Administration Système

Xavier MONNIN

Bureau A308
Université Paris 13
xm@lipn.fr

http://lipn.fr/~monnin/

Objectifs du cours

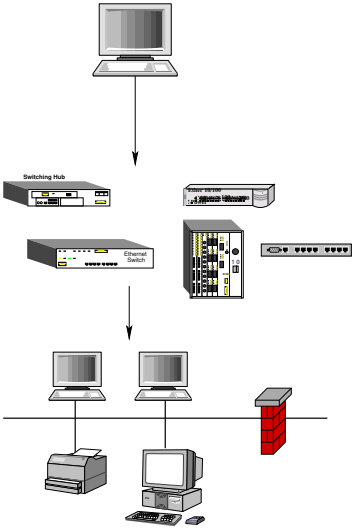
- Acquérir un savoir-faire théorique :
  - en matière d'administration des systèmes d'exploitation
  - au sein d'un réseau informatique
- Acquérir un savoir-faire pratique en TP via :
  - la mise en place et l'administration d'un petit réseau informatique (machines PC/Linux)
  - l'utilisation du logiciel d'émulation réseau Marionnet (`http://www.marionnet.org/`)
  - l'utilisation du logiciel de virtualisation VirtualBox

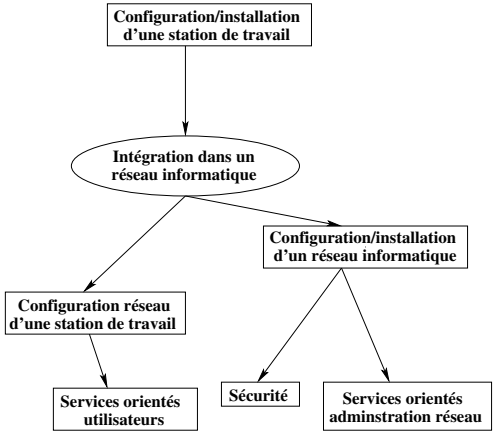
Programme des enseignements (1)

- Administration d'une station de travail
- Administration d'un serveur
- Administration d'un réseau informatique de taille moyenne

$\longrightarrow$ centrée sur les systèmes UNIX (Linux – Debian) Mais des concepts généraux

Programme des enseignements (2)





Répartition des enseignements

Cours :

- 6 séances de 1h30

TPs :

- 10 séances de 3h

Administration d'un réseau informatique

Des connaissances variées dans différents domaines :

- Systèmes d'exploitation
- Programmation au niveau du système
- Réseaux et services informatiques
- Sécurité

Administration système

Systèmes UNIX : homogénéité pour l'utilisateur
Configuration et administration des systèmes UNIX :

- De nombreuses divergences entre les constructeurs
  Chaque système :
  - possède ses spécificités
  - mais tente de suivre les standards (norme POSIX, LAN, ANSI C...)

Complexité de l'administration

- diversité des systèmes
- partage des ressources
- sécurité

Nécessite :

- une bonne connaissance des caractéristiques principales des différentes versions (et donc des commandes UNIX !)
- une expertise, les spécificités de chaque système

Définitions

- **Système :** ensemble des programmes permettant d'accéder à une machine et d'utiliser ses périphériques
- **Réseau :** ensemble des dispositifs (câbles, switchs, routeurs, stations de travail...) connectés entre eux et formant une entité globale vue de l'extérieur
- **Service :** ressource (DNS, Web, NIS, LDAP, Messagerie...) offerte par un programme situé sur une machine (serveur) et, accessible par des machines (clientes) situées sur le même réseau ou à l'extérieur

Bref historique des systèmes UNIX

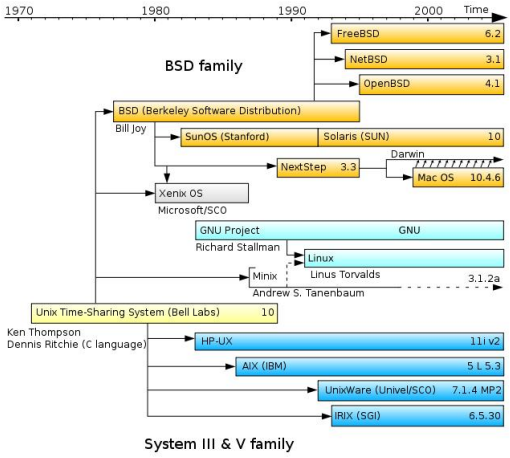
- Version 1 en 1970, (Laboratoire Bell, AT&T)
- Version 6 en 1975
- Version 7 en 1978

Trois branches principales :

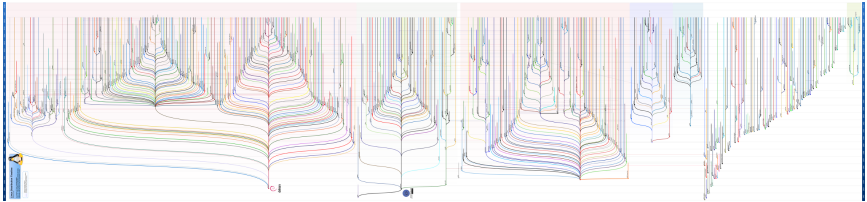
- BSD (Berkeley, Software Distribution)
- Recherche (Laboratoire Bell)
- System (AT&T, Commercial) dit System V

$\rightarrow$ Au total : plus de 150 UNIX depuis 1970

Arbre généalogique des UNIX



Arbre généalogique des Linux



Man

Les pages de manuel des commandes UNIX sont réparties en chapitres appelés des sections :

- section 1 : commandes normales
- section 2 : appels systèmes
- section 3 : fonctions de programmation C
- section 4 : périphériques et pilotes de périphériques
- section 5 : format de fichiers système
- section 6 : jeux
- section 7 : divers
- section 8 : commandes de gestion du système

$$\text{getopt}(3) \rightarrow \text{la commande getopt de la section 3 du manuel}$$

Autres sources

- *Request For Comments* (RFC)
  - `ftp.lip6.fr/pub/rfc/rfc`
  - `http://abcdrfc.free.fr`
- *Frequently Asked Questions* (FAQ)
- *Forums* et *Newsgroups* (`comp.unix.*`, `fr.comp.os.*`)
- *MailingLists*
- Les documentations constructeurs (`www.ibm.com`, `docs.sun.com`, `ftp.lip6.fr/pub/linux`...)

Quelques tâches d'un administrateur système

- Gérer les comptes utilisateurs (tâche simple et automatisable)
- Assister et éduquer les utilisateurs (répondre à leurs questions, documentation à jour pour les outils en place)
- Gérer les logiciels :
  - Installer
  - Configurer
  - Mettre à jour (*patcher*)
- Gérer le matériel :
  - Panne
  - Remplacement
  - Ajout

Quelques tâches d'un administrateur système

- Assurer la sécurité du système et des utilisateurs :
  - Sauvegardes fiables et régulières
  - Contrôle d'accès
  - Utilisations abusives de ressources
- Vérifier l'adéquation du matériel avec son utilisation (identifier les goulets d'étranglement)
- Assurer la maintenance de premier niveau :
  - Diagnostiquer une panne
  - Appel de la maintenance constructeur
- Gérer quotidiennement (multiples tâches, petites ou grosses)

Autres facettes du métier

- Diplomatie, police
- Aspects légaux (chiffrement, Cnil...)
- Enquêtes judiciaires (vol, saccage, piratage informatique, articles pédophiles...)
- Relations commerciales
- Politique d'utilisation des machines

$\rightarrow$ L'administrateur est en première ligne lorsqu'un problème surgit
C'est lui qu'on incrimine naturellement lorsque quelque chose ne fonctionne pas

Connaissances de base

$\rightarrow$ Expert Unix

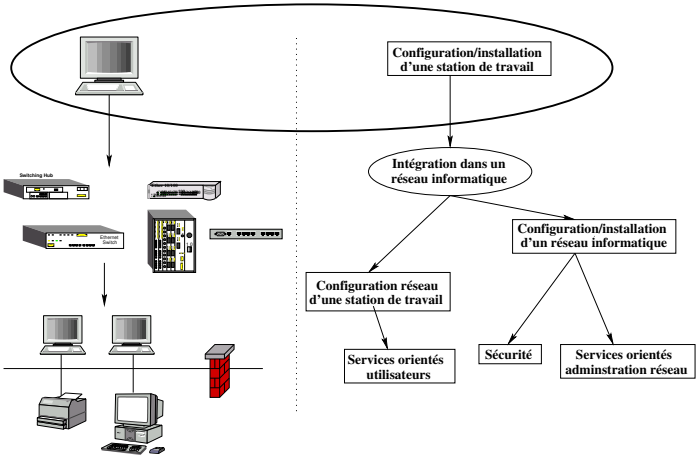
- Environnement utilisateur
- Aide en ligne
- Système de fichiers
- Utilisation du shell
- Utilisation d'un éditeur de texte
- Commandes de base
- Programmation shell

Administrateur système

3 qualités nécessaires :

- Technicité
- Rigueur
- Bon sens

Méthodologie d'administration



Administration système

- Administrer un système est une lourde responsabilité
- L'ampleur de la tâche est variable selon les sites

Dans tous les cas, il faut :

- Veiller au bon fonctionnement du réseau
- Avoir à l'esprit la sécurité du système et du réseau
- Mettre à disposition les outils nécessaires aux utilisateurs
- Gérer et tenir compte du comportement des utilisateurs :
  - Éviter les abus de pouvoir
  - Éviter la paranoïa : la plupart des utilisateurs gênant les fonctionnement d'une machine n'en ont pas conscience
  - Établir des règles de conduite avec les utilisateurs

Méthodologie d'administration (1)

- Garantir l'intégrité des bases de données système et leur mise à niveau
- Consigner :
  - Commandes tapées lors d'installation (notamment sur les serveurs)
  - Opérations effectuées sur les systèmes lors des configurations spécifiques
- Se tenir informer des évolutions des systèmes et du domaine (mesures à prendre en cas de problème de sécurité)
- Documenter

Méthodologie d'administration (2)

- Identifier les bases de données système
- Conserver une version de référence avant toute modification `/etc/inetd.conf.orig` pour `/etc/inetd.conf`
- Assurer une sauvegarde régulière (quotidienne ou hebdomadaire) automatique de ces fichiers sur des supports robustes (CD, bandes, disques externes...)
- Corriger les *bugs* des logiciels en appliquant les patches ou les mises à jour

Administration d'un système d'exploitation

Administration d'un système et d'un réseau :

$\longrightarrow$Différent de l'administration d'un ordinateur mono-utilisateur

- Utiliser les outils graphiques d'administration spécifiques proposés par les constructeurs ne permet pas d'acquérir l'expérience nécessaire pour s'adapter aux évolutions ou aux changements des systèmes
- Administration à l'aide de `scripts` (bash, perl, python...)
  - Création de centaines de comptes utilisateurs
  - Modification ou mise au jour de la configuration de plusieurs systèmes

Utilisation du compte `root`

Pour éviter les erreurs aux conséquences catastrophiques :
utiliser le compte `root` uniquement lorsque c'est nécessaire Quelques règles :

- Vérifier les commandes tapées avant leur exécution
- Utiliser `rm -i` plutôt que `rm` (placer un alias dans l'environnement du root)
- NB :
  Invite `root` : #
  Invite utilisateur ordinaire : $

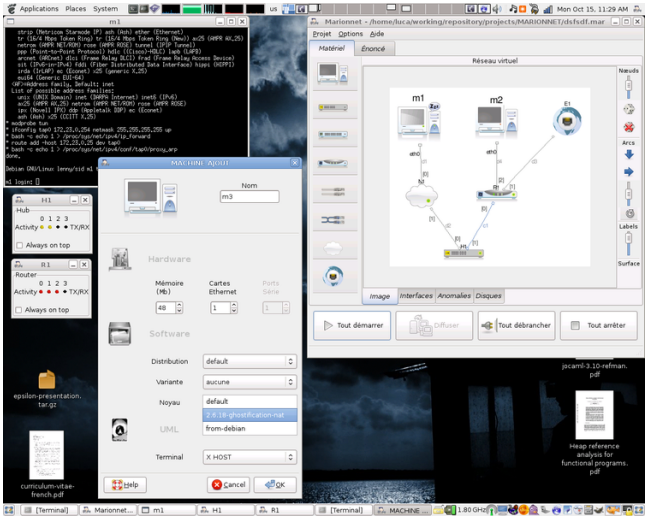
En travaillant sous root, vous ferez une erreur ...
... un jour ! ;-)

Présentation de Marionnet

- Système d'émulation d'un réseau informatique basé sur des machines Linux (Debian)
- Possibilité de configuration d'un réseau (switch, hub, station de travail)
- Simulation d'incidents
- Accès aux systèmes Linux pour une configuration complète

`http://www.marionnet.org`

Capture d'écran



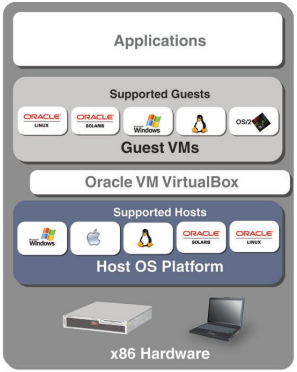
Exécution de Marionnet

- Dans les salles de TP, commande `marionnet`
- Dans une machine virtuelle VirtualBox
  Image disponible sur la page :

`http://marionnet.org/download/Marionnet.ova`

Présentation de Virtualbox : Virtualisation

$\longrightarrow$ Créer/Utiliser des machines virtuelles



Terminologie

Système hôte (host) : système d'exploitation principal qui permet de faire fonctionner VBox

Système invité (guest) : système d'exploitation installé à l'intérieur d'une VM

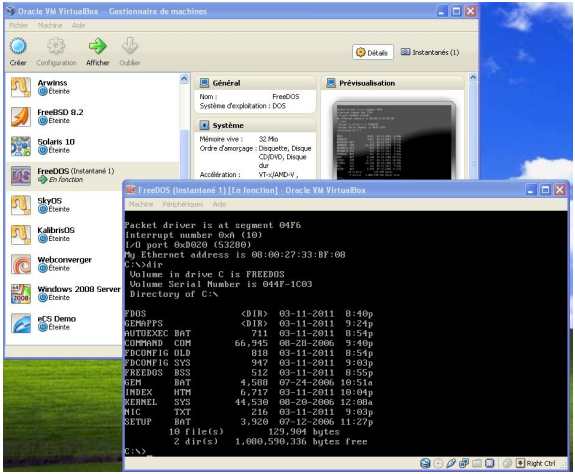
Machine virtuelle : ordinateur virtuel créé par VBox

VDI (Virtual Disk Image) : fichier (unique) contenant le Système invité

Présentation de VirtualBox

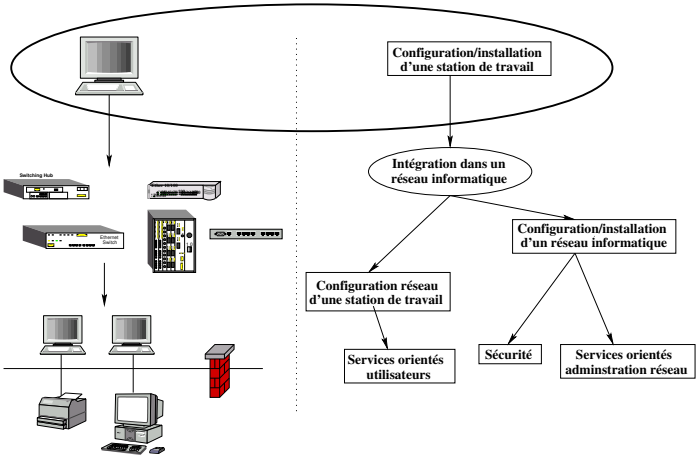
- `https://www.virtualbox.org`
- Logiciel libre (GPL)
- Système hôte :
  - Architecture Intel/AMD
  - Systèmes Linux, Windows, MacOS, Solaris.
- Système invité :
  - La plupart des SE disponibles sur architecture Intel : Linux, Windows, DOS, Unices, etc.
  - `https://www.virtualbox.org/wiki/Guest_OSes`

Capture d'écran



2 Administration d'une station de travail

Administration d'une station de travail



Installation et configuration d'une station de travail

Quatre grandes étapes :

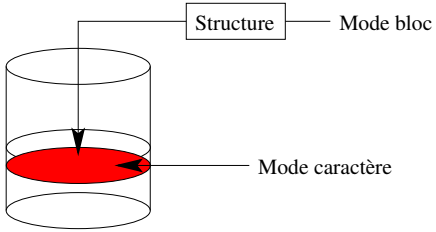
- Partitionnement de l'espace et installation du système de fichiers
- Chargement du système sur le disque
- (Configuration du système par l'administrateur)
- Redémarrage du système
- Configuration du système par l'administrateur (bis)

Périphériques (1)

Représentation des périphériques à travers des fichiers spéciaux (*Device Drivers*) : blocs ou caractères

Accès aux périphériques :

- Mode **b**loc : indirect, à travers les structures (partitions)
- Mode **c**aractère (raw) : direct



Périphériques (2)

- Ajout de périphériques : création de nouveaux fichiers spéciaux
- Certains UNIX offrent des mécanismes d'auto-configuration des périphériques (Solaris)
- Sous Linux, les fichiers spéciaux sont :
  - soit déjà créés
  - soit créés dynamiquement
- Sous UNIX, tout est fichier

Nomenclatures des périphériques

Regroupement des périphériques différents suivant le type d'UNIX :

- `/dev/sda1` : disque en mode bloc
- `/dev/sg` ou `/dev/rsda1` : disque en mode caractère
- `/dev/tty0` : terminal asynchrone
- `/dev/disk` : (*sous debian) accès aux disques par identifiant (indépendant du matériel), chemin, label, identifiant unique universel (UUID - lié au matériel)

Pourquoi partitionner ?

Découpage du disque physique en disques virtuels :

- cohabitation de plusieurs systèmes de fichiers
- isolement de certaines parties du système (`/usr`, `/var`, `/home`)
- facilité de réalisation de certaines tâches (sauvegarde de données, consultation en lecture seulement...)
- exportation de partitions vers d'autres machines

$\longrightarrow$ Changement de taille d'une partition : sauvegarde préalable des données

Identifier les besoins

$\longrightarrow$ Étape importante qui conditionne le bon fonctionnement du système

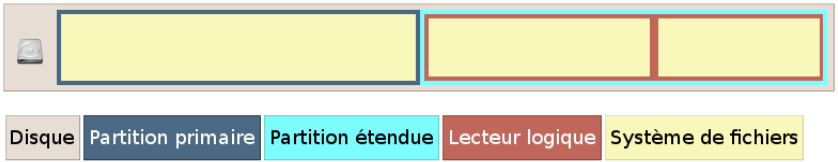
- Avoir une idée précise de l'utilisation future de la machine (serveur de fichier, station de travail, serveur de messagerie, machine de calcul)
- Évaluer :
  - la taille du système
  - les besoins en mémoire virtuelle (zone temporaire, swap)
  - l'espace disque alloué aux utilisateurs

Partitionnement

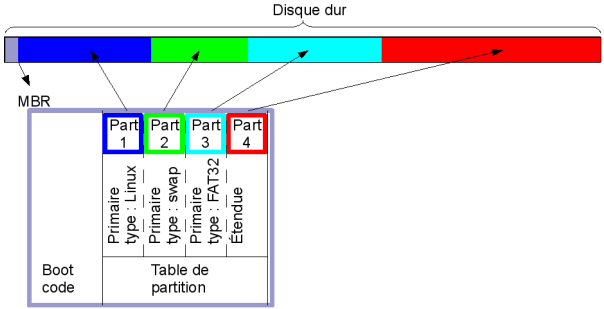
- Précautions :
  - Manuellement : Déterminer le bloc de départ et le bloc d'arrivée
  - Éviter les recouvrements de partition (risque d'erreur très important)
- Outils :
  - `fdisk`, `cfdisk`, `parted`
  - `gparted`

Partitions (type MBR)

→ Organiser les données

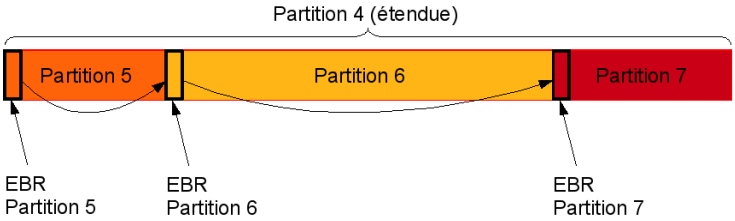


Partition primaire



- Contient au max 4 partitions :
  1 à 3 partitions principales puis une partition étendue
- Reconnue par le bios
- limitation à 2 To par partition

Partition étendue (logique)



- Contient des partitions secondaires
- Contenue dans l'Extended Boot Record (EBR)

Partitions (type GPT)

- Standard GPT : GUID Partition Table
- Nouvelle organisation des partitions
- Utilisation du système EFI (remplaçant du BIOS)
  - plus de limitation à 4 partitions
  - plus de partition étendue
  - limitation à 9,4 Zo par partition

$\rightarrow$ Système jeune

Gestion de volumes logiques

- RAID (Redundant Array of Independent Disks)
  - RAID 0 : Stripping (entrelacement de disques)
  - RAID 1 : Mirroring (miroir de disque)
  - RAID 1+0 (RAID 10) : Stripping et Mirroring
  - RAID 5 : Stripping sur disques indépendants avec parité répartie
  - RAID 6 : évolution du RAID 5, avec n informations redondantes ($n \geq 2$)
- LVM (Logical Volume Management)
  - resize partition
  - snapshot

(Voir description plus tard dans le cours)

Structure du système

- Tout est fichier
- Arborescence de fichiers unique
- Les fichiers ne sont pas typés
- Montage : intégrer les partitions dans l'arborescence
  Permet d'affecter tout système extérieur (disquette, cdrom, rép. réseau...) à un répertoire créé pour cela dans l'arborescence
- 6 catégories de fichiers
  - normaux
  - répertoires
  - périphériques
  - liens
  - pipes
  - sockets

Système de fichiers *local*

(formatage haut-niveau)

- Couche logique permettant :
  - la manipulation des blocs d'un disque
  - le stockage d'une arborescence de fichiers
- Structure d'inode : coordination du bloc de données logiques et l'emplacement sur disque

- Différentes fonctionnalités suivant les types de systèmes pour améliorer les entrées/sorties et le confort d'utilisation (fiabilité, taille limite...)

