Marc SCHAEFER
Epervier 15
2053 Cernier
Ing. inf. dipl. EPFL

Conseil et
réalisations en
informatique libre
http://www.cril.ch/

Email : schaefer@alphanet.ch
Tél. : +41 32 841 40 14
Fax : +41 32 841 40 81
Mobile : +41 79 502 56 92

Support de cours de préparation de l'examen

LPI-102

Marc SCHAEFER
schaefer@alphanet.ch

C+R
Informatique libre

25 avril 2008

Cernier, le 25 avril 2008

Avant-propos

Ce cours est basé sur mon expérience d'administration système UNIX depuis 1987, ainsi que de mes expériences faites dès 1995 sur GNU/Linux et en particulier avec **Debian GNU/Linux**, un système de qualité, à la fois à but personnel et en entreprise, comme conseiller indépendant, ingénieur système et développeur.

Il est également basé sur mes expériences de formateur en logiciels libres et notamment sur GNU/Linux, notamment dans le cadre des formations postgrades ES en cours du soir à l'**ESNIG** à Neuchâtel depuis 2000, ainsi que de nombreuses formations que j'ai proposées en entreprise ou dans le cadre d'associations comme **/ch/open** ou le **GULL**.

Il est également basé sur la lecture de documentations systèmes (page man, info), ainsi que de ressources Internet.

Réalisé dans le but d'une certification, le cours tente de suivre exactement le programme obligatoire, de donner les points essentiels, tout en proposant un contexte solide et des exemples à jour. J'essaie absolument d'éviter le bachottage uniquement en vue de la certification : le but est que la personne qui suit ce cours puisse en bénéficier largement pour son travail de tous les jours, notamment en apprenant comment trouver l'information et analyser les problèmes.

Ce cours écrit s'accompagne normalement d'extraits d'autres cours, dont les droits sont en général également en licence FDL, parfois co-écrits par d'autres auteurs. Il insiste surtout sur les compléments nécessaires pour la certification.

Bon apprentissage !

Licence et droits d'auteurs

Marc SCHAEFER
Ing. inf. dipl. EPFL
Conseil et réalisations en informatique libre (CRIL)
http://www.cril.ch/

Table des matières

1. Kernel

Contenu du chapitre

- comment interroger, charger et décharger les pilotes sous forme de *modules*
- comment reconfigurer, construire et installer un kernel et ses modules

Buts du chapitre

1

- savoir déterminer les modules kernel chargés et s'ils peuvent être déchargés, déterminer les paramètres acceptés par un module kernel donné, manuellement charger et décharger des modules et gérer les aliases ou la désactivation d'un module
- savoir particulariser/adapter la configuration d'un kernel, reconstruire un kernel et ses modules, l'installer avec ses modules
- savoir configurer le gestionnaire de démarrage de manière à utiliser un kernel nouvellement installé

support de cours additionnel : cours **Administration** et **Installation**

Ce premier chapitre traite principalement des pilotes sous forme de modules du kernel (qui sont chargeables et déchargeables dynamiquement), ainsi que de la configuration, regénération (compilation) et installation d'un kernel et de ses modules, y compris l'activation de celui-ci au démarrage par le chargeur de démarrage.

Gestion des modules – 1.1

Résumé des concepts importants

- les pilotes sont en général aujourd'hui fournis sous forme de *modules* kernel (livrés sous forme de fichiers séparés, p.ex. `ext3.o`[a]).
- un module peut dépendre d'un autre, la commande `modprobe` détermine automatiquement les dépendances via `/lib/modules/kernel-version/modules.dep` (généré via `depmod`)
- un module ne peut être déchargé que si aucun autre module chargé ne dépend de lui et qu'il n'est pas utilisé par le kernel (voir le compteur affiché dans `lsmod`)
- des aliases et paramètres de modules peuvent être configurés automatiquement dans `/etc/modules.conf` ou `/etc/modprobe.conf`[b]
- commandes : `depmod`, `insmod`, `lsmod`, `rmmod`, `modinfo`, `modprobe`, `uname`

[a] dès la version 2.6 du kernel, l'extension est désormais `.ko`.
[b] On utilisera plutôt un répertoire `/etc/modprobe.d/`

2

Les pilotes sous Linux Les pilotes (de l'anglais : *drivers*) sont des logiciels généralement de petite taille (quelques kilo-octets) qui gèrent des aspects relativement bas niveau d'un système GNU/Linux.

Par exemple :

msdos Support du système de fichiers MS-DOS

vfat Support du système de fichier VFAT

8139too Support des cartes réseau à chipset 8139 (Realtek p.ex.)

Ces pilotes peuvent être soit intégrés au kernel monolithique (ce qui signifie qu'il faut alors recompiler le kernel s'il y manque des pilotes), ou mis à dispositions sous forme de fichiers séparés, appelés *modules*.

Lorsqu'ils sont intégrés au kernel, il est possible de les configurer en spécifiant des options de la ligne de commande du kernel, p.ex. via des options du programme de démarrage (LILO ou GRUB). Par exemple, l'option `hdc=ide-scsi` propose de configurer le maître du secondaire IDE en tant que périphérique vu comme SCSI. On peut consulter ces arguments effectifs en affichant le pseudo-fichier `/proc/cmdline`.

Les pilotes sous forme de modules kernel Ces pilotes sont alors en général stockés dans des sous-répertoires du répertoire `/lib/modules/kernel-version`, où `kernel-version` est la version du kernel concerné (p.ex. `2.4.21-8`).

Par exemple, le module kernel du pilote de système de fichiers MS-DOS est `msdos.o` (v2.4 ; ou `msdos.ko` en v2.6), dans le répertoire `/lib/modules/2.4.21-8/kernel/fs/msdos`.

Plusieurs versions différentes des modules peuvent donc être installées si plusieurs kernels différents sont installés et peuvent ou non être activés depuis le programme de démarrage.

Lorsqu'ils sont disponibles en modules, deux problèmes se posent :

1. il faut charger le module (manuellement, statiquement via p.ex. `/etc/modules`, ou dynamiquement via hotplug/udev et `/etc/modprobe.d/`).
2. il faut indiquer les paramètres éventuels au chargement (manuellement ou via configuration)

Notons que si l'on n'utilise pas d'initrd, tous les pilotes nécessaires (matériel, disque, systèmes de fichiers, etc) doivent être compilés en dur dans le kernel. Avec un initrd, un sous-ensemble des modules nécessaires est intégré et les modules sont chargés dynamiquement.

Interrogation, chargement et déchargement de modules Quelques exemples : (en 2.6)

```
root@reliant:~# uname -r
2.6.15-29-686

root@reliant:~# lsmod | more
Module                  Size  Used by
ahci                   18020  12
speedstep_centrino      8752  1
cpufreq_userspace       6496  2
libata                 83440  1 ahci
scsi_mod              146088  6 sr_mod,sbp2,sg,sd_mod,ahci,libata
sg                     40160  0
sbp2                   25060  0
ieee1394              306520  2 sbp2,ohci1394
ide_cd                 35780  0
cdrom                  41408  2 sr_mod,ide_cd
sr_mod                 17988  0
ext3                  148456  5
jbd                    65876  1 ext3
[ ... ]

root@reliant:~# rmmod sr_mod; lsmod | egrep 'sr_mod|scsi_mod'
scsi_mod              146088  5 sbp2,sg,sd_mod,ahci,libata

root@reliant:~# modinfo sr_mod
filename:       /lib/modules/2.6.15-29-686/kernel/drivers/scsi/sr_mod.ko
license:        GPL
vermagic:       2.6.15-29-686 SMP preempt 686 gcc-4.0
depends:        scsi_mod,cdrom
srcversion:     8A9AC1284A6A880A9FC5193
parm:           xa_test:int

root@reliant:~# insmod sr_mod
insmod: can't read 'sr_mod': No such file or directory

root@reliant:~# modprobe sr_mod; rmmod sr_mod; modprobe sr_mod xa_test=1

root@reliant:~# rmmod cdrom
ERROR: Module cdrom is in use by ide_cd

root@reliant:~# rmmod ide_cd; rmmod cdrom
root@reliant:~# modprobe cdrom
```

Consulter également la sortie de la commande `dmesg`, les pilotes kernels y déposent de l'information. Attention, en 2.6 avec udev, il y a de grande chance que certains modules se rechargent tout seul.

Dépendances des modules On a vu dans l'exemple que comme `cdrom` dépend de `ide_cd`, on ne peut simplement l'insérer par un insmod, il faut utiliser `modprobe`. Cette commande résoudra les dépendances automatiquement, via les informations fournies dans le fichier `/lib/modules/kernel-version/modules.dep`.

Ce fichier est regénéré via la commande `depmod` (en général automatiquement, sauf si l'on copie un module à la main p.ex.).

Configuration des paramètres et du chargement automatique En particulier sur un système avec hotplug et udev, la plupart des modules seront chargés automatiquement et à la demande. Si des paramètres sont désirés, ou que l'on désire interdire le chargement d'un module, ou en charger un autre, il est possible de configurer.

Sous Debian, cela se fait dans le répertoire `/etc/modprobe.d`. On peut par exemple supprimer un module en remplaçant son alias par `off`, ou ajouter des paramètres (consultez par exemple `/etc/modprobe.d/options`).

Autres configurations Parfois, on désire charger statiquement des modules (par exemple pour garantir l'ordre au démarrage). Cela peut se faire, sous Debian et dérivés, via le fichier `/etc/modules`. Les paramètres éventuels des modules sont spécifiés comme dans le cas d'un `modprobe`.

Exercices

1. quelle est la version du kernel qui fonctionne actuellement ? où sont ses modules ?
2. essayez de décharger un module qui est utilisé (p.ex. en 2.6 `parport`). Que devez-vous faire ? Si vous avez des problèmes expliquez et résolvez-les.
3. comment configurer l'adresse I/O du ou des ports parallèle ? (indication : pilote `lp`) Comment feriez-vous pour que ce paramètre soit toujours utilisé ?
4. quel argument permet de regénérer les dépendances de modules ? vérifiez que le fichier a changé.
5. est-ce que le module `ipv6` est chargé ? comment s'assurer que ce module ne sera plus chargé ? redémarrez et vérifiez !
6. configurez votre système pour que la commande `modprobe bla` charge un module quelconque et testez.

Configuration, compilation et installation d'un kernel – 1.2

Résumé des concepts importants

- fichiers : `/usr/src/linux`, `/usr/src/linux/.config`, `/lib/modules/kernel-version/*` `/boot/*`
- commandes : `make`
- cibles de `make` : `all`, `config`, `menuconfig`, `xconfig`, `gconfig`, `oldconfig`, `modules`, `install`, `modules_install`, `depmod`, `rpm-pkg`, `binrpm-pkg`, `dev-pkg`

3

Lectures supplémentaires

- méthode Debian dans le cours **Installation**

Les étapes de regénération du kernel et des modules Les étapes pour regénérer un kernel et ses modules sont :

1. obtenir la *source* du kernel sous une forme ou une autre :
   - source pure (pristine, vanilla) de `http://www.ch.kernel.org/pub/linux/kernel/` (assez dangereux depuis v2.6)
   - package source de la distribution considérée (en général avec les patches de la distribution appliqués, p.ex. `apt-get install linux-source`)

   optionnellement : vérifier les signatures électroniques (GPG/PGP)
2. désarchiver éventuellement cette source, usuellement dans `/usr/src/linux`
3. appliquer des patches éventuels supplémentaires
4. installer les utilitaires de compilation nécessaires, p.ex. `kernel-package` et ces dépendances si l'on utilise la méthode Debian (ou au moins un compilateur, make, etc)
5. copier éventuellement une configuration compatible depuis `/boot/config-*` dans le fichier `.config` et lancer `make oldconfig` pour tester/importer ce fichier
6. configurer ce kernel avec une des méthodes désirées (questions, mode texte, divers modes graphiques)
7. compiler le kernel et les modules
   - `make all`
   - ou `make bzImage modules`
   - voir aussi `make help`
8. installer le kernel obtenu et ses modules

9. recompiler éventuellement des sous-systèmes kernel non intégrés dans la distribution du kernel

Cette procédure est largement automatisable grâce aux outils Debian du package `kernel-package`, notamment `make-kpkg`. Ces outils génèrent d'ailleurs un package Debian et ne nécessitent pas un répertoire système ni les droits root pour la compilation et la génération et permettent de préparer un package sur une machine différente de la machine cible, très facilement.

Méthode manuelle :

```
demo:~$ sudo bash
root:~# mkdir /usr/src/linux
root:~# cd /usr/src/linux
root:/usr/src# kup=http://www.ch.kernel.org/pub/linux/kernel/
root:/usr/src# wget $kup/v2.6/linux-2.6.23.12.tar.bz2
root:/usr/src# tar xjf linux-2.6.23.12.tar.bz2
root:/usr/src# ln -s linux-2.6.23.12 linux
root:/usr/src/linux# cp /boot/config-2.6.20-15-generic .config
root:/usr/src/linux/linux# apt-get install build-essential
root:/usr/src/linux# make oldconfig
root:/usr/src/linux# make all
```

Note : on n'a pas changé la configuration, on a simplement repris une configuration antérieure, et quelques questions supplémentaires pour de nouveaux pilotes ou fonctionnalités ont été posées. Le kernel tel que créé ci-dessus ne fonctionnera pas, sauf si on crée aussi un initrd, ou que l'on s'assure que tous les pilotes nécessaires (SATA, ext3, etc) ne sont pas en modules.

L'installation proprement dite Le kernel et les modules compilés peuvent être soit installés sous forme de package, soit via `make`, soit manuellement. Nous décrivons ci-après le cas `make` :

```
root:/usr/src/linux# make install
root:/usr/src/linux# make modules_install
```

On voit alors que les fichiers suivants sont créés (voire écrasés ...) :
- `vmlinuz-2.6.23.12`
- `/boot/config-2.6.23.12`
- `/boot/System.map-2.6.23.12`
- `/lib/modules/2.6.23.12/*`

Il y a aussi quelques liens symboliques qui sont positionnés dans /boot (et qui peuvent être utilisés pour configurer le démarreur, p.ex.)

La configuration éventuelle d'un démarreur La configuration du démarrage ayant été vue dans LPI-101, section 3.2, elle ne sera pas revue ici.

Exercices

1. essayez les diverses méthodes de configuration du kernel (oldconfig, config, menuconfig, xconfig, gconfig)
2. que trouvez-vous dans le fichier `.config`. Si vous le copiez d'une autre machine ou le changez à la main, que devriez-vous faire ?
3. comme la compilation de base du kernel ne crée pas d'initrd, que devez-vous absolument faire ?
4. quels sont les 4 fichiers ou répertoires résultant de la compilation manuelle du kernel et où les placez-vous (en supposant que vous n'employez ni `make install` ni `make modules_install`).
5. quel est le danger de modifier directement /boot ou /lib/modules sans passer par un package ?
6. à l'aide de `make help`, déterminez à quoi servent les cibles `deb-pkg`, `rpm-pkg` et `binrpm-pkg`.

