Formation Evolix Administration Linux

Focus sur Apache, PHP, MySQL et Varnish/Redis/Memcache

Evolix - Mai 2012

evolix
Informatique et Logiciels Libres

Table des matières

Chapitre 1

Introduction à Unix/Linux

1.1 Les débuts

En 1969, Ken Thompson [1], employé dans les laboratoires Bell [2], développe UNICS (UNiplexed Information and Computing Service), système d'exploitation mono-utilisateur écrit en langage assembleur. Rebaptisé UNIX, Ken Thompson tenta en 1971 de réécrire le système en langage FORTRAN ou en B. Entre-temps Dennis Ritchie [3], également employé dans les laboratoires de Bell a pris part au projet, et mis au point le successeur du langage B : le langage C. UNIX fut donc réécrit en langage C et nommé UNIX Time-Sharing System (UTS). UNIX commença également à être diffusé hors des laboratoires de Bell. Or les laboratoires de Bell appartiennent à la société AT&T qui ne peut commercialiser autre chose que des équipements téléphoniques ou télégraphiques. La décision fut prise de distribuer le système UNIX complet avec son code source complet dans les universités puis également dans les entreprises. De nombreuses contributions furent apportées par l'université de Berkeley (Californie) qui distribua ainsi UNIX BSD (Berkeley Software Distribution).

Les laboratoires Bell développèrent le système UNIX Time-Sharing System jusqu'en octobre 1989 (sa 10ème version). La branche commerciale d'AT&T développa les premières versions commerciales d'UNIX : System III puis System V.

Les droits d'UNIX appartenant à AT&T ont été rachetés par l'entreprise Novell [4]. En 1994, Novell a transféré les droits sur la marque UNIX ainsi que les spécifications à l'OpenGroup [5], et a également revendu le code source et l'implémentation UNIXWARE (dérivée de System V) à l'entreprise SCO [6].

L'université de Berkeley développa UNIX BSD jusqu'en 1993 (4.4BSD) dont dérivent les projets libres NetBSD, FreeBSD et OpenBSD mais également le système d'exploitation d'Apple MAC OS X. Du fait de la mise à disposition du code source d'UNIX, de nombreux dérivés propriétaires d'UNIX furent développés : AIX (IBM), Solaris (Sun Microsystems), HP-UX (Hewlett-Packard), Ultrix (DEC), Xenix (Microsoft), Unixware (SCO), Tru64 (DEC), IRIX (SGI), etc.

Les systèmes d'exploitation UNIX sont multi-tâches, multi-utilisateurs et en général ouverts

1. http://www.cs.bell-labs.com/who/ken/
2. http://cm.bell-labs.com/
3. http://cm.bell-labs.com/cm/cs/who/dmr/
4. http://www.novell.com/
5. http://www.opengroup.org/
6. http://www.sco.com/

(code source disponible).

Liens :
http://fr.wikipedia.org/wiki/UNIX
http://www.commentcamarche.net/unix/
http://www.unix.org/
http://www.levenez.com/unix/

1.2 Historique des logiciels libres

Au commencement de l'informatique, le matériel était volumineux et coûteux. On se focalisait surtout sur l'équipement : les codes des programmes étaient disponibles gratuitement et chacun pouvait les améliorer librement. Avec l'apparition de la micro-informatique dans les années 1980, les développements de logiciels furent de plus en plus nombreux. De nombreuses restrictions apparurent sur ces logiciels (licences, accord de non-divulgation) qui étaient souvent vendus uniquement sous forme de binaires. En 1984, suite à des problèmes liés à ces restrictions, Richard Stallman [7], informaticien au MIT(Institut de Technologie du Massachusetts), démissionne du laboratoire où il travaille pour fonder le projet GNU [8]. GNU est un acronyme récursif qui signifie GNU's Not Unix (GNU n'est pas Unix).

Son ambition est de développer un système d'exploitation complètement libre et promouvoir la liberté des logiciels. Ainsi, en 1985 est créée la Free Software Fundation (FSF) qui a écrit un grand nombre de paquetages logiciels, notamment GNU Compiler Collection (GCC) et Bourne-Again SHell (BASH [9]). Des licences dites libres ont également été rédigées, dont la GNU General Public License (GPL [10]) qui oblige les programmes dérivés à rester avec la même licence.

D'autres licences dites libres existent comme la licence BSD [11] qui permet la réutilisation du code sous la dite licence dans des programmes propriétaires. Le terme d'Open Source définit un ensemble de licences moins restrictives. L'Open Source Initiative [12] publie sur son site les licences dites Open Source.

La définition de liberté pour un logiciel n'est pas forcément triviale et diffère selon certains organismes.

La Free Software Fundation (FSF) spécifie les 4 libertés pour un logiciel libre :

- La liberté d'exécuter le programme, pour tous les usages **(liberté 0)**.
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins **(liberté 1)**. Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies, donc d'aider votre voisin, **(liberté 2)**.

7. http://www.stallman.org/
8. http://www.gnu.org/
9. http://www.gnu.org/software/bash/
10. http://www.gnu.org/copyleft/gpl.html
11. http://www.opensource.org/licenses/bsd-license.php
12. http://www.opensource.org/licenses/

– La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté **(liberté 3)**. Pour ceci l'accès au code source est une condition requise.

Lien : http://www.gnu.org/philosophy/free-sw.fr.html

1.3 Définitions des 'logiciels libres'

Le projet Debian spécifie ses propres principes du logiciel libre :

– redistribution libre et gratuite
– distribution du code source
– aucune discrimination de personne ou de groupe
– aucune discrimination de champ d'application
– distribution de licences
– la licence ne doit pas être spécifique à Debian
– la licence ne doit pas contaminer d'autres logiciels

Lien : http://www.debian.org/social_contract#guidelines

L'Open Source Initiative (OSI) spécifie les critères pour un logiciel Open Source, inspirés des principes du logiciel libre selon Debian :

– Libre redistribution
– Code source
– Travaux dérivés
– Intégrité du code source de l'auteur
– Aucune discrimination envers les personnes ou les groupes
– Aucune discrimination envers les champs d'effort
– Distribution de la licence
– Licence non spécifique d'un produit
– Licence non restrictive envers d'autres produits
– Licence neutre technologiquement

Lien : http://opensource.org/docs/definition.php

On se rend compte que la définition d'un logiciel libre est beaucoup plus complexe qu'on pourrait le croire. Elle s'appuie notamment sur la notion de respect des droits d'auteurs, comme tout écrit, et non sur la notion de brevets valables pour les inventions. Il faut noter que la notion de brevets logiciels existe désormais dans plusieurs pays (notamment aux États-Unis) et des discussions ont lieu ces dernières années au niveau de l'Europe pour l'adoption de ce principe. [13]

Pour bien se rendre compte des subtilités entre les différentes appellations, il suffit d'énumérer quelques catégories de logiciels : [14]

13. http://brevets-logiciels.info/
14. http://www.gnu.org/philosophy/category.fr.jpg

- Logiciel libre
- Logiciel Open source
- Logiciel du domaine public
- Logiciel copylefté (sous gauche d'auteur)
- Logiciel libre non-copylefté
- Logiciel couvert par la GPL
- Logiciel privé
- Logiciel propriétaire
- Shareware (Partagiciel)
- Freeware

Lien : http://www.gnu.org/licenses/license-list.fr.html

- **La licence GNU General Public License (GPL)** [15]
  Cette licence a été écrite par la FSF (Free Software Foundation) comptant pour membres Richard Stallman et Eben Moglen [16]. Elle a été écrite pour fixer les conditions légales de distribution des logiciels du projet GNU.
  Elle fait partie des licences restrictives (notion de copyleft). Il faut aussi savoir que c'est probablement la licence libre la plus utilisée aujourd'hui (Linux, GCC, KDE, Gnome, etc.).

  Liens :
  http://www.gnu.org/copyleft/gpl.html
  http://www.linux-france.org/article/these/gpl.html
  http://fsffrance.org/gpl/gpl.fr.html
  http://crao.net/gpl/

- **Les licences de type BSD**

  La licence BSD originale est une licence simple et permissive (elle n'est pas soumise au principe du copyleft). Néanmoins, elle comporte une clause de publicité qui a provoqué de nombreux débats (la licence impose un texte à mentionner chaque fois que le logiciel est cité). Cette clause a été supprimée par la suite dans ce qu'on appelera la licence BSD modifiée.
  Cette licence et des variantes de cette dernière sont utilisées notamment par les projets FreeBSD, NetBSD, OpenBSD, etc.
  Elle ne comporte aucune restriction, ce qui permet à des sociétés comme Microsoft de réutiliser le code source placé sous cette licence, ou d'inclure dans MacOSX des parties de FreeBSD et d'OpenBSD.

  L'université de Berkeley développa UNIX BSD jusqu'en 1993 (4.4BSD) dont dérivent MAC OS X (Apple) ainsi que de nombreux projets libres comme NetBSD, FreeBSD et OpenBSD, etc.

  Le projet OpenBSD définit un logiciel libre, comme un logiciel sous une licence n'apportant aucune restriction. Certaines licences peuvent être acceptées pour certaines parties.

  La capacité à disposer d'un Unix Berkeley librement distribuable permet d'avancer sur

15. http://www.gnu.org/copyleft/gpl.html
16. http://emoglen.law.columbia.edu/

une base compétitive par rapport aux autres systèmes d'exploitation, mais dépend directement de la volonté des différents groupes de développement qui échangent des sources, entre-eux et entre projets. Comprendre les implications légales qui entourent le concept de "copyright" est fondamental afin de pouvoir échanger et redistribuer du code source, et somme toute de promouvoir la coopération des personnes impliquées.

Chaque système se concentre sur des points particuliers, ou du moins en théorie :

- FreeBSD est orienté performances et applications ; l'enjeu n'est clairement pas de supporter un grand nombre d'architectures
- NetBSD mise sur la portabilité avant tout, avec un grand nombre d'architectures supportées.
- OpenBSD accentue ses efforts de développement sur la sécurité, de son système et des applications qui le composent. Une grande attention est portée au niveau des licences.

Il existe également d'autres systèmes BSD moins connus, comme DragonflyBSD, etc.

Lien : http://www.opensource.org/licenses/bsd-license.php

- **La licence Artistique**
  Licence mise en place par le créateur de Perl, Larry Wall. Outre les droits d'utilisation, de modification, et de distribution, l'auteur conserve certains droits (droit de négocier des arrangements au coup par coup, interdiction de diffuser une version entrant en conflit avec la distribution "standard" de l'auteur).

  Liens :
  http://www.opensource.org/licenses/artistic-license.php
  http://www.perl.com/pub/language/misc/Artistic.html
  http://linux-france.org/article/these/licence/artistic/fr-artistic.html

- **La GNU Lesser General Public License (LGPL)**
  LGPL signifie Licence publique générale limitée GNU, ou GNU LGPL (pour GNU Lesser General Public License) en anglais. Comme la Licence publique générale GNU (ou GNU GPL), elle a été écrite par la Free software foundation. La différence avec la GPL est que la LGPL permet de lier un programme tiers non libre à une bibliothèque LGPL, sans pour autant révoquer la licence (licence non copyleft). Elle est surtout utilisée pour des librairies (elle s'appellait initalement Library General Public License)

  Lien : http://www.gnu.org/licenses/lgpl.html

- **Licence Apache**
  La licence Apache en est à sa Version 2.0 (approuvée le 21 janvier 2004 par la fondation Apache). Tous les projets Apache (dont Ant, Jakarta, ou Cocoon) sont sous cette licence. Cette licence est libre, non-copyleft, et compatible GPL (en version 1.0 et 1.1, la Apache License n'était pas compatible GPL).

  Lien : http://www.apache.org/licenses/

- **Licence X11 (ou MIT)**

Cette licence est simple et permissive, sans copyleft, compatible avec la GPL de GNU. Les anciennes versions de XFree86 utilisaient cette licence, et quelques variantes actuelles de XFree86 l'utilisent également. Les licences ultérieures de XFree86 sont distribuées sous la licence XFree86 1.1 (qui est incompatible avec la GPL). La licence est parfois appelée «licence du MIT» mais ce terme est trompeur : le MIT a publié ses logiciels sous diverses licences.

Liens :
http://www.x.org/Downloads_terms.html
http://www.opensource.org/licenses/mit-license.php

– **Licence Mozilla Public Licence**
Cette licence n'est pas très stricte en terme de "copyleft" ; contrairement à la licence X11 elle présente des restrictions complexes qui la rendent incompatibles avec la GPL de GNU. Ainsi, on ne peut pas, légalement, lier un module couvert par la GPL et un module couvert par la MPL. Pour entrer dans le détail, la licence MPL 1.1 permet (section 13) à un programme ou à une portion de programme d'offrir le choix entre la MPL et une autre licence. La licence d'une partie de programme qui offre le choix de la GPL est alors compatible avec la GPL. Les logiciels de Mozilla mais également NVu (création de pages Web), Compière (ERP/CRM) sont sous licence MPL. SugarCRM utilise la SPL (SugarCRM Public Licence) qui est très proche de la MPL.

Liens :
http://www.mozilla.org/MPL/
http://www.sugarcrm.com/home/Public_License_FAQ/228/

– **Licence IBM Public License**
Cette licence en est à sa version 1.0 et est incompatible avec la GPL en raison de d'exigences spécifiques. Elle exige notamment que certains droits soient accordés en-dehors de ce que la GPL prévoit.

Lien : http://oss.software.ibm.com/developerworks/opensource/license10.html

– **Licence Sendmail** [17]
Il s'agit de la licence du serveur de messagerie électronique éponyme, le plus populaire sur Internet.

Lien : ftp://ftp.sendmail.org/pub/sendmail/LICENSE

– **Licence Common Public License**
Licence libre qui n'est pas compatible avec la GPL. La Common Public License est incompatible avec la GPL parce qu'elle énonce diverses exigences spécifiques qui ne se trouvent pas dans la GPL. Elle exige notamment que certaines licences de brevets soient données, ce que la GPL n'exige pas.

Lien : http://www.eclipse.org/legal/cpl-v10.html

Licences pour un usage particulier

17. http://www.sendmail.org/

- **Licence GNU Free Documentation License**
  Cette licence a été conçue pour les documents sous copyleft. Elle convient pour d'autres catégories d'oeuvres utiles telles que les manuels scolaires ou les dictionnaires, par exemple. Son domaine d'application n'est d'ailleurs pas exclusivement celui des oeuvres textuelles.

  Lien : http://www.gnu.org/copyleft/fdl.html

- **Les licences Creative Commons**
  Des juristes de l'université de Stanford ont créé un ensemble de licences destinées aux contenus tels que l'audio, la vidéo, les images, les textes et les ressources éducatives. Le but est de faciliter la mise à disposition et le partage de ces contenus aux auteurs avec les restrictions qu'ils veulent (usage commercial, possibilité de modifications, etc.) Il suffit en effet de visualiser deux ou trois pages illustrées d'images explicites sur le site Internet pour choisir la licence appropriée.

  Liens :
  http://creativecommons.org/
  http://fr.creativecommons.org/
  http://philippe.daigremont.free.fr/CreativeCommons/BD/

- **Licence Art Libre**
  Cette licence libre copyleftée est faite pour les oeuvres artistiques. Elle autorise la distribution commerciale, tout en précisant qu'une oeuvre de plus grande taille qui inclurait l'oeuvre soumise à la licence doit être elle-même libre.

  Lien : http://artlibre.org/licence.php/lal.html

1.4 Modèle de développement

Le modèle de développement des logiciels libres connaît un vif succès grâce à l'expansion des réseaux et d'Internet dans les années 1990. Ainsi, de nombreux projets (dont l'un des plus marquants est Linux) voient le jour avec une communauté répartie dans le monde entier. Des logiciels initiés à des buts commerciaux basculent également sous des licences libres afin de profiter des avantages de la communauté du logiciel libre (beta-testeurs nombreux, accroissement de l'équipe de développeurs, politique de sécurité, etc.). Parmi les logiciels libres, on distingue donc de nombreux projets issus de communautés (Apache, Debian, FreeBSD, NetBSD, Sendmail, etc.) et des projets proches d'entreprises commerciales (Mandrake, Red Hat, OpenOffice, etc.).

En ce qui concerne l'organisation, la philosophie des logiciels libres basée sur l'ouverture n'empêche pas une structuration précise. On peut ainsi dégager un modèle de développement typique qui comprend :

- Une ou plusieurs personnes responsables globalement du projet (souvent à l'initiative du projet ou élues)

- Plusieurs développeurs officiels du projet ayant été acceptés avec précaution
- Des contributeurs occasionnels qui font généralement parvenir leurs contributions aux développeurs officiels
- Des utilisateurs avancés qui participent activement aux forums, listes de diffusion, canaux IRC
- Des utilisateurs de base qui ont à leur disposition, outre le logiciel, une documentation et des moyens d'interaction (demande de fonctionnalités, rapport de bogues)
- On note aussi la nécessité de rédacteurs de documentation, de relecteurs, de traducteurs, de webmasters, etc.

On constate que ce modèle de développement nécessite l'utilisation de nombreux outils. Il existe des plateformes mises à la disposition des projets de développement proposant un panel d'outils : hébergement de site Web, hébergement des courriels électroniques, forums, listes de diffusion, outils de développement, miroirs de téléchargements, etc. Parmi ces plateformes, on note SourceForge [18] (plus de 100.000 projets), FreshMeat [19] (plus de 45.000 projets), Savannah [20] (plus de 3.000 projets) ou encore Gna [21], Tuxfamily [22] ou Berlios [23].

En ce qui concerne les outils de développements collaboratifs, on note le logiciel très utilisé CVS [24] (mises-à-jour contrôlées, téléchargement souvent public), mais aussi des outils plus récents tels que SVN [25] et Arch [26] ou encore GIT [27]. Pour la gestion des outils de communication, on note les outils de publication web (SPIP [28], Wiki [29]), les forums (PHPBB [30], Phorum [31]), les gestionnaires de listes de diffusion (Sympa [32], Mailman [33]), les serveurs IRC (Freenode [34], Undernet [35]), etc.

La connaissance de ces outils permet de mieux appréhender la diversité des membres de la communauté du logiciel libre. Des règles sont implicites aux utilisateurs finaux, à savoir lire la documentation officielle, les FAQ (Foires Aux Questions) ainsi que les archives des forums ou listes de diffusion avant d'utiliser les outils de communication. On se référera aux documents "Les règles de la Netiquette" [36], "Comment poser les questions de manière intelligente" [37], "Comment signaler efficacement un bug" [38] ou encore "Comment faire un rapport sans se faire lyncher" [39]. Il faut bien noter l'organisation souvent pyramidale du modèle de développement. Ainsi, même s'il est souvent possible de contacter directement les responsables d'un projet par

18. http://www.sourceforge.net/
19. http://freshmeat.net/
20. http://savannah.gnu.org/
21. https://gna.org/
22. http://www.tuxfamily.org/
23. http://www.berlios.de/
24. https://www.cvshome.org/
25. http://subversion.tigris.org/
26. http://www.gnu.org/software/gnu-arch/
27. http://git.or.cz/
28. http://www.spip.net/
29. http://en.wikipedia.org/wiki/Wiki
30. http://www.phpbb.com/
31. http://phorum.org/
32. http://www.sympa.org/
33. http://www.gnu.org/software/mailman/
34. http://freenode.net/
35. http://www.undernet.org/
36. http://www.sri.ucl.ac.be/SRI/rfc1855.fr.html
37. http://www.gnurou.org/documents/smart-questions-fr.html
38. http://www.chiark.greenend.org.uk/~sgtatham/bugs-fr.html
39. http://www.asktog.com/columns/047HowToWriteAReport.html

courrier électronique, cela se fera avec réserve et uniquement pour des questions majeures. Il semble aussi entendu que pour obtenir des droits ou des responsabilités dans un projet, les échelons se gravissent petit à petit.

Les logiciels en eux-mêmes, produits finaux des équipes de développement, sont souvent disponibles sous plusieurs formes. Généralement, on distingue une version dite stable et une version en cours de développement. La version stable est une version ayant subi plusieurs phases de tests et corrections. C'est cette version que le projet propose d'installer aux utilisateurs de base et c'est encore plus vrai en environnement professionnel.
La version en cours de développement (parfois accessible à partir des outils de développements collaboratifs comme CVS) est à réserver aux développeurs, aux utilisateurs désirant contribuer en rapportant les erreurs, ou bien aux impatients.
Il faut noter que les contraintes imposées à une version stable diffèrent d'un projet à un autre. Ce sera également le cas avec les numéros de version, qui n'ont plus grande signification du fait des politiques de numérotation différentes entre les projets. À l'exception des mentions "Version Beta", "Realease Candidate"... qui signifient qu'il s'agit de versions en cours de correction et validation avant une sortie officiellement stable. Sans que cela soit une règle absolue, on peut néanmoins distinguer certaines règles communes, voir par exemple une tentative de formalisation de certaines conventions [40].

Les utilisateurs de logiciels libres prennent donc part au modèle de développement grâce au support communautaire qui leur permet non seulement de trouver de l'aide mais de rapporter les erreurs éventuellement rencontrées et de demander l'ajout de nouvelles fonctionnalités. Parmi les utilisateurs de logiciels, on trouve également de nombreuses structures professionnelles. Ainsi, les contraintes engendrées par l'utilisation dans un environnement professionnel ont nécessité la création de support commercial pour certains logiciels libres. De même, on constate l'apparition de structures spécialisées dans le support de solutions libres.
En France, il existe ainsi de nombreuses SS2L (Société de Service en Logiciels Libres) réparties sur le territoire. D'une manière générale, l'utilisation des logiciels libres dans le monde professionnel est un phénomène en vogue actuellement ; il en résulte souvent de nombreux avantages (fiabilité accrue, correction de bogues, etc.) pour les logiciels libres concernés.

Liens :
http://www.gnu.org/prep/SERVICE
http://www.linux-france.org/article/pro/annuaire/

40. http://semver.org/

Chapitre 2

Systèmes GNU/Linux

2.1 Présentation de Linux

Linux est un noyau de système d'exploitation de type UNIX créé par Linus Torvalds et de nombreux développeurs.