Installation d'un système de fichiers

- Opération à peu près standard
- Caractéristiques nécessaires à l'installation (valeurs par défaut) :
  - Taille des blocs (8192 octets)
  - Nombre de cylindres par groupe de cylindres dans un système de fichiers (16)
  - Taille des fragments (1024)
  - Densité des inodes (1 inode pour 2048 octets)
  - Pourcentage réservé pour les utilisateurs normaux – quotas (10%)
  - Nombre de pistes par cylindre (16)

Particularités de certains systèmes de fichiers

(fonctionnalités supplémentaires)

- *Checkpointing/Update* : Écriture régulière sur disque des informations d'administration (maintien de l'état consistant du disque)
  `AdvFS`, `JFS`, `VxFS`, `efs`, `XFS`, `ext3`, `ext4`
- *Logging* : Enregistrement des transactions du système de fichiers
  `AdvFS`, `JFS`, `VxFS`, `XFS`
- *Sync-on-close* : Écriture du fichier sur disque sans attendre la synchronisation du cache

Particularités de certains systèmes de fichiers

- *Direct I/O* : Pas de données cachées (augmentation des performances)
- *Allocation par extends* : Allocations des blocs par blocs de multiple de *n* (Éviter la répartition des blocs sur le disque)
  `ext4`
- *Débit garanti (GRIO – Guaranteed rate I/O)* : fluidité du débit d'information (applications multimédia)
- *Chiffrement* : répertoire (utilitaire `encfs`), périphériques (utilitaire `cryptsetup`)
  `xfs`
- ...

Administration d'un système de fichiers

Principales commandes :

- Création : `mkfs`, etc...
- Affichage des informations : `dumpfs`, `fstyp`...
- Vérification : `fsck`
- Montage/Démontage : `mount`/`umount`
- Agrandissement : fonctionnalité disponible dans l'utilitaire de repartionnement, ou associée au système de fichiers (LVM)
- Modification des paramètres d'un système de fichiers : `tune2fs` (`ext2`, `ext3`, `ext4`)

Vérification d'un système de fichiers

5 phases pour vérifier 5 types d'informations :
(`/sbin/fsck /dev/sda1`)

- Table des inodes : suppression de inodes défectueux
- Structures des répertoires : destruction des structures défectueuses
- Connexion au répertoire père : rebranchement dans l'arborescence dans le répertoire `lost+found`
- Nombre de référence : réintégration des fichiers sans nom et mise à jour des liens
- Groupes de cylindres : mise à jour des listes de chaque groupe de cylindres

Organisation des répertoires du système d'exploitation

- / : racine de l'arborescence
- /boot : Noyau et configuration du noyau
- /dev : Périphériques
- /lost+found : Répertoire contenant les blocs et fichiers "perdus"
- /etc : Fichiers de configuration
- /lib : Librairies nécessaires au fonctionnement minimal du système (*single user*)

Organisation des répertoires du système d'exploitation (1)

- `/bin` : Exécutables nécessaires au fonctionnement minimal du système (*single user*) et complètement exploitables par un utilisateur
- `/sbin` : Exécutables système nécessaires au fonctionnement minimal du système (*single user*), avec accès ou utilisation restreint pour un utilisateur
- `/home` : Répertoires de connexion des utilisateurs
- `/root` : Répertoire de connexion du `root` (super utilisateur)

Organisation des répertoires du système d'exploitation (2)

- `/proc` : Répertoire contenant les processus s'exécutant sur le système et leur description
- `/sys` : Répertoire contenant les processus s'exécutant
- `/tmp` : Répertoire des fichiers temporaires sur le système et leur description (version évoluée)
- `/usr` : Répertoire contenant les applications supplémentaires généralement liées à des paquetages (contient les répertoires `bin`, `sbin`,`lib`, `share`)
- `/mnt` : Points de montage

Organisation des répertoires du système d'exploitation (3)

- `/var` : Répertoires dont le contenu varie pendant la session courante du système ou contenant des données sensibles (`/var/cache`, `/var/lib`), `/var/lock`, `/var/log`, `/var/mail`, `/var/run`, `spool`, `/var/tmp`, `/var/www`)
- `/opt` : Utilitaires et Applications supplémentaires quelconques
- `/usr/local` : Répertoire contenant les applications supplémentaires quelconques (contient les répertoires `bin`, `sbin`, `lib`, `share`)

Mémoire virtuelle

(*zone de swap*)

- Utilisation de l'espace disque comme mémoire virtuelle en supplément de la mémoire centrale
- Différentes manière de réserver la zone de swap :
  - Partition dédiée
  - Fichier local
  - Système de fichiers
  - Fichier distant

Gestion de la mémoire virtuelle

- Manipulation au niveau du noyau
- Sous Linux, la zone de swap n'est pas indispensable mais vivement conseillée (en principe, 2 fois la taille de la mémoire centrale)
- Opérations :
  - Détermination de la taille : `/sbin/swapon -s` – DEC OSF1, `/etc/swapinfo` – HP-UX, `/bin/free` – Linux,
    `/usr/sbin/swap -l` – Solaris
  - Ajout : `/etc/swapon`
  - Destruction : Impossible sur la plupart des systèmes, peu conseillée sur Linux, `/usr/sbin/swap -a` sous Solaris

Administration des utilisateurs

- Chaque utilisateur doit être défini sur la machine pour pouvoir l'utiliser
- Création d'un compte utilisateur par un administrateur (le super-utilisateur)
- Caractéristiques du super-utilisateur :
  - `root`
  - UID=0
  - tous les droits lui sont attribués
  - shell privilégié avec `su`

Caractéristiques d'un utilisateur (1)

- Identifiant : règles d'attribution dépendant de l'administrateur (nom de l'utilisateur, initiales, etc.)
- Mot de passe (8/10 caractères en général) : séquence complexe ne se trouvant dans aucun dictionnaire
- UID (valeur entière entre 0 et 65535 – SVR4 $2^{32} - 1$) : identification unique de l'utilisateur.
  - 0 à 999 : Comptes systèmes (`bin`, `daemon`, etc)
  - à partir de 1000 : Utilisateurs

Caractéristiques d'un utilisateur (2)

- GID primaire (entre 0 et 65535, >= 1000 en général) : groupe de l'utilisateur à la connexion
- Commentaires
- Répertoire de connexion
- Programme exécuté au login : un shell se trouvant sur la partition /

Gestion des comptes

- Gestion spécifique des utilisateurs au niveau de la sécurité
- Récapitulation de l'ensemble de ces informations : `/etc/passwd`
- Mot de passe pouvant être déporté dans `/etc/shadow` pour System V (uniquement lisible par le `root`)
- Opérations d'administration : Création, Suppression, Modification des propriétés
- Ne pas supprimer les comptes systèmes (`bin`, `daemon`, `sys`, `root`...)
- Éviter les modifications sans vérifier leurs incidences

Création d'un compte utilisateur

- Récupération et détermination des informations nécessaires à leur création
- Répertoire de création : `/export/home/`*login*, `/home/`*login*, etc
- Commande de création : `useradd`, `adduser` $\longrightarrow$ Ajout d'une entrée dans le fichier `/etc/passwd`
- Environnement initial : recopie des fichiers se trouvant dans `/etc/skel` :
  - famille `sh` : lecture des fichiers `/etc/profile`, `/.profile`
  - famille `csh` : lecture des fichiers `/etc/csh.login`, `/etc/csh.cshrc`, `/etc/.login`, `/etc/.cshrc`, `/etc/.login`

Suppression d'un utilisateur

Suppression :

- des fichiers utilisateurs (répertoire de connexion) et sauvegarde (sur bande, par exemple)
- de la boîte aux lettres et sauvegarde
- des alias courrier
- des tâches d'impression et quotidienne (`cron`, `at`)
- de l'entrée dans `/etc/passwd`
- du *username* des groupes dans lesquels il apparaît

Modification des propriétés

Modification des propriétés répertoriées dans `/etc/passwd`

- Modification du nom complet : `chfn`
- Modification du shell de login : `chsh`
- Modification du mot de passe : `passwd`

Gestion des groupes

- Attribution à chaque utilisateur d'un groupe primaire
- Possibilité de partage de répertoire
- Informations sur les groupes dans `/etc/group` :
  - Nom du groupe
  - Mot de passe (optionnel)
  - GID (entre 0 et 65535)
  - Membres du groupe

A priori, pas de modification des groupes créés lors de l'installation du système (`bin`, `sys`, `daemon`...)

Démarrage d'une station de travail (1)

Trois états possibles pour une machine UNIX :

- Le système d'exploitation n'est pas actif (`telinit 6` ou `telinit 0`) :
  Après la phase d'arrêt ou d'allumage (Machine en mode EEPROM ou sur le Bios)
  Pas de processus lancé (possibilité de redémarrer le système)
  Possibilité de test et de réglage

Démarrage d'une station de travail (2)

- Mode *Single-user* ou maintenance (`telinit 1`) :
  Le système est chargé et partiellement initialisé
  Seul le `root` peut intervenir
  Pas d'autorisation de connexion pour les utilisateurs
- Mode multi-utilisateurs (`telinit 3` ou `telinit 5`) : Initialisation totale du système
  Tous les processus nécessaires sont lancés
  Autorisation de connexion pour tous les utilisateurs

Procédure de démarrage

1. Chargement des programmes de boot (chargeur primaire)
2. Initialisation du noyau (chargeur secondaire) : tests matériel
3. Démarrage du processus `init` : exécution de différentes tâches et passage dans un mode ou un *run-level*
   - BSD : 3 modes (*single-user*, *multi-user*, *poweroff*)
   - System V : une dizaine de modes

Mode *single-user*

Mode commun à tous les UNIX
Passage dans un mode de type maintenance (*logiciel*)
Outils de base pour administrer un système défectueux
Pas d'initialisation des services réseaux

Mode multi-utilisateurs

Lancement de tous les services locaux et réseaux
Suivant les UNIX, possibilité de 2 niveaux multi-utilisateurs :

- Pas de possibilité d'être serveur
- Ensemble de fonctionnalités

Configuration de la procédure de boot

- BSD : utilisation de scripts `rc.*` situés dans `/etc/rc.d`
  - Chaque script est composé de sections lançant les services
  - Consultation des scripts de manière statique par `init`
- System V : utilisation de *run-level* à partir du fichier de configuration `/etc/inittab`, `/etc/init/rc-sysinit.conf`
  - Ensemble des scripts regroupés dans le répertoire `/etc/init.d`
  - Lancement des scripts référencés dans l'arborescence `/etc/rc`*N*`.d` où *N* représente le *run-level*

Arrêt et redémarrage du système

- Arrêt des systèmes UNIX à l'aide des commandes `halt`, `shutdown` ou `reboot`
- Passage en mode 0 ou 6 (`telinit`)
  - Vidage des tampons et écriture sur le disque
  - Démontage des systèmes de fichiers
  - Terminaison des processus

  ou lecture des scripts se trouvant dans `/etc/rc[06].d`

Exemple de script de démarrage de service

```
#!/bin/bash
#
# syslog        Starts syslogd/klogd.
#
#
# chkconfig: 2345 12 88
# description: Syslog is the facility by which many daemons use to log \
# messages to various system log files.  It is a good idea to always \
# run syslog.
### BEGIN INIT INFO
# Provides: $syslog
### END INIT INFO

# Source function library.
```

```
. /etc/init.d/functions

[ -f /sbin/syslogd ] || exit 0
[ -f /sbin/klogd ] || exit 0

# Source config
if [ -f /etc/sysconfig/syslog ] ; then
        . /etc/sysconfig/syslog
else
        SYSLOGD_OPTIONS="-m 0"
        KLOGD_OPTIONS="-2"
fi

RETVAL=0

umask 077
```

```
start() {
        echo -n $"Starting system logger: "
        daemon syslogd $SYSLOGD_OPTIONS
        RETVAL=$?
        echo
        echo -n $"Starting kernel logger: "
        daemon klogd $KLOGD_OPTIONS
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/syslog
        return $RETVAL
}
```

```
stop() {
        echo -n $"Shutting down kernel logger: "
        killproc klogd
        echo
        echo -n $"Shutting down system logger: "
        killproc syslogd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/syslog
        return $RETVAL
}
```

```
status() {
        status syslogd
        status klogd
}
restart() {
        stop
        start
}

case "$1" in
start)
start
        ;;
stop)
stop
;;
```

```
status)
rhstatus
;;
restart|reload)
restart
;;
condrestart)
[ -f /var/lock/subsys/syslog ] && restart || :
;;
*)
echo $"Usage: $0 {start|stop|status|restart|condrestart}"
exit 1
esac
exit $?
```

Lancement automatique de processus

Tâches à effectuer régulièrement : sauvegarde, mise à jour de base de données système, observation régulière du système...

Lecture par le démon `cron` du fichier `crontab` :

- System V et BSD récents : 3 types de files d'attente :
  - Exécution différée à une date et une heure précise : `at`
  - Exécution différée dans une file d'attente : `batch`
  - Exécution cyclique : définie dans le fichier `crontab`
- BSD anciens :
  - Exécution cyclique : définie dans le fichier `crontab`
  - Exécution différée à une date et une heure précise : `at`

Exécution différée avec at (1)

```
at -m HEURE
```

- Exécution à une heure précise d'une commande
- La commande envoie un mail à l'utilisateur
- Plusieurs options sont disponibles

Exécution différée avec at (2)

Format de l'heure :

- `HHMM`
- `HH:MM`
- `midnight` (0h)
- `noon` (12h)
- `teatime` (16h)
- `MMJJAA`
- `MM/JJ/AA`
- `JJ.MM.AA`
- `now + X minutes|hours|days|weeks`

Exécution différée avec at (3)

Exécution à une heure précise d'une commande lue sur l'entrée standard :

```
$ at 11:36
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 1 at 2012-01-10 11:36
#--------------------------
$ at now + 1 hour
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 3 at 2012-01-10 12:38
```

Exécution différée avec at (4)

```
$ at 11am tomorrow
warning: commands will be executed using (in order)
                  a) $SHELL b) login shell c) /bin/sh
at> scriptSauvegarde.sh
at> <EOT>
job 4 at 2012-01-11 11:00
```

Lecture de la commande dans un fichier avec l'option `-f` :

```
$ at -f scriptSauvegarde.sh 11:36
```

Exécution différée avec at (5)

Visualisation des commandes en exécution différée avec l'option `-l` ou la commande `atq` :

```
$ at -l
3       2012-01-10 12:38 a monnin
4       2012-01-11 11:00 a monnin
$ atq
3       2012-01-10 12:38 a monnin
4       2012-01-11 11:00 a monnin
```

Exécution différée avec at (6)

Suppression d'une commande avec l'option `-d` ou la commande `artrm`

```
$ at -d 3
$ at -l
4       2012-01-11 11:00 a monnin
$ atrm 4
$ at -l
$
```

Exécution cyclique avec cron (1)

Opération effectuées toutes les minutes par le programme `cron` :

- Examen du répertoire `/var/spool/cron/crontabs`
- Exécution des commandes placé dans les fichiers du répertoire.
  Le nom des fichiers correspond à des utilisateurs locaux ou déclarés sur le réseau
- Envoi d'un mail à l'utilisateur après exécution

Exécution cyclique avec cron (2)

Installation des commandes à exécuter cycliquement à l'aide de la commande crontab :

```
crontab Nom_de_fichier
```

Exemple :

```
$ crontab crontab_test
```

Visualisation de la liste des commandes à exécuter cycliquement :

```
crontab -l [-u nom_de_l'utilisateur]
```

Exemple :

```
crontab -l
```

Exécution cyclique avec cron (3)

Suppression de la liste des commandes à exécuter cycliquement :

```
crontab -r [-u nom_de_l'utilisateur}
```

Exemple :

```
crontab -r
```

Exécution cyclique avec `cron` (4)

Format du fichier `crontab` :

- Les commentaires sont marqués à l'aide du caractère #
- Chaque ligne (non commentée) correspond à une commande à exécuter
- Chaque ligne est composée de 6 colonnes
  - Les cinq premières définissent la date et l'heure d'exécution
  - La sixième colonne contient la commande à exécuter

Exécution cyclique avec cron (5)

- La date et l'heure sont définies de la manière suivante :
  - colonne 1 minute (0-59)
  - colonne 2 heure (0-23)
  - colonne 3 jour du mois (1-31)
  - colonne 4 mois de l'année (1-12)
  - colonne 5 jour de la semaine (0-6, 0 étant dimanche)

Chaque colonne peut contenir le caractère * (n'importe quelle valeur) ou une liste de valeurs séparées par des virgules

Exécution cyclique avec cron (6)

Exemple :

```
0 0 * * * find / -name core -print > /root/diskPicture.lst
5,20,35,50 * * * * /root/script-verif.sh
0 0 1 1 * /root/envoyerMailBonneAnnee.sh
0 19 * * 5 /root/envoyerMailBonWeekEnd.sh
```

Exécution cyclique avec cron (7)

Installation, affichage, suppression :

```
$ crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (crontab_test installed on Thu Jan 29 16:28:28 2011)
0 0 * * * find / -name core -print > /root/diskPicture.lst
5,20,35,50 * * * * /root/script-verif.sh
0 0 1 1 * /root/envoyerMailBonneAnnee.sh
0 19 * * 5 /root/envoyerMailBonWeekEnd.sh
$ crontab -r
$ crontab -l
no crontab for monnin
```

Génération d'un système (1)

Le noyau

- Cœur du système : accès au matériel, à la mémoire, aux systèmes de fichiers
- La plupart des systèmes sont fournis avec un noyau *générique*
- Chargement du noyau en mémoire après exécution du code primaire (`bootstrap`)
- Puis, reconnaissance du matériel et chargement des pilotes

Génération d'un système (2)

Le noyau

- contient les éléments de base du système et permet le chargement de modules :
  - les pilotes
  - les gestionnaires réseaux
  - ...
- Contenu dépendant des constructeurs

Types de noyau

- Noyau statique :
  - Génération du noyau avec tous les modules nécessaires à un moment donné
  - Chargement intégral au démarrage
  - La modification de la configuration du noyau implique un redémarrage de la machine
- Noyau dynamique :
  - Noyau minimal pouvant charger des modules dynamiquement
  - Pas de redémarrage nécessaire (sauf pour certaines modifications de certains paramètres)

Chargement/déchargement de modules sous Linux (1)

- `/sbin/modprobe` : Manipulation des modules chargeables du noyau
  - Chargement d'un module (et des modules en dépendant) : `/sbin/modprobe usb-uhci`
  - Déchargement d'un module (et des modules en dépendant) : `/sbin/modprobe -r usb-uhci`

  Fichier de configuration : `/etc/modules.conf`, `/etc/modprobe.d` définition d'alias (`alias usb-controller usb-uhci`)

  `/sbin/rmmod usb-uhci`

Chargement/déchargement de modules sous linux (2)

- `/sbin/insmod` : installation d'un module chargeable du noyau, passé en argument
  `/sbin/insmod usb-uhci`
- `/sbin/rmmod` : déchargement d'un module du noyau, passé en argument
  `/sbin/rmmod usb-uhci`

Génération d'un nouveau noyau

- Télécharger les sources du noyau utilisé, les dépendances nécessaires à la compilation
- Déterminer les modules nécessaires
- Conserver une version de sauvegarde du noyau actuel
- Description du noyau à générer dans un fichier de configuration (`/usr/sys/conf` – BSD `/etc/system` – System V)
- Compilation du noyau
- Remplacement de l'ancien noyau

Génération d'un nouveau noyau Linux (1)

- Nettoyage des fichiers objets :
  `make clean ; make mrproper`
- Configuration du noyau (et des modules chargeables) :
  `cd /usr/linux/src ; make xconfig` (puis choix des paramètres)
- Génération du noyau :
  `make bzImage` ou `make vmlinuz`
- Génération des modules :
  `make modules`

Génération d'un nouveau noyau Linux (2)

- Installation des modules :
  `make modules_install`
- Installation du nouveau noyau :
  Copie du noyau (situé dans `/usr/src/linux/arch/i386/boot`) dans `/boot` sous un autre nom spécifique
  NB : Éviter les remplacements, toujours conserver le noyau précédent qui fonctionne

Génération d'un nouveau noyau Linux (3)

- ajout d'une nouvelle entrée dans le chargeur de noyau Linux (LiLo – Linux Loader – ou GRUB – GNU GRand Unified Bootloader)
  Nouvelle entrée dans le fichier de configuration de LiLo (`lilo.conf`). Extrait de `lilo.conf` :

```
image=/boot/vmlinuz-test3
        label="linux"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="devfs=mount resume=/dev/hda6 \
                splash=silent"
        vga=791
        read-only
```

  Mise en place de la nouvelle de configuration de LiLo

Génération d'un nouveau noyau Linux (4)

En pratique

→ Suivre les recommandations de sa distribution

Administration des packages

Distribution de logiciels sous forme de packages :

- formats propres aux systèmes d'exploitation
- conformité de Solaris avec l'ABI SRV4 (*Application Binary Interface*)
- trace de l'installation des packages sur le système

Installation automatique des packages lors de l'installation du système
Visualisation, ajout, suppression des packages installés :

- Solaris : `pkginfo`, `pkgadd`, `pkgrm`, `pkgchk`, `admintool`
- Linux : `linuxconf`, `pkgtool`, `Yasp`, `rpm`, `autorpm`, `urpmi`, `apt-get`, `apt-cache`, `dpkg`, `aptitude`

Gestion des logiciels

- Logiciels non standard, non livrés avec le système ou nouvelles versions
- Répertoires : `/usr/local`, `/opt`
- Installation variable : Recopie de fichiers binaires, compilation des sources (utilisation d'`autoconf`, `automake`...)