2. Démarrage, arrêt et niveaux d'exécution

Contenu du chapitre

- comment influencer et analyser le démarrage du système via les options de démarrage interactives et les journaux systèmes
- comment gérer le niveau d'exécution du système (p.ex. mode maintenance, arrêt et redémarrage)

4

Buts du chapitre

- savoir trouver l'information de démarrage dans les journaux (logs) systèmes et le tampon cyclique de journal du kernel
- savoir configurer des options du kernel dans un démarreur au moment du démarrage
- savoir changer le niveau d'exécution, passer en mode maintenance (single-user) arrêter et redémarrer le système ; configurer le niveau par défaut
- savoir informer les utilisateurs et arrêter les processus correctement

support de cours additionnel : cours **Administration** et **Installation**

Ce chapitre traite principalement des options interactives au démarrage, de l'analyse des logs systèmes et des niveaux d'exécutions (run levels).

Démarrer le système – 2.1

Résumé des concepts importants

- les logs systèmes après le démarrage du kernel sont disponibles dans /var/log/messages[a]
- la commande `dmesg` donne accès au kernel cyclical log buffer : juste après le démarrage en général on obtient les messages du kernel du démarrage[b]
- on peut donner des options de démarrage pour LILO ou grub

Lectures supplémentaires

- http://www.debian.org/releases/stable/i386/ch05s02.html.fr
- http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html

[a] tous les logs, sauf peut-être les logs sensibles dans /var/log/auth.log par exemple, sont dans /var/log/syslog

[b] sous Debian on retrouvera ces logs sauvegardés dans /var/log/dmesg.

Exercices

1. Redémarrez votre système. Accédez au prompt du gestionnaire de démarrage (boot-loader : p.ex. LILO ou grub). Ajoutez une option kernel `mem=128m`. Que fait cette option (pour le savoir, consultez les logs systèmes !). Faites de même avec l'option `noapic`. Indication : vous pouvez trouver une liste des options kernel ici : http://www.cyberciti.biz/howto/question/static/linux-kernel-parameters.php (ou dans la documentation du kernel `kernel-parameters.txt`)
2. A quoi servent les options
   - `root=/dev/hdc3` ?
   - `single`
   - `init=/bin/sh`
3. Où trouvez-vous les journaux des programmes ayant démarré ?
4. Que se passe-t-il s'il y a trop d'informations au démarrage du kernel et comment y remédier ?

Changement des niveaux d'exécution – 2.2

Résumé des concepts importants

- gestion des niveaux d'exécution : `init` (configuration : `/etc/inittab`[a])
- niveau d'exécution par défaut : entrée `initdefault`
- changement de niveau d'exécution : `telinit X` (ou `init X`)
- rappel : niveaux 1, 6, 0 ; 2 à 5 ; `/etc/init.d`, `/etc/rc?.d`
- `shutdown` : `-h` (halt) ou `-r` (reboot) (autres options !)
- `wall`
- `ps`, `kill`, `killall`

[a] /etc/events.d dans une Ubuntu récente.

6

Exercices

1. Comment feriez-vous un redémarrage dans 15 minutes en avertissant automatiquement les utilisateurs qu'il s'agit d'un redémarrage pour mise à jour du kernel ? Comment l'annulez-vous ?
2. A quoi sert le fichier `/etc/nologin` ?
3. Comment déterminez-vous le niveau d'exécution actuel ?
4. Comment passez-vous en mode maintenance ? Comment contrôlez-vous qu'aucun processus utilisateur ou système ne tourne encore ?
5. Comment redémarrer le système *sans* utiliser `shutdown` (ou la commande `reboot`). Même question pour un arrêt (sans `halt`) ?
6. Comment envoyez-vous un message à tous les utilisateurs connectés (console) ?
7. Configurez le niveau d'exécution par défaut comme 3 et contrôlez si tous les services sont démarrés également. Sinon (avancé) adaptez la configuration dans `/etc/rc3.d` (voire aussi dans `/etc/initab`)
8. Que doit-on faire pour activer un service directement configuré dans `/etc/inittab` et que l'on vient de modifier ?

3. Impression

Contenu du chapitre

- gestion d'un système d'impression simple et complexe
- soumission de travaux
- configuration d'imprimantes locales et distantes
- configuration de filtres d'impression

7

Buts du chapitre

- savoir configurer et surveiller les serveurs CUPS et lpd
- savoir gérer les queues d'impression
- savoir imprimer des fichiers et les convertir en PostScript si nécessaire
- savoir configurer une imprimante locale ou distante, y compris des filtres d'impression éventuels, pour une imprimante PostScript, non-PostScript, locale ou via SMB

support de cours additionnel : cours **Impression**

Ce chapitre traite de l'impression sous UNIX : concepts de base, serveurs de queues, filtres d'impression, commandes d'impression et de conversion de format, installation d'imprimantes locales ou distantes.

Gérer les imprimantes et les queues d'impression – 3.1

Résumé des concepts importants

- l'impression sous UNIX se fait en général en PostScript
- la soumission de travaux se fait soit par réseau (protocole LPD, ou plus récemment le protocole IPP), soit localement via la commande `lp` (ou `lpr`)
- des filtres convertissent éventuellement de tout format soumis à PostScript et/ou effectuent des transformations
- si l'imprimante destinataire n'est pas PostScript, des filtres convertissent au format de l'imprimante (p.ex. via ghostscript)
- commandes d'administration : `lpc`, `lpq`, `lprm`
- un daemon gère les queues d'impression (cups, lpd, lprng) ainsi que les filtres éventuels
- fichiers et configurations de CUPS

Lectures supplémentaires

- http://tldp.org/HOWTO/Printing-HOWTO/

8

Différences entre LPD, LPRNG et CUPS LPD est le daemon de spooling/impression standard de UNIX BSD et il a été porté tel quel sur GNU/Linux. Il fonctionne de manière traditionnelle : toute sa configuration se trouve dans le fichier `/etc/printcap`, que les applications consultent pour déterminer les imprimantes disponibles. On peut y ajouter des filtres externes (ghoscript, apsfilter, magicfilter, printfilter, gimp-print) pour supporter l'impression sur des imprimantes non PostScript ou pour modifier des options d'impression. Certaines versions avancées de LPD supportent également les fichiers PPD.

LPRNG ajoute un peu de sécurité à un concept relativement identique.

CUPS supporte nativement un nouveau protocole d'impression (IPP) qui permet au client d'impression de choisir plus facilement des options d'impression (p.ex. bac, recto-verso, multipage, etc) sans passer par de multiples queues d'impression. Les capacités de l'imprimante sont dérivées d'un fichier PPD (*Printer Postscript Description*) fourni par le fabricant ou créé sur la base d'un exemple. L'impression IPP est également possible depuis Microsoft Windows. L'intégration des pilotes, PPD et filtres ainsi qu'une configuration par GUI simple sont également des avantages de CUPS. Par contre, CUPS, contrairement à LPD et LPRNG est tout sauf léger.

Exercices

1. Quel est le numéro de port et le protocole de transport (TCP ou UDP) du service d'impression par réseau LPD) ?
2. Quel est le format d'impression standard UNIX ? Quel est le problème si l'imprimante ne supporte pas ce format, et comment le résoudre ?
3. Quel est l'avantage pour les postes clients ?

Installation et configuration d'imprimantes – 3.2

Résumé des concepts importants

- imprimantes locales : `/dev/lp*`, USB `/dev/usb/lp*`
- imprimantes distantes : LPD, Samba
- `/etc/printcap` : fichier de configuration *classique*
- filtres d'impression
- répertoires : `/var/spool/cups/`, `/var/spool/lpd/*`
- compatibilité : http://www.linuxprinting.org/

Lectures supplémentaires

- http://www.linuxprinting.org/
- http://www.cups.org/documentation.php

9

Installation d'un daemon simple : lpd

```
apt-get install lpr magicfilter

magicfilterconfig # configuration de filtres
```

Le fichier /etc/printcap Originellement, `/etc/printcap` avait deux rôles :

1. configurer le daemon d'impression (queues disponibles, paramètres de l'imprimante et de la queue, filtres éventuels, lieu de l'imprimante, nom de périphérique UNIX éventuel)
2. indiquer aux applications les queues d'impression disponibles

Aujourd'hui, ce fichier n'est utilisé que si l'on installe les anciens serveurs (LPD, LPRNG) et non pas p.ex. CUPS (ou alors uniquement pour compatibilité avec d'anciennes applications, dans un format simplifié).

Chaque entrée de `/etc/printcap` a un format spécial :

```
rlp|Remote printer entry:\
        :lp=:\
        :rm=remotehost:\
        :rp=remoteprinter:\
```

```
        :sd=/var/spool/lpd/remote:\
        :mx#0:\
        :sh:

ljet4l|lp:\
    :lp=/dev/lp0:\
    :if=/etc/apsfilter/basedir/bin/apsfilter:\
    :sd=/var/spool/lpd/ljet4l:\
    :lf=/var/spool/lpd/ljet4l/log:\
    :af=/var/spool/lpd/ljet4l/acct:\
    :mx#0:\
    :sh:
```

Ici, on définit tout d'abord une imprimante distante (protocole LPD, nom remoteprinter@remotehost), puis une imprimante locale (adressable comme ljet4l ou lp), qui est locale, parallèle, et gérée via apsfilter. Des flags (p.ex. `sh`, suppress header) sont également utilisés.

GUI WWW de configuration de CUPS On se connecte avec un client WWW à l'URL `http://localhost:631/`.

Exercices

1. assurez-vous que CUPS n'est pas installé : `apt-get --purge remove cupsys cupsys-client`. Installez `lpr` et `magicfilter`. Sous le daemon lpd, configurez une imprimante avec `magicfilterconfig`. Consultez les filtres créés automatiquement dans `/etc/printcap`. Consultez l'état des imprimantes.
2. toujours sous lpd, désactivez, vérifiez et activez cette imprimante. Quelles sont les autres fonctions de la commande d'administration ? Pourriez-vous changer la priorité d'un job ?
3. supprimez le fichier `/etc/printcap`. Installez les packages `cupsys` et `cupsys-bsd`. En utilisant la configuration WWW CUPS, ajoutez une nouvelle imprimante parallèle de type HP 500c. Si vous avez des hésitations sur le pilote à employer, voyez `http://www.linuxprinting.org`. Consultez les fichiers de configuration qui ont été modifiés. Faites de même pour une impression réseau LPD et une impression SMB (indication : pour ce dernier cas, le package `smbclient` doit être installé). Enfin, installez une imprimante qui n'existe pas (et donc produira des erreurs). Indication : `touch /dev/lp0` si vous n'avez pas de port parallèle.
4. désactivez votre imprimante via le GUI WWW CUPS si ce n'est pas le cas (fonction `Stop Printer`), consultez son état avec la commande `lpc`. Activez l'imprimante via la commande spécifique à CUPS `cupsenable`. Comment faire l'inverse ? Vérifiez.
5. que font les commandes `accept` et `reject` Vérifiez avec le GUI WWW.
6. comment interdisez-vous l'annonce automatique des autres serveurs CUPS ? (indication : *browsing*)
7. comment autorisez-vous l'accès à votre serveur d'impression depuis un sous-réseau ?
8. comment faites-vous en sorte que `/etc/printcap` soit géré par CUPS ?
9. comment ajoutez-vous une imprimante avec la commande `lpadmin` ? comment la supprimez-vous ?
10. comment transformez-vous un serveur CUPS en serveur compatible protocole LPD ?

Imprimer des fichiers – 3.3

Résumé des concepts importants

- on soumet un job avec `lpr` (ou `lp`)
- conversion de texte en PostScript : `a2ps`[a]
- interrogation des queues : `lpq`

[a] des plus anciennes versions de l'examen LPI-102 parlaient d'`enscript`

Exercices

1. imprimez un fichier texte quelconque sur une imprimante réseau, que constatez-vous ? que faire ?
2. utiliser `a2ps` pour générer un fichier PostScript à partir d'un fichier texte (sans imprimer) et visualisez ce fichier avec `gv` p.ex. Changez des options de `a2ps` (p.ex. nombre de pages par feuille). Une fois que vous êtes satisfaits, imprimez la sortie. Alternativement, imprimez directement en spécifiant l'imprimante avec une option d'`a2ps`.
3. imprimez sur une imprimante inexistante locale ou arrêtée, consultez la queue d'impression, puis détruisez le job. Pouvez-vous supprimer le job de quelqu'un d'autre ?

4. Documentation

Contenu du chapitre

- utilisation des manuels **man** et de la documentation dans `/usr/share/doc`
- sources collaboratives sur Internet
- communication avec les utilisateurs

11

Buts du chapitre

- savoir trouver des manpages, connaître la structuration en sections, préparer pour l'impression, configurer la commande `man`
- savoir consulter et administrer la documentation de /usr/share/doc
- savoir où trouver de la documentation sur Internet
- savoir comment informer les utilisateurs

support de cours additionnel : cours **Self-help**

Ce chapitre traite de la documentation : comment rechercher une information à l'aide des pages man, de la documentation système et où la trouver sur Internet ainsi que comment informer les utilisateurs de travaux d'administration.

