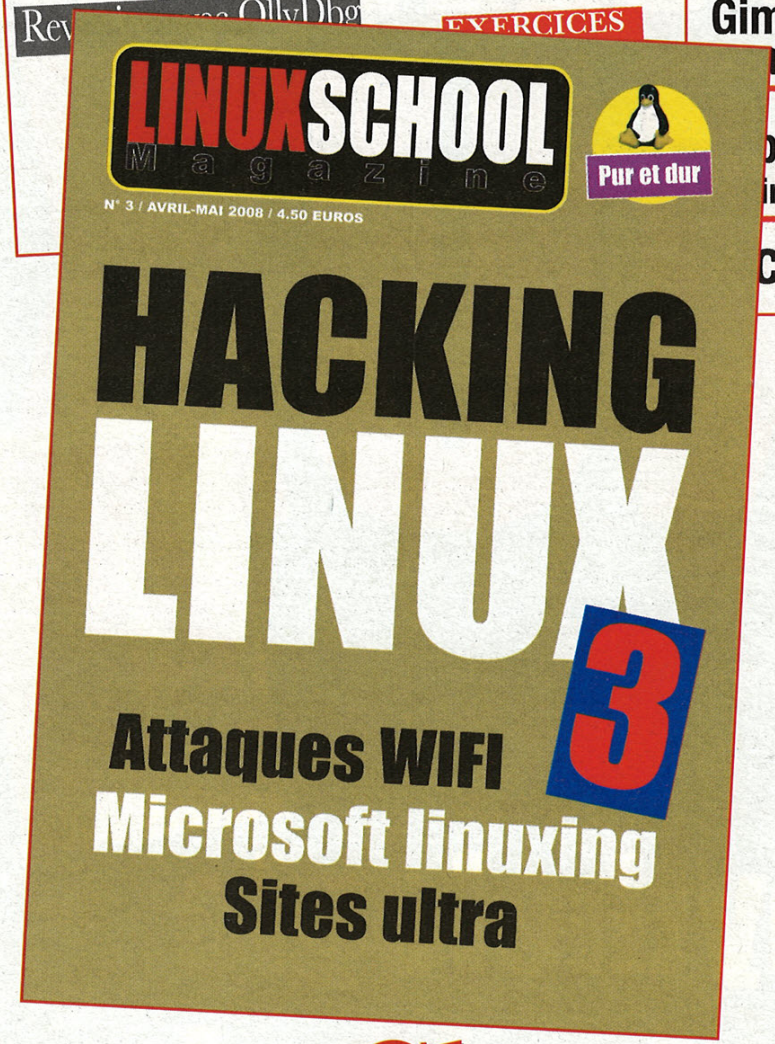
LINUX SCHOOL
Magazine

Pur et dur

N° 4 / JUILLET-AÖUT 2008 / 4.50 EUROS

# HACKING LINUX 4

## Tools
## Virus mythologie
## Plugins for hack



# Les rebelles sont de retour !

## Chez votre marchand de journaux

# Au programme



LINUXSCHOOL MAGAZINE est édité par LPN 15 RUE CHEVREUIL - 94 700 MAISONS-ALFORT

**Rédaction en chef :** Linux Community • **Directeur de Publication et représentant légal :** André Olivier

Imprimé en France par ROTO GARONNE 47310 Estillac • La Rédaction accepte toutes les contributions de la Communauté

Commission paritaire en cours • Dépôt légal à parution • ISSN en cours • 

# Télécharger sous Linux comme sous Windows !

Le problème de Linux, c'est le manque de logiciels… vous avez surement déjà entendu cet argument de la bouche d'un windowsien pour ne pas franchir le pas ! Et bien, en ce qui concerne le téléchargement, il est tout à fait possible d'utiliser Linux comme un Windows ! La preuve …

## Le grand eMule s'appelle aMule !

**aMule** est un client P2P multiplateformes (Linux, FreeBSD, Windows, Mac0S X, etc.) pour les réseaux edonkey2000 (ED2K) et eMule Kademlia (Kad). Il reprend tout les concepts de base du célèbre client Windows eMule. Les fonctionnalités d'aMule sont pratiquement les mêmes que celles d'eMule. L'extraordinaire succès du client eMule pour Windows trouve plusieurs explications : Ce client P2P est en effet recommandé pour l'échange de fichiers volumineux, il faut aussi avouer que comparer à certains de ses concurrents, il est fiable. Le programme ne contient ni spyware ni code malicieux. aMule en fait de même !

aMule est donc un client P2P pour les réseaux ED2K et Kad, chaque serveur de ce réseau est connecté aux autres serveurs du réseau.

Lorsqu'un ordinateur, le vôtre par exemple, se connecte à un serveur, ce serveur indexera les fichiers que vous aurez mis en partage. Il en va de même pour l'ensemble des ordinateurs connectés au réseau.

Lorsque vous souhaitez télécharger un fichier « toto », vous envoyez votre demande au réseau. Celui-ci vous répondra et vous indiquera la fréquence de la présence du fichier « toto » sur les ordinateurs connectés au réseau.

Cette fréquence s'appelle la disponibilité. Plus un fichier est disponible, plus vous aurez de sources et plus vous aurez la chance de le récupérer rapidement.

Une fois que vous aurez recherché votre fichier « toto », et que vous aurez double-cliqué dessus, le fichier apparaîtra dans l'onglet Transferts d'aMule.

Le serveur sur lequel vous êtes connecté va rechercher des « sources » pour télécharger votre fichier. Autrement dit, le serveur va rechercher les autres ordinateurs qui disposent de votre fichier « toto » et va envoyer une requête de connexion à ces ordinateurs.

Chaque ordinateur connecté au réseau dispose d'une file d'attente; vous ne commencerez à effectivement télécharger le fichier qu'une fois la file d'attente devant vous écoulée.

### Le « Low ID » et le « High ID » ?

Vous l'aurez sûrement remarqué, il s'agit d'un nombre, et si vos paramètres de connexion sont corrects, vous aurez tout simplement le bon nombre. Un problème de port, un pare-feu activé et non paramétré pour aMule, un problème avec votre routeur, et le serveur ne vous attribuera pas la bonne valeur, vous serez en Low ID.

Vous devez être impérativement en High ID, sinon, dans un premier temps, vous ne téléchargerez pas correctement, mais dans un second temps, et c'est surtout cela qui sera gênant, vous serez banni par les serveurs. Pour vous assurer que vous êtes en High ID, vérifiez que les flèches de la petite icône constituée d'une planète en bas à droite d'aMule, soient de couleur verte. Sinon, revérifier la configuration de vos équipements réseau !

### Kademlia

Le protocole Kademlia est inclus depuis la version 2.1.0 sur aMule. Il permet de ne plus utiliser des serveurs centralisés. Ainsi, il est possible de se connecter sur un nœud du réseau le plus proche, effectuer des recherches, télécharger et uploader sans avoir besoin de rester connecté à un serveur.

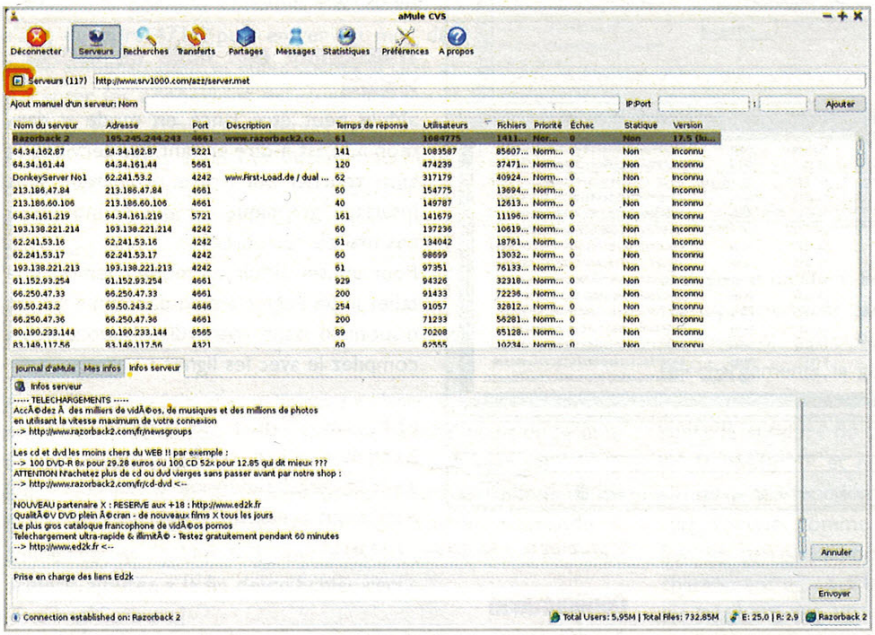
Kademlia utilise le port UDP 4672 pour fonctionner, pensez donc l'ouvrir dans votre pare-feu et à éventuellement configurer votre routeur. Il est possible d'utiliser les protocole ED2K et Kademlia simultanément comme séparément.

Pour activer Kademlia, rendez-vous dans « Préférences » puis

« Connexion » et enfin cochez la case nommée « Kademlia » dans la partie « Réseaux ».
Reste à se connecter comme d'habitude en cliquant sur le bouton « Se connecter », si tout est correctement configuré, le statut (KAD : ok) s'affiche en bas à droite.
Kademlia ne permet les connexions « anonyme » et vos données transitent en clair sur le réseau, tout comme avec le réseau ED2K.

## Installation et configuration d'aMule

Pour installer aMule, utilisez le gestionnaire de package de votre distribution (apt-get, emerge, yum…) ou rendez vous à l'adresse http://www.amule.org et dans la section nommée « download » à gauche du site, cliquez sur « Linux/*BSD etc. », cliquez ensuite sur le lien « download » et sélectionnez votre distribution dans la liste proposée. Il y a peu de chances que votre distribution ne soit pas présente. Pour les aficionados de la compilation, vous pouvez également récupérer le code source d'aMule en cliquant sur le lien « Sourcecode » de la section « download » sur la page d'accueil.
Maintenant qu'aMule est correctement installé, lancez le soit depuis un shell en tapant « aMule », soit directement depuis les menus si vous êtes passé par le système de package de votre distribution pour l'installation.




Au premier démarrage du logiciel, vous arrivez sur la fenêtre relative au serveur (correspondant à l'icône réseau et à l'onglet ED2K). La première chose à faire si vous souhaitez télécharger correctement est de commencer par mettre à jour votre liste de serveur EDK2, en cliquant sur le bouton représentant un triangle à coté de « Serveurs ».
Il est possible d'utiliser des fichiers de liste de serveurs disponible sur internet et mis à jour régulièrement. Rendez-vous donc sur : http://ed2kmet.x24hr.com/pl/slist.pl?download/server-max.met, http://www.gruk.org/server.met.gz ou encore http://www.emule-inside.net/files/server.met.
Une fois la liste des serveurs mis à jour, il ne vous reste plus qu'à vous connecter à un serveur en double-cliquant sur un serveur présent dans la liste. Il est recommandé de classer les serveurs par nombre d'utilisateurs et de choisir ceux qui acceuillent le plus de personne.
Dans l'onglet « Préférence » choisissez la rubrique « Connexion » et cochez la case Se connecter automatiquement au démarrage pour ne pas avoir à vous connecter vous-même.
Pour vous connecter au réseau Kad, allez sur l'onglet Kad (de l'icône réseaux), cliquez sur le bouton représentant un triangle à coté de **Nœuds**.
Dans le menu « Préférences », vous pouvez régler tous les paramètres de connexion, d'utilisation mémoire et CPU ou d'affichage. Certaines parties nécessitent tout de même une attention particulière… C'est le cas par exemple de la partie nommée « Répertoires ». En effet, c'est ici que vous indiquez les répertoires que vous souhaitez partager, ainsi que le répertoire où sont stockés les fichiers que vous téléchargez. Par défaut, le répertoire des fichiers temporaires est **/home/user/.aMule/Temp** et celui contenant les fichiers terminés **/home/user/.aMule/Incoming**. Vous pouvez, bien évidemment, changer ces répertoires.
Attention ! Il est impératif, pour quel le partage de fichier fonctionne correctement, que les répertoires « Incoming » et « Temp » soient différents.
Dans la partie concernant le lecteur vidéo, je vous conseille d'utiliser VLC pour lire les vidéos incomplètes.
Ensuite, la partie « Sécurité » est très importante puisque c'est dans cette partie que vous pouvez donner (ou non) un accès aux autres utilisateurs à la liste des fichiers que vous partagez. Cette option est réglée par défaut sur Personne. Vous pouvez le changer mais en veillant à ne pas partager l'intégralité de votre système.
Enfin, le dernier point important concerne le filtrage des ip. Il existe dans aMule un système permettant de bloquer un certain nombre d'ip indésirables. Celles-ci sont stockées le « ipfilter.dat » qu'il est possible de mettre à jour automatiquement depuis internet. Par exemple depuis http://ovh.dl.sourceforge.net/sourceforge/emulepawcio/ipfilter.dat. Copier-coller cette adresse en face de "URL", puis cliquer sur Mettre à jour maintenant.

## Configuration (indispensable) d'iptables

Si vous utilisez iptables, vous aurez à ajouter quelques règles pour qu'amule puisse fonctionner correctement. Pour les connexions entrantes, ouvrez donc un shell et tapez :

```
Sh# sudo iptables -A INPUT -i eth0 -p TCP --
dport 4662 -j ACCEPT
Sh# sudo iptables -A INPUT -i eth0 -p UDP --
```

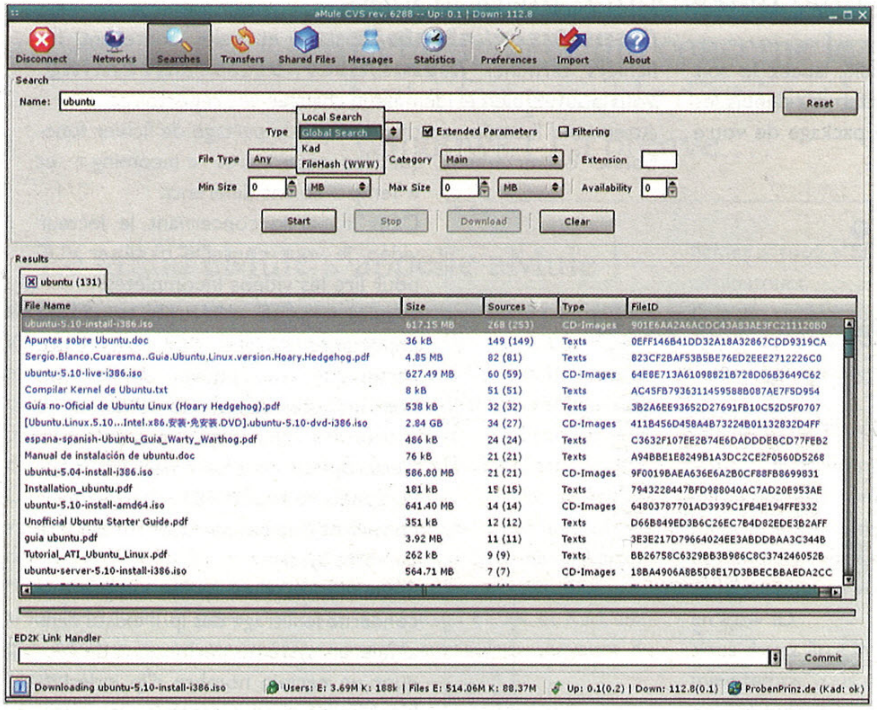
```
dport 4665 -j ACCEPT
Sh# sudo iptables -A INPUT -i eth0 -p UDP --dport
4672 -j ACCEPT
```

Si vous avez toutefois changé les ports de connexion d'aMule dans la configuration, modifiez aux besoins 4662, 4665 ou 4672. Enfin, pour configurer les connexions sortantes,

```
Sh# sudo iptables -P OUTPUT ACCEPT
```

Voilà, vous pouvez maintenant utiliser aMule au maximum de ces capacités. Vous n'avez plus qu'à vous rendre dans l'onglet « Recherches » pour trouver ce dont vous avez besoin.

Attention ! aMule ne doit vous servir qu'à télécharger des œuvres libres de droits !




## Les types de recherche sur aMule :

***Locale*** *: les recherches effectuées ne le seront que sur le serveur sur lequel vous êtes connecté.*

***Globale*** *: Votre recherche s'effectue sur l'ensemble des serveurs du réseau ED2K et Kad*

***Kad*** *: Votre recherche s'effectue uniquement sur le réseau Kad.*

*Remarque : Les résultats de recherche varient entre Kademlia et ED2K ! Pour effectuer une recherche avec Kademlia il faut spécifier le « Type » de Recherche sur « Kad » et non « Recherche globale ».*

*Attention ! Évitez d'effectuer trop souvent des recherches globales, cela a tendance à solliciter beaucoup les serveurs et vous risquez le bannissement de certains serveurs…*

## Récupérer un lien ed2k directement sur internet

La première méthode, la plus simple consiste à installer le package nommé « amule-utils » depuis votre gestionnaire de package et redémarrez Firefox s'il été en cours d'exécution. Ensuite, si lors du téléchargement d'un fichier, il vous est demandé quelle application lancer, sélectionnez aMule.

La seconde méthode, moins pratique mais tout aussi efficace (et même plus efficace si votre gestionnaire de packages ne connaît pas amule-utils…) consiste à passer par les registres de Firefox. Pour cela, lancez Firefox et tapez « about:config » dans la barre d'adresse. La page de configuration de Firefox s'ouvre alors ; faites un clic droit, sélectionnez « Nouvelle » puis « Valeur booléenne ». Ensuite, mettez en titre de valeur « network.protocol-handler.external.ed2k » et positionnez-la à « TRUE ». Créez ensuite une nouvelle chaine de caractères et nommez la « network.protocol-handler.app.ed2k » en lui donnant comme valeur le chemin du script ed2k, par défaut « /usr/bin/ed2k ». Si vous ne savez pas ou est passé ed2k, ouvrez un shell et tapez « whereis ed2k »

Si "/usr/bin/ed2k" ne convient pas mettre juste "ed2k"

Vous pouvez à présent cliquer sur un lien ed2k qui ira gentiment se mettre directement dans aMule.

## Transformez une mule en démon : aMuled

aMule peut être lancé en mode « daemon » (c'est-à-dire en tant que service), et ainsi tourner sur votre ordinateur sans interface graphique, ce qui économisera pas mal de ressources.

Pour utiliser aMule en mode daemon, installez le package « amule-daemon » s'il est disponible dans votre distribution, sinon compilez le avec les lignes :

```
Sh# ./configure --disable-mono-
lithic --with-toolkit=base --
enable-amule-daemon --enable-
amulecmd --enable-webserver
Sh# make && make install
```

## Configuration

Au démarrage, amuled récupère la configuration de aMule, et utilise donc les mêmes paramètres. Toute la configuration peut donc s'effectuer depuis l'interface graphique amule, que vous devrez toutefois quitter avant de lancer aMuled.

Pour une utilisation sur un serveur, sans interface graphique, la configuration se fera directement en éditant le fichier « ~/.aMule/amule.conf » ; fichier créé au un premier lancement de l'application.

Il sera juste nécessaire d'ajouter un utilisateur à la configuration du daemon. Pour des raisons de sécurité, évitez l'utilisation de root :

```
Sh# sudo gedit /etc/default/amule-daemon
```

Mettez la ligne nommée AMULED_USER à jour en lui spécifiant un nom d'utilisateur. Vous pouvez même créer un utilisateur spécifique à aMule en lui donnant comme répertoire home /dev /null et comme shell /bin/false ou /bin /nologin si votre distribution le permet.
Pour lancer le daemon, ouvrez un shell :

```
Sh# /etc/init.d/amule-daemon start
```

Vous pouvez changer « start » par « restart » ou « stop » !

