

Administration réseau local

Table des matières

ENVIRONNEMENT	3
CHOIX DU MATÉRIEL	3
CHOIX DU SYSTÈME D'EXPLOITATION	3
VIRTUALISATION	4
<i>Network</i>	5
INSTALLATION	8
TYPE D'INSTALLATION	8
AUTOMATISATION	8
<i>Kickstart</i>	8
LANGUAGE	8
KEYBOARD	9
DISK	9
CHOIX DES PACKAGES LOGICIELS	9
<i>Ajout des outils de développements</i>	9
PROGRAM.....	9
POST INSTALLATION.....	10
NTP	10
NO USER ADDED	10
SELINUX ENABLED (MODE ENFORCING)	10
ANACONDA-KS.CFG (FOR FURTHER REINSTALLATION)	10
RUN LEVEL : SWITCH FROM 5 TO 3	11
<i>/etc/inittab</i>	11
<i>telinit 3</i>	11
DISK / PARTITION MANAGEMENT	12
FDISK	12
MKE2FS	12
MOUNT	12
POLICIES	12
FILESYSTEM HIERARCHY STANDARD	12
USER MANAGEMENT.....	13
POLICY FOR USER DIRECTORIES	13
USERADD, USERDEL, GROUPADD, GROUPDEL	13
SU, SUDO, VISUDO	14
DISK SPACE MANAGEMENT	16
<i>du, df</i>	16
<i>quota</i>	16
STARTUP AND CONFIGURATION FILES	18
/ETC/INITTAB (RUN LEVEL & INIT SCRIPT)	18
/ETC/RC.D/RC.SYSINIT (← /ETC/RC.SYS.INIT)	19
/ETC/SYSTEM/SYSINIT	19
. /ETC/SYSCONFIG/NETWORK (EXPLAIN DOT)	19
/ETC/RC.D/RC\$RUNLEVEL.D	20

/ETC/RC.LOCAL	20
/ETC/RC\$RUNLEVEL.D/K* AND S*	20
/ETC/INIT.D/SERVICENAME	21
(NETWORK) SERVICES AND CONFIGURATION FILES	21
/etc/sysconfig	21
/etc/sysconfig/network-scripts/ifcfg-eth0	22
/ETC/INIT.D/NETWORK.....	22
/ETC/INIT.D/NFS.....	22
OTHER LESS USED DAEMONS FROM /ETC/INIT.D/XINETD	22
Missing from the base install	23
Yum install xinetd	23
CONFIGS IN /ETC/XINETD.CONF AND /ETC/XINETD.D/	23
Example rsync	23
NFS	23
File /etc/exports	23
/etc/init.d/nfs start	23
/sbin/services nfs start	23
/sbin/chkconfig --level 345 nfs on	24
CLIENT SYSTEM	24
IPLCLINT01 INSTALLATION.....	24
FEDORA CORE 13	24
AVOID EXPRESS INSTALLATION (KEYBOARD ...)	24
CREATE « LOCAL » LOGIN MANDATORY	24
NFS SERVER W/ CLIENT.....	24
CLIENT CANNOT CONNECT TO SERVER	24
FIREWALL.....	24
Disable by chkconfig --level 345 iptables off.....	27
PROBLEMS WITH SELINUX (ENFORCING → PERMISSIVE – SOLVED !)	27
Setenforce permissive	28
Upfate /etc/sysconfig/selinux	28
UPDATE /ETC/FSTAB.....	28
USERDEL –R NINA	28
LSOF DIRECTORY – NIS	28
NIS – NETWORK INFORMATION SERVICE	28
YP – YELLOW PAGE (PROTECTED TRADEMARK).....	28
ARCHITECTURE	28
Server.....	28
MASTER – SLAVE	28
DOMAINE.....	28
SERVER IPLSRV01	28
CLIENT IPLCLNT01	29
DIRECTORY OPENLDAP.....	29
DÉFINITION.....	29
ATTRIBUTS	30
COMMANDES	31
SAMBA.....	31
SERVEUR WEB APACHE	32
FICHIERS DE CONFIGURATION	32
PARE-FEU	32

Environnement

Choix du matériel

Le serveur est-il destiné à gérer plusieurs utilisateurs ou est-il utilisé dans la virtualisation ?

Si c'est le cas en fonction du nombre d'utilisateurs qui se connectent au serveur, avoir beaucoup de RAM est une bonne idée, je recommande au moins 12 à 15 Go selon la configuration du serveur et le niveau de virtualisation du serveur.

Est-ce que le serveur va effectuer des opérations complexes tels que des calculs complexes / compiler / faire du rendering ?

Si tel est le cas, je recommande au moins 12 à 24G de RAM et un processeur multi-cœur rapide, voir même un serveur possédant plusieurs processeurs.

Mettre en place un système de redondance ?

RAID¹ est crucial dans un serveur. En cas de défaillance du disque dur, cela pourrait coûter des centaines ou des milliers de dollars pour restaurer le serveur à son état précédent, à l'aide d'un swap disque RAID simple le serveur est de retour à son état précédent.

La solution la plus optimale pour un contrôleur RAID dans un environnement serveur est RAID 5 ou RAID 1. Un minimum de 3 disques est nécessaire pour le RAID 5 et au moins 2 disques dans une configuration RAID 1.

Recommandation : un contrôleur RAID hardware plutôt qu'un RAID software meilleur marché. Avoir un contrôleur hardware allège le CPU en particulier en cas de défaillance du disque dur, ce qui peut causer d'importants ralentissements du système s'il ne s'agit pas d'un contrôleur hardware.

Attention : « Mission critique » fait référence à n'importe quel facteur d'un système (équipements, processus, procédures, logiciels,...) dont la panne résulterait en l'échec d'opérations business. Ce qui veut dire que c'est critique pour l'organisation de l'entreprise.

Est considéré comme « Mission critique » toutes applications ou toutes plates-formes dont une interruption du service impacte directement un ou plusieurs des éléments suivants : le chiffre d'affaires, l'image de marque de l'entreprise, sa capacité à respecter ses obligations réglementaires. Ces systèmes ont généralement des exigences fortes en termes de : disponibilités, montée en charges, performance, sécurité ou sûreté, manageabilité, fiabilité.

Choix du système d'exploitation

Red-Hat Entreprise Linux (RHEL)

Distribution Linux produite par Red Hat et orientée vers le marché commercial et les serveurs d'entreprise. Red Hat prend en charge chaque version pour une durée de 7 à 10 ans après sa sortie. Toutes les formations et certifications Red Hat – RHCT, RHCE, RHCSS et RHCA – pour le déploiement de matériaux et de logiciels portent sur la plate-forme RHEL.

De nouvelles versions de RHL sont livrées tous les 18 à 24 mois. Quand Red Hat fournit une nouvelle versions, les clients peuvent mettre à jour leur version gratuitement à condition d'avoir un abonnement en cours.

Toutefois, les restrictions sur la marque déposée ne permettent pas la copie et la redistribution de la distribution complète.

¹ Redundant Array of Inexpensive Disks

CentOS

Très populaire parce qu'il s'agit essentiellement d'un clone de RHEL, mais toutes références propriétaires de Red Hat (nom, logo et images) ont été retirées. En fait, c'est un RHEL dont les sources ont été recompilées. Bien sûr, aucune mise à jour de Red Hat ni de support technique.

Debian

Organisation communautaire et démocratique dont le but est le développement d'un système d'exploitation basé exclusivement sur des logiciels libres.

Ubuntu

Distribution la plus populaire aujourd'hui. Elle satisfait aussi bien les nouveaux utilisateurs que les plus expérimentés. Il existe des versions adaptées aux différents supports disponibles (netbook, ordinateur, serveur, cloud). Cette distribution possède une forte communauté qui apporte des améliorations tous les jours.

Fedora

Distribution qui se veut être un système d'exploitation complet et généraliste composé uniquement de logiciels libres. Fedora dérive de la distribution RHEL et est destinée à la remplacer pour les utilisateurs finaux (utilisation non commerciale).

Le maintien de Fedora provient en grande partie de sa communauté d'utilisateurs. Les développements sont essentiellement tournés vers les nouveautés, cela permet à Red Hat un bon retour d'expériences.

Et d'autres ...

Debian et Fedora (anciennement Fedora Core) sont des distributions qui évoluent très vite. Pour le cours, nous avons utilisé CentOS 5.5.

Pour un usage professionnel, il est plus intéressant de se tourner vers une distribution professionnelle comme RHEL qui fournit un support technique payant et prend en charge chaque version du logiciel pour une durée de 7 à 10 ans après sa sortie.

Attention : Il est important de se poser la question : « Est-ce que ma version de Linux est supportée pour mon matériel ? » Il faut donc choisir sa distribution en fonction de son matériel.

Virtualisation

Quelle est l'utilité de la virtualisation ?