Utiliser et gérer la documentation locale système – 4.1

Résumé des concepts importants

- le concept des manpages
- la variable `MANPATH`
- les commandes `man`, `apropos` et `whatis`
- les documentations sous `/usr/share/doc` des programmes installés

12

Exercices

1. dans quel section des manuels systèmes se trouvent des informations sur :
   (a) les commandes pour les utilisateurs
   (b) les commandes pour l'administration
   (c) la documentation du périphérique `/dev/ram`
   (d) la documentation du fichier de configuration `/etc/fstab`
   (e) la documentation du code ASCII
   (f) la documentation des jeux
   (indication : `man man` ou `man 7 man`)
2. comment trouver toutes les manpages ayant un rapport avec le concept de réseau ?
3. documentez-vous sur le concept de **crontab** : comment s'informer sur le format de fichier ? sur la commande ?
4. comment obtenir un résumé court de ce que fait une commande ?
5. que doit-il se trouver dans le répertoire `/usr/share/doc` ? (indication : `man hier`)
6. supposez que vous avez installé un fichier de manuel (p.ex. `tar.1.gz` dans votre répertoire `~/man/man1/`). Donnez deux méthodes différentes pour y accéder. Indication : copiez un manuel quelconque de `/usr/share/man` et testez.
7. comment créer un fichier PostScript (p.ex. pour l'imprimer) avec la commande `man` ? (indication : option `-T`)
8. quelle différence y-a-t-il entre les pages man et les pages Info ? (indication : comparez `man screen` et `info screen` ou toute autre documentation accessible dans `/usr/share/info`)

Trouver de la documentation sur Internet – 4.2

Résumé des concepts importants

- Linux Documentation Project (LDP) http://www.tldp.org/
- serveurs WWW de distributeurs et de tiers
- newsgroups (forums USENET) et leurs archives
- mailing-list

13

Exercices

1. installez un client news et connectez-vous à un serveur de news[1], déterminez la liste des forums Linux en anglais et en français.
2. cherchez dans les archives Google de news
3. trouvez quelques mailing-lists sur divers sujets (p.ex. liés à une distribution, à une application) et cherchez dans ces listes
4. comment écrire une man-page (cherchez un HOWTO chez LDP)

[1] alternative : accédez à Google groups

Notifier les utilisateurs – 4.3

Résumé des concepts importants

- `/etc/issue`, `/etc/issue.net`
- `/etc/motd`

14

Exercices

1. que contiennent respectivement les fichiers `/etc/issue` et `/etc/issue.net` et à quels moments sont-ils affichés ?
2. même question pour `/etc/motd`
3. documentez-vous sur la commande `wall`
4. documentez-vous sur le fichier `/etc/nologin`

5. Scripts et programmation shell

Contenu du chapitre

- paramétrisation avancée de l'environnement
- développement de scripts **bash** plus avancés

15 *Buts du chapitre*

- savoir configurer l'environnement du shell au login ou à un nouveau sous-shell
- savoir développer des scripts shell plus avancés (fonctions, boucles, tests, etc)
- connaître quelques commandes internes et externes supplémentaires

support de cours additionnel : cours **UNIX**, **Shell**, **LPI-101** et **Sécurité**

Ce chapitre traite de la programmation shell dans le but d'écrire des scripts systèmes (p.ex. configuration de variables, écritures de petites fonctions, tests, boucles, etc).

Personnaliser et utiliser l'environnement – 5.1

Résumé des concepts importants

- savoir configurer des variables (p.ex. `PATH`) au login et/ou au démarrage d'un sous-shell
- savoir écrire des fonctions shell
- fichiers : `~/.bash_profile`, `~/.bash_login`, `~/.profile`, `~/.bashrc`, `~/.bash_logout`, `~/.inputrc`
- mots réservé shell : `function`
- commandes internes : `export`, `set`, `unset`
- commandes externes : `env`, `seq`

16

Exercices

1. (rappel) déterminez (p.ex. en mettant des `echo`[1] ou le man de bash) les conditions et l'ordre d'exécution des scripts `~/.bash_profile`, `~/.bash_login`, `~/.profile`, `~/.bashrc`, `~/.bash_logout`
2. à quoi sert le fichier `~/.inputrc` ? et le fichier `/etc/inputrc` ?
3. déposez un script dans votre `~/scripts` (à créer éventuellement) et modifiez l'environnement de manière à ce que l'exécution fonctionne sans spécifier le chemin.
4. faites comme précédemment, mais assurez-vous que la modification est active à chaque lancement de shell interactif.
5. (rappel) indiquez la différence entre `var=a` et `export var=a`. Que se passe-t-il si j'écris `var=a; export var`. Comment supprimer une variable ?
6. la commande `set` ne sert pas à affecter des variables. Documentez-vous sur celle-ci et testez les options `-e` puis `-v` dans un script. Comment activer ces options lors de l'exécution d'un script, sans le modifier ?
7. écrivez un script shell qui fait appel à une fonction qui compresse un fichier passé en paramètre, s'il existe et s'il ne l'est pas déjà (indications : il finira par `.gz` s'il est compressé avec `gzip` ; utilisez `case` c'est le plus simple).

[1] ou en lisant les commentaires déposés par Debian au début de ces fichiers

Adapter ou écrire des scripts – 5.2

Résumé des concepts importants

- boucles (`for`, `while`), tests (`[]`, `test`[a]), substitutions, résultat
- activation d'un script (she-bang `#!`, `chmod`)
- envoi conditionnel d'un mail à root

17

Lectures supplémentaires

- http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html

[a] `man test`

Exercices

1. compléter la fonction du sous-chapitre précédent pour qu'elle retourne un résultat (0 == ok, 1 == déjà compressé, 2 == erreur) et le traiter (indication : `return`)
2. complétez la fonction originale du sous-chapitre précédent pour supporter plusieurs nom de fichiers (indication : boucle `for`, il y a un raccourci, consultez le `help`)
3. que sont les deux pré-requis pour que l'on puisse directement exécuter un script shell ?
4. supposons que `~schaefer/.bashrc` ait été effacé et que le plus simple est de le copier d'un autre utilisateur `testuser`. Indiquez *toutes* les commandes nécessaires.
5. créez à l'aide de la commande `touch` un fichier dont le nom est `fichier-DATE`, où `DATE` est la date en format ISO-8601
6. à l'aide de la commande `mail`[1] envoyez un mail à root uniquement si le fichier `/tmp/flag` existe, est un fichier et est vide
7. testez si la variable `DISPLAY` est vide, et si oui configurez-la au nom de la machine suivi de `:0`
8. où mettriez-vous des scripts à usage général ?
9. faire une boucle for, puis une boucle while, qui compte de 10 à 1
10. consultez la manpage de la commande `fsck` et traitez quelques cas possibles de résultats de la commande, par exemple en vous aidant de `help case`
11. est-ce que la permission SUID fonctionne avec les scripts ?

[1] ou directement `/usr/sbin/sendmail`

6. Administration

Contenu du chapitre

- gestion des comptes (utilisateurs et groupes) et préconfiguration de l'environnement utilisateurs
- configuration, surveillance, réaction et archivage des journaux
- lancement planifié de travaux
- planification et mise en oeuvre des sauvegardes

18

Buts du chapitre

- savoir créer, supprimer, limiter, suspendre, modifier et préconfigurer des comptes
- savoir configurer `syslogd` et rechercher l'information utile
- savoir utiliser `at` et `cron` (ou `anacron`)
- savoir définir et configurer une sauvegarde automatisée
- savoir configurer le temps système et un daemon NTP

support de cours additionnel : cours **Administration** et **SIB**

Ce chapitre traite de l'administration système générale : gestion des comptes et de leur préconfiguration, des journaux, des travaux automatiques, des sauvegardes et du temps.

Gestion des comptes – 6.1

Résumé des concepts importants

- base de données : `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`
- ajouter, modifier et supprimer les utilisateurs et groupes : `useradd`, `userdel`, `usermod`, `groupadd`, `groupdel`, `groupmod`
- `passwd` et `gpasswd`
- `chage`

19

Exercices

1. (rappel) investiguez les commandes `{user,group}{add,del,mod}` listées dans le sous-chapitre
2. comparez le fichier `/etc/passwd` à `/etc/shadow`, notamment ses permissions.
3. expirez un compte et loguez-vous. A quoi cela peut-t-il servir ?
4. mettez un mot de passe à un groupe et utilisez `newgrp` pour y accéder. Y-a-t-il une méthode plus simple pour donner le groupe à un utilisateur ?
5. bloquez un compte, testez un login et débloquez-le, retestez (n'hésitez pas à consulter `/var/log/auth.log` pour vous convaincre)

Préconfiguration de l'environnement – 6.2

Résumé des concepts importants

- maintenance du répertoire de préconfiguration `/etc/skel`
- configuration de `/etc/profile`[a]
- commandes : `env`, `export`, `set`, `unset`

20

[a] ou de `/etc/environment`

Exercices

1. créez un compte `demo`, loguez-vous avec, préconfigurez des paramètres (p.ex. aliases de bash, configuration de GNOME ou KDE, icônes sur le desktop, etc). Copiez une partie des configurations dans `/etc/skel`. Créez un compte avec préconfiguration et vérifiez que les paramètres configurés ont été installés. Comment faire pour avoir des classes de configuration ?
2. ajoutez des aliases globaux dans `/etc/profile` ou des préconfigurations de variables (`LD_LIBRARY_PATH` ou `PATH` p.ex.) et testez-les

Configuration et gestion des journaux – 6.3

Résumé des concepts importants

- configurer syslogd (`/etc/syslog.conf`, type et niveau)
- chercher dans les logs : `grep qqch /var/log/*`
- archiver les logs : `logrotate`
- suivre les changements en temps réel : `tail -f`

21

Format de /etc/syslog.conf Chaque ligne (qui peut être prolongée via un backslash) contient deux parties : la partie gauche est formée des conditions de sélection (quel sous-système (*facility*), quel niveau (*priority*), la partie droite de l'action : la destination des logs (fichier, serveur distant UDP, console, pipe, programme, etc).

Les conditions de sélection sont séparées par des points-virgules, et comportent une partie facility et une partie priority, séparée par un point. La partie facility peut être multiple pour une seul priority, on sépare alors par des virgules.

Les facilities sont les suivantes : auth, authpriv, cron, daemon, ftp, kern, lpr, mail, mark et news. Les priorities (triées par ordre d'importance croissante) sont : debug, info, notice, warning, err, crit, alert, emerg et panic.

La partie priority dénote soit un niveau et tous les niveaux supérieurs, ou alors seulement un niveau (en présence d'un symbole =).

L'étoile remplace toute priority ou toute facility ; le mot-clé `none` spécifie aucune ; le point d'exclamation inverse la condition (de nouveau, s'applique pour une priority (=), ou sinon toutes depuis ce niveau).

Si la partie droite est préfixée par un −, le journal ne sera pas synchronisé sur disque à chaque écriture (on peut donc perdre des entrées récentes en cas de crash brutal).

Exercices

1. modifier la configuration de syslogd pour envoyer les logs sur un serveur centralisé
2. afficher tous les logs d'authentification de niveau exactement debug sur la console 8 (indication : utiliser la commande `logger` pour simuler des logs)
3. à quoi sert le caractère − dans la partie droite (sortie) dans la configuration de syslogd ?
4. expliquez le fonctionnement de la commande `logrotate` et de ses fichiers de configuration
5. expliquez la répartition des logs entre `/var/log/auth.log` et `/var/log/syslog`

Travaux exécutés automatiquement – 6.4

Résumé des concepts importants

- cron : `/etc/crontab`[a] `/etc/cron.allow`, `/etc/cron.deny`, `/var/spool/cron/*`
- anacron : `/etc/anacrontab`
- atd : `/etc/at.allow`, `/etc/at.deny`
- commandes : `at`, `atq`, `atrm`, `crontab`

22

[a] voir aussi les crontabs utilisateur éditées via `crontab -e`, y compris pour root, les crontabs séparés sous `/etc/cron.d/`, les scripts à lancer régulièrement (journellement : `/etc/cron.daily/`, une fois par semaine : `/etc/cron.weekly`, une fois par mois : `/etc/cron.monthly`)

Exercices

1. quelle est la différence entre `atd` et `cron` ?
2. à quoi sert `anacron`, et sur quel type de machines ?
3. expliquez le format de `/etc/anacrontab`
4. écrivez un petit script[1] qui envoie un mail à root et soumettez le pour dans 5 minutes à `at`. Vérifiez qu'il est dans la queue des jobs, supprimez-le, vérifiez et re-soumettez-le. Vérifiez qu'il s'est bien exécuté.
5. donnez tous les endroits où peuvent être configurées des crontabs sous root
6. comparez le format de `/etc/crontab` à celui gérée par la commande `crontab`
7. interdisez l'accès à `at` à votre utilisateur.
8. quel est le comportement si ni `/etc/cron.allow` existe mais est vide ?

[1] obligatoirement un script compatible `/bin/sh` sinon utiliser un *wrapper*

Maintenir une sauvegarde fonctionnelle – 6.5

Résumé des concepts importants

- sauvegarde/restaurations de périphériques bruts (*raw devices*)
- sauvegardes complètes et partielles
- vérification de sauvegardes
- restaurations complètes ou partielles
- commandes : `cpio` `dd`, `dump`, `restore`, `tar`

23

Les 2 types de programme de sauvegarde On distingue les programmes de sauvegardes comme `dump` qui accèdent le système de fichiers directement à celles comme `tar` et `cpio` qui sauvegardent à partir de n'importe quel répertoire, avec ou sans les systèmes de fichiers montés à l'intérieur de l'arborescence concernée. Ces derniers ont également l'avantage d'être plus portables et fiables et sont donc recommandés. Ce sont ceux qui sont en général utilisés par des logiciels intégrés de sauvegarde.

Les niveaux de sauvegarde

0 sauvegarde complète

1 sauvegarde différentielle depuis le niveau 0 concerné (uniquement ce qui a changé)

N sauvegarde différentielle depuis le niveau N - 1 concerné (uniquement ce qui a changé)

En conséquence, pour une restauration intégrale, il faut toujours restaurer le dernier niveau 0, puis tous les niveaux supérieurs associés, par ordre croissant.

Exercices

1. à l'aide de la commande `dd`, sauvegarder une petite partition dans un fichier
2. à quoi sert l'avant-dernier champ (le 5^{e}) de `/etc/fstab` ?
3. comment pourriez-vous, à l'aide de la commande `chattr` et des options, spécifier qu'un répertoire ou fichier doit être ignoré par `dump` ?
4. comment faire une sauvegarde complète du système de fichiers `/home` sur le lecteur DDS SCSI avec la commande `dump` ? Comment restaurer ?
5. quel est l'avantage de l'algorithme Tour de Hanoi implémenté par dump pour les niveaux supérieurs à 0 ?
6. donnez des exemples de sauvegardes et restaurations avec tar et cpio.
7. est-ce que tar peut faire des sauvegardes incrémentales ? et cpio ?
8. comment vérifier une sauvegarde avec `restore` ? et avec `tar` ?

Gérer le temps système – 6.6

Résumé des concepts importants

- configuration du temps système : `date`
- configuration des zones de temps : `/usr/share/zoneinfo`, `/etc/timezone`, `/etc/localtime`
- configuration du BIOS (CMOS) : `hwclock`
- configuration NTP (y compris la dérive de temps (*drift*)) : `ntpd`, `ntpdate`, `/etc/ntp.conf`, `/etc/ntp.drift`[a]

[a] /var/lib/ntp/ntp.drift sur les distributions compatibles FHS, comme Debian.

24

Horloge système et CMOS GNU/Linux charge une fois au démarrage l'horloge maintenue par le BIOS avec la commande `hwclock -s` (en général avec l'option `-u` pour UTC/GMT).

Ensuite, le temps est maintenu indépendamment et peut être changé via la commande `date`. Si l'on désire reporter les changements dans la CMOS, il faut alors utiliser `hwclock -w` (également en général avec `-u`). On peut aussi utiliser directement `hwclock --set --date=newdate`, ou voir l'heure de la CMOS avec `hwclock -r`.