Exemple d'installation de logiciel (debian)

- A partir d'une archive locale : `dpkg -i nom de l'archive`

  ```
  # dpkg -i zip_2.31-3_i386.deb
  ```

- A partir d'une archive présente dans les dépôts : `apt-get install nom de l'archive`

  ```
  # apt-get install zip
  ```

- Mise à jour des paquetages :

  ```
  # apt-get update
  # apt-get upgrade zip
  ```

- Recherche de paquetage :

  ```
  # apt-cache search zip
  ```

Exemple d'installation à partir des sources

- Exemple : `httpd-2.0.52.tar.gz`

```
tar xzvf httpd-2.0.52.tar.gz
(lecture du fichier INSTALL)
$ ./configure --prefix=/usr/local
$ make
$ make install
$ /usr/local/bin/apachectl start
```

Messages du système (1)

Consignation des opérations et problèmes rencontrés par le système :

- Messages des logiciels : le démon `syslogd` scrute le fichier `/dev/klog`, `/dev/log` et le port 514 (machines distantes)
- Journal : `/var/log/syslog` ou `/var/log/syslog`, et d'autres
- Fichier de configuration : `/etc/syslog.conf`
  - Définition de règles de journalisation : aiguillage des messages systèmes dans différents fichiers (`/var/log/user`, `/var/log/kernel/errors`, `/var/log/mail/info`, etc.)

Messages du système (2)

- Messages d'erreurs matériel :
  - BSD : commande `dmesg`
    Messages rangés dans `/usr/adm/messages`
  - System V : commande `dmesg`
    Messages rangés dans les fichiers présents dans `/var/log/*` (suivant les règles de journalisation de syslog, et les outils système)

Observation des activités du système

Tâche importante pour la sécurité et le confort des utilisateurs (éviter la surcharge du système) :

- Lister les utilisateurs (`who`, `finger`)
- Lister les processus du système (`ps`)
- Gestion des processus (`nice`, `kill`)
- Statistiques d'utilisation des ressources (`statcmd`, `vmstat`, `iostat` – BSD, `sar` – system V)
- Utilisation des disques (`du`, `df`)
- Gestion des IPC (System V) (`ipcs`, `ipcrm`)

Répertoires /proc et /sys

Origine : systèmes Unix, années 80

/proc :

- Présentation des différentes facettes des processus à un moment donné
- Regroupes des informations relatives aux processus dans des répertoires numériques (PIDs)
- Sous Linux : vue du système d'exploitation et du matériel
  Possibilité de paramétrage dynamique

/sys :

- Évolution de /proc
- Complète et réorganise /proc

Présentation générale de /proc (1)

- Contient des informations clés sur l'état du noyau et du système en général
- Varie en fonction des systèmes
- Pseudo système de fichiers : pas de surface de stockage
- Possibilité de montage/démontage
- Arborescence d'objets
- Possibilités d'effectuer les opérations classiques `ls`, `cat`, `cd`...

$\longrightarrow$ Interface de lecture/écriture de variables et structures internes au noyau

Présentation générale de /proc (2)

```
$ ls -l /proc/partitions
-r--r--r-- 1 root root 0 janv.  9 14:02 /proc/partitions
$ cat /proc/partitions
major minor  #blocks  name

3     0   78150744 hda
3     1    5116671 hda1
3     2          1 hda2
3     5    2040223 hda5
3     6    5116671 hda6
3     7   32941251 hda7
3     8   32933218 hda8
```

Contenu de /proc

Quatre types d'objets :

- Répertoires :
  - Nom numérique : représentation des processus
  - Sous-ensemble du système (`scsi`, `sys`...)
- Fichiers réguliers :
  - informations en ASCII
  - exploitable avec des commandes comme `cat`, redirections E/S
- Liens symboliques : `self` et `mounts`, `/proc/PID/exe` (lien vers le binaire)
- Fichiers spéciaux (rares) : correspondent à un périphérique à piloter

Manipulations (1)

Lecture/écriture : appel de fonction ou de méthode associés aux objets implantés en mémoire

```
$ ls -l /proc/cmdline
-r--r--r--  1 root root 0 Jan 23 15:46 cmdline
$ file /proc/cmdline
cmdline: empty
$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-36-generic root=UUID=94e951f3-2cf2-4384-b
```

NB : Pas de possibilité de modifier l'arborescence avec les commandes `mkdir`, `ln`, `touch`...

Manipulations (2)

- Commande `cat` : invocation de l'appel système `read()`
- Suivi de l'évolution du contenu d'un fichier :

```
$ watch cat /proc/loadavg

Every 2.0s: cat /proc/loadavg          Tue Jan 23 15:49:32 2007
0.04 0.05 0.08 2/110 28710
```

Manipulations (3)

- Modification d'objets
  Appel système `write()`

```
$ cat /proc/sys/kernel/threads-max
16378
$ echo 4096 > /proc/sys/kernel/threads-max
$ cat /proc/sys/kernel/threads-max
4096
```

Manipulations (4)

- Chargement/déchargement de modules

```
# ls -l /proc/sys
total 0
dr-xr-xr-x  2 root root 0 Jan 23 16:00 debug/
dr-xr-xr-x  7 root root 0 Jan 23 16:00 dev/
dr-xr-xr-x  5 root root 0 Jan 23 08:42 fs/
dr-xr-xr-x  4 root root 0 Jan 23 15:56 kernel/
dr-xr-xr-x  8 root root 0 Jan 23 16:00 net/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 proc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 vm/
# lsmod |grep sunrpc
# modprobe sunrpc
# lsmod |grep sunrpc
sunrpc                122788  0
# ls -l /proc/sys
total 0
dr-xr-xr-x  2 root root 0 Jan 23 16:00 debug/
dr-xr-xr-x  7 root root 0 Jan 23 16:00 dev/
dr-xr-xr-x  5 root root 0 Jan 23 08:42 fs/
dr-xr-xr-x  4 root root 0 Jan 23 15:56 kernel/
dr-xr-xr-x  8 root root 0 Jan 23 16:00 net/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 proc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 sunrpc/
dr-xr-xr-x  2 root root 0 Jan 23 16:00 vm/
```

Suivi de processus

- Répertoire numérique : pseudo-répertoire correspondant à un processus ou un thread du système
- Contenu :
  - Etat du système : `status`, `stat`, `wchan`, `auxv`
  - organisation de l'espace d'adresses : `statm`, `maps` `mem`
  - contexte d'exécution : `exe`, `cmdline`, `environ`
  - fichiers utilisés : `mounts`, `root`, `cwd`, `fd`
- Informations utilisées par `top` et `ps`

Analyse du système d'exploitation et du matériel (1)

Visualisation de l'utilisation courante des différents composants du systèmes (CPU, mémoire, disque...)

- exploitation du(des) processeur(s) : `/proc/stat`
- état de la mémoire : `/proc/meminfo`
- partitions : `/proc/partitions`
- zone de swap : `/proc/swaps`

Analyse du système d'exploitation et du matériel (1)

Visualisation de l'historique du système

- Paramètres de lancement du noyau : `/proc/cmdline`
- Utilisation de la machine : `/proc/uptime`

Egalement accès à travers des commandes :

- `lspci`, `lsusb`, `lspnp`...
- `uptime`, `free`, `tload`...

Paramétrage dynamique du système (1)

- Possibilité de modifications des fichiers de `/proc/sys` :
  - `echo`, `vi`...
  - `sysctl`
- Modification pour la session en cours
- Ajustement de paramètres de fonctionnement
  - du noyau (`kernel`)
  - de la gestion du système de fichiers (`fs`)
  - de la gestion de la mémoire virtuelle (`vm`)
  - du réseau (`net`)

Paramétrage dynamique du système (2)

Utilisation de `echo` :

```
$ cat /proc/sys/kernel/threads-max
16378
$ echo 4096 > /proc/sys/kernel/threads-max
$ cat /proc/sys/kernel/threads-max
4096
```

Utilisation de `sysctl` :

```
# sysctl kernel.threads-max
kernel.threads-max = 16378

# sysctl  -w kernel.threads-max=4096
# sysctl kernel.threads-max
kernel.threads-max = 4096
```

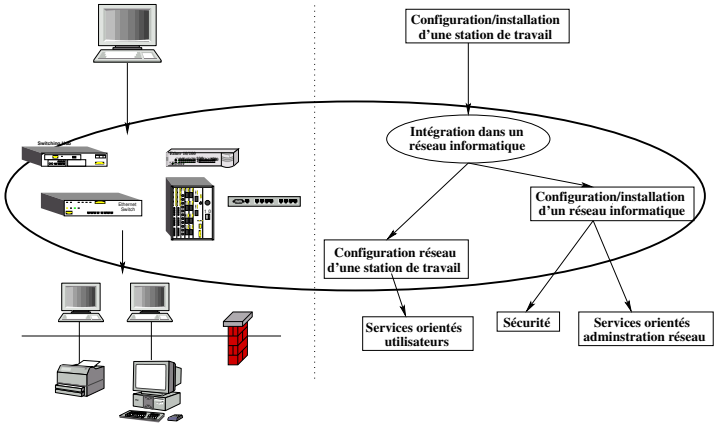
Présentation générale de /sys

- Nouveau système de fichiers orienté vers la description du matériel
- Publication de l'arborescence des composants matériels et des périphériques logiciels
- Similaire à /proc

Contenu de /sys

- `block` : utilisation des périphériques en mode bloc
- `bus` : énumération des différents bus de la machine
- `class` : organisation des périphériques suivant leur fonction
- `devices` : hiérarchisation des composants, identifiés en fonction de leur position dans le bus
- `firmware` : ACPI (gestion de l'énergie), EDD (disques visible par le BIOS)
- `power` : gestion d'énergie

Conception du réseau



Réseau : rappels

Protocoles sur les systèmes UNIX : TCP/IP et UDP/IP

- IP : Interconnexion des réseaux et routage des paquets
  Supporté par plusieurs couches physiques dont Ethernet
- TCP et UDP : protocoles de transport en mode connecté/non connecté s'appuyant sur les services de la couche IP

Interface Ethernet/IP

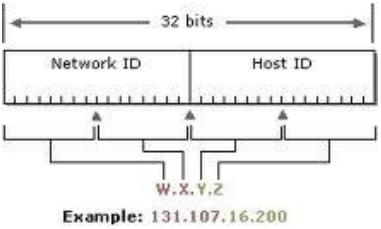
Communication entre deux machines à travers l'interface physique ethernet :

- IP : Protocole de convergence (Ethernet, PPP, ATM, ...)
- Applications : uniquement connaissance des adresses IP
- Établissement d'une correspondance adresse IP / Adresse physique Ethernet (MAC)
- Utilisation des protocoles ARP (Adress Resolution Protocol) et RARP (Revers ARP)
- Interrogation/manipulation du cache ARP/RARP au niveau du système d'exploitation : `arp`, `rarp`

Adresse IPv4

- 4 octets (32 bits) : notation décimale pointée A.B.C.D (par ex. 194.254.167.1)
- Unique au monde :
  - Configuration par logiciel
  - Associée au chaque interface réseau
- Plusieurs classes : A, B, C, D, E
  Attribution :
  - classes A et B : le RIPE (Réseau IP Européen)
  - classe C : en France le NIC

Adressage IPv4



- *Adresse réseau* (***network id***) :
  - correspond à la classe
  - détermine le réseau de la machine
  - assigné par une autorité nationale ou internationale
- *Adresse du host* (***host id***) :
  - correspond au masque réseau
  - détermine la machine sur le réseau
  - assignée par l'administrateur du réseau

Classes d'adressage (1)

- Classe A : 1 octet $\rightarrow$ réseau, 3 octets $\rightarrow$ machine
  - $2^{24} - 2 >$ 16 millions de postes
  - Premier octet compris entre 0 et 127
  - Premier bit toujours à 0
  - Ex : 48.27.49.13

- Classe B : 2 octet $\rightarrow$ réseau, 2 octets $\rightarrow$ machine
  - $2^{16} - 2 >$ 65534 postes
  - Premier octet compris entre 128 et 191
  - Deux premiers bits toujours égaux à 10
  - Ex : 131.16.1.23

Classes d'adressage (2)

- Classe C : 3 octet $\rightarrow$ réseau, 1 octets $\rightarrow$ machine
  - $2^8 - 2 > 254$ postes
  - Premier octet compris entre 192 et 223
  - Trois premiers bits toujours égaux à 110
  - Ex : 194.254.167.1
- Classe D : multicast
  - Premier octet compris entre 224 et 239
  - Quatre premiers bits toujours égaux à 1110
- Classe E : réservé IANA, adresses comprises entre 240.0.0.0 et 255.255.255.255

Adresses particulières

- 127.0.0.1 : loopback (pseudo-réseau), localhost
  Utilisé pour les tests logiciels et les communications internes inter-processus
- Adresse du réseau : tous les bits de la partie machine à 0
  194.254.167.0 désigne le réseau de classe C 194.254.167
- Adresse de diffusion (broadcast IP) : tous les bits de la partie machine à 1
  - 194.254.167.255 désigne toutes les machines du réseau 194.254.167.0
  - Permet la recherche d'une machine offrant un service, dont l'adresse est inconnue (serveur NIS, actualisation de la table de routage...)

Adresses particulières : les réseaux privés

Dans chaque classe d'adresse suivantes, il existe des plages particulières, dites « privées »

- Classe A : 10.0.0.0 à 10.255.255.255
- Classe B : 172.16.0.0 à 172.31.255.255
- Classe C : 192.168.0.0 à 192.168.255.255

Ces adresses ne sont pas directement utilisables sur Internet (non routées/routables) et ne peuvent donc servir que pour des réseaux locaux.

Masque de réseau et de sous-réseau

Le masque de **réseau** :

- Il s'écrit de la même manière qu'une adresse IP (4 octets en notation décimale pointée)
- Il permet de déterminer le réseau auquel appartient une adresse IP en faisant une opération binaire ET entre l'adresse et le masque.
- Il est déterminé par la classe d'adresse IP utilisée.

Le masque de **sous-réseau** :

- Il permet la subdivision logique d'un réseau de taille plus importante.
- Il est utilisé pour le routage au sein d'un même réseau.

Ces deux usages sont aujourd'hui généralement confondus : les réseaux ne sont que rarement utilisés d'un seul bloc.

Masque : Notation CIDR

La notation CIDR d'un masque permet son écriture sous une forme beaucoup plus courte que sa version décimale pointée. Elle correspond au nombre de bits du masque. Par exemple :

- /24 correspond à 255.255.255.0
  ou 11111111.11111111.11111111.00000000 sous forme binaire
- /16 correspond à 255.255.0.0
  ou 11111111.11111111.00000000.00000000 sous forme binaire
- /8 $\rightarrow$ 255.0.0.0
- ou encore /19 à 255.255.240.0

Familiarisez-vous avec cette notation, elle est régulièrement utilisée...

Administrer un réseau

- Concevoir (préliminaires) :
  - Plan du réseau
  - Mise en œuvre
- Assurer le bon fonctionnement :
  - Surveillance
  - Dépannage
- Offrir des services aux utilisateurs (ressources numériques, messagerie, stockage...)
- Recueillir les informations nécessaires à l'évolution du réseau

Conception du réseau (1)

- Réflexion sur l'utilisation du réseau
- Identifier les contraintes matérielles, financières
  $\longrightarrow$ Influence sur les choix techniques
- Quelle organisation matérielle et humaine ?

Une grande variabilité suivant le site

Conception du réseau (2)

- Plan du réseau
- Topologie et architecture
- Plan d'adressage, de nommage et de routage
- Architecture des services réseaux (Messagerie, Annuaire, DNS, etc...)
- Organisation des ressources humaines

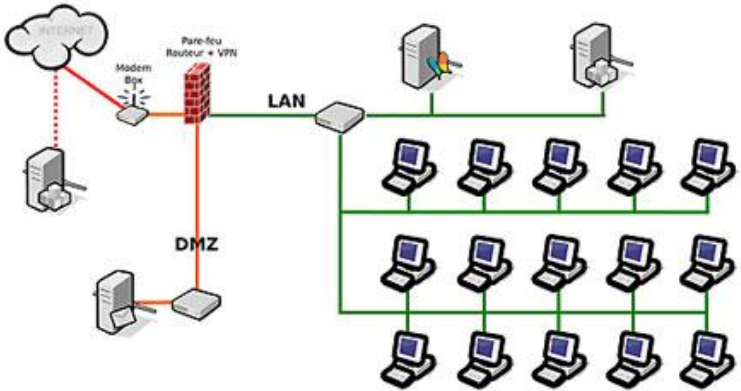
Plan du réseau

Élaboration du plan d'ensemble du réseau :

- Recenser les besoins actuels des utilisateurs
- Évaluer leurs besoins futurs
- Inventorier les ressources existantes
- Concevoir un réseau robuste aux évolutions

$\longrightarrow$ Éviter l'hétérogénéité des équipements

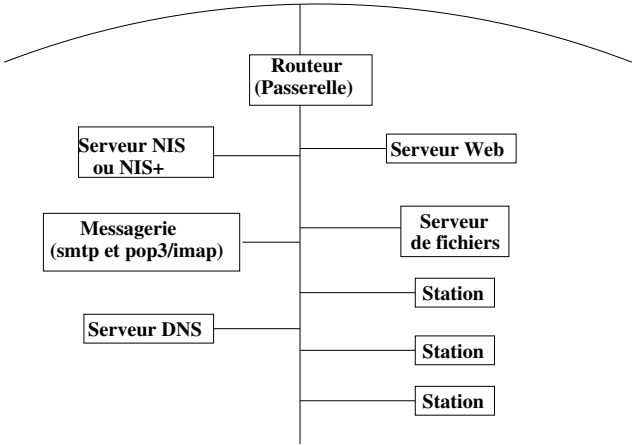
Exemple de plan de réseau



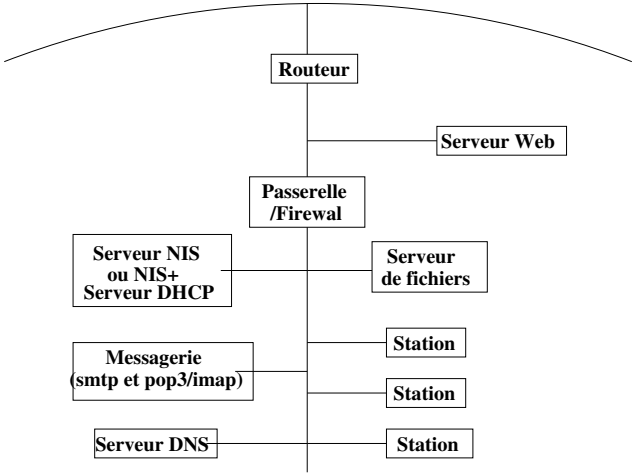
Organisation des ressources humaines

- Recenser les besoins en personnel en fonction de la taille du réseau, de la répartition géographique et des services nécessaires
- Définir la répartition des tâches
- Désigner une instance d'arbitrage
- Informer les utilisateurs sur les personnes à contacter
- Habituer les utilisateurs à contacter la personne compétente pour résoudre leur problème