Tout ordinateur inclue un ensemble basique de programmes appelé système d'exploitation. Le programme le plus important d'un système d'exploitation est appelé le noyau : il est chargé dans la mémoire physique (RAM) quand le système démarre et contient les instructions nécessaires à l'ordinateur pour fonctionner. Les autres programmes permettent d'interagir avec l'ordinateur mais ils sont moins importants car les possibilités d'interaction avec l'ordinateur sont déterminées par le noyau.

Le 5 octobre 1991, Linus Torvalds, un informaticien finlandais, annonce sur un forum Usenet la disponibilité du système d'exploitation Linux, inspiré de Minix.

```
From: Linus Benedict Torvalds (torvalds@klaava.Helsinki.FI)
Subject: Free minix-like kernel sources for 386-AT
Newsgroups: comp.os.minix
Date: 1991-10-05 08:53:28 PST

Do you pine for the nice days of minix-1.1, when men were men and wrote
their own device drivers? Are you without a nice project and just dying
to cut your teeth on a OS you can try to modify for your needs? Are you
finding it frustrating when everything works on minix? No more all-
nighters to get a nifty program working? Then this post might be just
for you

As I mentioned a month ago, I'm working on a free version of a
minix-lookalike for AT-386 computers. It has finally reached the stage
where it's even usable (though may not be depending on what you want),
and I am willing to put out the sources for wider distribution. It is
just version 0.02 (+1 (very small) patch already), but I've successfully
run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Sources for this pet project of mine can be found at nic.funet.fi
(128.214.6.100) in the directory /pub/OS/Linux. The directory also
contains some README-file and a couple of binaries to work under linux
```

```
(bash, update and gcc, what more can you ask for . Full kernel
source is provided, as no minix code has been used. Library sources are
only partially free, so that cannot be distributed currently. The
system is able to compile "as-is" and has been known to work. Heh.
Sources to the binaries (bash and gcc) can be found at the same place in
/pub/gnu.

[...]

I can (well, almost) hear you asking yourselves "why?". Hurd will be
out in a year (or two, or next month, who knows), and I've already got
minix. This is a program for hackers by a hacker. I've enjouyed doing
it, and somebody might enjoy looking at it and even modifying it for
their own needs. It is still small enough to understand, use and
modify, and I'm looking forward to any comments you might have.

I'm also interested in hearing from anybody who has written any of the
utilities/library functions for minix. If your efforts are freely
distributable (under copyright or even public domain), I'd like to hear
from you, so I can add them to the system. I'm using Earl Chews estdio
right now (thanks for a nice and working system Earl), and similar works
will be very wellcome. Your (C)'s will of course be left intact. Drop me
a line if you are willing to let me use your code.

Linus
```

Linux est multi-tâches, multi-utilisateurs et compatible Unix (il respecte les normes POSIX). Conçu au départ pour les machines de type Intel x86, Linux est maintenant disponible pour les architectures PowerPC, Alpha, MIPS, Sparc, etc.

Les sources de Linux sont disponibles (sous licence GPL) et l'explosion d'Internet a permis un mode de développement innovant : de nombreux développeurs (bénévoles ou rémunérés par des entreprises) participent au développement de Linux et forment avec tous les utilisateurs une communauté. Linux est d'ailleurs l'un des exemples les plus connus de logiciel libre.

L'un des objectifs du projet GNU est d'avoir un système d'exploitation complètement libre. Or, de très nombreux outils ont été développés par le projet GNU (GCC, Bash, etc.) mais Hurd, noyau développé par le projet GNU, tardant à sortir, Linux a été adopté par le projet GNU en 1993 pour être le noyau du système prôné par le projet GNU. L'ensemble formé par les outils du projet GNU et du noyau Linux est souvent appelé système GNU/Linux.

Une distribution est un système GNU/Linux avec un certain de nombre de choix et d'outils mis à disposition pour gérer au mieux les logiciels et leurs configurations. Il existe un grand nombre de distributions adaptées à un usage (ou un matériel) spécifique. Les distributions généralistes les plus connues sont Red Hat, Debian, Mandriva, SuSe, Gentoo, Slackware, Fedora.

2.2 Méthode d'installation

Il existe diverses méthodes d'installation :

Dans les années 1990, on installait un système GNU/Linux à l'aide de (nombreuses) disquettes. Aujourd'hui la méthode d'installation la plus répandue est d'utiliser un jeu de CD-ROM ou DVD-ROM (ou plus récemment via un périphérique USB) téléchargé sur Internet par HTTP ou FTP. D'autres méthodes existent selon les distributions et peuvent s'avérer très pratiques notamment des installations amorcées par le réseau.

En cas de problème à l'installation d'un système GNU/Linux, voici quelques suggestions :

- Prendre garde à la fiabilité des périphériques d'amorçage (CD-ROM, DVD-ROM, périphérique USB) en vérifiant systématiquement l'empreinte MD5 ou SHA1
- Prendre connaissance des paramètres spécifiques d'amorçage du noyau Linux permettant d'éviter certains dysfonctionnements.
- Pour les controleurs RAID ou cartes réseau exotiques (ou récentes), il faudra peut-être charger un module spécifique lors de l'installation.
- Vérifier la compatibilité du matériel. Il existe plusieurs sites dont le "HardWare Howto" [1] ou encore le "Debian GNU/Linux device driver check page" [2]
- Utiliser les moyens de support communautaire tels que les listes de diffusion ou les canaux IRC. En cas de bogue (ou pas) avec l'installation de Debian, il faut rapporter le bogue pour le pseudo-paquet "installation-reports" [3]
- Un support professionnel peut également être trouvé auprès de sociétés de services en logiciels libres... comme Evolix ;-)

Liens :
http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html
http://people.debian.org/~blade/install/preload/index.old.html
http://www.gnu.org/prep/service.html

2.3 Systèmes de fichiers

Il existe une norme appelée Filesystem Hierarchy Standard [4] à laquelle se conforment certaines distributions. Rappelons la hiérarchie du système de fichiers (cela varie d'une distribution à une autre) d'un système GNU/Linux :

1. http://www.tldp.org/HOWTO/Hardware-HOWTO/
2. http://kmuto.jp/debian/hcl/
3. http://www.debian.org/devel/debian-installer/report-template
4. http://www.pathname.com/fhs/

Arborescence	Contenu
bin	Binaires (exécutables) des commandes essentielles
boot	Fichiers statiques pour le chargeur d'amorçage (boot)
dev	Fichiers des pilotes de périphériques
etc	Configuration système propre à la machine
home	Répertoires personnels des utilisateurs
lib	Bibliothèques partagées et modules noyaux essentiels
mnt,media	Points de montage pour les montages temporaires
proc,sys	Répertoire virtuel pour les informations systèmes
root	Répertoire personnel de l'utilisateur root
sbin	Exécutables système essentiels
tmp	Fichiers temporaires
usr	Hiérarchie secondaire
var	Données variables
opt	Suites applicatives additionnelles
srv	Données pour les services

2.4 Partitionnement

Conseils de partitionnement

Pour une machine de type serveur, on appliquera un partitionnement [5] plus réfléchi qu'un poste de travail (où l'on se contente souvent d'isoler la partition contenant le répertoire home, la partition swap et le système). Il n'existe pas de vérité absolue (cela varie selon l'utilité de la machine et les habitudes des administrateurs) mais on passera en revue certaines recommandations. Voici quelques détails sur la taille des partitions destinées à accueillir certains répertoires :

- /boot/ : partition d'environ 100 Mo
- / : partition supérieure à 100 Mo (conseil : 500 Mo)
- /tmp : quelques centaines de Mo ou davantage dans des cas particuliers
- /var : partition supérieure à 250 Mo (conseil : plusieurs Go si possible)
- /usr : partition supérieure à 500 Mo (conseil : quelques Go)
- /home : à voir selon les quotas et le nombre d'utilisateurs
- swap : partition supérieure à 16 Mo et inférieure à 2 Go (systèmes 32-bits). On conseille souvent de mettre la même taille (ou le double) de la mémoire physique (sauf pour des applications spécifiques comme Oracle par exemple ou dans les cas de taille très importante de la mémoire physique) et si possible proche du centre du disque.
  D'autres espaces spécifiques notamment pour l'espace web, la base de données pourront être créés selon les utilisations.

Exemple de partitionnement pour un disque d'environ 40 Go :

5. http://www.tldp.org/HOWTO/Partition/

Partition	Rep.montage	Taille
sda1	/boot	100 Mo
sda3	/	500 Mo
sda4	/usr	3 Go
sda6	/var	5 Go
sda7	/tmp	500 Mo
sda8	/mnt/ftp	1 Go
sda9	swap	500 Mo
sda10	/home	20 Go

Programmes de partitionnement

fdisk
Programme de partitionnement en ligne de commande très répandu. Lire la page de manuel pour en savoir plus.
Lien : http://evolix.org/man/fdisk.html

cfdisk
Programme de partitionnement graphique en console. Il est utilisé par défaut lors de l'installation sous Debian Woody. Son utilisation est plutôt simple et intuitive.
Lien : http://evolix.org/man/cfdisk.html

sfdisk
Programme de partitionnement en ligne de commande. sfdisk a quatre utilisations principales : liste la taille d'une partition, liste les partitions d'un périphérique, vérifier une partition sur un périphérique, et repartitionner un périphérique.
Lien : http://evolix.org/man/sfdisk.html

parted
Programme de partitionnement en ligne de commande. GNU Parted est notamment indispensable pour gérer des partitions sur des volumes d'une taille importante (supérieure à 2 To) en permettant la gestion des tables de partitions GPT
Lien : http://www.gnu.org/software/parted/

2.5 Gestion des disques

Systèmes multi-disques
Lien : http://tldp.org/HOWTO/Multi-Disk-HOWTO.html

LVM

LVM (Logical Volume Manager) permet une gestion plus aisée que l'utilisation classique disques et partitions. Cela permet une plus grande flexibilité pour redimensionner les volumes en fonction des besoins et des nouveaux disques disponibles.
Lien : http://www.tldp.org/HOWTO/LVM-HOWTO/

RAID

Le RAID (Redundant Arrays of Inexpensive Disks) est une solution bien connue pour obtenir des disques redondants afin de sécuriser les machines demandant une qualité de service élevée. On parlera bien sûr ici de RAID logiciel.
Lien : http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html

Note : Nous recommandons d'utiliser le RAID logiciel et/ou LVM avec prudence pour les répertoires sensibles tels que /boot et le répertoire racine. Une solution de RAID matériel sera à privilégier bien que plus coûteuse.

Choix du système de fichiers journalisé : ext3/ext4, ReiserFS, XFS ou JFS

Il existe de nombreuses comparaisons entre ces systèmes de fichiers mais il est difficile d'en tirer des conclusions générales car cela dépend beaucoup de l'utilisation de la machine. La fiabilité de ces systèmes est désormais éprouvée même si la prudence naturelle des administrateurs poussera à conserver ext3[6], dérivé du système de fichier historique (Extended File System a été implémenté en 1992 et intégré à Linux 0.96c). En terme de performances, il semble se dégager quelques constations comme la performance de ReiserFS[7] pour les petits fichiers, un léger gain de XFS[8] pour la copie de fichiers de grande taille mais une lenteur à l'effacement et la bonne tenue d'ext3 pour des opérations classiques. JFS[9] semblant légèrement moins performant pour le moment.

Sachant la perpétuelle évolution des développements, les conditions dans lesquelles se déroulent les tests et les autres paramètres (processeur, mémoire...) on se gardera bien de tirer des conclusions définitives. Disons simplement que l'utilisation de ext3 est encore assez raisonnable pour des serveurs classiques. En ce qui concerne ext4, il tend à remplacer ext3 et l'on peut commencer à l'utiliser en production car il offre parfois de meilleures performances.

Liens :
http://www.linux-france.org/article/sys/ext3fs/Benchmarks/
http://fsbench.netnation.com/

2.6 Packages

Au fil des années, les distributions Linux se sont vu dotées de systèmes de packaging multiples, et sont devenus des moyens très utilisés permettant d'installer des logiciels plus rapidement et plus facilement qu'avec une compilation classique :

- le RPM initialement développé par Redhat
- les packages Slackware
- les packages Debian (.deb)

Pour plus d'informations sur les RPMs, vous pouvez consulter http://www.rpm.org/ et http://www.lilit.be/formations/systeme_fichier/node5.html.

6. http://www.zip.com.au/~akpm/linux/ext3/
7. http://www.namesys.com/
8. http://oss.sgi.com/projects/xfs/
9. http://www-124.ibm.com/developerworks/oss/jfs/

2.7 Configuration réseau

Au niveau des cartes réseau, il est désormais assez rare de rencontrer des problèmes de pilotes. Une fois les cartes réseau correctement détectées, on procédera à la configuration du réseau. Si un serveur DHCP est présent sur le réseau, la configuration est automatique. On préférera néanmoins une configuration statique pour des raisons de sécurité. Plus généralement la configuration réseau se définit souvent dans un fichier spécifique selon les distributions. Voici un exemple d'un fichier `/etc/network/interfaces` pour Debian :

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.12.67
netmask 255.255.255.0
gateway 192.168.12.254
```

En ce qui concerne la configuration DNS, c'est dans le fichier `/etc/resolv.conf` qu'on trouvera les adresses des serveurs de nom. On utilisera les outils `host` et `dig` pour s'assurer du bon fonctionnement. Voici un exemple de fichier :

```
search domain.tld
nameserver 192.168.12.71
nameserver 62.4.17.69
```

Les détails de la configuration réseau et DNS ne sont pas abordés ici, on se référera à de nombreuses documentations disponibles sur Internet ou dans les bonnes bibliothèques. Passons tout de même en revue quelques outils indipensables à tout administrateur réseau :

ifconfig permet de configurer les interfaces réseau
Exemple : `ifconfig eth0 192.168.13.47`
`ifconfig eth0:0 1.2.3.4 netmask 255.255.0.0`

route gestion de la table de routage
Exemple : `route add -net 192.168.100.10/24 gw 192.168.100.1`

ip gestion avancée de la configuration réseau (routage, périphériques, etc.)
Exemple : `ip addr add 10.0.0.1/24 dev eth0 label eth01`

netstat informations avancées sur l'état réseau
Exemple : `netstat -taupen`

traceroute pour tester la route vers un hôte du réseau
Exemple : `traceroute -v google.fr`

mtr un traceroute plus convivial Exemple : `mtr google.fr`

tcpdump l'outil ultime pour capturer les trames réseau

Exemple : `tcpdump -s2000 -XX -w mydump.pcap -i eth0 port 80 and tcp`

2.8 Réglages de base

Revenons sur quelques réglages de base :

Boot loader

Selon les distributions, le boot loader sera LILO ou Grub. Grub est plus flexible que LILO mais chacun possède des fonctionnalités différentes qui peuvent s'avérer utiles dans certains cas particuliers. Notons que sur un serveur en production, le choix du Boot loader n'est pas essentiel.

Réglage des locales

Les locales sont l'environnement de localisation du système. Cet environnement est utilisé par les programmes pour reconnaître la langue et le jeu de caractères que votre système utilise. On peut choisir ses locales grâce au paquet locales. Actuellement, un choix s'offre entre le codage en ISO ou en UTF-8 (Unicode sur 8 bits).
Bien que de plus en plus d'applications soient compatibles avec le codage unicode, les utilisateurs hésitent souvent à basculer en UTF-8. Pour un serveur, l'importance des locales est assez limitée. Si l'on pourrait rester en ISO-8859-1 ou ISO-8859-15 (extension du codage européen avec le support de l'euro), il n'est désormais plus imprudent d'utiliser de l'UTF-8.
En ce qui concerne le choix de la langue, il peut être judicieux d'utiliser une locale de langue anglaise pour obtenir des messages d'erreur en anglais, ce qui facilite les recherches Internet par exemple. Ainsi, l'utilisation la plus raisonnable semble la locale en_us.UTF-8 et l'ajout des locales françaises (fr_FR et fr_FR.UTF-8) est conseillée (pour PHP ou PostgreSQL par exemple).

Réglage du clavier

On pourra utiliser différents claviers. On chargera la table de traduction en utilisant la commande loadkeys. Les tables disponibles sont contenues dans le répertoire /usr/share/keymaps

Exemples pour passer en azerty ou en qwerty sur certaines distributions :

```
# loadkeys fr-latin0
# loadkeys us-latin1
```

Le programme kbdconfig peut aussi proposer une interface pour choisir son type de clavier.

Système de messagerie

Par défaut, un système du type Unix a besoin d'un système de messagerie, ne serait-ce que pour envoyer des alertes système à l'administrateur local. Si l'on n'a pas besoin de serveur de messagerie, on laissera donc un système de mail avec une configuration locale uniquement.

Mise à l'heure

Il est souvent essentiel pour un serveur d'être à l'heure. Pour cela le protocole NTP (Network Time Protocol) [10] permet de se synchroniser sur un serveur de temps. Si l'on possède plusieurs serveurs, il est intéressant d'avoir un serveur NTP [11] sur lequel les autres machines locales vont se synchroniser. La synchronisation du serveur de temps ou du serveur isolé se fera sur plusieurs serveurs de temps officiels (certains sont en libre accès comme `ntp.tuxfamily.net` ou `swisstime.ee.ethz.ch`, sinon il faut demander un accès en envoyant un courrier électronique).

Concrètement, on peut installer le paquet *ntpdate* et lancer une synchronisation horaire avec la commande suivante :

```
ntpdate ntp2.evolix.net
```

Sur un serveur, on utilisera plutôt le paquet `ntp` permettant une synchronisation permanente avec un ou plusieurs serveurs NTP.

Lien : http://www.cru.fr/NTP/serveurs_francais.html

2.9 Présentation des principales distributions

2.9.1 Red Hat

http://www.redhat.com/
http://www.europe.redhat.com/documentation/rhl9/rhl-ig-x86-fr-9/
http://www.com.univ-mrs.fr/ssc/info/linux/install_linux.html

Red Hat est la plus importante société avec une activité dédiée aux logiciels Open Source.
Fondée en 1993, cette société basée aux États-Unis propose le déploiement d'infrastructures réseau en se basant sur sa distribution : Red Hat Linux. Entrée en bourse en 1999, Red Hat a changé de stratégie en 2003 en se consacrant uniquement au monde professionnel : Red Hat propose désormais une gamme de distributions payantes (Red Hat Entreprise) et se contente de sponsoriser un dérivé de Red Hat : le projet Fedora. La société compte aujourd'hui plus de 700 salariés répartis dans plus de 20 pays.

Les versions :
- Redhat version 9, plus supportée, plus mise à jour
- Redhat Entreprise Advanced Server
- Redhat Entreprise Server
- Redhat Entreprise Linux WorkStation

Depuis quelques mois, Redhat ne fournit plus directement de versions gratuites. Cette tâche incombe désormais à un groupe de développeurs annexes qui maintient la distribution Fedora. Fedora est une version gratuite de Redhat, qui permet également aux développeurs Redhat de

10. ftp://ftp.rfc-editor.org/in-notes/rfc2030.txt
11. http://www.ntp.org/

"tester" certaines évolutions à grande échelle avant que celles-ci soient intégrées.

L'installation de logiciels peut être réalisée de plusieurs façons :
– via les sources, en les compilant
– via un RPM
– via *yum*, un système qui ressemble fortement à apt pour les utilisateurs de Debian

Les mises à jour sont assurées sur Fedora par *yum*, ainsi que part *uptodate* comme c'est le cas sur Redhat (utilisation texte ou graphique).

La sécurité pour la distribution Redhat est gérée via la page https://www.redhat.com/security/updates/, où toutes les mises à jour sont disponibles. Des RPMs sont disponibles.

2.9.2 Mandriva

http://www.mandrakelinux.com/
http://www.mandrivalinux.com/fr/fdoc.php3
http://www.mandrivalinux.com/fr/ftp.php3
http://fr.wikipedia.org/wiki/Mandriva_Linux

Mandriva est le produit d'une entreprise française, MandrakeSoft, créée en 1998. Grâce à une croissance rapide et de nombreux soutiens, la société MandrakeSoft compte plus de 120 salariés et est cotée en bourse. Elle impose sa distribution comme l'une des plus répandues dans le monde. Accessible dans plus de 50 langues, Mandriva est une distribution conviviale et accessible aux débutants. Initialement appelée Mandrake, elle fût rebaptisée Mandriva en 2005 (suite à des problèmes judiciaires concernant le nom et au mariage de MandrakeSoft avec Conectiva, une distribution Linux venant d'Amérique Latine).

Les versions actuelles :
Il existe de très nombreuses versions et Quelques principes :
Les versions "Free" ne contiennent que des logiciels libres, et sont téléchargeables gratuitement.
Les versions "Entreprise" sont, bien qu'Open Source, payantes et bénéficient d'un support spécial par Mandriva. La plupart permettent une inter-opérabilité entre les plates-formes Linux, Windows et Mac.

Quelques exemples :

– Mandriva Free : C'est tout un système complet livré avec tous les programmes qui peut être téléchargé gratuitement et librement distribuable, et gravé sur un DVD-ROM ou sur 3 CD-ROM.
– Mandriva Powerpack : La version en boîte (Powerpack) est payante et fournie avec des manuels imprimés, pour aider les utilisateurs à découvrir le système. Elle inclut également le droit à des services, comme de l'assistance technique.
– Mandriva One : version "Live CD" utilisable sans rien installer
– Mandriva Flash : version portable autobootable vendue sur une clé USB
– Multi Network Firewall (MNF) : version spéciale pour les machines de type routeur/firewall

Pour les problèmes de sécurité, Mandriva regroupe les informations sur http://www.mandriva.

com/security/, et les utilisateurs bénéficient d'une grande réactivité. Ainsi, une liste de diffusion et un flux RSS sont disponibles pour se tenir informé.

2.9.3 Gentoo

http://www.gentoo.org/
http://www.gentoo.org/doc/fr/handbook/index.xml
http://www.gentoo.org/doc/fr/gentoo-x86-quickinstall.xml

Gentoo est une distribution "orientée source". A l'image des systèmes BSD dont elle dérive, l'utilisateur compile son système et ses logiciels. Cependant, il existe également des paquetages binaires disponibles, afin d'éviter cette phase de compilation.

Gentoo a été originairement conçu pour fonctionner sur architecture x86 uniquement. Mais, elle a été portée sur de nombreuses autres architectures en raison de sa haute portabilité. L'installation se fait à partir d'un CD Live et une documentation d'installation et d'utilisation très complète est disponible. Notez qu'il existe un outil appelé Catalyst permettant de faire ses propres distributions basées sur Gentoo.

La distribution est très orientée compilation avec un système de "flags" interne, qui permettent d'utiliser certaines caractéristiques majeures, comme X (le serveur graphique), une version précise du compilateur, ou encore le support de tel ou tel protocole dans tous les logiciels installés. La compilation est omniprésente et des outils comme DistCC sont très utilisés afin d'en réduire les coûts. Cette distribution est basée sur le langage Python, utilisé dans la plupart des scripts inhérents à la distribution (démarrage, installation du réseau etc.)

Vous pouvez prendre connaissance des failles de sécurité et des éventuels problèmes apparaissant sur la distribution depuis http://www.gentoo.org/security/en/index.xml. La distribution est également axée sur la sécurité avec un certain nombre de mécanismes intégrés par défaut, et des ouvrages relatifs à la sécurité qui ont été réalisés par l'équipe (Gentoo Security Handbook). Les utilisateurs peuvent également prendre compte des changements oeuvrés par mail (gentoo-announce@gentoo.org).

2.9.4 Slackware

http://www.slackware.org/
http://www.trustonme.net/didactels/91.html

Cette distribution historique et simple a été créée, et est toujours maintenue par Patrick Volkerding (depuis 1993). Elle ne repose que sur peu d'outils de configuration automatisés, les outils traditionnels étant privilégiés. Les deux objectifs principaux sont la simplicité d'utilisation et la rapidité.

Le système de packages est géré grâce aux *slacktools*, remplaçants des *pkgtools* de Slackware. Ils ont pu bénéficier constamment d'améliorations, comme la gestion des dépendances etc. Le système de packaging RPM peut également être utilisé facilement afin d'utiliser des packages plus généralistes.

Contrairement à des distributions comme Debian, la compilation du noyau se fait de manière traditionnelle. Rappel :