Changement d'horloge UNIX a besoin d'un temps monotone croissant : c'est pour cela qu'en général on ne changera pas l'heure, sinon au démarrage (via `hwclock` depuis la CMOS, ou via `ntpdate` depuis un serveur de référence).

Une fois que le système est démarré, un daemon peut être utilisé pour assurer une horloge uniforme dans un réseau, voire correcte. Ce daemon va ajuster la durée des secondes pour adapter (plus rapide, ou plus lent) l'horloge de manière monotone croissante. Ce daemon se synchronisera sur d'autres serveurs, voire sur une horloge précise locale. Un système de strates (niveaux) indique la qualité relative de chaque source :

```
schaefer@shakotay:~$ ntptrace localhost
localhost: stratum 3, offset -0.007832, synch distance 0.082684
193.39.78.2: stratum 2, offset 0.000000, synch distance 0.026250
tik.cesnet.cz: stratum 1, offset 0.000000, synch distance 0.000010,
   refid 'GPS'
```

Exemple de configuration simplifiée :

```
server pool.ntp.org

# référence locale en cas de panne réseau
server 127.127.1.0
fudge 127.127.1.0 stratum 13

driftfile /etc/ntp.drift # emplacement en violation FHS
```

Le fichier `ntp.drift` est modifié par le serveur NTP en fonction des mesures correctives effectuées dans le passé. Il permet de précorriger et de moins dériver en cas de panne réseau.

Exercices

1. exécutez `TZ=GMT date` et expliquez
2. changez la date système en arrière via la commande `date`. Redémarrez. Observez et expliquez. Maintenant faites la même chose et utilisez ensuite la commande `hwclock`. Observez et expliquez.
3. que sont les fichiers `/etc/timezone` et `/etc/localtime` ? que contiennent-ils ? que feriez-vous pour changer la zone locale manuellement[1] ?
4. comment pourriez-vous modifier le fichier `/etc/default/rcS` pour collaborer efficacement avec la gestion d'heure d'été désastreuse[2] de Microsoft Windows ?
5. installez un `ntpdate` au démarrage si ce n'est pas déjà fait.
6. installez un serveur NTP et testez-le (p.ex. avec l'outil `ntptrace`)
7. trouvez les scripts de démarrage et arrêt qui s'occupent de l'horloge.

[1] Debian contient un outil de plus haut niveau appelé `tzselect`

[2] ils ne changent pas la zone de temps pour DST, ils changent la CMOS ... donc maintiennent toujours la zone locale et non GMT dans le CMOS – UNIX par défaut travaille en GMT et présente aux utilisateurs une vision *cohérente* – et toujours correcte – en fonction de la configuration système et de la variable TZ

7. Bases du réseau

Contenu du chapitre

- concepts de base du réseau IP
- protocoles
- configuration, détermination et correction de problèmes
- connexion via PPP

25

Buts du chapitre

- connaître les concepts d'adresses, routage, sous-réseaux, plages spéciales
- savoir configurer ces paramètres et déterminer les problèmes
- savoir configurer une connexion cliente via PPP

support de cours additionnel : cours **Réseau**

Ce chapitre traite des concepts de base du réseau sous TCP/IP, de la configuration, de la détermination et de la correction des problèmes usuels ainsi que de la connexion à un fournisseur d'accès Internet ou un réseau interne d'entreprise via le protocole PPP.

Rappels sur TCP/IP – 7.1

Résumé des concepts importants

- concepts : adresses, sous-réseau, masques réseau (netmask), diffusion (broadcast), routage, classes, CIDR[a], plages réservées ou spéciales, route par défaut
- protocoles : IP, ICMP, TCP, UDP
- services (ports) : 20 (FTP/DATA), 21 (FTP), 23 (TELNET), 22 (SSH), 25 (SMTP), 53 (DNS), 80 (HTTP), 110 (POP3), 119 (NNTP), 139 (SMB), 143 (IMAP2), 161 (SNMP)
- différences entre IPv4 et IPv6
- `/etc/services`
- `ftp`, `telnet`, `host`, `ping`, `dig`, `traceroute`, `whois`

26

Lectures supplémentaires

- http://www.bieringer.de/linux/IPv6/

[a] rappel : adresse-sous-réseau/nombre-de-bits-libres, p.ex. 192.168.1.0/24)

Différences entre IPv4 et IPv6

- adressage à 128 bits (plutôt que 32 bits pour IPv4), p.ex. `1fff:0000:0a88:85a3:0000:0000:ac1f` (ou formats simplifiés)
- plages allouables généreusement
- protocole ARP non nécessaire (les bits inférieurs de l'adresse IPv6 peuvent être configurés comme l'adresse MAC)
- NAT/PAT en théorie plus nécessaire
- champs DNS `AAAA`
- qualité de service
- mobilité
- authentification et chiffrement (optionnels)
- IPv6 sur IPv4 : possible (tunnel)

(la plupart des avantages d'IPv6 sont disponibles comme protocoles spécifiques d'IPv4 comme p.ex. IPsec ; sauf la taille d'adressage).

Le support GNU/Linux est fonctionnel (y compris firewall). A savoir : si IPv6 est activé, les règles du firewall IPv4 ne s'appliquent plus forcément, et la commande `netstat --inet6` doit être utilisée pour voir tous les services (qui pourraient être invisibles via `--inet`).

Le problème de la migration d'IPv4 à IPv6 est principalement politique.

Exercices

1. quelle est la taille en bits d'une adresse IPv4 ? Comment la représente-t-on usuellement ? Donnez un exemple.
2. renseignez-vous à l'aide de la commande `whois` sur le sous-réseau `193.72.186.0` et déduisez le netmask.
3. soit l'adresse `80.83.54.61` et le masque de sous-réseau 255.255.255.224. Indiquez la *classe* de cette adresse. Ensuite, déterminez le sous-réseau effectif : donnez l'adresse de sous-réseau et l'adresse de broadcast. Donnez cette information également sous forme CIDR
4. soit la machine `192.168.1.42` qui se trouve dans un sous-réseau CIDR `sous-réseau/26`. Indiquez l'adresse de sous-réseau, le netmask et l'adresse broadcast correspondante.
5. donnez des exemples de plages d'adresses réservées pour les réseaux privés. Que doit faire un routeur connecté à Internet lorsqu'il voit passer un datagramme IP avec une adresse source dans ces plages ?
6. à quoi sert la route par défaut ?
7. configurez une deuxième adresse IP sur votre poste (`ifconfig eth0:0 192.168.100.A`, où `A` est la dernière partie de l'adresse IP de votre machine : de manière à éviter des collisions avec d'autres). Que doit faire votre voisin pour pouvoir `ping`er cette adresse ? (indication : on suppose que les deux ordinateurs sont dans le même sous-réseau, soit non séparés par un routeur).
8. déterminez le chemin jusqu'à une autre adresse IP (avec `traceroute` ou `mtr`)
9. déterminez l'adresse IPv6 correspondant au nom `domreg.nic.ch`
10. avec quelle commande pouvez-vous lister les informations de la zone DNS `alphanet.ch` ?
11. quelle commande vous permet de faire des transferts de fichiers via le protocole situé sur le port 21/tcp ?
12. dessinez l'architecture des 4 protocoles des couches 3 et 4 de TCP/IP
13. déterminez via `/etc/services` le numéro de port et le protocole (TCP ou UDP) de quelques protocoles courants
14. déconfigurez le support IPv6 dans votre système (indication : soit une diversion sur le module kernel, soit, plus simple et plus portable, un alias dans `/etc/modprobe.d`)

Configuration et détermination de problèmes – 7.2

Résumé des concepts importants

- consulter, modifier et vérifier les états et configurations des interfaces réseau : `ifconfig`, `ifup`, `ifdown`, `route`
- configuration d'un client DHCP : `dhcpcd`, `dhclient`, `pump`
- configuration de base réseau : `/etc/hostname` (ou `/etc/HOSTNAME`), `/etc/hosts`, `/etc/networks`, `/etc/host.conf`, `/etc/nsswitch.conf`, `/etc/resolv.conf`
- debugging de problèmes réseau : `host`, `hostname`, `domainname`, `dnsdomainname`, `netstat`, `ping`, `traceroute`, `tcpdump`
- `/etc/init.d/networking`

27

Exercices

1. si vous êtes en DHCP, tuez le daemon DHCP, puis lancez-le manuellement
2. consultez les routes configurées sur votre machine (`route` ou `netstat -rn`)
3. que faut-il faire si `ping www.alphanet.ch` retourne l'erreur `unknown host` alors que `ping 80.83.54.2` fonctionne ? et si cette dernière commande retourne `network unreachable`
4. quelle est la différence entre `ifconfig eth0 down` et `ifdown eth0` ?
5. comment ajouter une route au réseau `192.168.100.0/24` via le routeur `192.168.42.35` sur `eth0` ?
6. quelle est la différence entre changer le nom de la machine avec la commande `hostname` et via `/etc/hostname` ?
7. quel est l'avantage de configurer un nom complètement qualifié (FQDN : machine.domaine.ch) pour l'adresse IP principale de la machine dans /etc/hosts ? comment y configurer des aliases ? comment le faire sur plusieurs machines d'un réseau ?
8. quelle est la différence entre `dnsdomainname` et `domainname` ?
9. listez tous les ports utilisés de la machine avec leur numéro de processus (sans résoudre les noms)
10. comment le système sait-il que `0.0.0.0` doit s'appeler `default` (sans l'option `-n` de `route` ou `netstat`) ? (alternative : ajoutez votre sous-réseau à ce fichier et testez)
11. à quoi servent les fichiers `/etc/host.conf` et `/etc/nsswitch.conf` ?
12. quels sont les scripts lancés au démarrage pour la configuration réseau ?
13. donnez deux idées pour déterminer quel serveur DNS est utilisé alors qu'il y a un doute sur la configuration.

Configurer un client PPP – 7.3

Résumé des concepts importants

- modem, ISDN, ADSL (ou MS-PPTP, ou PPP-sur-SSH, etc)
- concept de *chat script*
- configuration : `/etc/ppp/options.*`, `/etc/ppp/peers/*`, `/etc/wvdial.conf`, `/etc/ppp/ip-up`, `/etc/ppp/ip-down`,
- commandes (connexion, déconnexion) : `wvdial`, `pppd`
- options particulières : reconnexion automatique (p.ex.)

28

Lectures supplémentaires

- http://www.debianhelp.co.uk/ppp.htm (wvdial sur Debian)
- http://www.debian.org/releases/stable/i386/apds05.html.fr (PPPoE (ADSL ancien) sur Debian)
- http://tldp.org/HOWTO/PPP-HOWTO/ (obsolète)

Principes de base Le protocole PPP (Point-to-Point Protocol) est un protocole standard d'échange de paquets entre deux systèmes sur une liaison quelconque. Une application est la connexion à un fournisseur d'accès à Internet (FAI) ou au réseau interne de l'entreprise.

L'implémentation UNIX est le logiciel `pppd` (qui agit en tant que serveur ou client).

Lorsqu'un modem est utilisé, il y a en général un dialogue (les commandes AT de composition) ainsi qu'éventuellement un login et un mot de passe sur le concentrateur de terminal du fournisseur. Ce dialogue est fait par l'utilitaire `wvdial` (ou `chat`). Cet outil peut se configurer manuellement dans `/etc/wvdial.conf` ou semi-automatiquement via l'utilitaire `wvdialconf`.

Exemple de `chat` classique :

```
chat -v '' ATZ OK ATDT032841401 CONNECT '' ogin: pppmarc word: demo
```

Lorsqu'il s'agit d'ADSL, il existe, s'il s'agit d'un modem ADSL (plutôt qu'un routeur), d'un protocole particulier appelé PPP-over-Ethernet (évt. PPP-over-ATM) qui consiste à échanger des trames PPP qui sont converties d'Ethernet à ADSL/ATM/AAL5 par le modem.

Dans tous les cas, une phase d'authentification est en général prévue (le client s'authentifie au serveur dans le cas général). La configuration de cette authentification se fait dans `/etc/ppp/peers/*` ainsi que dans `/etc/ppp/chap-secrets` ou `/etc/ppp/pap-secrets`.

CHAP est un protocole challenge-response : le mot de passe doit être en clair dans tous les cas. PAP est un protocole en clair, le mot de passe peut être hashé (sur le serveur, en clair sur le client).

Des options spéciales sont configurées dans `/etc/ppp/options.*`. Des scripts peuvent être lancés lors de la connexion et/ou de la déconnexion (p.ex. pour lancer `fetchmail` ou d'autres services) via `/etc/ppp/ip-up` et `/etc/ppp/ip-down`.

Configuration de wvdial Cette configuration est la plus simple. Parfois elle est insuffisante. On se référera alors à la section suivante.

On lancera la commande `wvdialconf`. Cette dernière créera les configurations `/etc/wvdial.conf` et éventuellement des secrets CHAP ou PAP dans `/etc/ppp`.

Le fichier `/etc/wvdial.conf` pourrait ressembler à :

```
[Dialer Defaults]
     Modem = /dev/ttyS1
     Baud = 115200
     Init1 = ATZ
     Init2 = ATQ0 V1 E1 S0=0 S11=55 +FCLASS=0
     Phone = 0840555555
     Username = sunrise
     Password = freesurf
```

Configuration de pppd On peut partir p.ex. de `/etc/ppp/peers/provider`. On y configurera le nom d'utilisateur (p.ex. `sunrise`), et le mot de passe dans `/etc/ppp/chap-secrets`. Ou, plus simplement avec wvdial comme montré précédemment.

Le démarrage se fait alors via `pon provider`, l'arrêt par `poff provider`. On peut utiliser aussi la forme plus simple `pppd call provider`.

Options avancées L'option `Auto Reconnect` de `wvdial.conf` permet de réappeler indéfiniment en cas de coupure de la connexion. L'option `persist` de `pppd`, quant à elle, permet de redémarrer la connexion.

8. Services réseau

Contenu du chapitre

- configuration de daemons réseau via le super-serveur inetd/xinetd
- configuration de base et gestion de divers MTA (*Mail Transfer Agent*)
- configuration de base et gestion d'Apache, NFS, Samba, DNS et SSH

29

Buts du chapitre

- savoir configurer et gérer quelques services réseaux classiques (indépendants (*standalone*) ou via inetd/xinetd)
- connaître les bases de la configuration, mise en place et gestion de services réseau classiques comme un serveur de courrier électronique, le serveur WWW Apache, NFS, Samba, le DNS et SSH

support de cours additionnel : cours **Réseau**, **Administration**, **SMTP**, **Samba**

Ce chapitre traite des services de base installables en réseau TCP/IP, notamment du super-serveur inetd (ou xinetd), d'un serveur de courrier électronique (MTA), de la configuration de base de la gestion d'Apache, du partage de fichiers classique UNIX NFS ou compatible Microsoft Samba, du DNS ainsi que de la connexion sécurisée SSH.