Exemple de réseau public



Exemple de réseau privé



Plan d'adressage (1)

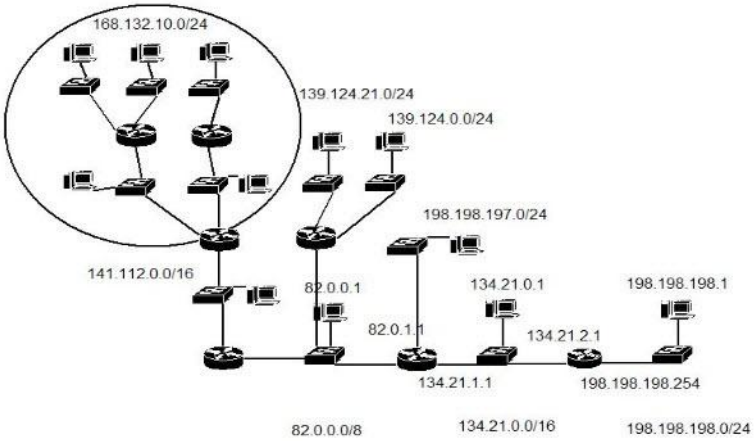
1) Site isolé : définir le plan d'adressage
   - Regroupement des entités qui peuvent l'être (économie de réseaux de classe C)
   - Découpage des réseaux de classe C en sous réseaux (facilité l'administration)
   - Définition des plages d'adresses par site, par entité, par protocole

Plan d'adressage (2)

2) Site proche d'un site existant
- intégration dans le site tout en conservant une autonomie :
  - Partage de plages d'adresses
  - Partage des informations et de l'expérience
- Nécessité d'indépendance et d'isolement du site
  Réseau privé plutôt que public :
  - Sécurité
  - Économie d'adresses publiques et routables

Mise en œuvre : Réservation des numéros de réseaux de la classe

Exemple de plan d'adressage



Plan de nommage

- Choix du nom de domaine pour le site ou le groupe de sites
- Concertation avec les administrateurs du réseau dans lequel s'intègre le réseau

$\longrightarrow$ Hiérarchisation du nommage pour les sites importants ou les structures distinctes (création de sous-domaine)

- Mise en œuvre : Réservation du nom de domaine (NIC en France)

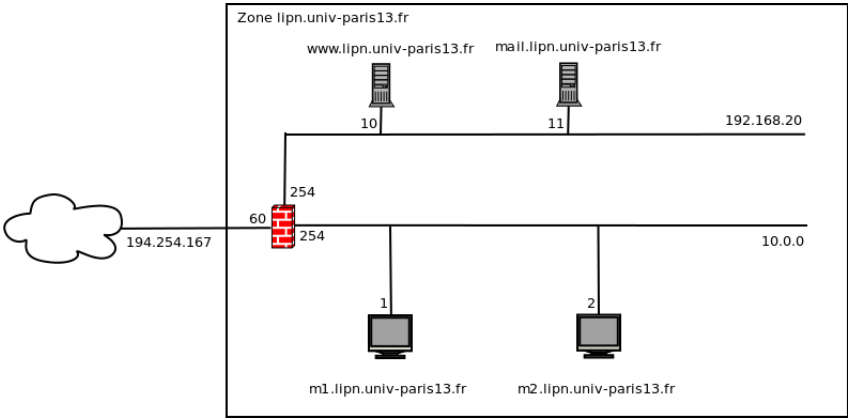
Noms de domaine

- Facilité d'emploi
- Manipulation de noms symboliques plutôt que d'adresses sur 32 bits
- Association adresse IP / nom principal et secondaire (cf. `/etc/hosts`)

$\longrightarrow$ Solution non satisfaisante avec l'explosion des réseaux

- Vers un modèle distribué : à partir de la manière de nommer les machines
- Une machine appartient à un réseau/domaine, un sous-réseau/sous-domaine : son nom en découle
- L'adresse IP d'une machine est connue en interrogeant le DNS du réseau (chaque administrateur gère ses machines sur son DNS)

Exemple : lipn.univ-paris13.fr



Routage

- Transmettre les informations d'un réseau (segment) à un autre

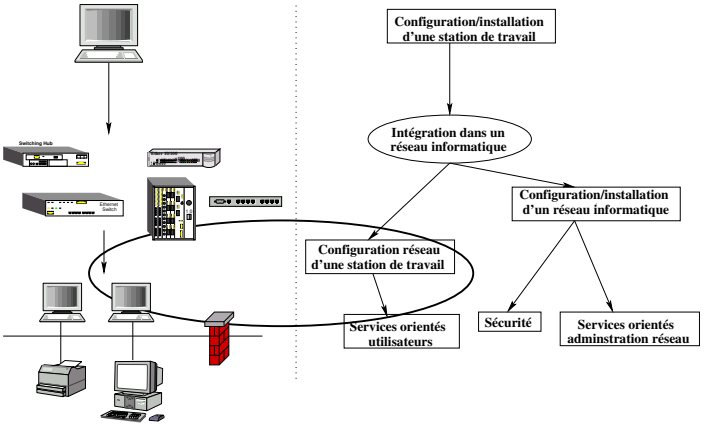
$\longrightarrow$ Trouver un chemin (une route) vers la destination finale à travers des relais (passerelles)

- Table de routage : Tableau contenant les passerelles permettant d'accéder aux réseaux connus
- Types de route :
  - vers une machine simple
  - vers un sous-réseau entier
  - vers une route par défaut (`default`)

Mise en œuvre du plan de routage

- Réservation d'un numéro de Système Autonome auprès de l'ICANN (www.afnic.fr pour le `.fr`)
- Préparation des configurations des équipements de routage
- Sauvegarde sur les stations de travail

Intégration au réseau



Configuration de l'interface réseau (1)

(Possibilité de configuration du réseau à l'installation)
Dépendant des systèmes d'exploitation, mais en général :

- `/etc/hosts` : Association d'adresse IP et de noms de machine (hostname)

```
127.0.0.1          localhost
194.254.163.3      mail
10.10.0.105        lipn-maple  lipn-maple.lipn.univ-paris13.fr
```

Noms particuliers :

- *localhost* : nom par défaut de la machine
- *mailhost* : utilisé par le logiciel *sendmail*

Configuration de l'interface réseau (2)

- `/etc/networks` : Association d'adresses de réseaux et de noms

```
loopback          127
arpanet           10                arpa     # Historical
ig-edu.univ-paris13.fr              194.254.167
```

  Nom particulier : `loopback`, réseau par défaut

- `/etc/protocols` : Association du numéro de protocole à des noms (udp, icmp, etc.)

```
ip      0       IP      # internet protocol, pseudo protocol number
icmp    1       ICMP    # internet control message protocol
tcp     6       TCP     # transmission control protocol
egp     8       EGP     # exterior gateway protocol
udp     17      UDP     # user datagram protocol
```

Configuration de l'interface réseau (3)

- `/etc/services` : Association de numéro de service à des noms

```
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
```

Configuration de l'interface réseau (4)

- commande `/sbin/ifconfig` : configure et affiche les interfaces réseaux
  `/sbin/ifconfig eth0 <ADRESSE IP> netmask <MASQUE RESEAU> up`
- Exemple :

```
/sbin/ifconfig eth0 192.168.0.12 \
broadcast 192.168.0.255 netmask 255.255.255.0
```

Configuration de l'interface réseau (5)

- `inetd` et `/etc/inetd.conf`
  Contrôle d'un ensemble de démons (`rlogind`, `rshd`, `ftpd`, `telnetd`, etc...)

`/etc/inetd.conf`

```
# Ftp and telnet are standard Internet services.
#
ftp     stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp  nowait  root  /usr/sbin/tcpd  in.telnetd
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell   stream  tcp  nowait  root  /usr/sbin/tcpd  in.rshd
login   stream  tcp  nowait  root  /usr/sbin/tcpd  in.rlogind
```

Configuration de l'interface réseau (6)

- Fichier(s) de configuration de l'interface réseau :
  Possibilité d'affectation de valeurs aux variables de configuration (masque réseau, adresse IP...)
  Installation de l'interface à l'aide de la commande `ifconfig`
  Dépendant des (types de) systèmes :
  - Solaris : `/etc/init.d/inetsvc` et `/etc/init.d/rootusr` (liens dans les répertoires `/etc/rc[0-6]/`)
  - Linux : `/etc/init.d/networking`
  - AIX : `/etc/rc.net`

Configuration de l'interface réseau (7)

- Lancement des démons internet :
  - `inetd/xinetd` : gestion des ports et des services
  - `routed` : gestion des routes
  - `rwhod` : gestion des utilisateurs présents sur le réseau local

  Tous ne sont pas obligatoirement lancés pour des raisons de sécurité

Configuration de l'interface réseau (8)

- Équivalence de machines :
  (mécanisme d'autorisation des *r-commandes* :
  `rlogin`, `rsh`, `rcp`, `rdump`, `rrestore`)
  Échec si pas d'autorisation mise en place sauf pour `rlogin`
  Mise en place des autorisations :
  - Déclaration de machines clientes dans `/etc/hosts` sur le serveur
  - Déclaration des autorisations concernant les utilisateurs dans le fichier `/etc/hosts.equiv`

Configuration de l'interface réseau (9)

Mise en place des autorisations (suite) :

- Autorisation de l'environnement de l'utilisateur avec `.rhosts` (dans leur répertoire `HOME`)
- Référencement dans `.rhosts` ou `/etc/hosts.equiv`

Exemples de fichier `.rhosts` :

```
nantes monnin
bourbaki monnin
painleve monnin
```

Contraintes supplémentaires au niveau du propriétaire, des droits et de la nature du fichier

Configuration de l'interface réseau (10)

- fichier /etc/network/interfaces

/etc/network/interfaces

```
auto lo eth0
iface lo inet loopback

iface eth0-home inet static
   address 192.168.1.21
   netmask 255.255.255.0
   gateway        192.168.1.254
```

- Redémarrage du service :

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

Installation de routes (1)

- Initialisation de la table de routage : `/sbin/ifconfig`
  Création d'une route vers son propre réseau (la machine est sa propre passerelle)

Installation de route (2)

- Ajout de route : commande `route`
  Routage statique :

```
route add host 192.33.182.68

route add net 192.33.182.0 0 gw 192.33.182.68

route add default 192.168.0.0 1
```

  Routage dynamique : démon `routed`, `gated`, etc.
- Suppression de route :

```
route del 192.33.182.0
```

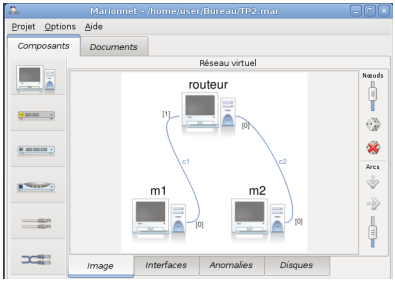
Configuration des routes

Après l'installation des interfaces ethernet
Dépendant du système :

- Solaris : `/etc/init.d/inetinit` avec consultation du fichier `/etc/defaultrouter`
- Linux : `/etc/rc.d/rc.inet1`
- HP-UX : `/etc/netlinkrc`

Création d'une passerelle sous Linux

- Machine Linux avec deux interfaces réseaux
- Activation du forwarding de paquets :
  `echo 1 > /proc/sys/net/ipv4/ip_forward`



- Configuration pour que les passerelles soient le routeur

Visualisation des routes (1)

Commande `traceroute` : Envoi de paquet du protocole ICMP
Récupération des réponses de chaque passerelle
Visualisation des routes définies sur la machine :
`/bin/netstat -r -n`
Linux :

```
$ netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.0.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG       40 0          0 eth0
```

Visualisation des routes (2)

Routes vers d'autres réseaux :

```
# netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.33.182.0    192.33.182.68   255.255.255.0   UG        40 0          0 eth0
192.33.182.0    0.0.0.0         255.255.255.0   U         40 0          0 eth0
10.10.0.0       0.0.0.0         255.255.0.0     U         40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         40 0          0 lo
0.0.0.0         192.33.182.254  0.0.0.0         UG        40 0          0 eth0
```

Visualisation des routes (3)

Visualisation des relais (les passerelles) : `traceroute`

```
traceroute to armen.biomath.jussieu.fr (134.157.72.23), 30 hops max, 40 byte
packets
 1  gw5-r.univ-paris13.fr (194.254.170.254)  2,792 ms  3,678 ms  1,718 ms
 2  195.83.240.205 (195.83.240.205)  8,323 ms  5,173 ms  5,505 ms
 3  aubervilliers1.rerif.ft.net (193.48.58.173)  221,376 ms  146,756 ms  131,263 ms
 4  stamand2.rerif.ft.net (193.48.53.189)  137,446 ms  210,720 ms  135,917 ms
 5  peer-renater.rerif.ft.net (193.48.53.217)  117,014 ms  148,342 ms  209,945 ms
 6  nio-n1.cssi.renater.fr (193.51.206.21)  150,090 ms  234,662 ms  151,598 ms
 7  jussieu.cssi.renater.fr (194.214.109.6)  167,079 ms  147,479 ms  185,449 ms
 8  rap-jussieu.cssi.renater.fr (193.51.12.78)  174,031 ms  229,732 ms *
 9  jussieu.rap.prd.fr (195.221.126.33)  229,504 ms  182,648 ms  183,462 ms
10  r-ps.reseau.jussieu.fr (134.157.254.3)  215,697 ms  249,651 ms  206,939 ms
11  r-biomath.chups.jussieu.fr (134.157.192.33)  132,791 ms  225,918 ms  214,403 ms
12  armen.biomath.jussieu.fr (134.157.72.23)  181,951 ms  290,827 ms  158,506 ms
```

Contrôle du réseau (1)

- /usr/sbin/ping
  Envoi d'un paquet avec écho à la machine spécifiée et, notification de la réception du paquet (protocole ICMP)
- Exemples :

```
$ ping 192.168.0.12
PING 192.168.0.12: 56 data bytes
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=0. time=2. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=1. time=18. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=2. time=27. ms
64 bytes from lipn.up13.fr (192.168.0.12): icmp_seq=3. time=9. ms
^C
----192.168.0.12 PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 2/14/27
```

Contrôle du réseau (3)

- `/sbin/ifconfig`
  Option `-a` : liste les interfaces présentes sur la machine

```
eth0      Link encap:Ethernet  HWaddr 00:04:76:94:07:A3
          inet addr:192.168.0.26  Bcast:192.168.0.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6165981 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18422117 errors:0 dropped:0 overruns:0 carrier:2
          collisions:4176156 txqueuelen:100
          RX bytes:2059858794 (1964.4 Mb)  TX bytes:2577606712 (2458.1 Mb)
          Interrupt:11 Base address:0xcc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:97723 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97723 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:83338668 (79.4 Mb)  TX bytes:83338668 (79.4 Mb)
```

Contrôle du réseau (4)

- `/usr/bin/netstat`
  Affichage des états des différents composants du réseau sur la machine locale
- Exemples :

```
$ netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR   TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0 6210183      0      0      0 018576108      0      0      0 BMRU
lo    16436   0   98687      0      0      0   98687      0      0      0 LRU
```

Contrôle du réseau (5)

```
$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:login                 *:*                     LISTEN
tcp        0      0 *:shell                 *:*                     LISTEN
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:telnet                *:*                     LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  3      [ ]         DGRAM                    103    /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     152    /dev/printer
unix  2      [ ACC ]     STREAM     LISTENING     195    /dev/gpmctl
unix  2      [ ACC ]     STREAM     LISTENING     24775  /tmp/.X11-unix/X0
```

Auscultation d'un réseau Ethernet (1)

Opérations dépendantes du système d'exploitation

DANGEREUX !

Nécessite

- le support du mode *promiscuous* : accès du niveau programmation à tous les paquets Ethernet
  Certains UNIX ne proposent pas ce mode *dangereux* pour la sécurité (possibilité de visualiser tous les paquets passant sur le réseau)
- le passage de l'interface en mode *promiscuous*

Auscultation d'un réseau Ethernet (2)

Capture des paquets (sniffers, bas niveau) : `/usr/sbin/snoop` sur Solaris, `tcpdump` sur Linux
Exemple de paquets capturés :

```
10.10.0.85 -> (broadcast)  ARP C Who is 10.10.0.22, 10.10.0.22 ?
    fwlipn -> lipn         NFS C GETATTR3 FH=0095
      lipn -> fwlipn       NFS R GETATTR3 OK
    fwlipn -> lipn         NFS C ACCESS3 FH=0095 (lookup)
      lipn -> fwlipn       NFS R ACCESS3 OK (lookup)
    fwlipn -> lipn         NFS C LOOKUP3 FH=0095 anass
      lipn -> fwlipn       NFS R LOOKUP3 OK FH=79B1
    fwlipn -> lipn         NFS C GETATTR3 FH=79B1
      lipn -> fwlipn       NFS R GETATTR3 OK
      lipn -> umr7030      TCP D=1134 S=22 Ack=865118141 Seq=1503225955 Len=284 Win=8760
         ? -> (multicast)  ETHER Type=0001 (LLC/802.3), size = 50 bytes
        ls -> BROADCAST    UDP D=161 S=4195 LEN=64
c.up13.fr -> (broadcast)  ARP C Who is 192.33.182.5, 192.33.182.5 ?
cc5.up13.fr -> 192.33.182.173 IP  D=192.33.182.173 S=194.254.164.5 LEN=34, ID=27277
```

Gestion des incidents

Pas de réseau : pourquoi ?
Plusieurs paramètres à évaluer :

- Problème matériel : carte, câble, machine allumée ou éteinte
- Interroger les machines distantes : `/usr/sbin/ping`
  - Interrogation par nom (problème de DNS)
  - Interrogation par adresse IP (problème de route, matériel)

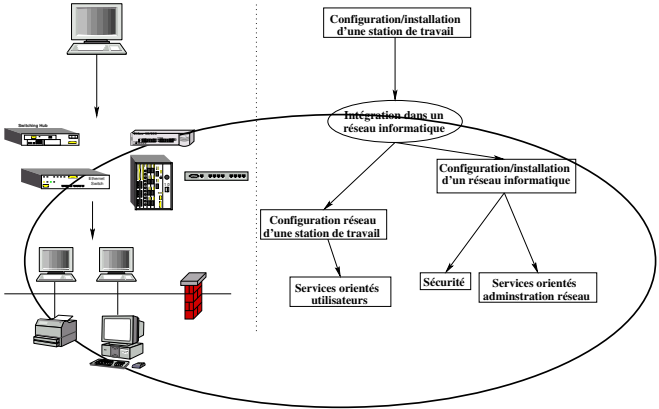
Mise en place de la station dans un réseau

- Configuration de l'interface réseau
- Configuration de la résolution de noms
- Installation de la connexion :
  - vers son serveur de compte utilisateur
  - vers son(ses) serveur(s) de données/stockage
- Installation de la messagerie
- Installation des services annexes

5 Services pour l'administration / les utilisateurs

- DHCP
- DNS
- NIS
- LDAP
- NFS
- Samba
- Telnet
- SSH
- FTP
- NTP

Services orientés administration



Services orientés administration

- Serveur de nom du domaine : DNS
- Annuaires : NIS, NIS+, LDAP
- Autres services : DHCP, Samba, Telnet, SSH, FTP, Serveur de temps (NTP)

Concepts d'annuaires

- Problème de l'administration système :

  Assurer la consistance des fichiers de base du système

- Solution : centraliser les informations

- Utiliser une structure client/serveur
  - Serveur : centralise les informations
  - Client : ne possède aucune donnée localement (ou un minimum) et demande les informations au serveur

DHCP : Dynamic Host Configuration Protocol

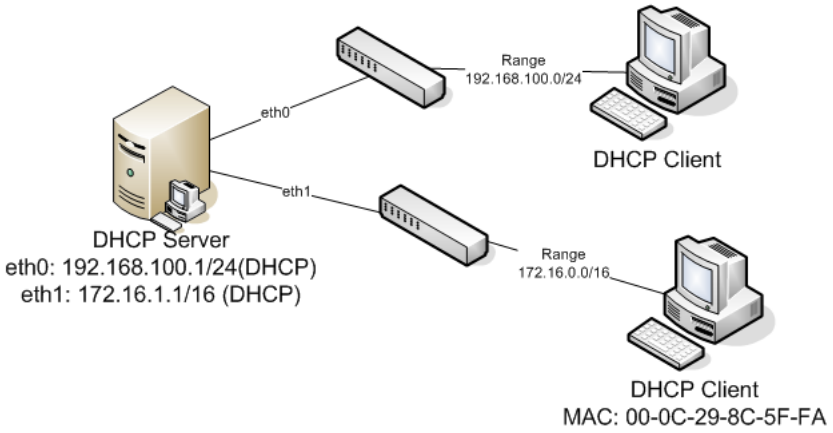
(Protocole de configuration dynamique de machines)

- Distribution de paramètres de configuration réseau (notamment adresse IP) par un serveur
- Obtention par une machine cliente de ses paramètres réseau auprès du serveur pour un temps donné (bail)
- Compatibilité ascendante avec BOOTP

Architecture logicielle :

- Serveur : démon `/usr/sbin/dhcpd`
- Client : processus `/sbin/dhcpcd`

DHCP


Range
192.168.100.0/24
eth0
DHCP Client
eth1
DHCP Server
eth0: 192.168.100.1/24(DHCP)
eth1: 172.16.1.1/16 (DHCP)
Range
172.16.0.0/16
DHCP Client
MAC: 00-0C-29-8C-5F-FA

Protocole DHCP (1)

Requêtes :

- `DHCPDISCOVER` : Localisatoin des serveurs DHCP disponibles
- `DHCPOFFER` : Réponse du serveur à un paquet DHCPDISCOVER – contient les premiers paramètres
- `DHCPREQUEST` : Requête quelconque du client – par exemple prolongement du bail
- `DHCPACK` : Réponse du serveur – contient des paramètres et l'adresse IP du client
- `DHCPNAK` : Réponse du serveur – annonce au client de la fin de son bail ou d'une mauvaise configuration réseau
- `DHCPDECLINE` : Annonce par le client au serveur que l'adresse est déjà utilisée

Protocole DHCP (2)

Requêtes :

- `DHCPRELEASE` : Libération de l'adresse IP par le client
- `DHCPINFORM` : Demande des paramètres locaux par le client – il possède déjà son adresse IP

Exemple de messages dans /var/log/syslog

```
Jun 16 15:03:04 dns-dhcp dhcpd: DHCPREQUEST for 192.168.60.73 from 00:e0:81:27:29:db via eth0
Jun 16 15:03:04 dns-dhcp dhcpd: DHCPACK on 192.168.60.73 to 00:e0:81:27:29:db via eth0
Jun 16 15:03:15 dns-dhcp dhcpd: DHCPREQUEST for 192.168.60.35 from 00:04:75:d5:7e:a3 via eth0
Jun 16 15:03:15 dns-dhcp dhcpd: DHCPACK on 192.168.60.35 to 00:04:75:d5:7e:a3 via eth0
Jun 16 15:03:16 dns-dhcp dhcpd: DHCPREQUEST for 192.168.60.69 from 00:e0:81:27:1b:52 via eth0
Jun 16 15:03:16 dns-dhcp dhcpd: DHCPACK on 192.168.60.69 to 00:e0:81:27:1b:52 via eth0
Jun 16 15:03:28 dns-dhcp dhcpd: DHCPREQUEST for 192.168.60.72 from 00:e0:81:27:29:dc via eth0
```

Serveur DHCP

- Adresse IP Fixe
- Package dhcp3-server
- Lancement du démon dhcpd dans le script de démarrage `/etc/init.d/dhcp3-server`
- Fichier de configuration : `/etc/dhcp3/dhcpd.conf`
- Déclaration des interfaces d'écoute : `/etc/default/dhcp3-server`
- De nombreuses options, groupes...