1. Optimisation de l'infrastructure

La virtualisation est une réponse à la prolifération des serveurs dans les centres de données. Nombre de ces serveurs utilisent à peine 15 % de la capacité et de la puissance de calcul disponibles. La virtualisation d'un serveur consiste à fractionner un serveur physique en différentes machines virtuelles se comportant chacune comme un serveur indépendant à part entière. Le matériel est ainsi exploité plus efficacement, et les serveurs peuvent être dotés facilement de davantage de ressources ; les applications fonctionnent indépendamment du matériel sur lequel elles tournent. L'ancien matériel (qui travaille moins efficacement ou qui présente davantage de risques de panne) peut ainsi être mis hors service. La virtualisation est également utile pour les scénarios anti sinistres (disaster recovery) et pour le groupage de serveurs.

2. Economiser

En réduisant le nombre de serveurs, les coûts de matériel sont limités, mais il y a plus de coûts d'énergie et engendre des problèmes de refroidissement - souvent des points épineux pour les centres de données de toutes tailles. Le groupage permet par ailleurs aux centres

de données plus modestes de se développer sur une surface moindre, et de limiter ainsi les frais de gestion. Un bénéfice dont profite bien entendu aussi le client !

Quand optez-vous pour la virtualisation ?

Bien que la virtualisation soit une technologie très prometteuse, elle ne convient pas encore pour toutes les applications. La virtualisation peut, entre autres, être utilisée pour les sites web, les applications web, les serveurs 'active directory', les serveurs 'terminal services', etc.

Attention : La virtualisation convient par contre moins pour les applications faisant intensivement appel aux CPU et Disk I/O. Les serveurs de base de données ou les serveurs Exchange très chargés et comportant des centaines de boîte de messagerie seront donc de préférence placés sur des serveurs physiques.

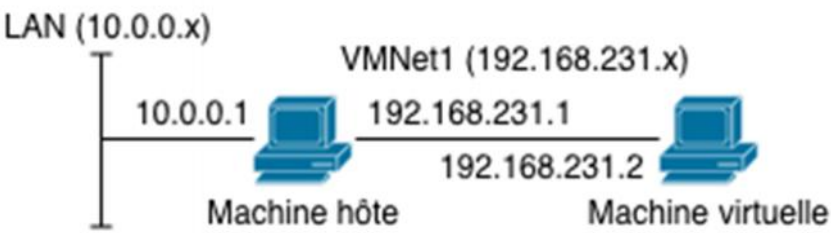

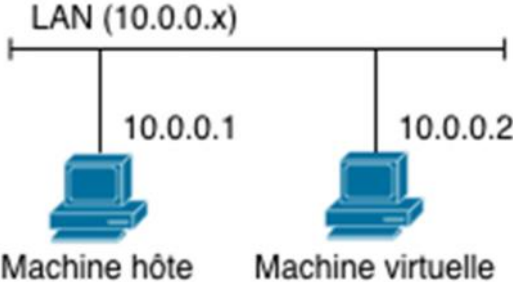
Quels sont les avantages et les inconvénients ?

Avantages	Inconvénients
Economie de coût	Panne Solution : <ul style="list-style-type: none"> • Contrat avec des SLA pour la maintenance et pour une durée maximum • Réplication des données • Distribution des charges
Achète moins de serveurs	
Sur un même matériel, permet de répartir les charges	
Existe que de manière logique. Backup facile puisqu'il y a qu'un seul fichier sur le serveur	

Network

Les types de réseau VMWare

	Machine virtuelle	
	Accès au LAN	Adresse IP de LAN
Host-only	NON	NON
NAT	OUI	NON
Bridged	OUI	OUI

Host-Only	 <p>La machine virtuelle a accès uniquement à la machine hôte sur un réseau privé virtuel (VMNetX). Vu du LAN, il n'y a aucune nouvelle machine. La machine hôte fait office de serveur DHCP pour le réseau VMNet1.</p> <p>Cela peut être utilisé pour créer un réseau contenant l'hôte et un ensemble de machines virtuelles, sans avoir besoin de l'interface réseau physique de l'hôte. Au lieu de cela, une interface réseau virtuelle (similaire à une interface loopback) est créé sur l'hôte, fournissant la connectivité entre les machines virtuelles et l'hôte.</p>
Shared Ethernet - NAT	 <p>La machine virtuelle a accès au LAN à travers la machine hôte par un routage de type NAT². Vu du LAN, il n'y a aucune nouvelle machine. La machine virtuelle envoie ses requêtes sur le LAN en utilisant l'adresse IP de la machine hôte. Nécessite un LAN opérationnel et connecté. La machine hôte fait office de serveur DHCP et NAT pour le réseau VMNet8.</p>
Bridge	 <p>La machine virtuelle a accès directement au LAN. Vu du LAN, il y a une nouvelle machine avec sa propre adresse IP. Nécessite un LAN opérationnel et connecté. La machine virtuelle utilise le serveur DHCP du LAN (si présent).</p> <p>C'est pour des besoins plus avancés de réseau tels que des simulations de réseau et des serveurs tournant dans des espaces invités. Lorsqu'elle est</p>

² Network Address Translation

	activé, VirtualBox se connecte à une vos cartes réseaux installées et échange les paquets directement sur le réseau, en contournant la pile réseau de votre système d'exploitation.
--	---

Shared Ethernet

Tous les utilisateurs du réseau ont les mêmes droits sous Ethernet. Tous les utilisateurs peuvent échanger des données de toutes les tailles avec un autre utilisateur à tout moment.

Parce qu'Ethernet a été conçu comme un système de bus logique, n'importe quel périphérique réseau qui est en train de transmettre est entendu par tous les autres utilisateurs. Chaque utilisateur Ethernet filtre les paquets de données qui lui sont destinés à partir du flux, en ignorant tous les autres.

Adresse Ethernet = adresse MAC = (adresse physique / matérielle)

Une adresse MAC³ est un identifiant physique stocké dans une carte réseau ou une interface réseau similaire et utilisé pour attribuer mondialement une adresse unique au niveau de la couche de liaison (couche 2 du modèle OSI). C'est la partie inférieure de celle-ci qui s'occupe d'insérer et de traiter ces adresses au sein des trames transmises. Elle est parfois appelée adresse Ethernet.

L'Address resolution protocol (ARP, protocole de résolution d'adresse) est un protocole effectuant la traduction d'une adresse de protocole de couche réseau (typiquement une adresse IPv4) en une adresse MAC (typiquement une adresse Ethernet), ou même de tout matériel de couche de liaison. Il se situe à l'interface entre la couche réseau (couche 3 du modèle OSI) et la couche de liaison (couche 2 du modèle OSI).

- ⇒ **Un contrôleur Ethernet peut répondre à plusieurs adresses.**
- ⇒ Adresse MAC/Ethernet : 48 bits (6 octets)
- ⇒ Adresse IPv4 : 32 bits

Bridged Ethernet

Cf. Les types de réseau VMWare

Host-Only

Cf. Les types de réseau VMWare

Configuration de notre serveur

Le nom du serveur doit être choisi de manière judicieusement :

- Soit un nom intelligent : iplserv001, iplclient001, ... Il faut cependant décrire la règle utilisée, c'est fondamentale ! Mais attention aux risques : si un attaquant repère un serveur iplservxxx il saura quelle cible il attaque et l'attaquera plus vite que s'il lit milou ou rouge, ...
- Soit un nom sans signification : Milou, Black, Tintin, ...
- Soit un nom sans signification à l'extérieur et intelligent en interne

Quant à la configuration :

- Séparer l'OS du core fichier. Il faut donc 2 disques !
- Disque de 16G + Disque supplémentaire de 8G
- RAM : 1024MB
- 2CPUs

³ Mac Access Control

Installation

Type d'installation

3 types d'installation sont proposées :

- ⇒ Desktop Gnome ;
- ⇒ Desktop KDE ;
- ⇒ Server

Une installation est appelée **Desktop** quand elle est destinée à un PC d'utilisateur. Elle dispose d'une interface graphique et de logiciels type bureautique.

Une installation est appelée **Server** quand elle est destinée à un PC serveur. En général, elle ne dispose pas d'une interface graphique (→ Pas de programmes inutiles).

Automatisation

Kickstart

La version 5 de RHEL est livrée avec un utilitaire peu connu (et jusqu'à aujourd'hui quasiment pas documenté) appelé KickStart. Il vous permet d'automatiser (presque) toute l'installation d'une distribution RHEL et notamment:

- ⇒ la sélection de la langue;
- ⇒ la configuration réseau et la sélection des sources de la distribution;
- ⇒ la sélection du clavier;
- ⇒ l'installation de l'utilitaire de démarrage (ex: lilo);
- ⇒ le partitionnement du disque et la création du système de fichiers;
- ⇒ la sélection de la souris;
- ⇒ la configuration du serveur X-Window;
- ⇒ la sélection de la zone géographique;
- ⇒ la sélection du mot de passe de l'utilisateur root;
- ⇒ la sélection des paquetages à installer.

Il s'agit des principales étapes de l'installation manuelle d'une distribution RedHat. KickStart vous permet d'automatiser le processus d'installation en plaçant les informations que vous rentreriez normalement au clavier dans un fichier de configuration.

Mais attendez, il y a mieux!