```
# make menuconfig
# make dep
# make bzImage
# make modules
# make modules_install
```

Cependant, même si la distribution est relativement à jour, les personnes qui y contribuent sont peu nombreuses, et certaines améliorations peuvent mettre du temps à être incorporées. Le projet Slackware a pour but de rester le plus Unix-like possible.

En cas de problème de sécurité, une liste de diffusion dédiée est accessible sur http://www.slackware.org/security/. Les mises à jour sont faites sous forme de packages à installer.

2.9.5 Suse

http://www.novell.com/linux/suse/
http://www.novell.com/documentation/suse93/pdfdoc/user93-screen/user93-screen.pdf

Suse est une société allemande qui appartient désormais au groupe Novell. La distribution du même nom est basée sur Slackware et le système de paquets RPM de Redhat. La configuration peut être facilement réalisée par le Centre de Control YAST, qui est passé, depuis plus d'un an sous licence GPL.

Vu l'orientation de Novell sur le monde des réseaux, le rachat de Suse a marqué l'intégration de nombreux protocoles réseau etc.

Novell propose actuellement plusieurs versions :

- SUSE LINUX Professional : version professionnelle principalement mise en avant par Novell
- Novell Linux Desktop : version orientée poste de travail
- SUSE LINUX Enterprise Server : version spéciale pour les entreprises (certifiée par exemple pour les machines IBM Xseries ou le logiciel Oracle)

Novell propose également de nombreux produits ou services basés sur Linux. On notera des sortes de packages comprenant une distribution Linux et des services :

- Novell Linux Small Business Suite : basée sur Novell Linux Desktop
- Novell Open Enterprise Server : basée sur SUSE LINUX Enterprise Server

Pour les problèmes de sécurité, Suse dispose d'un service en ligne
(à l'adresse http://www.novell.com/linux/security/securitysupport.html).
Une liste des failles ayant affecté Suse est disponible à la même adresse de façon textuelle, ou en flux RSS à l'adresse (http://www.novell.com/linux/security/suse_security.xml).

2.9.6 Debian

Le projet Debian a été lancé en août 1993 par Ian Murdock. Debian est alors une nouvelle distribution produite de façon ouverte, dans l'esprit de Linux et de GNU. Debian a la réputation d'être soigneusement et consciencieusement mise en place, maintenue et supportée.

Cela a commencé par la constitution d'un petit groupe très soudé de hackers (codeurs) de logiciels libres. Graduellement, le groupe s'est agrandi pour devenir une vaste communauté de développeurs et d'utilisateurs bien organisée [12]. Le projet Debian est basé sur un contrat social [13], des directives définissant les logiciels libres, DFSG - Debian Free Software Guidelines [14] et une constitution [15] définissant l'organisation du projet. Organisés de façon démocratique, le leader, le comité technique et le secrétaire ainsi que chaque contributeur Debian possèdent un rôle bien défini (responsable de paquets, de projets, etc.). Ainsi Debian n'est pas soumis à des exigences commerciales et se permet d'avoir un avenir serein. Aujourd'hui, on compte près d'un millier de développeurs Debian répartis dans le monde entier [16], des dizaines de milliers de paquets [17], le support de plusieurs architectures [18] ainsi que des versions utilisant un noyau différent de Linux (GNU/Hurd ou FreeBSD).

Lien : http://www.debian.org/doc/manuals/project-history/

Les versions :

Debian propose trois ou quatre versions de distributions aux utilisateurs.
La distribution "stable" contient la dernière distribution officiellement sortie de Debian. Il s'agit de la dernière production de Debian qu'il est recommandé d'utiliser.
La distribution "testing" contient des paquets en attente d'entrée dans la distribution "stable". Elle contient donc des versions assez récentes des logiciels mais moins bien surveillés par les développeurs Debian.
En pratique, la distribution "testing" passe par une période de gel (freeze) où son évolution est figée et seules les corrections de bogues ou mises-à-jours mineures sont acceptées (on parle parfois de distribution "frozen"). Après de minutieuses vérifications, la version "testing gelée" peut remplacer la version stable.

La troisième distribution est appelée "unstable". C'est celle sur laquelle se concentre les activités de développement. Elle est utilisée par "les développeurs et par ceux qui aiment vivre sur le fil" car c'est une sorte de laboratoire destiné à tester et corriger les bogues des paquets. Son nom Sid (nom de l'enfant qui casse les jouets dans *Toys Story*) signifie officieusement : Still In Development.

Les nouveaux paquets des développeurs arrivent en effet directement en unstable après inspection et validation des ftpmasters (les personnes responsables des mises en ligne sur le serveur FTP principal). Chaque mise en ligne est répercutée sur les miroirs en moins de 24h (on peut retouver les nouveaux paquets des développeurs directement dans le répertoire Incoming [19]. Les paquets de la version "unstable" remplissant certains critères (absence de bogues critiques, durée supérieure à 10 jours dans "unstable", support de toutes les architectures et dépendances toutes satisfaites) basculent dans la distribution "testing".

Dernière distribution plus particulière, il s'agit de la version "experimental" : il s'agit d'une archive mise à la disposition par des développeurs pour avoir des retours d'utilisateurs expérimentés sur des modifications importantes des logiciels (la version "experimental" n'est pas une

12. http://www.debian.org/intro/organization
13. http://www.debian.org/social_contract.fr.html
14. http://www.debian.org/social_contract#guidelines
15. http://www.debian.org/devel/constitution
16. http://www.debian.org/devel/developers.loc
17. http://www.debian.org/distrib/packages
18. http://www.debian.org/ports/
19. http://incoming.debian.org/

distribution complète). Cette étape est indépendante du processus de validation des paquets Debian.

On se rend donc compte des précautions qui entourent le statut des paquets Debian. En contrepartie, ce processus prend du temps et les versions de paquets de la distribution "stable" ont souvent du retard par rapport aux dernières versions des logiciels.

Voici le tableau des noms de distribution Debian (tirés du Film Toy Story) :

-	0.01->1.1	août 1993->juin 1996	-
Buzz	1.1	juin 1996	Le ranger de l'espace
Rex	1.2	décembre 1996	le tyrannosaure
Bo	1.3	juillet 1997	La bergère
Hamm	2.0	juillet 1998	Le cochon-tirelire
Slink	2.1	mars 1999	Le chien à ressort
Potato	2.2	août 2000	Monsieur Patate
Woody	3.0	mai 2002	Le cow-boy
Sarge	3.1	juin 2005	Le chef des soldats
Etch	4.0	avril 2007	L'écran magique
Lenny	5.0	mars 2009	La paire de jumelles
Squeeze	6.0	février 2011	L'extraterrestre à trois yeux
Wheezy	7.0	?	Le pingouin au noeud papillon
Sid	-	toujours unstable	L'enfant qui casse les jouets

2.10 Focus sur Debian GNU/Linux

2.10.1 Méthodes d'installation

Il existe donc diverses méthodes d'installation [20] :

- *À partir de CD-ROM (ou DVD-ROM)* [21]
  Pour obtenir un jeu de CD-ROM (ou DVD-ROM) de Debian, on passera par l'un des revendeurs [22] ou par le réseau en téléchargeant des images [23].
  Les moyens de téléchargement sont sur des images complètes par HTTP/FTP sur des mirroirs [24], une construction d'image grâce à l'outil Jigdo [25] qui télécharge séparément tous les fichiers du CD-ROM (ou DVD-ROM) [26] ou encore grâce au Peer-to-peer comme le système BitTorrent [27].

- *À partir des disquettes* [28]
  On peut amorcer une installation de Debian à partir d'un nombre très réduit de disquettes (pratique notamment en cas d'absence de lecteur CD-ROM utilisable). Deux disquettes sont nécessaires pour débuter l'installation (disquettes Root et Rescue) puis quatre autres pour charger le noyau et ses modules.

- *À partir du réseau*
  Suite à l'installation à partir de disquettes ou à partir d'un CD-ROM minimal, on poursuit souvent l'installation par le réseau pour installer le système de base ainsi que les logiciels supplémentaires. En téléchargeant directement les paquets sur internet ou sur un mirroir local, il suffit donc d'une image minimale pour installer Debian GNU/Linux. Néanmoins, il est préférable de ne pas installer directement par internet car il faudrait ne pas connecter un serveur avant de l'avoir totalement sécurisé (et parce que les versions des paquets installés par défaut peuvent comporter des failles de sécurité). On préférera des mirroirs locaux, voir par exemple apt-proxy [29] ou apt-move [30].

  On peut également amorcer complètement l'installation de Debian par le réseau [31] si l'on possède un périphériphérique réseau amorçable.

Remarques sur les possibilités d'installation

Les possibilités d'installation de Debian sont nombreuses. On notera que le programme debootstrap [32] permet d'installer Debian à partir d'un système de fichiers, qu'une installation

20. http://www.debian.org/distrib/
21. http://www.debian.org/CD/
22. http://www.debian.org/CD/vendors/
23. http://www.debian.org/distrib/cd
24. http://www.debian.org/CD/http-ftp/
25. http://atterer.net/jigdo/
26. http://www.debian.org/CD/jigdo-cd/
27. http://www.debian.org/CD/torrent-cd/
28. http://www.debian.org/distrib/floppyinst
29. http://apt-proxy.sourceforge.net/
30. http://ptitlouis.dyndns.org/~ptitlouis/doc/
31. http://www.debianplanet.com/node.php?id=818
32. http://www.debian.org/releases/stable/i386/ch-rescue-boot.fr.html#s-dbootstrap-intro

automatisée est possible en passant l'argument *preseed*[33] à l'amorce de l'installation par CD-ROM (ou DVD-ROM) et que la personnalisation de l'installation est possible.

2.10.2 Installation et réglages de base

La version stable de Debian est toujours celle à privilégier pour de nouvelles installations. Dans certains cas, dans les mois précédents la sortie de la future version stable, celle-ci peut être installé si le serveur n'est pas trop critique. Dans ces périodes charnières, il faut bien prendre en compte qu'il existera toujours des administrateurs qui affirmeront le contraire. Il faut donc essayer de se faire un avis objectif à partir de plusieurs serveurs de test avant toute décision importante.

Voici les recommandations que nous faisons à nos clients :

Pour un serveur en production

Nous conseillons d'utiliser Debian stable, à moins de vraiment nécessiter beaucoup de fonctionnalités présentes dans la prochaine version stable, et dans ce cas la version Debian testing peut être envisagée.

Pour un serveur en semi-production

Dans un environnement non critique et dans l'optique est de préparer une future mise en production, Debian testing peut être utilisée si la version stable date de plus de 12 à 18 mois ; sinon l'utilisation de Debian stable reste à privilégier.

Pour un poste de travail

Nous conseillons l'utilisation de Debian testing (voire unstable) afin d'avoir des versions récentes (mais néanmoins testées) des logiciels de bureautique. Cela permet de passer de façon souple vers la prochaine version stable lors de sa sortie et de l'utiliser pendant quelques mois avant de rebasculer vers Debian testing.

Installation de Debian

Revenons brièvement sur l'installation de Debian gérée par le Debian-Installeur[34] :

- Démarrage, détection des cartes réseau, des paquets udebs (.udeb est un format particulier de paquet pour l'installeur)
- Chargement éventuel de pilotes ou firmware
- Outils de partitionnement, installation de base
- Détection avancée des périphériques grâce à discover[35] ce qui permet de charger automatiquement les modules adéquats
- Installation de GRUB, redémarrage
- Configuration de base

33. http://d-i.alioth.debian.org/manual/fr.i386/ch04s07.html
34. http://www.debian.org/devel/debian-installer/
35. http://d-i.alioth.debian.org/manual/fr.i386/index.html

Après cette installation, on procédera à une mise-à-jour générale (sur un miroir local si possible), une installation d'outils pratiques, une compilation du noyau puis une sécurisation. Voici quelques outils pratiques à installer : ssh vim less mailx metche sudo munin log2mail apt-listchanges apticron evocheck

Voici un récapitulatif de nos préconisations d'installation : http://trac.evolix.net/infogerance/wiki/HowtoDebian/Install

2.10.3 Système de packages Debian

L'installation de nouveaux logiciels Open Source sous environnement Linux peut toujours se faire selon la méthode classique de recompilation des sources. Néanmoins cette méthode est peu aisée pour la gestion des dépendances, des mises-à-jour, etc. Ainsi les distributions utilisent souvent des systèmes de paquetage. Debian utilise des paquets à l'extension `.deb` qui fournissent non seulement les binaires précompilés (le plus souvent) mais également des méthodes de gestion pour faciliter la manipulation de ces paquets.

L'utilitaire basique de manipulation des paquets Debian porte le nom de `dpkg`. Il est important de bien maîtriser les différentes options de `dpkg` :

`dpkg --unpack` :

le paquet est dépaqueté mais n'est pas configuré

`dpkg --configure`

configuration d'un paquet

`dpkg -i`

installation complète

`dpkg -r`

supression du paquet

`dpkg -r -P`

suppression complète (fichiers de configuration compris)

`dpkg -L`

affiche la liste des fichiers appartenant au paquet

`dpkg -S`

recherche un fichier dans les paquets installés

`dpkg -l`

liste tous les paquets installés sur le système.

`dpkg --get-selections / --set-selections`

Donne/installe la/une liste des paquets installés
Debian possède un programme de gestion avancée de paquets appelé APT (Advanced Packaging Tool).
Ce système gère les paquets d'après une liste (Packages.gz) et permet de gérer les dépendances, les mises-à-jour particulières ou globales ou encore les conflits. Le fichier qui référence toutes les sources de paquets disponibles est `/etc/apt/sources.list`. On peut gérer le contenu de ce fichier à l'aide de la fonction `apt-setup`.

Exemple de fichier `/etc/apt/sources.list` pour Debian Squeeze :

```
deb http://security.debian.org/ squeeze/updates main contrib non-free
deb http://mirror.evolix.org/debian squeeze main contrib non-free
deb-src http://mirror.evolix.org/debian squeeze main contrib non-free
```

`dselect` est l'outil de gestion des paquets historique (dpkg a une dépendance envers dselect !). Son utilisation n'est pas forcément aisée pour les novices.

Lien : http://www.debian.org/releases/woody/i386/dselect-beginner

La commande `aptitude` est une interface pour APT. Voici quelques commandes souvent utilisées :

`aptitude update`
Cette commande resynchronise les informations sur les fichiers disponibles à partir des endroits spécifiés dans le `sources.list`. Cette commande récupère donc les fichiers Packages.gz et les analyse de manière à rendre disponibles les informations concernant les nouveaux paquets.

`aptitude safe-upgrade`
Cette commande met à jour tous les paquets dont une version plus récente est disponible sans supprimer de paquets, ni en ajouter.

`aptitude full-upgrade`
Cette commande met à jour les paquets (comme upgrade) en utilisant une gestion intelligente des changements de dépendances. Ainsi des paquets pourront être supprimés et des nouveaux paquets pourront être installés.

`aptitude install <paquet>`
Cette commande installe le paquet <paquet> ainsi que toutes ses dépendances nécessaires.

`apt-get remove --purge <paquet>`

Cette commande supprime le paquet <paquet> ainsi que tous ceux qui en dépendent.

`aptitude clean`
Cette commande supprime les paquets installés du répertoire de cache d'APT (libère de l'espace disque). On peut également utiliser autoclean qui va supprimer uniquement les paquets les plus anciens et inutiles.

D'autres interfaces utilisent APT :

apt-get C'est l'ancien outil qui est petit à petit remplacé par `aptitude` : toujours utilisable, il est notamment toujours très partique pour les commandes `apt-get source` ou `apt-get -f install`.

synaptic Outil graphique de gestion des paquets en GTK+ basé sur APT. Il permet d'utiliser la plupart des fonctions (recherche, installation, mise-à-jour, etc.)

Enfin de nombreux outils supplémentaires existent pour gérer les paquets :

apt-cache utilitaire permettant d'obtenir un certain nombre d'informations sur les paquets et le cache d'APT.

apt-file utilitaire permettant d'effectuer des recherches dans le système de paquets d'APT. À la différence d'apt-cache, on peut rechercher des informations précises (noms des fichiers du paquet) sur l'ensemble des paquets même si ils ne sont pas installés.
Exemple : `apt-file search stdio.h`

apt-listbugs outil qui liste automatiquement les bogues critiques avant d'installer les nouveaux packages, et, si des bogues critiques sont référencés, qui vous demande de confirmer ou non la mise à jour. Il permet de "blacklister" les paquets ayant des bogues critiques et donc de ne pas les installer.

apt-show-versions liste des paquetages installés.

apticron script pour signaler les mises-à-jour disponibles.

Lien : http://www.debian.org/doc/manuals/apt-howto/

Problèmes dans la gestion des paquets

Bien connaître le système de gestion de paquets Debian est une condition nécessaire pour un administrateur. Ainsi, s'il arrive un problème lors de l'installation ou la mise-à-jour (assez rare avec la version stable), on pourra déboguer les conflits. On gardera à l'esprit quelques commandes essentielles à utiliser avec précaution :

```
# apt-get -f install
```

commande qui tente de corriger les éventuels problèmes lors d'une installation ou mise-à-jour

```
# dpkg --ignore-depends --force-all
```

commande qui permet de forcer l'installation d'un paquet

```
# dpkg --configure -a
```

commande qui relance les étapes de configuration de tous les paquets présents sur le système

2.11 Le noyau Linux

2.11.1 Présentation

Andrew Tanenbaum, professeur, développa en 1985 Minix, un système d'exploitation minimal inspiré du système UNIX Time-Sharing System version 7. Destiné à enseigner le concept des systèmes d'exploitations, Minix fut la source d'inspiration de Linus Torvalds, un étudiant finlandais, qui développa un nouveau système d'exploitation baptisé Linux. Complètement réécrit (principalement en langage C), Linux fut tout d'abord développé pour les ordinateurs de type i386. Linus Torvalds choisit la licence GPL pour son développement et rapidement, grâce à Internet, il reçut l'aide de plusieurs informaticiens. L'explosion des réseaux et d'Internet a largement favorisé le développement collaboratif de Linux, au point que certains le considèrent comme le premier produit d'Internet. Linux est généralement compilé avec GCC et plus accompagné d'outils provenant du projet GNU afin de fournir un système d'exploitation utilisable. C'est pourquoi on a tendance à qualifier ce système d'exploitation de "GNU/Linux" [36]
La première version de Linux sortit en 1991, la version 1.0 sortit en 1993 et la version 2.0 sortit en 1996. Aujourd'hui, Linux supporte plusieurs architectures (Alpha, MIPS, SPARC, PPC, ...) et compte plusieurs millions de lignes de code. On peut télécharger Linux sur le site http://www.kernel.org. Sa branche stable actuelle est la 2.6.x

x.y.z

x désigne la branche (numéro de version majeure), y complète le numéro de version et z est le numéro de release (c'est-à-dire le nombre de publications de cette version). Si y est pair, il s'agit d'une version stable, si y est impair il s'agit d'une version en cours de développement.
Depuis la version 2.6.11, la numérotation a un peu changé avec l'introduction d'un quatrième chiffre indiquant des changements mineurs de versions (quelques patches peu conséquents).

Des suffixes tels que preN ou rcN (prépatches), bkN (snapshots), acN (patches d'Alan Cox) ou mmN (patches d'Andrew Morton) indiquent des versions particulières du noyau.

On peut connaître la version actuelle grâce à la commande `uname` qui affiche des informations concernant la machine et le système d'exploitation sur lequel il est invoqué.

Exemple :
```
$ uname -r
2.6.10-rc2laptop221104
```

Liens :
http://www.kernelnewbies.org/
http://www.bertolinux.com/

36. http://www.gnu.org/gnu/why-gnu-linux.fr.html

2.11.2 Compilation

Sous Debian, le noyau compilé se trouve dans le répertoire /boot sous le nom vmlinuz-x.y.z Les sources du noyau se trouvent dans le répertoire /usr/src/linux-x.y.z (ou kernel-source-x.y.z). Historiquement on retrouve un lien /usr/src/linux pointant vers le répertoire des sources du noyau. On trouve également les modules dans /lib/modules/x.y.z, les entêtes dans /usr/src et les tables de symboles [37] du noyau dans system.map-x.y.z (souvent dans le répertoire /boot)

Une fois l'installation de Linux terminée, on procédera éventuellement (surtout sur un serveur) à la compilation d'un noyau adapté aux besoins de la machine. On privilégiera un noyau débarrassé des modules inutiles : toutes les options strictement nécessaires devront être compilées en dur (sauf exception).

On va donc récupérer les sources du noyau Linux. la procédure la plus classique est de prendre le noyau sur le site officiel http://www.kernel.org/. On prendra garde à bien vérifier l'intégrité des sources téléchargées :