## aMuleweb, la mule à distance !

aMuleweb est une autre application permettant de controler aMule, et aMuled, depuis votre navigateur, sur la machine locale mais également à distance ! Pratique si vous faites tourner aMuled sur une machine dédiée !
aMuleweb est installé lorsque vous installé les « amule-utils ».
Ouvrez aMule et rendez vous dans le menu « Préférences » puis « Contrôle à distance »et cliquez sur Démarrer amuleweb au lancement.
Spécifiez le port de contrôle, par défaut, il s'agit du port 4711, puis entrer un mot de passe administrateur, et enfin, cochez la case « Accepter les connexions externes » si vous désirez vous connecter depuis une autre machine. Si vous souhaitez accéder à aMule depuis n'importe quel point du globe, pensez à ouvrir le port 4711 en TCP sur votre routeur/pare feu.

```
Sh# sudo iptables -A INPUT -i eth0 -p TCP --
dport 4711 -j ACCEPT
```

Attention toutefois, si vous faites ce choix, il est préférable de mettre un mot de passe ultra sécurisé ! L'idéal étant de se limiter au réseau local !
Éditez ensuite le fichier /home/users/.aMule/amule.conf. (Si ce fichier n'existe pas, vous pouvez le créer avec la commande : « amuleweb –w »). Une fois ouvert dans votre éditeur de texte préféré, ajoutez a la fin de ce fichier ceci :

```
[ExternalConnect]
AcceptExternalConnections=1
ECUseTCPPort=1
ECPassword=votre_mot_de_passe_crypté
```

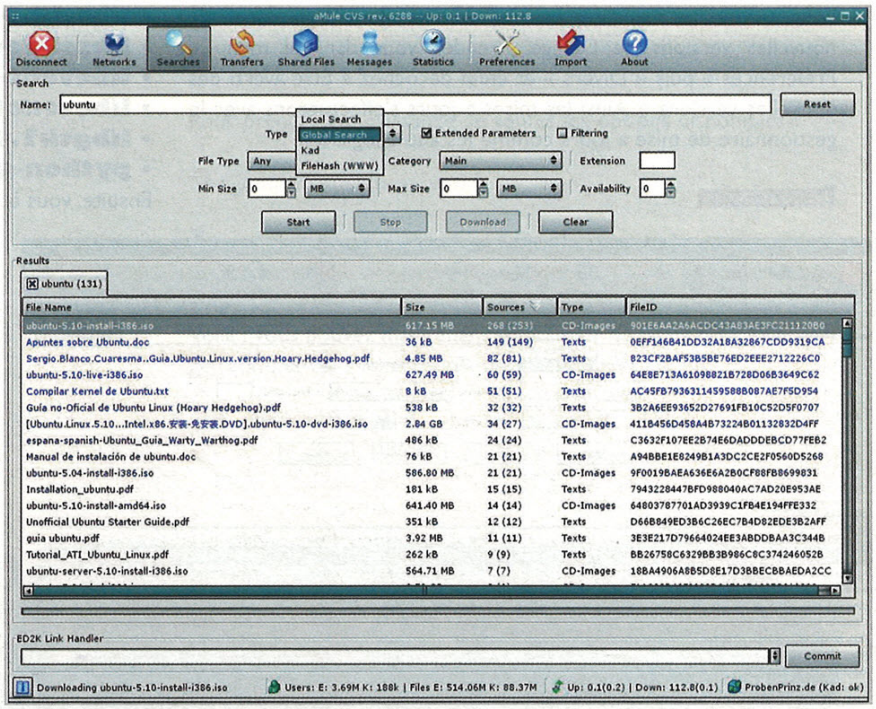
Il faut aussi également communiquer ce mot de passe à amuleweb : editez le fichier /home/users/.aMule/remote.conf et entrez votre mot de passe crypté après "=" de Password, Adminpassword et de Guestpassword.
Pour crypter votre mot de passe, rien de plus simple :

```
Sh# echo –n monM0td3p4sSeUltr4S3curE !!! |
md5sum | cut –d ' ' –f 1
```

# Inondez votre Linux avec un client torrent !

De nombreux clients torrent existe sous linux (peut être même plus que sur Windows…), mais tous n'offre pas les même fonctionnalités. Voici donc une aide pour faire le bon choix….

## Deluge



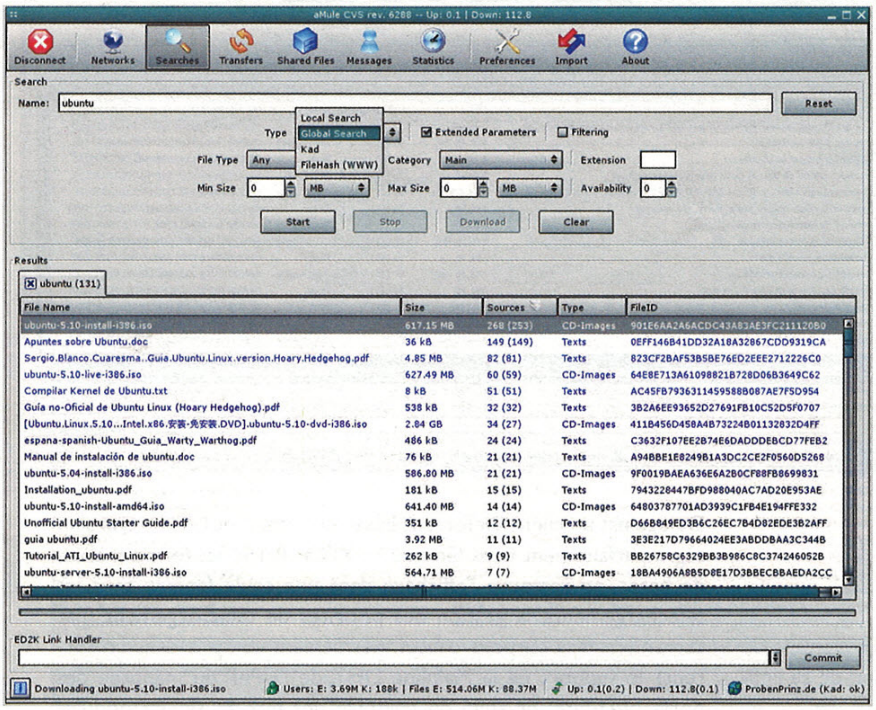

Deluge est un client BitTorrent basé sur Python et GTK+. Qui s'intègre parfaitement dans Gnome et XFCE. Parmi les foncionnalités de bases, on retrouve l'affichage dans une seule fenêtre tous les téléchargements, la gestion des priorités de téléchargement (par torrent, mais aussi pour chacun des fichiers compris dans un torrent), le support de la Mainline DHT, de l'UPnP, du mappage des ports NAT-PMP et enfin du chiffrage des flux. Plusieurs modules ajoute à ces fonctionnalités de bases plusieurs « features » bien sympathique comme par exemple la possibilité de créer des fichiers torrent, le déplacement des torrents pendant leurs téléchargement et, coté sécurité, un filtre d'IP par importation de listes noires. Enfin, vous pouvez afficher des graphiques statistiques sur les flux entrants et sortants et même planifier des tâches !
Pour installer Deluge, utilisez votre gestionnaire de package (apt-get install deluge-torrent ?)
Plusieurs problèmes peuvent survenir lors de l'utilisation de Deluge. Parmi les problèmes les plus courants, il est possible que Deluge ne démarre tout simplement pas. Il peut y avoir plusieurs raisons à cela. Lancer Deluge dans un terminal pour voir un peu plus en détail ce qui se passe.

```
Torrent Size 71904023.0
Available Space 5355061248
Raising error:
deluge_core; using libtorrent 0.11.0.0. Compiled
```

```
with NDEBUG value: 1
terminate called after throwing an instance of
'boost::filesystem::filesystem_error'
  what():  boost::filesystem::default_name_check:
default name check already set
Abandon (core dumped)
Sh# rm ~/.config/deluge/*.state && rm
~/.config/deluge/torrentfiles/*.fastresume
```

Enfin, si vous ne souhaitez pas être informé lors de la sortie de nouvelles versions de Deluge , rendez vous dans le menu « Préférences » puis « Divers » et enfin décochez « Être averti des nouvelles versions ». Ainsi les mises a jours s'effectuerons avec le gestionnaire de mise a jours comme les autre logiciels.

## Transmission




Transmission est un client BitTorrent léger en GTK2 avec une interface très épurée inclut dans ubuntu par défaut depuis la version Hardy 8.04. Ce client offre la possibilité de choisir le nombre de connexions totales et par torrent, une option permettant de désactiver le cryptage, un mode minimal pour montrer plus de torrents dans moins d'espace de bureau, le filtrage de torrent. Une fonctionnalité qui s'avère bien pratique pour tester la configuration de votre par feu est le test du port dans les préférences. Une fenêtre de statistiques proposant le tri par activité, par progrès, par état, et par tracker est également disponible.
Comme d'habitude, pour installer transmission, utilisez votre système de package. Par exemple, sur Ubuntu, vous pouvez lancer l'installation avec un « aptitude install transmission » qui installera à la fois l'interface graphique et le mode ligne de commande, « aptitude install transmission-gtk » n'installera que l'interface graphique et enfin « aptitude install transmission-cli » installera transmission uniquement en mode ligne de commande.

Enfin, si vous souhaitez toujours utiliser la dernière version de transmission, vous pouvez le compiler. Pour cela, vous devez d'abord disposer des packages suivant :

- **build-essential**
- **fakeroot**
- **checkinstall**
- **svn**
- **libgettext-ruby1.8**
- **libgettext-ruby-util**
- **libssl0.9.8**
- **libssl-dev**
- **libevent-dev**
- **libevent1**
- **libgtk2.0-dev**
- **python-gtk2-dev**

Ensuite, vous êtes prêt à installer transmission, ouvrez un shell :

```
Sh# svn co
svn://svn.m0k.org/Transmission/trunk
Transmission
```

Après avoir entré cette commande, la liste des fichiers téléchargés défile et se termine par « Révision xxxx extraite ». De plus, parmi les fichiers téléchargés se trouve un fichier « NEWS » dans lequel se trouve le numéro de version. Il ne nous reste plus qu'à lancer la compilation :

```
Sh# cd Transmission
Sh#./autogen.sh
Sh#./configure -q
Sh# make
Sh# sudo checkinstall
```

Après avoir lancé la commande checkinstall, on vous demande d'entrer un résumé ; tapez ici ce que vous souhaitez... client bittorrent semble convenable...
Vous aurez peut-être un message d'erreur : **Warning: The package name "Transmission" contains upper case...** Ce n'est en fait pas un problème, appuyez donc sur Entrée pour passer. Vous pouvez ensuite modifier différents paramètres :

```
This package will be built according to these
values:
0 -  Maintainer: [ root@ubuntu ]
1 -  Summary: [ Client Bittorrent simple et
leger ]
2 -  Name:     [ transmission ]
3 -  Version: [  ]
4 -  Release: [ 1 ]
5 -  License: [ GPL ]
6 -  Group:    [ checkinstall ]
7 -  Architecture: [ i386 ]
8 -  Source location: [ Transmission ]
9 -  Alternate source location: [  ]
10 - Requires: [  ]
```

```
Enter a number to change any of them or press
ENTER to continue:
```

Tapez 3 pour modifier le numéro de version, 4 pour modifier le numéro de révision. Attention, vous devez renseigner les numéros de versions indiqués dans le fichier NEWS ! Vous obtenez quelque chose qui ressemble à ceci :

```
This package will be built according to these
values:
0 -  Maintainer: [ root@ubuntu ]
1 -  Summary: [ Client Bittorrent simple et
leger ]
2 -  Name:     [ transmission ]
3 -  Version:  [ 1.10 ]
4 -  Release:  [ svn-r4800 ]
5 -  License:  [ GPL ]
6 -  Group:    [ checkinstall ]
7 -  Architecture: [ i386 ]
8 -  Source location: [ Transmission ]
9 -  Alternate source location: [  ]
10 - Requires: [  ]
Enter a number to change any of them or press
ENTER to continue:
```

Vous pouvez maintenant sereinement appuyez sur « Entrée » pour créer et installer automatiquement un paquet deb.
Vous êtes maintenant prêt a utiliser transmission. Vous pouvez le lancer depuis le menu si vous avez utilisé le gestionnaire de packages de votre distribution ou depuis un shell :

```
Sh# transmission-gtk
```

Ou encore :

```
Sh# transmissioncli /chemin/de/votre/fichier.tor-
rent
```

Pour prendre connaissance de toutes les options disponibles :

```
Sh# man transmissioncli
```

À noter que transmissioncli sauvegarde ses téléchargements dans le dossier où il a été lancé. Pensez donc à ne pas le lancer depuis « / »...
Voilà ! vous pouvez maintenant vous servir de Linux pour télécharger comme vous le faisiez avec votre vieux Windows....

# Installez votre propre serveur FTP

Créer son serveur FTP permet de partager des fichiers avec le monde entier. Il est possible d'envoyer et de recevoir des fichiers sans aucune limitation de taille, contrairement aux emails par exemple. Nous verrons ici comment installer et configurer au mieux un serveur FTP sur votre distribution linux afin de partager des fichiers avec un niveau de sécurité maximal !

## Un serveur FTP qui fait tout : glFTPd

glFTPd est un serveur FTP gratuit entièrement paramétrable (un peu trop ? A vous de juger...). Il possède sa propre base de données utilisateurs, qui peut être complètement gérée en ligne avec des commandes SITE. Son environnement chroot le rend assez sécurisé et les scripts et add-ons qu'il est possible de lui ajouter facilement permettent d'y ajouter grand nombre d'options que vous ne soupçonniez même pas.

Parmi ses fonctions les plus communes, il dispose d'une gestion de groupes et utilisateurs virtuels, de support de réseaux multiples, de configuration par IP, de réglage de la bande bassante, du calcul des CRC à la volée quand le fichier est en cours d'upload, du support des scripts sur quasi toutes les commandes, d'une fonction de « dup check » vérifiant l'existence d'un fichier, de gestion des utilisateurs en ligne, d'un moteur de statistiques. Coté sécurité, glFTPd inclut également le support du cryptage SSL/TLS ainsi que des ACL (Access Control List).

## Installation

Pour télécharger glFTPd, il suffit de vous rendre sur le site http://www.glftpd.com et de prendre la dernière version ou alors, si votre machine visant à accueillir le serveur ftp ne dispose pas d'interface graphique :

```
Sh# wget http://www.glftpd.com/files/glftpd-LNX_2.01.tgz
```

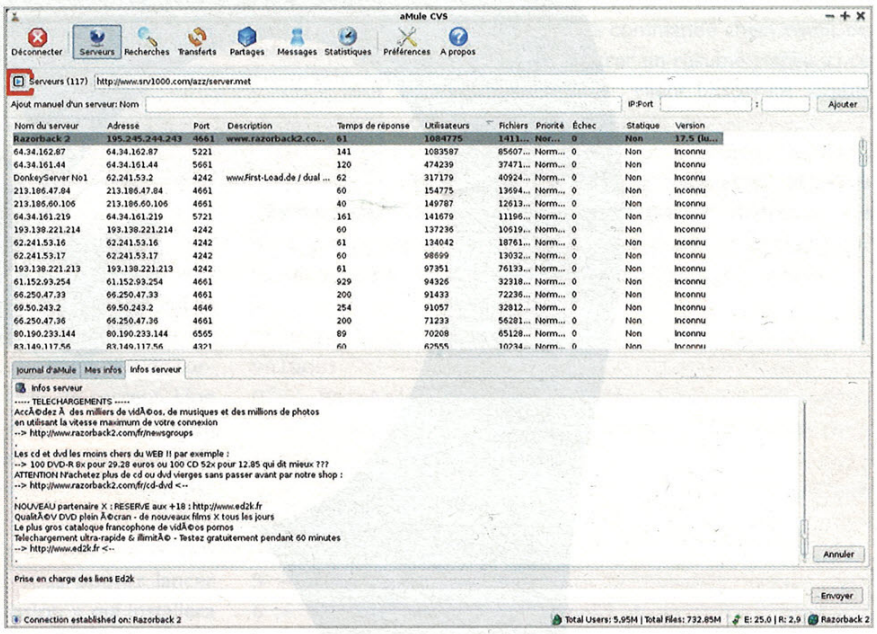
Ensuite lancez la décompression de l'archive téléchargée avec la commande tar -xvzf :

```
Sh# tar xzvf glftpd-LNX_2.01.tgz
```

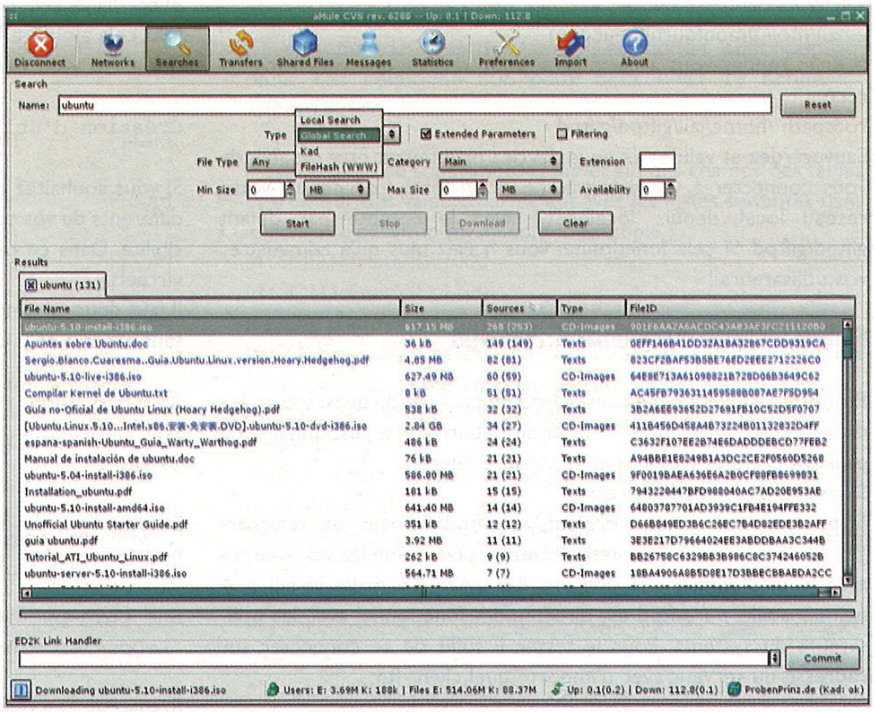
Une fois ceci réalisé, rendez-vous dans le répertoire de glFTPd et lancez la commande ./installgl.sh

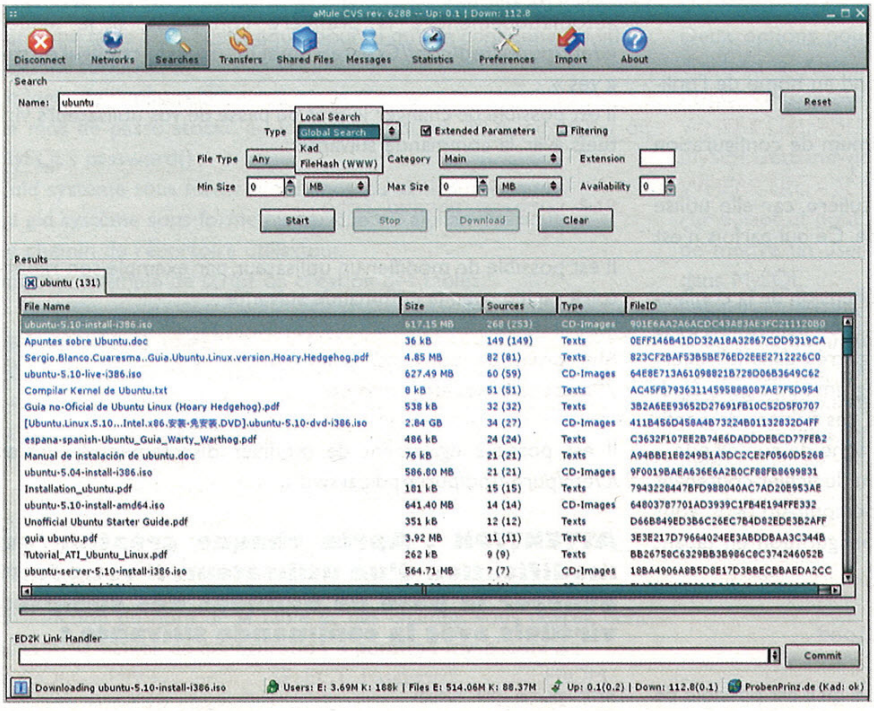
Arrivé ici, vous allez lancer la commande ./installgl.sh qui est le script d'installation. Il va vous permettre d'effectuer le plus gros de l'installation et configuration de glftpd simplement.