Client DHCP

- Package dhcp3-client
- Fichier de configuration `/etc/dhcp3/dhclient.conf`
- Configuration de la machine en tant que client DHCP via le fichier de configuration des interfaces réseau `/etc/network/interfaces`

Serveur de nom (DNS) (1)

Domain Name Serveur

- Gestion de machines nommées dans un espace de nom (domaine, zone)
- FQDN – Fully Qualified Domain Name
- Résolution des adresses IP
- Implémenté sur quasiment toutes les plates-formes

Serveur de nom (DNS) (2)

- Structure hiérarchique permettant une grande souplesse d'administration
- Zone : ensemble de machines
  - clients de cette zone
  - serveurs de cette zone
  - clients d'une autre zone
-

Service DNS (1)

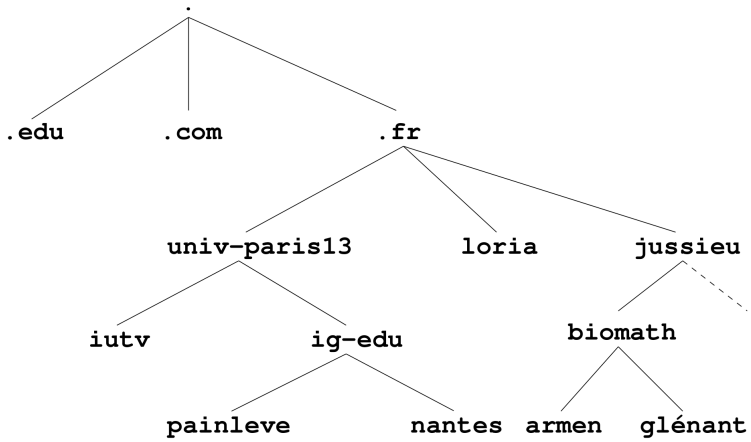
Assurer la correspondance entre les adresses IP et le nom des machines d'une zone

- Le serveur DNS
  - possède tous les renseignements sur la zone
  - peut faire autorité (serveur primaire)
  - délègue l'autorité sur les zones de niveau inférieur

Service DNS (2)

- Organisation hiérarchique suivant plusieurs critères :
  - Par secteur d'activité ou type de contenu : nom de domaine sur 3 lettres ou plus
  - Par pays : 2 lettres
- Domaines particuliers :
  - « . » : la racine de l'arbre
  - Top-level domain : domaine juste sous la racine, géré par les prestataires de connexion, les hébergeurs, des entreprises spécialisées, etc.

Hiérarchie


.
.edu
.com
.fr
univ-paris13
loria
jussieu
iutv
ig-edu
biomath
painleve
nantes
armen
glénant

Types de serveurs DNS

- Serveur primaire :
  Contient la liste des correspondances entre les adresses IP et les noms de machines
  Les mises à jour doivent être effectuées sur le primaire
- Serveur secondaire :
  Contient une copie de la base du serveur primaire
  Mise à jour automatique assurée par le serveur primaire
  Évite la surcharge du serveur primaire
- Serveur cache :
  Stocke en mémoire principale une partie de la liste
  Les serveurs primaires et secondaires sont des serveurs caches
- Autres serveurs : *forwarding server*, *stealth server*

Informations

Informations contenues et délivrées par le(s) serveurs :

- adresses en notation décimal (`A`)
- alias (`CNAME`)
- informations (`HINFO`)
- nom du DNS pour le domaine indiqué (`NS`)
- centre de tri de la messagerie (`MX`)
- etc.

Stockage

Types de stockages des informations délivrées par les serveurs DNS :

- Fichiers locaux
- NIS
- LDAP
- Bases de données (MySQL, PgSQL, Oracle, ...)

Configuration d'un client

- Serveurs à interroger et domaine d'appartenance
  `/etc/resolv.conf`
- Résolutions statiques
  `/etc/hosts`
- Ordre d'interrogations
  `/etc/host.conf`

Interrogation

Quelques outils d'interrogation :

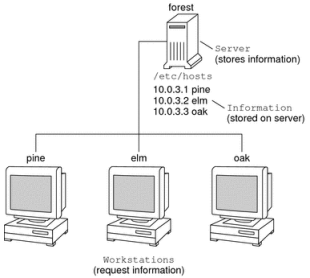
- `nslookup`
- `host`
- `dig`

Exemple :

```
xm@xm-laptop $ nslookup www.lipn.fr 8.8.8.8
Server: 8.8.8.8
Address: 8.8.8.8#53

Non-authoritative answer:
Name: www.lipn.fr
Address: 194.254.163.36
```

Problématique : Centraliser les informations



- Identification
- Authentification
- Localisation des données

NIS : Network Information Service

Distribuer sur un réseau les informations contenues dans des fichiers de configuration (`/etc/hosts`, `/etc/passwd`, ...)

- Service d'administration centralisé des principales bases de données système (`passwd`, `group`, `hosts`, `networks`, ...)
- Appelé à l'origine *Yellow Page* (YP)
- Implémentation par SUN dans les années 80
- Portabilité : PC, et sur la plupart des systèmes UNIX

Fonctionnement général

- Déploiement sur un réseau local
- Autour d'un machine centrale : `NIS Master Server`
- Clients :
  - Référence aux informations présentes sur un `domaine NIS`
  - Initialisation en broadcast
- Limitations :
  - Absence de hiérarchie
  - Impossibilité de communiquer entre deux domaines NIS
- Nom : quelconque, pouvant être différent (souvent) du nom du domaine
  Exemple
  Nom de domaine : `adsys.univ-paris13.fr`
  Nom du domaine NIS : `enseignement.galilee`

Machines sur un réseau NIS

3 types de machines sur un domaine NIS :

- Serveur maître NIS (`NIS Master Server`)
  Gestion des informations disponibles sur le domaine
- Serveur esclave NIS (`NIS Slave Server`)
  Possession d'une copie de la base du serveur maître
  Remplacement en cas de défaillance du serveur maître NIS
  Répartition de la charge sur le réseau
- Client NIS
  Exploitation des services de nom (`/etc/nsswitch.conf`)
  Dialogue avec les serveurs (maître ou esclave)

Informations gérées par NIS

- Utilisateurs (`passwd`)
- Groupes (`group`)
- Résolution de nom (`hosts`)
- `aliases`, `services`, `rpc`, `protocols`, `netgroup`
- tables d'automontage
- ...

Sécurité sous NIS

- Connaissance du nom du serveur : suffisant pour être client NIS
- L'interrogation d'un serveur NIS ne suppose aucune action de la part du serveur
- Faille importante dans la sécurité des mots de passe

$\rightarrow$NIS+

NIS+

Network Information Service +

- Successeur de NIS
- Résolution de certains problèmes liés à NIS :
  - Sécurité (cache, niveaux d'accès)
  - Structure hiérarchique permettant une administration répartie
  - Administrateurs explicitement nommés (différenciés de `root`)
  - Modification d'une entrée : uniquement diffusion de cette entrée
  - Chaque domaine peut posséder plusieurs sous-domaines

Problématique des administrateurs systèmes

Système d'information est composé :

- des utilisateurs (nom, prénom, mot de passe, email, droits d'accès)
- des groupes d'utilisateurs (liste précise d'utilisateur pré existants)
- des serveurs (description, adresse IP, service rendu)
- des postes de travail (description adresse IP, listes de logiciel, licences)
- un système de téléphonie (numéro de téléphone, répondeur)
- des services numériques (Messagerie, pages web...)

Problématique du responsable informatique

Comment avoir la bonne information au bon endroit et au bon moment ?

→ Problème complexe

Exemple : création d'un compte informatique

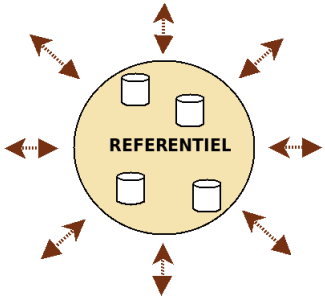
Le nouvel arrivant a besoin d'un compte informatique, mais très souvent l'informatique est souvent le dernier service prévenu.

- Comment créer un compte informatique dans l'urgence ?
- Où sont toutes les donnée nécessaires à la création ?
- Comment être sur que tous les systèmes informatiques ont les bons accès pour la personne ?
- Comment maintenir l'information cohérente chez tous les éléments d'un système d'information ?
- Comment savoir quel poste informatique va lui être affecté ?
- Quels vont être ses besoins en logiciels ou ressources réseaux ?

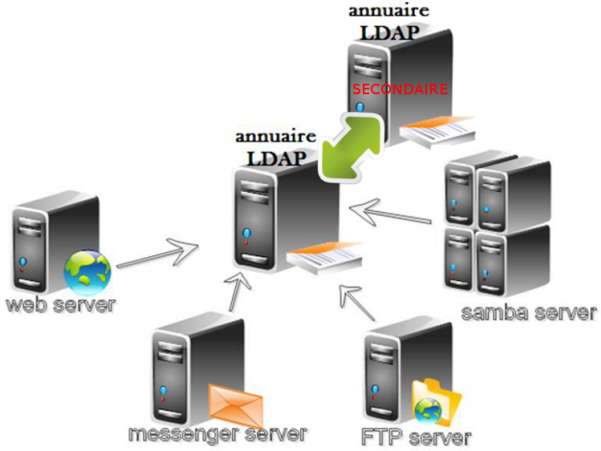
Solution : référentiel unique

Travail de l'administrateur est inversé :

- L'administrateur système ne remplit plus l'information sur X systèmes au risque de faire des erreurs.
- L'administrateur rempli une fois l'information sur ce référentiel, et ce sont les X systèmes qui viennent consulter et récupérer les informations dont ils ont besoin.



Centralisation des informations et tolérance aux pannes



Concepts de LDAP (1)

LDAP : Lightweight Directory Access Protocol

- Protocole d'annuaire sur TCP/IP
- Caractéristiques :
  - Stockage d'une grande quantité de données mais de faible volume
  - Accès en lecture très rapide grâce au modèle hiérarchique
- Annuaire standard et extensible fournissant :
  - protocole (accès à l'information dans l'annuaire)
  - modèle d'information (définition du type de données contenues dans l'annuaire)
  - modèle de nommage (définition de l'organisation et du référencement de l'information)

Concepts de LDAP (2)

LDAP : Lightweight Directory Access Protocol

- Annuaire standard et extensible fournissant (suite) :
  - modèle fonctionnel (définition de l'accès à l'information)
  - modèle de sécurité (définition de la protection des données et des accès)
  - mécanisme d'authentification pour le client
  - modèle de duplication (définition de la répartition de la base entre les serveurs)
  - APIs (développement d'applications clientes)
  - LDIF (format d'échange des données)

Logiciels LDAP (Serveurs)

- Slapd (Openldap, 389)
- Apache Directory Server
- Netscape Directory Server
- Directory Services (Sun Microsystems)
- DSSeries LDAP Directory (IBM)
- Serveurs acceptant des requêtes au format LDAP :
  - NetWare Directory Services (NDS) 3.0 (Novell)
  - Active Directory (AD) (Microsoft)
  - Lotus Domino

Logiciels LDAP (Clients)

- Navigateur Web utilisant URLs LDAP
- Microsoft Outlook
- Mozilla Thunderbird
- Clients de gestion (GUI)
  - Apache Directory Studio
  - phpLdapAdmin
  - ldapbrowser

Système de fichiers en réseau NFS (1)

- Service d'accès transparent à des données distantes
- Fonctionnement au-dessus des couches UDP/IP (et TCP/IP pour la version 3) : couche TI-RPC, indépendante de la couche transport
- Service réseau indépendant du matériel
- NFS version 2 : RFC 1094 $\rightarrow$ version 4
- Adopté par tous les constructeurs car SUN a rendu ses sources publiques

Système de fichiers en réseau NFS (2)

- Implémentation de type client/serveur :
  - Client : accès aux données distantes
  - Serveur : exportation des données se trouvant physiquement sur la machine

Protocole NFS

Protocole sans état (pour le serveur) : toutes les opérations sont mémorisées par le client

- Chaque requête NFS doit être accompagnée de l'ensemble des données nécessaires à son exécution
- Pas de mémorisation des opérations successives effectuées par le serveur
- Mémorisation de l'opération d'ouverture du fichier par le client et non le serveur

Utilisation de NFS (1)

- Administration des fichiers `/etc/passwd` et `/etc/group` de l'ensemble des machines par une même autorité (contrôle d'accès par UID et GID et non par nom de login)
  $\rightarrow$ Utilisation d'annuaire type NIS ou LDAP recommandée
- Droits des fichiers exportés par NFS :
  - Droits des fichiers de l'utilisateur déclaré sur le serveur
  - Une exception : le super-utilisateur (uid = 0)
  - Sans option d'exportation spécifique, le `root` est considéré comme `anonymous`
    Il ne possède quasiment aucun droit sur la machine cliente

Utilisation de NFS (2)

- Opérations (client NFS) :
  - Accès aux données distantes : montage des répertoires partagés par le serveur NFS
  - Envoi des requêtes au server NFS pour la lecture/écriture des fichiers
  - Mise à jour des informations concernant les fichiers ouverts
  - Mémorise les requêtes d'écriture sur les fichiers
  - Utilisation d'un cache pour limiter le trafic réseau lors d'accès en lecture seule à des fichiers montés par NFS
  - Mise en place du gestionnaire de fichiers du client dans la table de montage de son noyau
  - Utilisation du gestionnaire de fichiers lors de l'envoi de requêtes par le client

Possibilité de monter les systèmes de fichiers de plusieurs serveurs NFS

Utilisation de NFS (3)

- Opérations (serveur NFS) :
  - Partage des répertoires : les rendre accessibles de manière transparente pour les applications lancées sur les clients
  - Réponse aux requêtes NFS : lecture ou écriture des attributs et du contenu des fichiers
  - Pas de conservation des informations relatives aux fichiers ouverts
  - Possibilité de servir les clients d'autres serveurs NFS

Auto-montage (1)

- Objectif : remédier au problème de l'effondrement d'un client (nécessité de remonter le système de fichiers à la main)
- Proposition d'un système de montage dynamique
- Temporisation du montage : démontage du système de fichiers s'il n'est pas accédé pendant 5 minutes (valeur par défaut)
- Système de fichier auto-monté : *autofs*

Auto-montage (2)

- L'auto-montage ne concerne que les clients
- L'auto-montage n'apporte rien de plus par rapport au montage permanent en cas de problème (blocage de l'application effectuant les lectures/écritures)
- Utilisation des auto-montages : comptes (`home`) utilisateurs par exemple

Samba (1)

- Service de partage de ressources sous UNIX pour les machines sous Windows
- Début en 1991 (Andrew Tridgell, Australie)
- Machines Unix visibles dans le voisinage réseau
- Architecture client/serveur

Samba (2)

Serveur Samba : serveur de fichiers et de services *classiques*

- Partage de fichiers et de répertoires
- Partage d'imprimantes
- Prise en compte des comptes utilisateurs
- Gestion des permissions d'accès
- Exécution de scripts de connexion

Samba (3)

- Nécessite les protocoles NetBios, et TCP/IP (pour les clients)
- Protocole SMB : basé sur NetBios
- SMB : Server Message Block
- On parle aussi de CIFS (Common Internet File System)

Telnet : TELetype NETwork

- Protocole réseau utilisé sur tout réseau prenant en charge le protocole TCP/IP
  - Telnet : port TCP 23
- Commande permettant de créer une session et d'accéder à un terminal distant pour taper des commandes
- Identifiant et mot de passe nécessaire pour se connecter, mais...
- ... protocole non sécurisé (pas de chiffrement des communications)

Plus généralement aujourd'hui :

- Application permettant à l'administrateur de jouer le rôle de client sur une machine pour un service donné (débogage d'un service)

SSH : Secure SHell

- Protocole réseau utilisé sur tout réseau prenant en charge le protocole TCP/IP
  - SSH : port TCP 22
- Commande permettant de créer une session et d'accéder à un terminal distant pour taper des commandes
- Protocole de communication sécurisé entre un client et un serveur

Deux types d'authentification

- Par mot de passe :
  - + aucune configuration
  - - chaque connexion nécessite de retaper le mot de passe
  - - certaines fonctions peuvent nécessiter une authentification par clé (car pas de prompt)
  - - si quelqu'un connaît votre mot de passe, il a accès à votre machine
- Par clé :
  - + demande d'une phrase de passe une seule fois par session
  - + double sécurité
  - - nécessite de la configuration

Principe des clés

1. Quelqu'un prétend que c'est à lui qu'appartient une clé publique
2. Pour le prouver, il signe son message avec sa clé privée
3. Son interlocuteur peut alors vérifier avec la clé publique que le message a bien été signé avec la clé privée associée

Principe des clés : Sécurité

On ne peut pas à partir de la clé publique trouver la clé privée associée...