Super-serveur inetd/xinetd – 8.1

Résumé des concepts importants

- définition des services : `/etc/services`
- configuration de services : `/etc/inetd.conf` (ou `/etc/xinetd.conf` et `/etc/xinetd.d/`)
- configuration des règles d'accès basées sur les adresses IP avec les `tcpwrappers` : `/etc/hosts.allow`, `/etc/hosts.deny`
- consultation des logs : `/var/log/xinetd.log`

Principes de base du super-serveur réseau L'idée du super-serveur réseau (que cela soit `inetd` ou `xinetd`) est de sortir les fonctionnalités d'écoute réseau et de gestion d'instances multiples des divers daemons.

Le super-serveur écoute sur les ports assignés et lance des instances de services (multiples ou uniques : en général multiples pour TCP et uniques pour UDP) qui ne doivent gérer que l'entrée et la sortie standard.

Des limites permettent d'assurer une charge convenable (p.ex. rapidité successive de lancement, etc). Le support optionnel **tcpwrappers** permet de configurer des règles simples de sécurité (basées sur les adresses IP seulement !) avant le lancement de services.

Exemple de l'ajout d'un service Soit le service suivant :

```
#! /bin/bash

cat <<EOF
Il est actuellement `date`.
Veuillez tapez RETURN pour quitter le service.
EOF

read ignore
```

stocké par exemple comme `/usr/local/bin/service-simple` et rendu exécutable (`chmod a+rx /usr/local/bin/service-simple`).

Ce service peut être ajouté au super-daemon `inetd` comme suit :

```
echo >> /etc/services 'mon-service 12345/tcp'
echo >> /etc/inetd.conf \
   'mon-service stream tcp nowait nobody \
       /usr/sbin/tcpd /usr/local/bin/service-simple'
```

Pour qu'`inetd` relise sa configuration, il faut lui envoyer le signal `HUP` (ou passer par les scripts `init.d`) :

```
killall -HUP inetd
```
ou
```
/etc/init.d/inetd reload  # ou similaire
```

Exercices

1. installez le service proposé et testez (avec `telnet localhost 12345`) qu'il fonctionne.
2. empêchez l'adresse IP de votre voisin (via `/etc/hosts.deny`) de se connecter à ce service (indication : `man tcpd`). Quel est le symptôme ?
3. installez un serveur telnet dépendant d'inetd (p.ex. `telnetd`) et regardez ce que le système préconfigure pour vous dans `/etc/inetd.conf`. Consultez le script de post-installation[1]. Quelle commande spécifique est utilisée ?
4. déinstallez l'inetd actuellement installé et installez `xinetd`. Ajoutez une redirection de port et testez-là. Par exemple `localhost:2222` vers `localhost:22` (indication : `man xinetd.conf`, `redirect`).
5. quelles facilités sont disponibles pour limiter la fréquence des connexions et d'autres paramètres ?
6. dans quels cas configure-t-on des services en mode *standalone* plutôt que via `inetd` ?
7. quelle est la différence entre `stream` et `dgram` ?

[1] indication : `/var/lib/dpkg/info/telnetd.postinst`

Configuration et gestion de base d'un MTA – 8.2

Résumé des concepts importants

- MUA, MTA, MDA
- configuration de base : `/etc/mail/*`[a], `/etc/aliases`, `newaliases`, `~/.forward`
- serveurs Postfix, exim, sendmail et qmail
- commandes de base : `mailq`, `sendmail`, `runq`
- anti-spam : concept de relais ouvert (*open relay*)

31

Lectures supplémentaires

- http://www.flounder.net/qmail/qmail-howto.html

[a]surtout pour sendmail : sendmail.cf, local_host_names (sendmail.cw), relay_domains, mail_access, mailertable, virtusertable, trusted_users

Concepts de base L'outil utilisé pour rédiger ou consulter ses mails est un **MUA** (*Mail User Agent* : par exemple `mail`, `mailx`, `mutt`, `mozilla-thunderbird`, `evolution`). Cet outil lit ses mails via POP, IMAP ou système de fichier local. Il envoie ses mails via SMTP (ou SMSP, récemment, RFC-2476), ou plus simplement via `/usr/sbin/sendmail` local (qui correspond toujours à une interface compatible au MDA/MTA local installé, au moins sur Debian).

Sur Internet, les mails sont échangés entre **MTA** (*Mail Transfer Agent* : par exemple Postfix, exim, sendmail, qmail). Le protocole utilisé lors de ces échanges est SMTP. Les enregistrement de type MX (*Mail Exchanger*) du DNS sont utilisés pour déterminer les serveurs gérant le mail pour un domaine donné (à défaut : les champs A).

Enfin, à un moment donné, un mail atteint sa destination finale. Un **MDA** va délivrer le mail dans la boîte-aux-lettres concernée (*Mail Delivery Agent*, p.ex. une partie du MTA, ou un logiciel spécifique comme `procmail` – très pratique pour le tri des messages dans des boîtes multiples par exemple – ou `deliver`).

La configuration et la gestion d'un système de mail Les aspects généraux sont :
- `/etc/aliases` contient les redirections locales. La commande `newaliases` crée l'index associé et est donc obligatoire à chaque changement.
- le fichier `.forward` dans un répertoire d'un utilisateur UNIX permet de spécifier une ou plusieurs redirections.
- on peut en général obtenir la liste des mails dans la queue avec `mailq`
- l'envoi de mail passe en général par une interface de compatibilité (`/usr/sbin/sendmail`)

Les aspects spécifiques sont :
– `runq` (exim) et `postqueue -f` servent à relancer la queue
– la configuration de chaque MTA/MDA dans son répertoire de `/etc`

Exemple de fichiers `~/.forward` complexe :

```
schaefer@shakotay:~$ cat .forward
/home/schaefer/Mail/mail.received, \
   "|/usr/bin/procmail /home/schaefer/.procmailrc-MANUAL", \
   auto-rt@alphanet.ch
```

Le spam Le but ici est simplement de rappeler quelques concepts de base de l'anti-spam :
– aucun serveur de mail de devrait être un relais ouvert : un MTA ne doit accepter que les mails dont la destination finale est locale (gérée par lui : domaines locaux, aliases, envoi éventuel à un serveur interne par SMTP, etc)
– seule exception : on peut faire relais ouvert pour ses utilisateurs (soit basé sur l'adresse IP si cette méthode est sûre, soit basée sur une authentification SASL par mot de passe)

Exemple d'un serveur ouvert : (en fait il est ouvert uniquement pour le réseau privé)

```
schaefer@reliant:~$ telnet 192.168.1.1 smtp
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
220 shakotay.alphanet.ch ESMTP Postfix (Debian/GNU)
HELO abcd
250 shakotay.alphanet.ch
MAIL FROM: <whatever@earth.org>
250 Ok
RCPT TO: <anyone@truc.org>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: test

bla
.
250 Ok: queued as 30C2B58B01B
QUIT
221 Bye
Connection closed by foreign host.
```

Même exemple depuis l'extérieur :

```
outside% telnet smtp.alphanet.ch smtp
Trying 80.83.54.2...
Connected to shakotay.alphanet.ch.
Escape character is '^]'.
```

```
220 shakotay.alphanet.ch ESMTP Postfix (Debian/GNU)
HELO abcd
250 shakotay.alphanet.ch
MAIL FROM: <whatever@earth.org>
250 Ok
RCPT TO: <anyone@truc.org>
554 <anyone@truc.org>: Relay access denied
```

On peut aussi utiliser le service http://www.abuse.net/relay.html pour tester.

Autres informations

exim http://www.exim.org/howto/relay.html

sendmail http://www.akadia.com/services/sendmail_relay.html

qmail http://www.palomine.net/qmail/relaying.html

Exercices

1. installez exim, testez que l'envoi de mail et la lecture fonctionnent (p.ex. utiliser `mail` ou `mailx` pour envoyer un mail, listez `/var/mail`, utilisez more ou `mail` pour lire vos mails)
2. documentez-vous sur la commande `sendmail`.
3. ajoutez-vous un `.forward` qui redirige à un autre utilisateur. Testez que cela fonctionne.
4. faites de même via `/etc/aliases`
5. modifiez le type de serveur comme étant capable d'envoyer à tout Internet, envoyez un mail à `schaefer@alphanet.ch`. Consultez les logs et la queue des mails. Que se passe-t-il ? Supprimez-le de la queue.
6. consultez les fichiers de configuration d'exim
7. empêchez votre voisin d'envoyer des mails via votre serveur (ou faites le contraire suivant la configuration par défaut).
8. même question pour Postfix et sendmail.
9. comment configurez-vous un smart-host (relais par défaut) avec sendmail ? testez et consultez les logs.

Configuration et gestion de base d'Apache 2 – 8.3

Résumé des concepts importants

- daemon `httpd`[a]
- configuration : `/etc/apache2/`, `httpd.conf`
- commandes : `apache2ctl` (ou `apachectl`)

32

[a]sous Debian, le nom du daemon est apache ou apache2, et la configuration est répartie dans divers fichiers et répertoires sous /etc/apache2.

Configuration d'Apache 2 de base Une configuration de base d'Apache 2 :

```
Listen 80

ServerRoot "/etc/apache2"

User www-data
Group www-data
Alias /icons "/usr/share/apache2/icons/"
<Directory "/usr/share/apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
ErrorDocument 404 /not_found.html

AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset UTF-8       .utf8

ServerAdmin webmaster@localhost
DocumentRoot /var/www/
<Directory />
```

```
        Options FollowSymLinks
        AllowOverride None
</Directory>
<Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Il faut savoir que les distributions facilient grandement le travail de configuration grâce à leur préconfiguration et à l'organisation des divers fichiers et répertoires. Les fichiers de configuration sont grandement commentés et des commandes supplémentaires (p.ex. `a2enmod`) permettent l'ajout facilité de modules, etc.

Exercices

1. quelles sont les opérations possibles à l'aide d'`apache2ctl` ? testez-les.
2. faites écouter votre serveur WWW sur un port supplémentaire et testez (p.ex. 8080, `http://localhost:8080/`)
3. ajoutez le module `userdir` si ce n'est pas encore fait et testez l'url `http://localhost/~user/` qui doit correspondre à `~user/public_html/`.
4. à quoi sert un fichier `.htaccess` ?
5. à quoi peut servir l'URL `http://localhost/server-status/` ?
6. consultez les logs Apache et expliquez les champs. Pouvez-vous modifier le format ?

Bases de NFS et Samba – 8.4

Résumé des concepts importants

- configurations : `/etc/exports`, `/etc/fstab`, `/etc/smb.conf` (ou `/etc/samba/smb.conf`),
- commandes : `mount`, `umount`
- autres commandes : `showmount`, `rpcinfo`, `smbclient`, `nmblookup`, swat
- LPI ne couvre que les concepts les plus simples

33

Lectures supplémentaires

- http://nfs.sourceforge.net/nfs-howto/
- http://tldp.org/HOWTO/SMB-HOWTO.html

Concepts de NFS NFS (*Network File System*) est un système de fichiers réseau supportant POSIX dans les grandes lignes. Il a été développé dans les années 80 par Sun Microsystems et est basé sur le système RPC (*Remote Procedure Call*).

Les différences principales avec Samba sont les suivantes :
- protocole plus simple
- support de la norme POSIX (y compris le verrouillage si configuré)
- montage normalement par le système (p.ex. un `/home` distant monté comme `/home`) pour tous les utilisateurs, p.ex. dans `/etc/fstab` ou via l'automounter.
- modèle de sécurité UNIX (UID, GIDs, éventuellement ACLs), vérifié sur le client
- protocole UDP (NFSv4 : supporte TCP également)
- pas de chiffrement ni authentification (sauf SecureRPC, sécurisation couche IP (p.ex. IPsec, VPN) ou tunnelling en TCP)
- protocole sans-état (sauf évt. verrouillage) : le serveur peut redémarrer sans affecter le client
- fiabilité : montage *soft* (erreur) ou *hard* (comportement réessai infini)
- charge serveur plus faible qu'avec Samba

De nombreuses améliorations ont été apportées à NFS. La version la plus récente est NFSv4. D'autres systèmes de fichiers POSIX réseau existent : AFS, codafs, etc, mais NFS est le plus répandu.

Services nécessaires pour NFS Au minimum, il faut que le service **portmap** soit accessible (lancé et non bloqué via les règles `/etc/hosts.{allow,deny}`) : ce daemon gère l'attribution dynamique des numéros de ports aux services RPC.

On peut tester que ce service fonctionne et quels sont les services RPC qui se sont enregistrés :

```
schaefer@asterix:~$ rpcinfo -p localhost
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   udp  32773  nlockmgr
    100021    3   udp  32773  nlockmgr
    100021    4   udp  32773  nlockmgr
    100021    1   tcp  32912  nlockmgr
    100021    3   tcp  32912  nlockmgr
    100021    4   tcp  32912  nlockmgr
    100005    1   udp    947  mountd
    100005    1   tcp    950  mountd
    100005    2   udp    947  mountd
    100005    2   tcp    950  mountd
    100005    3   udp    947  mountd
    100005    3   tcp    950  mountd
    100024    1   udp  32774  status
    100024    1   tcp  50018  status
```

De plus, un serveur NFS doit être installé et lancé (soit en mode kernel, soit en mode utilisateur – le mode kernel est préféré aujourd'hui).

Exportation de systèmes de fichiers en NFS De manière à pouvoir exporter des systèmes de fichiers, le daemon `rpc.mountd` doit être démarré.

On configurera les systèmes de fichiers à exporter dans `/etc/exports`. N'importe quel répertoire peut en fait être exporté. Ensuite, il faut annoncer au serveur `rpc.mountd`, via la commande `exportfs -a`, que `/etc/exports` a été modifié.

Par exemple :

```
/opt/ltsp/i386      192.168.0.0/255.255.255.0(ro,no_root_squash)
/data               192.168.0.0/255.255.255.0(rw)
/scratch            192.168.0.0/255.255.255.0(rw)
```

On peut tester que l'exportation est correcte ainsi :

```
schaefer@asterix:~$ /sbin/showmount -e localhost
Export list for localhost:
/data          192.168.0.0/255.255.255.0
/scratch       192.168.0.0/255.255.255.0
/opt/ltsp/i386 192.168.0.0/255.255.255.0
```

Les paramètres d'exportation définissent une liste d'accès (adresse IP), la permission maximale (lecture/écriture ou lecture seule) et d'autres paramètres, comme par exemple la suspension de la translation de l'accès root implicite en nobody ou des paramètres de performance.

Montage de systèmes de fichiers NFS On peut monter n'importe quel répertoire se situant à l'intérieur d'une exportation (ou la racine de celle-ci).

