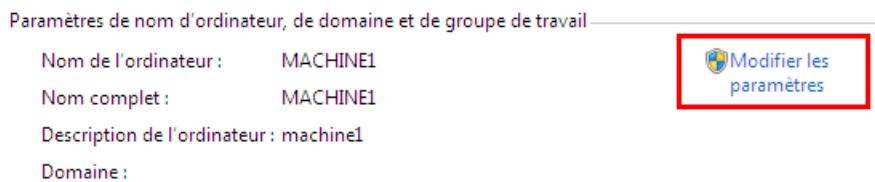
Normalement, la jonction au domaine est immédiate.

Néanmoins des expériences d'autres utilisateurs montrent que ces clés peuvent être nécessaires dans certains cas (avec regedit) :

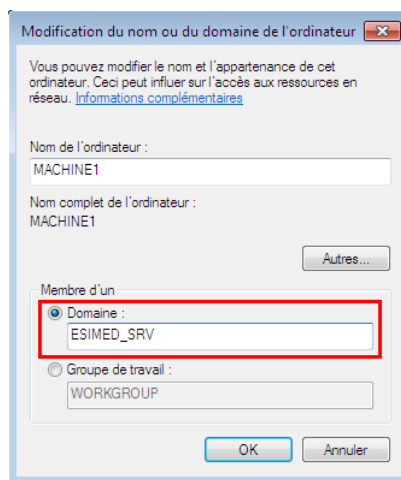
```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\System]
"CompatibleRUPSecurity"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\parameters]
"enableplaintextpassword"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters]
"requiresignorseal"=dword:00000000
```

Screenshots

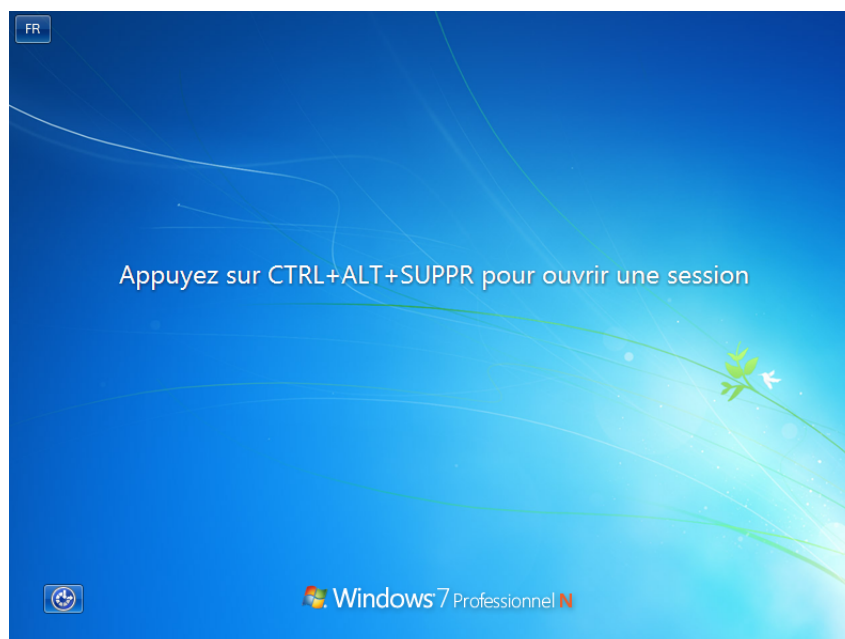
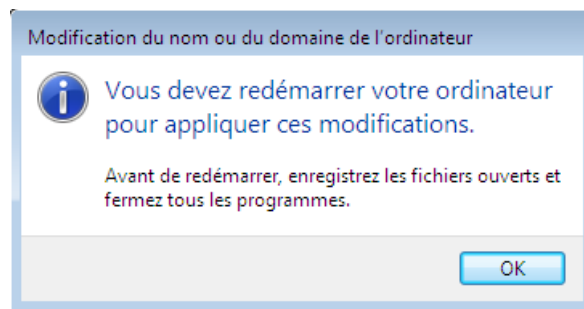
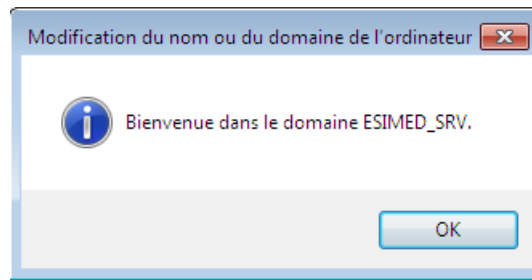
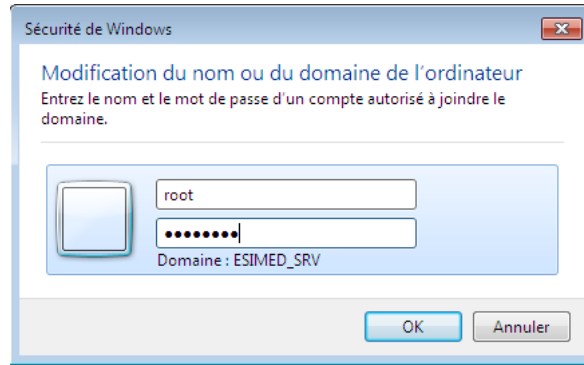
Aller dans "Propriétés" de l'"Ordinateur" :