```
$ wget http://kernel.org/pub/linux/kernel/v2.6/linux-2.6.z.tar.bz2.sign
$ gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
$ gpg --verify linux-2.6.z.tar.bz2.sign linux-2.6.z.tar.bz2
```

La recompilation dite classique d'un noyau consiste à vérifier les dépendances (`dep`), nettoyer les sources (`clean`), compiler le noyau en lui-même (`bzImage`), puis les modules (`modules`) et les installer (`modules_install`) :

`make dep` vérifie les dépendances

`make clean` fait un peu le ménage

`make bzImage` compile le noyau

`make modules` compile les modules

makes modules_install
cp arch/i386/boot/bzImage /boot/vmlinuz-new

Avec Debian, il existe les sources Debian du noyau Linux. Il s'agit des sources originales patchées par Debian. On téléchargera ces sources grâce à APT :

```
apt-cache search kernel-source*
```

Lorsque l'on veut patcher les sources du noyau, il faudra éviter d'appliquer des patches incompatibles. C'est pourquoi il peut être préférable de patcher à partir des sources originales. Il existe de nombreux patches pour le noyau Linux. Pour un serveur, on s'intéressera au patches

37. http://www.dirac.org/linux/system.map/

de sécurité grsecurity[38] proposant un certain nombre de fonctionnalités. Voici la procédure pour l'installer :

```
$ tar -jxvf linux-2.6.z.tar.bz2
$ patch -p0 < grsecurity-2.0-2.6.z.patch
```

La phase la plus délicate est en réalité le choix des options du noyau. En effet, il faut lire attentivement les explications de chaque option pour déterminer si on doit l'activer pour notre machine. Pour lancer le choix des options du noyau, on fera :

```
cd linux-2.6.z.tar.bz2
make menuconfig
```

On pourra également utiliser `make config` mais son utilisation est moins aisée. Le choix des options est stocké dans le fichier `.config`. En cas de changement de sources (par exemple en cas de changement de version du noyau), on peut réutiliser ce fichier `.config` en ne choisissant que les nouvelles options. Pour cela on le transférera à la racine des nouvelles sources et on lancera la commande :

```
make oldconfig
```

Debian propose des outils pour compiler simplement. La commande `make-kpkg` permet de construire des paquets Debian contenant un noyau compilé prêt à être installé. On procédera ainsi :

```
make-kpkg clean
make-kpkg kernel_image kernel_headers kernel_source kernel_doc
dpkg -i ../kernel-*-2.6.z.xxx.deb
```

Les options de `make-kpkg` permettent de spécifier un nom particulier pour le noyau (- -append-to-version), de compiler avec initrd (- -initrd), de compiler avec des modules (- -added-modules), etc.

38. http://www.grsecurity.net/

Chapitre 3

Administration Système et Réseau

3.1 Gestion des droits

Sous les systèmes de type Unix ou Linux, il existe plusieurs types de fichiers : les fichiers, les répertoires, les liens symboliques, les fichiers-périphériques.

Un fichier appartient à un utilisateur (en fait un numéro d'utilisateur) et à un groupe (en fait un numéro de groupe).
On peut changer l'utilisateur par la commande :

```
$ chown <new_user> fichier
```

On peut changer le groupe par la commande :

```
$ chgrp <new_group> fichier
```

On peut changer l'utilisateur et le groupe par la commande :

```
$ chown <new_user>.<new_group> fichier
```

Les 3 droits fondamentaux sont la lecture, l'écriture et l'éxecution.
Pour un fichier, ces 3 droits sont définis pour 3 catégories : l'utilisateur, le groupe et le "reste du monde". Pour chacune de ces catégories : on note les droits sous la forme rwx.
r correspond à l'écriture, **w** à l'écriture et **x** à l'exécution. Par exemple, **r-x** correspond à des droits de lecture et d'exécution autorisés, mais l'écriture interdite.
Un notation pratique est de voir **rwx** comme 3 bits. L'autorisation correspond à 1 et l'interdication à 0.
Par exemple **r-x** correspond à 101. En base 10, 101=1*4+0*2+1*1=5. Cela correspond donc à 5.On notera côte à côte les droits des 3 catégories.
Par exemple **rwxr-xr–** correspond à des droits **rwx** pour l'utilisateur, **r-x** pour le groupe et **r–** pour le reste du monde. Si vous avez bien suivi, **rwxr-xr–**=111101100=754
On ajoute (ou retire) des droits avec la commande :

```
$ chmod <catégorie>+<nouveau_droit> fichier
```

`<catégorie>` peut être u (utilisateur), g (groupe) ou a (all=reste du monde). `<nouveau_droit>` peut être r,w ou x
Pour retirer, on mettra un - à la place du +
On peut changer complètement les droits d'un fichier par la commande :

```
$ chmod <nouveaux_droits> fichier
```

Par exemple, <nouveaux_droits>=777 pour autoriser tout par tout le monde.

Pour un répertoire, on pourra lister les fichiers dans ce répertoire si on a les droits **r** du répertoire (et ceux des répertoires supérieurs). Si on a pas ces droits, on ne pourra pas lister. Le droit **rx** permet en plus d'entrer dans ce répertoire. Sans lui, on ne pourra pas effectuer des opérations sur un des fichiers contenus dans ce répertoire. Par exemple, le répertoire /root/ n'a souvent aucun droit pour "all" et donc on ne peut rien faire dans ce répertoire si l'on est pas root.

Notons que le droit **x** seul permet de traverser le répertoire mais pas d'y entrer ou de le lister

Attention, pour pouvoir effacer un fichier ou un répertoire vide, il suffit d'avoir les droits **wx** sur le répertoire contenant ! (avoir uniquement de droit **w** ne sert -a priori- à rien pour un répertoire). Notez que dans un répertoire avec droit **wx**, on peut effacer les fichiers (ou répertoires vides) sur lesquels on n'a pas les droits (par contre, on ne peut pas effacer les répertoires non vides sur lesquels on a pas les droits **wx**).

En plus de cela, il existe des droits spéciaux : setuid, setgid et sticky bit
- Un fichier exécutable peut être setuid, c'est-à-dire qu'au lieu d'être exécuté avec les droits de l'utilisateur qui le lance, il sera exécuté avec les droits du propriétaire de l'exécutable. Ceci s'avère assez dangereux notamment pour les exécutables setuid root. L'exemple-type est le programme passwd qui permet de changer de mot de passe. Il est exécutable par un utilisateur mais il est setuid root car seul root peut écrire dans les fichiers /etc/passwd et /etc/shadow
- Un fichier exécutable peut être setguid. Il s'agit de la même notion que celle vue ci-dessus pour le groupe. Le fichier est donc exécuté avec les droits du groupe auquel il appartient.
- Un fichier exécutable peut être sticky, c'est-à-dire avoir le sticky bit positionné. Cela signifie qu'il reste en mémoire même après la fin de son exécution afin d'être relancé plus rapidement. Attention, seul root peut positionner le sticky bit. - Un répertoire peut être setgid. Cela signifie que tous les fichiers créés dans ce répertoire appartiendront au même groupe que le répertoire.
- Un répertoire peut être sticky bit. Cela signifie dans ce répertoire, un utilisateur ne pourra effacer que les fichiers qui lui appartiennent. L'exemple-type est le répertoire /tmp où tout le monde peut écrire mais où l'on ne peut effacer que ce que l'on a créé.
Ces droits spéciaux sont notés sst où le premier s correspond au setuid, le second au setgid et le t au sticky bit.
On écrira également cela sous la forme de bits. Par exemple s-t=101=5 On ajoute (ou retire) des droits spéciaux avec la commande :

```
$ chmod +<droit_spécial> fichier
```

<droit_spécial> peut être s (setuid+setgid) ou t (sticky bit)
On change complètement les droits spéciaux et les droits d'un fichier par la commande :

```
$ chmod <nouveaux_droits_spéciaux><nouveaux_droits> fichier
```

<nouveaux_droits_spéciaux> s'écrit en base 10. Par exemple, 5 pour setuid et sticky bit.

Il faut aussi définir la politique de gestion des droits de la machine, c'est-à-dire se poser la

question "Qui a le droit de faire quoi ?"
Définissons tout d'abord les droits au niveau des données utilisateurs. Les droits par défaut sont gérés par la directive `umask`. On mettra donc, selon sa politique, dans le fichier */etc/profile* (et dans login.defs pour gérer lors de la création de l'utilisateur) :

umask 022 : pour que les données utilisateurs soient visibles par tous les utilisateurs

umask 027 : pour que les données d'un utilisateur soient visibles entre eux par les utilisateurs du même groupe

umask 077 : pour que seul l'utilisateur puisse lire ses fichiers par défaut

Il faut ensuite gérer les droits sur les périphériques. Les différents périphériques appartiennent souvent à un groupe (cdrom, audio, video, etc.) et on peut gérer les droits en ajoutant ou non les utilisateurs à ces groupes. On ne détaillera pas trop cette procédure car elle est surtout valable dans le cas où il s'agit de postes de travail accessible physiquement aux utilisateurs.

Il faut également penser à mettre des protections sur les répertoires et partitions accessibles à l'utilisateur (on traitera le cas des journaux systèmes à part : voir par la suite). Ces protections ne sont pas infaillibles mais constituent un premier rempart dans le cas d'attaques d'un utilisateur inexpérimenté (souvent appelés script-kiddy). Voici un exemple de protection que l'on peut mettre dans le fichier */etc/fstab* :

```
/dev/sda8 /tmp ext3 defaults,nodev,nosuid,noexec,usrquota,grpquota 0 2
# disques amovibles
/dev/fd0 /mnt/fd0 ext2 defaults,users,nodev,nosuid,noexec 0 0
/dev/fd0 /mnt/floppy vfat defaults,users,nodev,nosuid,noexec 0 0
/dev/hdc /mnt/cdrom iso9660 ro,users,nodev,nosuid,noexec 0 0
```

Remarque : certains paquets (screen, PostgreSQL, etc.) nécessitent d'exécuter un script dans */tmp/*. On fera donc :

```
# mount -o exec,remount /tmp
```

Avant d'installer des paquets, puis on refera :

```
# mount -o remount /tmp
```

Plus généralement on peut vérifier ce que peut voir un utilisateur par les commandes :

```
$ find / -type f -a -perm +006 2>/dev/null}
$ find / -type d -a -perm +007 2>/dev/null}
```

On prendra garde par exemple aux données sensibles des fichiers de configuration ou des scripts (notamment des données web). On peut par exemple voir les fichiers du répertoire /etc non visibles par un utilisateur :

```
# find /etc -type f -a -perm 600 -a -uid 0
```

On peut également restreindre les droits sur certaines applications dangereuses.
Exemple :

```
chmod o-x /usr/bin/nmap
chmod -s /bin/ping
```

On pourra autoriser l'utilisation de n'importe quel programme (même des programmes administrateurs) grâce au logiciel sudo [1]. Son fichier de configuration est /**etc/sudoers**. Voici un exemple :

```
User_Alias STAFF=jo,zette
Cmnd_Alias NET=/bin/ping,/usr/bin/traceroute,/usr/bin/nmap

root ALL=(ALL) ALL
STAFF ALL=(ALL) NET
```

 # /etc/init.d/sudo restart

3.2 Quotas

Il est souvent intéressant de définir des quotas :

Configuration du noyau : CONFIG_QUOTA

Ajoutez les options *usrquota* et *grpquota* pour les partitions concernées dans le fichier */etc/fstab* :

```
/dev/hdc8 /home ext3 rw,nosuid,nodev,exec,nouser,auto,async,usrquota,grpquota 0 2

# touch /home/aquota.user /home/aquota.group
# chmod 600 /home/aquota.* && chown root:root /home/aquota.*
# mount -v -o remount /home
# apt-get install quota quotatool -> warnquota
# update-rc.d -f quotarpc remove
# quotacheck -auvg
# quotaon -auvg
```

On peut maintenant créer un utilisateur qui va servir d'exemple pour les quotas des autres utilisateurs :

```
# adduser --home /dev/null --shell /bin/false --ingroup nogroup --disabled-password forquota
# edquota -u forquota
```

La taille d'un "block" est habituellement de 1024 octets sous Linux. Elle peut varier selon les options des systèmes de fichiers.

Les inodes représentent le nombre maximum de fichiers ou de répertoires que l'on pourra créer.

1. http://www.courtesan.com/sudo/

On pourra également définir le délai (grace period) qui définit le temps au-delà duquel la limie douce devient limite dure. On change ce délai avec la commande :

```
# edquota -t
```

On peut ensuite appliquer les quotas à chaque utilisateur par la commande :

```
# edquota -p forquota USER
```

On peut utiliser le fichier adduser.conf pour imposer un quota lors de la création de l'utilisateur [voir plus haut].
D'autres commandes intéressantes permettent de vérifier le bon fonctionnement des quotas (quotacheck), d'envoyer des messages d'avertissement aux utilisateurs dépassant la limite douce (warnquota) et de générer des statistiques :

```
# repquota -ugva
```

Note :
Attention, l'utilitaire dd semble mal gérer les quotas.
Liens :
http://www.freenix.fr/unix/linux/HOWTO/mini/Quota.html
http://linux.developpez.com/cours/securedeb/?page=page5#L5.4

3.3 Crontab

Cron est démon qui permet d'exécuter automatiquement des commandes ou des scripts à une date et une heure spécifiées à l'avance.

C'est évidemment très utile pour toutes les tâches d'aministration. Le démon cron se base sur le fichier */etc/crontab* pour lancer des actions toutes les heures (cron.hourly), tous les jours (cron.daily), toutes les semaines (cron.weekly) et tous les mois (cron.monthly).

Déjà, il peut être intéressant de personnaliser les paramètres du */etc/crontab* afin d'éviter l'utilisation des horaires par défaut.
Ensuite on peut placer des scripts (exécutables) dans les cron.*ly ou bien dans le répertoire cron.d afin de personnaliser l'horaire :
Squelette :
"minute" "heure" "date" "mois" "jour" "utilisateur" "commande"

Exemple signifiant du lundi au vendredi, toutes les 3h à la 5ème minute :

```
5 */3 * 1-5 root ntpdate serveur-ntp
```

3.4 Gestion des Journaux

Les journaux sont des fichiers qui contiennent des informations d'activité datée. Ils sont essentiels pour un serveur pour de nombreuses raisons : vérifier des actions passées, générer des statistiques, déboguer un programme. La vérification des actions passées est notamment

importante en cas de problème (piratage, service défectueux). La justice oblige également à conserver certains journaux pendant une certaine durée. Un flou concerne ce qu'il faut réellement conserver (apparemment seules les informations d'entêtes mais pas le contenu en lui-même) et la durée (cela varie entre 3 mois, 1 an et 3 ans si l'on se base sur les lois françaises ou européennes). Des décrets d'application devraient éclaircir ces points dans les prochains mois.

Sous Debian, les journaux se trouvent généralement dans le répertoire */var/log*. On va distinguer les journaux systèmes et les journaux applicatifs. Les journaux systèmes sont gérés par le démon SYSLOG[2]. Sa configuration se trouve dans le fichier *syslog.conf*. Voici quelques ligne extraites de ce fichier :

```
daemon.*                                -/var/log/daemon.log
kern.*                                  -/var/log/kern.log
mail.*                                  -/var/log/mail.log
*.=debug;auth,authpriv.none;news.none;mail.none  -/var/log/debug
*.emerg                                 *
*.*;auth,authpriv.none                  /dev/tty8
```

Voici les fichiers pincipaux générés par SYSLOG :

auth.log : authentification système (login, su, getty)
daemon.log : relatif aux daemons
mail.* : messages relatifs aux mails
kern.log : messages générés par le noyau
user.log : message généré par des programmes utilisateur
debug : messages de bogues
messages : messages d'info
syslog : tous les messages

Les journaux applicatifs sont générés par chaque application. Ils sont souvent dans un répertoire du nom de l'application situé dans */var/log*.

L'un des points essentiels est la rotation des journaux, c'est-à-dire l'action de fermer le journal actuel (et éventuellement le compresser) et d'en ouvrir un autre. Il existe actuellement deux programmes qui se chargent d'effectuer ce travail. Le script `savelog` (un outil Debian) et le programme `logrotate`. Par défaut `savelog` gère les journaux système et `logrotate` gère les journaux applicatifs (apache, mysql, ppp, etc.).
Logrotate est exécuté tous les jours (cron.daily)
Savelog est exécuté tous les jours (crond.daily) pour syslog notamment.
Savelog est exécuté toutes les semaines (crond.weekly) pour les autres journaux système.

L'option "-d" de savelog permet d'utiliser la date lors de la rotation des journaux et de ne pas les effacer. On pourra donc ajouter ses propres règles dans les scripts cron pour faire une sauvegarde distante des journaux (éventuellement dans une base de données). Pour des serveurs dédiés (applications clés), on peut augmenter la fréquence des rotations et des sauvegardes distantes, mais également utiliser des scripts afin de détecter toutes alertes ou anomalies et les envoyer par courrier électronique ou même SMS.

2. ftp://ftp.rfc-editor.org/in-notes/rfc3164.txt

Il existe des programmes analysant les journaux permettant de détecter des problèmes, avoir des statistiques (lire logtool prelude).

Il faut donc bien insister sur la nécessité de surveiller les services en production à l'aide d'outils adaptés. On dispose donc d'outils s'appuyant souvent sur le protocole SNMP (Simple Network Management Protocol) qui permet de gérer et diagnostiquer les problèmes en transférant des informations système (réseau, charge, état, etc.).
Pour exploiter ces informations, on pourra tracer des courbes MRTG [3] ou RRDtool [4]. Ces logiciels permettent de produire des courbes totalement personnalisées pour tracer des courbes.

3.5 OpenSSH

SSH signifie Secure SHell (Shell sécurisé). Le protocole SSH est en cours de standardisation par l'IETF [5]. Les outils du protocole SSH sont utilisés par un nombre croissant de personnes. Ils sont destinés à sécuriser le login à distance, sécuriser les transferts de fichiers et sécuriser les TCP/IP et X11 forwardings. Il peut automatiquement chiffrer, authentifier et compresser des données transmises.

La principale implémentation du protocole est OpenSSH [6]. OpenSSH est une version libre de la suite d'outils du protocole SSH. De nombreux utilisateurs de telnet, rlogin, ftp et autres programmes identiques ne réalisent pas que leur mot de passe et leurs données sont transmis de façon non chiffrée. OpenSSH chiffre tout le trafic (mots de passe inclus) de façon à déjouer les écoutes réseau, les prises de contrôle de connexion, et autres attaques. De plus, OpenSSH fournit toute une palette de possibilités de tunnel et de méthodes d'authentification. OpenSSH **doit** être utilisé à la place de telnet et autres logiciels non-sûrs.

Pour la plupart de ses fonctionnalités cryptographiques, OpenSSH s'appuie sur la bibliothèque OpenSSL. La suite logicielle OpenSSH inclue les programmes ssh qui remplace telnet et rlogin, scp qui remplace rcp, et sftp qui remplace ftp. De plus sshd, la partie serveur, est inclus ainsi que d'autres utilitaires tels que ssh-add, ssh-agent, ssh-keygen, ssh-keysign, ssh-keyscan, et sftp-server. OpenSSH supporte les protocoles SSH 1.3, 1.5 et 2.0. Intéressons nous à sa configuration, qui se trouve dans le fichier *sshd_config* :

```
Protocol 2
LoginGraceTime 30
PermitRootLogin no
AllowUsers moi admin
ClientAliveInterval 15
ClientAliveCountMax 45
```

3. http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
4. http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/
5. http://www.ietf.org/html.charters/secsh-charter.html
6. http://www.openssh.org

On peut également imposer des restrictions supplémentaires dans le fichier pam.d/ssh :

```
auth required pam_listfile.so sense=allow onerr=fail
item=user file=/etc/loginusers
```

Si des utilisateurs normaux sont destinés à utiliser SSH, il peut être intéressant de l'installer dans une prison Chroot. Voici quelques liens qui expliquent cette mise en place :

Liens :
http://chrootssh.sourceforge.net/index.php
http://www.debian.org/doc/manuals/securing-debian-howto/ap-chroot-ssh-env.fr.html
http://vince.kerneled.org/files/ssh_chroot.txt

3.6 Transfert de fichiers

Le transfert de fichiers par le protocole FTP fait circuler identifiants, mot de passe et données en clair sur le réseau. Il est vraiment préférable d'utiliser scp ou sftp, du projet OpenSSH, qui permet de transférer des fichiers de façon plus sécurisée. Mais dans certains cas, notamment dans le cas de serveur web mutualisé, il est de coutume d'offrir un accès FTP. Il faut donc prendre quelques précautions. Par exemple, avec le serveur ProFTPD [7], certaines directives du fichier de configuration *proftpd.conf* sont importantes :

