L'adresse du réseau est constituée en remplaçant la partie de l'hôte dans l'adresse IP par des « 0 » (zéros). En ce qui concerne l'adresse de diffusion, on l'obtient en remplaçant la partie de l'hôte par des « 1 ».

Un autre méthode, plus mathématique, permet d'obtenir ces deux adresses :

```
<Adresse réseau> = <Adresse IP> AND <Masque réseau>
<Adresse de diffusion> = <Adresse IP> OR NOT(<Masque réseau>)
```

où AND, OR et NOT sont les opérateurs logiques classiques.

c) Les classes IP

Les réseaux d'adresses IP ont été regroupés en classes. Les 2 premiers bits de l'adresse IP déterminent la classe IP à laquelle cette adresse IP appartient. De ce fait, la classe IP détermine la taille du réseau (*tableau 3*).

Tableau 3. Les classes en IPv4

Premiers 2 bit	Classe IP	Taille du réseau	Nombre d'adresses IP du réseau	Exemple
00 ou 01	A	1 octet	224 = 16777216	115.0.0.1 **01**110011.00000000.00000000.00000001
10	B	2 octets	216 = 65536	130.1.1.1 **10**000010.00000001.00000001.00000001
11	C	3 octets	28 = 65536	212.50.14.82 **11**010100.00110010.00001110.01010010

Dans chaque classe IP il existe un ou plusieurs réseaux réservés pour des besoins privés. Ce sont des adresses qui ne sont pas prises en compte par les routeurs de l'Internet et peuvent être utilisées uniquement pour des réseaux internes. Tandis que les adresses « normales » sont, en principe, uniques dans tout l'Internet (leur distribution est contrôlée), la même adresse privée peut être présente dans plusieurs réseaux privés.

Tableau 4. Réseaux privés par classe en IPv4

Classe IP	Adresses privées
A	10.x.x.x (un réseau de classe A)
B	de 172.16.x.x à 172.31.x.x (16 réseaux de classe B)

C	192.168.x.x (255 réseaux de classe C)

d) Les sous-réseaux

La division en classes IP standard provoque un gaspillage important d'adresses IP (très peu d'organisations ont besoin de 16 milliards d'adresses IP).

En utilisant des masques réseaux différents, il est possible de subdiviser un réseau en sous-réseaux en fonction des besoins. Un fournisseur d'accès le fait fréquemment pour distribuer des adresses à ses clients, mais un administrateur peut être amené à le faire pour gérer plusieurs sites.

Par exemple un réseau avec un masque /24 (normalement la taille d'une classe C) peut être divisé en 2 sous-réseaux de taille /25 ou en 4 sous-réseaux de taille /26.

B. La suite TCP/IP

TCP/IP est le nom commun d'un ensemble de protocoles. Ce sont les protocoles sur lesquels sont basées toutes les communications Internet – IP, ICMP, TCP, UDP.

Les communications par TCP/IP sont organisées en couches, comme dans le modèle OSI. La différence est qu'ici le nombre de couches est réduit à 4, sinon l'idée est la même : chaque couche réalise des fonctionnalités spécifiques et communique avec les couches voisines.

Les quatre couches du modèle TCP/IP sont décrites dans le *tableau 5*.

Tableau 5. Pile TCP/IP

Couche	Description	Protocoles
Application	La couche supérieure – comme son nom l'indique. On y trouve les applications réseau	FTP, HTTP, SMPT, POP, IMAP, ….
Transport	Transmission des données	TCP, UDP
Internet	Datagrammes et routage – connexion des réseaux	IP, ICMP, ARP(?), RIP, BGP, IGMP
Network Access	Communication au niveau physique	Ethernet, Token Ring, etc.

Les protocoles principaux assurant le fonctionnement de l'Internet sont présentés dans le *tableau 6*.

Tableau 6. Protocoles associés à la pile TCP/IP

Protocole	Description
IP	C'est le protocole utilisé pour le découpage de l'information (les segments TCP ou UDP) en paquets (datagrammes) et pour le routage de ces paquets de l'émetteur au destinataire. C'est aussi le protocole qui définit l'adressage des différentes machines connectées à l'Internet et le regroupement de ces machines en réseaux.
TCP	*Transmission Control Protocol* – c'est le protocole fiable de transmission des données. Il travaille en mode connecté pour transmettre les données d'une application à une autre tout en assurant le contrôle de l'intégrité des données.
UDP	*User Datagram Protocol* – ce protocole assure aussi la transmission des données entre applications mais en mode « non fiable » : l'intégrité des données n'est pas contrôlée. Les applications utilisant ce protocole doivent elles-mêmes vérifier cette intégrité. Ce protocole est plus performant que TCP.
ICMP	*Internet Control Message Protocol* – ce protocole est destiné à gérer les informations relatives aux erreurs pouvant survenir sur le réseau.

C. Les ports

Les protocoles TCP et UDP assurent la communication entre deux applications (*tableau 6*).

Nous savons déjà que pour distinguer les machines nous utilisons les adresses IP (chaque machine dispose de sa propre adresse IP unique). Il faut aussi pouvoir distinguer les connexions entre applications.

La solution est l'utilisation des ports. Une connexion TCP entre deux applications est identifiée par 4 informations : l'adresse IP de la première machine, le port TCP de la première application, l'adresse IP de la deuxième machine et le port TCP de la deuxième application.

Le même principe est valable pour les connexions UDP. Elles sont identifiées par des ports UDP.

Il faut remarquer qu'une connexion peut être réalisée entre deux applications qui se trouvent sur la même machine. De même une application donnée (donc utilisant toujours le même port) peut participer à plusieurs connexions (ce sont les applications serveurs). Dans tous les cas la combinaison des quatre informations est unique.

Pour initier une connexion TCP (ou respectivement UDP) il faut au préalable que l'une des deux applications commence à « écouter » sur un certain port TCP. Cette application est le serveur. La deuxième application, que l'on appelle le client, va initier la connexion vers ce port TCP (ou port UDP dans le cas d'une communication par UDP).

Les ports de 1 à 1023 sont réservés aux serveurs, les ports de 1024 à 65 535 sont utilisés dynamiquement par les clients (et par les serveurs parfois).

Une liste des services réseaux classiques peut être trouvée dans le fichier /etc/services. Voilà un extrait de ce fichier :

```
ftp-data   20/tcp
ftp        21/tcp
fsp        21/udp  fspd
ssh        22/tcp              # SSH Remote Login Protocol
ssh        22/udp              # SSH Remote Login Protocol
telnet     23/tcp
# 24 - private
smtp       25/tcp  mail
# 26 - unassigned
time       37/tcp  timserver
time       37/udp  timserver
rlp        39/udp  resource    # resource location
nameserver 42/tcp  name        # IEN 116
whois      43/tcp  nicname
re-mail-ck 50/tcp              # Remote Mail Checking Protocol
re-mail-ck  50/udp             # Remote Mail Checking Protocol
domain      53/tcp  nameserver # name-domain server
domain      53/udp  nameserver
mtp         57/tcp              # deprecated
bootps      67/tcp              # BOOTP server
bootps      67/udp
bootpc      68/tcp              # BOOTP client
bootpc      68/udp
tftp        69/udp
gopher      70/tcp              # Internet Gopher
gopher      70/udp
rje         77/tcp  netrjs
finger      79/tcp
www         80/tcp  http        # WorldWideWeb HTTP
www         80/udp              # HyperText Transfer Protocol
```

Les informations spécifiques à chaque interface se trouvent dans le répertoire /etc/sysconfig/network-scripts.

Le fichier se nomme ifcfg-<nom de l'interface>, par exemple ifcfg-eth0.

```
#/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=134.157.9.52
NETMASK=255.255.255.0
NETWORK=134.157.9.0
BROADCAST=134.157.9.255
ONBOOT=yes
```

Il contient essentiellement quatre informations :

- le nom de l'interface ;
- la manière d'affecter l'adresse IP (statiquement ou dynamiquement) ;
- les informations IP (adresse, masque, réseau, diffusion) ;
- s'il faut démarrer automatiquement l'interface.

Enfin, il est possible de nommer les interfaces, mais ce n'est pas normalisé et reste très dépendant de la distribution Linux utilisée.

La distribution Debian stocke les informations de toutes les interfaces réseau dans le fichier /etc/network/interfaces. On y indique pour chaque interface son nom et les informations de réseau associées.

```
auto eth0
   iface eth0 inet static
   address 192.168.10.10
   netmask 255.255.255.0
   network 192.168.10.0
   broadcast 192.168.10.255
   gateway 192.168.10.1
```

B. Démarrage et arrêt du réseau

a) Démarrage « classique »

On utilise la commande traditionnelle de Unix `ifconfig`.