```
mount serveur:/chemin /mount/point/local # -t nfs implicite
```

Le démontage se fait p.ex. avec

```
umount /mount/point/local
```

On peut spécifier un montage directement dans `/etc/fstab`.

Les options de montage sont diverses : citons p.ex. l'option `soft`, qui permet de retourner une erreur en cas d'inaccessibilité du serveur plutôt que d'attendre indéfiniment.

Concepts de Samba Samba est une implémentation libre du protocole **Lan Manager SMB**, sous forme de deux daemons : `smbd` (gère les connexions, l'authentification et les transferts de données) et `nmbd` (serveur de nom spécifique à SMB). Les versions modernes de Samba implémentent en fait CIFS (*Common Internet File System*), en présentant diverses interfaces suivant la version du client. L'implémentation Samba ne supporte que TCP/IP (ce qui est également le cas recommandé aujourd'hui sous Microsoft).

Ce qui est important pour Samba :
- le serveur porte un ou plusieurs nom et répond en TCP/IP à une ou plusieurs adresses sur des ports standards (`smbd` : TCP 139, 445 ; `nmbd` : UDP 137, 138)
- un service de nom (travaillant par broadcast ou via un protocole spécial appelé WINS) permet de trouver l'adresse IP en connaissant son nom `\\NOM-SERVEUR`
- un ou plusieurs partages sont accessibles `\\NOM-SERVEUR\PARTAGE` (avec quelques cas spéciaux)
- un client s'identifie à la connexion en choisissant plusieurs modes possibles (pour Samba, en général cela signifie un utilisateur et un mot de passe)
  - mode share : authentification ancienne par partage
  - mode user : authentification usuelle (utilisateur/mot de passe)
  - mode domain : comme user, mais à un serveur distant
- les droits d'accès s'appliquent comme si l'utilisateur s'était connecté sous UNIX (UID, GID principal et secondaires), y compris les ACLs si désiré
- il est possible de forcer des droits d'accès plus ou moins restrictifs que les droits d'UNIX pour gérer des cas particuliers

Il n'y a pas vraiment de différence fondamentale entre un serveur de domaine ou un serveur de workgroup Samba. La différence vient surtout dans la façon dont on y intègre les clients.

Debugging Samba Trois outils sont très pratiques : `smbclient`, `nmblookup` et `smbstatus` :

```
schaefer@asterix:~$ smbclient -L localhost
Password:
Anonymous login successful
Domain=[ENTERPRISE] OS=[Unix] Server=[Samba 3.0.24]

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (asterix server (Samba 3.0.
        fax             Printer   Fax Printer
        scratch         Disk      Temporary Files
        data            Disk      Enterprise data
        brother         Printer

        Server               Comment
        ---------            -------
        CLIENT1
        CLIENT2

        Workgroup            Master
        ---------            -------
        ENTERPRISE           ASTERIX
        WORKGROUP            MAC000A95CECF4C

schaefer@asterix:~$ nmblookup asterix
querying asterix on 192.168.0.255
192.168.0.6 asterix<00>
schaefer@asterix:~$ nmblookup MAC000A95CECF4C
querying MAC000A95CECF4C on 192.168.0.255
192.168.0.247 MAC000A95CECF4C<00>

schaefer@asterix:~$ smbclient '\\asterix\scratch' -U schaefer
Password:
Domain=[ASTERIX] OS=[Unix] Server=[Samba 3.0.24]
smb: \> dir
  .                                   DH        0  Fri Jan 21 11:20:59 200
  ..                                  DH        0  Tue Nov 27 23:27:43 200
  tmp                                  D        0  Tue Sep 11 07:49:40 200
  lost+found                           D        0  Fri Jun  6 13:31:44 200

                50396 blocks of size 2097152. 18899 blocks available
smb: \> cd tmp
smb: \tmp\> put /etc/motd bla
putting file /etc/motd as \tmp\bla (18.1 kb/s) (average 18.1 kb/s)
smb: \tmp\> del bla
smb: \tmp\> quit
```

Exercices

1. vérifiez que vous avez les outils nécessaires pour NFS
2. exportez votre `/home` et vérifiez
3. montez un système de fichier exporté par votre voisin. Pouvez-vous effectuer des modifications à l'intérieur sous root ? Quel est le problème et comment le corriger ? Pouvez-vous vous faire passer pour un utilisateur valide ?
4. configurez ce montage dans `/etc/fstab` et redémarrez
5. installez Samba, utilisez `smbclient` pour déterminer les services disponibles
6. configurez votre mot de passe avec `smbpasswd -a UTILISATEUR`
7. en utilisant les exemples dans `/etc/samba/smb.conf`, ajoutez un nouveau partage (p.ex. `/tmp`) et testez la connexion (d'abord anonyme : `public=yes` puis non) (indication : un `/etc/init.d/samba reload` est nécessaire)
8. copiez `/etc/samba/smb.conf`, puis à l'aide de l'outil SWAT configurez Samba en mode graphique. Comparez enfin les modifications effectuées (p.ex. avec `diff`)
9. configurez votre `nmblookup` pour utiliser le daemon de votre voisin. Testez avec l'outil `nmblookup`. (indication : un `/etc/init.d/samba restart` est nécessaire.
10. comment fonctionne le partage des imprimantes ?
11. comment accéder à son répertoire personnel via Samba ?

Bases du DNS – 8.5

Résumé des concepts importants

- configuration : `/etc/hosts`, `/etc/resolv.conf`, `/etc/nsswitch.conf`, `/etc/named.conf` [a]
- daemon : `named`
- outils : `host`, `dig` et `whois` (voire `nslookup`)
- délégation, enregistrement, cache et *forwarder*

[a] En général `/etc/bind/named.conf`

34

Exercices

1. expliquez ce qu'il se passe quand j'essaie de résoudre `www.alphanet.ch` (en supposant qu'il n'y a pas de cache ni de forwarder) dans le cas général
2. copiez la configuration actuelle du client de résolution (le *resolver*)
3. installez un serveur `bind9` et assurez-vous qu'il est bien utilisé par votre système
4. une optimisation courante est de configurer un *forwarder*[1] sur un serveur de nom : un autre serveur qui sera utilisé (plutôt que d'effectuer la résolution via l'arborescence) pour répondre aux questions dont les réponses ne sont pas encore cachées localement. Configurez de préférence celui que vous aviez sauvegardé et testez.
5. créez une zone `test.ch` en utilisant les exemples dans `/etc/bind` et testez
6. créez une zone de résolution inverse pour le sous-réseau 192.168.100.0/24 et testez
7. expliquez ce que vous devriez faire pour enregistrer, activer et utiliser un nouveau nom de domaine sous `ch.`
8. expliquez la différence entre `allow-query` et `allow-recursion`.

[1] en pratique : souvent les DNS du FAI, s'ils sont fiables !

SSH – 8.6

Résumé des concepts importants

- configuration : `/etc/hosts.allow`, `/etc/hosts.deny`, `/etc/ssh/sshd_config`, `/etc/ssh_known_hosts` (ou dans /etc/ssh/), `/etc/sshrc` (ou dans /etc/ssh/
- fonctions spéciales : `/etc/nologin`
- daemon : `sshd`
- commandes : `ssh-keygen`
- concept : clé publique/privée

35

Exercices

1. installez `sshd` si ce n'est pas déjà fait (indication : `telnet localhost 22`)
2. effectuez la manipulation suivante pour pouvoir vous connecter de votre utilisateur à un autre utilisateur sans mot de passe :
   (a) sous votre utilisateur, générez un couple de clé privée et publique : `ssh-keygen -t dsa` (sans spécifier de passphrase)
   (b) sauvez ces clés normalement (usuellement `~/.ssh/id_dsa` pour la clé privée (garder secrète ! !) et `.pub` pour la clé publique (qui peut être visionnée (texte encodé) ou copiée ailleurs) – consultez les permissions respectives de ces fichiers
   (c) transférez, p.ex. avec `scp` la clé publique dans `~autre-utilisateur/.ssh/authorized_keys`
   (d) vérifiez que vous pouvez vous connecter sans mot de passe

   (indication : `ssh -v` permet d'avoir plus d'informations)
3. ajoutez une passphrase à votre clé privée et tentez à nouveau de vous connecter : la passphrase doit vous être demandée
4. vérifiez qu'un `ssh-agent` tourne sous votre utilisateur
5. ajoutez la passphrase à l'agent via `ssh-add` et vérifiez que vous pouvez vous connecter sans ré-entrer la passphrase

9. Sécurité

Contenu du chapitre

– tâches d'administration concernant la sécurité (revue, politiques, application, vérification (journaux p.ex.))

Buts du chapitre

36

– savoir déterminer une politique de sécurité, vérifier la configuration du système, appliquer des politiques restrictives, auditer les résultats

support de cours additionnel : cours **Sécurité**, **Administration**

Ce chapitre traite de quelques tâches d'administration de sécurisation de base.

Tâches administratives de sécurité – 9.1

Résumé des concepts importants

- **tcpwrappers** : `/etc/hosts.{deny,allow}`
- utiliser `find` pour trouver des fichiers SUID/SGID
- gérer les mots de passe (mettre, changer, modifier les paramètres)
- savoir quoi mettre à jour (selon la distribution, CERT, BUGTRAQ, etc)
- connaître quelques notions de base du firewall : `iptables`
- trouver les ports ouverts sur une machine : `nmap`, `netstat`
- auditer les packages
- `/proc/sys/net/ipv4/ip_*`, `socket(7)`

37

Lectures supplémentaires

- http://www.debian.org/doc/manuals/securing-debian-howto/

Exercices

1. trouvez les fichiers qui ont le bit SUID ou SGID mis, ou qui n'appartiennent à aucun utilisateur, ou qui sont modifiables par n'importe qui
2. quel sont les rôle respectifs des packages : `debsums`, `integrit`, `nessus` et `logcheck` ?
3. vous lisez sur la liste BUGTRAQ qu'une vulnérabilité a été trouvée sur le plugin GPG pour Squirrelmail. Vous avez ce package installé depuis Debian etch standard mais ne savez pas si Debian a fait le travail. (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1924). Est-ce que cette vulnérabilité a été corrigée dans Debian ? Comment savoir ? (indications : http://www.securityfocus.com/, http://security.debian.org/, http://www.cert.org/, http://cve.mitre.org/)
4. utilisez l'outil `nmap` pour déterminer les ports ouverts sur la machine de votre voisin
5. votre serveur de mail est assailli par un autre serveur (une seule adresse IP). Proposez une solution (avec le firewall de Linux) pour bloquer le trafic SMTP en provenance de cette adresse. Que faire pour la supprimer ? (indication : http://www.brandonhutchinson.com/iptables_fw.html)
6. installez l'outil `nessus` et testez quelques vulnérabilités. Avez-vous des fausses alarmes ? si oui, pourquoi ?

Sécurisation de la machine – 9.2

Résumé des concepts importants

– configuration : `/etc/xinetd.d/*`, `/etc/xinetd.conf`, `/etc/inetd.d/*`, `/etc/inetd.conf`, `/etc/nologin`, `/etc/passwd`, `/etc/shadow`, `/etc/syslog.conf`

38

Exercices

1. faites en sorte que les mail pour root aille à une adresse donnée (NB : c'est normalement déjà le cas dans Debian, mais changez l'alias)
2. assurez-vous que les mots de passe sont géré via `/etc/shadow`. Documentez-vous sur la commande `pwconv`.
3. déconfigurez tous les services réseau inutiles (aidez-vous de `netstat -anp --listen`)
4. y-a-t-il des comptes sans mot de passe ?
5. comment feriez-vous pour créer des comptes UNIX qui n'autoriseraient pas de connexion SSH ?
6. pour quelle raison les erreurs d'authentification ne sont pas logguées dans `/var/log/syslog` ou `/var/log/messages` mais dans un fichier séparé ? Adaptez les permissions sur ce fichier si nécessaire (ainsi que la configuration `logrotate`).

Restrictions des utilisateur et processus – 9.3

Résumé des concepts importants

- savoir configurer des limites : `ulimit`
- commandes : `quota`, `usermod`

39

Exercices

1. documentez-vous sur le module PAM `pam_limits` (p.ex. : `http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/sag-pam_limits.html`) ainsi que sur le fichier `/etc/security/limits.conf`. Limitez le nombre de processus à 5 pour un utilisateur et testez. Où peut-on également configurer des limites sans utiliser PAM ?
2. donnez quelques exemples d'utilisation de la commande `ulimit` (indication : commande interne !)
3. quelle est la différence entre les limites **soft** et **hard** de `ulimit` ?

10. Corrigé des exercices

Kernel

Gestion des modules

1. `uname -r; /lib/modules/$(uname -r)/`
2. Il faut décharger tous les modules qui utilisent ce module (ils sont listés à droite), et vérifier ensuite que le compteur est à zéro. Sinon cela signifie qu'il y a des applications qui utilisent le périphérique (p.ex. utiliser `fuser /dev/lp0`), ou le kernel (p.ex. dans le cas d'un montage et d'un pilote de système de fichiers).
3. `modinfo -p lp`[1]`; modprobe lp parport=0x378;`
   `echo >> /etc/modprobe.d/options 'options lp parport=0x378'`
4. `depmod -a`
5. `lsmod | grep ipv6`; changer `ipv6` en `off` dans `/etc/modprobe.d/aliases`
6. `echo >> /etc/modprobe.d/aliases 'alias bla plip'`

Configuration, compilation et installation d'un kernel

1. manipulation simple
2. les variables de configuration pour la compilation du kernel, telles que générées par les interfaces de configuration du kernel comme p.ex. `make config`. Les sous-systèmes sont activés avec `Y` ou `M` (en module), et désactivés avec `N`. D'autres valeurs sont possibles pour des configurations spécifiques. Vous devriez lancer `make oldconfig`, au minimum.
3. intégrer *tous* les pilotes nécessaires au démarrage et montage des systèmes de fichiers (pas comme modules : avec `Y`, pas `M`).
4. `arch/i386/boot/bzImage` (kernel), `.config` (configuration), `System.map` (liste des fonctions et adresses kernel), évt. les modules à copier dans `/lib/modules/kernel-version/`
5. plus géré par le packaging, peut être écrasé en cas de mise à jour, nécessite de modifier la configuration de GRUB ou LILO manuellement.
6. création d'un package Debian binaire installable (pas aussi bon qu'un package créé via `make-kpkg`), d'un package RPM source et d'un package RPM binaire.

Démarrage, initialisation, arrêt et niveaux d'exécution

Démarrer le système

1. Cette option limitera la mémoire disponible (on verra la détection mémoire assez tôt dans le log `dmesg`). L'option `noapic` quant à elle supprimera la gestion d'interruption avancée. On ne verra que les 15 niveau d'interruption classiques.

[1] l'option -p a disparu dans les versions récentes : elle permettait de ne voir que les paramètres du module.

2. (a) spécifier un autre / qui sera monté par le kernel (utile p.ex. avec un CD de démarrage)
   (b) passage en mode mono-utilisateur (single-user mode), un mode de maintenance qui sera détaillé dans le sous-chapitre suivant.
   (c) `/var/log/messages`[2]
   (d) S'il y a trop de messages, la gestion de tampon circulaire écrasera les premières informations avant même qu'elles puissent être sauvegardées. L'option `log_buf_len=100k` permet de passer ce buffer à 100 kilo-octets.

Changement des niveaux d'exécution

1. `shutdown -r` 15 'mise a jour du kernel' ; CTRL–C ou `shutdown -c`
2. il interdit la connexion aux non-superutilisateurs. Peut être créé manuellement ou automatiquement pendant le démarrage ou lors d'un shutdown 5 minutes avant le shutdown effectif. Plus de détail : `shutdown(8)`
3. `runlevel` (sortie : précédent actuel)
4. `telinit 1` (ou `init 1`); `ps auxw`
5. `telinit 6` respectivement `telinit 0`
6. `echo message | wall`
7. modification de `/etc/inittab` (entrée `initdefault`)
8. `telinit q` (ou `init q`)

Impression

Gérer les imprimantes et les queues d'impression

1. `grep print /etc/services` ou `netstat -anp | grep lp` : 515/TCP
2. PostScript ; on utilise un filtre PostScript vers le format de l'imprimante (avec ghostscript, gimp-print, ou les filtres de CUPS)
3. Il n'est pas nécessaire d'installer les filtres d'impression sur les clients ni de pilotes spécifiques aux imprimantes (on peut le faire si nécessaire)

Installation et configuration d'imprimantes

1. manipulations et observations ; `lpc status`
2. `lpc disable lp; lpc status; lpc enable lp; lpc help; lpc topq JOB`
3. les fichiers modifiés sont : `/etc/cups/printers.conf`, `/etc/cups/ppd/NOM.ppd` (`find /etc/cups/ -mmin -5`) et `/var/run/cups/printcap`[3]
4. manipulation simple. `cupsdisable`
5. accepte ou non les travaux d'impression sur une queue
6. `/etc/cups/cupsd.conf` : `Browsing Off` (ne pas oublier de recharger la config de CUPS)

[2] Aussi /var/log/syslog.
[3] en tous cas sous etch, `/etc/printcap` n'est plus maintenu, voir une autre question plus bas.

7. `/etc/cups/cupsd.conf` : section (ditto)
8. `/etc/cups/cupsd.conf`
```
<Location />
Order Allow,Deny
Allow from 192.168.1.0/24
Deny from All
</Location>
```
(ditto)
9. le plus simple : `ln -s /var/run/cups/printcap /etc/printcap` (alternative : modifier cupsd.conf)
10. `lpadmin -p impr1 -E -v parallel:/dev/lp0 -m laserjet.ppd`
`lpadmin -x impr1`
11. package `cupsys-bsd`, lire `cups-lpd(8)` (configuration via `xinetd` ou `inetd`)

Imprimer des fichiers

1. les fins de lignes UNIX (linefeed, LF, ASCII 10) provoquent un escalier à l'impression (NB : uniquement si l'imprimante supporte encore l'impression texte simple, ce qui n'est pas gagné). Les accents éventuels ne sont pas bien imprimés. (solution : ajouter un filtre qui ajoute des carriage return, CR, ASCII 13, et qui transcode les caractères en CP-437 – ou passer par le format PostScript, PCL5 ou bitmap de l'imprimante)
2. `a2ps -o /tmp/a.ps /etc/motd`
`a2ps -1 -Pimprimante /etc/motd`
3. `lpr /tmp/a.ps`
`lpq -Pimprimante`
`lprm -Pimprimante NUMERO_JOB`

Documentation

Utiliser et gérer la documentation locale système

1. 1, 8, 4, 5, 7, 6
2. `apropos network` (ou `man -k network`)
3. `apropos crontab`
`man 5 crontab`
`man 1 crontab`
4. `whatis commande`
5. les documentations des programmes installés[4]
6. `export MANPATH=~/man:`[5]
`man -M ~/man tar` (voir aussi l'option `-l ~/man/man1/tar.1.gz`)

[4] sous Debian, chaque répertoire correspond à un package et contient divers fichiers obligatoires, voire une documentation plus étendue, qui peut aussi résider dans un package nommé `NOM-doc`.

[5] le : final permet de chercher ensuite dans le système également, si pas trouvé, voir `/etc/manpath.config`

7. `man -Tps intro > a-imprimer.ps`[6]
8. man : format standard UNIX ; GNU Info : plutôt FSF/Projet GNU. En général, GNU Info va plus en profondeur que man et contient parfois de petits tutoriaux.

Trouver de la documentation sur Internet

1. p.ex. Thunderbird, ou plus simple : tin. On peut aussi lire les news archivées sur Internet via Google groups
2. manipulation simple
3. manipulation simple
4. http://www.tldp.org/HOWTO/Man-Page/index.html

Notifier les utilisateurs

1. un texte affiché sur une des consoles textes locales, séries (aussi modem) ; respectivement par réseau (via `telnet` ou `rlogin`), usuellement identifie le système sur lequel on se connecte
2. un texte affiché au login (y compris `ssh`)
3. une commande qui permet d'afficher un message sur tous les terminaux textes des utilisateurs (employée notamment par `shutdown`)
4. un fichier qui, s'il existe, sera affiché juste avant de refuser le login aux utilisateurs non root (employé notamment lors du démarrage du système ainsi que par `shutdown` dans les dernières 5 minutes)

Scripts et programmation shell

Personnaliser et utiliser l'environnement

1. section INVOCATION de la manpage de bash, il y a 2 cas possibles :

   shell de login `/etc/profile` est sourcé ; ensuite le premier fichier existant et lisible dans la liste qui suit est lu : `~/.bash_profile`, `~/.bash_login` et `~/.profile` ; lorsque le shell quitte, `~/.bash_logout` est sourcé

   nouveau sous-shell interactif `/etc/bash.bashrc` et `~/.bashrc`

   Il y a encore le cas de shells non interactifs (lancement de scripts) qui est spécial. `~/.profile` est aussi sourcé en mode compatibilité `sh`.
2. configurer **libreadline** (entrée d'une ligne au clavier dans bash et ailleurs)
3. `export PATH=${PATH}:~/scripts`
4. `echo >> .bashrc 'export PATH=${PATH}:~/scripts'` (les apostrophes sont nécessaires, sinon le `${PATH}` risque d'être étendu au mauvais moment)
5. (a) sans `export`, sera disponible pour le shell (et donc les commandes internes), mais pas aux futurs sous-shells ou aux commandes externes[7].

   (b) c'est équivalent à la forme courte `export var=a`

[6] investiguer aussi `groff -Tps -man tar.1`

[7] par convention, on nomme souvent les variables exportées en majuscules, ce qui n'est pas le cas ici

(c) unset var

6. `-e` : quitter au premier résultat non zéro
   `-v` : voir les commandes exécutées.
   vous pouvez utiliser `bash -v nom-du-script`
7.
```
#! /bin/bash