Une fois le processus d'installation achevé, KickStart vous permet de spécifier une liste de commandes shell que vous souhaitez voir exécutées. Cela signifie que vous pouvez automatiquement installer des logiciels locaux qui ne font pas partie de la distribution RedHat et procéder aux derniers réglages nécessaires pour rendre votre nouveau système d'exploitation parfaitement opérationnel.

Il y a deux manières d'utiliser KickStart :

- ⇒ La première est de copier le fichier de configuration de KickStart sur une disquette d'amorçage RedHat.
- ⇒ La seconde est d'utiliser une disquette d'amorçage classique et de récupérer ce fichier de configuration via le réseau.

Language

Toujours installer sa distribution de Linux en anglais pour une solution professionnelle.

- ⇒ Meilleur support (la communauté anglophone est plus grande, plus de réponses lorsque les messages d'erreurs sont en anglais)
- ⇒ Pas de problèmes d'accents

Keyboard

⇒ Be-latin 1

Disk

1ère partition = root	<i>/ : Fichiers système dans /etc</i>	1.0 GB	disk 1
2ème partition	/usr	8.0 GB	
3ème partition	/usr/local ou /opt : Les logiciels extérieurs doivent y être installés	restant	
4ème partition	/var : Fichiers de configuration du système	2.0 GB	
5ème partition	swap : Taille du swap = 2 * taille de la RAM	2.0 GB	
6ème partition	/disks/home	2.0 GB	disk 2
7ème partition	/disks/share	restant	

Il faut donc toujours réfléchir à son partitionnement !

Sur un serveur il faut toujours séparer l'OS (disque 1) du corps business (disque 2) → 2 disques physiques différents (sinon ça n'a pas de sens).

/var/spool/mail contient les mails d'un serveur de mails

- ⇒ S'il n'y a qu'une seule partition sur le serveur et qu'on envoie plein de mails, c'est l'entière du serveur qui plante (et pas juste la partition /var) !
- ⇒ Si le **/var** est bourré et qu'il est sur une partition séparée on a toujours accès au /home
- ⇒ Si le **/var** est bourré il n'y a plus aucun log (on peut donc hacker le serveur sans laisser de traces) → D'où l'utilité d'utiliser une partition pour éviter ça !

/var/log contient tous les logs

- ⇒ **/var/log/syslog** contient les logs systèmes archivés entre 6 mois et 2 ans.

System clock uses UTC pour passer à l'heure d'été/hiver automatiquement.

Choix des packages logiciels

- ⇒ Serveur Apache, MySQL, SSH, ...
- ⇒ Encryption

Ajout des outils de développements

Program

1. Basic Installation
2. Local system – Users – Management
3. Systems in network – NFS
4. Network directories
5. Windows
6. Security – Web servers

Post installation

NTP

Il est important d'établir une date et une heure précise pour les logs du système. Une excellente habitude est d'utiliser NTP qui permet de mettre à jour l'heure via l'horloge atomique. Exemples de serveurs NTP : ntp.centos.org / ntp.belnet.be

No user added

Après l'installation (lors du 1^{er} démarrage du serveur) il n'y a qu'un seul utilisateur : « root », il faut donc créer au moins un compte utilisateur non-administrateur.

SELinux enabled (mode enforcing)

Security Enhanced **ON**

Il y a 2 modes :

- Enforcing (sécurisé)
- Permissive (non sécurisé, s'il y a violation laisse faire et on la log)

SELinux⁴ est un modèle de sécurité que l'on peut ajouter au système standard de Linux afin d'augmenter la sécurité face aux diverses attaques que subit le système. On peut configurer les accès de chaque processus pour les restreindre à un strict minimum. Cela permet de rendre inexploitable certaines failles et, en cas de piratage, de limiter l'étendue des dégâts.

- **/usr/sbin/setenforce 0** : Désactive SELinux (le mode passe à permissive).
- **/usr/sbin/getenforce** : Commande qui renvoi « Permissive » quand SELinux est activé, mais la politique de SELinux n'est pas « Enforced » et seules les règles DAC sont utilisées. La commande **getenforce** renvoie « Disabled » si SELinux est désactivé.

2 modes principaux:

- **DAC⁵** : 666 c'est le créateur des objets qui choisit, il fait ce qu'il veut.
- **MAC⁶** : Les décisions de protection ne doivent pas être prises par le propriétaire des objets concernés.

Anaconda-ks.cfg (for further reinstallation)

/root/anaconda-ks.cfg

KickStart est la méthode d'installation automatique de RedHat. Elle est utilisée aussi pour des installations semi-automatiques. Sa configuration repose sur un unique fichier qui peut être écrit à la main, en partant de zéro ou bien à partir du fichier **anaconda-ks.cfg** que l'installation manuelle aura généré pour vous dans le répertoire **/root**.

De plus, CentOS (comme aussi des cousines RedHat, Fedora et ScientificLinux) dispose d'un outil graphique pour générer des fichiers KickStart (-system-config-kickstart).

Générer un fichier kickstart : Partitionnement des disques afin d'éviter la réinstallation infinie des noeuds, avec KickStart il faut rajouter un script de post-installation, qui désactive le démarrage depuis le réseau en renommant le fichier correspondant à l'adresse IP du client.

⁴ Security-Enhanced Linux

⁵ Discretionary Access Control

⁶ Mandatory Access Control

➔ Lors de la première installation de CentOS un fichier `anaconda-ks.cfg` est créé, ce fichier contient les informations de configuration de la première installation pour pouvoir la reproduire de manière identiques sur toutes les autres machines. L'installation devient donc automatique, ce qui fait gagner beaucoup de temps.

Ex : conserver les services, partitions, infos réseau,...

Run level : switch from 5 to 3

Le choix du démarrage du système Linux dans un niveau ou dans l'autre s'effectue avec le fichier de configuration « `/etc/inittab` », et consiste à choisir le niveau d'exécution.

Il existe différents niveaux d'exécution (**run level**) de Linux :

0	« Halt » fermer le système
1	Mono utilisateur
2	Multi utilisateur sans réseau NFS
3	Multi utilisateur avec réseau NFS (pas d'environnement graphique)
4	Inutilisé
5	Interface graphique, xdm
6	« reboot »

Un bon serveur est Level 3 !

Comment changer le run level d'un serveur ?

```
# ssh 10.0.0.192           // Se connecte sur le serveur

# vi /etc/hosts            // Ajoute son adresse MAC pour se connecter plus vite

# vi /root/anaconda-ks.cfg // Informations sur le serveur

# vi /etc/inittab          // Edite pour mettre en level 3

# telinit 3                // Passe le serveur en level 3 sans devoir redémarrer
```

/etc/inittab

Le fichier contient une suite d'instructions sous la forme :
code:niveau_d'action:action:commande

```
# Le niveau d'exécution par défaut
id:3:initdefault:
```

⇒ Le run level est de 3

telinit 3

On peut changer de niveau d'exécution grâce à la commande `telinit` (qui n'est qu'un lien sur `init`) :

telinit 2 placera le système en Mode multi-utilisateurs sans réseau. Bien entendu, seul l'administrateur système (root) peut exécuter cette commande...

Disk / Partition Management

fdisk

- L'utilitaire **fdisk** permet de créer des partitions sur un disque dur.
- Le partitionnement avec fdisk peut entraîner la perte de toutes les données présentes sur le disque sur lequel vous effectuez les opérations.
- fdisk prend comme argument le chemin du fichier spécial associé au disque. À défaut, il utilisera le premier disque trouvé : **fdisk /dev/sda**

Il doit y avoir 1 à 4 partitions primaires par disque !

Partitions primaires → étendues → logiques

mke2fs

mke2fs crée un système de fichiers étendu Linux version 2 (second extended file system) sur un périphérique (habituellement une partition de disque).

mount

Tout fichier accessible par un système Unix est inséré dans une grande arborescence, la hiérarchie des fichiers, commençant à la racine /. Ces fichiers peuvent résider sur différents périphériques.

La commande **mount** permet d'attacher un système de fichiers trouvé sur un périphérique quelconque à la grande arborescence du système. A l'inverse **umount(8)** le détachera à nouveau.

Policies

- Data separated from systems
- /disks/home, /disks/share, /usr/local

Filesystem Hierarchy Standard

Filesystem Hierarchy Standard (« norme de la hiérarchie des systèmes de fichiers », abrégé en **FHS**) définit l'arborescence et le contenu des principaux répertoires des systèmes de fichiers des systèmes d'exploitation GNU/Linux et de la plupart des systèmes Unix.