Syntaxe

```
/sbin/ifconfig interface [informations reseau] [options]
```

Trois exemples :

```
ifconfig eth0 193.54.85.245 netmask 255.255.255.224 up
```

```
ifconfig eth0 up
```

```
ifconfig eth0 down
```

b) Démarrage en utilisant les fichiers de configuration

On utilise la commande `/sbin/ifup` qui récupère la configuration à partir de `/etc/sysconfig/network` et `/etc/sysconfig/network-scripts/`, fichiers cités précédemment.

```
/sbin/ifup eth0
```

Pour fonctionner, la commande recherche les informations pour l'interface « eth0 » dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` qui doit exister et être renseigné.

c) Démarrage de toutes les interfaces

Comme pour la plupart des services sous Linux, on peut démarrer le service « réseau » (donc toutes les interfaces) avec le script `/etc/rc.d/init.d/network` qui utilise la commande `ifup` vue précédemment pour chaque interface définie. Sur les systèmes de la famille Red Hat, les interfaces sont définies par des fichiers de configuration dans le répertoire /etc/sysconfig/network-scripts/ (ifcfg-eth0 pour l'interface « eth0 »).

```
/etc/rc.d/init.d/network start
```

Ce script récupère des informations supplémentaires dans le fichier /etc/sysctl.conf. Par exemple, on peut trouver dans ce fichier la ligne `net.ipv4.ip_forward=1`, le système va ainsi relayer (*forward*) les paquets IP entre les interfaces pour se transformer en routeur. Ceci a pour effet de modifier le contenu du fichier `/proc/sys/net/ipv4/ip_forward` en remplaçant « 0 », qui est la valeur par défaut », par « 1 ».

d) Renouvellement de bail DHCP

Les outils permettant de récupérer une nouvelle adresse IP depuis un serveur DHCP (*Dynamic Host Configuration Protocol*) sont, selon la distribution utilisée, `pump` ou `dhcpclient`.

Attention, le serveur DHCP de Linux s'appelle **dhcpd**, alors que le client se nomme **dhcpcd**.

C. Routage

Lorsque l'on utilise la commande `ifup` ou le script de démarrage du réseau, les informations concernant la passerelle sont lues directement depuis le fichier /etc/sysconfig/network (champ GATEWAY).

Lorsque l'on utilise `ifconfig` ou si la passerelle n'est pas renseignée, il faut indiquer cette passerelle (et d'autres routes éventuellement) par la commande `/sbin/route` qui permet de configurer la table de routage.

Pour ajouter une route statique vers le réseau 192.168.100.0/24 en utilisant l'interface physique eth2 qui permet de joindre la machine 192.168.2.1 qui, elle, permet d'atteindre un réseau, le 192.168.100.0/24 :

```
/sbin/route add -net 192.168.100.0/24 gw 192.168.2.1 eth2
```

Pour plus de commodité, on peut définir des réseaux par un nom au lieu de mettre la notation CIDR qui dans l'exemple est 192.168.100.0/24. Il suffit pour cela de l'indiquer dans le fichier /etc/networks.

```
mon_reseau 192.168.100.0/24
```

Pour ajouter une route par défaut vers la machine 192.168.1.1 joignable par l'interface physique eth0 :

```
/sbin/route add default gw 192.168.1.1 eth0
```

Pour afficher la table de routage du noyau Linux, on utilise la commande `/sbin/route` sans options. L'option `-n` permet d'éviter simplement la résolution de nom DNS, ce qui est commode lorsque l'on travaille sur le réseau et qu'elle n'est pas opérationnelle.

```
/sbin/route -n
Table de routage IP du noyau
Destination     Passerelle   Genmask         ...  Iface
192.168.1.0     0.0.0.0      255.255.255.0   ...  eth0
192.168.2.0     0.0.0.0      255.255.255.0   ...  eth2
192.168.100.0   192.168.2.1  255.255.255.0   ...  eth2
127.0.0.0       0.0.0.0      255.0.0.0            lo
0.0.0.0         192.168.1.1  0.0.0.0              eth0
```

La route vers le réseau 0.0.0.0 signifie « vers n'importe quel réseau » : c'est la route par défaut ou passerelle par défaut (*route add default* ... ou celle se trouvant dans le fichier /etc/sysconfig/network). Cette ligne est évidemment la dernière consultée par le système lors d'une demande d'accès à une machine : cela signifie que l'adresse demandée ne se trouve sur aucun des sous-réseaux décrits par les routes précédentes.

L'autre route peut être ajoutée automatiquement à chaque démarrage en l'indiquant dans le fichier /etc/sysconfig/static-routes.

Les informations de routage peuvent être créées automatiquement à l'aide de démons spéciaux plutôt que de les indiquer statiquement. C'est le rôle des logiciels `routed` et `gated` qui peuvent mettre à jour dynamiquement les routes en récupérant les informations depuis le réseau. Cela permet d'utiliser automatiquement des passerelles de secours en cas de défaillance d'une des passerelles.

D. Les outils associés au réseau

Voici quelques outils permettant d'effectuer des tests de fonctionnement du réseau.

a) `ping`

Cette commande envoie un paquet ICMP (ECHO_REQUEST) à une machine et attend sa réponse (ECHO_RESPONSE). Cela permet de vérifier qu'une machine est joignable et qu'elle est capable de répondre. Si c'est bien le cas, cela signifie que sa configuration réseau et la nôtre sont correctes.

Quelques options utiles pour la commande `ping` :

- `-c <N>` : envoie N paquets et stoppe ;
- `-q` : mode « calme » (*quiet*), rien n'est affiché à part les lignes de résumé au démarrage et à la fin de l'exécution ;
- `-b` : envoie le ou les paquets à un ensemble de machines (*broadcast*).

```
ping www.transfer-tic.org
PING www.transfer-tic.org (81.80.122.16) 56(84) bytes of data.
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=1
ttl=238 time=4.42 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=2
ttl=238 time=4.29 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=3
ttl=238 time=11.6 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=4
ttl=238 time=4.20 ms
©64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=5
ttl=238 time=4.88 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=6
ttl=238 time=4.35 ms
```

```
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=7
ttl=238 time=3.81 ms
--- www.transfer-tic.org ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6060ms
rtt min/avg/max/mdev = 3.815/5.373/11.635/2.573 ms
```

b) `netstat`

Donne des informations générales sur la configuration réseau, à savoir les tables de routage, les statistiques des interfaces, etc.

Quelques options de `netstat` :

- `-n` : ne pas résoudre les noms ;
- `-r` : affiche la table de routage, équivalent à la commande `route` ;
- `-v` : mode verbeux ;
- `-l` : liste des connexions/interfaces ;
- `-c` : mise à jour permanente.

```
netstat -n
Connexions Internet actives (sans serveurs)
Proto  Recv-Q  Send-Q  Adresse locale    Adresse distante    Etat
tcp    0       0       134.157.9.32:22   134.157.9.52:46903  ESTABLISHED
tcp    0       0       134.157.9.32:22   134.157.9.52:46639  ESTABLISHED
tcp    0       0       134.157.9.32:800  134.157.9.11:2049   ESTABLISHED
tcp    0       0       134.157.9.32:643  134.157.9.11:111    TIME_WAIT
```

On remarque deux connexions sur le port 22, une sur le port 800 et une sur le port 643.

c) `arp`

Cette commande permet d'afficher la table cache de résolution d'adresses du noyau (*i.e.* l'association avec l'adresse physique MAC d'une machine) pour les machines présentes sur le même réseau.

```
arp
Address                  HWtype  HWaddress           Flags Mask  Iface
cerbere.lodyc.jussieu.f  ether   00:B0:D0:D1:8B:2A   C           eth0
nestor.lodyc.jussieu.fr  ether   00:04:76:E4:BB:33   C           eth0
```

d) `traceroute`

Affiche les différents nœuds (ou passerelles) traversés pour joindre une machine.