... dans un temps raisonnable

Principe des clés : Besoin de deux clés

- Chaque client génère une paire de clé :
  - une clé publique, que tout le monde peut connaître
  - une clé privée, qui doit être protégée et ne doit jamais être envoyée sur le réseau
- Ces deux clés sont mathématiquement liées

Mécanisme des clés pour SSH

- Un serveur SSH possède une liste de clés publiques autorisées à se connecter
- Le client possède sa clé privée, chiffrée par une passphrase
- Lors de la demande de connexion, le client veut utiliser sa clé privée
  - Si c'est la première utilisation de la clé lors de la session, l'utilisateur doit la déchiffrer, en tapant sa passphrase
  - Une fois la clé déchiffrée, la passphrase ne sera plus demandée
- L'authentification est réciproque : les clients possèdent un fichier contenant les serveurs qu'il connaît

Transferts de fichiers via SSH

Il est possible de faire des transferts de fichiers au travers d'SSH

- En ligne de commande sous Unix/Linux : scp, sftp
- Quelques outils graphiques :
  - FileZilla le célèbre client FTP, multi-plates-formes
  - Winscp, sous Windows
  - gFTP sous Linux/Gnome

Précautions

On a accès à tous les fichiers de la machine, avec les droits de l'utilisateur connecté :

- OK : pour récupérer nos propres fichiers de n'importe quel endroit
- NOK : pour donner l'accès à quelqu'un d'autre

FTP : File Transfert Protocol

- Destiné uniquement aux transferts de fichiers
- Utilisation de 2 ports de communication :
  - 20 : Port de données
  - 21 : Port de contrôle
- Nombre limité de commandes sur une machine distante
- Différentes implémentations : `wuftpd, proftpd`
- Existence de nombreux problèmes de sécurité (entre autres, la connexion n'est pas chiffrée...) : il vaut mieux éviter (ou restreindre les connexions par FTP sur les machines

Services de synchronisation horaire

Notion de temps et d'heure importante pour les systèmes informatiques :

- Horodatage de fichiers
- Corrélation de messages de "logs" de plusieurs systèmes
- Oblitération des courriers
- Gestion des caches DNS

Problèmes :

- Remise à l'heure régulière des oscillateurs à quartz (dérive de temps)
- Sur un réseau, heure différente entre diverses équipements pouvant accéder aux mêmes fichiers
- Réception de fichiers avant les avoir émis !
- NIS+ : utilisation de l'horaire pour l'authentification

Protocoles de synchronisation horaire

1. Time Protocol
2. Network Time Protocol (NTP)
3. Simple Network Time Protocol (SNTP)
   Version simplifiée de NTP

Time Protocol

- Système simple d'interrogation pour obtenir le temps d'un serveur (1983 !)
- Utilisation du port 37 (UDP ou TCP)
- Envoi par les serveurs d'un paquet contenant le temps en secondes écoulé depuis le 1er janvier 1900 à 0h00
- Utilisation par le démon Unix `timed`

Mais :

- faible précision
- absence de mécanisme de compensation du délais de transit

Network Time Protocol (NTP) (1)

- Synchronisation de l'heure avec un serveur en ligne
- Protocole beaucoup plus sophistiqué
- Synchronisation permanente avec plusieurs serveurs
- Redondances multiples pour assurer une synchronisation permanente et fiable
- Implémentation client/serveur

Network Time Protocol (NTP) (2)

- Utilisation de serveurs géographiquement proches
- Correction des délais de
  - transmission
  - dérive des horloges locales
- Représentation du temps NTP sur un entier de 64 bits
- Débordement prévu en 2036

Fonctionnalités

Définition dans NTP :

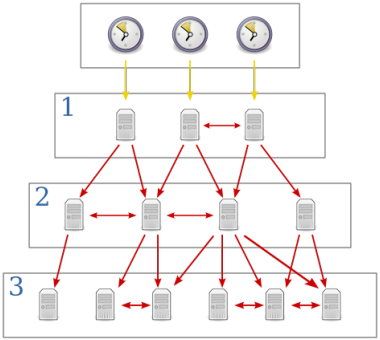
- Algorithmes de filtrage et de sélection
- Modèles d'implémentation

Permet :

- Détermination par les clients NTP de la meilleure source de synchronisation
- Élimination des sources suspectes
- Correction les temps de transit dans le réseau

Organisation des serveurs de temps

- Structure pyramidale
- Synchronisation des serveurs NTP grâce aux références de temps directement raccordées



Fonctionnement

- Mode serveur simple : Réponse aux requêtes de ses clients
- Mode broadcast : destiné aux réseaux locaux
- Mode client : envoie des requêtes à un ou plusieurs serveurs

SNTP : Simple Network Time Protocol

- Version simplifiée de NTP
- Pas de mécanismes de sélection
- Destinée à des utilisations pour lesquelles user une précision de l'ordre de la seconde est suffisante
- Synchronisation possible d'un client SNTP sur un serveur NTP

Configuration de NTP (1)

- Synchronisation immédiate d'un client : `ntpdate`
- idem avec indication du(des) serveur(s) sur lesquels synchroniser : `ntpdate -b ntp.tuxfamily.net (ntp.univ-lyon1.fr)`
- Fichier de configuration : `/etc/ntp.conf`
- Programme-démon-commande : `ntpq`
- La commande `ntpq` permet également de connaître l'état de synchronisation Par exemple :

```
# ntpq -npw
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 79.143.250.152  .GPS.            1 u   87h 1024    0  1434.58 -712.25    0.000
+158.227.98.15   .GPS.            1 u   667 1024  377    47.584    6.158   5.511
*130.149.17.21   .PPS.            1 u   706 1024  377    36.078    2.458   3.073
+164.132.235.192 131.188.3.221    2 u   459 1024  377     7.401    2.824   3.285
-129.175.34.43   193.79.237.14    2 u   280 1024  377     4.194   -0.419   4.985
-193.55.120.12   145.238.203.14   2 u   710 1024  377     7.806    2.719   3.302
```

Stockage des données

- Disques (bus IDE/SATA/SCSI/FibreChanel)
- Systèmes de sauvegarde au niveau matériel : RAID
- Centralisation de stockage et archivage lié au réseau (SAN/NAS)

Disques

Différents bus :

- IDE/ATA : 1 à 1064 Mb/s (2 périphériques)
- SATA (ATA série) : 1 à 6 Gb/s (1 périphérique)
- SCSI : 40 à 4 Gb/s (7/15 à 128 périphériques)
- SCSI-SAS : jusqu'à 12 Gb/s
- FibreChanel (série) : 1 à 128Gb/s
- FireWire : de 100 Mb/s à 3,2 Gb/s (63 périphériques par bus – 1024 bus)
- USB : (1) 1,5/12 Mb/s, (2) 480 Mb/s, (3) 5 Gb/s

Rappel : (Réseau ethernet) 10 Mb/s, 100 Mb/s, 1 Gb/s, 10 Gb/s, 40 Gb/s, 100 Gb/s

Systèmes RAIDs

Redundant Array of Independent/Inexpensive Discs

Niveaux de RAID définis par l'Université de Berkeley :

- RAID 0 : Stripping (entrelacement de disques)
- RAID 1 : Mirroring (miroir de disque)
- RAID 1+0 (RAID 10) : Stripping et Mirroring
- RAID 4 : Stripping sur plusieurs disques avec parité sur disque dédié
- RAID 5 : Stripping sur disques indépendants avec parité répartie

Niveaux logiciel ou matériel

RAID 0 (1)

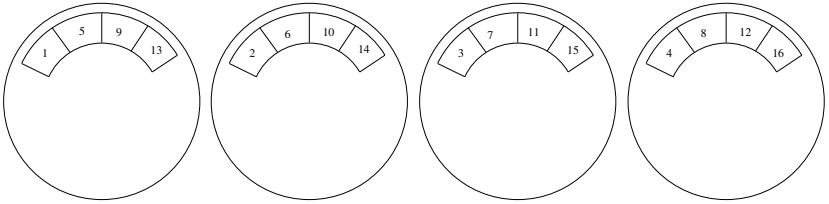
Stripping

- Entrelacement des données
- Assemblage de plusieurs petites unités de disques pour former une grande unité logique
- Facteur d'entrelacement : taille du fragment stocké sur le disque physique (en général 1 octet)

RAID 0 (2)

Problème : la défaillance d'un disque entraîne l'impossibilité d'accès aux données

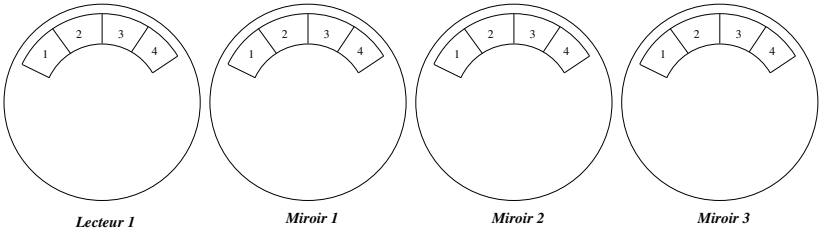
→ La sécurité n'est pas améliorée



RAID 1

Mirroring

- Duplication des données d'un disque sur tous les autres disques du groupe RAID
- Accessibilité des données même en cas de défaillance



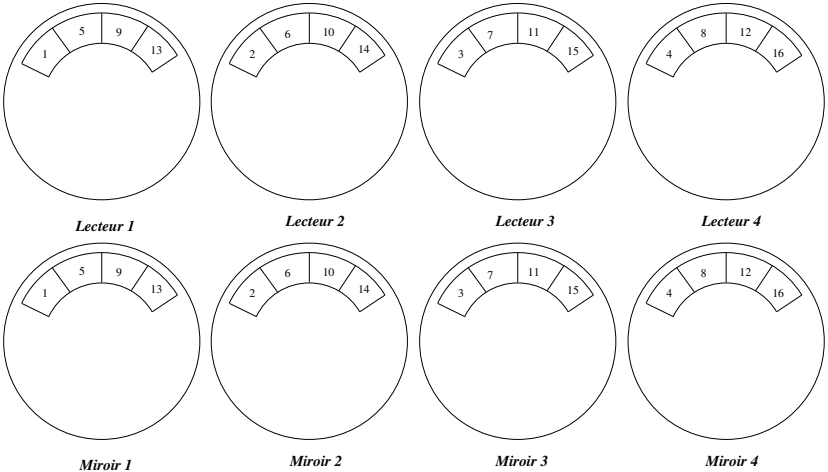
$\rightarrow$ Technique onéreuse

RAID 0+1 (RAID 10) (1)

Stripping et Mirroring

- Haute sécurité
- Performance de RAID 0
- Coût élevé

RAID 0+1 (RAID 10) (2)

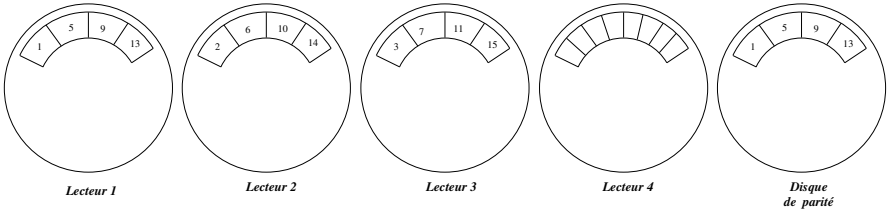


RAID 4 (1)

Stripping avec contrôle de parité sur disque dédié

- RAID 1 et RAID 0+1 : sécurité contre les pannes implique le doublement des disques
- RAID 4 : utilisation du concept de parité
- Pour chaque groupe de x disques entrelacés, ajout d'un disque
- Disque supplémentaire : n-ième bit (bit de parité) formé à partir des n-ième bit des x disques
- Crash d'un disque : restauration du contenu bit pour bit à partir de $x - 1$ autres disques et du disque de parité
- Écriture de données : actualisation du disque de parité
- Rapidité inférieure à RAID 0 et RAID 1

RAID 4 (2)

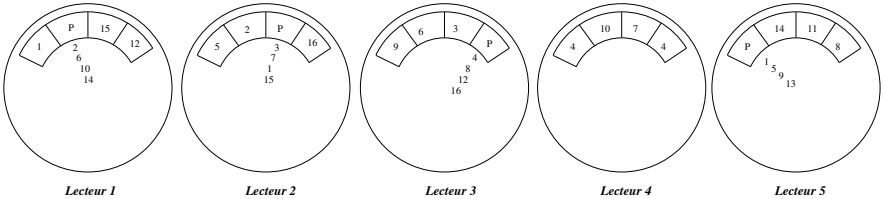

1
5
9
13
2
6
10
14
3
7
11
15
1
5
9
13
Lecteur 1
Lecteur 2
Lecteur 3
Lecteur 4
Disque
de parité

RAID 5 (1)

Stripping sur disques indépendants avec parité répartie

- RAID 4 : Chaque accès disque implique un accès au disque de parité
- RAID 5 : Toutes les données et les indicateurs de parité sont répartis par bloc sur l'ensemble des disques
- Chaque disque : disque de parité pour une zone de blocs
- Amélioration des accès en lecture/écriture
- Pour de petites requêtes, RAID 5 est moins rapide qu'un disque dur unique

RAID 5 (2)


P
15
1
12
2
6
10
14
Lecteur 1
P
16
5
2
3
7
1
15
Lecteur 2
6
3
9
P
4
8
12
16
Lecteur 3
10
7
4
4
Lecteur 4
14
11
P
8
1
5
9
13
Lecteur 5

RAID logiciel

- Possibilité en standard sous de nombreux systèmes d'exploitation : Windows, Solaris, Linux
- Solution peu onéreuse et simple
- Mais réduction des performances du système d'exploitation
- Données relatives à la configuration du RAID sur le disque de démarrage : perte du système RAID en cas de défaillance de ce disque

RAID matériel

- Utilisation de contrôleur RAID PCI (RAID interne)
- Également RAID externe (solution très coûteuse)
- Disques SCSI ou Fibre Channel
- Mais aussi de plus en plus de disque IDE (réduction du coût de stockage)

Réseaux de stockage (1)

- Remplacement des stockages centralisés (type Mainframe) par une répartition sur de serveurs locaux et stations de travail
- Désormais retour à une centralisation des données :
  - bus rapide
  - mise en réseau de volume de données élevés
- Exigence de centralisation des données : une disponibilité élevée
  1. immobilisation d'une entreprise en cas de défaillance
  2. $\Longrightarrow$ chemin de données et stockage redondants
     Mais aussi copie de données (instantanés) et réplique synchrone/asynchrone

Réseaux de stockage (2)

Trois solutions :

- DAS (classique)
- NAS
- SAN (ou NAS + SAN)

DAS (1)

Direct Attachment Storage

- Plates-formes de stockage (*classique*) : chaque disque, ou ensemble de disques, est physiquement connecté à un serveur de fichiers
- Mémoire disque directement rattachée aux serveurs par du SCSI ou du Fibre Channel
- Manipulation de volumes de données pour un utilisateur de l'ordre de 10 à 20 Go
- Avantages :
  - peu coûteux
  - faciles à déployer et pas de travail d'intégration

DAS (2)

Inconvénients :

- Peu de possibilités d'évolution
- Peu performants pour la gestion des redondances

Solutions : RAID SCSI ou Fibre Channel

NAS (1)

Network Attached Storage

- Serveur de fichiers optimisé
- Réponse à l'augmentation du nombre de disques
- Repose sur les protocoles IP (de type Ethernet) Windows (CIFS) et Unix (NFS)
- Mutualisation des données stockées sur plusieurs systèmes et serveurs de fichiers d'un réseau Ethernet (LAN)
- Adapté à l'archivage et la réplication

NAS (2)

- Limites : croissance des volumes de données archivées : augmentation des charges au niveau IP et chute des systèmes

$$\Longrightarrow \text{Combinaison possible de NAS et DAS}$$

SAN (1)

Storage Area Network

- « Un réseau derrière le serveur »
- Réseau de stockage secondaire à vitesse élevée
- Existe parallèlement à un LAN pour le transfert de données entre clients et serveur
- Communication entre les systèmes centraux de stockage et un serveur

SAN (2)

- Utilisation simultanée d'un système de stockage (RAID) par plusieurs serveurs
  - partage de la capacité de stockage par les serveurs
  - réduction des coûts d'achat et d'administration
- Mise en œuvre de technologie réseau adéquate : Network Attached Storage (NAS)
  $\Longrightarrow$ Mise à disposition des données pour des serveurs ou des clients via le réseau

SAN (3)

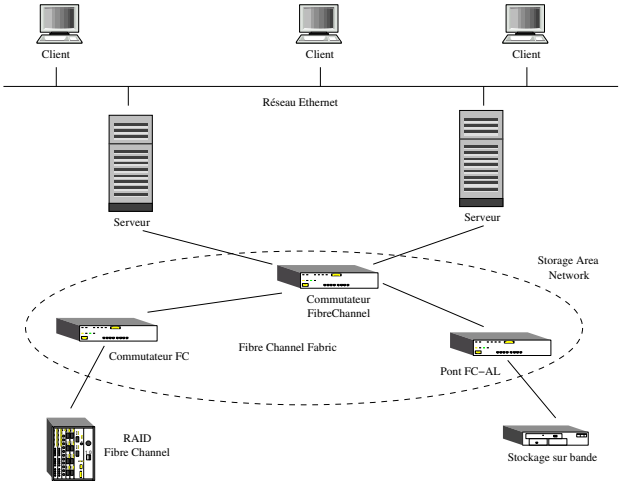
- Assure des performances élevées pour des quantités importantes de données :
- fourniture de données au niveau des blocs (comme un disque local)
- Transport de données par Ethernet Gigabit
- La carte Ethernet apparaît comme un périphérique SCSI (grâce à un pilote spécifique)
- Généralement utilisation de Fibre Channel :
  - Jeu de commande compatible avec le SCSI
  - Protocole (FC-AL - Arbitrated Loop) sans rapport avec le SCSI

Architecture SAN (1)

5 éléments de base :

- Serveurs
- Infrastructure (Fibre Channel Fabric)
- Stockage sur disques (RAID Fibre Channel, iSCSI/SCSI)
- Stockage sur bande
- Logiciel de gestion

Architecture SAN (2)



Composition d'un SAN

- commutateur ou ensemble de commutateur
- utilisation de Fibre Channel ou du SCSI

Liaison série point à point :

- sur un brin : trafic d'un CPU vers un espace disque
- commutateur : impression de multiplexage
- commutateur indispensable pour créer le réseau

Importante largeur de bande et performance E/S très élevée (jusqu'à 16 Go/s)
Nécessite un faible taux d'erreur (pas de correction)
Les données arrivent dans l'ordre de départ

Avantages du SAN

- Mise en commun des ressources de stockage en réseau
- Grande disponibilité des ressources informatiques
  Les éléments prennent le relais automatiquement sur ceux qui tombent en panne
- Unité de stockage RAID partagées
  Gestion conviviale et grande disponibilité
- Sauvegarde et restauration des données
  - sans impact sur le réseau local
  - sans l'utilisation de serveurs
    Archivage des données directement des disques durs sur des bandes magnétiques

Inconvénients du SAN

- peu de compatibilité
  - solution propriétaire
  - absence de norme
- peu de flexibilité et de facilité de gestion
- difficulté à avoir des SAN logiques

$\Longrightarrow$ en constante évolution
Utilisation du iSCSI (données SCSI transportées par Ethernet)

SAN et sécurité (1)

- Pas de fonction sécurité réelles/dédiées
- Uniquement contrôle d'accès (généralement pas utilisées)
- Possibilité de définition de zone (*Zoning*) :
  - réalisée par les commutateurs
  - Similaire aux VLANs
  - Création de zones/groupes de partitions et de serveurs
  - Identification par numéro de port ou *World Wild Name* (*WWN* – similaire à l'adressage Ethernet)

SAN et sécurité (2)

Commutateur SAN $\Longrightarrow$ relié au réseau Ethernet par IP :

- Problème avec SNMP
- Serveur web pour la gestion du commutateur
- Accès interactif par `telnet` ou `ssh`

Sauvegarde et restauration des données

- Protection des données
- Archivage des données

- Optimisation des sauvegardes
- Automatisation des sauvegardes

Sauvegarde : Reprise rapide du fonctionnement du système en cas de crash majeur avec une perte de données réduite
Exigences différentes s'il s'agit de l'administration système ou des données des utilisateurs
Sauvegarde d'un système : redescendu sur disque après un crash disque

Sauvegarde et archivage

- Sauvegarde :
  - Gestion d'index
  - Protection et restauration des données utilisées couramment
- Archivage :
  - Pas d'index
  - Déplacement de données inutilisées sur des supports moins coûteux

Compression

- Utilisation de procédures spéciales de compression de données :
  - Exploitation maximale des périphériques de sauvegarde
  - Optimisation des temps de sauvegarde et d'archivage
- Compression matérielle : plus efficace que les compressions logicielles

Procédés de sauvegarde (1)

Définition de niveaux de sauvegarde :

- Sauvegarde totale (Full dump) :
  Sauvegarde de l'intégralité des données
- Sauvegarde différentielle :
  Sauvegarde des données ayant été modifiées depuis la dernière sauvegarde de niveau strictement inférieur
- Sauvegarde incrémentale :
  Sauvegarde des données ayant été modifiées depuis la dernière sauvegarde

Procédés de sauvegarde (2)

Deux derniers types de sauvegarde :

- Réduction du temps de sauvegarde
- Mais la restauration est plus longue que pour la sauvegarde totale

Sauvegarde distante

- Utilisation de serveurs de sauvegarde : Souple et sûr
- Protection de différents systèmes présents sur le réseau
- Centralisation des informations de sauvegarde
- A l'aide d'applications spécifiques ou de la commande rsh

Média de sauvegarde (1)

- Cartouche 1/4 de pouce :
  - La plus répandue
  - Plusieurs formats d'écriture
  - Jusqu'à 2,5 To
- Cartouche 8mm
  - meilleur rapport qualité / prix
  - Jusqu'à 10Go (25Go compressé)
- Cartouche DAT 4mm
  - vitesse de recherche importante
  - Jusqu'à 40 Go compressé