/	Répertoire racine
/bin	Commandes de base pour tous les utilisateurs (par exemple : cat, ls, cp) <i>Abréviation de binaries (binaries en anglais).</i>
/boot	Chargeur d'amorçage (par exemple : noyaux, initrd : image mémoire du ramdisk utilisé par le processus init).
/dev	Fichiers correspondant (directement ou non) avec un périphérique. Les fichiers de périphériques : <ul style="list-style-type: none"> • Périphériques physiques : disque, réseau, bande, disquette • Périphériques virtuels : /dev/null, /dev/zero

	<i>Abréviation de device</i>
<code>/lib</code>	Bibliothèques logicielles nécessaires pour les exécutables de <code>/bin</code> et <code>/sbin</code> <i>Abréviation de librairies</i>
<code>/usr</code>	Contient certains dossiers semblables à ceux présents à la racine mais qui ne sont pas nécessaires au fonctionnement minimal du système. <i>Abréviation de Unix System Resources</i>
<code>/opt</code>	Logiciels optionnels (logiciels non inclus dans la distribution, installés manuellement)
<code>/etc</code>	Fichiers de configuration <i>Abréviation de Editing Text Configuration</i>
<code>/var</code>	Fichiers variables, tels que base de données, boîte de messagerie, ...
<code>/tmp</code>	Fichiers temporaires (voir aussi <code>/var/tmp</code>) <i>Abréviation de temporary</i>
<code>/sbin</code>	Exécutables pour les administrateurs <i>Abréviation de system binaries</i>

User Management

Policy for user directories

On ne va pas donner aux user des noms explicites : pas de `/home/ipl/info/Julie`

On va utiliser des noms suivant une règle définie qui anonymise les utilisateurs :

`u2bin042`

De cette façon il est moins facile de faire une attaque sur un utilisateur particulier. On va aussi regrouper les utilisateurs dans des dossiers commençant par la première lettre de leur nom :

`/home/c/carbon`

`/home/t/titane`

`/home/c/cuivre`

Cela permet une recherche plus aisée.

useradd, userdel, groupadd, groupdel

On va automatiser la création des comptes utilisateurs : on part d'un fichier liste

```
# cat ipl_list_user

carbone : pw_carbonne : Pierre CARBONNE

titane : pw_titane : Claude TITANE
```

```

...

# cat ipl_list_adduser.sh

#!/bin/bash

#set -x                                // mode debug

if [ !-W / ] ; then
    echo "$0 vous n'êtes pas root";
    exit 1;
fi

if [ $# -ne 3 ] ; then
    echo "Usage : $0 user pass realName"
    exit 2;
fi

user=$1
pass=$2
realN=$3

hdir=$(perl -e "print 'home/'.substr($user,0,1).'/${user}'")
groupadd $user
mkdir -p /disks/$hdir
ln -s /disks/$hdir /home/$user          // on crée un lien symbolique
usradd -g $user -p INACTIVE -d /home/$user -c "realN" $user
chown $user.$user /disks/$hdir
su -c "cd /home/$user; cp -np /etc/skel/.??* ." $user
echo $pass | passwd -stdin $user

exit 0

```

su, sudo, visudo

En tapant la commande **su**, l'utilisateur est invité à rentrer le mot de passe du root et, après authentification, on lui donne un accès au shell root.

Une fois connecté via la commande **su**, l'utilisateur est l'utilisateur root et possède alors les accès administrateurs au système. De plus, une fois que l'utilisateur est devenu root, il lui est possible d'utiliser la commande **su** pour se connecter au compte de n'importe quel utilisateurs du système sans devoir indiquer le mot de passe.

sudo⁷ est un logiciel permettant à un utilisateur lambda d'exécuter des commandes nécessitant les droits administrateur (root). Concrètement, pour exécuter une commande avec les droits de root, il faut taper cette commande : `sudo commande` puis taper votre mot de passe utilisateur.

➔ **/etc/sudoers** : fichier de configuration de la commande sudo.

Vous pouvez vous apercevoir que lorsque vous êtes logué en root avec **sudo su**, votre home devient **/root** et non plus **/home/login**.

sudo -s Vous êtes désormais toujours root mais vous avez conservé votre home utilisateur. Par conséquent, vous conservez également les paramètres de

⁷ Substitute User DO (et non SuperUser DO)

votre `/home/login/.bashrc` (ou de tout autre fichier de configuration présent dans votre `/home/login`). Cela évite, dans le cas du `.bashrc`, d'avoir à éditer le fichier `/etc/bash.bashrc` ou bien de devoir éditer le `/home/login/.bashrc` et `/root/.bashrc`.
Note : C'est également pratique pour le répertoire `.ssh`.

Je vous conseillerais donc `sudo -s` plutôt que `sudo su`, mais c'est à vous de voir :)

Le fichier de configuration de sudo est `/etc/sudoers`.

Attention : Il ne faut **jamais** éditer `/etc/sudoers` « à la main ». Pour modifier ce fichier, nous utiliserons la commande `visudo`. L'avantage de `visudo` est qu'il détecte les éventuelles erreurs de syntaxe avant d'enregistrer le fichier (c'est un peu le même principe que le `crontab`).

Voici, pour l'exemple, un fichier de configuration et ses explications :

```
# /etc/suders
# This file MUST be edited with the 'visudo' command as root
#
# See the man page for details on how to write a sudoers file.
Defaults      env_reset
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root ALL = (ALL) ALL
# Uncomment to allow members of group sudo to not need a password
# (Note that later entries override this, so you might need to move it
# further down)
# %sudo ALL = NOPASSWD: ALL

# Members of the admin group may gain root privileges
%admin ALL = (ALL) ALL
```

Spécifier les privilèges :

```
utilisateur machine = () commandes
```

Par exemple :

- `benjamin localhost = /usr/bin/wireshark` : permet d'autoriser l'utilisateur benjamin à exécuter Wireshark en tant que root sur sa propre machine
- `benjamin localhost = (nagios) /usr/bin/pkill, /usr/bin/kill`, permet d'autoriser l'utilisateur benjamin à exécuter pkill et kill en tant qu'utilisateur nagios (par exemple, grâce à cette commande : `sudo -u nagios pkill nagios`)
- `%admin ALL=(ALL) ALL`, permet d'autoriser aux membres du groupe admin d'exécuter toutes les commandes sous n'importe quel compte (=ALL), à partir de n'importe quelle machine

- **#%sudo ALL=NOPASSWD: ALL**, autoriserait (s'il n'était pas commenté) les utilisateurs du groupe sudo à exécuter n'importe quelle commande à partir de n'importe quelle machine en tant que n'importe quel utilisateur sans entrer de mot de passe.

Attention : Veillez à ne pas donner de droits root à vos utilisateurs qui leur permettraient de modifier leurs propres droits. Par exemple, si vous leur donnez les droits root pour l'exécution de gedit, rien ne les empêcherait de modifier le fichier /etc/group pour s'ajouter dans le groupe admin.

Disk space management

du, df

du – Statistiques sur l'utilisation du disque

- `cd /disks/home`
- `du -s -k */*` // Espace disque utilisé par l'utilisateur à qui « home » appartient.
- `du -s -k */* | sort -nr | head -10` // On ne voit que la taille du répertoire utilisateur.

df – Indique les quantités d'espace disque utilisées et disponibles sur les systèmes de fichiers.

-h, --human-readable

Ajouter à chaque valeur une lettre comme M pour Mega-octet, afin d'améliorer la lisibilité

-H, --si

Comme **-h**, en utilisant les unités SI officielles (puissances de 1000 plutôt que 1024, ainsi M signifie 1000000 et non 1048576)

quota

La gestion des quotas permet de limiter le volume en termes de nombre de blocs ou de nombre de fichiers (**inode**) pour un utilisateur ou pour un groupe d'utilisateur. Cette limitation se fait sur un système de fichier.

Il existe deux sortes de limitations :

- **La limitation fixe** : Permet d'empêcher l'utilisateur de dépasser le seuil mis en place. Passé cette limite toute écriture pour cet utilisateur sur ce système de fichier est interdite. L'inconvénient de cette limitation est que l'utilisateur pourrait avoir besoin de rapatrier un fichier de grande taille pour un certain temps et qui risque de dépasser le seuil autorisé.
- **La limitation souple** : Cette limitation permet à l'utilisateur de dépasser pour un laps de temps (appelé période de grâce) le seuil autorisé. Lorsqu'un user dépasse son seuil, le système peut en avertir l'administrateur.

Il est possible d'utiliser ces deux limitations en même temps, c'est à dire donner une certaine souplesse d'utilisation de l'espace disque à un utilisateur, mais en lui donnant tout de même un seuil à ne pas franchir.

usrquota, grpquota (/etc/fstab) (need to remount)

1. Editer le fichier /etc/fstab pour poser les quotas sur une partition.

```
/dev/sdb2    /home    ext3    defaults,usrquota,grpquota    0    0
```


2. Afin d'utiliser les quotas, il faut créer deux fichiers, dans chaque système de fichier soumis aux quotas. Il faudra, juste après ça, leur donner les droits d'accès uniquement aux administrateurs afin que les utilisateurs ne puissent pas modifier leurs quotas. Ces fichiers se trouvent dans le /home.

```
# touch /home/aquota.grp  
  
# touch /home/aquota.usr  
  
# chmod 600 /home/aquota.grp  
  
# chmod 600 /home/aquota.usr
```

3. Maintenant nous pouvons démarrer le daemon de gestion de quota. Au préalable, après avoir modifié le fichier /etc/fstab, il faut toujours le recharger, on fait cela grâce à la commande mount -a

quotacheck -cug filesystem

quotacheck : Vérifie les quotas pour un système de fichier

-a

Pour tous les systèmes de fichiers

-u

Pour les systèmes de fichier étant soumis à la gestion des quotas pour utilisateur

-g

Pour les systèmes de fichier étant soumis à la gestion des quotas pour groupe

-c

Don't read existing quota files. Just perform a new scan and save it to disk. **quotacheck** also skips scanning of old quota files when they are not found.

repquota -a

repquotas : Affiche la configuration des quotas

-a

Sur tous les systèmes de fichier

-u

Sur les quotas utilisateurs

-g

Sur les quotas des groupes

edquota [-p] user

edquota : Attribution des quotas pour les différent user/groupe/files système

-u

Pour un ou plusieurs utilisateurs

-g

Pour un ou plusieurs groupes

-t

Permet de définir le temps de grâce (limite souple, soft limit)

-p --protoname

Duplicate the quotas of the prototypical user specified for each user specified.
This is the normal mechanism used to initialize quotas for groups of users.