Lors de l'installation, ce script vous offre la possibilité de configurer glFTPd en vous posant une suite de questions auxquelles vous devrez répondre. La première d'entre elles concerne l'utilisation de TCPD. Mettez oui uniquement si vous savez ce que c'est et que vous en avez l'utilité.



La deuxième étape vous demandera si vous souhaitez utiliser glftpd dans un environnement "jailed". Il s'agit d'un répertoire privé dans lequel les utilisateurs classiques du shell ne pourront accéder. Si votre objectif est avant tout la sécurité, sélectionnez ici « oui ». Dans ce cas, il vous sera alors demandé de sélectionner le répertoire de 'jail' (« prison » en anglais). Sélectionnez ici un répertoire qui ne risque pas d'affecter le système (/home/jail/glftpd par exemple)
A la question suivante, tapez « Y » (pour oui) si vous souhaitez créer un groupe privé qui pourra accéder à glftpd, dans le cas contraire, il n'y a que root qui pourra y accéder ce qui posera un problème de sécurité. Il est fortement conseillé de créer un groupe « ftpgroup » qui assurera vos arrières.
Choisissez ensuite les utilisateurs qui auront accès à l'administration du glftpd. Énumérez ici la liste des utilisateurs auxquels vous souhaitez attribuer ces droits. Ne mettez que un ou deux utilisateurs en qui vous avez entière confiance (donc vous seul devrait suffire) en plus du root et séparez-les par des espaces...
glFTPd vous demande ensuite le répertoire ou vous souhaitez placer le binaire. Laissez par défaut semble tout à fait convenable à cette étape.
Vous devez ensuite donner un nombre au service (le nom qui s'affichera dans la liste des processus lors d'un 'ps' ou d'un 'top'.Ici encore, la valeur par défaut fera parfaitement l'affaire. L'installation suit ensuite son cours jusqu'à vous demander le port d'écoute. 21 est la valeur « conventionnelle », mais si toutefois vous désirez le changer, libre à vous ! Vous devrez penser à toujours spécifier ce port aux personnes à qui vous donnerez votre adresse ;)
Ensuite, si vous désirez que les semaines commencent le lundi, sélectionnez « Y ». Si vous préférez qu'elles commencent le dimanche, tapez « N »




L'étape suivante est une des plus intéressantes car elle permet la création d'un certificat qui sera utilisé pour le cryptage SSL/TLS. Même si il n'est pas pour vous nécessaire d'utiliser le SSL (serveur FTP local avec connexion par RJ45 par exemple), vous devez créer ce certificat. Laissez ici les valeurs par défaut si vous n'avez pas pour habitude de centraliser vos certificats. Puis donnez un nom à ce certificat (glftpd semble acceptable).
Vous avez maintenant bien mérité que glFTPd vous félicite de cette installation sans accroc !

## Configuration

glFTPd est connu dans le monde linux/*BSD pour sa grande souplesse mais aussi pour sa configuration parfois... rébarbative. Prenez donc un remontant avant de partir à la conquête du fichier glftpd.conf. Une fois votre remontant prêt à vous soutenir dans cette épreuve, ouvrez le fichier glftpd.conf se trouvant dans le répertoire choisi pour l'installation (/home/jail/glftpd dans notre exemple) avec votre éditeur unix habituel (donc vi ;))
Nous allons nous concentrer sur les fonctionnalités de base de

glFTPd. Utilisez la documentation officielle si vous souhaitez paramétrer glFTPd avec une grande minutie.
# mettez ici le nom de votre serveur ftp (=sitename), version longue et courte.
sitename_long Mon Super Nouveau FTP
sitename_short MSNFTP
# ici votre adresse email
email moi@monfournisseur.com
# Enfin rendez-vous à cette ligne pour vérifier que tout est correcte.
rootpath /home/jail/glftpd/glftpd
Sauvegardez et validez, et vous devriez maintenant être capable de vous connecter à votre serveur FTP en local (pas depuis votre réseau local, depuis 'localhost') avec le compte par défaut glftpd/glftpd. Si cela fonctionne, vous n'avez plus qu'à administrer vos utilisateurs !

## Simple, sécurisé et performant : Pure-ftpd

Pure-ftpd est disponible dans la liste de package de quasi toutes les distributions linux. Installez-le donc depuis votre gestionnaire de package préféré, urmpi, rpm, aptitude, emerge…
Sinon, rendez vous à l'adresse http://download.pureftpd.org/pub/pure-ftpd/releases et téléchargez la dernière version, désarchivez-la, puis compilez les sources avec le traditionnel « ./configure && make && make install ». A peine installé, pure-ftpd est directement fonctionnel avec les utilisateurs du système. Pour le tester, il suffit de se connecter sur l'adresse du serveur avec n'importe quel client ftp:

```
Sh# ftp localhost
```

Le répertoire ftp utilisé par défaut correspond au home de l'utilisateur.
Cependant, il est conseillé d'opérer un minimum de configuration sur pure-ftpd pour avoir un serveur sécurisé !
La configuration de « pure-ftpd » est particulière, car elle utilise des paramètres passés en ligne de commande. Ce qui parfois n'est pas vraiment pratique.
Heureusement, pour nous faciliter la compréhension de la configuration, pure-ftpd utilise un fichier de configuration par option de la ligne de commande. Ces fichiers sont enregistrés dans le répertoire « /etc/pure-ftpd/conf ». C'est le programme « pure-ftpd-wrapper » qui se chargera pour vous de lire ces fichiers de configuration pour modifier les paramètres de la ligne de commande.
Il est par exemple possible de modifier le nom du fichier contenant les logs en modifiant le fichier « /etc/pure-ftpd/conf/AltLog ». Vous pouvez forcer pure-ftpd à relire toute la configuration en tapant dans un shell :

```
Sh# /etc/init.d/pure-ftpd force-reload
```

Pour avoir la liste des fichiers de configuration qu'il est possible de créer ou de modifier, ainsi que la syntaxe pour les renseigner, ouvrez la page de manuel de pure-ftpd-wrapper.

```
Sh# man pure-ftpd-wrapper
```

Il faut faire attention à la case des caractères lors de la création des fichiers. Chaque fichier correspondant à une option de configuration que vous retrouverez au chapitre « Annexe » de ce document.
Par exemple, en créant le fichier « /etc/pure-ftpd/conf/ChrootEveryone » et en mettant « yes » sur sa première ligne, cela permet de « Chrooter » les utilisateurs. Pour info, cela ajoutera le paramètre « -A » sur la ligne de commandes après avoir rechargé la configuration.

### Création d'utilisateurs virtuels

Si vous souhaitez que les comptes de votre serveur FTP soient différents de vos comptes UNIX, pure-ftpd vous offre cette possibilité. Dans ce cas, il vous faudra donc créer des utilisateurs virtuels.
Il faut donc commencer par créer un vrai compte système mais sans accès à celui-ci :

```
Sh# groupadd ftpgroup
Sh# useradd -g ftpgroup -d /dev/null -s /etc
ftpuser
```

Ensuite, vous devez créer les utilisateurs virtuels avec la commande suivante :

```
Sh# pure-pw useradd toto -u ftpuser -d
/LeDossierDeToto
```

Remarque : Le dossier « LeDossierDeToto » indiqué sera créé automatiquement à la première connexion si le fichier « /etc/pure-ftpd/conf/CreateHomeDir » est configuré sur « yes ».
Il est possible de changer le mot de passe de vos utilisateurs virtuels avec la commande suivante :

```
Sh# pure-pw passwd toto
```

Il est possible de modifier un utilisateur, par exemple son répertoire racine, avec la commande suivante :

```
Sh# pure-pw usermod toto -d
/UnautreDossierPourToto
```

Il est possible également de modifier directement le fichier « /etc/pure-ftpd/pureftpd.passwd »

**ATTENTION : Après chaque création ou modification d'un utilisateur, il vous faut générer la base de données des comptes virtuels avec la commande suivante :**

```
Sh# pure-pw mkdb
```

## Gérer les utilisateurs de Pure-FTPd dans MySQL

Il faut commencer par créer un fichier de configuration lisible uniquement par root (chmod 600) pour des raisons de sécurité.

Ce fichier de configuration doit être enregistré dans le répertoire spécifié dans le fichier /etc/pure-ftpd .conf. Par défaut : /etc/pureftpd-mysql.conf
Exemple de fichier de configuration :

```
#MYSQLServer    localhost
#MYSQLPort      3306
MYSQLSocket     /tmp/mysql.sock
MYSQLUser       root
MYSQLPassword   rootpw
MYSQLDatabase   pureftpd
MYSQLCrypt      cleartext
MYSQLGetPW              SELECT Password         FROM
users   WHERE User="\L"
MYSQLGetUID             SELECT Uid              FROM
users   WHERE User="\L"
MYSQLGetGID             SELECT Gid              FROM
users   WHERE User="\L"
MYSQLGetDir             SELECT Dir              FROM
users   WHERE User="\L"
```

Ensuite, vous devez lancer pure-ftpd avec l'option suivante :

```
Sh# /usr/local/bin/pure-ftpd -login
mysql:/etc/pureftpd-mysql.conf
```

Il est possible d'utiliser à la fois une authentification avec MySQL et une authentification sur les comptes UNIX. Pour cela :

```
Sh# /usr/local/bin/pure-ftpd -login
mysql:/etc/pureftpd-mysql.conf use -login unix
```

Pure-FTPd est très flexible et les utilisateurs peuvent être stockés dans une table SQL quelconque, pourvu qu'elle contienne les informations suivantes :
Le login de l'utilisateur
Le mot de passe stocké au format : plaintext, MD5, crypt()ed ou MySQL's password()
L'uid système sous forme numérique ou alphanumérique
Le gid système sous forme numérique ou alphanumérique
Le chemin du répertoire utilisateur.
Voici un exemple de script de création des tables :

```
CREATE TABLE users (
 User varchar(16) NOT NULL default '',
 Password varchar(64) NOT NULL default '',
 Uid varchar(11) NOT NULL default '',
 Gid varchar(11) NOT NULL default '',
 Dir varchar(128) NOT NULL default '',
 PRIMARY KEY (User)
);
```

En fonction du nom de la table et des champs, vous devrez donc adapter le fichier de configuration « pureftpd-mysql.conf » :

```
MYSQLGetPW              SELECT Password         FROM
users   WHERE User="\L"
MYSQLGetUID             SELECT Uid              FROM
users   WHERE User="\L"
MYSQLGetGID             SELECT Gid              FROM
users   WHERE User="\L"
MYSQLGetDir             SELECT Dir              FROM
users   WHERE User="\L"
```

```
Codes des requêtes SQL :
  \L représente le login
  \I représente l'adresse IP du client
  \P représente le port de connexion
  \R représente l'adresse IP du client
```

Il est possible d'utiliser les codes précédents dans les requêtes SQL pour retrouver la bonne information.

Il est possible de supprimer les uid et gid des champs des tables et d'utiliser des valeurs par défaut avec les lignes suivantes dans le fichier « pureftpd-mysql.conf ». Exemple :

```
MYSQLDefaultUID 1000
MYSQLDefaultGID 1000
```

Il existe d'autres options facultatives :
MySQLGetQTAFS permet de déterminer le nombre de fichiers qu'un utilisateur peut enregistrer dans son dossier. Exemple :

```
MySQLGetQTAFS  SELECT QuotaFiles FROM users WHERE
User="\L"
```

MySQLGetQTASZ permet de limiter la taille maximum en Mo, qu'un utilisateur peut mettre dans son dossier. Exemple :

```
MySQLGetQTASZ  SELECT QuotaSize FROM users WHERE
User="\L"
```

Il y a également deux options pour les ratios :

```
MySQLGetRatioUL  SELECT ULRatio FROM users WHERE
User="\L"
MySQLGetRatioDL  SELECT DLRatio FROM users WHERE
User="\L"
```

Deux options pour limiter la bande passante à l'upload et au dowload en KB/s :

```
MySQLGetBandwidthUL  SELECT ULBandwidth FROM users
WHERE User="\L"
MySQLGetBandwidthDL  SELECT DLBandwidth FROM users
WHERE User="\L"
```

Par défaut et pour des raisons de sécurité, il n'est pas possible de mettre un user "root" ou un iud/id à 0 comme utilisateur dans MySQL.
Enfin, il est possible de lancer Pure-FTPd uniquement à partir de la ligne de commande. Il est tout de même vivement conseillé de sauvegarder votre « ligne de commande parfaite » dans un script Shell (pourquoi pas qui s'exécute au démarrage sur le système ?) si vous ne souhaitez pas la réécrire à chaque lancement...

| Option courte | Option Gnu | Commentaire |
|---|---|---|
| -0 | —notruncate | Cette option permet de renommer automatiquement un fichier si un autre du même nom existe déjà sur le serveur.Cette option est incompatible avec les "Quotas virtuels" |
| -1 | —logpid | Log le PID de chaque session dans Syslog |
| -4 | —ipv4only | Écoute uniquement les connections IP V4 |
| -6 | —ipv6only | Écoute uniquement les connections IP V6 |
| -a | —trustedgid | Seulement les membre du groupe peuvent se connecter. |
| -A | —chrooteveryone | Chroot tous les utilisateurs |
| -B | —daemonize | Démarrer le serveur en mode Standalone (en daémon) |
| -c | —maxclientsnumber | Autorise au maximum clients à se connecter simultanément au serveur (Défaut = 50) |
| -C | —maxclientsperip | Nombre maximum de connexions simultanées par client (adresse IP) |
| -d | —verboselog | Envoie des messages de débugage dans Syslog (A utiliser seulement en cas de problème)Pour avoir également les log de réponses, il faut doubler ce paramètre. |
| -D | —displaydotfiles | Affiche les fichiers commençant par un point (Paramètre à activer pour les puristes et à désactiver pour les simples utilisateurs) |
| -e | —anonymousonly | Autorise uniquement les utilisateurs anonymes. A utiliser uniquement pour les sites FTP publiques. |
| -E | —noanonymous | Autorise uniquement les utilisateurs authentifiés. Les utilisateurs anonymes sont interdits. |
| -f | —syslogfacility | Indique la « facility » à utiliser pour syslog (Défaut = local2). « -f none » permet de désactiver les logs. |
| -F | —fortunesfile | Affiche une ligne de texte aléatoire à la connexion en provenance du fichier « fortune file » (ex : Astuce du jour). Pour afficher un texte à la connexion, il faut également utiliser cette option. |
| -g | —pidfile | Change l'adresse du fichier pid quand l serveur est exécuté en mode standalone. (Défaut = /var/run/pure-ftpd.pid) |
| -H | —dontresolve | Par défaut, les noms pleinement qualifiés sont logués en faisant une requête DNS inverse sur l'adresse IP. Cette option permet d'éviter de rechercher le nom pleinement qualifié dans le but d'accélérer le temps de connexion. Cette option est donc conseillée pour les sites publiques très chargés. |
| -i | --anonymouscantupload | Interdit l'accès en écriture sur le serveur (upload) pour les utilisateurs anonymes. |
| -I | —maxidletime | Change la durée de connexion maximum en minutes (Défaut = 15) |
| -j | —createhomedir | Crée automatiquement le home de l'utilisateur s'il n'existe pas. |
| -k | —MaxDiskUsage | Interdit les chargements sur le serveur (upload) si la partition utilisée est chargée à plus de |
| -K | —keepallfiles | Autorise les utilisateurs à charger des fichiers sur le serveur, mais les interdit de les renommer ou les supprimer. |
| -l | —login or : | Ajoute une nouvelle règle d'authentification. |
| -L | —limitrecursion | Cette option permet d'éviter le nombre de fichiers listes et le nombre de niveau de recherches pour éviter les déni de services.(Défaut = 2000 fichiers et 5 niveaux de répertoire) |
| -m | —maxload | Interdit les téléchargements anonymes si la charge du CPU atteint |
| -M | —anonymouscancreatedirs | Autorise les utilisateurs anonymes à créer des répertoires. |
| -N | —natmode | Mode NAT |
| -O | --altlog : | Enregistre les logs des fichiers transférés dans un fichier spécifique. Plusieurs formats sont supportés : CLF (Apache-like), Stats, W3C and xferlog.Par exemple le format "Stats" ressemble à : |
| -p | —passiveportrange | Permet de modifier les ports sur lesquels le serveur Écoute |
| -P | —forcepassiveip | Force l'adresse IP ou le nom de la machine pour le retour |
| -q | —anonymousratio : | Active les ratios pour les utilisateurs anonymes |
| -Q | —userratio : | Active les ratios pour tous les utilisateurs |
| -r | --autorename | Ne jamais écraser un fichier existant. Le fichier est automatiquement renommé si c'est nécessaire. |
| -R | --nochmod | Interdit aux utilisateurs d'utiliser les commandes CHMOD |
| -s | --antiwarez | Interdit aux utilisateur anonymes de télécharger des fichiers |
| -T | —userbandwidth or  :  | Active la limitation de la bande passante en kilobytes/seconds. Il est possible d'avoir une bande passante différente avec la syntaxe "-t  :  " |
| -t | --anonymousbandwidth | Active la limitation de la bande passante en kbps pour les utilisateurs anonymes. Il est possible d'avoir une bande passante différente avec la syntaxe "-t  :  " |
| -u | --minuid | Interdit les accès pour les utilisateurs ayant un uid inférieur à celui indiqué. |
| -U | --umask : | Change le mask de création des dossiers et des fichiers. (Défaut = 133.022) |
| -V | --trustedip | Autorise les accès non anonymes seulement pour l'adresse IP spécifiée. Les autres adres ses IP auront un accès anonyme. |
| -x | --prohibitdotfileswrite | Interdit les utilisateurs anonymes à écrire des fichiers commençant par un point. |
| -X | --prohibitdotfilesread | Interdit les utilisateurs anonymes à lire et écrire des fichiers commençant par un point. |
| -y | --peruserlimits : | Permet de limiter le nombre de sessiond simultanées qu'un utilisateur peut ouvrir (Défaut = 0 = illimités) |
| -z | --allowdotfiles | Autorise les utilisateurs anonymes à lire les fichiers et les dossiers commençant par un point. |


# Réservé aux grands-parents !

**Information et abonnement**

**www.ordisenior.com**

# Et Linux fut...

Vous connaissez Linux, du moins, vous pensez connaître Linux, mais savez-vous vraiment comment et pourquoi est né Linux ? Connaître l'histoire du système est en effet indispensable à la bonne connaissance du système !