Quelques options de `traceroute` :

- `-n` : ne pas résoudre les noms (DNS) ;
- `-v` : mode verbeux ;
- `-f <ttl>` : change le TTL (*Time To Live*) ;
- `-w <sec>` : change le *time-out* sur les paquets retournés.

```
traceroute -n www.transfer-tic.org
traceroute to www.transfer-tic.org (81.80.122.16), 30 hops max,
38 byte packets
1  134.157.9.126     0.640 ms   0.265 ms   0.242 ms
2  134.157.247.238   5.000 ms   0.655 ms   0.545 ms
3  134.157.254.126   1.010 ms   0.830 ms   0.826 ms
4  195.221.127.181   1.656 ms   1.130 ms   1.083 ms
5  193.51.181.102    1.447 ms   1.028 ms   1.097 ms
6  193.51.180.158    1.888 ms   1.826 ms   1.771 ms
7  193.51.179.1      1.030 ms   1.443 ms   1.383 ms
8  193.51.185.1      1.815 ms   1.384 ms   1.100 ms
9  193.251.241.97    1.361 ms   1.675 ms   1.438 ms
```

La commande `traceroute` force les nœuds intermédiaires traversés à renvoyer une réponse qui est un message d'erreur (ICMP TIME_EXCEEDED), en positionnant une valeur de TTL trop basse. Dès qu'un message d'erreur est reçu, la commande incrémente cette valeur et renvoie le message ce qui lui permet de passer au nœud suivant et ainsi de suite.

L'utilitaire `tcpdump` permet de visualiser tous les échanges se produisant sur le réseau auquel votre machine est connectée.

Cette commande permet le débogage de problèmes réseau, mais aussi de récupérer certaines informations qui circulent « en clair » sur le réseau.

```
# surveillance de la machine 192.168.10.1
tcpdump src 192.168.10.1
```

E. Exercices

1. Quel est le fichier utilisé pour associer les noms symboliques et les adresses IP des machines de votre réseau ?

- ❑ /etc/nsswitch.conf
- ❑ /etc/resolv.conf

- ❑ /etc/hosts
- ❑ /etc/services

2. Quelle commande va créer une route par défaut avec comme passerelle 192.168.1.1 ?

- ❑ netstat-add default gw
- ❑ route default 192.168.1.1
- ❑ ip route default 192.168.1.1
- ❑ route add default gw 192.168.1.1
- ❑ ifconfig default gw 192.168.1.1 eth0

3. Lesquelles des commandes suivantes sont utilisées pour activer une interface réseau? (deux réponses)

- ❑ ifconfig
- ❑ netstat
- ❑ ifup
- ❑ ifstart

4. Vous soupçonnez que l'une des passerelles de votre réseau ne fonctionne plus mais vous ne savez pas laquelle. Quelle commande va vous aider à résoudre le problème ?

- ❑ ps
- ❑ netstat
- ❑ nslookup
- ❑ ifconfig
- ❑ traceroute

Chapitre 9. Services systèmes de base

Objectifs	⇒ Connaître les principaux serveurs SMTP et être capable de faire suivre les courriers (*forwarding*) et de configurer les alias. ⇒ Être capable de conserver l'heure système et de synchroniser l'horloge via le protocole NTP.
Mots clés	~/.forward, /etc/ntp.conf, date, Exim, hwclock, newaliases, mail, mailq, ntpd, ntpdate, pool.ntp.org, Postfix, Qmail, Sendmail

A. Maintien de l'horloge du système

Il existe deux types d'horloges sur l'architecture x86, une horloge matérielle et une horloge logicielle.

L'horloge matérielle, conservée par le BIOS, maintient l'heure lorsque l'ordinateur est éteint.

Lorsque le système démarre, il lit l'horloge matérielle et règle l'horloge logicielle à la valeur qu'il récupère. Il utilise ensuite l'horloge logicielle pour ses besoins et ceux de ses processus.

a) Configuration manuelle des horloges matérielle et logicielle

La commande `date` permet de gérer l'horloge logicielle du système, alors que la commande `hwclock` permet de régler l'horloge matérielle à partir de l'horloge logicielle et vice-versa.

La syntaxe de la commande `date` est :

```
date [option] … [+Format]
```

```
date [-u|--utc|--universal] [MMJJhhmm[[CC]YY][.ss]]
```

La commande `date` sans arguments permet d'afficher l'horloge logicielle :

```
$ date
lun. déc. 14 11:00:25 CET 2009
```

On peut personnaliser cet affichage par l'utilisation des options de formatage de la commande `date` :

```
$ date +"%d-%m-%Y"
14-12-2009
[zied@ankara ~]$ date +"%A,%d %B %Y"
lundi,14 décembre 2009
```

Pour modifier uniquement la date du système :

```
# date -s 01/01/2010
ven. janv.  1 00:00:00 CET 2010
```

Pour modifier uniquement l'heure du système :

```
# date -s 12:12:59
ven. janv.  1 12:12:59 CET 2010
```

Pour afficher l'horloge matérielle :

```
# hwclock -r
lun. 14 déc. 2009 12:02:30 CET  -0.504123 secondes
```

Pour mettre à jour l'horloge système par rapport à l'horloge matérielle :

```
# hwclock -s (ou bien hwclock --hctosys)
# date
lun. déc. 14 12:02:42 CET 2009
```

Pour modifier l'heure du système et affecter cette modification à l'horloge matérielle :

```
# date -s 11:00:00
lun. déc. 14 11:00:00 CET 2009
# hwclock -w
# date
lun. déc. 14 11:00:25 CET 2009
# hwclock -r (ou bien hwclock --systohs)
lun. 14 déc. 2009 11:00:42 CET  -0.895758 secondes
```

b) Le protocole NTP : *Network Time Protocol*

Maintenir une horloge précise est important sous Linux. Plusieurs services et programmes ont besoin ou tirent partie de l'horloge du système. En effet, l'horodatage est utilisé dans les journaux. Les programmes `cron` et `make` ont besoin des dates précises des modifications des fichiers. L'horodatage est également inclus dans les en-têtes des courriers électroniques. Certains protocoles d'authentification, tel que Kerberos, doivent s'assurer de la synchronisation des horloges des machines.

Le protocole NTP permet de synchroniser l'horloge d'un ordinateur avec celle d'un serveur de référence. Il crée une hiérarchie à plusieurs niveaux de sources de temps. Au sommet, une ou plusieurs sources de temps très précises telles que des horloges atomiques. Ces sources sont désignées par **strate 0**, elles sont directement reliées aux serveurs NTP de **strate 1**.

Les serveurs NTP de **strate 1** offrent le temps aux serveurs de **strate 2**, qui fournissent le temps aux serveurs **strate 3**, et ainsi de suite.

Le protocole NTP prévoit jusqu'à 16 strates, mais la plupart des clients se situent dans les strates 3 et 4.

c) Configuration de base du serveur NTP

Le paquetage NTP comporte plusieurs paquetages, notamment le démon `ntpd` et un certain nombre de programmes utilisés pour configurer et interroger le serveur NTP.

`ntpd` utilise le fichier de configuration /etc/ntp.conf qui contient plusieurs options dont les plus importantes sont :

- `restrict`, pour définir des contrôles d'accès au serveur `ntpd` ;
- `server`, pour rediriger le serveur `ntpd` vers un serveur NTP.

Voici quelques options extraites du fichier de configuration /etc/ntp.conf :

```
driftfile /var/lib/ntp/drift
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this
system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
server 3.pool.ntp.org
```

Lorsque le démon `ntpd` démarre, il contacte tous les serveurs spécifiés dans le fichier /etc/ntp.conf par l'option `server`, il compare la précision de leurs

horloges et s'ajuste par rapport à un seul serveur qui le marque comme sa source de temps primaire.

La commande `ntpq` est utilisée pour envoyer des messages de contrôle NTP à un hôte pour vérifier l'état du démon `ntpd` ou changer sa configuration. La syntaxe est la suivante :

```
ntpq [options] [host]
```

La commande `ntpq` peut être exécutée soit en mode interactif, soit avec des arguments passés à la ligne de commande.

Si l'argument `host` n'est pas spécifié, la requête est envoyée à la machine locale.

L'exemple suivant montre quatre serveurs externes connectés au serveur NTP local.

```
# ntpq -c peers
remote           refid         st t  when …  delay    …  jitter
==============================================================
196.216.249.2    146.164.48.5  2  u  243  …  486.574  …  12.956
+147.137.46.196 146.64.8.7     3  u  178  …  348.080  …  7.811
*196.7.156.83    146.64.8.7    3  u  404  …  223.383  …  5.933
ntp.dts.mg       193.50.27.66  3  u  174  …  678.084  …  42.805
```

La colonne « refid » montre le serveur sur lequel chaque système est synchronisé et la colonne « st » indique la strate des serveurs externes.

Le serveur avec lequel le serveur NTP local est synchronisé est marqué par une astérisque (*), les serveurs avec de bonnes précisions sont marqués par un signe plus (+) et les autres symboles (x ou -) désignent les serveurs rejetés pour diverses raisons.

La commande `ntpdate` est utilisée pour synchroniser l'horloge du système avec un serveur NTP.