Startup and configuration files

/etc/inittab (run level & init script)

Lorsque vous démarrez le système ou changez les niveaux d'exécution avec la commande `init` ou `shutdown`, le démon `init` démarre les processus en lisant des informations à partir du fichier `/etc/inittab`. Ce fichier définit les éléments importants pour le processus `init` :

- Le fait que le processus `init` va redémarrer ;
- Les processus à démarrer, surveiller et redémarrer s'ils se terminent ;
- Les actions à entreprendre lorsque le système entrera dans un nouveau niveau d'exécution.

Chaque entrée du fichier `/etc/inittab` contient les champs suivants :

```
id:rstate:action:process
```

Le tableau suivant décrit les champs dans une entrée `inittab`.

id	Identificateur unique pour l'entrée
rstate	Répertorie les niveaux d'exécution auxquels cette entrée s'applique
action	<p>Indique la manière dont les processus spécifiés dans le champ <code>process</code> doivent être exécutés.</p> <p>Les valeurs possibles principales sont les suivants : <code>sysinit</code>, <code>boot</code>, <code>bootwait</code>, <code>wait</code> et <code>respawn</code>.</p> <p>Pour obtenir une description des autres mots clé d'action, reportez-vous à la page manuel <code>inittab(4)</code>. Sinon, la liste détaillée se trouve ici.</p>
process	Définit la commande ou le script à exécuter

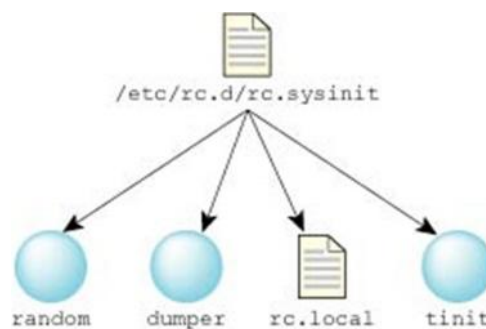
/etc/rc.d/rc.sysinit (← /etc/rc.sys.init)

Se lance une seule fois au démarrage (boot)

```
# Définir des variables d'environnement
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
HOSTNAME='/bin/hostname'
#Le fichier de configuration du réseau existe-t-il ?
if [ -f /etc/sysconfig/network ]; then
    ./etc/sysconfig/network
else
    NETWORKING=no
fi
```

/etc/system/sysinit

Le script /etc/system/sysinit exécute /etc/rc.d/rc.sysinit en initialisant les variables locales du système.



Le script /etc/system/sysinit fait les choses suivantes :

1. Il démarre un générateur de nombres aléatoires sécurisé, **random**, pour fournir un nombre aléatoire utilisé dans l'encryption et ainsi de suite.
2. Si le dossier /var/dumps existe, rc.sysinit démarre l'utilitaire « dumper » pour capturer (dans /var/dumps) les dumps des processus qui se terminent anormalement.
3. Si /etc/host_cfg/\$HOSTNAME/rc.d/rc.local existe et est exécutable, rc.sysinit l'exécute. Sinon, si /etc/rc.d/rc.local existe et est exécutable, rc.sysinit l'exécute. Il n'existe pas une version par défaut de ce fichier, vous devez le créer si vous voulez l'utiliser.
4. Finalement, rc.sysinit exécute tinit. Par défaut, le système démarre « Photon » mais si vous créez un fichier nommé /etc/system/config/nophoton, alors rc.sysinit dit à tinit d'utiliser le mode texte.

./etc/sysconfig/network (explain dot)

Le fichier /etc/sysconfig/network est utilisé pour spécifier des informations sur la configuration réseau désirée. Les valeurs suivantes peuvent être utilisées :

- **NETWORKING=<valeur>**, où **<valeur>** est une des valeurs booléennes suivantes :
 - **yes** — la connexion au réseau devrait être configurée.
 - **no** — la connexion au réseau ne devrait pas être configurée.
- **HOSTNAME=<valeur>**, où **<valeur>** devrait être un *nom de domaine complet*, tel que hôte.domaine.com, mais vous pouvez choisir le nom d'hôte que vous voulez.

Remarque : Pour assurer la compatibilité avec des logiciels plus anciens que certaines personnes risqueraient d'installer (tels que **trn**), le fichier `/etc/HOSTNAME` devrait contenir la même valeur qu'ici.

- **GATEWAY=<valeur>**, où **<valeur>** est l'adresse IP de la passerelle du réseau.
- **GATEWAYDEV=<valeur>**, où **<valeur>** est le périphérique de passerelle, tel que `eth0`.
- **NISDOMAIN=<valeur>**, où **<valeur>** est le nom de domaine NIS.

Attention : Dans un script qui doit éditer des fichiers, il doit être défini au niveau du père avec « . »

/etc/rc.d/rc\$runLevel.d

`$runLevel` permet de récupérer le niveau d'exécution (run level) du serveur (pour notre serveur `$runLevel` renvoie 3)

Dans notre cas : `/etc/rc.d/rc3.d`

- **rc0.d** : Contient les scripts exécutés quand le système s'éteint. Techniquement, « halt » ou « shutdown » amène le système au niveau 0. Ce répertoire est le plus souvent constitué de commandes « kills ».
- **rc1.d à rc3.d** : Contient les scripts exécutés par le système aux changements de niveau d'exécution.
 - Le niveau 1 d'exécution est habituellement le mode mono utilisateur.
 - Le niveau 2 d'exécution est configuré pour les multi utilisateur sans NFS.
 - Le niveau 3 d'exécution est pour les installations full multi utilisateur et réseau.
- Le niveau 4 d'exécution n'est pas utilisé.
- **rc5.d** : Contient les scripts pour démarrer le système en mode X11. C'est le même que le niveau 3, avec l'exception que le programme `xdm` démarre (fournit l'interface graphique).
- **rc6.d** : Contient les scripts à exécuter lorsque le système redémarre. Ces scripts sont appelés par la commande « reboot ».

/etc/rc.local

Le fichier `/etc/rc.d/rc.local` est le dernier script à être exécuté au démarrage du PC.

Pour qu'un script ou une commande Linux soit automatiquement lancé au démarrage du PC, il suffit d'ajouter la commande ou l'appel au script à la fin du fichier `/etc/rc.d/rc.local`. Pour cela, sous `root` tapez la commande suivante :

```
# echo "commande_ou_script_a_appeler" >> /etc/rc.d/rc.local
```

/etc/rc\$runLevel.d/K* and S*

`$runLevel` permet de récupérer le niveau d'exécution (run level) du serveur (pour notre serveur `$runLevel` renvoie 3 → `/etc/rc.d/rc3.d`).

Tous les fichiers d'un niveau d'exécution contiennent une liste de fichiers qui comment par :

- la lettre K : Kill
- la lettre S : Start

Les fichiers répertoriés ci-dessus indiquent les sous-systèmes qui ne sont pas à exécuter pour ce niveau d'exécution (si ils sont dans un fichier qui commence par la lettre «K») et les sous-systèmes qui sont à exécuter pour ce niveau d'exécution (si ils sont dans un fichier qui commence par la lettre "S").

/etc/init.d/serviceName

Ces programmes s'exécutent en arrière-plan et sont couramment appelé « services » ou « daemon ». Les serveurs requièrent habituellement des numéros de services pour s'exécuter en arrière-plan, comme un serveur web, un serveur mail, un serveur de base de données, etc.

(Network) services and configuration files

Le répertoire **init.d** contient un certain nombre de scripts de démarrage/arrêt pour les différents services de votre système.

Mais il y a des moments où vous avez besoin de démarrer ou arrêter un processus proprement sans utiliser les commandes kill ou killall. C'est là que le répertoire **/etc/init.d** vient à point.

Afin de contrôler manuellement n'importe quel script de **init.d**, vous devez avoir les accès root (ou sudo). Chaque script sera exécuté comme une commande et la structure de la commande devrait ressembler à:

```
# /etc/init.d/command OPTION
```

Où commande est la commande réelle à exécuter et OPTION peut être une des suivantes:

- start
- stop
- reload
- restart
- force-reload

Le plus souvent, vous utiliserez soit démarrer, arrêter ou redémarrer. Donc, si vous voulez arrêter votre réseau, vous pouvez exécuter la commande:

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

Ou si vous faites un changement à votre réseau et avez besoin de le redémarrer, vous pourriez le faire avec la commande suivante:

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

Certains des scripts d'initialisation plus communs dans ce répertoire sont:

- networking
- samba
- apache2
- ftpd
- sshd
- dovecot
- mysql

/etc/sysconfig

- Plus ou moins équivalent à la registry windows
- Contient de nombreux fichiers de configuration (configuration système) différents pour RHEL.