On saisit le nom de domaine :



On doit s'authentifier en root :

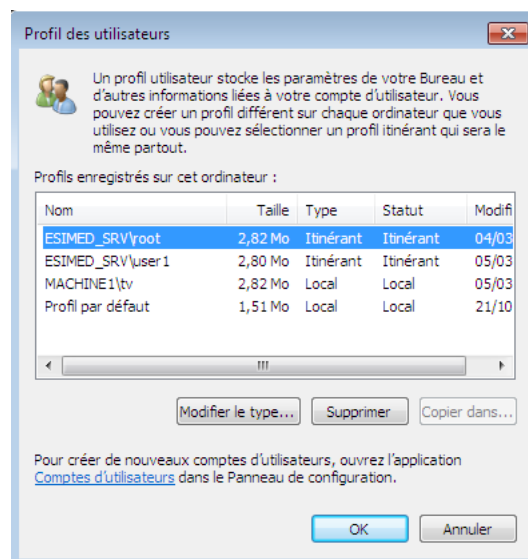
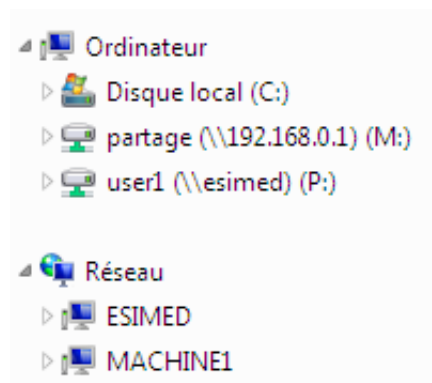


 Dans VirtualBox, vous pouvez faire Host (Ctrl droite) + Suppr

On ouvre une session sur le domaine :



Une fois connecté, on peut obtenir par exemple :



Personnalisation des clients Windows

Il est possible d'ajouter les options ci-dessous dans le fichier de configuration `/etc/samba/smb.conf` :

```
# It specifies the location of the user's
# profile directory from the client point of view) The following
# required a [profiles] share to be setup on the samba server (see
# below)
    logon path = \\%L\profiles\%U

# The following setting only takes effect if 'domain logons' is set
# It specifies the location of a user's home directory (from the client
# point of view)
    logon drive = P:
    logon home = \\%L\%U

# The following setting only takes effect if 'domain logons' is set
# It specifies the script to run during logon. The script must be stored
# in the [netlogon] share
# NOTE: Must be store in 'DOS' file format convention
    logon script = logon.bat

# Un-comment the following and create the netlogon directory for Domain Logons
# (you need to configure Samba to act as a domain controller too.)
[netlogon]
    comment = Network Logon Service
    path = /home/samba/netlogon
    guest ok = yes
    read only = yes

# Un-comment the following and create the profiles directory to store
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
[profiles]
    comment = Users profiles
    path = /home/samba/profiles
    guest ok = no
    browseable = no
    writable = yes
    read only = no
    create mask = 0600
    directory mask = 0700

[partage]
    comment = Partage
    browseable = yes
    writeable = yes
    path = /home/samba/partage
```



Il vous faut redémarrer le serveur Samba.

Il faut maintenant créer les répertoires associés au partage Samba :

```
# mkdir -p /home/samba/netlogon
# mkdir /home/samba/profiles
# mkdir /home/samba/partage

# touch /home/samba/netlogon/login.bat
```



Il vous faudra gérer les droits UNIX sur ces répertoires.

À titre d'exemple, voici un script `login.bat` qui sera exécuté au démarrage de la session :

```
net use m: \\192.168.0.1\partage /persistent:no
```

Les profils itinérants des utilisateurs seront stockés dans `/home/samba/profiles` :

```
/home/samba/profiles/user1.V2:
AppData
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
NTUSER.DAT
ntuser.ini
Pictures
Saved Games
Searches
Videos
```