Exemple :

```
# ntpdate -s ntp.loria.fr
```

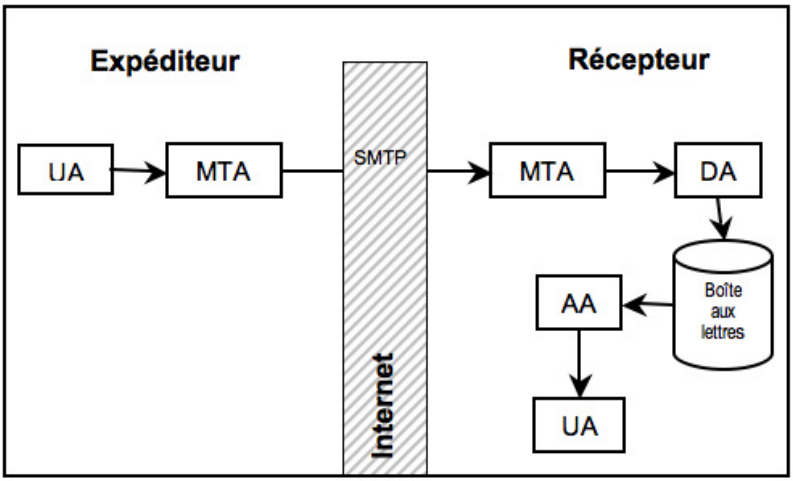
Mais la commande `ntpdate` est dépréciée, et pourrait disparaître du paquetage NTP à tout moment. À sa place on peut utiliser la commande `ntpd` avec l'option `-g`.

B. Le courrier électronique

Un système de messagerie sous Linux se compose de quatre éléments distincts (*figure 5*) :

- **MUA**, *Mail User Agent*, qui permet aux utilisateurs de lire, écrire et gérer leurs messages. Exemples de programmes MUA : Thunderbird de Mozilla, Evolution de Aka Novell, Outlook de Microsoft, la commande `/bin/mail` avec Unix et Linux ;
- **MTA**, *Mail Transport Agent* ou Agent de transport de courrier, dont le rôle est de router les messages entre les machines ;
- **MDA**, *Mail Delivery Agent*, qui permet de stocker le courrier dans la boîte aux lettres du destinataire ;
- **AA**, *Access Agent*, composant optionnel dont le rôle est de connecter le MUA à la boîte aux lettres à travers par exemple les protocoles IMAP et POP.

Figure 5. Architecture du système de messagerie



a) MTA ou Agent de transport de courrier

L'agent MTA accepte le message à partir de l'agent UA, il se charge alors de l'acheminer vers sa destination, pour cela il doit vérifier l'existence de l'expéditeur ainsi que du ou des destinataires.

Les agents MTA communiquent à travers le protocole SMTP (*Simple Mail Transport Protocol*).

Linux et Unix supportent plusieurs agents MTA, quatre agents sont les plus populaires :

- **Sendmail** : c'est le système le plus utilisé et le plus ancien. Il transmet plus de 50 % du courrier sur Internet. La configuration de Sendmail est

très complexe à cause de son fichier de configuration abscons. Sendmail est un système monolithique : il utilise un seul programme pour tous les traitements. La performance et la sécurité ont été améliorées avec les dernières versions ;

- **Postfix** : conçu comme une alternative modulaire à Sendmail, Postfix utilise plusieurs programmes qui gèrent chacun une tâche spécifique. En théorie cette conception améliore la sécurité. Postfix est rapide et plus facile à configurer que Sendmail ;
- **Exim** : les distributions Debian et Ubuntu utilisent Exim comme serveur de courrier par défaut. C'est un serveur monolithique, facile à configurer et avec un petit nombre de fichiers journaux ;
- **Qmail** : c'est un système modulaire, avec la sécurité comme objectif de conception majeur. Le code source de Qmail est disponible gratuitement mais, pour des raisons de sécurité, il est interdit de distribuer une version modifiée de Qmail sans l'accord préalable de l'auteur. Avec cette interdiction, la maintenance est devenue assez difficile. L'installation est assez compliquée, en effet il faut utiliser des *patches* pour obtenir un serveur sans bogues.

b) Gestion des courriers électroniques

La commande `mail` est un MUA permettant à l'utilisateur d'envoyer et de recevoir des courriers textuels à partir de l'invite de commandes du shell.

Par exemple l'utilisateur mejdi envoie un message à nicolas et une copie de ce message à zied :

```
[mejdi@ankara ~]$mail -s "Confirmation de réunion" -cc
zied@ankara nicolas@ankara
La réunion d'aujourd'hui est confirmée pour 15h
```

Les utilisateurs nicolas et zied peuvent ainsi consulter leurs boîtes aux lettres :

```
[zied@ankara ~]$ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/zied": 1 message 1 new
>N  1 mejdi@localhost.loca  Sun Dec 13 08:12  21/877
"Confirmation de réuni"
&
[nicolas@ankara ~]$ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/nicolas": 1 message 1 new
>N  1 mejdi@localhost.loca  Sun Dec 13 08:12  21/877
"Confirmation de réuni"
```

```
&
```

Les messages reçus sont enregistrés dans des boîtes aux lettres qui sont de simples fichiers. Ils se trouvent dans le répertoire /var/spool/mail/.

L'utilisateur peut gérer sa boîte aux lettres de façon interactive avec la commande `mail`. Il peut ainsi lire et supprimer les messages reçus ou répondre aux expéditeurs. Mais la gestion d'une boîte aux lettres avec le MUA `mail` n'est possible que sur la machine locale.

L'exemple précédent illustre la description des messages dans la boîte aux lettres de l'utilisateur : les boîtes de nicolas et de zied contiennent un message, non lu (*new*), l'expéditeur est mejdi, la date d'arrivée le 13 décembre à 08h12, le message contient 21 lignes, a une taille de 877 octets et son sujet est « Confirmation de réuni ».

Pour lire le message, on tape son numéro (ici c'est le numéro 1). On peut aussi le supprimer par la commande `d` ou bien y répondre avec la commande `r`.

Les messages en attente d'envoi sont stockés dans `/var/spool/mqueue`. Leur liste peut être affichée avec la commande `mailq`.

Le MTA Sendmail permet d'avoir plusieurs noms, ou alias (*aliases* en anglais), pour la même boîte aux lettres. On utilise pour cela le fichier /etc/aliases.

Par exemple, pour créer l'alias « nicolas.larrousse » pour un utilisateur dont le nom réel est « nicolas » :

```
nicolas.larrousse:  nicolas
```

Après avoir modifié le fichier /etc/aliases il faut lancer la commande `newaliases` pour mettre à jour la base des alias de Sendmail. Cette commande génère le fichier /etc/aliases.db qui est un fichier binaire indexé.

Un utilisateur peut rediriger les messages reçus vers la boîte aux lettres d'un autre utilisateur. Pour cela il indique l'adresse électronique de l'autre utilisateur dans le fichier `~/.forward` créé dans son répertoire personnel.

Par exemple l'utilisateur zied veut rediriger tous les messages reçus vers la boîte aux lettres niry. Il ajoute alors l'adresse électronique de la boîte niry dans le fichier ~/.forward.

```
niry@ankara
```

C. Exercices

5. Quelle commande allez-vous utiliser afin de mettre à jour l'horloge machine par rapport à l'horloge système ?

- ❑ date --sethwclock
- ❑ ntpdate
- ❑ hwclock --utc --systohc
- ❑ time --set –hw

6. Quels sont les moyens valides pour modifier l'heure de votre système ?

- ❑ ntpdate serverntp
- ❑ date -s
- ❑ date
- ❑ ntp -update

7. Salah va partir en vacances pendant deux semaines et veut que son courrier soit transmis à Ali pendant son absence. Quels changements devrait-il faire ?

- ❑ Ajouter « ali » à /etc/aliases.
- ❑ Ajouter « ali » à ~/.forward.
- ❑ Ajouter « ali » à ~/.aliases.
- ❑ Ajouter « ali » à /etc/mail.