## Contexte de la naissance de Linux

1991. Les PC dominent le marché des ordinateurs personnels et fonctionnent pour la plupart sur les systèmes d'exploitation MS-DOS, Windows et OS/2. Le prix du microprocesseur Intel 80386 ne cesse de baisser, et de plus en plus de personnes l'utilisent. Le problème est qu'aucun de ces trois systèmes n'est en mesure de pleinement exploiter les capacités 32 bits de ce processeur. GNU est alors déjà bien connu des hackers pour avoir produit de nombreux logiciels libres, dont la plupart des commandes Unix, l'éditeur de texte Emacs et le compilateur C GCC. Ces logiciels sont généralement utilisés sur des stations de travail fonctionnant sous UNIX propriétaire, car le noyau de système d'exploitation Hurd n'est qu'à l'état de projet.

En juin 1991, la Berkeley Software Distribution (BSD) sort la Networking Release 2 (Net/2), qui constitue un système UNIX BSD presque complet. Mais Unix System Laboratories ne l'entend pas de cette oreille. Il lance un procès contre Berkeley Software Distribution et l'avenir de cette distribution est alors incertain pendant presque deux ans !

D'autre part, le professeur Andrew Tanenbaum (dont le livre « Architecture de l'ordinateur » est une véritable mine d'or pour tout informaticien) développe le système d'exploitation « Minix » à des fins pédagogiques. Ce système s'inspire des grands fondements d'UNIX, il est totalement gratuit, ses sources sont disponibles mais non libres, et il privilégie la simplicité aux performances.

Un étudiant finlandais, du nom de Linus Torvalds, entreprend d'écrire un noyau de système d'exploitation qui plus tard s'appellera « noyau Linux ». Ce qui a poussé Linus Torvalds à une telle entreprise est la mauvaise disponibilité du serveur UNIX de son université à Helsinki.

Linus Torvalds fait alors son apprentissage sur le système d'exploitation Minix. Puisqu'Andrew Tanenbaum refuse toutes les contributions visant à améliorer Minix, Linus décide tout simplement de le remplacer. Il commence le développement de son système par un

émulateur de terminal, qu'il utilise pour se connecter via modem au serveur informatique de son université. La principale motivation de Linus est alors la compréhension du fonctionnement de son ordinateur, un compatible PC basé sur un microprocesseur Intel 80386. Linus a pour son système de grandes ambitions dès le départ puisqu'il souhaite développer un noyau entièrement compatible avec les normes POSIX.
Le 5 octobre 1991, il annonce sur le forum Usenet news:comp.os.minix la disponibilité d'une ébauche version 0.02 de son système d'exploitation (NDLR : le message original de cette annonce est sur la page suivante), capable d'exécuter Bash, gcc et quelques autres commandes unix mais nécessitant Minix pour sa propre compilation. La version 0.01 n'a, quant à elle, pas été annoncée officiellement, sa diffusion s'est limitée à quelques personnes de l'entourage de Linus

## Linux de 1991 à nos jours...

Depuis 1991, Linux a grandement évolué (ça, vous vous en doutiez…). Des centaines de passionnés et des entreprises, petites ou géantes, ont apporté leur contribution au projet, dont Linus Torvalds est encore aujourd'hui le coordinateur. Eric S. Raymond décrit le modèle de développement du noyau Linux et d'une partie des logiciels libres consultable sur http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar.html.
Le noyau avait initialement pour nom « Freax » mais Ari Lemmke, administrateur du serveur FTP ftp.funet.fi, qui héberge le travail de Linus Torvalds, dépose le travail de Linus dans un répertoire ayant pour nom la contraction de Linus et UNIX, « Linux ». Linux est né et devient alors une marque déposée au nom de Linus Torvalds. Le petit manchot que vous connaissez tous, « Tux », est dessiné par Larry Ewing en 1996. Il fait l'unanimité et devient le symbole du projet.
Parmi les étapes marquantes de la vie de Linux, on peut d'abord citer le lancement en octobre 1996 par « Matthias Ettrich » de l'environnement graphique KDE, puis en août 1997 par « Miguel de Icaza » de son concurrent GNOME, tous deux basés sur le système de fenêtrage X11 issu des travaux du Massachusetts Institute of Technology (MIT, université ou a étudier entre autre Richard Stallman). L'intérêt commercial de Linux prend sa source lors du lancement, en février 1998, de l' « Open Source Initiative ». Puis, en en juillet 1998, le géant Oracle Corporation porte et supporte sa célèbre base de données sous Linux.
Des grands noms du monde Linux ont pu bâtir de vraie entreprise sur un système pourtant gratuit ! Par exemple, Red Hat (entrée en bourse le 11 novembre 1999 » ou encore, depuis novembre 2003 et l'acquisition de SuSE par l'entreprise américaine Novell.
Linux s'est forgé sa réputation surtout grâce aux serveurs, notamment avec le très populaire LAMP. Linux a peu à peu supplanté tous les autres UNIX sur le marché toujours croissant des serveurs d'entreprise. Plus simple à administrer qu'OpenBSD et tout aussi performant, il a fait ses preuves et est désormais quasiment le seul sur le marché des serveurs avec Windows.
Linux a également fait ses preuves dans le domaine de l'embarqué. Léger, capable de support d'un grand nombre d'architecture matérielle, il est fréquemment utilisé avec les outils uClibc et BusyBox qui ont été développés pour le matériel particulièrement limité en capacité mémoire. En outre, le fait de pouvoir compiler le noyau Linux avec des options spécialement adaptées au matériel cible donne aux développeurs de nombreuses opportunités d'optimisation (qui parle de gentoo ?).
Cependant, la part de marché de Linux sur les postes clients reste faible. Les diverses estimations étant comprises en général entre 0,3 % et plus de 3 % en fonction des méthodes de relevé et de calcul. Ce chiffre est en réalité difficilement estimable, car d'une part, de nombreux navigateurs web modifient leur identité par défaut, et d'autre part, de très nombreux utilisateurs de Linux disposent également d'un Windows sur leur ordinateur. De plus, il n'est pas rare qu'un utilisateur Linux expérimenté configure son système de manière à ce qu'il ne communique pas d'informations concernant le système, car elles pourraient permettre de faciliter la recherche de failles éventuelles par une personne mal attentionnés.

## L'utopie Linux

Ce qui fait l'originalité de Linux par rapport aux systèmes d'exploitation concurrents comme Microsoft Windows, Mac OS, ou les autres UNIX propriétaires est qu'il est constitué d'un noyau libre et de logiciels libres.
Le noyau est diffusé sous licence GNU GPL (écrite elle aussi par Richard Stallman). Cette licence est dite basée sur le principe de copyleft (en opposition au copyright). Une œuvre dérivée d'un logiciel sous copyleft (GPL) doit elle-même être libre.
L'ouverture du code source, l'un des quatre critères correspondant à la notion de logiciel libre, a prouvé ses nombreux avantages notamment en matière de correction rapide des bogues qui sont la plaie de l'informatique, et la correction des failles de sécurité. C'est l'exact opposé de la politique menée par Microsoft qui préfère le principe de sécurité par l'obscurité.
En plus des développeurs, Linux se veut être un système communautaire. De nombreuses associations, connues sous le nom de Linux Users Group, Groupe d'Utilisateurs Linux (LUG ou GUL), cherchent à promouvoir Linux et les logiciels libres, en organisant par exemple des rencontres où des démonstrations de Linux sont faites.
Internet est également un moteur pour la communauté Linux. Nombreux sont les forums, les blogs, les channels IRC ou les utilisateurs expérimentés aident les débutants, et même parfois les professionnels. Par exemple, les sites tels que lea-linux, Linuxfr.org et Linux-Québec, aide les utilisateurs québécois comme français dans leur apprentissage des bases de Linux grâce à un réseau IRC très actif. Il existe aussi deux projets nommés respectivement « Proselux » et « Parrains.Linux » qui permettent aux linuxien(ne)s de se rencontrer pour s'entraider.

## Linux et les logiciels libres à l'assaut des ordinateurs !

Grand nombre de projets ayant vu le jour initialement sous Linux ont été portés sur Windows tant leurs réputations été bonne ! C'est par exemple le cas du serveur web Apache ou du logiciel de traitement graphique Gimp. Ce qui contribue à l'utilisation massive

des logiciels libre est en grande partie l'utilisation de formats ouverts, des formats de données dont les spécifications techniques sont publiques et sans restriction d'accès ni de mise en œuvre, afin de ne pas dépendre d'un seul logiciel et de pouvoir adapter les logiciels à des besoins spécifiques.

Mais si Linux est installé sur un grand nombre de postes aussi bien serveurs que station de travail, il le doit avant tout aux « distributions », solutions prêtes à être installées par l'utilisateur final comprenant un noyau Linux, des programmes d'installation et d'administration de l'ordinateur ainsi qu'un mécanisme facilitant l'installation et la mise à jour des logiciels (les gestionnaires de packages parmi les plus connus : RPM et APT).

Une distribution peut par exemple choisir de se spécialiser sur GNOME ou KDE. Elle est également responsable de la configuration par défaut du système (graphisme, logiciels préinstallés, etc.), du suivi de sécurité (installations de mise à jour) et plus généralement de l'intégration de l'ensemble.

La diversité des distributions permet de répondre à des besoins divers. Ainsi, suivant la distribution choisie, vous pouvez disposer en quelques clics, installer un système très stable pour un serveur ou simplement pour faire de la bureautique. Il existe également plusieurs distribution ultra spécialisé comme par exemple SMOOTHWALL ou IPCOP qui sont idéaux pour faire un pare-feu sur une machine dédiée ou GeeXboX, distribution à caractère embarqué qui se propose de transformer votre ordinateur en Media player !

### L'annonce de linux par Linus Torvalds

*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.*

*ALERT! WARNING! NOTE! These sources still need minix-386 to be compiled (and gcc-1.40, possibly 1.37.1, haven't tested), and you need minix to set it up if you want to run it, so it is not yet a standalone system for those of you without minix. I'm working on it. You also need to be something of a hacker to set it up (?), so for those hoping for an alternative to minix-386, please ignore me. It is currently meant for hackers interested in operating systems and 386's with access to minix.*

*The system needs an AT-compatible harddisk (IDE is fine) and EGA/VGA. If you are still interested, please ftp the README/RELNOTES, and/or mail me for additional info.*

*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 enjoyed 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*

*PS. to PHIL NELSON! I'm unable to get through to you, and keep getting "forward error - strawberry unknown domain" or something.*

# Démarrez vos applications Windows sous Linux

Wine est une solution permettant aux applications développées et compilées pour Microsoft Windows de tourner sous Linux. WINE est un acronyme récursif pour « Wine Is Not an Emulator ». Si WINE n'émule pas un processeur ou un environnement matériel, comme Bochs ou VMWare, WINE émule l'API de Windows et prend en charge les fichiers .exe/.dll. En ce sens, WINE est bien un émulateur…

## La brève histoire de Wine

Wine est né en 1993, à l'époque où la priorité pour ses développeurs était de permettre aux applications Windows 16 bits (Windows 3.1) de tourner sur d'autres systèmes d'exploitation.
Depuis cette époque, Wine a suivi l'évolution de Windows et fournit une compatibilité suivie avec les applications de Microsoft : en 1995, Wine supporte les applications Win32 au lieu du simple 16-bits, et en 2001, Wine supporte DirectX et le démontre en permettant l'exécution de plusieurs jeux vidéos.
En l'espace de 10 ans, Wine s'est enrichi de près d'1,4 million de lignes de code et a été soutenu par plus de 700 contributeurs. L'aventure Wine est belle et bien lancée et continue toujours en 2008 : le 13 Juin 2008 est le jour de la dernière version 1.0 de Wine : Wine 1.0-rc5. Désormais, les aficionados que nous sommes doivent se tenir prêts pour les futures améliorations de Wine ! On parle notamment d'un support pour les applications Windows 64 bits.
Aujourd'hui Wine gère :

- Les applications MS-DOS, Windows 16 bits (Win 3.1) et Win32 (Windows 95/98/NT/2000/XP)
- La compatibilité avec des fichiers .DLL tierces parties
- La conversion de l'affichage graphique Windows avec un affichage graphique Linux (XWindow)
- DirectX

Alors, Wine, solution idéale pour se débarasser définitivement de Windows ? C'est ce que nous allons vérifier dans ce numéro.

## Installation de Wine

Wine est diffusé sous forme d'archives pour votre système d'exploitation ou de code source compilable sur votre système. Vous pouvez trouver les différents packages ou manuel d'installation pour votre système à l'adresse : http://www.winehq.org/site/download. Les pages sont en anglais. Sous Debian, un apt-get update puis apt-get install wine sous root devraient toutefois faire l'affaire.

## Configuration

Le fichier de configuration de wine, wine.conf, est extrêmement long. Heureusement pour nous, il ne faut en modifier que quelques lignes : en effet, le fichier **wine.conf** par défaut est très bien, mais il n'est pas adapté à vos différents disques. Nous partirons donc du fichier fourni avec votre installation de Wine. Pour commencer, vous pouvez éditer ce fichier et y supprimer toutes les sections nommées [Drive X] où X est une lettre entre A et Z pour plus de clarté, mais laissez les autres, elles sont très bien.
Dans notre exemple, nous disposons de deux disques dédiés à Windows qui sont montés sur **/mnt/win_c** et **/mnt/win_d**, un lecteur de disquette (**/dev/fd0**) monté sur **/mnt/disquette** et un lecteur de cdrom (**/dev/hdc**) monté sur **/mnt/cdrom.** Le principe restera le même si vous avez une autre configuration.

Pour chacun des disques dont vous disposez, il faut ajouter une section nommée **[Drive X]** dans le fichier **wine.conf** où **X** est le nom du disque sous Windows. Par exemple, votre lecteur de disquette se trouvera sur la section nommée **[Drive A]**. Pour chaque section, vous pouvez définir les variables suivantes :

**'Path' :** le répertoire dans lequel est monté le lecteur en question (cette variable est obligatoire).

**'Type' :** le type de lecteur en question. Elle peut prendre les valeurs suivantes : **'floppy'** pour un lecteur de disquette, **'hd'** pour un disque dur, **'cdrom'** pour un lecteur de cdrom, et **'network'** pour tous les autres cas. (cette variable est obligatoire)

**'Label' :** le nom de ce lecteur sous Windows (cette variable est facultative)

**'Serial' :** le numéro de série de ce lecteur sous Winodws (cette variable est facultative)

**'Filesystem' :** le type de système de fichier de ce lecteur. Les valeurs possibles sont : **'msdos'** ou **'fat'** pour un disque en **FAT16**, **'win95'** ou **'vfat'** pour les autres types de disque. Il existe une valeur **'unix'** mais elle est déconseillée. (cette variable est facultative)

**'Device' :** le nom de périphérique sous Linux. Vous ne devez renseigner cette variable que si vous voulez que Wine accède directement au périphérique sans passer par les routines de gestion de fichier de Linux. Ce n'est conseillé que pour les lecteurs de disquettes et de cdrom.

Fichier wine.conf :

```
[Drive A]
Path=/mnt/floppy
Type=floppy
Label=Floppy
Serial=87654321
Device=/dev/fd0 [Drive C]
Path=/mnt/win_c
Type=hd
Label=win-c
Filesystem=win95 [Drive D]
Path=/mnt/win_d
Type=hd
Label=win-d
Filesystem=win95 [Drive E]
Path=/mnt/cdrom
Type=cdrom
Label=CD-Rom
Filesystem=win95
Device=/dev/hdc
```

Attention : l'utilisateur devra avoir le droit d'écrire sur les disques pour lesquels la variable **'Device'** est renseignée. Je vous conseille d'ajouter deux autres lecteurs (qui n'existe pas pour Windows) : un disque pour les fichiers temporaires (le même que le répertoire /tmp de Linux) et le disque qui correspondra au répertoire home de l'utilisateur :

```
[Drive F]
Path=/tmp
Type=hd
Label=Tmp Drive
Filesystem=win95 [Drive G]
Path=${HOME}
Type=network
Label=Home
Filesystem=win95
```

Voilà, il ne reste plus qu'à modifier la section [wine] de façon à ce que le disque temporaire soit pris en compte correctement :

```
[wine]
Windows=c:\windows
System=c:\windows\system
Temp=f:\
Path=c:\windows;c:\windows\system
# Profile=c:\windows\Profiles\Administrator
GraphicsDriver=x11drv
```

Si vous utilisez des profiles sous Windows vous pouvez décommenter la ligne profile en la modifiant pour qu'elle corresponde à votre installation. Maintenant, Wine doit pouvoir fonctionner.

## Test d'installation : Adobe Photoshop 9

Voilà Wine installé, voyons maintenant si nous pouvons installer Adobe Photoshop, application aussi populaire qu'indispensable à de nombreux graphistes.

Notons tout d'abord qu'après l'installation, il est recommandé de faire tourner Wine sous le nom de l'utilisateur sous lequel votre session graphique est installée.

Munissez-vous ensuite du CD-ROM de Photoshop ou copiez ses fichiers dans un répertoire de votre disque puis assurez-vous que votre utilisateur a les droits de lecture sur le répertoire d'installation. Pour monter votre CD-ROM, si votre distribution ne le fait pas automatiquement, utilisez simplement mount dans une console avec l'option unhide en tant que root :

```
# mount -t iso9660 -o unhide /dev/hdc /mnt/cdrom
```

Vous prendrez bien sûr soin de remplacer /dev/hdc par le nom de périphérique adéquat. L'option –t n'est pas toujours nécessaire et l'option unhide permet d'afficher les fichiers cachés. Si vous ne connaissez pas le nom du périphérique associé à votre CD-ROM alors lisez le fichier /var/log/dmesg pour l'obtenir.

```
root@linux:/mnt# cat /var/log/dmesg | grep ROM
hdc: VMware Virtual IDE CDROM Drive, ATAPI CD/DVD-ROM drive
hdc: ATAPI 40X DVD-ROM DVD-R-RAM CD-R/RW drive, 2048kB Cache
Uniform CD-ROM driver Revision: 3.20
root@linux:/mnt#
```

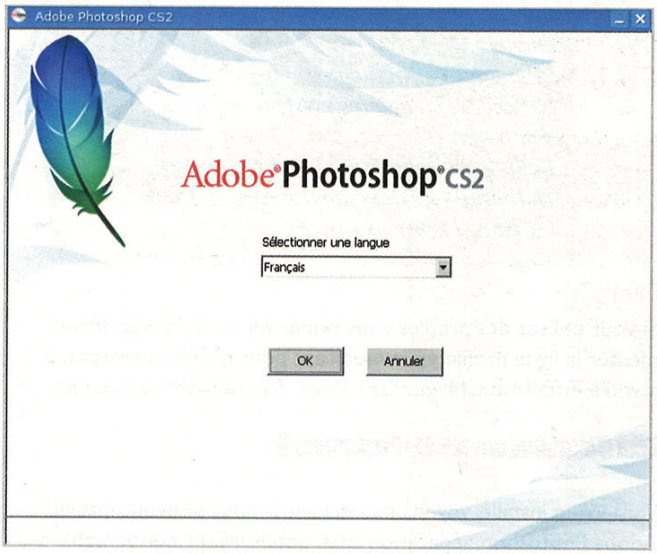
Une fois le CD-ROM ou les fichiers de Photoshop disponibles dans un répertoire de votre système Linux vous allez pouvoir lancer le Setup. Rien de plus simple pour y parvenir : placez-vous dans le répertoire contenant le Setup.exe puis lancez Wine avec comme paramètre le nom du setup. N'oubliez pas de repasser avec l'utilisateur qui utilise la session graphique :

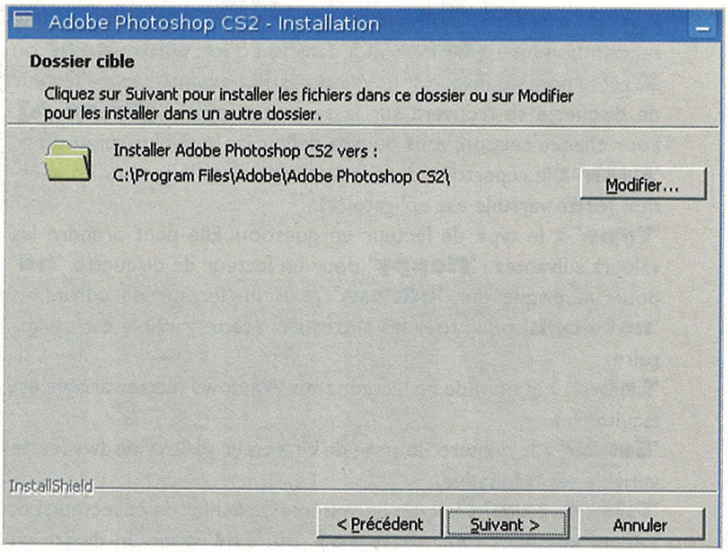
```
$ wine Setup.exe
```

Le lancement du Setup a l'air de fonctionner convenablement. On procède aux différentes étapes en cliquant « Next ». Sur la fenêtre qui nous demande de renseigner le nom d'utilisateur, l'organisation et le numéro de série on modifiera bien évidemment les valeurs d'origine proposées par défaut.