```
DefaultRoot ~
DenyFilter \*.*/
```

On pourra également utiliser PAM pour limiter l'accès selon les utilisateurs. Dans le fichier */etc/pam.d/proftpd* :
auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed

On aura donc un fichier */etc/ftpusers* qui spécifiera les identifiants ne pouvant pas utiliser les services FTP (on pourrait faire l'inverse aussi).

Dans le but de chiffrer les connexions FTP, on peut également utiliser une couche SSL. Les serveurs Pure-FTP [8] ou linux-ftpd-ssl [9] permettent cela. Il faut bien noter qu'il faut utiliser un programme client capable de l'utiliser (mais c'est le cas de plus en plus de clients).

3.7 Authentification

Le fichier `/etc/passwd` contient la liste des utilisateurs avec le mot de passe chiffré. Ainsi lors de la procédure d'authentification d'un utilisateur, le système teste si le chiffrement du mot

7. http://www.proftpd.org/
8. http://www.pureftpd.org/
9. http://freshmeat.net/projects/linux-ftpd-ssl/

de passe entré correspond au mot de passe chiffré (`/etc/passwd` est accessible en lecture aux utilisateurs). Il est souhaitable d'utiliser l'algorithme de chiffrement MD5 pour chiffrer les mots de passe. Il est également conseillé d'utiliser l'authentification shadow. Avec cette authentification, les mots de passe du fichier /etc/passwd sont remplacés par 'x' et sont stockés dans le fichier `/etc/shadow`, inaccessible en lecture aux utilisateurs. D'autres méthodes d'authentification locale existent comme sous OpenBSD [10] mais l'utilisation de la méthode shadow est répandue sur les distributions Linux. Il faut bien avoir conscience que cela repose sur la solidité de l'algorithme de chiffrement [11].

Le fichier `/etc/group` stocke la liste des groupes, c'est-à-dire des entités regroupant plusieurs utilisateurs et permettant de donner à ce groupe d'utilisateurs les mêmes droits sur des fichiers. Un utilisateur peut connaître les groupes auxquels il appartient en tapant la commande `groups` ou encore `id`.
Le fichier /etc/adduser.conf contient les valeurs par défaut pour les programmes *adduser addgroup deluser* et *delgroup*. Chaque option est de la forme option = valeur. Les simples ou doubles guillemets sont autorisés autour de la valeur. Les lignes de commentaires doivent avoir un caractère dièse (#) au début de la ligne.

Exemple :

```
DSHELL=/bin/bash
DHOME=/home
SKEL=/etc/skel
QUOTAUSER="forquota"
DIR_MODE=0755
```

Le répertoire /etc/skel/ contient le profil par défaut qui sera copié dans le répertoire personnel d'un nouvel utilisateur. Il contient souvent les fichiers .bashrc, .bash_profile, etc. Pour configurer un profil pour les utilisateurs, on ajoutera des fichiers dans /etc/skel (par exemple des boites à mail, paramètres pour certaines applications, etc.)

La gestion des utilisateurs se fait grâce aux commandes adduser et addgroup. Notez bien que les commandes useradd, groupadd, userdel et groupdel ont une syntaxe différente et n'utilisent pas forcément les mêmes configurations.

Exemple :

```
# addgroup --gid 107 student}
# adduser --home /home/jean --shell /bin/bash
  --uid 1057 --ingroup student jean
```

L'utilisation d'un mot de passe aléatoire contenant des lettres majuscules et minuscules, des chiffres et des caractères spéciaux (au total plus de 8 caractères) est fortement recommandée. Toute trace écrite devra être bannie si possible surtout pour un administrateur - c'est son boulot ;) - à l'exception d'endroits sécurisés (coffre-fort, banque, etc.) accessibles par des supérieurs hiérarchiques. Une procédure en cas d'accident corporel de l'administrateur pourra être mise en place.

10. http://openbsd.org/
11. http://evolix.org/man/crypt.html

Pour les mots de passe utilisateurs, il peut être utile d'empêcher les mots de passe trop simples (cracklib) et de faire tourner des crackeurs de mots de passe afin de vérifier en permanence qu'aucun utilisateur n'a un mot de passe trop simple etc.

Voici quelques programmes de génération de mot de passe aléatoire disponibles sous Debian :
`otp apg makepasswd pwgen`

Il existe également des outils pour tenter de cracker les mots de passe. L'un des plus connu est le programme `john` qu'il est intéressant de faire tourner régulièrement afin de détecter si des utilisateurs ont des mots de passe trop simples.

3.8 Gestion de l'authentification

Nous savons comment fonctionne la procédure d'authentification [voir plus haut] mais au-delà de l'authentification, il est nécessaire d'imposer des restrictions au niveau de la procédure de login pour éviter des désagréments. Pour cela on dispose de plusieurs moyens de restreindre les accès et imposer d'autres paramètres.

Commençons par le fichier `/etc/login.defs` qui définit les paramètres d'authentification. Certains paramètres sont notamment importants :

```
# delai minimum entre deux tentatives de login
FAIL_DELAY 10
# journaliser les tentatives ratées
FAILLOG_ENAB  yes
# retenir les identifiants iconnus essayés
LOG_UNKFAIL_ENAB yes
# retenir les tentatives réussies
LOG_OK_LOGINS yes
# delai maximim pour authentification
LOGIN_TIMEOUT  60
```

Le fichier `/etc/securetty` contient la liste des terminaux sur lesquels la connexion de 'root' est autorisée. Il est conseillé de mettre ce fichier vide afin de désactiver le login de 'root'. Seul 'su' permettra donc de devenir superutilisateur. Faire donc :

```
# sed -e 's/^/#/' /etc/securetty > /tmp/sed
# mv /tmp/sed /etc/securetty
```

Linux-PAM
La plupart des distributions Linux utilisent l'authentification PAM par défaut.
De nombreuses applications utilisent l'authentification PAM et notamment login, su qui sont assez importantes...
On vérifiera qu'une application utilise l'authentification PAM grâce à `ldd`
Ex :

```
ldd $(which su)
```

Lien : http://www.kernel.org/pub/linux/libs/pam/

La configuration se passe dans le répertoire `/etc/pam.d/`
Si ce répertoire est absent, la configuration pourra avoir lieu dans un fichier unique : `/etc/ldap.conf` (ignoré sinon)
Chaque application est configurée dans un fichier portant son nom.
Le répertoire `/etc/security/` contient des fichiers de configuration sécurité pour PAM.

Le comportement par défaut est dans le fichier `/etc/pam.d/other` La syntaxe des fichiers est constituée de lignes telles que : "type" "niveau" "module" "arguments"

"type" peut être :

auth : authentification
account : vérification des types de services autorisés
session : tâches à effectuer avant/après l'accès
password : mécanismes d'authentification

"niveau" peut être :

required : le succès à cette étape est nécessaire
requisite : le succès est nécessaire et l'accès est refusé en cas d'erreur
sufficient : le succès à cette étape suffit
optional : l'accès pourra être refusé en fonction d'autres paramètres

"module" :
(Ils se trouvent par défaut dans le répertoire /lib/security/)

pam_access.so : restriction d'accès avec le fichier access.conf

Le fichier `access.conf` permet de gérer les permissions de login (si activé dans PAM) selon la syntaxe suivante :

[*] permission : users : origins
[*] = + (accès permis) ou - (accès refusé)

Exemples :

- *:mechant pasbeau :ALL* -> les comptes mechant et pasbeau ne peuvent pas se loguer

- *:ALL EXCEPT admin :ALL EXCEPT LOCAL* -> seuls admin peut se loguer à distance

Sur un serveur où seuls les administrateurs peuvent se connecter :

- *:ALL EXCEPT admin admin2 :LOCAL*

pam_deny.so : interdiction d'accès

pam_env.so : utilise les variables d'environnement de pam_env.conf en plus du fichier

/etc/environment

pam_filter.so : utiliser divers filtres

pam_ftp.so : avoir un accès de type FTP anonyme

pam_group.so : utilise group.conf pour imposer des restrictions selon le groupe

pam_issue.so : pour afficher /etc/issue lors d'un login

pam_lastlog.so : donne des informations sur la dernière connexion de l'utilisateur

pam_ldap.so : module pour l'authentification LDAP

pam_limits.so : restrictions particulières avec le fichier limits.conf

Le fichier limits.conf permet d'imposer des limites diverses sur les groupes ou les identifiants :

Structure : "qui" "type" "quoi" "combien"

qui = compte, @groupe, *

type = soft (soft limits) ou hard (hard limits)

quoi = core, data, fsize, memlock, nofile, rss, stack, cpu, nproc, as, maxlogins, priority, locks

Exemples :

```
ftpusers - maxlogins 3
@invite hard cpu 5
@users hard data 10000
```

pam_listfile.so : permet d'autoriser ou non d'après une liste

pam_mail.so : pour indiquer si de nouveaux mails sont arrivés
login session optional pam_mail.so dir= /Maildir/

pam_mkhomedir.so : pour créer le home des utilisateurs authentifiés (pratique pour les utilisateur NIS ou LDAP)
`session required pam_mkhomedir.so skel=/etc/skel/ umask=022`

pam_nologin.so : empécher tout login ! = root si le fichier /etc/nologin existe

pam_permit.so : autoriser l'accès (dangereux)

pam_rootok.so : autoriser si l'utilisateur est root (id=0)

pam_securetty.so : permettre l'accès de root seulement si le PAM_TTY figure dans le fichier `/etc/securetty`

pam_shells.so : permettre l'accès si le shell de l'utilisateur est listé dans le fichier `/etc/shells`

pam_tally.so : permet de bloquer les tentatives au bout d'un certain nombre d'échecs

```
account required /lib/security/pam_tally.so per_user deny=5
```

pam_time.so : permet de restreindre certains services selon le temps d'après le fichier time.conf

```
games ; * ; !waster ; Wd0000-2400 | Wk1800-0800
```

pam_unix.so : module standard d'authentification Unix

pam_userdb.so : module d'authentification sur une Berkeley DB

```
auth sufficient pam_userdb.so icase db=/etc/id.db
```

pam_warn.so : journalise certains paramètres pour syslog

pam_wheel.so : permettre l'accès root uniquement au membre de whell (gid=0)

pam_cracklib.so : vérifie que le mot de passe répond à certains critères (taille, simplicité)

```
password required pam_crackedlib.so type=Evolix retry=0 minlen=8
dcredit=2 ucredit=2 lcredit=2 ocredit=1
password required pam_pwdb.so use_authtok nullok md5
```

Lien : http://perso.wanadoo.fr/alexandre.vidal/pam/

On rappelle que pour avoir des renseignements sur l'utilisateur actuellement connecté, on fera :
 # whoami
 # id

On peut également avoir divers renseignements sur les utilisateurs :

 # who : donne les utilisateurs actuellement connectés
 # last : donne les dernières connections grâce à fichier wtmp
 # w.procs : donne les utilisateurs actuellement connectés et des renseignements complémentaires

3.9 Sécurité

La sécurité informatique englobe non seulement la sécurité réseau pour se prémunir d'attaque locale ou extérieure, mais également la sécurité physique, la gestion des utilisateurs, des droits, des journaux et des sauvegardes.

Sécurité physique au niveau des infrastructures

Même si les gens ont plutôt tendance à l'oublier, la sécurité physique de la machine est très importante. Il est nécessaire qu'un disque dur ne soit pas en évidence, prêt à être volé avec toutes les informations qu'il contient. Il est important que l'alimentation d'un serveur soit protégée, et qu'il ne suffise pas juste au pirate de débrancher le cordon à la prise pour que des dizaines de couches de sécurité logicielles soient anéanties.

Sécurité physique au niveau de l'utilisation

Lorsque vous tapez un mot de passe, il n'est pas superflu de vous assurer que personne ne soit penché ou dessus de votre épaule, ou qu'aucun élément d'écoute a été rajouté (keylogger logiciel ou physique).

Au niveau du BIOS

Une première méthode de sécurisation consiste à autoriser uniquement le disque dur à démarrer. Il faut aussi définir un mot de passe au niveau du BIOS empêchant de changer les paramètres de ce dernier pour un démarrage sur un autre périphérique dans l'espoir de pouvoir ensuite avec différents outils accéder aux données du disque dur.

Il est également possible d'avoir des protections complexes, comme une authentification mutuelle entre le BIOS et le disque, mais ces protections sont rarement mises en place.

Au niveau du Boot Loader

Lilo
Le fichier de configuration principal est lilo.conf (*/etc/lilo.conf*)
Dans ce fichier la possibilité est donnée de mettre en place un mot de passe par l'ajout de la ligne : `password = motdepasse`. A remarquer que contrairement à grub que l'on va décrire, une modification de lilo.conf pour être prise en compte doit être suivie de la commande `#lilo` exécutée en tant que root.

Grub
De même que précedemment il est possible avec grub d'ajouter un mot de passe qui peut même être crypté :

Lancer le shell GRUB :

```
# grub
grub> md5crypt
```

entrez votre mot de passe :

Password : *********

Encrypted : 1gxRBf0$pe0rH7/nG9KPJLvc.XV7V.

puis, copiez le mot de passe crypté dans votre fichier de configuration `/boot/grub/menu.lst` :

```
password --md5 $1$gxRBf0$pe0rH7/nG9KPJLvc.XV7V.
```

L'argument password peut être utilisé pour restreindre certaines entrées ; dans ce cas il est inséré juste sous la ligne "title" à démarrer.

Dans le fichier */etc/inittab* permet à un utilisateur physique de redémarrer la machine. Vous pouvez donc supprimer cette ligne :
`ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now`
Pour demander à init (le processus d'initialisation) de réexaminer `/etc/inittab`, faire :
`# init q`

Néanmoins, malgré toutes ces protections possibles, il faut considérer qu'avoir un accès physique à une machine permet d'en faire ce que l'on veut. C'est pourquoi on se contente souvent des protections basiques comme fermer la salle des machines à clé, mettre en place des cadenas sur les boitiers ou encore avoir un système de vidéosurveillance.

Firewall

Sous Linux, il existe un programme puissant qui permet d'avoir des règles de firewall complexes. Ce programme se nomme (depuis le noyau Linux 2.4.x) Netfilter/IPTables [12]. Il permet d'écrire des règles en ligne de commande, et donc le lancement de règles se fait souvent sous forme de script.

Un script détaillé se trouve en annexe à titre d'exemple.

Détection d'intrusion

Au-delà des précautions prises, il faut avoir conscience des diverses possibilités d'intrusion et d'attaque. Dans le but de détecter et prévenir ce type de danger, il existe des programmes de détections d'intrusion (IDS) qui permettent d'obtenir des rapports de trafic, des alertes et éventuellement des réactions dynamiques.
Les logiciels Prelude [13] et Snort [14] sont les IDS libres les plus connus.

Ils peuvent être couplés avec une base de données (MySQL, PostGreSQL) et respectivement avec les logiciels Piwi et ACID (Analysis Console for Intrusion Databases) [15] pour obtenir les rapports par une interface web.

Il existe un autre type de logiciels de sécurité : le *pot de miel*. Il s'agit de serveur (ou serveur virtuel) laissé sous surveillance étroite sur un réseau et dont le rôle est d'être attaqué et compromis afin d'étudier le comportement et les outils des pirates.
Le programme libre le plus connu est honeyd [16] mais il existe d'autres implémentations (tiny-honeypot [17], etc.) D'autres programmes moins élaborés permettent de surveiller des actions

12. http://www.netfilter.org/
13. http://www.prelude-ids.org/
14. http://www.snort.org/
15. http://acidlab.sourceforge.net/
16. http://www.honeyd.org/
17. http://www.alpinista.org/thp/

d'utilisateurs :

TTYSnoop [18] permet de dériver des terminaux de connexion,
Snoopy [19] pour intercepter les commandes sur le système,

On citera également les nombreux outils qui permettent d'écouter et d'analyser le trafic réseau :

`tcpdump, ethereal, dsniff, sniffit` [20]`, ngrep, hunt, tcpick nast, karpski, vnstat, ndiff, etc.`

Piratage ?
On peut vérifier l'intégrité du système grâce à des outils de détection des rootkits : Chkrootkit [21] :
ou Rootkit Hunter [22]

Isoler les services

Chroot
Chroot est un utilitaire permettant d'emprisonner un service dans une arborescence limitée. Cela permet de limiter l'accès aux services offerts et éviter d'avoir un accès complet au système en cas de failles logiciel. Sous Debian, il existe quelques services tournant dans une prison Chroot par défaut (Postfix notamment) mais si l'on veut emprisonner des services comme Bind, Apache, serveur FTP, OpenSSH, il faudra créer la prison Chroot manuellement. Le script makejail [23] permet d'aider à la création de prisons Chroot à l'aide de fichiers de configuration. On notera que les systèmes *BSD (notamment OpenBSD [24]) sont en avance pour les services d'emprisonnement. De nombreux services sont ainsi emprisonnés par défaut.

D'autres programmes permettent d'émuler plusieurs systèmes sur une seule machine :

Vserver
Linux-VServer est projet libre proposant un patch pour le noyau Linux ainsi qu'un ensemble d'outils pour permettre de lancer plusieurs distributions Linux au dessus d'un seul noyau et partager les ressources.
Lien : http://www.linux-vserver.org/

UML
UML (User Mode Linux) est un projet libre permettant de lancer plusieurs noyaux Linux comme un simple programme (en mode utilisateur donc) ce qui permet d'avoir des machines virtuelles Linux. Le contenu du disque du système virtuel est stocké dans un seul fichier.
Lien : http://user-mode-linux.sourceforge.net/

18. http://freshmeat.net/projects/ttysnoop/
19. http://sourceforge.net/projects/snoopylogger/
20. http://reptile.rug.ac.be/~coder/sniffit/sniffit.html
21. http://www.chkrootkit.org/
22. http://www.rootkit.nl/projects/rootkit_hunter.html
23. http://www.floc.net/makejail/
24. http://www.openbsd.org/

Vmware
Vmware est un programme propriétaire plutôt réputé permettant de lancer plusieurs machines virtuelles (MS-Windows, Linux, *BSD) Il existe une version Workstation permettant de lancer des machines virtuelles comme des applications et une version Serveur permettant de lancer à distance des machines virtuelles sur un serveur.
Lien : http://www.vmware.com/
Mises-à-jour logicielles de sécurité
"Nous ne cacherons pas les problèmes. Nous garderons toujours notre base de données des rapports de bogues ouverte au public. Les rapports que les utilisateurs remplissent en ligne seront immédiatement visibles par les autres." extrait du Contrat social Debian

Comme pour la plupart des logiciels, la politique de sécurité de Debian est d'avoir une gestion transparente des failles et des bogues. Plusieurs études ont montré que cette politique était la plus efficace en terme de sécurité. Néanmoins, cela a pour conséquence pour l'administrateur d'avoir une réactivité accrue pour faire le nécessaire dès qu'un problème est annoncé. Il existe donc une équipe de sécurité Debian qui est chargée de publier des paquets corrigés pour la version stable de Debian dès qu'un problème est découvert. Pour pouvoir profiter de ces paquets mis à jour, il faut avoir la ligne suivante dans son fichier *sources.list* :
`deb http://security.debian.org/ stable/updates main contrib non-free`
Le minimum est de s'abonner à la liste de diffusion debian-security-announce. Mais pour aller plus loin, il est conseillé de suivre de près les listes de diffusion des logiciels utilisés en production et également des listes traitant de sécurité notamment la liste Bugtraq [25].

3.10 Réseau

Au niveau d'un serveur, il est préférable d'opter pour une configuration réseau statique. En effet dépendre d'un tiers (serveur DHCP) pour des paramètres aussi essentiels que les paramètres réseau n'est pas prudent. Dans le cas où l'on voudrait quand même avoir un configuration par DHCP, il faudra au minimum une authentification par adresse MAC.
Tout d'abord, parlons de quelques outils réseau utiles :

netstat : netstat -a -u
lsof : lsof -i 4
tcpdump : tcpdump -X -i eth2 port 22 > log_ssh
nmap : http ://www.insecure.org/nmap/ nmap -O -P0 192.168.23.27
hping : outil générateur de paquet réseau
scapy : outil de manipulation de paquets
arping : arping -a -S 192.168.3.12 -c 3 -i wlan0 192.168.54.22
dsniff : dsniff -i eth1

On va donc commencer par contrôler les services réseau ouverts sur la machine et les fermer si nécessaires :

Sur un système installé depuis peu, on trouve souvent ces services ouverts :

25. http://www.securityfocus.com/archive/1

```
# nmap localhost  9/tcp   open  discard
13/tcp  open  daytime
37/tcp  open  time
111/tcp open  rpcbind
113/tcp open  auth
```

discard, daytime et time sont démarrés par inetd

Inetd est historiquement un "super-serveur" permettant de configurer plusieurs services. Pour désactiver les services précédents on procédera à la commande suivante :

```
# update-inetd --disable discard,daytime,time,ident

WARNING!!!!!! /etc/inetd.conf contains multiple entries for
the ‘discard’ service. You’re about to disable these entries.
Do you want to continue? [n] y
```

Le super-démon

Si vous avez l'intention d'utiliser le serveur Inetd, il est préférable d'utiliser Xinetd[26], qui remplace Inetd en offrant des fonctionnalités plus complètes notamment en terme de sécurité. Pour installer Xinetd, on installera le paquet xinetd tout simplement. Voici un exemple du fichier de configuration xinetd.conf :

```
service imap3
{

        socket_type         = stream
        protocol            = tcp
        wait                = no
        user                = root
        passenv             =
        server              = /usr/bin/tcpd
        server_args         = /usr/sbin/imapd
}
```

On fermera également le port 111 en supprimant le programme portmap (utile pour NIS ou NFS notamment) :

```
# aptitude remove portmap nfs-comon
```

26. http://www.xinetd.org/

On supprimera tous les services inutiles.
On vérifie les services TCP ouverts par :

```
# nmap localhost
# netstat -a -t
```

Et les services UDP par :

```
# nmap -sU localhost
# netstat -a -u
```

On peut vérifier quels sont les processus qui ouvrent les ports par :

```
# lsof -i 4
# lsof -i 6
```

Note : dans le cas où l'on utilise une configuration réseau par DHCP, on ne s'étonnera pas de trouver le port 68 ouvert en UDP.

3.11 Monitoring

Voyons un exemple simple qui trace des courbes en fonctions du temps de réponse de `ping`.
On installera les programmes snmpd et mrtg.
Dans `/etc/snmp/snmpd.conf`, ajouter cette ligne :
com2sec readonly default public
Attention, bien veillez à supprimer cette ligne :
com2sec paranoia default public

`/etc/mrtg/ping :`

```
#!/bin/sh

P=‘ping  -c3 -q google.fr |grep avg|cut -d" " -f4‘
MIN=‘echo $P|cut -d"/" -f1‘
MAX=‘echo $P|cut -d"/" -f2‘
echo $MAX
echo $MIN
```

mrtg.conf :

```
WorkDir: /var/www/mrtg
Language: French
```

```
Target[ping]: `/etc/mrtg/ping`
Options[ping]: nopercent,growright,gauge,noinfo, nobanner
MaxBytes[ping]: 10000
AbsMax[ping]: 10000
YLegend[ping]: Latence
ShortLegend[ping]: ms
Legend1[ping]: Latence max en ms
Legend2[ping]: Latence min en ms
LegendI[ping]: Latence Max:
LegendO[ping]: Latence Min:
Title[ping]: Ping sur Google
PageTop[ping]: <h1>Latence Google.fr</h1>
WithPeak[ping]:wmy
Legend4[ping]: Max de la latence min
Legend3[ping]: Max de la latence max
```

Il suffit ensuite de lancer la commande :

```
# mrtg /etc/mrtg.conf
```

On la placera dans un cron pour obtenir des courbes de statistiques régulières.

On pourra créer des courbes de statistiques en s'appuyant sur les nombreux outils disponibles sous Linux (sysstat, smartmontools, etc.)

Il existe de nombreux programmes évolués permettant de générer des courbes et statistiques. Citons Nagios[27], Cacti[28] et Ntop[29]. Le plus connu d'entre eux est certainement Nagios qui permet de surveiller de nombreux services (SMTP, POP3, HTTP, NNTP, PING, etc.) mais également les ressources (charge processeur, utilisation des disques, etc.). On peut visualiser les résultats, historiques des problèmes, journaux par interface web et obtenir des alertes personnalisées et écrire ses propres plugins pour des vérifications spécifiques. La mise en place de Nagios (ou d'un équivalent) pour un nombre de serveurs dépassant la dizaine est fortement conseillée.

Pour une surveillance système en direct, on peut utiliser GKrellM[30] en mode client-serveur : chaque serveur fait tourner un serveur gkrellmd et pour surveiller tous les serveurs, on démarre les clients sur un poste de travail. On peut également sécuriser les transmissions en encapsulant le trafic dans un tunnel SSH.

http://www.debian.org/doc/manuals/securing-debian-howto/index.fr.html
http://entreelibre.com/scastro/debian-secinst/debian-secinst.txt

27. http://www.nagios.org/
28. http://www.cacti.net/
29. http://www.ntop.org/
30. http://www.gkrellm.net/

3.12 Scripts shell

En informatique, on distingue les langages compilés et les langages interprétés. On peut considérer qu'un "script" est un programme écrit dans un langage interprété. Cela comprendra donc les shells (sh, csh, ksh, tcsh, bash, pdksh) les outils de manipulation de texte (sed, awk), Perl, Tcl, Ruby et Python.

La plupart de ces langages pourraient mériter une formation entière aussi nous nous concentrons sur quelques fonctionnalités intéressantes du shell :

cut Utilitaire qui sélectionne des sections sur chaque ligne d'un fichier sur la sortie standard.