Chapitre 10. La sécurité

Objectifs	⇒ Comprendre les risques et être capable d'appliquer les consignes de sécurité de base. ⇒ Savoir aborder la sécurité selon ses deux aspects fondamentaux : la sécurité machine et la sécurité réseau.
Points importants	Garantir la sécurité d'une machine connectée à un réseau n'est pas quelque chose d'évident. Cela dépend d'un grand nombre de paramètres qui ne sont jamais purement techniques. La sécurité du point de vue de la machine pose la question suivante : comment protéger le système contre quelqu'un qui dispose d'un accès à la machine ? En d'autres termes, le ver est déjà dans le fruit, avec soit un accès physique soit un compte qui lui permet de se connecter au système. La sécurité réseau aborde le problème d'un autre point de vue : comment nous protéger de quelqu'un qui va attaquer notre machine par le réseau (par exemple en utilisant les services que notre système offre à l'extérieur) ?
Mots clés	/etc/fstab, /etc/hosts.allow, /etc/hosts.deny, /etc/nologin, /etc/ssh_known_hosts, /etc/ssh/sshd_config, /etc/sshrc, /etc/security/access.conf, /etc/security/limits.conf, /proc/net/ip_fwchains, /proc/net/ip_fwmasquerade, /proc/net/ip_fwnames, authorized_keys, BIOS, ipchains, iptables, known_hosts, lilo, netfilter, OpenSSH, rsa, SSH, ssh-keygen, syslog, TCP wrapper

A. Les fichiers de configuration

Nous allons donner ici quelques éléments constitutifs d'une démarche de sécurisation de l'accès à notre machine.

a) Configuration du BIOS

Le cas où quelqu'un de malveillant dispose d'un accès physique à la machine est grave et quasiment insoluble. Seul le cryptage des données du système peut alors représenter une protection relativement fiable.

Il est tout de même possible de créer certaines difficultés. Le premier pas est de configurer le BIOS pour que la machine ne puisse démarrer qu'à partir du disque dur. Cela va empêcher le pirate de démarrer simplement d'une disquette ou d'un cédérom pour obtenir ensuite l'accès au système.

b) Restrictions de LILO

Le chargeur de démarrage LILO permet de passer des options au démarrage.

L'une de ces options permet de démarrer le système en « *single user mode* ». Dans certaines distributions de Linux, cette option ouvre un accès *shell* au système muni des permissions de l'utilisateur *root* sans même demander un mot de passe.

Pour éviter cela nous disposons de deux options de configuration de LILO : `restricted` et `password = '...'`. La première indique que seul l'utilisateur qui connaît le mot de passe, spécifié par la deuxième option, peut donner des options au démarrage.

c) Permissions des fichiers et répertoires

Une bonne pratique générale est de ne pas permettre à des programmes d'être exécutés à partir de /home et /tmp. Cela empêche les utilisateurs d'utiliser des logiciels qu'ils auraient installés et d'essayer d'attaquer le système par des outils téléchargés de l'Internet ou programmés par eux-mêmes.

Pour ce faire, nous pouvons utiliser les options de montage de systèmes de fichiers suivantes (/etc/fstab) :

```
/tmp /tmp ext2 nosuid 1 2
/home /home ext2 noexec 1 2
```

d) Analyser le système

Pour assurer la sécurité du système, il faut surveiller son fonctionnement. La source d'information la plus fiable pour visualiser l'activité de notre système est la consultation des fichiers de *logs*. Le démon `syslog` enregistre normalement ces informations dans /var/log/messages. Certaines distributions de Linux « loguent » l'information relative à la sécurité dans /var/log/secure : nouveaux utilisateurs ajoutés au système, échecs de connexion, etc.

La commande `w` ou `who` affiche les utilisateurs connectés au système. Elle utilise les informations du fichier de logs /var/log/utmp.

La commande `last` affiche la liste des derniers utilisateurs du système. Elle cherche ces informations dans le fichier /var/log/wtmp.

e) Des limites pour les utilisateurs

Certaines limites et restrictions peuvent être imposées aux utilisateurs.

Les restrictions d'accès pour certains utilisateurs et groupes d'utilisateurs sont configurées dans le fichier /etc/security/access.conf.

Les limitations d'utilisation sont configurées dans le fichier /etc/security/limits.conf dont chaque ligne est de la forme suivante :

```
<domaine>        <type> <élément> <valeur>
```

- `<domaine>` peut être un nom d'utilisateur, un nom de groupe d'utilisateurs, ou le signe « * » pour indiquer une entrée par défaut ;
- `<type>`, définit le type de limite à mettre en place : soit d'avertissement, avec le type `soft`, soit de blocage, avec le type `hard` ;
- `<élément>` définit la ressource à limiter pour le domaine en question (utilisateur ou groupe) : `core`, `data`, `fsize`, `memlock`, `nofile`, `rss`, `stack`, `cpu`, `nproc`, `as`, `maxlogins`, `priority`...

Le fichier /etc/nologin permet d'empêcher toute connexion au système autre que celle du super-utilisateur. Il contient le message que recevront les utilisateurs lors de la tentative de connexion, par exemple « système en maintenance ».

On peut également définir des quotas par utilisateurs (*cf.* support LPI 101). La commande `quota` (ou `repquota`) permet de vérifier l'utilisation des quotas par les utilisateurs.

B. Sécurité réseau

Comment contrôler l'accès à un serveur ? Nous pouvons le faire de deux manières différentes :

- restreindre l'accès en utilisant l'adresse de la machine qui se connecte au serveur ;
- restreindre l'accès en utilisant le port TCP ou UDP (service réseau) auquel le client essaie de se connecter.

a) TCP wrappers

Ce contrôle est assuré par la bibliothèque **libwrap**. Une grande partie des applications réseaux est compilée en utilisant cette bibliothèque. Cela permet de configurer l'accès à ces applications par deux fichiers, /etc/hosts.allow et /etc/hosts.deny. Chaque ligne de ces fichiers contient deux champs (ou éventuellement trois comme on verra un peu plus bas) séparés par le signe « : » (deux points) :

- le premier champ décrit le service pour lequel on ajoute des restrictions ou des permissions d'accès ;
- le deuxième décrit la liste des machines pour lesquelles cette règle s'applique.

Les mots-clés ALL et EXCEPT servent pour la spécification de ces deux champs.

Pour donner la liste des adresses des machines clientes on peut utiliser soit des noms de domaines commençant éventuellement par un point, soit des adresses IP se terminant éventuellement par un point. Par exemple la spécification .auf.org comprend tous les noms de domaines suffixés par « .auf.org ». De la même manière 10.1.1. donne la liste de toutes les adresses IP dans le réseau 10.1.1.0/24 (toute adresse IP commençant par 10.1.1.).

```
/etc/hosts.deny
ALL:       ALL    EXCEPT .auf.org

/etc/hosts.allow
ALL:       LOCAL  192.168.0.
in.ftpd:          ALL
sshd:     .auf.org
```

Par les « *tcp wrappers* » il est aussi possible de configurer l'exécution d'une commande. Pour faire cela on utilise le mot-clé spawn.

L'exemple suivant va garder une trace de chaque tentative de connexion à un des services de notre machine :

```
/etc/hosts.deny
ALL: ALL : spawn (/bin/echo `date` client : %c service : %d >>
/var/log/tcpwrap.log)
```

La page de manuel **host_access (5)** est une bonne source d'information sur la configuration des *tcp wrappers* et en particulier sur les macros commençant par « % ».

Il faut tout de même garder en mémoire que cette technique de contrôle d'accès n'est fiable que si elle est utilisée conjointement avec d'autres démarches de sécurisation du réseau qui vont empêcher le camouflage de l'identité de la machine cliente.

b) Filtrage de paquets

Le noyau de Linux dispose d'un filtre de paquets très puissant. Ce filtre nous permet de contrôler l'accès à notre machine par le réseau en utilisant plusieurs critères comme :

- l'adresse source de l'émetteur du paquet (l'adresse IP du client de notre serveur) ;
- le port TCP ou UDP source ;
- le port TCP ou UDP de destination (le port de notre service réseau) ;
- etc.

La commande qui permet de gérer les règles de filtrage dans les noyaux de Linux de la série 2.2 est `ipchains`. La commande correspondante dans les noyaux 2.4.x et 2.6.x est `iptables`, elle configure le filtre **netfilter**.

`Ipchains` et `iptables` utilisent trois chaînes de règles : `input`, `output` et `forward` pour `ipchains` et `INPUT`, `OUTPUT` et `FORWARD` pour `iptable`. L'utilisateur peut définir des chaînes supplémentaires pour mieux structurer son filtre.

Avec `ipchains` chaque paquet qui n'est pas destiné à la machine filtre passe par la chaîne `forward`. En revanche, avec `iptables` la chaîne `FORWARD` est traversée uniquement par des paquets qui ne proviennent pas de la machine filtre et ne lui sont pas destinés.

Les chaînes `input` et `output` (ou respectivement `INPUT` et `OUTPUT` pour `iptables`) représentent les paquets qui entrent ou sortent de la machine filtre.