Média de sauvegarde (2)

- CD-ROM / DVD
- Disque
- Autres : plus ou moins dépassé, mais pouvant être utilisé pour la sauvegarde longue durée (disque magnéto-optique) ou l'échange de fichiers (bande magnétique 1/2 pouce)

- Exemple de nom des médias de sauvegarde : `/dev/rmt5`, `/dev/rst0`
- Outils de manipulation de bande (bobinage, rembobinage, positionnement) : `mt`

Commandes de sauvegarde de données (1)

- `tar` :
  - format portable
  - mono-volume la plupart du temps
  - simple à utiliser et à manipuler
  - surtout employé au niveau utilisateur

```
cd /etc/
tar cvf /tmp/archive_tar_etc *

cd ..
tar cvf /tmp/archive2_tar_etc ./etc

cd /usr/restore
tar xvf /tmp/archive_tar_etc
```

Commandes de sauvegarde de données (2)

- `cpio` :
  - format portable
  - multi-volume
  - options complexes
  - possibilité d'utilisation de la commande `find`
  - conforme au standard XPG[23]

```
cd /etc/
ls | cpio -o > /tmp/archive_cpio_etc

find . -print | cpio -o > /tmp/archive2_cpio_etc

cd /usr/restore
cpio -ivd < /tmp/archive_cpio_etc
```

Commandes de sauvegarde de données (3)

- `pax` :
  - *portable archive interchange*
  - compatible avec `tar`
  - présent sur la plupart des UNIX (origine OSF)
  - conforme au standard XPG4 et à la norme POSIX
- `dump` / `restore`

dump / restore

- Sauvegarde de disques entiers et systèmes de fichiers de plusieurs Go
- Condition préalable : si le système de fichiers est modifié pendant la sauvegarde, le contenu de la sauvegarde n'est pas garanti
  Réalisation d'un `fsck` sur le système de fichiers afin de vérifier sa cohérence

Fonctionnalités de dump (1)

- Sauvegarde totale et différentielle (incrémentale)
  Utilisation de niveau de sauvegarde
- Sauvegarde optimisée en utilisant les périphériques en mode caractère
  Sauvegarde de la structure du système de fichiers
  Sauvegarde des blocs à l'aide du *raw device*

Fonctionnalités de dump (2)

- Sauvegarde multi-volume
  Détection des fins de bandes
  Sauvegarde sur plusieurs bandes
- Sauvegarde à travers le réseau
- Sauvegarde de partitions entières

Sauvegarde avec dump

2 étapes :

1. Structures d'inodes
   Sauvegarde des blocs référant les inodes
2. Blocs de données
   Accès aux périphériques en mode caractère
   Accélération de la sauvegarde

```
dump 0udsf 6250 2400 /dev/rmt8 /usr
```

Fréquence de sauvegardes

Différente suivant les systèmes de fichiers :

- Système de fichiers `root` :
  Après configuration du système
  Après chaque modification de la configuration
- Système de fichiers `usr` :
  Après installation du système et des applications
  Après ajout de nouvelles applications
- Système de fichiers contenant les données utilisateurs
  Sauvegarde quotidienne
  Archivage régulière (hebdomadaire)
  Nécessite une réflexion sur la stratégie de sauvegarde

Restauration d'un système de fichiers

Restauration totale :

1. Utilisation de `newsfs` pour refaire un système de fichiers vierge
2. Montage la partition dans l'arborescence
3. Utilisation de la commande `restore`
   Restauration des données depuis la dernière sauvegarde de niveau 0 et des sauvegardes incrémentales

```
cd /usr
restore r
restore xh /usr
```

Fonctionnalité de `restore`

- Restauration complète
- Restauration interactive
- Restauration de fichiers à la demande

Synchronisation d'arborescence

Arborescence locale ou distante

- unidirectionnelle `rsync`

  `/usr/bin/rsync OPTIONS SOURCE CIBLE`

  Exemple :

```
/usr/bin/rsync -azu  -r --delete -v -e ssh
        /export/home/monnin/
        monnin@194.254.167.19:/export/home/users/Enseignant/monnin
```

NB : il est nécessaire de disposer des deux outils sur les deux machines

Synchronisation d'arborescence

Arborescence locale ou distante

- bidirectionnelle `unison`

  `/usr/bin/unison OPTIONS REPERTOIRE1 REPERTOIRE2`
  Exemple :

```
/usr/bin/unison -batch
        /export/home/monnin
        ssh://monnin@194.254.167.19//export/home/users/Enseignant/monnin
```

NB : il est nécessaire de disposer des deux outils sur les deux machines

Sécurité informatique (1)

Thème large :

- Confidentialité : Accessibilité des données informatiques uniquement aux personnes habilitées (organismes gouvernementaux, bancaires)
- Intégrité : Modification des données uniquement par les personnes habilitées (pertes de données)
- Disponibilité : Garantie permanente de l'accès à l'outil informatique aux personnes habilités

Sécurité informatique (2)

- Existence de solution dans ces trois domaines.
- Mais à l'encontre des objectifs de base d'UNIX : un système pour programmeur et primauté de l'échange d'information
- Au départ, utilisation de protocoles de communication non chiffré
- Un système de plus en plus complexe au niveau sécurité (réseau)

Sécurité informatique (3)

```
# snoop | grep -v RLOGIN | grep omaha
Using device /dev/hme (promiscuous mode)
       machine1 -> machine2     TELNET C port=35966 t
    machine2 -> machine1        TELNET R port=35966 t
       machine1 -> machine2     TELNET C port=35966
       machine1 -> machine2     TELNET C port=35966 h
    machine2 -> machine1        TELNET R port=35966 h
       machine1 -> machine2     TELNET C port=35966
       machine1 -> machine2     TELNET C port=35966
    machine2 -> machine1        TELNET R port=35966
       machine1 -> machine2     TELNET C port=35966
    machine2 -> machine1        TELNET R port=35966 Password:
       machine1 -> machine2     TELNET C port=35966
       machine1 -> machine2     TELNET C port=35966 a
    machine2 -> machine1        TELNET R port=35966
       machine1 -> machine2     TELNET C port=35966 b
    machine2 -> machine1        TELNET R port=35966
```

Niveau utilisateur

- Sensibilisation à la sécurité par les administrateurs
- Essentiellement au niveau de la connexion en tant qu'utilisateur et de l'accessibilité des données
- Mettre en place des règles de conduite pour
  - Définir une charte informatique
  - Protéger les données
  - Protéger son compte (mot de passe sûr)
- Mais aussi de plus en plus (notamment sous Windows) : bien utiliser sa messagerie.

Protection des données

- Ne pas stocker n'importe où des données sensibles
  Utiliser les zones de stockage protégées (et sauvegardées régulièrement)
- Restreindre l'accès à des données sensibles
  Mise en place de zone de stockage protégées : sous réseau à accès restreint (par connexion et/ou origine)
- Ne pas diffuser son mot de passe

Mot de passe (1)

Point souvent négligé par les utilisateurs

- Changement des mots de passe assez régulièrement (fréquence dépendante du niveau de sécurité du réseau) :
  - Tous les mois
  - Toutes les semaines
  - Tous les jours
  - A chaque connexion (utilisation de carte génératrice de mot de passe)

Mot de passe (2)

$\rightarrow$ une faille de sécurité importante

Bien choisir son mot de passe :

- facile à mémoriser, mais difficile à retrouver
- utiliser une combinaison semblant être définie au hasard
  Nécessite un recours à la force brutale pour la décrypter
- utiliser tous les caractères possibles (chiffres et ponctuation) :
  augmentation de la combinatoire donc du délai de décryptage

Mot de passe (3)

Exemple de méthodes classiques :

- Prendre une phrase ou une ligne d'un poème ou d'une chanson qui doit être d'au moins long de 8 mots,
- Prendre la première lettre de chaque mot et l'utiliser dans le mot de passe.
- Utiliser les signes de ponctuation.

Par exemple :
C'est un trou de verdure où chante une rivière $\Longrightarrow$ `C1t2v|cur`

Mot de passe (4)

Interdire les mots de passe dérivés d'un dictionnaire ou d'informations personnelles (même modifiés)
Exemple :

- nom de login : xavier.monnin
- noms famille : monnin
- prénom : xavier
- mots écrits à l'envers : ninnom, reivax
- mots d'un dictionnaire : reseau, maison, utilisateur
- mots avec des capitales : ReZo, ReSeau, MaIsOn
- mots des dictionnaires de cracking : 123456789, azerty, abcxyz, starwars, darkvador
- mots de langues étrangères : hello, gutentag

Messagerie

Principal problème : les virus

- Rappeler certaines règles aux utilisateurs :
  - ne pas cliquer sur les liens présents dans les messages
  - ne pas ouvrir les messages d'expéditeur inconnu
- Éviter l'exécution de programmes à partir de la messagerie
- Filtrer les messages (ou indiquer comment le faire)

Niveau système d'exploitation (1)

Différentes failles :

- Authentification et utilisateurs
- Faille de sécurité dans les programmes (dépassement de pile, ports ouverts)
- Incertitude concernant la sécurité sur certaines machines (ordinateurs portables) et certaines parties du réseau (service Web et FTP, Wifi)

Niveau système d'exploitation (1)

Solutions :

- Établir de règles de sécurité (voir précédemment)
- Crypter les connexions (SSH)
- Connaître son réseau et ses faiblesses (audit)
- Restreindre les accès (services, tcpwrappeur, pare-feu)
- Limiter la diffusion vers l'extérieur d'informations concernant votre réseau
- Se tenir au courant des évolutions de sécurité : CERT (`cert-advisory@cert.org`), BugTraq (`bugtraq-request@crimlab.com`)

Authentification

- Gérer correctement les comptes utilisateurs
  $\Longrightarrow$ Autant de portes ouverts pour les pirates
- Vérifier régulièrement /etc/passwd ou la base des comptes utilisateurs
  - Supprimer les comptes qui ne servent pas
  - Bloquer les comptes non utilisés pendant un certain temps
  - Vérifier les mots de passe des utilisateurs

Vérification des mots de passe (1)

1. Apprendre au utilisateur à ne pas mettre n'importe quel mot de passe
2. Identifier les mots de passe faibles
3. Ajouter si possible des contraintes sur les mots de passe (longueur minimale, augmenter la taille, etc.)
4. Tester la difficulté d'un mot de passe
   Principe : si l'administrateur peut craquer un mot de passe, un pirate pourra également

Vérification des mots de passe (2)

Difficulté d'un mot de passe

Utiliser des outils destinés à craquer les mots de passe :

- John the Ripper (`http://www.openwall.com/john/`)

Utilisation d'un dictionnaire de "cracking" : suite de mots servant pour trouver des mots de passe "faciles"

Vérification des mots de passe (3)

Principe du test :

- mot de passe crypté
- cryptage de tous les mots de passe du dictionnaire (plus quelques heuristiques)
- Comparaison avec le mot de passe à décrypter
- si le mot de passe et un des mots du dictionnaire sont identiques, alors le mot de passe est découvert

Taille indicative d'un dictionnaire : 1.4 million de mots

Les "R" commandes

Connexion sans mot de passe :
Problèmes :

- fichier `/etc/hosts.equiv` (ne concerne pas le `root`), `~/.rhosts`
- Outils très dangereux
  authentification rudimentaire : très faciles à attaquer

Solutions :

- Supprimer le fichier `/etc/hosts.equiv`, vérifier la cohérence du contenu du fichier `~/.rhosts`
- Empêcher leur utilisation (préférer SSH)

Mécanismes d'authentification (1)

- Connexion avec mot de passe :
  Problème : fichier de stockage des mots de passe
  Solutions : *shadow passwords*, NIS+, LDAP
- Gestion des fichiers locaux :
  Problèmes : fichiers *suid*, fichier type périphérique
  Solution : contrôle lors du montage par ajout des options `nodev` et `nosuid` (sinon vérifier périodiquement ces fichiers)

Sécuriser les serveurs (1)

- Identifier les points faibles des serveurs
- Vérifier régulièrement les versions, les avis de sécurité et les corrections
  - Serveur FTP : FTP anonyme.
  - Serveur Web :
    Peu de faille de sécurité dans sa version classique (exécution avec un utilisateur ayant très peu de droit sur le système)
    Problèmes importants : CGI, PHP, etc...
    (exécution de commandes possibles sous couvert d'un utilisateur)

Sécuriser les serveurs (2)

- Serveurs NFS :
  Problèmes :
  1. Détermination des machines vers lesquels on exporte l'arborescence
  2. Manière d'exporter et d'importer l'arborescence

  Solutions :
  1. N'exporter que le strict minimum
  2. Définir le maximum de restriction au niveau des accès
  3. Définition dans le fichier `/etc/exports` de la liste des machines, soit explicitement, soit à l'aide de `netgroups`, NIS ou NIS+
  4. Définition d'options de montage (`nodev`,`nosuid`)

Mais pas ou très peu de contrôle possible sous NFS 3
$\rightarrow$ Passer à Kerberos + NFS 4

Sécuriser les serveurs (3)

Ne pas faire tourner des services inutiles sur une machine : Supprimer les services non nécessaires

- Examiner le fichier /etc/inetd.conf
- Désactiver les services non nécessaires
- Ne lancer au démarrage que les services nécessaires
  Examen des scripts de démarrage dans `/etc/rc5.d` ou `/etc/rc3.d`
  $\Longrightarrow$ Supprimer les liens symboliques `SXXxxxx` non nécessaires
  (ou utiliser `chkconfig --del xxxx`)

Sécuriser les serveurs (3)

Surveiller les services :

- effectuant des statistiques, notamment : finger, systat, netstat
  (peuvent fournir des informations à des personnes extérieures)
- permettant la configuration à l'aide d'un navigateur internet : `swat` (samba), `linuxconf`

Techniques de détection des failles et des intrusions

- Vérification de l'état des fichiers et programmes systèmes
- Audit du réseau
- Scan des ports des (UNIQUEMENT) machines du réseau

Analyses statistiques du système

- Session utilisateur :
  Problème : Détermination des connexions des utilisateurs
  Solution : systèmes de mouchard (`audit`, `wtmp`, `utmp`, `last`)
- Outils d'analyse statique d'un système : Vérifier l'état des fichiers systèmes
  - COPS (Computer Oracle Passwd and Security System) – pas d'analyse du réseau
  - ISS (Internet Security Scanner) – spécialisé dans le diagnostic des problèmes de sécurité du réseau
  - Ntop : Analyse du trafic sur le réseau
  - Crack ou John The Ripper – Analyse des mots de passe en fonction de leur degré de sécurité

Vérification de l'intégrité du système

Éviter les modifications de programmes sensibles (`passwd`, `ping`, `find`, `login`, etc.)

- Tripwire :
  - cliché du système sûr (après installation ou mise à jour)
  - comparaison régulière avec un cliché de référence

Analyse dynamique du fonctionnement du système

Avoir une trace des opérations des programmes systèmes

- Traitement/création des fichiers systèmes : Syslog, Swatch (System Watcher), Trimlog
- Restriction d'accès : TCP-Wrappers
  Encapsulation des démons
  Définition des machines autorisées ou non à se connecter sur une machine (`/etc/hosts.allow` et `/etc/hosts.deny` à l'aide d'un service donné
- TCPlogd (détection des attaques de type scan)
- Portsentry
- Fail2ban

Autres : `SATAN/Sara`, `Nmap`, `Nessus`, `ethereal`

Audit de la sécurité du réseau

- Objectif : tests pour détection de failles de sécurité sur le réseau
- Opération importante pour vérifier la sécurité du réseau
- Quelques outils classiques : `Nessus`, `Nmap`, `Tcplog`, `PortSentry`
- IMPORTANT : n'utiliser ces outils que pour vérifier la sécurité de votre réseau

$\Longrightarrow$ Tests sur des machines extérieurs : attaques potentielles

Audit avec Nmap (1)

- `Nmap` : scan des ports
- Identification des ports ouverts sur une machine (ports sur lesquels des serveurs écoutent)
- Exemple : acceptation des connexions `telnet`, `ftp` ou `www`
- Première action d'un pirate
- Considérer comme une tentative d'attaque
- Exécutables :
  - `nmap` : outil de scan
  - `nmapfe` : interface graphique

Audit avec Nmap (2)

- Utilisation : `nmap [-sT -sU -O -p port -v ] cible`
- Cible : nom ou adresse d'une machine, ou classe d'adresse IP ou de réseau
- Exemple :

```
# nmap -O 192.168.0.2

Starting nmap 3.55 ( http://www.insecure.org/nmap/ ) at 2005-05-12 22:33 CEST
Interesting ports on 192.168.0.2:
(The 1654 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
22/tcp   open  ssh
110/tcp  open  pop3
111/tcp  open  rpcbind
2401/tcp open  cvspserver
3306/tcp open  mysql
6000/tcp open  X11
```

Audit avec Nmap (3)

```
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.5.25 - 2.6.3 or Gentoo 1.2 Linux 2.4.19 rc1-rc7)
Uptime 0.163 days (since Thu May 12 18:38:00 2005)

Nmap run completed -- 1 IP address (1 host up) scanned in 5.438 seconds
```

Détection d'intrusion : Tcplog

- Détection passive : uniquement avertissement d'une tentative d'intrusion (en cours ou non)
- Deamon
- Signale les scans de port
  Conserve la trace dans un fichier de log
- Nécessite un consultation fréquente et régulière
  (console ouverte avec `tail -f /var/log/attaques`

(utilisation de nmap pour tester)

Détection active des intrusions : PortSentry (1)

- Détection active : réaction aux attaques
  Identification des intrusions et mise en place d'une défense
- Comportements possibles :
  Utilisation de
  - `tcpwrapper` : routage des paquets de l'attaquant vers `/dev/null`
  - `ipchains` : blocage des paquets
- Possibilité d'exécuter une commande lors d'une attaque (envoi d'un message à l'administrateur)

Fichier de configuration : `portsentry.conf`

Détection active des intrusions : PortSentry (2)

Utilisation des `iptables` :

```
# KILL_HOSTS_DENY='ALL : $TARGET$'
KILL_ROUTE='/sbin/iptables -P INPUT DROP
```

Fichier `portsentry.ignore` : liste des adresses IP à ne pas prendre en compte (*scaneur connu*)
Plusieurs modes d'exécution :

- comportement comme `tcplogd` : `Basic port-bound TCP mode` ou `Basic port-bound UDP mode`

Détection active des intrusions : PortSentry (3)

- Protection active (sur tcp et udp) : `Advanced UDP stealth (audp)`, `Advanced TCP stealth (atcp)`
  Exécution :

  ```
  portsentry -audp
  portsentry -atcp
  ```
- Information dans les logs en cas d'attaque

Système de connexion sécurisé SSH

- Alternative sécurisée à des outils de connexion (`telnet`, `rlogin`)
- Tunnel de connexion crypté entre deux machines
- Cryptage RSA
- Serveur : démon `sshd`
- Client : ensemble de commandes `slogin`, `sftp`, `ssh`

Type de connexion SSH (1)

1. Possibilité d'équivalence : `/etc/hosts.equiv`, `/etc/ssh/shosts.equiv`
   Méthode non sécurisée
2. Authentification par machine (RSA) :
   Connexion autorisée dans `$HOME/.rhosts`, `$HOME/.shosts`, `/etc/hosts.equiv`, `/etc/ssh/shosts.equiv`
   Vérification de la clé par le serveur (consultation des fichiers `/etc/ssh/ssh_known_hosts` ou `$HOME/.ssh/known_hosts`)
   Éviter l'usurpation d'adresse IP, falsification de DNS ou de routage

Type de connexion SSH (2)

3 Authentification basée sur RSA
Cryptographie par clé publique (déduction de la clé de décryptage à partir de la clé de cryptage impossible)
Système de clés publique/privée
Procédure (ssh1) :

- Création d'une paire clés publique/privée
  Connaissance de la clé publique par le serveur
  Liste des clés publiques : `$HOME/.ssh/authorized_keys`
  Connaissance de la clé privée par le client
- Connexion : Envoi au serveur, par l'utilisateur, de la paire clés publique/privée qu'il souhaite utilisée
- Vérification par le serveur si la clé est autorisée

Type de connexion SSH (3)

- Vérification que l'utilisateur a bien la clé privée
  Envoi d'un défi : décryptage d'un nombre aléatoire crypté à l'aide de la clé publique de l'utilisateur
  Décryptage du nombre uniquement possible à l'aide de la clé privée
  Si le décryptage effectué, l'utilisateur a bien la clé privée (sans l'envoyer au serveur)

4 Utilisation d'un mot de passe
  Transmission cryptée

Type de connexion SSH (4)

Procédure ssh2 :

- identique à ssh1
  1. Authentification par machine connue `known_hosts`
  2. Authentification par clé publique
  3. Authentification par mot de passe

  Paramètre `PreferredAuthentications` : ordre de préférence des types de connexion
- Utilisation des algorithmes DSA ou RSA
  Chiffrement des données (tripleDES, Blowfish, CAST128, Arcfour)