A noter que nous avons eu un plantage conséquent de Wine 1.0-rc5 lors de l'étape suivante en voulant modifier le répertoire d'installation :

>>

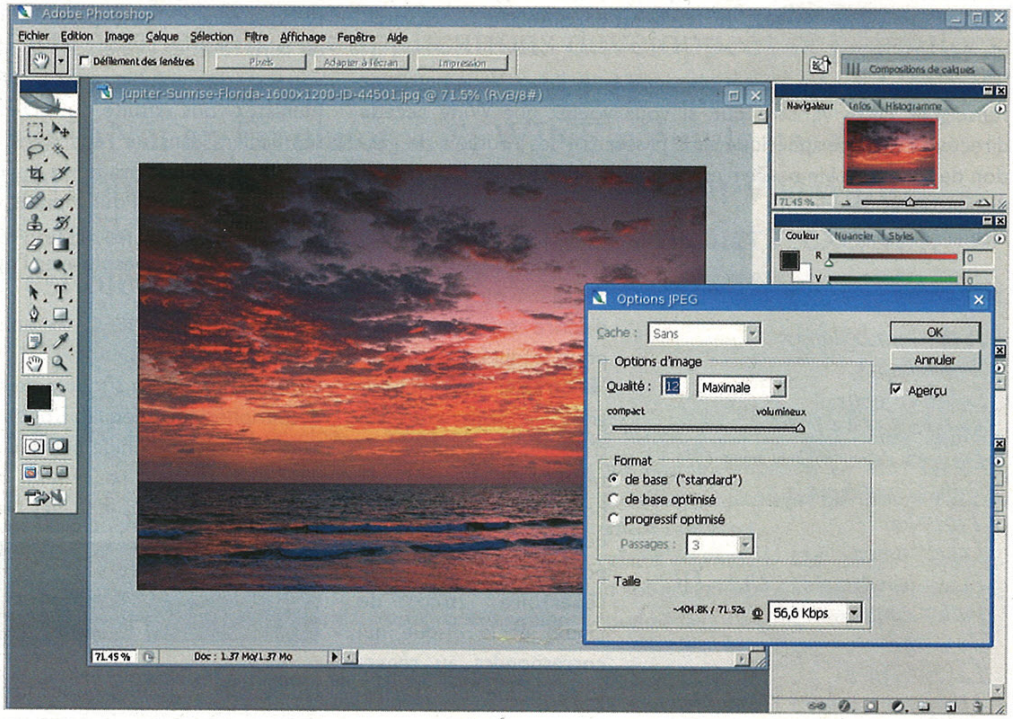



On se contentera donc de cliquer sur « Next » sans trop vouloir fignoler sur l'installation. Celle-ci se déroule sans autre accroc.

Maintenant que nous avons installé Photoshop, il va falloir le lancer puis s'assurer qu'il est bel et bien fonctionnel. Mais où donc Wine a-t-il caché Photoshop ?

En effet, contrairement à Windows, il n'y a pas nativement de raccourci qui mène vers Photoshop. Pas de menu démarrer ou autre. Libre à vous de créer vos raccourcis sur le bureau ou scripts shell qui vous permettront de lancer Photoshop.




Les fichiers de Photoshop sont installés dans le répertoire home de l'utilisateur ayant procédé à l'installation. Vous vous souvenez du C:\Program Files\Adobe\... que vous proposait le Setup.exe lors de l'installation ? Cela se traduit par une installation des fichiers dans le répertoire « ~/.wine/drive_c/Program Files/… ». Ou, autrement dit, en tapant les deux commandes suivantes, vous parviendrez à lancer Photoshop sans difficulté :

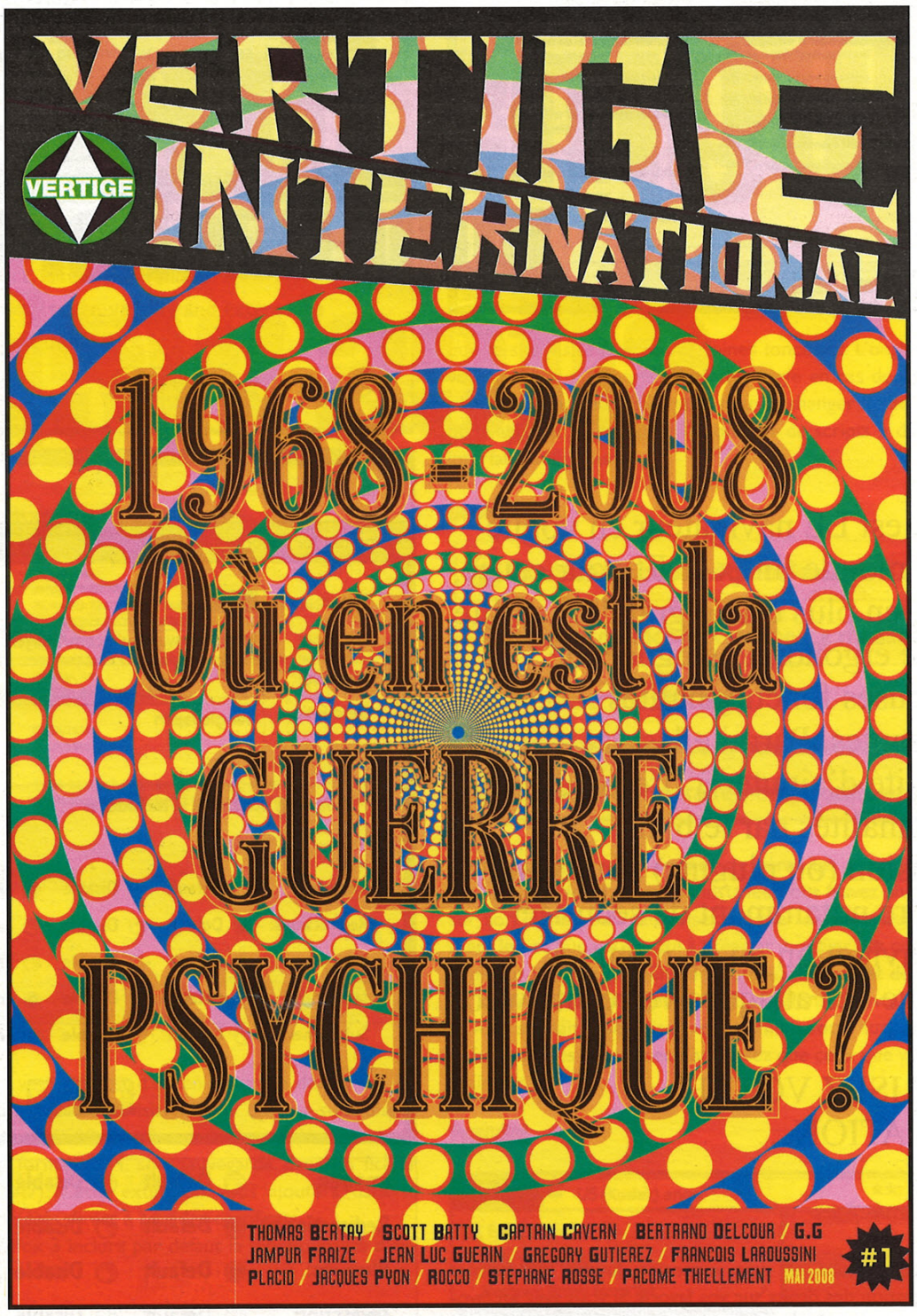
```
clad@linux:~$ cd ~/.wine/drive_c/Program\ Files/Adobe/Adobe\ Photoshop\ CS2/
clad@linux:~/.wine/drive_c/Program Files/Adobe/Adobe Photoshop CS2$ wine Photoshop.exe
```

Nous avons procédé à différents tests sur les images (ajouts de calques, redimensionnement, utilisation des outils Photoshop, …) et n'avons pas trouvé de bug immédiat dans l'utilisation du logiciel. Le parcours de l'arborescence du disque, contrairement au bug rencontré lors de l'installation, ne s'est pas répété lorsque nous avons voulu enregistrer notre travail. En conséquence nous pouvons affirmer que Photoshop tourne très bien sous Linux.

## Test numéro 2 : Microsoft Office 2007

Poursuivons nos aventures avec Wine. La deuxième « killer application » qui justifie bien souvent l'utilisation de Windows est la suite Office de Microsoft. Bien qu'égalée par la suite OpenOffice, l'utilisation de cette dernière en milieu professionnel est encore marginale. Est-il possible de se passer de Windows et de bénéficier néanmoins des atouts de la suite Office de Microsoft ?

# Le 1er magazine de contre-culture




Chez votre marchand de journaux

# Utilisez les plugins firefox pour hacker vos sites web !

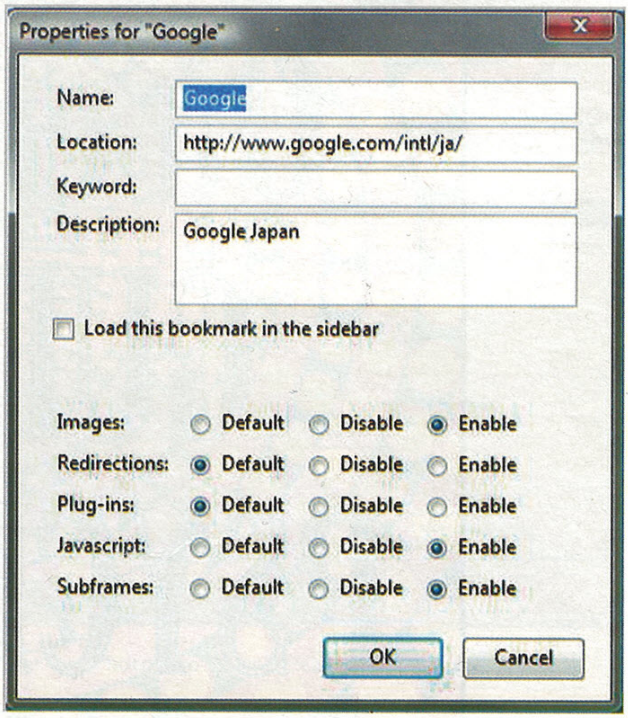
Firefox est LE navigateur en vogue dans le monde linux… Et pour cause, il offre, en plus d'une grande stabilité et d'une ergonomie soignée, de nombreuses fonctionnalités intéressantes. Parmi elles, la possibilité d'ajouter à firefox des fonctionnalités par le biais d'extension (ou plugins) pour vous permettre notamment de sécuriser votre navigation ou de mettre à l'épreuve des attaques votre site web.

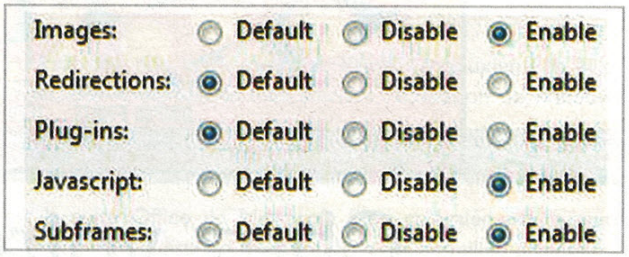
## SÉCURISEZ VOTRE NAVIGATION…

### Bookmark permissions

Bookmark permissions est une extension firefox qui permet d'avoir une gestion des droits sur vos favoris. Si vous avez en favoris quelques sites « douteux » que vous suspectez d'exécuter certain script pouvant être nuisible, alors cette extension est faite pour vous ! En effet, vous pouvez paramétrer 5 réglages pour tous vos favoris :

• **Images** vous permet d'autoriser ou non l'affichage d'images sur un site. Cela peut s'avérer utile si vous en avez marre des centaines de pubs qui traînent sur certains sites.

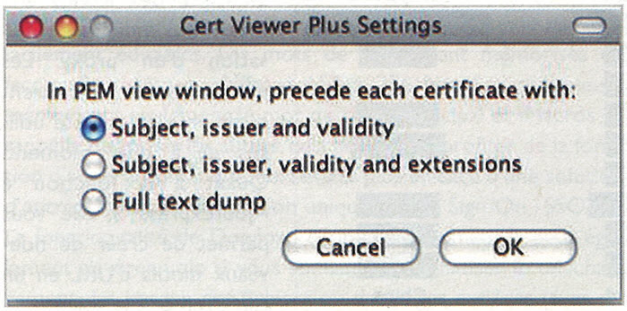


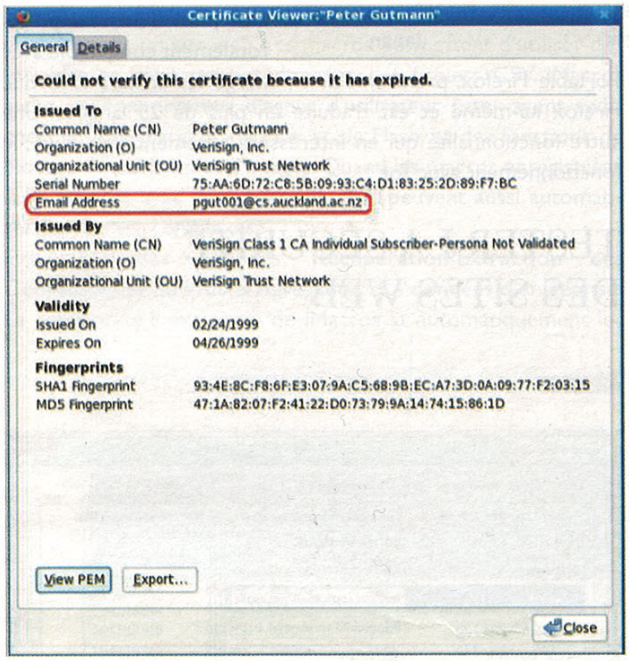




• **Javascript** permet le contrôle des scripts javascript, utile pour certains sites de warez qui utilise parfois certains scripts plus que douteux.

• **Redirection** évite que vous ne soyez automatiquement redirigé vers un site que vous ne souhaitez pas visiter.
• **Plug-ins** et **Subframes** ont, quant à eux, un rôle moins indispensable.
Pour chacun de ces paramètres, vous avez le choix entre 3 options : enable, disable et default.

## Certs Viewer Plus

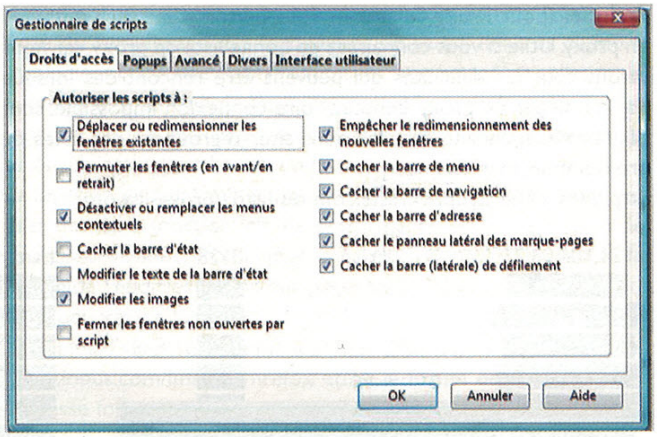



Cette extension vous permet d'obtenir plus de détails sur les certificats des sites que vous visitez. Deux options sont disponibles : un certificat X.509 peut être soit ouvert dans une nouvelle fenêtre au format PEM, soit sauvegardé dans un fichier (PEM/DER/PKCS#7). Cette extension vous ajoutera dans le menu option un accès direct à tous les certificats.
A noter que Firefox 3 inclura par défaut la fonctionnalité d'export de ce plugin.

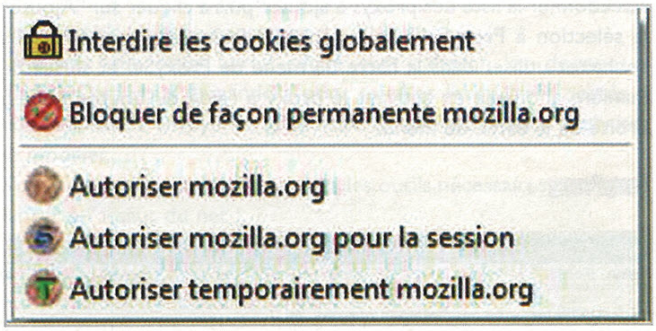
## Contrôle de scripts

Cette extension permet d'ajouter de nombreuses fonctionnalités à l'interface de contrôle JavaScript native de firefox. Vous pouvez notamment spécifier à quelles fonctions ou propriété JavaScript un site à droit d'accès ou de modification. Vous pouvez également dans l'onglet « popups » générer des alertes lors-



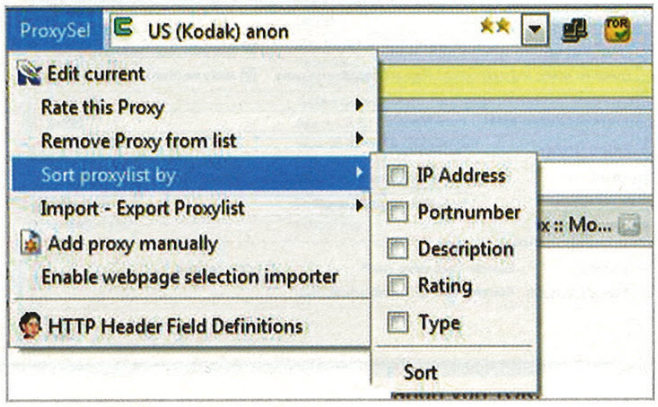

que le site visité exécute certaines fonctions. L'onglet « Avancé » vous permet de créer de véritable règle en cas de tentative d'accès à une propriété. Enfin, vous pouvez changer le raccourci clavier pour accéder au menu de « Contrôle de scripts » dans l'onglet « interface utilisateur ».