[/etc/sysconfig/network-scripts/ifcfg-eth0](#)

Le fichier **/etc/sysconfig/network-scripts/ifcfg-ethN** (où **N** est le numéro de carte Ethernet, **0** pour la première, **1** pour la seconde, ...), qui configure l'interface Ethernet, devrait ressembler à ceci :

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
TYPE=Ethernet
USERCTL=no
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
PEERDNS=no
```

Le script de démarrage du réseau est le fichier **/etc/rc.d/init.d/network**. Les scripts de démarrage des services réseau se trouvent dans le répertoire **/etc/rc.d/rc3.d/**.

[/etc/init.d/network](#)

La configuration d'une interface avec ifconfig n'est pas enregistrée sur le disque, et en particulier, elle n'est pas conservée en cas de réinitialisation du système (reboot). Pour enregistrer la configuration de manière permanente, il faut créer cette configuration dans un fichier de configuration.

Pour initialiser le réseau après configuration, il faut faire :

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

Pour réinitialiser le réseau après un changement dans les fichiers de configuration :

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

[/etc/init.d/nfs](#)

NFS⁸ est un protocole développé par Sun Microsystems qui permet à un ordinateur d'accéder à des fichiers via un réseau. Il fait partie de la couche application du modèle OSI. Ce système de fichiers en réseau permet de partager des données principalement entre systèmes UNIX.

[Other less used daemons from /etc/init.d/xinetd](#)

Dans l'absolu, **inetd** et **xinetd** ont le même rôle, à savoir de piloter l'accès à un ou plusieurs services réseaux. Ils agissent comme une standardiste. Ils reçoivent des requêtes de clients, extérieurs pour la plupart, qui demandent un accès à un service réseau déterminé (ex : ftp, telnet, ssh...). Le super démon va, en fonction des instructions qu'on lui aura données (fichiers de configuration) transmettre ou rejeter l'appel.

⁸ Network File System

inetd, permettait de paramétrer l'accès aux services en l'autorisant/interdisant totalement ou partiellement. (cf. les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`).

xinetd apporte des fonctionnalités bien plus importantes et permet d'affiner les paramètres d'accès aux services. On citera dans le désordre :

- Possibilité d'affiner les logs des services gérés
- Paramétrage d'accès par service et non global
- Paramétrage des plages horaires de disponibilité des services
- Possibilité de chrooter les services (ex : ftp)
- Possibilité de limiter les attaques de type deny of service (contrôle de la priorité d'un serveur, contrôle de la charge CPU, contrôle du nombre de connexions par service, ...)
- redirection de ports

Pour configurer xinetd, vous aurez à connaître la syntaxe, commune de `/etc/xinetd.conf` et, selon les cas de figure, les fichiers situés dans le répertoire `/etc/xinetd.d`.

L'arborescence de la configuration de xinetd est relativement simple. On en rencontre 2 types :

- **un seul fichier de configuration** : `/etc/xinetd.conf` qui comprendra la configuration générale de xinetd **et** la configuration des services gérés par xinetd. (exemple plus loin dans l'article)
- **Un fichier réservé à la configuration général** de xinetd, nommé aussi `/etc/xinetd.conf`. La **configuration des services** est déportée dans des fichiers situés dans le répertoire `/etc/xinetd.d`. Ce répertoire comprend un fichier de configuration par service géré par xinetd. Le fichier porte le nom du service. Pour utiliser ce deuxième cas de figure, le fichier `/etc/xinetd.conf` doit contenir la ligne suivante : `includedir /etc/xinetd.d`

C'est ce deuxième cas de figure qui est le plus couramment utilisé dans les distributions.

[Missing from the base install ...](#)

[Yum install xinetd](#)

[Configs in /etc/xinetd.conf and /etc/xinetd.d/ ...](#)

[Example rsync](#)

[NFS](#)

[File /etc/exports](#)

Le fichier **/etc/exports** contient une table de systèmes de fichiers physiques locaux sur un NFS serveur qui est accessible aux clients NFS.

[/etc/init.d/nfs start](#)

[/sbin/services nfs start](#)

[/sbin/chkconfig --level 345 nfs on](#)

Client system

[iplclint01 installation](#)

[Fedora Core 13](#)

[Avoid express installation \(keyboard ...\)](#)

[Create « local » login mandatory](#)

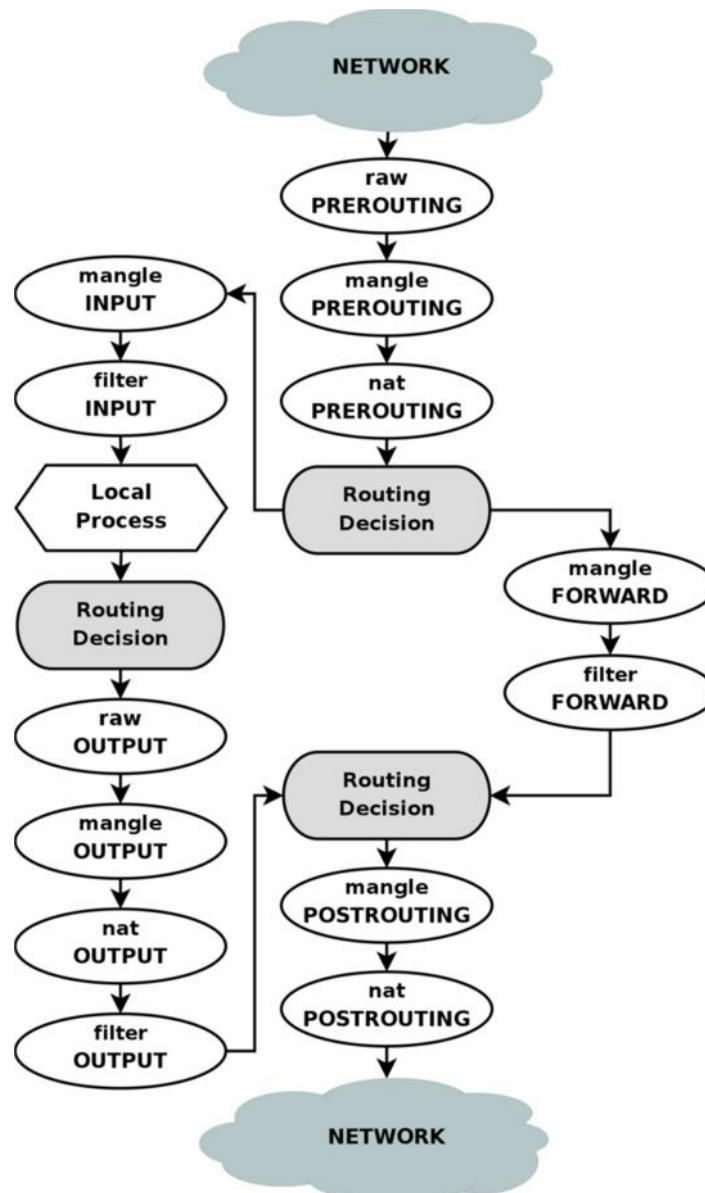
NFS server w/ client

[Client cannot connect to server](#)

Firewall

Ce chapitre décrit comment utiliser iptables, le firewall de Linux. Un script facilitant son utilisation aux quotidiens est également présenté.

Le fonctionnement d'iptables est le suivant :



Chaque paquet doit traverser une série de filtres en fonction de sa source et sa destination. De base, tous les paquets sont autorisés. Voici comment protéger un serveur simplement :

- **Forward** concerne les paquets traversant un serveur en mode routeur, il faut donc le bloquer.
- **Output** analyse le paquet émis par le serveur directement, c'est à dire les sessions initiées par le serveur mais aussi les réponses aux sessions reçues, il n'est pas nécessaire de le filtrer.
- **Input** filtre les demandes de connexions à destination du serveur mais aussi les réponses aux sessions émises par le serveur, c'est cette table qu'il faut construire.

Pour bloquer FORWARD, il suffit d'utiliser la commande suivante :

```
# iptables -P FORWARD DROP
```

Avant de bloquer INPUT, il est important de laisser quelques flux passer :

- Les flux provenant de l'interface interne :

```
# iptables -A INPUT -i lo -j ACCEPT
```

- Les flux provenant des connexions établies (les réponses aux requêtes du serveur)

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

Vient ensuite les flux autorisés, par exemple :

- Les flux provenant de l'interface du réseau interne

```
# iptables -A INPUT -i eth1 -j ACCEPT
```

- Les flux icmp

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

- Le SSH

```
# iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

- L'accès au serveur web en HTTP

```
# iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

- L'accès au serveur web en HTTPS