```
cut -d " " -f 1 fichier
```

head Utilitaire qui renvoie les premières parties d'un fichier sur la sortie standard.

```
head -n 7 fichier
```

tail Utilitaire qui renvoie les dernières parties d'un fichier sur la sortie standard.

```
tail -f fichier
tail -n 7 fichier
```

sort Utilitaire qui trie les lignes d'un fichier sur la sortie standard.

```
sort -d fichier
```

tr Utilitaire pour convertir ou supprimer des caractères d'un fichier sur la sortie standard.

```
echo -e "plop\nplop" | tr -d "\n"
```

wc Utilitaire qui renvoie le nombre de lignes, fichiers ou octets d'un fichier.

```
wc -l fichier
```

grep Utilitaire renvoyant les lignes correspondant au modèle indiqué.

```
grep -i HtTp /etc/services
```

seq Utilitaire renvoyant une séquence de nombres.

```
seq 4 3 20
```

for Boucle

```
for i in 1 2 5; do echo $i; done
```

```
for i in ls .php; do mv $i $($i%%.php).html; done
```

3.13 Procédures de sauvegarde

La mise en place de procédures de sauvegarde nécessite souvent un audit précis pour évaluer le juste milieu entre le niveau de sécurité et le coût.
Prenons quelques exemples concrets.
Il est impensable qu'une entreprise, même petite, doive mettre la clé sous la porte en cas de petite castrophe naturelle (tempête, foudre, incendie).
Des sauvegardes dans un endroit physiquement différent sont donc obligatoires. À l'inverse, pour une petite entreprise, mettre en place des sauvegardes sur bande rapatriée dans un coffre-fort toutes les heures en fourgon blindé sera probablement disproportionné. Il faut donc évaluer chaque risque et le chiffrer. Les paramètres à prendre en compte sont :

- les risques
- le coût de mise en place et le coût régulier
- le temps d'administration de la solution
- le temps de redéploiement ou de recherche

Il existe principalement deux types de préventions :

- Prévention crash matériel
La solution est à choisir parmi la solution RAID, les sauvegardes système (disque/bande/périphérique amovible) et sauvegarde des fichiers système.

- Prévention erreurs logicielles ou humaines
La solution est de faire des sauvegardes régulières.

tar est l'outil le plus utilisé pour la sauvegarde.

Exemple :

```
tar -czvps --same-owner --atime-preserve backup.tar.gz /rep/
```

Table des partitions :

Sauvegarde : `dd if=/dev/hda of=NOM_FIC bs=512 count=1`

Restauration : `dd if=NOM_FIC of=/dev/hda bs=1 count=64 skip=446 seek=446`

Partimage[31] :

31. http://www.partimage.org/

Sauvegarde :

```
partimage -z1 -o -d save /dev/hda12 /mnt/backup/sav.partimg.gz
```

Restauration :

```
partimage restore /dev/hda12 /mnt/backup/sav.partimg.gz
```

L'outil rsync [32] est très puissant car il permet de mettre à jour en local ou à distance uniquement les fichiers modifiés.

client rsync <-> serveur rsynd

```
22/tcp    open  ssh
873/tcp   open  rsync
```

Options intéressantes :

`-v, --verbose`

-> plus verbeux

`-a, --archive`

-> mode archive (equivalent to -rlptgoD), ne préserve pas les liens hard

`-r, --recursive`

-> visite récursive des répertoires

`-l, --links`

-> copie les liens symboliques comme liens symboliques

`-p, --perms`

-> préserve les permissions

`-o, --owner`

-> préserve le propriétaire (root uniquement)

`-g, --group`

-> préserve le groupe

`-t, --times`

-> préserve les dates

`-S, --sparse`

-> traite les fichiers à trous efficacement

32. http://samba.anu.edu.au/rsync/

`-C, --cvs-exclude`

-> ignore automatiquement des fichiers, comme le ferait CVS

`--delete`

-> efface les fichiers qui n'existent pas du coté expédition

`--partial`

-> conserve les fichiers partiellement transférés

`--progress`

-> affiche la progression

`-z, --compress`

-> compresse les données

`-e ssh`

-> utilise ssh

`-D, --devices`

-> préserve les devices

L'astuce suprême consiste à utiliser des "liens hards" grâce à la commande `cp -al`
Un fichier est donc supprimé lorsqu'aucun lien matériel ne pointe vers lui :

```
mv backup.1 backup.2
cp -al backup.0 backup.1
rsync -e -a --delete source backup.0/
```

Chapitre 4

Apache

4.1 Rappel de l'architecture client/serveur

L'architecture client-serveur [1] se résume à la demande de services d'un programme client à un programme serveur. Il s'agit de l'extension logique du partitionnement des logiciels importants en modules donnant la possibilité de développement et de maintenance plus aisés. Les modules "demandeurs" sont appelés client et les modules appelés sont appelés service. Ainsi les différents modules fonctionnent sur des plateformes différentes et appropriées à leur fonction. Par exemple, les systèmes de gestion de base de données tournent sur des plateformes logicielles et matérielles conçues pour optimiser les requêtes, ou les serveurs de fichiers tournent sur des plateformes adaptées pour la gestion de fichiers.

Le client est donc un programme qui envoie un message à un programme serveur, demandant au serveur un service. Les programmes client sont en général constitués d'une interface permettant de valider les données entrées par l'utilisateur et d'un programme permettant de traiter et d'envoyer les requêtes aux programmes serveur.

Le programme contient donc un certain nombre de facilités pour interagir avec l'utilisateur. Ainsi, il accède aux ressources locales (écran, clavier, processeur, périphériques, etc.).
Un des éléments souvent présent sur une machine de type poste de travail est une interface graphique : GUI (Graphical User Interface).
Normalement, c'est le Windows Manager qui détecte les actions de l'utilisateur, gère les différentes fenêtres et affiche les données.

Le serveur est un programme qui répond aux demandes du client en réalisant la tâche demandée. Les programmes serveur recoivent en général des requêtes des programmes client, exécutent des requêtes et mises-à-jour sur une base de données, contrôlent l'intégrité des données et répondent aux programmes clients. Le programme serveur devrait être sur une machine indépendante sur le réseau mais souvent plusieurs programmes serveur sont sur la même machine et dans certains cas, la machine hébergeant le service est un poste de travail. Le programme serveur peut souvent accéder à des resources locales telles que les bases de données, imprimantes, interfaces et processeur(s).

1. http ://www.faqs.org/faqs/client-server-faq/

4.2 Le protocole HTTP

4.2.1 Différentes versions

HTTP/0.9 : première version du protocole HTTP, très simple, permettant uniquant une requête GET et une réponse sans méta-données.
HTTP/1.0 : ancienne version du protocole HTTP, encore utilisée par certains logiciels. Le serveur HTTP ferme encore la connexion dès qu'il a envoyé sa réponse.
HTTP/1.1 : version la plus répandue du protocole HTTP. Elle permet notamment les connexions persistantes, la négociation du contenu, et une meilleure gestion du cache.

4.2.2 Méthodes :

GET : requête d'une ressource
HEAD : requête uniquement des entête d'une ressource
POST : envoi de données à une ressource
Il existes d'autres méthodes moins utilisées (PUT, DELETE, TRACE, CONNECT)

4.2.3 Codes d'état :

- 1xx : Information (peu utilisé)
- 2xx : Succès, notamment le code 200 correspondant à OK
- 3xx : Redirection, notamment 301 (déplacement défintif) et 302 (déplacement temporaire)
- 4xx : Erreur du client, notamment 404 (non trouvé) et 403 (non autorisé)
- 5xx : Erreur du serveur, notamment 500 (erreur interne)

4.2.4 Champs d'entête :

- Allow
- Authorization
- Content-Encoding
- Content-Length
- Date
- Expires
- From
- If-Modified-Since
- Last-Modified
- Location
- Pragma
- Referer
- Server
- User-Agent
- WWW-Authenticate
- etc.

4.3 Présentation

Le logiciel Apache est un serveur HTTP. Apparu en 1995, il est dérivé de nombreux patches pour le serveur NCSA HTTPD [2]. Complètement réécrit, son nom serait tiré officieusement de l'appelation "a patchy server", c'est-à-dire un serveur fait de patches. La version officielle indique que le nom a été choisi en l'honneur de la tribu Apache, bien connue pour son sens aigu de la stratégie guerrière et pour son endurance. Dès 1996, il devenait le serveur HTTP le plus répandu sur Internet et sa popularité ne cesse de croître car en 1999, il était présent sur 57% des serveurs et en 2004, le chiffre atteind 67% [3].
La fondation Apache, Apache Software Foundation [4], a été créée en 1999 afin de soutenir le développement d'Apache mais aussi de nombreux autres projets orientés web (Jakarta, Spamassassin, etc.).
Apache est l'un des logiciels libres - sous licence Apache [5] souvent cité en exemple quand on parle des logiciels libres car il est notamment réputé pour sa sécurité et sa fiabilité.

Liens :
http://www.apache.org/
http://en.wikipedia.org/wiki/Apache_HTTP_Server

On distingue actuellement la version 1.x de la version 2.x qui comprend de nombreuses avancées telles qu'une nouvelle API, le support natif de l'IPv6 et la possibilité d'installation sur des plateformes non UNIX. Apache possède également de nombreux modules (CGI, Perl, PHP, authentification avancée, etc.) offrant des possibilités de mise en oeuvre de services complexes.

4.4 Installation

4.4.1 Compilation

Comme la plupart des logiciels libres, il est possible de compiler Apache à partir des sources. Cela permet de compiler uniquement avec les options que l'on a besoin et d'avoir des binaires bien adaptés à sa machine.
Pour la compilation en elle-même, on applique donc la procédure classique. On va reprendre en détail cette procédure.

On télécharge les sources mais également le hash MD5 des sources ainsi que la signature PGP (et les clés des développeurs Apache) de ces sources :

```
$ wget apache_x.y.z.tar.gz
$ wget apache_x.y.z.tar.gz.md5
$ wget KEYS
$ wget apache_x.y.z.tar.gz.asc
```

On vérifie le bon déroulement du téléchargement des sources en comparant le résultat des commandes suivantes :

2. http ://hoohoo.ncsa.uiuc.edu/
3. http ://news.netcraft.com/archives/web_server_survey.html
4. http ://www.apache.org/foundation/
5. http ://www.apache.org/licenses/

```
$ md5sum apache_x.y.z.gz
$ cat apache_x.y.z.tar.gz.md5
```

On importe les clés des développeurs Apache et on vérifie l'intégrité des sources :

```
$ gpg --import KEYS
$ gpg --verify apache_x.y.z.tar.gz.asc
```

On peut ensuite décompresser et désarchiver les sources :

```
tar -zxvf apache_x.y.z.tar.gz
cd apache_x.y.z.tar.gz
```

On prend ensuite connaissance des options qui s'offrent à nous grâce à la commande :

```
./configure --help
```

On distinguera les options d'administration (noms des répertoires, chemins des librairies, etc.). Par exemple :

```
--sysconfdir=/etc/apache2 --sbindir=/usr/sbin ;
```

Et les options relatives aux fonctionnalités, par exemple :

```
--with-mpm=worker --enable-ssl --enable-rewrite
--enable-cgi --enable-dav-fs --enable-dav
```

On aura bien sûr besoin de nombreuses librairies de développement pour compiler Apache (l'étape suivante sert bien sûr à vérifier leurs présences). Ensuite, on spécifie les options choisies avant de lancer l'étape de vérification :

```
./configure [options]
```

On compile :

```
make
```

Et on procède à l'installation :

```
make install
```

4.4.2 Paquets

Les paquets offrent plusieurs avantages sur la compilation à partir des sources. Ils permettent notamment de gagner du temps, et parfois de gérer les dépendances. On distinguera les paquets RPM[6], DEB[7], etc.

6. http ://www.rpm.org/
7. http ://www.debian.org/distrib/packages

Par exemple, sur un système Debian :

Pour Apache 2 :

```
apt-get install apache2-mpm-prefork
```

Paquets principaux :

apache2.2-common : modules de base, documentations et icones pour Apache

Plusieurs choix pour Apache MPM (Multi-Processing Module) :

apache2-mpm-worker : version par défaut. Adapté aux serveurs à fort trafic
apache2-mpm-prefork : implémentation "non-threaded" (similaire à l'historique Apache 1.3.x)
apache2-mpm-itk : similaire au prefork, avec la possibilité de préciser l'utilisateur et le groupe pour chaque VirtualHost

Dépendances directes :

libapr0 :	librairie "Apache Portable Runtime"
openssl :	librairies "Authentication abstraction"
ssl-cert :	surcouche pour générer des certificats
libldap2 :	librairies OpenLDAP
libgnutls11 :	librairies GNU TLS
libgcrypt11 :	librairies cryptographiques LGPL
libgpg-error0 :	librairie pour erreurs/messages composants GnuPG
liblzo1 :	librairies de compression LZO
libopencdk8 :	Kit "Open Crypto Development"
libtasn1-2 :	librairies structures ASN.1
zlib1g :	librairies de compression gzip
libsasl2 :	librairies SASL v2

4.5 Configuration

On vérifiera sa configuration grâce à la commande :

```
apache2ctl configtest
```

La configuration d'Apache 2 se trouve dans le répertoire /etc/apache2/
La configuration principale est dans le fichier apache2.conf

Note : Selon les systèmes (distributions Linux, BSD, etc.), cela peut varier : la commande peut être apachectl, le répertoire de configuration peut être /etc/httpd ou /usr/local/etc/apache22 par exemple, et la configuration peut être dans un fichier httpd.conf

Passons en revue quelques options à connaître grâce à un exemple de fichier de configuration.
Détaillons la première partie correspondant à l'environnement et aux modules :

```
### Section 1: Environnement

# mode d'execution du serveur : inetd ou standalone
ServerType standalone

# repertoire de configuration
ServerRoot /etc/apache2

# lock and PID file
LockFile /var/lock/apache.lock
PidFile /var/run/apache.pid

# temporisation pdt laquelle Apache attend temps total réception requête GET
# ou entre réception paquets TCP lors d'une requête POST ou PUT etc.
Timeout 300

# connexions persistentes
#KeepAlive On
# nombre de requêtes permises pour une connexion unique
# lorsque la directive KeepAlive est activée
#MaxKeepAliveRequests 100
# nombre de secondes pendant lesquelles Apache
# attendra une requête postérieure avant de rompre une connexion.
#KeepAliveTimeout 15

# nombre minimum de processus fils en attente qu'un serveur pourra conserver
#MinSpareServers 5
# nombre maximal de processus fils en attente
#MaxSpareServers 10
# nombre de processus fils créés dès le démarrage du serveur
#StartServers 5

# nombre limite de requêtes simultanées pouvant être acceptées par le serveur
MaxClients 150
# nombre limite de requêtes qu'un processus serveur fils peut traîter
MaxRequestsPerChild 100

# modules
LoadModule ...
LoadModule ...
LoadModule ...

# MIME
<IfModule mod_negotiation.c>
    LanguagePriority fr en da nl et de el it ja pl pt pt-br ltz ca es sv
```


```
</IfModule>

AddType application/x-httpd-php .html .php .php3
AddType application/x-httpd-php-source .phps
AddType application/x-tar .tgz
...

# avoir le maximum d'informations (mod_status)
<Location /server-status-0906>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    Allow from 1.2.3.4
</Location>
ExtendedStatus On
<Location /server-info-0906>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    Allow from 1.2.3.4
</Location>


# xxx
ReadmeName README
HeaderName HEADER
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

# redirection vers fichiers index (mod_dir)
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi index.php
</IfModule>

# Répertoires utilisateurs (mod_userdir)
<IfModule mod_userdir.c>
    #nom du répertoire public
    UserDir public_html
    #root n'a pas de site perso
    UserDir disabled root
</IfModule>

<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes FollowSymLinks IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
```


```
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

# navigateurs particuliers (mod_setenvif)
<IfModule mod_setenvif.c>
    BrowserMatch "Mozilla2" nokeepalive
    BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0
    force-response-1.0
    BrowserMatch "RealPlayer 4.0" force-response-1.0
    BrowserMatch "Java/1.0" force-response-1.0
    BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>
```

Détaillons maintenant la seconde partie :

```
### Section 2: configuration principale

#numéro du port
Port 80
#utilisateur et groupe propriétaire d'apache
User www-data
Group www-data

#adresse e-mail que le serveur peut inclure dans un message d'erreur
#retourné au client
ServerAdmin webmaster@domaine.tld

#nom d'hote (sert pour redirection)
ServerName www.example.com
ServerAlias example.com tmp.example.com

#Répertoire racine du serveur
DocumentRoot /var/www

#configuration par defaut
<Directory />
    #pas d'acces par defaut
    Order Deny,Allow
    Deny from all
    #possibilite de liens symboliques ssi liens et destinations
    #ont meme proprio
    Options SymLinksIfOwnerMatch
    #htaccess desactive
    AllowOverride None
</Directory>

<Directory /var/www/>
    Options Indexes Includes FollowSymLinks MultiViews
```


```
    AllowOverride AuthConfig FileInfo
    Allow from all
</Directory>

#HTACCESS si directive AllowOverride
AccessFileName .htaccess
<Files ~ " ^.ht">
    Order allow,deny
    Deny from all
</Files>

# resolution inverse double
#HostnameLookups Off
# Desactive version verbeuse
ServerTokens Prod
# ajoute une ligne contenant ServerName et ServerAdmin
# en bas des pages d'erreurs notamment
ServerSignature On

#CGI
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory /usr/lib/cgi-bin/>
    AllowOverride None
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>

# icones
Alias /icons/ /usr/share/apache/icons/
<Directory /usr/share/apache/icons>
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# forcer
#AddDefaultCharset UTF-8
```

Revenons sur certaines options :

Dans **<Directory ...></Directory>** :

- Gestion des accès

http://httpd.apache.org/docs-2.2/mod/mod_access.html

```
Order [option]
```


```
Allow/Deny from [nom de domaine, adresse IPv4/v6, réseau]
```

```
Order Deny,Allow
Allow from 10.1.0.0/255.255.0.0
Deny from all
```

- Options :
Options [options]
None : rien
MultiViews : rediriger les demandes selon les préférences du navigateur (mod_negotiated)
All : toutes les options ci-dessous
Indexes : lister le répertoire si il n'y a pas de fichier index (mod_index)
FollowSymLinks : suit les liens symboliques
SymLinksIfOwnerMatch : suit les liens symboliques ssi liens et destinations ont le même propriétaire
Includes : possibilité de filtres Server-side (mod_include)
IncludesNOEXEC : Includes mais sans scripts exécutables
ExecCGI : l'exécution de scripts CGI est permise (mod_cgi)
Possibilités de faire +/- [options] par rapport à une directive supérieure (répertoire contenant ou racine)

- AllowOverride :
AllowOverride [options]
Permet de spécifier certains paramètres dans des fichiers .htaccess :
AuthConfig : pour les directives d'authentification (Auth*, Require, etc.)
FileInfo : pour les directives de contrôle des types de fichier (DefaultType, ErrorDocument, SetHandler, etc.)
Indexes : pour les directives d'indexation de répertoire (DirectoryIndex, DefaultIcon, etc.)
Limit : permet de spécifier les directives de gestion d'accès (Allow, Deny, Order)
Options : permet de spécifier les options d'Options

Fichiers .htaccess
http://httpd.apache.org/docs-2.2/howto/htaccess.html

Exemple de fichier .htaccess :

```
AuthUserFile .htpasswd
AuthGroupFile /dev/null
AuthName "Acces reserve"
AuthType Basic
<LIMIT GET POST>
Require valid-user
</LIMIT>
```

Voir mod_auth
<Location ...></Location> est similaire à <Directory></Directory> à la différence que les directives sont valables sur les chemins d'URL

Exemple :

```
<Location /status>
SetHandler server-status
Order Deny,Allow
Deny from all
Allow from 192.168.176.53
</Location>
```

Dans **<Files ...></Files>** :

Directives portant sur les fichiers.

Exemple :

```
<Files ~".(mp3|ogg|avi|mpeg)\$">
    Order allow,deny
    Deny from all
</Files>
```

<Limit ... ></Limit> impose des restrictions sur certaines méthodes du protocole HTTP.
Exemple :

```
<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
    Order deny,allow
    Deny from all
</Limit>
```

Par défaut, la configuration d'Apache est souvent répartie dans plusieurs fichiers pour une meilleure gestion. Ainsi, sous Debian, on retrouve le partitionnement suivant, indiqué dans le fichier de configuration principal :

```
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf

Include /etc/apache2/httpd.conf

Include /etc/apache2/ports.conf

Include /etc/apache2/conf.d/[^.#]*
```

```
Include /etc/apache2/sites-enabled/[^.#]*
```

Les modules :

On trouve les répertoires mods-available et mods-enabled dans le répertoire de configuration. mods-available contient des fichiers NOM.load et NOM.conf : un fichier NOM.load contient la directive permettant le chargement d'un module disponible :

```
LoadModule /chemin/NOM.so
```

Le fichier NOM.conf contient les éventuelles options de configuration du modèle, par exemple :

```
<IfModule mod_NOM.c>
...
</IfModule>
```

Pour activer un module, on fait simplement un lien symbolique du fichier NOM.load (et NOM.conf si il existe) vers le répertoire mods-enabled.

Exemple :

```
# ln -s /etc/apache2/mods-available/ssl.load /etc/apache2/mods-enabled/ssl.load # ln -s
/etc/apache2/mods-available/ssl.conf /etc/apache2/mods-enabled/ssl.conf
```

Voir les modules

-Le fichier `httpd.conf` :

Utilisé pour les directives supplémentaires (vide par défaut)

-Le fichier `ports.conf` :

```
Listen 80
Listen IP:80
Listen domain.tld:80
```

La troisième possibilité est à éviter si possible [8]

4.5.1 VirtualHost

Le terme de VirtualHost se réfère à la pratique de faire tourner plusieurs sites Internet sur une

8. http ://httpd.apache.org/docs-2.2/dns-caveats.html

seule machine alors que l'utilisateur final ne se rend pas compte que les différents sites tournent physiquement sur la même machine.

Apache est capable d'avoir des VirtualHost basés sur les adresses IP et sur les noms. Attention, les ports d'écoute sont définis avec le paramètre Listen. Les VirtualHost ne font que "rediriger" les requêtes entrantes.