Les options suivantes permettent de gérer les règles de filtrage :

- `-A` pour ajouter une nouvelle règle dans une chaîne ;
- `-D` pour supprimer une règle d'une chaîne ;
- `-P` pour modifier la politique par défaut ;
- `-I` pour insérer une règle ;
- `-F` pour effacer les règles d'une chaîne ;
- `-N` pour créer une nouvelle chaîne ;
- `-X` pour supprimer une chaîne créée par l'utilisateur ;
- `-L` pour afficher la liste des règles.

Pour stopper l'accès de toutes les machines du réseau 10.1.1.0/24 au service `ssh` de la machine filtre nous allons ajouter la règle suivante :

```
iptables -A INPUT -s 10.1.1.0/24 -p tcp --dport 22 -j DROP
```

Le grand avantage de `iptables` est la possibilité de filtrer les paquets en considérant l'état de la connexion à laquelle le paquet appartient. Cette fonctionnalité est réalisée par le module *state* d'`iptables` (« `-m state` » sur la ligne de commande pour indiquer l'utilisation de ce module). À l'aide de ce module, il est par exemple possible de savoir si un paquet appartient à une connexion déjà établie ou bien s'il s'agit d'un paquet qui essaie d'ouvrir une nouvelle connexion en relation avec une autre connexion déjà établie (exemple du fonctionnement du protocole FTP qui ouvre des connexions dynamiques). Cette fonctionnalité permet de résoudre le problème du non filtrage des connexions sortantes et de celles qui leur sont associées. Cela concerne tous les services qui ouvrent des ports dynamiques.

Pour ajouter une règle qui laisse passer tous les paquets appartenant à une connexion déjà établie (ESTABLISHED) ou à une connexion en relation (RELATED) avec une autre déjà établie :

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

L'exemple ci-dessous montre un script bash réalisant un filtrage simple sur une machine utilisée comme passerelle pour un réseau local. Aucune connexion vers cette machine n'est autorisée, toutes les connexions qui proviennent de la machine même sont acceptées et tous les paquets du réseau local sont « masqués » (MASQUERADE), c'est-à-dire que leurs adresses IP sources sont remplacées par l'adresse IP de l'interface externe de notre passerelle :

```
#!/bin/bash

I_INT=eth0 # l'interface locale
I_EXT=eth1 # l'interface externe
IP_LOCAL=10.1.1.0/24 # le réseau IP interne

# les politiques par défaut
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP


# le masquerading des adresses privées
iptables -t nat -A POSTROUTING -s $IP_LOCAL -i $I_INT -o $I_EXT
-j MASQUERADE
```

```
# accepter toute connexion initiée par le réseau local
iptables -A FORWARD -s $IP_LOCAL -i $I_INT -o $I_EXT -j ACCEPT
iptables -A FORWARD -m state -state RELATED,ESTABLISHED -j
ACCEPT


# Permettre toutes les connexions initiées par le pare-feu
iptables -A OUTPUT -j ACCEPT
iptables -A INPUT -m state -state RELATED,ESTABLISHED -j ACCEPT
```

Les informations concernant `ipfilter` sont transmises au noyau par trois fichiers situés dans l'arborescence `proc` :

- /proc/net/ip_fwchains qui contient les règles ;
- /proc/net/ip_fwnames qui contient le nom des chaînes ;
- /proc/net/ip_fwmasquerade qui contient les informations pour le « *masquerading* ».

c) Le shell sécurisé (SSH)

Le protocole SSH permet de se connecter en mode sécurisé à une machine à distance. Il effectue :

- le chiffrement de la connexion ;
- l'authentification de la machine serveur ;
- l'authentification des utilisateurs.

Il se décline en deux versions : la version 1 du protocole utilise l'algorithme de chiffrement **RSA1**, et la version 2 utilise soit l'algorithme **RSA** soit l'algorithme **DSA**.

L'authentification et le chiffrement sont réalisés sur la base d'algorithmes de chiffrement asymétriques. Une paire de « clé publique/clé privée » est générée pour le serveur et éventuellement une paire de clés est générée pour chaque utilisateur pour s'authentifier de cette manière.

Sous Linux on utilise en général OpenSSH, version libre du protocole SSH.

Par défaut la configuration de OpenSSH ainsi que les clés publique/privée sont enregistrés sous l'arborescence /etc/ssh.

La connexion SSH tient compte de la présence du fichier /etc/nologin décrit plus haut.

d) Authentification du serveur

Quand nous ouvrons une connexion SSH vers un serveur, nous devons nous assurer de l'identité de ce serveur. Pour cela, il envoie sa clé publique. Si c'est la première fois que nous nous connectons à ce serveur la question suivante apparait :

```
The authenticity of host machin (10.1.1.8)' can't be
established.
RSA key fingerprint is
8f:29:c2:b8:b5:b2:e3:e7:ec:89:80:b3:db:42:07:f4.
Are you sure you want to continue connecting (yes/no)?
```

Si nous acceptons, la clé publique sera enregistrée dans le fichier $HOME/.ssh/known_hosts.

Lors de la prochaine connexion à la même machine la question n'apparaîtra plus car l'identité de la machine sera connue.

Il faut admettre qu'accepter une clé publique envoyée par le réseau n'est pas parfaitement fiable. Il est préférable de récupérer cette clé directement sur une disquette ou une clé USB par exemple et de l'ajouter directement dans le fichier known_hosts.

Il est possible de configurer les clients SSH pour qu'ils n'enregistrent pas de clé dans ce fichier et de ne permettre la connexion qu'à des machines décrites dans le fichier /etc/ssh_known_hosts de façon à limiter les risques d'usurpation d'identité par une machine qui répondrait à la place de celle que l'on voulait contacter.

Sur le serveur, ses propres clés publiques/privées sont enregistrées par défaut dans /etc/ssh. Pour la version 1 le fichier contenant la clé privée est nommé ssh_host_key. Pour la version 2 il est nommé ssh_host_rsa_key pour l'algorithme RSA et ssh_host_dsa_key pour l'algorithme DSA.

Les fichiers contenant les clés publiques utilisent la même désignation avec l'extension .pub (par exemple ssh_host_dsa_key.pub).

e) Authentification de l'utilisateur

L'authentification de l'utilisateur peut se faire soit par mot de passe, soit, de façon beaucoup plus fiable, par une paire de clés publique et privée.

Pour une authentification par nom d'utilisateur et mot de passe l'utilisateur est invité à les entrer au clavier lors de la connexion. Le serveur effectue classiquement la vérification à partir des fichiers /etc/passwd et/ou /etc/shadow.

Pour réaliser une authentification par clés publique/privée il est nécessaire de générer ces clés. On utilise pour cela la commande ssh-keygen.

```
ssh-keygen -t rsa -b 1024
```

Par défaut cette commande enregistre les clés dans le répertoire $HOME/.ssh/ sous les noms **id_rsa** pour la clé privée et **id_rsa.pub** pour la clé publique.

Pour permettre au serveur d'authentifier un utilisateur à l'aide de sa clé publique, il faut enregistrer cette clé dans le fichier $HOME/.ssh/authorized_keys de l'utilisateur.

Il ne faut pas oublier de sécuriser les fichiers contenant les clés privées (dans les répertoires /etc/ssh et $HOME/.ssh) en leurs donnant les permissions d'accès 600.

f) Configuration de OpenSSH

La configuration du serveur sshd se fait par le fichier /etc/ssh/sshd_config.

Voici quelques options intéressantes extraites de ce fichier :

```
Port 22
Protocol 2,1
ListenAddress 0.0.0.0

# Clé d'authentification de l'hôte pour la version 2
HostKey /etc/ssh/ssh_host_key
# Clés d'authentification de l'hôte pour la version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
```

La configuration du client SSH se fait dans le fichier /etc/ssh/ssh_config génériquement pour tous les utilisateurs ou dans $HOME/.ssh/config pour une configuration spécifique pour un utilisateur.

Le fichier /etc/sshrc contient les commandes à exécuter lors de la connexion d'un client SSH, donc selon le même principe qu'un fichier /etc/profile, mais adapté aux connexions SSH.

C. Exercices

1. **Avec le protocole SSH, pour réaliser une authentification par clé publique/clé privée, l'utilisateur doit enregistrer sa clé publique RSA dans le serveur SSH dans le fichier :**
   - ❑ ~/.ssh/id_rsa
   - ❑ ~/.ssh/known_hosts

- ~/.ssh/authorized_keys
- /etc/ authorized_keys

2. **Créer un script pour la mise en place des règles de filtrage simples suivantes :**

   – les politiques par défaut sont DROP ;

   – masquer les connexions initiées par le réseau local ;

   – accepter toute connexion initiée depuis le réseau local.