```
# iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Et enfin, la règle de DENY sur les autres paquets entrants :

```
# iptables -P INPUT DROP
```

Au final, la configuration est la suivante :

```
bender:~# iptables-save

# Generated by iptables-save v1.3.6 on Sat Jul 14 16:22:53 2007

*mangle

:PREROUTING ACCEPT [1882:245141]

:INPUT ACCEPT [1882:245141]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [1271:188201]
```

```
:POSTROUTING ACCEPT [1271:188201]

COMMIT

# Completed on Sat Jul 14 16:22:53 2007

# Generated by iptables-save v1.3.6 on Sat Jul 14 16:22:53 2007

*filter

:INPUT DROP [332:115556]

:FORWARD DROP [0:0]

:OUTPUT ACCEPT [1271:188201]

-A INPUT -i lo -j ACCEPT

-A INPUT -i eth1 -j ACCEPT

-A INPUT -p icmp -j ACCEPT

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

-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT

COMMIT

# Completed on Sat Jul 14 16:22:53 2007

# Generated by iptables-save v1.3.6 on Sat Jul 14 16:22:53 2007

*nat

:PREROUTING ACCEPT [126:14961]

:POSTROUTING ACCEPT [86:8122]

:OUTPUT ACCEPT [87:8195]

COMMIT

# Completed on Sat Jul 14 16:22:53 2007
```

[Disable by chkconfig --level 345 iptables off](#)

[Problems with Selinux \(enforcing → permissive – solved !\)](#)

Setenforce permissive

Upfate /etc/sysconfig/selinux

Update /etc/fstab

Userdel -r nina ...

Lsof Directory – NIS

NIS – Network Information Service

YP – Yellow Page (protected trademark)

Network Information Service (NIS) nommé aussi Yellow Pages est un protocole client-serveur développé par Sun permettant la centralisation d'informations sur un réseau UNIX.

Son but est de distribuer les informations contenues dans des fichiers de configuration contenant par exemple les noms d'hôte (/etc/hosts), les comptes utilisateurs (/etc/passwd), etc. sur un réseau.

Un serveur NIS stocke et distribue donc les informations administratives du réseau, qui se comporte ainsi comme un ensemble cohérent de comptes utilisateurs, groupes, machines, etc.

À l'origine, NIS est sorti sous le nom de « Yellow Pages » (YP) ou Pages jaunes mais le nom étant déposé par la compagnie britannique British Telecom, Sun a renommé son protocole NIS. Cependant, les commandes NIS commencent toutes par yp.

NIS est réputé pour être faible en termes de sécurité.

Architecture

Server

Master – Slave

Domaine

```
# domainname ipl  
  
# /etc/sysconfig/network
```

Server iplsrv01

```
# yum install ypserv // Paquets logiciels à installer  
  
# /usr/lib64/yp/ypinit -m // Initialisation
```

```
# /etc/init.d/ypserv start      // Démarre ypserv  
  
# chconfig -level 345 ypserv on // Redémarre avec le service  
  
# /etc/init.d/ypbind start     // Démarre ypbind  
  
# chconfig -level 345 ypbind on // Redémarre avec le service
```

Ensuite mettre à jour /etc/hosts et le fichier dbm de /var/yp.

Configurer le firewall en « bind » ypserv sur le port 714 :

```
# /etc/sysconfig/network  
  
# YPSERV_ARGS="-p 714"
```

Met à jour /etc/sysconfig/iptables.

Client iplcInt01

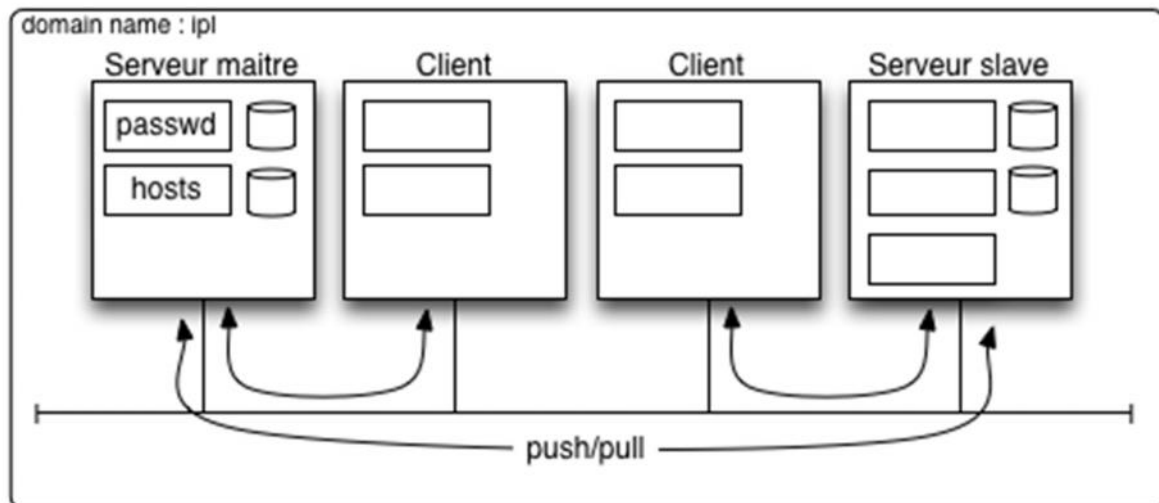
```
# /etc/sysconfig/network  
  
# /etc/yp.conf  
  
# /etc/init.d/iptables stop  
  
# /etc/nsswitch.conf
```

Directory OpenLDAP

Définition

LDAP (Lightweight Directory Access Protocol, Protocole d'accès aux annuaires léger) est un protocole standard permettant de gérer des annuaires, c'est-à-dire d'accéder à des bases d'informations sur les utilisateurs d'un réseau par l'intermédiaire de protocoles TCP/IP.

Le protocole LDAP définit la méthode d'accès aux données sur le serveur au niveau du client, et non la manière de laquelle les informations sont stockées.



Qu'est-ce qu'un annuaire ?

- Stockage et consultation d'informations
- Dédié à la lecture plus qu'à l'écriture
- Accès se fait par recherche multicritères

Un annuaire électronique, c'est en plus :

- Un protocole d'accès
- Un modèle de distribution
- Un modèle de duplication de l'information
- Un contenu évolutif

Exemple : DNS

Le protocole définit :

- Comment s'établit la communication client-serveur (bind, unbind, abandon)
- Comment s'établit la communication serveur-client
 - Synchronisation (réplication service)
 - Liens entre différents annuaires(referral service)
- Transport des données : pas l'ASCII (http, smtp, ...) mais Basic Encoding Rules
- Les mécanismes de sécurité
 - Méthodes de chiffrement et d'authentification
 - Mécanismes d'accès aux données
- Les opérations de base (search, add, delete, etc.)

Attributs

Caractérisés par un nom, un nom alternatif, un type et un Object Identifier (OID). Le type le plus employé est la chaîne de caractères mais aussi des champs d'octets pour stocker des images.

dn	Distinguished name	cn	Common name
o	Object	sn	Surname
ou	Organizational unit	uid	Identifieur unique obligatoire
c	Country	mail	Mail

Commandes

```
# yum install openldap-servers // Paquets logiciels à installer pour le serveur

# yum install openldap-client // Paquets logiciels à installer pour le client

# tune2fs -m 1 /dev/sda2 // Ajuste les param des sys de fichiers ext2/ext3

# cd /etc/openldap // Dossier de config de openldap

# vi slapd.conf // Met à jour le fichier de config
```

Modifications apportées au fichier de config slapd.conf

Directory hierarchy

Suffix : o=ipl,c=be // alternative dc=ipl,dc=be

Rootdn = cn=Manager,o=ipl,c=be

Copier /etc/openldap/DB_CONFIG.example dans /var/lib/ldap/DB_CONFIG

```
# slappasswd // Copie dans le .config l'attribut dans rootpw

# /etc/init.d/ldap start // Démarre ldap

# chconfig -level 345 ldap on // Redémarre avec le service

# vi base.ldif // Permet de construire l'arborescence

# ldapadd -x -D "cn=Manager,o=ipl,c=be" -W -f base.ldif // Création d'une fiche

# ldapdelete -x -D "cn=Manager,o=ipl,c=be" -W "cn=secretariat,ou=group,o=ipl,c=be"

# ldapsearch -x -D "cn=Manager,o=ipl,c=be" -b "o=ipl,c=be" -W uid=titane

# vi passwd.ldif // Comprend les différentes fiches

# slapcat | more // Contenu complet de l'annuaire
```

Samba

Implémentation « open source » des protocoles MS.

```
# /etc/init.d/samba start // Démarre samba

# vi /etc/samba/smb.conf // Edite le fichier de configuration pour partage

# smbpasswd -a titane // Associe un compte à samba
```

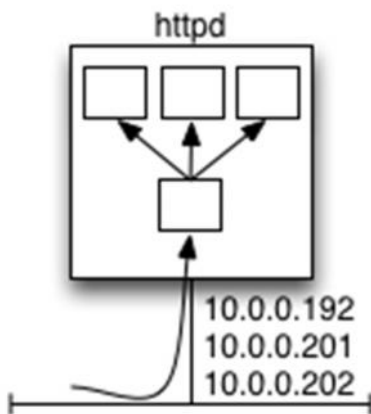
```
# /etc/sysconfig/iptables

# nmlookup -A 10.0.0.192

# smbclient -L 10.0.0.192 -U titane

# smbclient //10.0.0.192/secretariat -U titane
```

Serveur Web Apache



Apache : serveur virtuel

- 10.0.0.192 iplsrv01
- 10.0.0.201 iplsrv02
- 10.0.0.202 iplsrv03

Serveur virtuel donne des adresses identiques pour chaque serveur :

- 10.0.0.192 A
- 10.0.0.192 CNAME
- 10.0.0.192 CNAME

Fichiers de configuration

- /etc/httpd/conf/httpd.conf
- /etc/httpd/conf.d/*

Pare-feu



2 types de parefeu :

- Stateless : Envoi une requête vers le port 80 mais reçoit pas les réponses si le port est différent. Pas terrible niveau sécurité.
- Statefull : Sauvegarde le port d'entrée et de sortie pour permettre l'échange. Plus sure.