```
NameVirtualHost IP:*
NameVirtualHost *

<VirtualHost 10.1.2.3:>
ServerAdmin webmaster@host.foo.com
DocumentRoot /www/docs/host.foo.com
ServerName host.foo.com
ErrorLog logs/host.foo.com-error_log
TransferLog logs/host.foo.com-access_log
</VirtualHost>
```

Exemple complexe :

```
Listen IP1:80
Listen IP2:8080

NameVirtualHost IP1:80

<VirtualHost IP1:80>
DocumentRoot /www/ip1
ServerName www.name1.tld
</VirtualHost>

<VirtualHost IP1:80>
DocumentRoot /www/ip2
ServerName www.name2.tld
</VirtualHost>

#basé sur l'IP

<VirtualHost IP2:8080>
DocumentRoot /www/ip3
ServerName www.name3.tld
</VirtualHost>
```

À l'intérieur d'un VirtualHost, on peut spécifier de nombreuses directives. Souvent il s'agira de :

```
DocumentRoot
ServerAdmin
```


```
ServerName
ServerAlias
ErrorLog
TransferLog
LogLevel
CustomLog
ServerSignature
ErrorDocument
Rewrite*
etc.
```

Ainsi que le <Directory /></Directory> spécifiant les droits par défaut sur les répertoires concernés (DocumentRoot, script CGI, script Perl, icones, manuel, ...)

Voir dans la documentation, les paramètres pouvant s'appliquer dans un VirtualHost.

Lien :http://httpd.apache.org/docs/vhosts/

4.5.2 Configuration des sites en ligne

La configuration d'Apache fonctionne souvent avec des VirtualHost... même pour un seul site mis en ligne ! On trouve les répertoires *sites-enabled* et *sites-available* dans le répertoire de configuration. Par exemple, le fichier *default* :

```
NameVirtualHost *
<VirtualHost *>
ServerName www.example.com
ServerAlias example.com
        ServerAdmin webmaster@example.com
        DocumentRoot /var/www/

        <Directory />
                Order Deny,Allow
                Deny from all
                Options None
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
        </Directory>

        ErrorLog /var/log/apache2/error.log
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
        ServerSignature On
</VirtualHost>
```


Pour activer un site, on fait simplement un lien symbolique du fichier dans le répertoire sites-available vers le répertoire sites-enabled. Par contre, Apache passe en revue les liens du répertoire sites-enabled dans l'ordre alphanumérique/alphabétique. Il faut donc nommer les liens selon ses préférences. Ainsi, on créra un lien :
ln -s /etc/apache2/sites-available/default /etc/apache2/sites-enabled/000-default # a2ensite test

4.6 Modules

Un grand nombre de modules sont préinstallés. On cherchera les paquets des modules supplémentaires avec la commande :

```
apt-cache search ^libapache2-mod
```

4.6.1 mod_cgi

Lien : http://httpd.apache.org/docs-2.2/mod/mod_cgi.html

Ce module permet l'exécution de scripts CGI (les scripts CGI peuvent être écrits en C, Perl, Shell, etc.).

Configuration :

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory /usr/lib/cgi-bin/>
    AllowOverride None
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

Exemple :
date.cgi :

```
#!/bin/sh
tmp=‘/bin/date‘
cat << EndFile
Content-type: text/html

<HTML><HEAD><TITLE>Date du serveur</TITLE></HEAD>
<BODY>
<CENTER>
<H1>La date du serveur est</H1>
$tmp
</CENTER>
```

```
</BODY>
</HTML>

EndFile
```

4.6.2 mod_perl

Lien :http://perl.apache.org/

Ce module permet d'exécuter des scripts Perl. Il offre de nombreux avantages par rapport aux scripts CGI en Perl (rapidité, optimisation, etc.)

```
apt-cache search ^libapache perl
```

4.6.3 mod_php5

voir PHP

4.6.4 mod_auth

Lien :http://httpd.apache.org/docs-2.2/mod/mod_auth.html

Exemple :

```
AuthUserFile /var/apache/passwd/.htpasswd
AuthGroupFile /dev/null
AuthName "Accès reservé"
AuthType Basic
<LIMIT GET POST>
Require valid-user
</LIMIT>
```

Exemple :

```
mkdir /etc/apache2/pass
htpasswd -c /etc/apache2/pass/.htpasswd user1

htpasswd /etc/apache2/pass/.htpasswd user2
```

Souvent dans un fichier .htaccess

4.6.5 mod_proxy

Lien : http://httpd.apache.org/docs-2.2/mod/mod_proxy.html

Ce module implémente un proxy/cache pour Apache. Il gère les fonctionnalités de proxy pour FTP, CONNECT (pour SSL), HTTP/0.9, et HTTP/1.0.

4.6.6 mod_rewrite

Lien : http://httpd.apache.org/docs-2.2/mod/mod_rewrite.html

```
RewriteEngine On

RewriteCond

Variables :

RegEx :

^ : début
$ : fin
. : tous les caractères
* : nombre infini de fois

RewriteRule
```

Exemple : forcer le nom SERVER_NAME pour le serveur :

```
RewriteEngine On
RewriteLog  "/var/log/apache/rewrite.log"
RewriteLogLevel 3
RewriteCond %{HTTP_HOST} !^%www.domaine.tld$
RewriteRule ^/(.*) http://%{SERVER_NAME}/$1 [L,R]
```

4.6.7 mod_dav

Lien : http://httpd.apache.org/docs-2.2/mod/mod_dav.html

mod_dav et mod_dav_fs
dav_fs.conf :

```
DAVLockDB /var/lock/apache2/DAVLock/DAVLockDB
```


4.6.8 mod_ssl

SSLEngine On

Lien :http://httpd.apache.org/docs-2.2/mod/mod_ssl.html

On rappelle la procédure de génération d'un certificat auto-signé :

On crée une "demande" de certificat en se basant sur des paramètres aléatoires ainsi que sur une clé privée `privkey.pem` protégée par un mot de passe :

```
$ openssl req -new > demande.csr
```

Si l'on veut supprimer ce mot de passe de protection (utile dans le cas d'un serveur), on ajoute l'argument `-out cleprivee.pem` et l'on obtient une clé privée `cleprivee.pem` non protégée :

```
$ openssl rsa -in privkey.pem -out cleprivee.pem
```

Enfin, on génère le certificat basé sur la demande et signé par la clé privée :

```
$ openssl x509 -in demande.csr -out certificat.pem -req -signkey cleprivee.pem -days 365
```

On peut ajouter ensuite les lignes suivantes dans le VirtualHost :

```
#activation SSL
  SSLEngine on
#certificats
  SSLCertificateFile /path/to/certs/certificat.pem
#cle privee
  SSLCertificateKeyFile /path/to/certs/cleprivee.pem
```

Exemple 1 : avoir un site disponible avec HTTP et HTTPS

```
ports.conf :
Listen 80
Listen 443

sites-available/default :

NameVirtualHost *:80
<VirtualHost *:80>
...
</VirtualHost>

sites-available/default-ssl :
```

```
NameVirtualHost *:443
<VirtualHost *:443>
...
#SSL
  SSLEngine on
#certificats
  SSLCertificateFile /etc/apache2/ssl/certificat.cert
#cle privee
  SSLCertificateKeyFile /etc/apache2/ssl/cle-privee.key
</VirtualHost>
```

ln -s /etc/apache2/sites-available/default /etc/apache2/sites-enabled/000-default
ln -s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/001-default-ssl

Exemple 2 : avoir un site disponible en HTTPS et HTTP redirigé vers HTTPS

```
sites-available/default-ssl :
NameVirtualHost *:80
NameVirtualHost *:443

<VirtualHost *:80>
        RewriteEngine On
        RewriteRule ^/(.*) https://webmail.domain.tld/ [L,R]
</VirtualHost>

<VirtualHost *:443>
...
</VirtualHost>
```

Lien :http://home.earthlink.net/~fjhirsch/Papers/wwwj/article.html

4.6.9 autres options

UseCanonicalName : On|Off|DNS (défaut=On) Permet de spécifier que l'on se réfère à l'option ServerName pour déterminer les variables SERVER_NAME et SERVER_PORT

4.7 Optimisation

Lien : http://httpd.apache.org/docs-2.2/misc/perf-tuning.html

```
HostnameLookups off
<Files ~ ".(html|cgi)$">
```

```
HostnameLookups on
</Files>
```

`AllowOverride None` : partout où l'on peut (évite de chercher .htaccess partout)

`Options SymLinksIfOwnerMatch` à utiliser le moins possible (pas par défaut)

DirectoryIndex index.php index.html index.html index.cgi index.pl

4.8 Sécurité

Pour améliorer la sécurité d'Apache, on peut installer mod_security :

```
# aptitude install libapache2-mod-security2
```

Avec le fichier de configuration `conf.d/mod-security2.conf` resemblant à :

```
# enable mod_security
SecRuleEngine On
# access to request bodies
SecRequestBodyAccess On
#SecRequestBodyLimit 134217728
#SecRequestBodyInMemoryLimit 131072
# access to response bodies
SecResponseBodyAccess On
#SecResponseBodyLimit 524288
SecResponseBodyMimeType (null) text/html text/plain text/xml
#SecServerSignature "Apache/2.2.0 (Fedora)"

SecUploadDir /tmp
SecUploadKeepFiles Off

# default action
SecDefaultAction "log,auditlog,deny,status:406,phase:2,t:none"

SecAuditEngine RelevantOnly
#SecAuditLogRelevantStatus "^[45]"
# use only one log file
SecAuditLogType Serial
# audit log file
SecAuditLog /var/log/apache2/modsec_audit.log
# what is logged
SecAuditLogParts "ABIFHZ"

#SecArgumentSeparator "&"
SecCookieFormat 0
```

```
SecDebugLog /var/log/apache2/modsec_debug.log
SecDebugLogLevel        0

SecDataDir /tmp
SecTmpDir /tmp

#########
# RULES
#########

# File name
SecRule REQUEST_FILENAME "modsecuritytest1"
# Complete URI
SecRule REQUEST_URI "modsecuritytest2"
SecRule REQUEST_FILENAME "(?:n(?:map|et|c)|w(?:guest|sh)|cmd(?:32)?|telnet|rcmd|ftp)\.exe"
```

Afin de sécuriser les requêtes vers l'extérieur, il est recommandé d'installation un proxy tel que Squid. Voici le fichier squid.conf :

```
# ports
http_port 8888 transparent
icp_port 0

# ACL
acl all src 0.0.0.0/0.0.0.0
acl localhost src 127.0.0.1/255.255.255.255
acl INTERNE src 1.2.3.4/32 127.0.0.0/8
acl Safe_ports port 80          # http
acl SSL_ports port 443 563

acl WHITELIST url_regex "/etc/squid/whitelist.conf"

http_access deny !WHITELIST
http_access allow INTERNE
http_access deny all
```

Avec le fichier /etc/squid/whitelist.conf suivant :

```
http://.*debian.org/.*
http://zidane.evolix.net/.*
http://pub.evolix.net/.*
http://www.kernel.org/.*
http://pear.php.net/.*
http://.*akismet.com/.*
http://.*wordpress.org/.*
http://etc.inittab.org/.*
http://.*twitter.com/.*
http://feeds.feedburner.com/.*
http://feeds2.feedburner.com/.*
```


```
http://sync.openx.org/.*
http://oxc.openx.org/.*
http://code.openx.org/.*
http://pc.openx.com/.*
http://api.pc.openx.com/.*
http://bid.openx.net/.*
http://blog.openx.org/.*
http://forum.openx.org/.*
http://www.backports.org/.*
```

Pour l'activer, on ajoute les regles suivantes dans le firewall :

```
#HTTPSITES='0.0.0.0/0'
# Proxy
/sbin/iptables -t nat -A OUTPUT -p tcp --dport 80 -m owner --uid-owner proxy -j ACCEPT
/sbin/iptables -t nat -A OUTPUT -p tcp --dport 80 -d 1.2.3.4 -j ACCEPT
/sbin/iptables -t nat -A OUTPUT -p tcp --dport 80 -d 127.0.0.1 -j ACCEPT
/sbin/iptables -t nat -A OUTPUT -p tcp --dport 80 -j REDIRECT --to-port 8888
```

4.9 Surveillance

Apache génère donc les logs selon votre configuration. Généralement, on retrouvera les erreurs dans error.log et les logs des VirtualHost là où on veut :)

Analyse des logs : awstats, webalizer, scanerrlog, webdruid, vlogger

Outils :

ab - ApacheBench

```
ab -n 5000 -c 100 http://www.domaine.com/index.html
```

siege - outil de benchmark semblable à ab

tsung - outil de benchmark très puissant

apachetop - surveillance Apache en temps réel

Munin - surveillance notamment d'Apache via divers graphes

Installer le paquet libwww-perl et configurer mod_status pour assurer le bon fonctionnement des courbes.

awstats - analyse de logs

awstats.VHOST.conf :

```
LogFile="/var/log/apache/access.VHOST.log"
SiteDomain="www.VHOST.tld"
Lang="fr"
[...]
```


cron.d/awstats.VHOST

```
30 * * * * root [ -x /usr/lib/cgi-bin/awstats.pl -a -f
/etc/awstats/awstats.VHOST.conf -a -r /var/log/apache/access.VHOST.log ]
&& /usr/lib/cgi-bin/awstats.pl -config=VHOST -update >/dev/null
```

Voici un récapitulatif de nos préconisations d'installation : http://trac.evolix.net/infogerance/wiki/HowtoLAMP/Apache

Chapitre 5

PHP

http://www.php.net/
http://www.zend.com/

5.1 Présentation

Un danois, Rasmus Lerdorf [1], décidant de mettre en valeur sa page personnelle a créé une collection de scripts en Perl/CGI appelée PHPTools (Personel Home Page Tools). Réécrit en langage C, l'outil fut renommé PHP/FI (Forms Interpreter). PHP/FI version 2 peut s'insérer sous forme de module au serveur Apache et permet d'insérer directement des instructions dans du code HTML. En 1997, plusieurs développeurs s'associent à Rasmus Lerdorf pour sortir un an plus tard PHP 3 qui intégrait de nombreuses fonctionnalités comme le support de systèmes de gestion de base de données. PHP étant cette fois un acronyme récursif de "PHP : Hypertext Preprocessor". La sortie de PHP 4 en 2000 puis de PHP 5 en 2004 apporta de nombreuses fonctionnalités supplémentaires.
PHP est donc un langage de programmation Open Source principalement utilisé par les développeurs web pour créer des pages dynamiques. En effet c'est un langage de script exécuté du côté serveur générant principalement du HTML. Sa syntaxe empruntée aux langages C, Java et Perl est assez simple à apprendre.

5.2 Installation

5.2.1 Compilation

Comme la plupart des logiciels libres et Apache, il est possible de compiler PHP à partir des sources. Cela permet de compiler uniquement avec les options que l'on a besoin et d'avoir des binaires bien adaptés à sa machine :

```
./configure [options]
make
make install
```

1. http://www.lerdorf.com/

5.2.2 Paquets

Pour Debian GNU/Linux :
Paquets principaux :

php5-common : fichiers courants pour PHP
libapache2-mod-php5 : module mod_php pour Apache2

Dépendances directes :

libc6 :	librairies "GNU C"
libbz2-1.0 :	librairie de compression bzip2
libdb4.2 :	librairies "Berkeley v4.2 Database"
libexpat1 :	librairie "XML parsing C"
libpcre3 :	librairie Perl 5 Compatible Regular Expression
libssl0.9.7 :	librairies "SSL shared"
zlib1g :	librairies de compression gzip
mime-support :	support MIME (mime.types, mailcap)
apache2-mpm-prefork :	implémentation "non-threaded" (similaire à Apache 1.3.x)
libmagic1 :	librairie pour les types de fichiers utilisant les numéros magiques

5.3 Configuration

5.3.1 Fichier de configuration

La configuration par défaut de PHP fonctionne en général assez bien. Néanmoins un certain nombre de paramètres peuvent être configurés dans le fichier nommé php.ini (sa syntaxe rappelle en effet les fichiers INI d'applications Microsoft Windows). Il est important de se familiariser avec cette configuration pour pouvoir la modifier pour des questions de sécurité ou pour certaines applications.

`php.ini :`

```
; activer l'interpretation du Code PHP
engine = On
; active le support des balises courtes <? ... ?>
short_open_tag = On
; quantité maximale de memoire qu'un script peut reserver
memory_limit = 8M

; ajoute des restrictions et controle sur les scripts
; http://www.php.net/manual/fr/features.safe-mode.functions.php
; obsolete
;safe_mode = On

;  permet de ne changer que certaines variables d'environnement
```

```
safe_mode_allowed_env_vars = PHP_

; messages d'erreur non visibles
display_errors = Off

; duree maximale d'execution d'un script en seconde
max_execution_time = 30

; permet d'eviter l'execution de scripts distants
allow_url_fopen = Off

; ajoute des quotes pour empecher certaines injections sql
magic_quotes_gpc = On

; ne pas indiquer la presence de PHP dans entete HTTP
expose_php = Off

; enregistrer les erreurs
log_errors = On

; journal des erreurs
error_log = /var/log/php.log

; desactiver des fonctions dangereuses
disable_functions = exec, shell_exec, system, passthru, putenv, popen

; specifie si uploads possibles ou non
file_uploads = On

; la taille maximale des uploads
upload_max_filesize = 2M

; repertoire temporaire pour les uploads
upload_tmp_dir = /var/tmp/

; limiter les acces de PHP sur le systeme
open_basedir = /home
```

Lien : http://www.php.net/manual/fr/features.safe-mode.functions.php

Malgré l'intégration du moteur Zend (parseur de code), certains "optimiseurs" peuvent accélérer l'exécution de script PHP en optimisant code, cache, etc.
Zend Optimizer
Eaccelerator
APC

Les problèmes de sécurité en relation avec PHP proviennent bien souvent d'erreurs de code. Ils concernent donc principalement les scripts du type PHP-Nuke, PHPBB2, etc. Néan-

moins, PHP peut comporter des failles de sécurité. Outre l'utilisation des fonctionnalités disponibles dans la configuration de PHP, il existe des patches comme hardened-php qui permettent d'éviter certains problèmes. Le suivi des listes de sécurité et la mise-à-jour de PHP reste indispensable.

Liens :
http://www.hardened-php.net/
http://www.phpsecure.info

Exécution de scripts en mode de commande :
list.php

```
<?php echo "Hello World\n"; ?>

$ php /tmp/php.php
X-Powered-By: PHP
Content-type: text/html

bonjour
```

5.3.2 Utilisation avec Apache

Module mod_php5

LoadModule php5_module /usr/lib/apache2/modules/libphp5.so

```
<IfModule mod_php5.c>
AddType application/x-httpd-php .php .html .phtml .php5
AddType application/x-httpd-php-source .phps
</IfModule>
```

On vérifie avec le script PHP suivant :

```
echo "<?php phpinfo(); ?>" > /var/www/info.php
```

5.3.3 Gestion des sessions

Lorsque des sessions PHP sont utilisées (fonctions session_*() dans le code), des informations sont stockées côté serveur. Le navigateur conserve uniquement identifiant pour accéder à ces informations, stockés dans un cookie ou une variable du type PHPSESSID dans l'URL (cela tend à être obsolète).

Par défaut, ces informations sont conservées dans des fichiers sur le disque (un fichier par session) mais il est conseillé d'utiliser une méthode plus performante si vous avez un serveur dédié : monter simplement le répertoire des sessions pour un mono-serveur, ou déléguer la gestion des sessions à un service annexe (Memcached, Redis ou Kyoto Tycoon).

Pour plus de détails : http://trac.evolix.net/infogerance/wiki/HowtoLAMP/PHP#SessionsPHP

5.4 Base de programmation

Il existe de nombreuses extensions de PHP (MySQL, PostGreSQL, LDAP, FTP, IMAP, GD, XML, etc.) et également des bibliothèques dont la plus connue est PEAR (PHP Extension and Application Repository) qui comprend classes et librairies mais introduit également un style de codage.

Liens :
http://www.nexen.net/docs/php/
http://pear.php.net/
http://pecl.php.net/
http://www.lephpfacile.com/manual_pear/index.php
http://gtk.php.net/

Voici un récapitulatif de nos préconisations d'installation : http://trac.evolix.net/infogerance/wiki/HowtoLAMP/PHP

Chapitre 6

MySQL

Lien : http://www.mysql.com/

6.1 Présentation

MySQL est un serveur de base de données relationnelles SQL (Structured Query Language) multi-thread et multi-utilisateur. MySQL est soumis à la licence GPL (GNU General Public License) ou à une licence commerciale distribuée par l'entreprise Mysql AB (fondée par les développeurs de MySQL).

6.2 Installation

6.2.1 À partir des sources

Il est bien sûr possible de compiler MySQL à partir des sources. Cela permet de compiler uniquement avec les options que l'on a besoin et d'obtenir des binaires pour sa machine. On rappelle la procédure :

```
./configure [options]
make
make install
```

6.2.2 Par paquet

Dans la plupart des cas, on utilisera des paquets pour installer MySQL.

Paquets principaux :

mysql-common : fichiers communs pour MySQL
php5-mysql : module MySQL pour PHP

Dépendances directes :

libc6 : librairies "GNU C"
libmysqlclient10 : librairie client mysql
zlib1g : librairies de compression gzip
php4-common : fichiers courants pour php4

6.3 Administration

On note bien que les utilisateurs MySQL n'ont rien à voir avec des utilisateurs UNIX.
La première fois, en root, taper :

```
# mysql
```

On obtient alors un message ressemblant à :

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 133 to server version: 3.23.49-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Vous êtes donc connecté à votre serveur MySQL.
Déconnectez vous rapidement :

```
mysql> quit
Bye
```

On retrouve la main en ligne de commande.
Il faut ABSOLUMENT mettre un mot de passe à root (ou bien lui supprimer les droits, voir ci-après) :

```
# mysqladmin password "motdepasse"
```

On peut maintenant se connecter a la base mysql en root sous :

```
#mysql -p
Enter password:*****
mysql>
```

Il est conseillé de créer un utilisateur MySQL nommé adminmysql avec tous les droits pour éviter d'utiliser root.

La gestion des permissions des utilisateurs (en fonction de l'hôte de connexion) se trouve donc dans la table "user" de la base mysql.