3. **Installer et configurer le serveur sshd.**

   Créer une paire de clés pour un utilisateur et configurer la connexion au serveur par clés publique/privées

Annexe 1 : exemple d'examen de certification 102

Voici un exemple d'énoncé d'examen 102, suivi des réponses.

Cet exemple est destiné à vous aider à évaluer vos connaissances, et également à vous préparer au style des questions posées lors de l'examen de certification. Il est en effet nécessaire de s'habituer à la formulation des questions, qui peut parfois paraître ambiguë, ainsi qu'aux questions à choix multiples qui sont courantes dans le monde anglo-saxon.

Lors de l'examen, vous disposez d'environ une minute par question.

Questions

1. **Quelle variable d'environnement allez-vous utiliser pour afficher le code de retour de la dernière commande ?**

2. **Quel ordre SQL allez-vous utiliser pour modifier des tuples dans une base de données SQL ?**

3. **Vous essayez d'exécuter la commande ls mais il existe un alias de la commande « ls ». Quelle est la manière la plus simple pour exécuter la commande originale ls et non pas son alias ?**

4. **Donnez le nom (chemin complet) du fichier qui contient le message qui est affiché à l'utilisateur au moment de l'ouverture d'une session shell.**

5. **Quel et le nom du démon NTP utilisé pour synchroniser l'horloge du système?**

6. **Quel est le nom de l'utilitaire permettant d'ajuster les différents modes graphiques ?**

7. **Quel fichier doit modifier l'utilisateur dans son répertoire personnel pour configurer la variable d'environnement PATH ? Donnez seulement le nom du fichier, sans le chemin d'accès.**

8. **Quel est le fichier qui contient la configuration de l'environnement standard pour tout le système ? Ce fichier contient normalement la variable PATH, ainsi que umask et ulimit. Donnez le chemin d'accès complet.**

9. **Quelle commande indique au serveur X d'accepter les connexions des clients X à partir de la machine tunis ?**

10. **Quel est le chemin d'accès complet au fichier qui contient la configuration du démon utilisé pour constituer les logs du système ?**

11. **Vous voulez redémarrer le service réseau d'un serveur Red Hat. Quelle commande allez-vous exécuter pour accomplir cette opération sans avoir besoin d'utiliser un chemin absolu ?**

12. **Quelle est la section du fichier xorg.conf qui contient le chemin d'accès vers les polices de caractères ?**

13. **Quel est le protocole utilisé pour offrir une interface de connexion graphique sur un réseau TCP/IP ?**

14. **Quel fichier contient les adresses IP des serveurs DNS que la machine va utiliser pour la résolution de noms ? Donnez le chemin d'accès complet.**

15. **Quel programme vous permet d'avoir un clavier d'écran ?**

16. **Quel est le fichier de configuration du fuseau horaire du système ? Donner le chemin complet.**

17. **Quelle est la variable environnement qui permet de définir le fuseau horaire du système ?**

18. **Quelle commande allez-vous utiliser pour afficher le contenu des variables de localisation LC_* ?**

19. **Quelle commande allez-vous utiliser pour changer l'encodage d'un fichier de codage UTF-8 au codage ISO-8859-1 ? Donner seulement la commande sans options et sans arguments.**

20. **Quelle est la commande utilisée pour tester le bon fonctionnement de DNS côté client ?**

21. **Donner le nom (chemin complet) du fichier de configuration de CUPS ?**

22. **Quelle commande est équivalente à route –n ?**

23. **Vous voulez régler l'horloge du système à partir de l'horloge matérielle. Quelle commande (sans options et sans arguments) allez-vous exécuter ?**

24. **Quelle commande allez-vous utiliser pour modifier votre table de cron personnelle ? Donnez la commande la plus simple.**

25. **Quelle est la commande la plus simple qui vous permet d'afficher le contenu de votre table de cron ?**

26. **L'utilisateur nicolas de votre serveur a oublié son mot de passe. Quelle commande allez-vous utiliser pour changer son mot de passe en supposant que vous avez ouvert une session en tant que root ?**

27. **L'utilisateur mejdi a été déplacé dans le département BECO. Vous voulez changer son groupe principal en beco. Quelle est la commande la plus simple pour réaliser cela ?**

28. **Quelle commande allez-vous utiliser pour verrouiller le compte de l'utilisateur nicolas ?**

29. **Quelle est la commande la plus simple pour supprimer le compte de l'utilisateur nicolas y compris son dossier personnel ?**

30. **Vous voulez ajouter l'utilisateur mejdi à votre système en créant son dossier personnel. Donnez la commande la plus simple avec ses options et arguments.**

31. Quel est le chemin d'accès complet du fichier de configuration du démon ntpd ?

32. Dans l'ordre SQL SELECT, quelle option faut-il utiliser afin de ne conserver que des lignes distinctes ?

33. En langage SQL, pour tester l'égalité de deux chaînes de caractères, quel caractère de remplacement faut-il utiliser afin de remplacer zéro à n caractères quelconques ?

34. La requête SQL suivante permet d'afficher les livres dont le prix est supérieur à 70, en ordre décroissant de prix :

- ❑ A `SELECT * FROM Livre WHERE 'Prix' >=70 ORDER BY 'Prix' DESC;`
- ❑ B `SELECT * FROM Livre WHERE Prix >=70 ORDER BY Prix DESC;`
- ❑ C `SELECT "*" FROM Livre WHERE Prix >=70 ORDER BY Prix DESC;`
- ❑ D `SELECT * FROM Livre WHERE Prix >=70 ORDER BY Prix ASC;`

35. Votre adresse IP est 170.35.13.28 et votre masque réseau est 255.255.255.192. Quelle adresse IP N'APPARTIENT PAS à votre réseau ?

- ❑ A 170.35.13.33
- ❑ B 170.35.13.88
- ❑ C 170.35.13.62
- ❑ D 170.35.13.55

36. Quelle commande vous permet de voir l'adresse MAC et la configuration IP de votre carte réseau ?

37. Laquelle des commandes suivantes permet de synchroniser l'horloge du système avec un serveur NTP ?

- ❑ A `date -q -g`
- ❑ B `hwclock -q -g`
- ❑ C `ntpdate -s ServeuNTP`
- ❑ D `ntpd Update`

38. Vous voulez que tous vos utilisateurs BASH puissent accéder aux programmes contenus dans « /opt/bin ». Vous allez ajouter PATH=$PATH:/opt/bin; export PATH dans quel fichier ?

39. La ligne correcte de la table cron qui permet d'exécuter le script /usr/local/sbin/chklog une fois par heure entre trois heures et cinq heures de l'après-midi chaque lundi et jeudi est :

- ❑ A 0 3,4,5 * * 2,5 usr/local/sbin/chklog
- ❑ B 0 3,4,5 * * 1,4 /usr/local/sbin/chklog
- ❑ C * 15,16,17 * * 1,4 /usr/local/sbin/chklog
- ❑ D 0 15,16,17 * * 1,4 /usr/local/sbin/chklog
- ❑ E 0 15,16,17 1,4 * * /usr/local/sbin/chklog

40. À quoi sert la commande export de bash ?

- ❑ A Permettre de monter les disques à distance
- ❑ B Lancer une commande dans un sous-shell
- ❑ C Mettre l'historique des commandes à la disposition des sous-shells
- ❑ D Permettre à une variable d'être accessible dans l'environnement des processus fils
- ❑ E Partager une partition NFS avec les autres ordinateurs sur le réseau

41. Quelles sont les permissions correctes pour le fichier /etc/shadow ?

- ❑ A –rw--w--w-
- ❑ B –rwxrw-rw-
- ❑ C –rw-r--r--
- ❑ D –rw-------

42. Quelle commande du bash vous empêche d'écraser un fichier avec « > » ou « >> » ?

- ❑ A set -o safe
- ❑ B set -o noglob
- ❑ C set -o noclobber

- ❑ D set -o append
- ❑ E set -o nooverwrite

43. Vous venez d'installer un système et vous voulez vous assurer que chaque utilisateur créé aura un sous-dossier bin/ dans son répertoire personnel. Dans quel répertoire allez-vous mettre le répertoire bin/ pour permettre sa création automatique au moment de l'ajout d'un nouvel utilisateur ?

44. Lesquels de ces deux fichiers dans le répertoire personnel de l'utilisateur sont utilisés pour configurer l'environnement bash ?

- ❑ A bash et .bashrc
- ❑ B bashrc et bash_conf
- ❑ C bashrc et bashprofile
- ❑ D .bashrc et .bash_profile
- ❑ E bash.conf et .bash_profile

45. Quel est le fichier dont le contenu est affiché pour des utilisateurs qui ouvrent une session localement sur la machine AVANT l'ouverture de la session ?