## CookieSafe

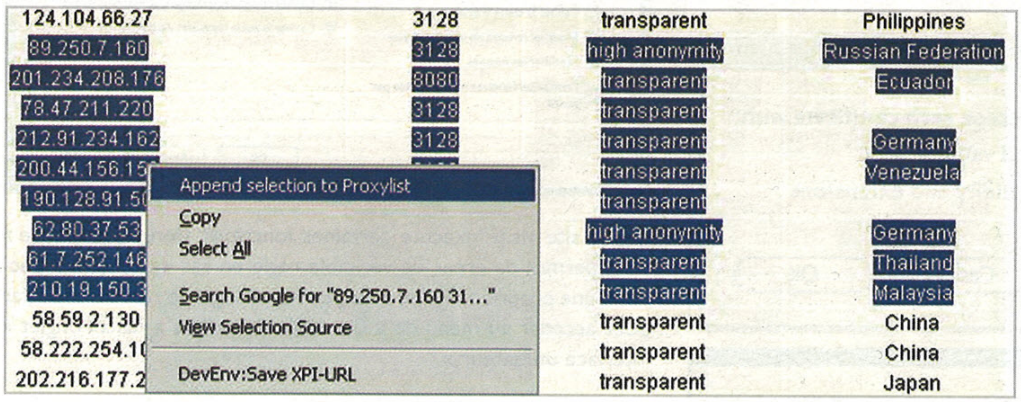



Cette extension vous permet de gérer les permissions sur vos cookies. Une fois l'extension installée, vous n'avez plus qu'à cliquer sur son icône pour : autoriser, bloquer ou autoriser temporairement le site que vous visité à créer et modifier des cookies. Un clic droit sur la barre de menu de CookieSafe permet de visionner ou de supprimer les cookies et les exceptions que vous avez créées. Si vous souhaitez sécuriser au maximum votre surf, désactivez par défaut tous les cookies et n'autorisez que ceux que vous souhaitez vraiment.

## ProxySel



ProxySel (pour Proxy Selection) vous permet de surfer à travers un proxy. Utile si vous connaissez de bonne liste de proxy, car nous savons tous les difficultés qui peuvent être rencontrées lorsque l'on passe par un proxy. Lenteur, voire connexion impossible, sont courants. Cependant, certains bons sites regroupent des listes de proxys anonymes tels que :
http://www.checkedproxylists.com ou http://proxy-list.org

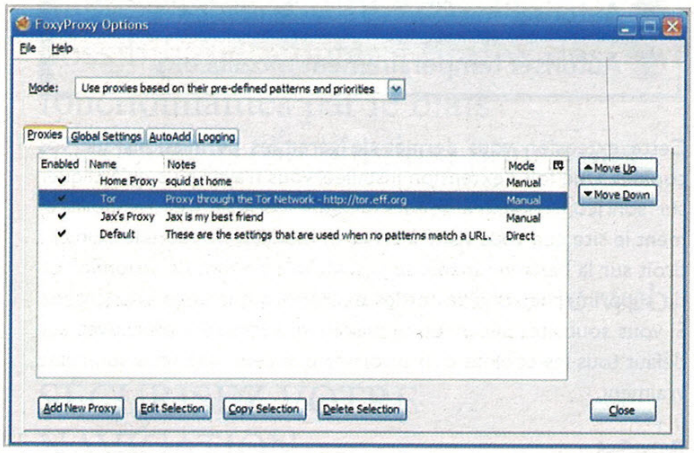


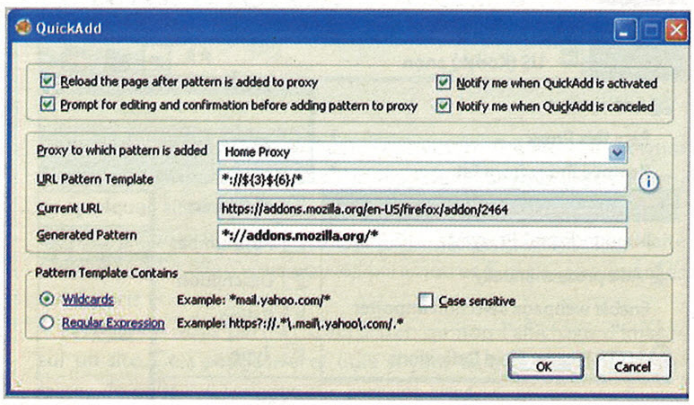

L'ajout des nouveaux proxys est très simple puisqu'il suffit de sélectionner la liste des proxys à ajouter puis à cliquer sur "Ajouter la sélection à ProxySel". Sélectionnez ensuite le proxy que vous souhaitez utiliser dans la barre de menu de ProxySel et surfez de manière anonyme en activant le proxy à l'aide du bouton situé à droite de la barre de menu.

## FoxyProxy

FoxyProxy est une extension pour Firefox permettant de se connecter à Internet via des serveurs proxies en se basant sur des motifs d'URL. En utilisation basique, Foxyproxy automatise le réglage manuel opéré dans la boîte de dialogue des paramètres proxy de Firefox. Le basculement de proxy se fait à partir du chargement de l'URL et des règles de basculement que vous avez définies.

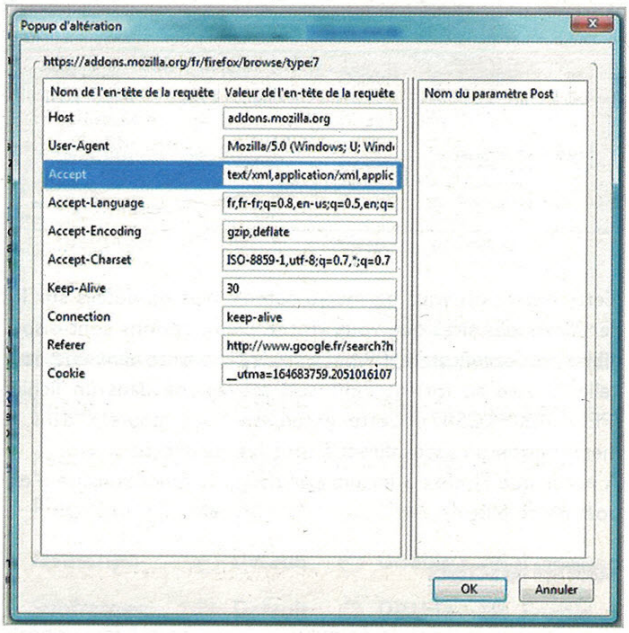
Parmi les fonctionnalités avancées, des icônes animées vous indiquent l'utilisation d'un proxy. Les menus vous montrent quels proxies ont été utilisés et à quel moment. Quant à la fonction « AjoutExpress », elle vous permet de créer de nouveaux motifs d'URL en un clin d'œil et à la volée.
De plus, FoxyProxy est totalement compatible avec Portable Firefox, prend mieux en charge les fichiers PAC que Firefox lui-même et est traduite en plus de 25 langues. Une autre fonctionnalité qui en intéressera sûrement plus d'un : le fonctionnement avec Tor !




# TESTER LA SÉCURITÉ DES SITES WEB

## iMacros

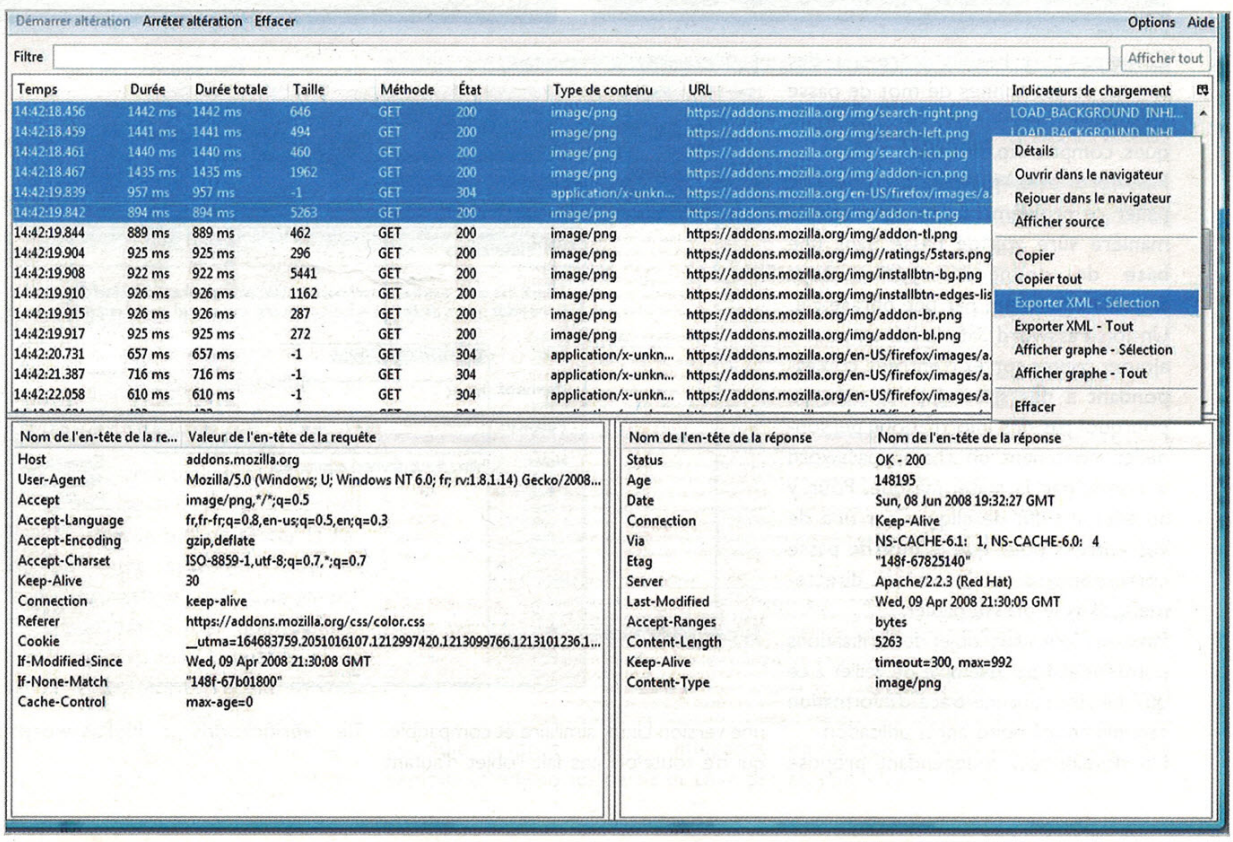



Voici quelques exemples sur comment utiliser iMacros pour automatiser votre navigateur web et enregistrer et refaire les activités répétitives. Très utile donc pour automatiser tout type de taches répétitives et ennuyeuses que certains sites vous infli-

gent… Ou pour lancer des tests de résistance de charge… (qui a dit DOS ?!)
Parmis les fonctionnalités d'iMacros, les plus intéressantes sont : Grâce au remplisseur de Formulaires & Gestionnaire de Mots de Passe, vous n'avez plus besoin de contrôler les mêmes sites quotidiennement, en vous rappelant les mots de passe et en remplissant les formulaires de saisie. iMacros est le seul remplisseur de formulaire qui peut remplir automatiquement les formulaires longs de plusieurs pages. Toutes les informations sont enregistrées dans un format lisible, des fichiers texte en clair facilement éditables. Les mots de passe sont mémorisés de façon sûre avec un chiffrage AES à 256 bits. Les utilisateurs mémorisent seulement le mot de passe principal et iMacros se rappelle de toutes les autres, cela permet de profiter de la fonction d'accès automatique, qui résulte plus efficace d'une solution d'entreprise d'authentification unique (Single Sign-On : SSO).
La fonctionnalité de Dowload & upload Automatisés peut également s'avérer utile si vous souhaitez automatiser le téléchargement des images, des fichiers ou des pages entières (avec ou sans images). Dans l'autre sens, il peut automatiser l'upload des données vers un site web. Les macros permettent d'utiliser des variables et importer des données des fichiers CSV. iMacros inclut un commutateur d'agent d'utilisateur (user agent switcher), un téléchargeur de PDF et de Flash, et des fonctions de blocage de publicité et d'images. Quand les macros enregistrées sont utilisées avec des Javascript, elles peuvent aussi automatiser des taches complexes.
Extraction des données, Récupération/Extraction des Connaissances du Web & Agrégation des données d'entreprise
La commande « extraire » de iMacros lit automatiquement les données d'un site web et les exporte dans des fichiers CSV – l'opposé du remplissage des formulaires de saisie. iMacros supporte totalement l'Unicode et gère toutes les langues, même celles multibyte comme le chinois. Vous pouvez par exemple utiliser cette fonction pour télécharger et comparer les prix des magasins virtuels sur internet et beaucoup plus…
La fonction de Test Web est, quant à elle, incontournable pour réaliser des tests fonctionnels, de performance et de régression des applications web. La commande intégrée STOPWATCH capture les temps de réponses précis des pages web... (à vous de voir ce que vous pourrez en faire ;))
Pour plus d'idées sur comment utiliser iMacros, vous pouvez visiter www.iopus.com/imacros/firefox et/ou le forum de support actif à l'adresse forum.iopus.com.

## TamperData

Si il n'y avait qu'une seule extension firefox à installer, ce serait celle là ! Cette extension est INDISPENSABLE pour quiconque souhaite tester la sécurité des formulaires de son site web…
En effet, TamperData vous permet d'altérer toutes les données postées à la volée, d'afficher l'en-tête de réponse du serveur, le tout dans la même fenêtre. De plus, vous pouvez exporter ce qui vous semble utile de conserver ;)
Une fois que vous aurez cliqué sur « altérer les données », TamperData vous demandera pour tous les formulaires postés si vous souhaitez envoyer le formulaire tel quel ou si vous souhaitez le modifier.
Vous avez maintenant en main tous les outils nécessaires pour vous lancer à l'assaut du net !

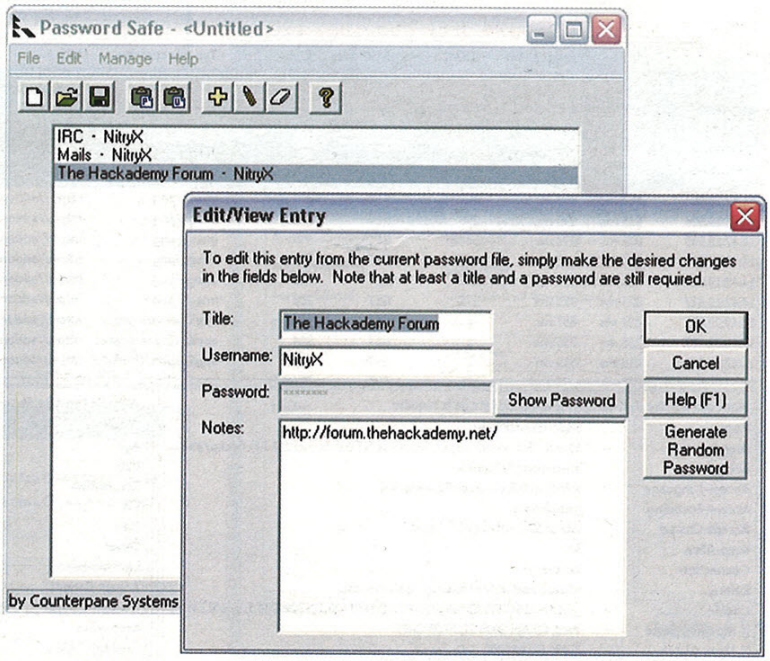


# 3 sites pour les initiés

## Password Safe

**OS :** Windows 9x/2000/XP/CE
**TAILLE :** 172 KiB • *Logiciel libre*
**URL :** http://www.schneier.com/passsafe.html

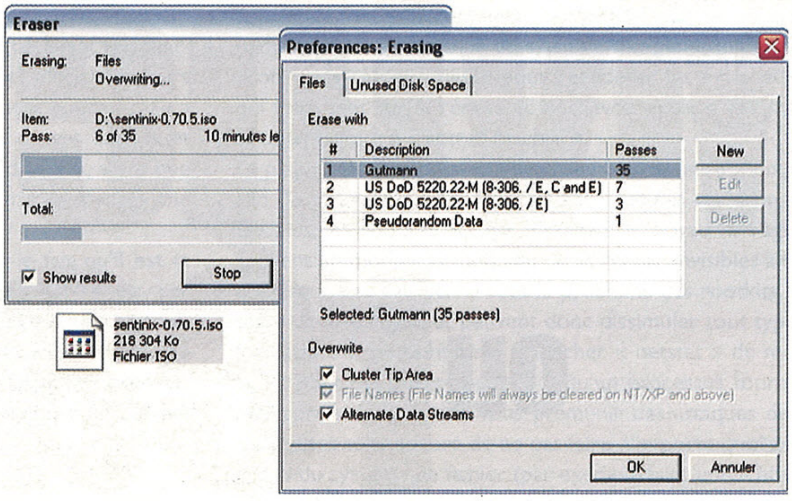
On a toujours besoin de retenir des tonnes et des tonnes de mot de passe : travail, forums, service en ligne, banques, comptes ftp, mail, etc.
Password Safe, permet en effet de palier ce problème en enregistrant de manière sûre vos de passe dans une base de données cryptée (avec Blowfish) protégée par une phrase clé.
Un fois Password Safe lancé on peut y ajouter toutes sortes d'entrées, correspondant à des sites, des logiciels, ou pourquoi pas des informations personnelles, contenant un champ password qui sera, par la suite, masqué. Pour y accéder, il suffit de cliquer sur une de vos entrées pour que le mot de passe correspondant soit transféré directement dans le presse papier.
Passworf Safe a fait l'objet de vérifications poussées afin, par exemple, de veiller à ce qu'il ne reste aucune trace d'information sensible en mémoire, après utilisation.
Un développeur indépendant propose une version Linux, similaire et compatible, qui n'a toutefois pas fait l'objet d'autant de vérifications : MyPasswordSafe (http://www.semanticgap.com/myps/).



# Eraser

**URL :** http://www.tolvanen.com/eraser/
**Taille :** 2.7 MiB • *Logiciel Libre*

Ce programme s'est fait un nom dans le monde de la sécurité Windows. Il s'agit du meilleur outil de suppression sécurisée de fichiers.
Grâce à Eraser vous pouvez enfin supprimer efficacement vos données confidentielles. Vous savez, toutes celles que vous ne voulez absolument pas qu'un tiers puisse retrouver grâce à un logiciel tel que DiskInternals Uneraser. Pour arriver à écraser des données, un des modèles utilisé par Eraser a été élaboré à partir de l'article de Peter Gutmann ("Secure Deletion of Data from Magnetic and Solid-State Memory"). Celui-ci défini précisément comment faire disparaître toute trace magnétique d'un disque dur.
En outre, Eraser ne se contente pas d'ajouter une entrée « Erase » lors du clic droit d'un fichier. Il comporte également un gestionnaire de taches qui pourra vous permettre de supprimer automatiquement les fichiers contenus, par exemple, dans un répertoire temporaire.




# Frsirt

**LANGUE :** Francais
**URL :** http://www.frsirt.com

Trouver de l'information à jour sur la sécurité en français n'est pas évident. Frsirt.com (anciennement connu sous le nom de k-otik) est un site qui répond à ces critères puisqu'il est entièrement en français et totalement dédié à la sécurité informatique. Vous aurez la possibilité de consulter leur base de donnée d'advisories, d'exploits ou de papers. De nombreuses news et articles sont disponibles en ligne. Ce site offre de plus, moyennant une petite participation, de recevoir des alertes personnalisées (par mail ou sms), mais aussi des exploits privés (0dayz ?!) dans le but de tester la fiabilité de votre système. Tous les exploits disponibles sont classés par date et sont mis à jour quotidiennement. Vous aurez aussi la possibilité de recevoir les advisories récent en vous inscrivant à leur mailing liste. Ce site est donc une excellente alternative à Security Focus pour se tenir informer quotidiennement des différents exploits et advisories parus et tout ça en français. Bref, un site à bookmarker au plus vite.