Il faut configurer les modes de sortie du pare-feu : bloquer tous les ports TCP/UDP qui peuvent être ciblés par des virus, trojans, ...

Il faut bloquer les ports Windows : 137 → 139, 445

Notes de cours :

NTP (Network Time Protocol) : est un protocole qui permet de synchroniser, via un réseau informatique, l'horloge locale d'ordinateurs sur une référence d'heure. Pour synchroniser l'heure le système se connecte à un serveur ftp (ex : ntp.belnet.be).

Le fichier **/etc/hosts** contient une liste d'adresses IP et les noms des machines ("hostnames") correspondants. En général, /etc/hosts ne contient que les entrées de votre propre machine, et celles de quelques autres systèmes "importants" (comme votre serveur de noms ou votre passerelle). Votre serveur de noms local s'occupera du reste. Le fichier hosts est d'une grande utilité pour interdire l'accès à certains sites par exemple pour protéger vos enfants des sites adultes ou bien accélérer l'accès à vos sites préférés.

Nessus est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées. Ceci inclut, entre autres :

- ➔ les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service...
- ➔ les fautes de configuration (relais de messagerie ouvert par exemple)
- ➔ les patches de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée
- ➔ les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra pour attaquer les mots de passe à l'aide d'un dictionnaire.
- ➔ les services jugés faibles (on suggère par exemple de remplacer Telnet par SSH)
- ➔ les dénis de service contre la pile TCP/IP

Nessus détecte les machines vivantes sur un réseau, balaie les ports ouverts, identifie les services actifs, leurs versions, puis tente diverses attaques.

S'il y a beaucoup d'utilisateurs sur un serveur la commande **ls /home** va être très lourde car il y a beaucoup d'entrées à afficher en même temps. Pour éviter des ralentissements inutiles **il faut répartir** les différents utilisateurs dans le système.

Ex :

- ➔ **/home/ucl/sysi/ninan**
- ➔ **/home/ipl/adm/dupuis**
- ➔ **/home/ipl/info/demuylder**

Rouge = partition (3 partition différentes).

Comment savoir dans un script si on est root ?

➔ **if [! -w /]** Si on ne peut pas écrire dans la racine c'est qu'on n'est pas root !

group 0 contient les utilisateurs :

- ➔ root
- ➔ halt (si on se connecte avec ce login le serveur s'arrête)
- ➔ rpc

Politique des uid : de 500 → ... les uids représentent des personnes physiques (ça facilite l'envoi de mails à tous les utilisateurs en même temps).

Sous Linux, le répertoire **/etc/skel** contient des fichiers qui remplaceront automatiquement le répertoire /home de l'utilisateur quand ce dernier a été créé avec le programme useradd. Le répertoire /home archive les données personnelles de l'utilisateur telles que la configuration de l'environnement de travail et des applications courantes. Dans un système d'exploitation Linux, c'est aussi le répertoire auquel le système se réfère en premier après la connexion d'un utilisateur, parce qu'il contient les nombreux fichiers de configuration importants du système.

Le programme useradd est situé dans le répertoire /usr/sbin/, et n'est accessible qu'en mode root (il faut alors posséder les droits d'administrateur ou de super-utilisateur).

/etc/skel permet donc à l'administrateur la création d'un répertoire /home par défaut qui fixe les paramètres généraux d'un ordinateur ou réseau afin que tout nouvel utilisateur bénéficie d'un environnement de travail stable.

Vipw édite le fichier de mots de passe après avoir mis en place les verrous appropriés, et lance tous les processus nécessaires après que le fichier de mots de passe ait été déverrouillé. Si le fichier de mots de passe est déjà bloqué par un autre utilisateur qui l'édite, vipw vous demandera de réessayer plus tard. L'éditeur par défaut pour vipw est vi.

Les services doivent être démarrés lorsque le système démarre (lors du boot).

Attention : les pirates peuvent rentrer par ces services démarrés, pour éviter ça on n'exécute plus aucun service au démarrage du serveur excepté pour le service SSH.

Il est donc essentiel de s'occuper de la sécurité lors du boot !

ls /etc/init.d liste les services.

Port SSH = 22

Port FTP = 21

Port TELNET = 23

/etc/services contient la liste des ports.

Les divers numéros de ports standards correspondent univoquement à des types particuliers de services.

/sbin/service nomDuService est la manière officielle de lancer un service.

Ex :

- /sbin/service xinetd start
 - /sbin/service iptables off
-

LDAP (Lightweight Directory Access Protocol, traduisez Protocole d'accès aux annuaires léger et prononcez "èl-dap") est un protocole standard permettant de gérer des annuaires,

c'est-à-dire d'accéder à des bases d'informations sur les utilisateurs d'un réseau par l'intermédiaire de protocoles TCP/IP.

Le protocole LDAP définit la méthode d'accès aux données sur le serveur au niveau du client, et non la manière de laquelle les informations sont stockées.

Suffixes LDAP :

- dn : distinguished name
- o : object
- ou : organizational unit
- c : country
- cn : common name
- givenName : prénom
- sn : surname

Droits spéciaux: setuid, setgid, sticky bit

Quand un fichier est exécutable par son propriétaire, il peut de plus être **setuid**. Cela signifie que lorsqu'il est exécuté, il l'est avec les droits de son propriétaire, et non avec ceux de l'utilisateur qui le lance. Par exemple, le programme passwd, qui permet à un utilisateur de modifier son mot de passe, est setuid root (c'est à dire qu'il est setuid et qu'il appartient à l'utilisateur root): il doit pouvoir écrire dans le fichier /etc/passwd (ou /etc/shadow), dans lequel seul root peut écrire.

De la même façon, un exécutable peut être **setgid**, et s'exécuter avec les droits du groupe auquel il appartient.

Enfin, un exécutable peut être "sticky" (on dit aussi "avoir le **sticky bit** positionné"): cela signifie qu'il reste en mémoire même après la fin de son exécution, pour pouvoir être relancé plus rapidement. Alors qu'un exécutable peut être déclaré setuid et setgid par son propriétaire, seul l'administrateur système peut positionner le sticky bit.

Pour les répertoires, la notion de setuid n'existe pas à ma connaissance. Les setgid et sticky bit prennent une signification différente. Quand un répertoire est setgid, tous les fichiers créés dans ce répertoire appartiennent au même groupe que le répertoire. C'est utilisé par exemple quand plusieurs personnes travaillent sur un projet commun: ils ont alors un groupe dédié à ce projet, et un répertoire setgid appartenant à ce groupe, et ils créent leurs fichiers dans ce répertoire avec les permissions 664: tout le groupe peut alors écrire n'importe quel fichier, vu que tous les fichiers appartiennent au groupe.

Voyons maintenant l'utilisation du **sticky bit**. Comme je l'ai écrit plus haut, un utilisateur qui a le droit d'écrire dans un répertoire peut effacer n'importe quel fichier de ce répertoire. Ça peut être très gênant par exemple pour le répertoire /tmp, dans lequel tout le monde a généralement le droit d'écrire. Pour y remédier, on positionne le sticky bit; ainsi, un utilisateur ne peut effacer que les fichiers qui lui appartiennent.

Quand on écrit les permissions en octal, setuid, setgid et sticky bit sont représentés par une nouvelle série de 3 bits, qui se place avant les 3 autres séries: setuid=4, setgid=2, sticky=1. Ainsi, sur ma machine, le serveur de mail /usr/sbin/sendmail a les droits rwsr-sr-x (rwxr-xr-x, setuid, setgid); en octal, ça donne 6775.

Il existe 2 types de pare-feu (firewall) :

- stateless, ex : ipchain → Pas terrible niveau sécurité
- statefull, ex : iptables → Beaucoup plus sure

Il faut configurer les modes de sortie du pare-feu : bloquer tous les ports TCP/UDP qui peuvent être ciblés par des virus, trojans,...

Il faut bloquer les ports Windows : 137 → 139, 445

2 modes FTP :

- actif
- passif