```
| Host            | char(60) binary
| User            | char(16) binary
| Password        | char(16) binary
| Select_priv     | enum('N','Y')
| Insert_priv     | enum('N','Y')
| Update_priv     | enum('N','Y')
| Delete_priv     | enum('N','Y')
| Create_priv     | enum('N','Y')
| Drop_priv       | enum('N','Y')
| Reload_priv     | enum('N','Y')
| Shutdown_priv   | enum('N','Y')
| Process_priv    | enum('N','Y')
| File_priv       | enum('N','Y')
| Grant_priv      | enum('N','Y')
| References_priv | enum('N','Y')
| Index_priv      | enum('N','Y')
| Alter_priv      | enum('N','Y')
```

La gestion des droits des utilisateurs sur des bases de données est opérée dans la table db :

```
| Host            | char(60) binary
| Db              | char(64) binary
| User            | char(16) binary
| Select_priv     | enum('N','Y')
| Insert_priv     | enum('N','Y')
| Update_priv     | enum('N','Y')
| Delete_priv     | enum('N','Y')
| Create_priv     | enum('N','Y')
| Drop_priv       | enum('N','Y')
| Grant_priv      | enum('N','Y')
| References_priv | enum('N','Y')
| Index_priv      | enum('N','Y')
| Alter_priv      | enum('N','Y')
```

D'autres tables (host, tables_priv, columns_priv) permettent d'ajuster plus finement les droits...

Donc pour créer un utilisateur `adminmysql` qui a tous les droits, on se connecte avec un utilisateur qui a les droits nécessaires :

```
mysql> use mysql
mysql>  GRANT ALL PRIVILEGES ON *.* TO adminmysql@localhost
-> IDENTIFIED BY 'mot2passe' WITH GRANT OPTION;}
```

On vérifie :

```
mysql> select * from user where User='adminmysql';}\\
```

On pourrait même insérer des utilisateurs "manuellement" :

```
INSERT INTO user VALUES ('localhost','login',password('motdepasse'),
'N','N','N','N','N','N','N','N','N','N','N','N','N','N','N');
```

Lors d'interventions sur ces tables de gestion des droits MySQL, on pensera à lancer la commande :

```
mysql>  FLUSH PRIVILEGES;
```

Ou encore effacer des utilisateurs :

```
mysql> delete from user where Host='vilain';}
```

On effacera les entrées que l'on ne désire pas dans cette table en fonction de sa politique de sécurité. On notera sous Debian la présence d'un utilisateur `debian-sys-maint` qui sert à certains scripts Debian : il ne doit pas être supprimé ! Au niveau sécurité, le mot de passe est généré à l'installation (stocké dans `/etc/mysql/debian.cnf`) par la commande :

```
perl -e 'print map{("a".."z","A".."Z",0..9)[int(rand(62))]}(1..16)'
```

Notons qu'il est bien sûr primordial d'interdire la lecture du fichier `/etc/mysql/debian.cnf` Bref, pour administrer notre base MySQL, on peut désormais éviter de se connecter en root et faire :

```
 # mysql -u adminmysql -p
Enter password:*****
mysql>
```

Souvent la principale action à faire par un administreur MySQL est de créer une base de données et un utilisateur ayant tous les droits dessus. Ceci est assez courant pour faire une page en PHP/Mysql ou intégrer un logiciel "tout fait" (forum, chat, etc.)
Pour faire cela, nous allons donc faire en tant qu'adminmysql :

```
mysql> create database forum;
mysql> grant all privileges on forum.* to moderateur@localhost
mysql> identified by 'mot_de_passe_du_moderateur';
```

Pour créer un utilisateur en lecture seule sur une base donnée :

```
mysql> INSERT INTO user (Host,User,Password) VALUES('localhost','readprod',PASSWORD('xxx'))
mysql> insert into db (Host,Db,User,Select_priv) values ('localhost','production','readprod',
```

6.4 Sauvegarde

Il existe diverses méthodes de sauvegarde. On peut par exemple sauvegarder le répertoire contenant les bases : /var/lib/mysql par défaut sous Debian. Il existe également plusieurs outils dont le plus utilisé est le programme `mysqldump`[1]

Sauvegarde d'une base :

```
$ mysqldump --opt BASE -u mysqladmin -p > database.sql
$ mysql -u mysqladmin -p  BASE < database.sql
```

Sauvegarde générale :

```
$ mysqldump --opt --all-databases -u mysqladmin -p > all_databases.sql
$ mysql -u mysqladmin -p < all\_databases.sql
```

Lien : http://www.nexen.net/docs/mysql/annotee/backup.php

6.5 Base de programmation

Voici quelques commandes SQL de base :

```
mysql> use essai ;
```

-> utiliser la base de données essai

```
mysql> create table test (nom type [options],..);
```

-> créer une table nommé test

Exemple :

```
CREATE TABLE clients_tbl (id INT not null AUTO_INCREMENT, prenom VARCHAR
(50) not null , nom VARCHAR (50) not null , ne_le DATE not null ,
ville VARCHAR (90) not null , enfants INT not null , PRIMARY KEY (id))
```

```
mysql> alter table test add nom type [-options];
```

-> ajouter un champ à une table

```
mysql> alter table test drop nom;
```

-> effacer un champ à une table

```
mysql> desc test;
```

1. http://www.nexen.net/docs/mysql/annotee/mysqldump.php

-> décrire la table test

```
mysql> insert into test values ('a','b',...);
```

-> remplir la base de donnée

```
mysql> select * from test;
```

-> sortir tous les champs de la table test

```
mysql> delete from teste where (condition)
```

-> effacer des données

```
mysql> delete from test;
```

-> effacer toutes les données de test

```
mysql> drop table test ;
```

-> effacer la table test
Liens :
http://nexen.net/docs/mysql/
http://www.mysql.com/

6.6 Autres SGBD libres

Il existe PostGreSQL[2], moins répandu que MySQL sur le web, mais qui propose certaines fonctionnalités plus avancées (on le cite parfois en tant qu'alternative à Oracle[3]). SQLite[4] est un autre système de gestion de base données libre. SQLite est très léger et stocke ses bases dans des fichiers binaires, il est intéressant (plus rapide, moins lourd que les autres SGBD, etc.) pour des applications n'opérant que de simples requêtes (select, insert etc.).

En conclusion, voici notre documentation avancée à propos de MySQL : http://trac.evolix.net/infogerance/wiki/HowtoMySQL

2. http://www.postgresql.org/
3. http://www.oracle.com/
4. http://www.sqlite.org/

Chapitre 7

Redis

Site officiel : http://redis.io/

7.1 Présentation

Redis est un jeune projet (première version sortie en 2009) de système de gestion de bases de données no-SQL (*Not Only SQL*) de type clé-valeur. Il est écrit en C et son but est d'être hautement performant.
Les données stockées peuvent être des chaines de caractères, tableaux, listes, etc. . .

Une caractéristique notable de Redis est que tout est stocké dans la RAM. Des synchronisations sont faites de manière régulière sur le disque afin de rendre les données persistantes.

7.2 Installation

Sous Debian Squeeze, la version 1.2.6 est présente dans les dépôts, et peut être installée simplement :

```
# aptitude install redis-server
```

Cependant, il peut être utile d'avoir une version plus récente, et on peut utiliser un backport du paquet qui fourni la version 2.4.2. Pour cela, si ce n'est pas le cas, il faut ajouter le dépôt backport au `sources.list` :

```
deb http://backports.debian.org/debian-backports squeeze-backports main
```

Pour s'assurer d'avoir les mises à jour ultérieures sur le paquet `redis-server`, il est nécessaire de rajouter dans le fichier `/etc/apt/preferences.d/redis` :

```
Package: redis-server
Pin: release a=squeeze-backports
Pin-Priority: 999
```

7.3 Configuration

La configuration de Redis se fait dans le fichier `/etc/redis/redis.conf`, dont voici un exemple :

```
daemonize yes
pidfile /var/run/redis.pid
port 6379
unixsocket /var/run/redis/redis.sock
bind 127.0.0.1
timeout 300
loglevel notice
logfile /var/log/redis/redis-server.log
databases 16
save 900 1
save 300 10
save 60 10000
dbfilename dump.rdb
dir /var/lib/redis
#requirepass <password>
maxclients 128
maxmemory 104857600
```

7.4 Sécurité

Il faut bien faire attention à restreindre l'accès aux données stockées dans les bases de données Redis. Par défaut, aucune authentification n'est configurée.

Redis peut être configuré pour écouter soit sur un socket réseau (par défaut `127.0.0.1:6379`) soit via un socket unix. Dans le second cas, il est aisé de positionner les bons droits unix sur le socket pour restreindre l'accès à tout le monde. Pour cette raison, il est préférable de faire écouter Redis sur un socket unix.
Si le choix ne peut se poser et que Redis doit obligatoirement écouter sur un socket réseau (en local ou non), il est important de configurer une authentification sur les bases de données.

Redis implémente une couche d'authentification extrêmement minimaliste : pas de gestion de comptes, mais un unique mot de passe défini en clair dans le fichier de configuration de Redis. Il sera alors nécessaire, après chaque connexion à la base, d'exécuter la commande Redis `AUTH` suivie du mot de passe.

Étant donné que Redis est capable de traiter un nombre très élevé de requêtes par seconde, les attaques par brute-force se font plus rapide et il est donc conseillé de définir un mot de passe très long.

7.5 Utilisation

Voici un aperçu de l'utilisation de Redis :

- Connexion au serveur (via un socket unix) :
  ```
  $ redis-cli -s /var/run/redis/redis.sock
  ```
- Ajout d'une entrée :
  ```
  redis> set foo 3
  OK
  ```
- Ajout de plusieurs entrées :
  ```
  redis> mset un 1 deux 2 trois 3 quatre 4
  OK
  ```
- Récupération d'une entrée :

```
redis> get foo
(nil)
```

– Listage des clés :

```
redis> keys *
1) "un"
2) "foo"
3) "deux"
4) "trois"
5) "quatre"
```

– Récupération des entrées qui contiennent *r* :

```
redis> *keys *r*
1) "quatre"
2) "trois"
```

7.6 Sauvegardes

Redis sauvegarde régulièrement le contenu de sa base de donnée (en RAM) sur le disque, dans le seul fichier `/var/lib/redis/dump.rdb` (par défaut). Il suffit donc de copier ce fichier pour en faire une sauvegarde.

La restauration consiste à éteindre Redis, copier le fichier sauvegardé à son bon emplacement, et redémarrer Redis.

7.7 Réplication

7.7.1 Fonctionnement

Redis supporte la réplication master-slaves, avec quelques caractéristiques intéressantes : un master peut avoir plusieurs slave, et un slave peut être master d'un autre slave (ce qui permet de faire de la réplication cascadée).

Après sa mise en place, la réplication se passe en deux temps. Premièrement, le master va envoyer l'intégralité de sa base de données au(x) slave(s). Les données seront alors créés sur le disque, puis chargées en mémoire. Une fois que les données sont identiques entre le master et le(s) slave(s), le master transmet au(x) slave(s) les modifications qui sont effectuées. Les commandes transmises sont les mêmes que pour interagir avec le master.

7.7.2 Configuration

La mise en place de la réplication est extrêmement simple : il suffit de rajouter la directive `slaveof <IP> <port>` dans la configuration de Redis, en prenant bien sûr soin d'adapter l'IP et le port du master.

Si le master demande une authentification (ce qui devrait être le cas si il est correctement configuré), il faut ajouter la directive `masterauth <password>`, avec le mot de passe du master.

Chapitre 8

Varnish

Site officiel : https://www.varnish-cache.org/

8.1 Présentation

Varnish est un reverse proxy HTTP dans le but premier est la mise en cache de contenu. Il est également capable de gérer plusieurs backend, avec répartition de charge et détection de panne.

Varnish, développé en C, se concentre principalement sur la performance sur des infrastructures à haut et très haut trafic.

Un autre point fort est son langage de configuration, qui permet de paramétrer finement le comportement de Varnish aux différentes étapes du traitement de la requête.

Le développement de Varnish a commencé en 2005, et il est distribué sous licence BSD.

8.2 Installation

Varnish est disponible dans les dépôts de Debian Squeeze en version 2.1.3. Il existe également un backport du paquet de Wheezy, qui fournit la version 3.0.2. Cette version apporte de nombreuses amélioration et fonctionnalité dans la gestion du load-balancing entre les backends.

Installation du paquet :

```
# aptitude install varnish
```

8.3 Configuration

Les possibilités offertes pour la configuration de Varnish sont assez vastes, elles seront abordés par grands thèmes.

8.3.1 Paramétrage de base

Tout d'abord, il est nécessaire de renseigner quelques informations de base au démon `varnishd`. Cette configuration se passe dans le fichier `/etc/default/varnish`. Plusieurs cas de figure sont proposés à titre d'exemple dans ce fichier, en voici un autre avec quelques optimisations supplémentaires :

```
DAEMON_OPTS="-a 192.0.2.1:80 \
             -T localhost:6082 \
             -f /etc/varnish/default.vcl \
             -S /etc/varnish/secret \
             -s malloc,3G
             -s file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,10G
             -p thread_pools=<Number of CPU cores>
             -p thread_pool_add_delay=2
             -p thread_pool_max=5000"
umask 022
```

Et voici quelques explications sur les paramètres :

`-a 192.0.2.1:80`
: Il s'agit du couple IP,port sur lequel Varnish attendra les requêtes HTTP à traiter.

`-T localhost:6082`
: Il s'agit du couple IP,port sur lequel sera accessible l'interface d'administration de Varnish (traité plus loin dans ce chapitre).

`-f /etc/varnish/default.vcl`
: Cette option indique le fichier de configuration à utiliser.

`-S /etc/varnish/secret`

`-s malloc,3G`

`-s file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,10G`
: On indique ici qu'une partie du cache sera stocké en mémoire 3 Go, ainsi que dans un fichier plat sur le disque, qui sera limité à 10 Go.

`-p thread_pools=<Number of CPU cores>`

`-p thread_pool_add_delay=2`

`-p thread_pool_max=5000`
: L'option `-p` permet de modifier différents paramètres d'exécution. De nombreux paramètres peuvent être modifiés, la liste complète avec leur description se trouve ici : https://www.varnish-cache.org/docs/2.1/reference/varnishd.html.

  `thread_pools` indique le nombre de groupe de threads à lancer. Cette valeur ne devrait pas dépasser le nombre de cœur disponible sur le système (pour des raisons de performance). Pour `threa_poo_ad_delay`, il s'agit du temps en milisecondes à attendre avant la création d'un nouveau thread. Et enfin `threa_poo_max` représente le nombre total de thread maximum à ne pas dépasser, tout pool confondus.

`umask 022`
: Varnish s'attend à avoir un umask à 022 pour s'exécuter correctement. Étant donné qu'il n'est pas forcé dans le script d'init, nous le plaçons ici manuellement.

8.3.2 Aperçu de la syntaxe du langage VCL

8.3.3 Gestion du cache

En se positionnant entre le client et le serveur applicatif, Varnish permet de lire et surcharger si besoin les entêtes HTTP de contrôle du cache. Par défaut, celle ci sont lues et pris en compte, mais on peut redéfinir le comportement dans la configuration.

Voici quelques exemples d'utilisation typique :

Forcer le TTL pour certains contenu

```
sub vcl_fetch {
    if (req.url ~ "\.(png|gif|jpg)$") {
        set beresp.ttl = 5d;
        set beresp.http.magicmarker = "1";
    }
}

sub vcl_deliver {
    if (resp.http.magicmarker) {
        unset resp.http.magicmarker;
        set resp.http.Age = "0";
      }
}
```

`beresp.http.magicmarker` permet de marquer l'objet pour pouvoir ensuite remettre son age à 0 (dans `vcl_deliver`.

Pour que le changement de TTL le soit également coté client, on réécrit le header HTTP *Cache-Control* en ajoutant (dans le premier `if` :

```
set beresp.http.cache-control = ‘‘max-age=432000’’;
```

Indiquer si un objet provient du cache ou pas dans les headers HTTP Dans un but de debugage, il peut être intéressant d'indiquer si un contenu provient du cache de Varnish ou non. Cela se fait simplement comme ceci :

```
sub vcl_deliver {
    if (obj.hits > 0) {
        set resp.http.X-Cache = "HIT";
    } else {
        set resp.http.X-Cache = "MISS";
    }
}
```

8.3.4 Gestion du load-balancing

Tout d'abord, il faut définir au moins un backend pour que l'ensemble puisse fonctionner correctement. Cela se fait à l'aide de la directive `backend`, comme ceci :

```
backend www00 {
    .host = "192.0.2.8";
    .port = "80";
}

backend www01 {
    .host = "192.0.2.14";
    .port = "80";
}
```

Il est ensuite possible de grouper ces backends dans un cluster, appelé *director* dans le langage de Varnish :

```
director baz round-robin {
    { .backend = www00; }
    { .backend = www01; }
}
```

Et enfin, on indique dans quel cas il sera utilisé (dans l'exemple il sera utilisé dans tous les cas, pas de condition) :

```
sub vcl_recv {
    set req.backend = baz;
}
```

Il s'agit ici de la configuration la plus simple possible. Maintenant, il peut être intéressant d'ajuster certains paramètres :

- Dans l'exemple ci dessus, le director est en mode round-robin. Le trafic est alors réparti équitablement entre les backend. On peut définir un «poids» pour chacun des backends, afin de jouer sur la répartition du trafic entre eux :

  ```
  director baz random {
      {
          .backend = www00;
          .weight = 6;
      }
      {
          .backend = www01;
          .weight = 4;
      }
  }
  ```

  Pour cela, on change le mode du director pour *random*.
- Une directive importante est `.max_connections`. Elle permet de limiter le nombre de connexions concurrentes envoyées sur un backend. En en positionnant une sur chacun des backends, Varnish saura qu'il devra ignorer le backend saturé et en choisir un autre, afin de ne pas le surchargé.

  ```
  backend www00 {
      .host = "192.0.2.8";
      .port = "80";
      .max_connections = 80;
  }
  ```
- Il est possible également de répartir les requêtes sur les backends suivant des critères sur la requête. Le mode du director à utiliser est alors *client* :

  ```
  director baz client {
      { .backend = www00; }
      { .backend = www01; }
  }

  sub vcl_recv {
     set req.backend = baz;
     set client.identity = req.ip;
  }
  ```

  Dans l'exemple ci-dessus, le critère utilisé est l'IP du client (`client.identity = req.ip`. Les autres critères possibles sont le user-agent (`req.http.user-agent`), l'URL (`client.url`) ou encore un cookie de session (`req.http.cookie`).

8.3.5 Gestion du failover

probe sur les backend saint et grace mode

8.4 Administration

8.5 Gestion des logs

Chapitre 9

À propos de ce document

Ce support de formation s'inspire de documentations officielles, de pages Internet, de livres ou de magazines soumis à des droits d'auteurs. Dans la mesure du possible, les liens vers les sources ont été cités.

Copyright (c) 2004,2005,2006,2007,2008,2009,2010,2011,2012 Evolix, Grégory Colpart, Sébastien Dubois, Romain Dessort et Alexandre Anriot

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.2 ou ultérieure publiée par la Free Software Foundation ; ce document ne comporte pas de section inaltérable. Cette licence est disponible à l'adresse suivante :
http://www.gnu.org/copyleft/fdl.html