# Les virus Linux... Mythe ou réalité ?

Un des points qui fait le succès de Linux est la sécurité. En effet, vous avez sûrement déjà dit à un de vos amis Windowsien, « Installe un Linux, tu n'auras plus de virus ! ». Mais Linux est-il vraiment insensible aux attaques virales ? C'est ce que nous allons voir…

## Un virus, c'est quoi ?

Un virus est un programme informatique qui a pour but de s'insérer dans un programme « légitime » dans le but de se propager. Ses effets sont la plupart du temps nocifs pour le système hôte et amène pour la majorité à l'effacement total ou partiel de données, à la récupération d'informations personnelles et à des ralentissements significatifs pour le système. Les virus se propagent la plupart du temps via Internet mais peuvent également utiliser des supports tels que les CDRom ou les clés USB pour se propager. On utilise trop souvent le terme virus pour désigner tout type de logiciel malveillant (**malware**). Ce terme désigne, quant à lui, tout programme informatique ayant pour but de nuire à un système. Les malwares peuvent ainsi être des virus, des vers, des trojans, des backdoors, les spyware, des rootkits, des exploits ou des dialers. Tous ces types de programmes sont différents dans leurs fonctionnements et dans leurs effets. Ainsi, les trojans et les backdoors ont pour but de permettre le contrôle de l'ordinateur cible à distance alors qu'un ver a pour seul objectif la prolifération sur tous les postes d'un réseau (Internet est un réseau).

## Alors ? Insensible aux virus ?

Tout d'abord, il faut admettre que Linux n'est PAS insensible au virus. Certes, les virus sont beaucoup moins nombreux sous la plate-forme UNIX, mais cela ne signifie pas qu'il n'y en a pas. Les virus UNIX existent même si leur nombre est plutôt limité. D'ailleurs, quelques-uns des vers les plus anciens prennent pour cible UNIX et pas Windows ! Si vous considérez vos données comme importantes, alors vous devez accepter ces faits.

Par exemple, Bliss est un virus qui démontre clairement que Linux peut être vulnérable aux attaques virales. Ce n'est pas néanmoins le dernier. Par exemple, Winux, avait fait son apparition en 2001. Si, en pratique, le virus découvert par Kaspersky et nommé Virus.Linux.Bi.a/Virus.Win32.Bi.a, est particulièrement simple et d'un impact mineur sur la santé du système, il est le témoin d'une nouvelle tendance qui consiste à attaquer plusieurs système d'exploitation à la fois. En effet, ce virus attaque aussi bien Windows que Linux. Si moins de virus sont disponibles sur Linux, c'est avant tout parce que Linux est moins répandu que le système venu de Redmond. Mais ce n'est pas la

seule raison ! Plusieurs autres explications peuvent être trouvées. Entre autres, le fait que les utilisateurs ne disposent pas par défaut des droits administrateurs et évite ainsi qu'un simple utilisateur modifie le système de fichier. De plus, la gestion des droits UNIX ayant un système de read/write/execute, impossible de se faire berner par un virus « paris_hilton_nude.jpg.exe » ;)
Une autre raison à cela est que les utilisateurs de Linux téléchargent généralement leurs logiciels dans des dépôts officiels de leurs distributions Linux dont le contenu est strictement contrôlé. Il y a rarement besoin de prendre des logiciels hors de ces dépôts, et donc moins de risques de tomber sur un site douteux, alors que sur Windows, il est nécessaire d'avoir dans ces marques pages une bonne dizaine de sites de téléchargements !
Un des autres avantage de Linux repose dans le fait qu'il est le dérivé d'un UNIX (minix, si vous avez suivi ;-)). UNIX est, par essence, axé sur la sécurité, contrairement a DOS qui, lui, à sa création, n'avait pour seul objectif que la simplicité ! Et windows (même Vista !) comportant toujours quelques briques DOS (les exécutables Windows au format WinPE comportent toujours un en-tête DOS), vous imaginez la suite….

## Bon ! J'installe un anti virus alors ?!

Bon, alors, pas de parano inutile, les virus Linux, même s'il existe, ne sont guerre « en liberté ». La plupart des hackers qui développent des virus ou des vers Linux ne souhaitent pas mettre en péril un système qu'ils affectionnent particulièrement. En revanche, les attaques par rootkits sont un peu plus développées. En effet, les rootkits sont des outils exploités par les pirates qui permettent de s'approprier le compte root. Et ça, c'est plus intéressant ! La plupart des rootkits sont installés par une personne qui a un accès physique (ou SSH) à l'ordinateur cible. Le rootkit exploite une faille du système d'exploitation et/ou des protocoles réseau utilisés pour récupérer les mots de passe qui transitent sur le réseau et ainsi avoir accès à des services sécurisés, voire à d'autres machines.
La plupart des rootkits installent une backdoor sur la machine afin de pouvoir se reconnecter discrètement sur la machine. De plus, les rootkits actuels modifient le noyau (il s'agit bien souvent un module kernel) afin de se rendre invisibles à l'administrateur. Et c'est bien là tout le problème des rootkits. Il s'intègre dans le noyau et peuvent donc dissimuler tout type d'information. Ils peuvent ainsi empêcher « netstat » de montrer qu'un port est ouvert ou « ps » qu'un processus tourne ! La meilleure solution pour vous prémunir des attaques de code malveillant sont encore de ne pas faire n'importe quoi avec les droits du système de fichier (par exemple faire un « chmod –R 777 / »), laisser le moins de service possible tourner, qu'il s'agisse de SSH, FTP ou http, et de lancer de temps à autre « rootkit hunter » si vous avez des doutes sur un ami à qui vous permettez gentiment de se connecter en SSH sur votre machine.

# Maîtriser les outils Linux

Dans un environnement où les fichiers sont très nombreux, et où les besoin de recherche et traitement sur ceux-ci sont très divers, il est plus que nécessaire de maîtriser quelques autres outils console puissants et disponibles sur tous les systèmes UNIX/Linux.

## Les redirections

Sous linux, la plupart des entrées sorties se font sous forme de flux, canalisés vers l'une ou l'autre sortie. Tout comme il est possible de détourner un fleuve, il est possible, et même pratique, de détourner ces flux de données pour agir dessus en temps réel. Il est par exemple fort utile de rediriger les informations qui s'affichent à l'écran vers un fichier texte, ou vers un périphérique spécial dans le cas par exemple des utilisateurs non voyants. Ces redirections sont effectuées à l'aide de ce que l'on appelle les opérateurs de redirection.

• L'opérateur '>' : il permet de rediriger un flux vers la sortie de son choix. La sortie est créée, et si elle existe déjà, elle sera créée, sauf dans un cas très particulier qui est /dev/null. /dev/null est un périphérique extrêmement utile, que l'on pourrait comparer au néant : c'est un trou sans fond, et ce qui y est redirigé y est donc englouti. Une sorte de tonneau des danaïdes version unix, quoi.

```
bash-2.05$ ls
img
lib
404.php
code.php
contact.php
index.php
links.php
unix.php
bash-2.05$ ls > toto
```

Rien ne s'affiche car le flux a été redirigé vers le fichier toto au lieu de la sortie standard.

```
bash-2.05$ ls
img
lib
404.php
code.php
contact.php
index.php
links.php
toto
unix.php
```

On s'aperçoit que le fichier toto qui n'existait pas lors de notre premier ls a maintenant été créé.

```
bash-2.05$ cat toto
img
lib
404.php
code.php
contact.php
index.php
links.php
toto
unix.php
bash-2.05$
```

Le contenu du fichier toto est absolument identique à ce qu'avait donné notre premier ls, à une différence près : lorsque nous lançons notre commande, le fichier toto est immédiatement créé, mais il ne contient rien quand la commande est lancée. Lorsqu'il est listé, il fait donc 0 octets, et, à la fin du processus de listing, il fait 912 octets.

• L'opérateur '>>': il a la même fonction que l'opérateur '>' si ce n'est que lorsque la sortie indiquée existe déjà, les données sont ajoutées à la suite de celles existant déjà, au lieu que celles-ci soient écrasées. Si la sortie n'existe pas, elle est créée.

```
bash-2.05$ ls /tmp/* >> toto
```

On s'aperçoit que la sortie standard a bien été redirigée vers notre fichier toto.

```
bash-2.05$ cat toto
img
```

```
lib
404.php
code.php
contact.php
index.php
links.php
toto
unix.php
/tmp/list
/tmp/tutu
bash-2.05$
```

Le listage de notre répertoire s'est ajouté à la suite des données précédemment entrées. Le fichier toto n'a pas été écrasé.
• L'opérateur '|': ou pipe. C'est sans doute le plus puissant des opérateurs de redirection. En effet, il permet de rediriger le flux de sortie d'une commande en entrée d'une autre commande. On peut ainsi faire des lignes de commande contenant une dizaine de pipes et faisant des trucs hallucinants une fois qu'on sait se servir de certains utilitaires de formatage de chaînes de caractères. La notion de pipes semble souvent un peu confuse, mais un exemple simple permet de mieux comprendre.

```
bash-2.05$ cat toto | grep tu
/tmp/tutu
bash-2.05$
```

grep est un utilitaire permettant de faire une recherche de correspondance dans un fichier ou une chaîne de caractères. Il nous a permis d'afficher le contenu de notre fichier toto, mais en le filtrant au travers de l'utilitaire grep de manière à n'avoir en sortie que les lignes contenant la chaîne de caractères "tu".

## Les caractères de contrôle

• Le ';': dans une chaîne d'instructions placées sur la même ligne, le caractère ';' permet de séparer les instructions les unes des autres. Il n'y a aucune relation entre chacune des commandes lancées, et il s'agit simplement d'une liste de commandes à exécuter. Il n'y a pas de contrôle de succès de la première commande avant de passer à la seconde.

```
bash-2.05$ rm *.exe ; echo "toto"
rm: cannot remove `*.exe': No such file or directory
toto
bash-2.05$
```

Même s'il n'existe pas de fichier .exe dans le répertoire courant, la seconde instruction est quand même lancée (le contraire aurait d'ailleurs été un peu bizarre sur un PC n'ayant pas vu un fichier .exe depuis.... hou la la, au moins tout ça).

• Le '&&': le double "et commercial" se place entre deux commandes dans une ligne où l'on désire exécuter plusieurs instructions. Contrairement au ';', le '&&' vérifie le succès d'une tâche avant de passer à la suivante, sinon la ligne de commande s'interrompt et un message est affiché sur la sortie d'erreur.

```
bash-2.05$ rm *.exe && echo "toto"
rm: cannot remove `*.exe': No such file or directory
bash-2.05$
```

• Le '&': il permet de lancer un processus, de le passer en tâche de fond avant de rendre la main à l'utilisateur. La tâche peut cependant être rappelée par la suite, grâce à la commande fg, comme foreground.

```
bash-2.05$ emacs &
[1] 673
bash-2.05$
```

## Les utilitaires ultimes

Ils sont nombreux dans les environnnements *nix les utilitaires très puissants et directement intégrés au shell. S'il est possible d'utiliser linux sans jamais voir une seule ligne de commande, ce serait vraiment très très dommage et ce serait passer à côté de choses VRAIMENT ultimes, notamment grâce aux pipes. Les explications et exemples donnés ci-dessous ne dispensent pas de consulter vos deux meilleurs amis : google et le man. Ils sont en effet une mine de renseignements incomparables dès que vous avez besoin de savoir quelque chose.
Tous les utilitaires dont je vais vous parler ici ne sont pas directement intégrés au shell mais on les trouve sur toutes les distributions et ils sont vraiment utiles ; c'est pour cela qu'il m'a semblé important de les décrire.

• **grep :** avec son cousin egrep, ce sont des outils permettant de faire ce que l'on appelle du "pattern matching", c'est-à-dire une recherche de correspondance au sein soit d'une chaîne, soit d'une expression régulière. Cette seconde possibilité donne alors toute sa puissance à grep. Je vous conseille fortement de faire un man regexp pour étudier les expressions régulières car elles ne sont pas étudiées dans cet article.

```
bash-2.05$ cat /etc/passwd | grep toto42
toto42:x:100:100::/home/toto42/:/bin/tcsh
```

On fait une recherche sur toutes les chaînes contenant l'occurence 'toto42'

```
bash-2.05$ cat /etc/passwd | grep ^n
news:x:9:13:news:/usr/lib/news:
nobody:x:99:99:nobody:/:
bash-2.05$
```

On fait ici une recherche dans le fichier /etc/passwd sur toutes les chaînes commençant par un 'n'.
Dans l'exemple ci-dessus, on utilise directement grep sur tous les fichiers c à la recherche de la string "sock". Grep nous affiche alors les lignes correspondantes avec le fichier dont elles ont été extraites.

• **cut :** cet utilitaire permet de faire des coupes franches dans

des fichiers texte, par exemple pour afficher tous les champs d'une liste située entre deux délimiteurs. Combiné à grep et à d'autres utilitaires du même genre, via les pipes, cet utilitaire permet de faire des trucs de fou (comme tout ce qu'on peut faire en console en fait).

Dans notre exemple, nous allons faire une recherche sur le fichier /etc/passwd, sur les chaînes commençant par n, mais en ne retournant que leur UID et le nom des utilisateurs. La ligne étant un peu compliquée (très simple en fait :-), je vais vous la décrire pas à pas.

- "cat /etc/passwd" parcourt le fihier /etc/passwd.
- "grep ^n" fait une recherche sur les lignes commençant par 'n'.
- "cut -f 3,5" prend les champs 3 et 5.
"-d :" déclare que ces champs sont séparés par le délimiteur ':'.

Ce qui donne :

```
bash-2.05$ cat /etc/passwd | grep ^n
  | cut -f 3,5 -d : > toto
bash-2.05$ cat toto
9:news
99:nobody
100:
bash-2.05$
```

• **sed :** cet utilitaire fait tout sauf le café, ou presque. Nous l'utiliserons ici pour remplacer des occurences dans des chaînes de caractères, ce qui n'est qu'une de ses très nombreuses fonctionnalités. En fait, pour dire tout ce que fait sed, il faudrait un livre entier (il existe : Sed et hawk, éditions o'reilly(9)).

Notre exemple va consister à parcourir le fichier /etc/passwd, extraire les logins UIDs et GIDs des utilisateurs dont le nom commence par 'n', d'afficher ces informations, mais en remplaçant le separateur ':' par des ',', puis, de tout rediriger vers notre fichier.... toto (original n'est-ce pas ?). Une fois de plus, c'est un peu compliqué, donc je vais commmenter pas à pas. Jusqu'à sed, pas de problème, on l'a déjà vu.
sed -e va appliquer sed sur une expression.
'y/:/,/' remplace toutes les occurences de ':' par ','. L'option -s aurait appliqué ce changement sur la première occurence remplacée.

```
bash-2.05$ cat /etc/passwd | grep ^n
   | cut -f 1,3,4 -d : | sed -e 'y/:/,/'
   > toto
bash-2.05$ cat toto
news,9,13
nobody,99,99
bash-2.05$
```

• **sort :** permet de trier les lignes d'un fichier par ordre alphabétique. L'option -n trie dans le bon ordre l'option -r dans l'ordre inverse.

```
bash-2.05$ sort -n /etc/passwd
adm:x:3:4:adm:/var/log:
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
ftp:x:14:50::/home/ftp:
games:x:12:100:games:/usr/games:
```

```
bash-2.05$ grep sock *.c
xaccept.c:#include <sys/socket.h>
xaccept.c:int            xaccept(int sockfd, struct sockaddr *addr, int *addrlen)
xaccept.c:  len = sizeof(struct sockaddr);
xaccept.c:  if ((new_fd = accept(sockfd, addr, &len)) == -1)
xbind.c:#include <sys/socket.h>
xbind.c:#include "libsocks.h"
xbind.c:int      xbind(int sockfd, struct sockaddr *my_addr, int addr_len)
xbind.c:  if ((bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr))) == -1)
xconnect.c:#include <sys/socket.h>
xconnect.c:#include "libsocks.h"
xconnect.c:int  xconnect(int sockfd, struct sockaddr *serv_addr, int addr_len)
xconnect.c:  if ((connect(sockfd, (struct sockaddr *)&serv_addr, sizeof (struct sockaddr))) == -1)
xlisten.c:#include <sys/socket.h>
xlisten.c:#include "libsocks.h"
xlisten.c:int   xlisten(int sockfd, int backlog)
xlisten.c:  if ((listen(sockfd, backlog)) == -1)
xsocket.c:#include <sys/socket.h>
xsocket.c:#include "libsocks.h"
xsocket.c:int   xsocket(int domain, int type, int protocole)
xsocket.c:  int sock;
xsocket.c:  if ((sock = socket(domain, type, protocole)) == -1)
xsocket.c:      printf("Error: socket\n");
xsocket.c:    return (sock);
bash-2.05$
```

```
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
halt:x:7:0:halt:/sbin:/sbin/halt
lp:x:4:7:lp:/var/spool/lpd:
mail:x:8:12:mail:/:
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash
news:x:9:13:news:/usr/lib/news:
nobody:x:99:99:nobody:/:
operator:x:11:0:operator:/root:/bin/bash
root:x:0:0::/root:/bin/tcsh
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
sync:x:5:0:sync:/sbin:/bin/sync
uucp:x:10:14:uucp:/var/spool/uucppublic:
```

• tr : dans le genre utilitaire utile, tr se pose là : il sert à remplacer toutes les occurences d'un caractère affichable par une autre. Si cela peut être parfaitement inutile (comme le montrera notre exemple 1), il peut aussi être très pratique lorsqu'on travaille sur de gros fichiers.

```
$ echo "Je suis un linuxien d'elite, le pingouin
       c'est bien, linux aussi" | tr eslota
       351074
J3 5ui5 un 1inuxi3n d'31i73, 13 ping0uin c'357
bi3n, 1inux 4u55i
```

Voilà, vous savez maintenant écrire en warlordz sans vous fatiguer. Complètement inutile, donc complètement indispensable !!! Mais voyons quelque chose d'un peu plus sérieux maintenant.

```
bash-2.05$ cat /etc/passwd | grep ^n
 | cut -f 1,3,4 -d : | sed -e 'y/:/,/'
 | tr n N> toto
bash-2.05$ more toto
News,9,13
Nobody,99,99
bash-2.05$
```

Voilà, nous avons repris l'exemple vu plus haut, mais en mettant les n en majuscule, parce que ça fait plus propre.

• **wc :** word count est un utilitaire bien pratique qui permet de compter le nombre de lignes, de mots et de lettres d'un fichier texte. Ça peut sembler un peu inutile, mais vous comprendrez très vite que ça peut se montrer bien utile au contraire.

```
  wc -w compte le nombre de mots dans un fichier.
  wc -l compte le nombre de lignes.
  wc -L donne la longueur de la plus longue
ligne.
  wc -c compte le nombre de caractères.

bash-2.05$ wc /etc/passwd
      18       18      579 /etc/passwd
bash-2.05$
```

Mon fichier /etc/passwd contient 18 lignes, 18 mots et 579 caractères. Les mots sont comptés à chaque espace, c'est pour cela que wc en compte 18.