function compresseur {
   if [ -f $1 ]; then
      case $1 in
         *.gz) ;;
         *) gzip -9 $1;;
      esac
   fi
}

compresseur /tmp/bla1
compresseur /tmp/truc.gz
```

Adapter ou écrire des scripts

1.
```
#! /bin/bash

function compresseur {
   if [ -f $1 ]; then
      case $1 in
         *.gz) return 1;;
         *) gzip -9 $1 && return 0
            return 2;;
      esac
   else
      return 2
   fi
}

compresseur /tmp/bla1; echo $?
compresseur /tmp/truc.gz; echo $?
```
2.
```
#! /bin/bash

function compresseur {
   local i

   for i
   do
      if [ -f $i ]; then
         case $i in
            *.gz) ;;
            *) gzip -9 $i;;
         esac
      fi
```

```
    done
}

compresseur /tmp/bla1 /tmp/truc.gz
```

3. (a) la ligne she-bang, du style : `#! /bin/bash` pour choisir l'interprète
   (b) la permission d'exécution : `chmod u`x un-script+
4. (a) `cp ~testuser/.bashrc` schaefer+
   (b) `chown schaefer:schaefer ~schaefer/.bashrc`
   (c) `chmod 700 ~schaefer/.bashrc` (ou moins restrictif, comme 755)
5. `touch 'date --iso-8601'`[8]
6. `[ -f /tmp/flag ] && [ ! -s /tmp/flag ] && echo "test" | mail root`
7. `if [ "$DISPLAY" = "" ]; then DISPLAY='hostname':0; fi`
8. `/usr/local/bin`
9. `for i in 'seq 10 -1 1'; do echo $i; done`
   `i=10; while [ $i -gt 0 ]; do echo $i; i=$(($i - 1)); done`
10.
```
fsck /tmp/blah
case $? in
   0) echo "Pas d'erreur";;
   1) echo "Petites erreurs corrigées";;
   2) echo "Vous devriez rebooter";;
   *) echo "Pas eu envie de traiter ces cas";;
esac
```
11. non, il faut alors utiliser un wrapper, en Perl, C, ou, plus simple, configurer `sudo`.

Administration

Gestion des comptes

1. voir les exercices et solutions du cours Administration
2. `/etc/passwd` doit être lisible par tous pour que les commandes comme `ls` puissent convertir l'UID en nom ; `/etc/shadow` contient les mots de passe hashés et ne devrait pas être accessible autrement qu'à root (p.ex. commande passwd SUID), ou éventuellement en lecture à un groupe spécifique pour vérifications.
3. `chage -E 0 demo`. On peut limiter un compte en durée de validité.
4. oui, ajouter le groupe à l'utilisateur (p.ex. `usermod -G groupe1,groupe2,...,groupeN`, où `groupeN` est le nouveau groupe à ajouter et les autres sont les groupes à maintenir
5. `passwd -L demo`, `passwd -U demo`

Préconfiguration de l'environnement

1. manipulation simple ; option `-m` de `useradd` pour copier depuis `/etc/skel`. Utiliser l'option `-k` pour spécifier un autre répertoire skel pour implémenter des classes d'utilisateurs.
2. manipulation simple

[8] avec bash on peut remplacer les *backticks* par `$()`

Configuration et gestion des journaux

1. client : `echo "*.* @serveur" > /etc/syslog.conf`
   serveur : option `-r` au démarrage
2. `echo "auth.=debug /dev/tty8" >> /etc/syslog.conf`
3. supprime le `fsync(2)` implicite à chaque log : cela augmente les performances mais ne garantit plus la présence des derniers logs sur le disque en cas de crash.
4. configuration de base : `/etc/logrotate.conf`, avec éventuellement un répertoire `/etc/logrotate.d`. On y spécifie la fréquence de déplacement des logs et les paramètres de compression, voire des scripts à lancer avant ou après le déplacement des logs, les permissions et le nombre de copies qui sont maintenues. logrotate est lancé depuis `/etc/cron.daily/logrotate`
5. les facilités entières auth et authpriv vont dans auth.log ; tout va dans syslog sauf tout auth et authpriv

Travaux exécutés automatiquement

1. atd : gère les travaux à exécuter une fois en mode batch (p.ex. en respectant une charge maximum de la machine) ; cron : les travaux à exécuter régulièrement (p.ex. chaque heure, chaque 1er jour du mois, etc)
2. à lancer des travaux journaliers, hebdomadaires ou mensuels à la première occasion. Sert principalement sur des machines qui ne sont pas enclenchées 24h/24 et 7j/7 (p.ex. des postes clients)
3. périodicité (jours) ; délai après le boot (minutes) ; identification ; script
4. `echo 'echo "test" | /usr/bin/mail root' | at now` 5 min+
   `atq`
   `atrm NUMERO`
5. `/var/spool/cron/crontabs/root` (via `crontab -e`), `/etc/crontab`, `/etc/cron.d/*`, `/etc/cron.{daily,weekly,monthly}/*`
6. la colonne *utilisateur* est implicite
7. `echo >> /etc/at.deny "demo"` (uniquement si `/etc/at.deny` n'existe pas)
8. aucun utilisateur (sauf root) ne sera autorisé

Maintenir une sauvegarde fonctionnelle

1. `dd if=/dev/hda1 of=/tmp/un_fichier`
2. utilisé en conjonction avec le fichier `/var/lib/dumpdates` et l'option `-W` de `dump`, il permet d'indiquer automatiquement quels systèmes de fichiers devraient être sauvegardés et à quel niveau.
3. `chattr +d /tmp` ; fonctionne également pour les sauvegardes complètes `dump` si `-h 0` est spécifié.
4. `dump 0nf /dev/nst0 /home`
   `restore rf /dev/nst0`
5. permet de faire plus rarement des sauvegardes de niveau 1, car chaque fichier sauvegardé est au moins sauvé deux fois, statistiquement.

6. `tar -czf /tmp/bla.tar.gz /etc`
   ```
   (cd /tmp && tar -xzf /tmp/bla.tar.gz
   find . -type f -print | cpio -o > /tmp/bla.cpio
   cpio -imBmdu < /tmp/bla.cpio
   ```
7. oui, tar peut utiliser un fichier d'état (pas nécessaire de le sauvegarder) pour implémenter des niveaux > 0, via l'option `--listed-incremental`. cpio peut être associé, grâce à find, à un fichier témoin. tar peut également travailler par ce biais.
8. option `-C` de `restore`. Option `--compare` de `tar`

Gérer le temps système

1. nous avons ainsi reconfiguré la notion de fuseau horaire pour cette commande seulement
2. il se peut que certains programmes aient de la peine (p.ex. screen saver, cron, etc). On verra qu'au démarrage, la modification n'est pas prise en compte[9].
3. `/etc/timezone` : nom textuel de la zone
   `/etc/localtime` : lien symbolique à la zone en format compilé `zic` : contient des indications de changement d'heure d'été et d'hiver, des directives spéciales (p.ex. la *leap second* servant à compenser le ralentissement de la Terre), le tout sur plusieurs années :
   ```
      zdump -v /etc/localtime
   Sun Mar 29 01:00:00 1981 UTC = Sun Mar 29 03:00:00 1981 CEST
      isdst=1 gmtoff=7200
   Sun Sep 27 00:59:59 1981 UTC = Sun Sep 27 02:59:59 1981 CEST
      isdst=1 gmtoff=7200
   [ ... ]
   Sun Mar 31 01:00:00 1996 UTC = Sun Mar 31 03:00:00 1996 CEST
      isdst=1 gmtoff=7200
   Sun Oct 27 01:00:00 1996 UTC = Sun Oct 27 02:00:00 1996 CET
      isdst=0 gmtoff=3600
   ```
4. `UTC=no`. Notons que la "méthode Microsoft" ne permet plus, par la suite, d'identifier de manière univoque l'heure de création d'un fichier dans tous les cas (que cela soit sous Microsoft ou sous GNU/Linux)
5. manipulation simple
6. `apt-get install ntp-simple` (trop facile !)
7. ```
   $ grep -l hwclock /etc/rc?.d/* /etc/init.d/*
   /etc/rc0.d/K25hwclock.sh
   /etc/rc6.d/K25hwclock.sh
   /etc/rcS.d/S20checkroot.sh
   /etc/rcS.d/S50hwclock.sh
   /etc/init.d/checkroot.sh
   /etc/init.d/hwclock.sh
   ```