- ❑ A /etc/issue
- ❑ B /etc/issue.net
- ❑ C /etc/motd
- ❑ D /etc/local.banner

46. Dans la liste suivante, quels sont les « Window Manager » (plusieurs réponses) ?

- ❑ A WindowMaker.
- ❑ B KDM.
- ❑ C Xwindow.
- ❑ D twm.

47. Que va faire la commande suivante: cat hosts | lpr -#2

- ❑ A Imprimer le fichier hosts sur l'imprimante par défaut deux fois
- ❑ B Classer hosts et imprimer le classement comme tâche #2

- ❑ C Envoyer le fichier hosts à l'imprimante et le mettre dans la queue numéro 2.
- ❑ D Envoyer le fichier hosts sur la sortie standard puis envoyer la tâche en cours à l'imprimante 2.

48. Quelle ligne de la table du cron permettrait la mise à jour régulière de la date du système à partir d'un serveur de temps ?

- ❑ A 10 * * * date dt$24
- ❑ B 10 * * * settime dt$24
- ❑ C 10 * * * date<ntpl.digex.net
- ❑ D 10 * * * /usr/sbin/runcron date <ntpl.digex.net
- ❑ E 10 * * * /usr/sbin/ntpdate ntp1.digex.net \ >/dev/null 2>&1

49. Quelle commande utilisez-vous pour suspendre ou mettre en attente une queue d'impression ?

- ❑ A lpr
- ❑ B lpq
- ❑ C lpc
- ❑ D lpd
- ❑ E lprm

50. Que contient le fichier xorg.conf (plusieurs réponses) ?

- ❑ A La résolution de l'écran.
- ❑ B Le(s) chemin(s) pour trouver les polices de caractères.
- ❑ C La taille du moniteur.
- ❑ D Le nom du Window Manager à lancer

51. Vous avez décidé de basculer vos mots de passe de type standard vers des mots de passe de type MD5. Après avoir configuré les fichiers contenus dans /etc/pam.d vous devez également faire les opérations suivantes :

- ❑ A Rien, les mots de passe seront modifiés au moment de l'ouverture d'une nouvelle session par l'utilisateur
- ❑ B Rien, les utilisateurs seront avertis automatiquement de changer leur mot de passe au moment de l'ouverture d'une nouvelle session.

- ❑ C Vous devez ré-entrer un à un tous les mots de passe avec la commande passwd
- ❑ D Vous devez supprimer et recréer tous les utilisateurs
- ❑ E Vous devez revenir vers la configuration précédente de /etc/pam.d pour réinitialiser les mots de passe en MD5.

52. Vous êtes en train de vérifier la sécurité de votre système et vous vous apercevez que la plupart des enregistrements dans /etc/passwd ont des « x » dans le champ du mot de passe et que certains ont des longueurs de 13 caractères. Que faites-vous dans ce cas ?

- ❑ A Rien, les utilisateurs avec « x » comme mot de passe sont bloqués.
- ❑ B Vous utilisez la commande pwconv pour convertir les mots de passe unix standard vers des mots de passe shadow.
- ❑ C Vous utilisez la commande passwd pour créer des mots de passe shadow aux utilisateurs ayant des mots de passe unix standard.
- ❑ D Vous utilisez la commande passwd pour créer des mots de passe aux utilisateurs ayant des « x » comme mot de passe.

53. Dans quel fichier configurez vous les alias du shell pour tous les utilisateurs ?

- ❑ A /etc/bashrc
- ❑ B /etc/profile
- ❑ C ~/.bash_profile
- ❑ D /etc/skel/.bashrc
- ❑ E /etc/skel/.bash_profile

54. Avec le protocole SSH, pour réaliser une authentification par clé publique / clé privée, l'utilisateur doit enregistrer sa clé publique RSA dans le serveur SSH dans le fichier :

- ❑ A ~/.ssh/id_rsa
- ❑ B ~/.ssh/known_hosts
- ❑ C ~/.ssh/authorized_keys
- ❑ D /etc/ authorized_keys

55. Vous êtes en train de configurer un routeur mais les réseaux connectés à ce routeur ne parviennent pas à communiquer entre eux. Finalement vous en déduisez que le problème provient du fait que le « *forwarding* » n'est pas activé sur votre routeur. Pour vérifier cela vous utilisez la commande

#cat /proc/sys/net/ipv4/_____

56. La commande netstat -a stoppe un long moment sans rien afficher à l'écran. Vous soupçonnez le problème suivant :

- ❑ A problème avec NTP
- ❑ B problème avec le DNS
- ❑ C problème avec SMTP
- ❑ D problème de routage
- ❑ E le démon netstat ne fonctionne plus

57. Après avoir modifié le fichier /etc/aliases, quelle commande faut il exécuter pour mettre à jour le fichier binaire indexé /etc/aliases.db :

- ❑ A. aliases
- ❑ B. forward
- ❑ C. newaliases
- ❑ D mailq

58. Vous soupçonnez qu'une passerelle de votre réseau ne fonctionne plus mais vous ne savez pas laquelle. Quelle commande va vous aider à résoudre le problème ?

- ❑ A ps
- ❑ B netstat
- ❑ C nslookup
- ❑ D ifconfig
- ❑ E traceroute

59. Quelle commande va créer une route par défaut avec comme passerelle 192.168.1.1 ?

- ❑ A netstat-add default gw
- ❑ B route default 192.168.1.1

- ❑ C ip route default 192.168.1.1
- ❑ D route add default gw 192.168.1.1
- ❑ E ifconfig default gw 192.168.1.1

60. _____ est utilisé par la machine pour identifier quelle machine se trouve sur le même réseau et quelle machine est sur un autre réseau.

- ❑ A DNS
- ❑ B ARP
- ❑ C La passerelle
- ❑ D Le masque réseau
- ❑ E Le protocole de routage

Réponses

1. ?
2. UPDATE
3. \ls
4. /etc/motd
5. ntpd
6. xvidtune
7. .bash_profile ou .profile
8. /etc/profile
9. xhost +tunis
10. /etc/syslog.conf
11. service network restart
12. Files
13. XDMCP
14. /etc/resolv.conf
15. GOK
16. /etc/localtime
17. TZ

18. locale
19. iconv
20. dig ou host
21. /etc/cups/cupsd.conf
22. netstat -nr
23. uname –r (alternative: uname –a)
24. crontab -e
25. crontab -l
26. passwd nicolas
27. usermod -g beco mejdi
28. passwd -l nicolas (alternative: usermod -L nicolas)
29. userdel -r nicolas
30. useradd -m medji
31. /etc/ntp.conf
32. DISTINCT
33. %
34. B
35. B
36. ifconfig
37. C
38. /etc/profile
39. D
40. D
41. D
42. C
43. /etc/skel
44. D
45. A
46. A, D

47. A
48. E
49. C
50. A, B
51. C
52. B
53. A
54. C
55. ip_forward
56. B
57. C
58. E
59. D
60. D

Index des mots clés

L

M

N

O

P

Q

R

S

T

U

W

X

Table des figures et des tableaux

Les auteurs

Zied Bouziri (Tunisie) : enseignant depuis 2003 au département Informatique de l'Institut supérieur des études technologiques de Charguia (ISET, Tunis), option réseaux informatiques. Il est ingénieur en informatique, diplomé de l'École nationale des sciences de l'informatique de Tunis (ENSI). Entre 1999 et 2003, il a été ingénieur conception et développement au département recherche et développement chez Alcatel.

Niry H. Andriambelo (Madagascar) : membre fondatrice de l'Association malagasy des utilisateurs de logiciels libres et formatrice de formateurs GNU/Linux, Niry Andriambelo est ingénieur informatique diplômée de l'Université d'Antananarivo et, depuis 2004, coordinatrice pour les systèmes et réseaux universitaires francophones dans l'océan Indien.

Andrei Boyanov (Bulgarie) : directeur d'Active Solutions et membre de la commission technique et développement de l'Institut professionnel Linux. Andrei Boyanov est ingénieur en informatique issu de l'Universtié technique de Sofia et formateur des personnels d'encadrement de l'enseignement supérieur dans le domaine des systèmes et des réseaux Linux.

Nicolas Larrousse (France) : concepteur de programmes Transfer dans les systèmes et réseaux sous Linux ainsi que de formations à la certification LPI. Nicolas Larrousse est ingénieur informatique. Il enseigne les systèmes d'information à l'Université de Versailles et exerce, depuis 1992, au Centre national de la recherche scientifique (CNRS), à Paris.

Véronique Pierre (France) : consultante indépendante en édition scientifique multimédia.