Fichiers de configuration

Fichier `/etc/ssh/sshd_config`
Quelques options importantes :

- AllowHosts : Liste des adresses IP autorisées à se connecter
- DenyHosts : Liste des adresses IP non autorisées à se connecter
- AllowUser : Liste des utilisateurs autorisés à se connecter
- DenyUser : Liste des utilisateurs non autorisés à se connecter
- X11Forwarding : Redirection graphique (X11)
  Autorisation d'ouverture de fenêtres graphiques

Nécessite un redémarrage du service

Client SSH (1)

- Fichier de configuration : `/etc/ssh/ssh_config`
  Définition de caractéristiques des sessions clientes
  Spécifique ou non à une machine :

```
Host 192.168.60.10
  ForwardX11 yes
  ForwardX11Trusted yes
  Protocol 2,1
    StrictHostKeyChecking yes
```

Client SSH (1)

```
Host *
  ForwardX11 yes
  ForwardX11Trusted yes
  Protocol 2,1
    StrictHostKeyChecking no
```

(`StrictHostKeyChecking` : pas d'ajout des clés des machines, refus de connexion lors d'un changement de clé)

- Connexion : `ssh utilisateur@machine`

  ```
  ssh monnin@lipn
  slogin -l monnin lipn
  ```

Équivalence par clé publique/privée

Échange de clé pour authentifier l'utilisateur sur le serveur

1. Création de la paire de clé publique/privée
   Chez l'utilisateur :

   ```
   ssh-keygen -t rsa
   ssh-keygen -t dsa
   ```

   Génération des fichiers `$HOME/.ssh/id_rsa` ou `$HOME/.ssh/id_dsa` (clé privée) et `$HOME/.ssh/id_rsa.pub` ou `$HOME/.ssh/id_dsa.pub` (clé publique)
2. Envoi de la clé publique au serveur (`$HOME/.ssh/id_rsa.pub` ou `$HOME/.ssh/id_dsa.pub`)

Cryptage des données avec PGP

- `GnuPG` : version gratuite
- Cryptage de données
- Authentification de l'origine d'un document
- Utilisation d'un système de clés publique/privée :
  - Diffusion de la clé publique à tout le monde
  - Conservation de clé privée
  - Signature de clés publiques (authentification de la clé publique)
- Algorithmes : IDEA, RSA, TripleDES

Création d'une clef (1)

`pgp -kg ou gpg --gen-key`

1. Choix de l'algorithme de cryptage : RSA (incompatible avec GnuPG) ou DSS/DH
2. Définition de la taille de la clé : 1024 bits
3. Persistance de la clé : délai de validité
4. Identifiant : xavier.monnin@lipn.univ-paris13.fr
5. Phrase utilisée comme mot de passe (voir principe sur les mots de passe) : utiliser une phrase complexe (augmente la sécurité)
   Clé générée : utilisée pour s'identifier

Création d'une clef (2)

6 Génération d'une clé de cryptage des données si nécessaire
7 Ajout d'un nombre aléatoire à l'aide de la frappe d'une touche sur le clavier
8 Création d'une clé publique (`pubring.pkr` ou `pubring.gpg`) et d'une clé privé (`secring.pkr` ou `secring.gpg`)

Exportation de la clé publique

Suite de caractères ASCII
Indiquer l'identifiant de la clé

```
pgp -kxa 'Xavier Monnin (cle pgp)' pubring.pkr
gpg -a --export 'Xavier Monnin (cle gpg)
```

Exemple de clé GPG

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)

mIsEQoYJkwEEAKBF1NYYW+EhM3RTrpS/nyFyolXJ9Ut+L7QaLMhl63VtXPaA+nHG
P7pZj1+1/se3sN7wd+Aoo1TFO2GZbIsQyujIm/libhS/r3n9LElM4oAuYJCVp4V6
ZOvCQlASYx4KtsSEJOoaE87qGh77SU4pn284aJelQtikTWTjtFxBZezDAAYptEZU
aGllcnJ5IEhhbW9uICh0ZXNOIGRlIGNsZSBwZ3AgMikgPHRoaWVycnkuaGFtb25A
bGlwbi51bml2LXBhcmlzMTMuZnI+iLQEEwECAB4FAkKGCZMCGwMGCwkIBwMCAxUC
AwMWAgECHgECF4AACgkQi/9RQA+h+dQLKgP/VgMQ/ane6KVrF4Wm6ca9HvFiMvwB
twRl6x36mIP+k8jhdQOdA7+ORNsjWCvS7WudcSjYIwVpPxtGBPTfZR+NMdwqUOtI
4fyzrr8MOpTSW8d419gPeDAgnMXFFMsrCJZgHqoZzzOXBEDtPPhKY/kSiYP0tg04
vxnnS8WdIjHvYAk=
=XhGV
-----END PGP PUBLIC KEY BLOCK-----
```

Importation d'une clé publique

Vérification de l'authenticité d'une donnée (ou crypter une donnée)

- Source d'un programme
- Document

Nécessite la clé (dans un fichier)

```
pgp -ka fichier_contenant_la_cle
```

```
gpg --import fichier_contenant_la_cle
```

Exemple :

```
$ gpg --import KEYS
gpg: clé 2719AF35: clé publique "Ben Laurie <ben@gonzo.ben.algroup.co.uk>" impo
gpg: clé A99F75DD: clé publique "Rodent of Unusual Size <coar@ACM.Org>" importé
```

Cryptage de données (1)

Syntaxe :

```
pgp -ea fichier Destinataire [message]
```

```
gpg -e[a] fichier [message]
```

```
   + indication du destinataire
```

Destinataire : sélection de la clé publique à utiliser

Cryptage de données (2)

Exemple :

```
$  gpg -ea test.html
Vous n'avez pas spécifié de nom d'utilisateur. (vous pouvez
utiliser «-r»)


Entrez le nom d'utilisateur, en terminant par une ligne vide: Xavier Monnin
Added 1024g/B62FFEEF 2005-05-14 "Xavier Monnin (test de cle pgp) <xavier.monnir

Entrez le nom d'utilisateur, en terminant par une ligne vide:
```

Génération d'un fichier au format ASCII correspondant au fichier `test.html` crypté

Cryptage de données (3)

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.2.4 (GNU/Linux)

hQEOAz7OBz7P72EJEAP+OZORv+1ANSrO4KLI0qJDSJn9BiHr4RKTZi69Zx5p8eOp
wASp6ZBkFhkd3INpTc/SWHT1HZ26Hu2ReRPzrhvf4KT8aEdq/8MUQxI9C/e0e9tQ
uvXf+0o5ydWO0P+nlmDv3Nne5FVXBx4AXcP2HkFxi2Jbg3cM/rZudVkRA0Y1qbgD
/A7+N8+Uy3riWIgwg1iYOET1kz7hG5xrjOTVOGZJg8/RfmhM7enqG0g2vty+sNTZ
uR1BJDDxI1Dh6Ht1d8bLu1AZS3AOy/mfd520DuR4e7MF2pFqwsGUlwAX1vl2NfZS
0blEI0r5KBQEJmfj+HymRxzVpvuRduMFo30+JtAEPy7a0l4Bn55KE+pgqw2BHABW
kiqvhGYiqQjI+aVYvrhCqgPrCy8/7+TRQ9BKLyfLpZiokS/Q4pty9W7Ety+8fKlI
rfN4U2HwtGX2hyIrzxxkV4i+0DqN5BuIVy2VAUKzG09h
=Xlcu
-----END PGP MESSAGE-----
```

Important : Signer les messages cryptés (éviter l'usurpation d'identité)

Décrypter des données

Pour décrypter c'est encore plus simple :

```
pgp fichier_crypté
gpg fichier_crypté
```

Décodage du fichier

```
$ gpg  -d test.txt.asc

Vous avez besoin d'un mot de passe pour déverrouiller la clé secrète pou
l'utilisateur: "Xavier Monnin <xavier.monnin@lipn.univ-paris13.fr>"
clé de 1024 bits ELG-E, ID CFEF6109, créée le 2005-05-14 (ID clé princip

gpg: chiffré avec une clé de 1024 bits ELG-E, ID CFEF6109, créée le 2005
      "Xavier Monnin <xavier.monnin@lipn.univ-paris13.fr>"
Fichier de test de cryptage GPG
```

Signer des données (1)

Signature d'un document : Assurance sur l'expéditeur
Création d'une signature :

- Pour PGP :

  `pgp -sta signature.txt -u Identifiant`
  `gpg -sa fichier`

  fichier `signature.txt`

  `xavier.monnin@lipn.univ-paris13.fr`

  Fichier contenant la signature chiffrée : `signature.txt.asc`

Signer des données (1)

- Pour GPG :

```
# signature et compression du message
gpg -sa [message]

# signature uniquement
gpg --detach-sign [message]
```

Chiffrement et sgnature :

```
gpg -u Expéditeur -r Destinataire -sa -sign
                                --encrypt [message]
```

Vérifier une signature

Vérification d'une signature : nécessite la clé publique de l'expéditeur

- Pour PGP

  ```
  pgp signature.txt.asc
  ```

- Pour GPG

  ```
  # importation d'une clé
  gpg --import Fichier_clé_expéditeur

  # Vérification

  gpg --verify message
  ```

Exemple de mauvaise signature (1)

Utilisé notamment pour la vérification de paquetage de programme (source ou binaire)

```
$ cat test.txt
Fichier de test de cryptage GPG
!

$ gpg --detach-sign -a test.txt

Vous avez besoin d'un mot de passe pour déverrouiller la clé secrète pour
l'utilisateur: "Xavier Monnin <xavier.monnin@lipn.univ-paris13.fr>"
clé de 1024 bits DSA, ID D44F3D5D, créée le 2005-05-14

$ ls -l test.txt*
-rw-r--r--  1 monnin monnin  34 mai 14 17:52 test.txt
-rw-r--r--  1 monnin monnin 189 mai 14 18:27 test.txt.asc
```

Exemple de mauvaise signature

```
$ gpg --verify test.txt.asc
gpg: Signature faite sam 14 mai 2005 18:27:57 CEST avec la clé DSA ID D44F3D5D
gpg: Bonne signature de "Xavier Monnin <xavier.monnin@lipn.univ-paris13.fr>"

$ cat !$
cat test.txt
Fichier de cryptage GPG
!

$ gpg --verify test.txt.asc
gpg: Signature faite sam 14 mai 2005 18:27:57 CEST avec la clé DSA ID D44F3D5D
gpg: MAUVAISE signature de "Xavier Monnin <xavier.monnin@lipn.univ-paris13.fr>"
```

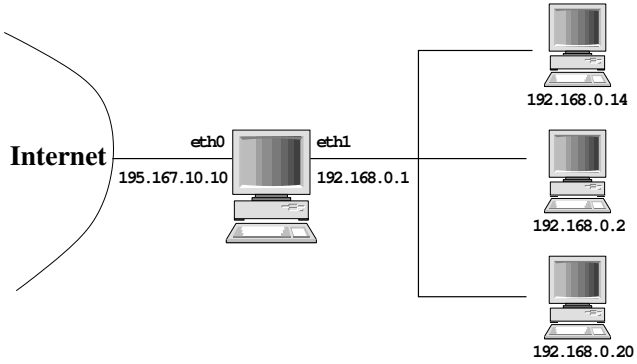
Firewall et NAT

- Firewall : Pare-feu
  Filtrage des paquets entrants et sortants
  Protection contre les intrusions
- NAT : Masquage d'adresse IP
  Objectif : connecter un réseau privé à l'internet avec un seul point d'entrée

$\Longrightarrow$ Deux outils indispensables pour la mise en place d'un réseau privé sécurisé

Firewall

Protection des réseaux d'entreprise des attaques venant de l'Internet
Machine entre le réseau et l'internet



Filtrage du trafic réseau

Principe de fonctionnement

- Manipulation des paquets (couche réseau de l'OSI)
- Analyse de paquets entrants et sortants (type, adresses source et destination, ports)
- Travail sur la couche IP (nécessite de la mémoire)
  Sous Linux, filtrage IP intégré au noyau $\Longrightarrow$ réduction des besoins en mémoire

Avantages et inconvénients

Avantages :

- transparents pour les utilisateurs
- pas d'authentification nécessaire pour sortir
- pas de paramétrage spécifique

Inconvénients :

- Pas d'authentification par utilisateur mais par adresse IP
- Pas de possibilité d'interdire la sortie à certains utilisateurs (sauf si bijection entre utilisateur et machine/adresse IP)

Firewall proxy

Contrôle et analyse le trafic réseau avec l'extérieur
Utilisation d'un cache (stockage de données en local pour réduire le trafic réseau)
Deux types de fonctionnement :

- Proxy d'applications
- Proxy «*sock*»

Exemple de Proxy : Squid, TIS Firewall toolkit (FWTK), Socks

Proxy d'applications

Fonctionnement : intermédiaire entre l'application (locale) et l'extérieur
Avantages :

- Très sécurisé pour la machine client
- Réduction du trafic réseau
- Authentification possible

Inconvénients :

- Nécessite une configuration spécifique des clients
- Installation sur le proxy des applications serveurs (gestion de chaque protocole)
- Grande consommation de ressources

Proxy «*sock*»

Pas de travail sur les applications
Fonctionnement comme un Firewall filtrant
Pas d'authentification, mais enregistrement de l'utilisateur demandant une connexion

IP Masquerading

Masquage d'adresse IP
Translation d'adresse IP : Network Address Translation (NAT)
Partage d'une adresse IP pour plusieurs machines
Principe :

- Machine connectée à l'Internet (modem, ADSL, Liaison Spécialisée)
- Machine interne au réseau (adresse privée)
- Machine interne doit passer pour la machine connectée à l'Internet

Mise en place d'un pare-feu (1)

Filtrage des paquets Ethernet :

- entrants (`input`)
- sortants (`output`)
- transmis (`forward`)

Règles définies suivant l'interface utilisée (eth0)

Mise en place d'un pare-feu (2)

Trois comportements :

- Rejet (la source est prévenue) : REJECT
  Comportement déconseillé
- Rejet (la source n'est pas prévenue) : DENY ou DROP
- Acceptation : ACCEPT
- Mise en attente : QUEUE

Chaîne : suite de règles
Mise en place d'un Firewall : définition de règle pour chaque chaîne
Si aucune règle ne s'applique, application de politique par défaut de la chaîne

Préparation

- Firewall : Machine avec deux cartes réseau, ou au moins deux interfaces réseau
  - Adresse privée (plage 192.168.XXX.XXX par exemple)
    (passerelle/firewall : 192.168.0.1)
  - Adresse(s) publique(s) (routable sur Internet)
- Définir les services devant être disponible pour l'extérieur (Web/http, ssh, smtp, pop3/pop3s, imap/imaps

$\Longrightarrow$ Mise en place conjointe du filtrage, masquerading, routage

Configuration de l'IP masquerade

Configuration du noyau :

- Chargement des modules pour l'IP Masquerade (`ip_masq*`)

Activation :

Accès vers l'extérieur pour toutes les machines de réseau

Exemple (1)

Exemple avec les `ipchains` (noyau linux 2.2) :

```
# Aucun paquet n'est transmis

ipchains -P forward DENY

# Transmission des paquets issus du réseau 192.168.0.0
ipchains -A forward -s yyy.yyy.yyy.yyy/x -j MASQ
```

`yyy.yyy.yyy.yyy` : adresse réseau (192.168.0.0)
`x` : maque réseau

Exemple (2)

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.0.0/24 -j MASQ
ou
ipchains -P forward DENY
ipchains -A forward -s 192.168.0.0/255.255.255.0 -j MASQ
```

Configuration d'un Firewall

Exemple avec les Iptables (noyau 2.4)
Utilisation de trois tables (chaînes et cibles associées :

- `NAT`
- `FILTER`
- `MANGLE`

La table NAT

Network Address Translation – IP Masquerading

table utilisée pour la translation d'adresse ou la translation de port
Chaînes :

- PREROUTING : opération à effectuer à l'entrée du firewall
- POSTROUTING : opération à effectuer à la sortie du firewall

Cibles :

- SNAT : modification de l'adresse source du paquet
- DNAT : modification de l'adresse destination du paquet
- MASQUERADE : Transformation des paquets sortants Les paquets semblent provenir du firewall (allocation dynamique d'un port)
  Réponse sur le port : utilisation d'une table de correspondance (fait suivre le paquet sur la bonne machine)

Les tables FILTER et MANGLE

- `FILTER` : table par défaut (si aucune n'est spécifiée)
  Contient toutes les règles de filtrage (pour FORWARD, INPUT et OUTPUT)
  Cibles disponibles : ACCEPT, DENY, DROP et REJECT
- `Mangle` : table contenant les règles pour la modification de paquets
  Peu utilisée

Opérations

- -P (policy) : définition de la politique d'acheminement
- -N (New chain) : Définition d'une nouvelle chaîne
- -L (list) : affichage de la liste des règles d'une chaîne
- -D (delete) : suppression des règles d'une chaîne
- -F (flush)) : Vidage des règles d'une chaîne
- -A (add) : Ajout d'une règle à une chaîne

Syntaxe des iptables

Exemple d'ajout de règles :

```
iptables -A chaine -i interface -s source --sport port_source
-d destination --dport port_destination -j police
```

- chaîne : input,output,forward
- interface : eth0, lo, eth1
- source : adresse IP particulière (192.168.1.1/24) ou classe d'adresse IP entière (192.168.1.0/24).
  /24 : masque de sous réseaux (255.255.255.0)
- polices : ACCEPT,DENY,REJECT

Exemple de filtres

Anti-spoofing :
spoofing : changement de l'adresse IP source (par un pirate)
Le filtre doit rejeter tous paquets IP d'adresse 192.168.0.XXX venant de eth0 (interface extérieure) :

```
iptables -A INPUT -i eth0 -s 192.168.0.0/24 -d 0.0.0.0 -j DENY
```

Visualisation des règles :

```
iptables -L INPUT
```

Configuration d'un Firewall/NAT

Politiques :

1. Rejet de tous les paquets. Ouverture au fur et à mesure
   Politique fortement conseillée
2. Ouverture à tous les paquets. Fermeture aux paquets dangereux

Initialisation :

```
# Autorisation pour le forwarding dans le noyau
echo 1 > /proc/sys/net/ipv4/ip_forward

# Insertion des modules
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_conntrack

# Mise à zéro
iptables -F
iptables -F -t mangle
iptables -F -t nat
iptables -X
iptables -X -t mangle
iptables -X -t nat
```

```
# Initialisation de la politique par défaut
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT

iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
```

Arrêt :

```
iptables -F
iptables -F -t mangle
iptables -F -t nat
iptables -X
iptables -X -t mangle
iptables -X -t nat

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Définition de protection au niveau ICMP (*Ping Flood)* :

```
iptables -N ICMP_CHAIN

# Protection Ping Flood.

iptables -A ICMP_CHAIN -p icmp --icmp-type echo-request
        -m limit --limit 5/s --limit-burst 10 -j ACCEPT

# Blocage des ICMP-reply
iptables -A ICMP_CHAIN -p icmp --icmp-type echo-reply
     -m limit --limit 5/é --limit-burst  10 -j ACCEPT

# Blocage des ICMP-time-exceeded
iptables -A ICMP_CHAIN -p icmp --icmp-type time-exceeded
         -m limit --limit 5/s --limit-burst 10 -j ACCEPT
```

```
# Blocage des ICMP-parameter-problem
iptables -A ICMP_CHAIN -p icmp --icmp-type
           parameter-problem -m limit --limit 5/s
           --limit-burst 10 -j ACCEPT
```

Mise en place du NAT :

```
iptables -A PREROUTING -t nat -i eth0 -p tcp -m tcp
            -d 195.167.10.10 --dport http -j DNAT
            --to-destination 192.168.0.20

iptables -A POSTROUTING -t nat -o eth0 -p tcp -m tcp
               --sport http -s 192.168.0.20 -j SNAT
               --to-source 195.167.10.10
```

Mise en place de logs :
ATTENTION : peut nécessité beaucoup de place disque (ou une rotation des logs adaptée)

```
iptables -A INPUT -m limit --limit 3/s --limit-burst 10
                      -j LOG --log-prefix "fp= INPUT "
```

Filtrage :

```
# Suppression des broadcasts
iptables -A INPUT -s 0.0.0.255/0.0.0.255 -j DROP
iptables -A INPUT -d 0.0.0.255/0.0.0.255 -j DROP

# Suppression des broadcasts
iptables -A FORWARD -s 0.0.0.255/0.0.0.255 -j DROP
iptables -A FORWARD -d 0.0.0.255/0.0.0.255 -j DROP

# On fait suivre les connexions déja établies
iptables -A FORWARD -i eth0 -o eth1 -m state
           --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -o eth0 -i eth1 -m state
       --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
# On fait suivre les connexions internes aux réseaux
iptables -A FORWARD -o eth1 -p tcp -m tcp
          -d 192.168.0.20 --dport http -j ACCEPT
```