Bases du réseau

Rappels sur TCP/IP

[9] sauf hwclock -w -u dans un des fichiers de /etc/rc6.d ...
   ```

1. 32 bits. Sous forme d'un quadruplet (4 x 8 bits), p.ex. 157.26.173.31
2. CIDR : 193.72.186.0/24, d'où netmask 255.255.255.0
3. 80.83.54.61 : classe A (0-127). netmask 224 (256 - 32, soit 5 bits de liberté, donc CIDR 80.83.54.32/27), donc sous-réseau 80.83.54.32, broadcast 80.83.54.63.
4. /26 correspond à $(32 - 26 == 2^6)$, donc netmask 255.255.255.192. L'adresse sous-réseau est donc `192.168.1.0`, le broadcast `192.168.1.63`
5. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16. Le routeur devrait jeter ces datagrammes s'ils sortent d'un réseau privé et ne pas en accepter de l'extérieur.
6. spécifier une route qui sera utilisée en dernier recours. En général pointera sur le routeur du sous-réseau.
7. `route add 192.168.100.A dev eth0` (ou `route add -net 192.168.100.0/24 dev eth0` p.ex.)
8. manipulation simple
9. `host -t aaaa domreg.nic.ch`
10. `dig -t axfr alphanet.ch @ns1.ecoweb.ch`
11. `ftp`
12.

couche	protocoles
4	TCP et UDP
3	IP (et ICMP)

13. `grep NUMERO-DE-PORT /etc/services`
14. remplacer l'entrée `ipv6` à `off` dans `/etc/modprobe.d/aliases`

Configuration et détermination de problèmes

1. `killall dhcpcd` (ou pump, etc, cf ps (ou encore ifdown eth0))
2. manipulation simple
3. (a) problème DNS, vérifier `/etc/resolv.conf`
   (b) vérifier les routes, notamment la route par défaut, ou un firewall éventuel
4. ifconfig travaille en bas-niveau *sans* tenir compte de la configuration effectuée (p.ex. Debian dans `/etc/network/interfaces`)
5. `route add 192.168.42.35 dev eth0` (si pas déjà dans les routes)
   `route add -net 192.168.100.0/24 gw 192.168.42.35`
6. la commande ne change que l'état du kernel, au prochain démarrage la modification sera perdue
7. certains programmes (p.ex. Apache) ont besoin d'un nom correct en résolvant l'adresse IP (vers le nom) sur chaque interface active. Les aliases sont ajoutés en colonnes supplémentaires. Sur un réseau, le DNS est préféré.
8. `domainname` concerne le domaine NIS.
9. `netstat -pan`
10. `echo 'enterprise-net 192.168.100.0' >> /etc/networks`
11. `host.conf` : configuration de l'ordre de résolution entre `/etc/hosts`, `DNS` et NIS.
    `nsswitch.conf` : configuration de l'endroit où certaines bases de données (`/etc/passwd`, etc) sont stockées (fichiers dans `/etc`, bases de données NIS, LDAP, etc)
12. `/etc/init.d/networking` (p.ex.)
13. `tcpdump` ou `strace`

Services réseau

Super-serveur inetd/xinetd

1. manipulation simple
2. `echo "service-simple: 1.2.3.4" >> /etc/hosts.deny`
   la connexion s'ouvre et se ferme tout de suite
3. `update-inetd` (plutôt que de toucher le fichier directement)
4.
```
echo 'ma-redirection 2222/tcp' >> /etc/services
cat > /etc/xinetd.d/redirect-demo <<EOF
service ma-redirection
{
   socket_type = stream
   wait        = no
   redirect    = localhost 22
   user        = nobody
}
EOF
```

5. avec `xinetd`
   - `instances=` : nombre d'instances d'un service
   - `nice=` : priorité
   - `per_source=` : nombre d'instances par adresse IP
   - `cps=` : connexions/secondes et durée d'arrêt
   - `max_load=` : charge système maximum
6. dès le moment où le service est très souvent accédé ou a sa propre gestion, il vaut mieux qu'il soit en mode indépendant
7. `stream` est pour le type de socket(7) format un flux de données en mode connecté (p.ex. TCP) alors que `dgram` s'utilise dans le cas de datagrammes isolés (p.ex. UDP).

Configuration et gestion de base d'un MTA

1. manipulation simple
2. exemple : `ls -1 | sendmail -oi schaefer`
3. `echo autre-utilisateur > ~/.forward`
4.
```
echo utilisateur: autre-utilisateur >> /etc/aliases
newaliases
```
5. il est très possible que mon serveur refuse votre message pendant quelques minutes pour des raisons d'anti-spam.
6. manipulation simple
7. exim : par défaut le relaying est interdit, du moins sous Debian. Les variables concernées sont gérées par debconf, visibles dans `/etc/exim4/update-exim4.conf.conf` et modifiables via `dpkg-reconfigure exim4-config`. Alternativement, dans le fichier `/etc/exim4/exim4.conf.template`, on peut trouver les variables suivantes qui ont un rapport avec le relaying :
   - `MAIN_LOCAL_DOMAINS` (domaines locaux)
   - `MAIN_RELAY_TO_DOMAINS` (domaines relayés)

– `MAIN_RELAY_NETS` (autorisation par adresses IP)

8. Postfix : `/etc/postfix/main.cf`
   – `mydestination` (domaines locaux)
   – `relay_domains` (domaines relayés)
   – `mynetworks` (autorisation par adresses IP)
   sendmail
   – `/etc/mail/access` (autorisations par adresses IP ou noms)
   (nécessite une regénération des maps via `makemap`)
9. entrée `DSsmtp.fai.ch` du fichier `/etc/sendmail.cf`

Configuration et gestion de base d'Apache 2

1. démarrer, arrêter, voir l'état, tester la configuration avant de l'activer
2. ajouter `Listen 8080` et recharger (`/etc/init.d/apache2 reload` ou via `apache2ctl`)
3. `a2enmod userdir` puis recharger
4. configurer des restrictions d'accès par répertoire, p.ex. des mots de passe ou des exécutions particulières
5. voir par exemple http://httpd.apache.org/docs/2.0/mod/mod_log_config.html

Bases de NFS et Samba

1. manipulation simple
2.
```
echo "/home 157.26.173.0/255.255.255.0(rw)" >> /etc/exports
exportfs -v -a
showmount -e localhost
```

3. `mount adresse-ip-voisin:/home /mnt`
   non, les droits du root local sont transformés en droits de nobody sur le serveur NFS (sauf si l'option d'exportation `no_root_squash` est activée)
   oui, il est possible de faire un simple `su utilisateur` en tant que root sur le client pour avoir accès aux fichiers de l'utilisateur.
4. `echo "adresse-ip-voisin:/home /mnt nfs defaults 0 0" >> /etc/fstab`
5. `smbclient -L localhost` (RETURN à la question du mot de passe)
6. manipulation simple
7. par exemple
```
[public]
comment = Test
writable = yes
public = yes ; puis no
path = /tmp
```

8. manipulation simple
9. `wins server = 1.2.3.4`
10. chargement automatique d'une configuration LPR/BSD/CUPS, partage éventuel spécial `print$` (où sont stockés des pilotes Windows éventuels), partage `printers` avec génération des noms automatiques
11. `\\SERVEUR\NOM` ou `\\SERVEUR\HOMES`

Bases du DNS

1. le resolver consulte `/etc/host.conf` pour déterminer si `/etc/hosts` doit être consulté d'abord, puis éventuellement `/etc/nsswitch.conf` pour voir si ce fichier est en fait une base de données réseau NIS ou autre. En cas d'échec, on consulte `/etc/resolv.conf` et on essaie en séquence les serveurs de noms (`nameserver`) qui y sont configurés. Ces serveurs de noms, s'ils sont sympathiques (recurse) vont se charger de la résolution entière. Sinon, ils vont simplement donner un pointeur et c'est le client qui devrait faire les étapes. Ces étapes consistent tout d'abord à obtenir le serveur qui gère la racine du DNS (`.`). Ils sont en général configurés en dur (p.ex. `/etc/bind/db.root`). On en choisit un. On lui demande de résoudre `www.alphanet.ch`. En général, il ne répondra que "où est `ch.`". On contacte alors le NS de `ch.`, qui nous donne le NS de `alphanet.ch` : on continue jusqu'à la réponse (type A).
2. manipulation simple
3. `apt-get install bind9`
   `cat /etc/resolv.conf`
4. mettre `forwarders { 1.2.3.4; 5.6.7.8; }` dans `/etc/bind/named.conf.options` (p.ex.)
5. dans `/etc/bind/test.ch.zone` (p.ex.) :

```
$TTL    86400
@                       IN  SOA ns1 postmaster.test.ch. (
                                2007081801  ; serial number
                                18000       ; refresh
                                3600        ; retry
                                604800      ; expire
                                43200     ) ; minimum TTL
@               IN      NS              ns1 ; ns1.test.ch
@               IN      NS              ns1.imp.ch. ; externe

@               IN      MX      10      smtp

www             IN              A               1.2.3.4
ns1             IN              A               4.5.6.7
smtp            IN              A               8.9.0.1
```

   ne pas oublier de charger la zone, p.ex. dans `/etc/bind/named.conf.local`

```
zone "test.ch.ch" {
   type master;
   file "/etc/bind/test.ch.zone";
};
```

   puis de recharger (évt. utiliser `named-checkconf` ou `named-checkzone` avant).
6. voir par exemple `/etc/bind/db.127`
7. il vous faut deux serveurs de noms enregistrés chez NIC/CH (en particulier s'ils résident dans le nouveau domaine), qui répondent la même chose (p.ex. configurer le 2e en secondaire, qui télécharge de temps en temps la zone du primaire)

8. `allow-query` : définir quels clients peuvent faire des requêtes.
   `allow-recursion` : définir quels clients peuvent charger le serveur DNS de chercher une réponse qui n'est ni disponible localement (*authoritative*) ni dans le cache. Les autres se verront référer simplement à un serveur de l'arborescence de recherche.

SSH

1. `apt-get install ssh`
2. manipulation simple
3. `ssh-keygen -p`
4. manipulation simple
5. manipulation simple

Sécurité

Tâches administratives de sécurité

1. `find / -type f \( -perm +6000 -o -nouser -o -nogroup -o -perm -2 \)`
   (on pourrait avoir envie de vérifier les répertoires inscriptibles également via un `type d`)
2. vérifier les md5sums des fichiers installés en package.
   créer une base de données de fichiers (aussi données ou logiciels installés localement, configurations, etc si désiré) à fin de vérification automatique.
   trouver des vulnérabilités automatiquement.
   résumer les logs avec des expressions régulières et envoyer par mail les entrées inquiétantes.
3. dans ce cas, cette partie vulnérable n'est pas packagée dans le package Debian standard pour etch, voir http://www.debian.org/security/nonvulns-etch (la liste des NON vulnérabilités, référencée de http://security.debian.org/.
4. `$ nmap -A -T4 login`

```
Starting Nmap 4.03 ( http://www.insecure.org/nmap/ ) at 2008-01-31 17:08 CET
Interesting ports on login.alphanet.ch (80.83.54.2):
(The 1660 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 3.8.1p1 Debian-8.sarge.6 (protocol 2.0)
25/tcp   open  smtp       Postfix smtpd
53/tcp   open  domain     ISC Bind 9.2.4
80/tcp   open  http       Apache httpd 1.3.33 ((Debian GNU/Linux) mod_ssl/2.8.22
 OpenSSL/0.9.7e mod_perl/1.29)
81/tcp   open  http       Apache httpd 1.3.33 ((Debian GNU/Linux) mod_ssl/2.8.22
 OpenSSL/0.9.7e mod_perl/1.29)
110/tcp  open  pop3       Courier pop3d
113/tcp  open  ident      OpenBSD identd
119/tcp  open  nntp       INN nntpd 2.4.3 (posting ok)
143/tcp  open  imap       Courier Imapd (released 2004)
443/tcp  open  http       Apache httpd 1.3.33 ((Debian GNU/Linux) mod_ssl/2.8.22
 OpenSSL/0.9.7e mod_perl/1.29)
993/tcp  open  ssl/imap   Courier Imapd (released 2004)
995/tcp  open  ssl/pop3   Courier pop3d
5432/tcp open  postgresql PostgreSQL DB
6667/tcp open  irc        Hybrid-based ircd
Service Info: Hosts:  shakotay.alphanet.ch, news.alphanet.ch; OS: OpenBSD

Nmap finished: 1 IP address (1 host up) scanned in 19.037 seconds
```

5. `iptables -I INPUT -i eth0 -s 1.2.3.4/32 -p tcp --dport 25 -j DROP` (pour supprimer, remplacer `-I` par `-D`)
6. les signatures sur lesquelles se base nessus sont basées sur des numéros de version : si la vulnérabilité du programme X est corrigée dans la version Y, il se peut très bien que Debian montre que la version installée est U < Y (avec backport du patch vers la version Y) : non vulnérable.

Sécurisation de la machine

1. `vi /etc/aliases;newaliases`
2. manipulation simple
3. manipulation simple
4. manipulation simple
5. p.ex. via `DenyGroups` ou `AllowGroups` dans `/etc/ssh/sshd_config`.
6. si la personne tape son mot de passe à un prompt `login`, on ne veut pas que ces informations soient lisibles par n'importe qui.

Restrictions des utilisateur et processus

1. on peut imaginer configurer ces limites dans les scripts de démarrage système du shell bash, par exemple.
2. `ulimit -a` : liste de toutes les limites
   `ulimit -c unlimited` : autoriser les core dumps
   `ulimit -v 500000` : limiter l'utilisation mémoire à 500MB
3. (voir `man 2 setrlimit`) : la limite soft est celle gérée par le kernel. La limite hard est la limite supérieure de configurabilité de la limite soft. Seul un processus avec des droits spéciaux (capacité `CAP_SYS_RESOURCE`) peut augmenter la limite hard. En conséquence, un script système devrait toujours changer les limites hard.