# INFORMATIONS SUR LES COMMANDES

## Quelle commande utiliser pour faire ... ceci ou cela ?

**apropos mot_clef ou man -k mot_clef**
• affiche les commandes, brièvement définies, en rapport avec mot_clef.

**apropos copier**
• affiche les commandes en rapport avec la copie d'un fichier, d'une chaîne, d'une zone mémoire ...

**apropos permission**
• affiche les commandes liées à la vérification et à la modification des permissions.

Notes : les noms communs et les verbes à l'infinitif permettent généralement de trouver facilement la commande recherchée. En cas d'échec, pensez aux synonymes : apropos supprimer fait apparaître la commande rmdir (supprimer un répertoire), alors que apropos effacer fait apparaître la commande rm (effacer un fichier).

## Comment se définit cette commande ?

**whatis nom_commande ou man -f nom_commande**
• affiche une brève définition de nom_commande.

**whatis whatis**
• affiche la définition de whatis.

**whatis arch**
• affiche la définition de la commande arch.

**cd /bin ; for i in * ; do whatis %i ; done | more ; cd**
• se positionne dans le répertoire /bin/, affiche page par page la définition de chacune des commandes qui s'y trouve, retourne au répertoire personnel.

## Quelles sont et comment utiliser les commandes internes ?

**help**
• affiche la liste des commandes internes et leur syntaxe.

**help nom_commande**
• affiche une aide sommaire sur nom_commande.

**help help**
• affiche une aide sur help.

**help alias**
• affiche une aide sur la commande alias.

## Où sont et quelles sont les commandes externes ?

**s /bin**
• affiche le contenu du répertoire /bin/, et donc la liste des commandes externes usuelles communes à tous les utilisateurs.

**ls /sbin**
• affiche le contenu du répertoire /sbin/, et donc la liste des commandes externes usuelles réservées à l'administrateur (root).

**whereis nom_commande**
• affiche le chemin de nom_commande ainsi que celui de sa page man.
**whereis cat**
• affiche le chemin de la commande cat (/bin/cat) et celui de sa page man (/usr/share/man/man1/cat.1.bz2).
**which nom_commande**
• affiche le chemin de nom_commande.
**which tac**
• affiche le chemin de tac (/usr/bin/tac).

## Comment obtenir un aide mémoire sur cette commande ?

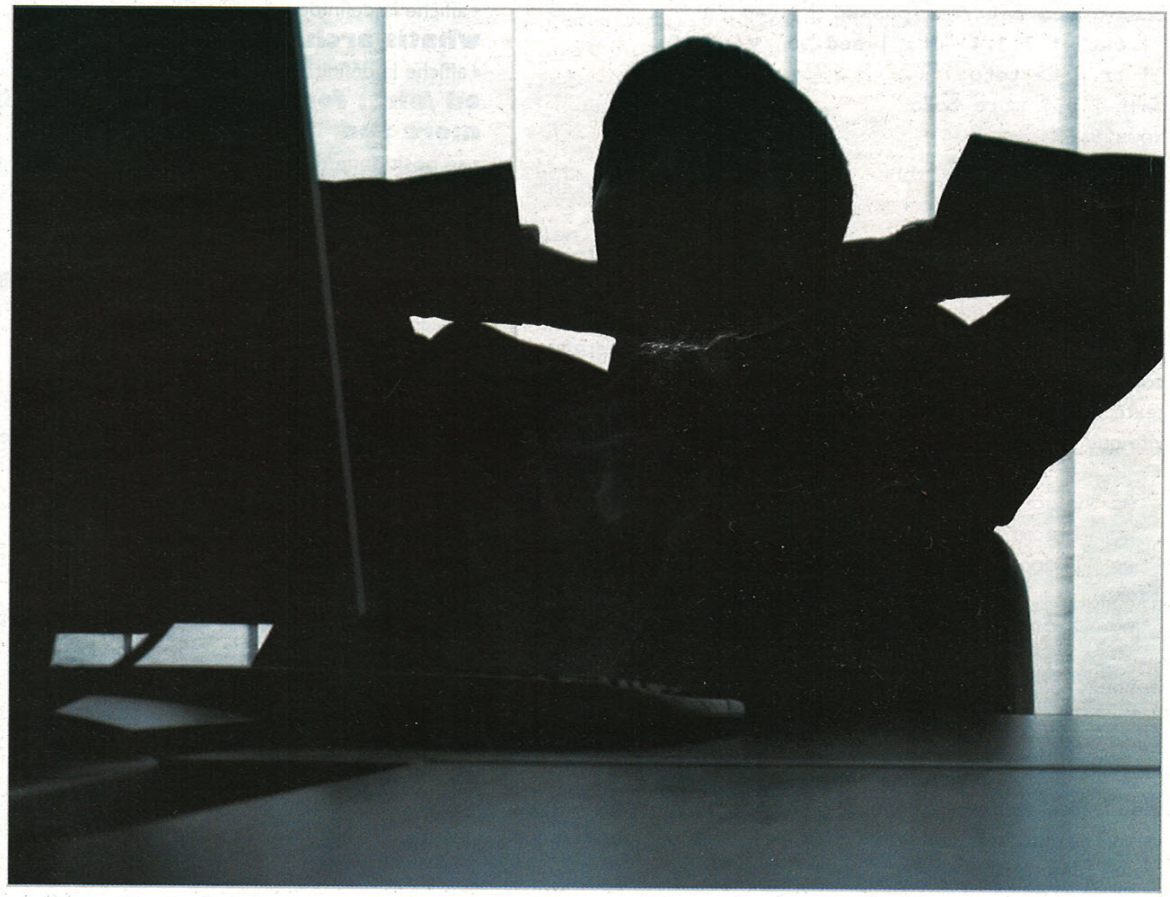
**nom_commande --help**
• affiche l'aide mémoire de nom commande.
**ls --help**
• affiche l'aide mémoire de la commande ls.

## La dernière commande s'est-elle bien terminée ?

**echo $?**
• affiche le code de retour de la dernière commande effectuée, 0 si elle s'est bien terminée, un autre nombre dans le cas contraire.
**clear ; echo $?**
• efface l'écran et affiche 0.
**sl / ; echo $?**
• affiche un message d'erreur et le code 127 (bash ne connaît pas la commande sl).
**nom_commande 2>/dev/null && echo "ok" || echo "m'enfin"**
• exécute nom_commande en redirigeant les erreurs vers /dev/null (périphérique fictif) puis affiche "ok" si tout s'est bien passé ou "m'enfin" dans le cas contraire.
**: && echo "ok" || echo "m'enfin"**
• ne fait rien puis affiche "ok" (la commande : ne fait rien et se termine toujours bien).
**bof 2>/dev/null && echo "ok" || echo "m'enfin"**
• affiche "m'enfin" (la commande bof n'existe pas).

## Quels sont le nom, la taille et le contenu du fichier d'historique ?

**echo $HISTFILE $HISTFILESIZE**
• affiche le nom et la taille du fichier d'historique des commandes.
**cat $HISTFILE | more**
• affiche le contenu du fichier d'historique page par page.
**tail -n 24 $HISTFILE**
• affiche les 24 dernières lignes du fichier d'historique.



# Shell l'interprète idéal pour votre machine

Depuis toujours, vous êtes en manque de conversation avec votre machine… Heureusement, Linux est très bavard et va vous obliger à l'être aussi. Et pour parler à votre système, l'interface la plus courante reste le "shell", un outil de saisie de commandes en mode texte, un peu comme DOS, mais en bien plus puissant…

## Top départ !

Au démarrage de votre Linux, un shell est lancé. De ce poste de contrôle, vous pouvez absolument tout faire. Si vous démarrez directement sur une jolie interface graphique, lancez donc un terminal (une fenêtre où s'affiche un shell), tel "xterm". Vous devriez à présent obtenir un prompt - la ligne d'attente de saisie de commandes :

```
votre_login@host:$
```

(ou simplement [$], [#], [>] ou encore [%])

Le type de prompt dépend du shell choisi (les plus courants étant "bash", "tcsh", "zsh"…), de sa configuration actuelle, et de votre statut. Pour les exemples qui vont suivre, nous prendrons le symbole [$] pour représenter le prompt.

Le shell est un animal simple : c'est un programme qui vit sur votre Linux, et ne sert qu'à lancer d'autres applications. En fait, toute commande que vous ordonnez à votre shell d'exécuter est un binaire (un programme) stocké quelque part sur votre disque. Subsistent quelques rares exceptions que l'on appelle "built-ins", ou en Français de patakweque : "les fonctions internes"… En voici un aperçu :

- cd
- exit
- les fonctions de gestion de l'environnement (relatives à chaque shell)
- etc. (selon les shells).

Comme votre shell est un animal propre, il se charge de gérer ce que l'on appelle un "environnement". Cet environnement est une liste de variables dynamiques qui peuvent être modifiées par l'utilisateur, et qui servent à l'exécution du shell. Diverses informations comme l'allure de votre prompt, la liste des répertoires où chercher les binaires, le nom de votre machine, votre nom d'utilisateur, etc. sont stockées dans ces variables.

Chaque animal (heu… pardon), chaque shell offre des outils de gestion d'environnement différents. Nous ne les aborderons malheureusement pas cette fois, mais vous pouvez toutefois afficher le contenu de votre environnement avec la commande "env" bien souvent.

## Se promener dans les branches du système

Pour afficher le contenu d'un répertoire, tapez "ls" suivi ou non du chemin du répertoire que vous voulez visualiser :

```
$ ls /home
```

Notez que vous pouvez ajouter des options à chaque commande Unix. Par exemple pour "ls", on dispose, entre autres, des options suivantes : a, l, d, F, i, R. En aval, vous trouverez plus de détails quant aux procédures pour s'informer sur les options disponibles des différents outils du système. Avant cela, essayez par exemple la commande suivante :

```
$ ls –aR /home
```

Dans cet exemple, on applique l'option "–a", qui permet de voir les fichiers cachés, et "–R", une option de récursion permettant de lister les fichiers contenus dans les sous-répertoires. Nota bene : le répertoire racine (root) de votre système est "/".

La commande qui permet de naviguer dans l'arborescence est " cd " (Change Directory), suivie du chemin du répertoire. Ainsi, si vous voulez aller dans le répertoire /home, il vous suffit d'entrer :

```
$ cd /home
```

Il existe deux liens permanents et non effaçables qui vous permettent de vous déplacer plus facilement dans vos répertoires. Il s'agit de "." et de "..", désignant respectivement le répertoire courant dans lequel vous opérez, et le répertoire parent. Essayez donc :

```
$ cd .. && pwd
```

Taper "pwd" vous indique le chemin complet du répertoire dans lequel vous vous trouvez.

## Jouer à Dieu sur son système de fichiers

Pour créer un répertoire, tapez la commande "mkdir" suivie du (ou des) nom(s) de répertoires que vous voulez créer :

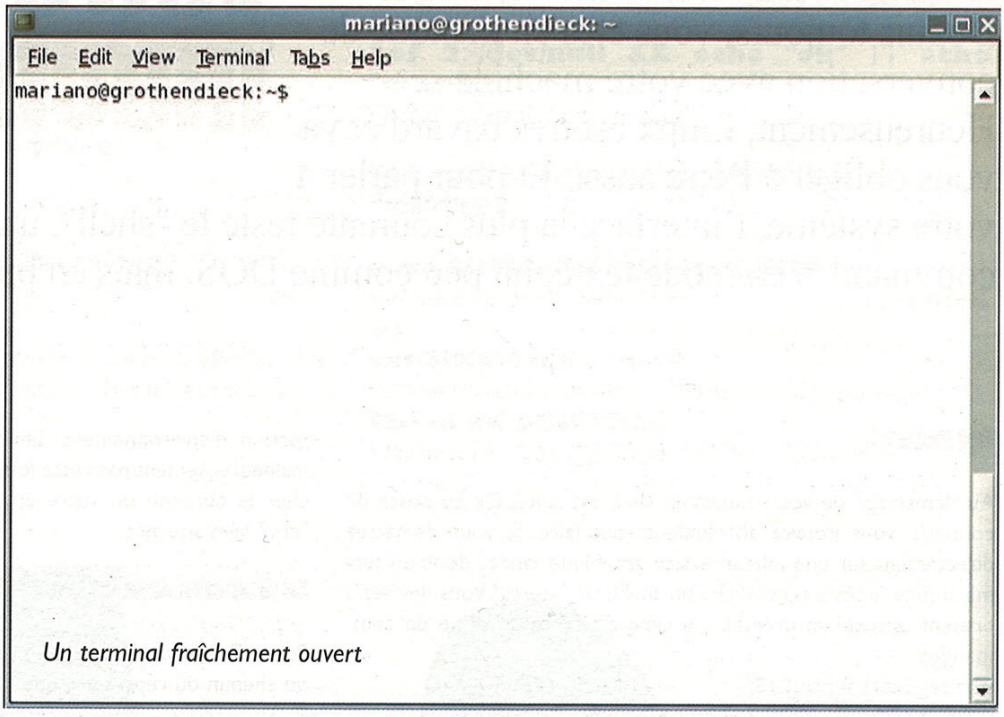
```
$ mkdir /home/
blagues ./annexes
```

De cet exemple, il découle que l'on peut créer des répertoires partout où on le souhaite si l'on en précise bien le chemin.

Effacer un répertoire (un répertoire est un fichier, rappelons-le) se fait avec " rmdir ", qui ne supprime un répertoire que s'il est vide. Pour effacer un répertoire non-vide, il faut utiliser la commande "rm –R" (Recursive ReMove). Cette dernière est toutefois plus spécifique aux fichiers standards.

Notez que "rm" ne vous demande aucune confirmation avant la suppression - irréversible ! - de vos fichiers, sauf avec l'option "-i". Cet état de fait a engendré de nombreux désastres un peu partout dans le monde des maladroits.

Sinon, vous pouvez déchaîner vos passions créatrices à l'aide de "touch". On l'utilise suivi du nom du fichier afin d'engendrer ce dernier. "touch" sert également à modifier les dates des derniers accès aux fichiers.




*Un terminal fraîchement ouvert*

On peut copier fichiers et répertoires à l'aide de la commande "cp" (CoPy), de la façon suivante :

```
$ cp /etc/passwd /tmp/password
```

La commande "mv" (MooVe) dont la syntaxe est à l'identique de "cp", permet quant à elle de déplacer (et accessoirement renommer) des fichiers et répertoires.

## Jongler avec son shell

Là où le shell devient réellement puissant, c'est dans les différentes façons qu'a l'utilisateur de lui faire exécuter ses commandes. Par exemple, il est possible de rediriger très simplement le résultat d'un programme (une sortie) vers l'entrée (ce que vous auriez saisi) d'un autre programme, ou bien d'utiliser la sortie d'un programme comme paramètre pour lancer un autre programme, de chaîner des exécutions, etc.

Les shells modernes comprennent même des langages de script qui leur sont propres. Cela dit, nous nous pencherons uniquement sur les fonctionnalités de redirection (d'une sortie vers n'importe où) offertes par tout bon shell.

Pour cela, voyons tout d'abord comment afficher le contenu d'un fichier à l'écran. A cet effet, on utilise généralement "cat" (conCATénation):

```
$ cat toto
Alors c'est
l'histoire
de Toto
...
```

Si le fichier est trop long et que vous ne pouvez pas en voir le début car la fenêtre de votre shell est trop petite, utilisez "more" (ou "less") ou une redirection, afin d'afficher le fichier page par page :

```
$ more compiling_tiger.txt
```

Pour rediriger un résultat, nous utiliserons les opérateurs suivants : [>], [>>], [<], [|] (" alt_gr+6 " sur votre clavier Français) et les backquotes [`] (" alt_gr+7 " sur ce misérable clavier toujours en français).

Ici, on se sert de [>]. Par exemple, on veut enregistrer le résultat de la commande " ls " dans le fichier listing :

```
$ ls > listing
```

Notez qu'il importe peu que le fichier ait déjà été créé ou pas : cette commande efface le contenu du fichier ou le crée, respectivement.

Si, par contre, on ne veut pas que le contenu du fichier (s'il existe) soit effacé, on utilise non pas > mais >>, ce qui écrira à la fin du fichier.

Un petit pipe ? Hum… Pour les esprits mals tournés, un pipe c'est ça : [|]. En fait, cela sert à rediriger la sortie d'une commande vers l'entrée d'une autre. Explications ? Vous voulez rechercher une chaîne de caractères (ici : "z66w") dans un fichier ("log_irc"), vous allez faire :

```
$ cat log_irc
  | grep 'z66w'
<zaizaide> z66w: c'est toi que je viens embê-
 ter :)
```

```
<z66w> erff
```

Ici l'outil " grep " sert à sélectionner les lignes qui ne contiennent que le texte indiqué en argument.
Autre exemple, vous voulez lire un fichier page par page mais en faisant un cat :

```
$ cat toto   |   less
```

### Les man pages

A tout moment, vous pouvez taper "man nom_de_la_commande" afin d'obtenir une aide sur la commande précisée. Les manuels (manpages) sont très complets, mais leur aspect un peu technique reboutera sûrement quelques rares manchots. Tapez "man man" pour commencer sur quelque chose de concret.
Vous pouvez également taper "nom_de_la_commande –-help" pour afficher une aide succincte. En général, la plupart des applications supportent cette option ou affichent une aide en cas d'erreur, ainsi, toutes les routes mènent à Rome. Il reste toujours bien pratique de comprendre la syntaxe de ces aides. Étudions une sortie fictive de message d'aide pour l'outil "ping" :

```
$ ping --help
ping [-LRUbdfnqrvVqQB]
 [-c count]
 [-h host]
 destination
```

Tout d'abord, les options entre crochets indiquent leur caractère optionnel. Les imbrications signifient que pour utiliser l'option la plus enracinée, vous devez aussi utiliser celle qui la contient. Ainsi, pour indiquer un host2 à cet outil, vous devrez au préalable spécifier "host1" avec l'opérateur "–h".
Dans la suite incohérente de lettres, chaque option est individuellement optionnelle : il s'agit de drapeaux d'état (on utilise une ou plusieurs options qui n'ont pas besoin de paramètres).
Enfin, les options écrites sans spécifications ou entre "< >" sont obligatoires.

### On achève !

A force de tâtonnements, d'essais manqués et de documentation lue à la fois dans les manpages et sur Google (vous ne vous en sortirez pas sans ça), un utilisateur débutant s'amusera bien vite plus sur son shell que dans un environnement graphique. A moins que ce dernier, dans un ultime effort de survie, ne serve plus qu'à exécuter des shells.

# Les caractères spéciaux

Ci-dessous la liste des caractères spéciaux du shell :

**&** processus en arrière-plan

**~** home directory

**;** séparateur de commandes

**** annulation d'un caractère spécial

**"** doubles quotes : encadre une chaîne de caractères et annule la signification de $, \ et '

**`** back quotes : substitution de commandes

**'** simples quotes : encadre une chaîne de caractères et annule la signification de tous les caractères spéciaux

**#** commentaire

**( )** exécution d'un shell fils

**[ ]** test

**|** pipe

**$** variable

***** remplace 0 ou n caractères

**!** négation d'un test

**?** remplace 1 caractère

**< >** redirections entrée, sortie

**$0...$9** variables de position
[modifier]

Les shells Unix disposent de petits "raccourcis" très astucieux et utiles, qui vous épargneront de taper sur quelques touches.

### Dernière ligne de commande : !!

Vous pouvez la désigner par '!!', ce qui peut être très intéressant.
[user@localhost user]$ vi
[user@localhost user]$ which !!
which vi
/bin/vi

### Arguments de la dernière commande : !*

Les arguments de la dernière commande peuvent être représentés par '!*'.
[user@localhost user]$ mkdir test
[user@localhost user]$ cd !
cd test

### Utiliser la sortie d'une commande comme argument

Vous pouvez réutiliser directement ce qu'une commande écrit à l'écran comme argument pour une autre commande. Pour ce faire, vous devez encadrer la commande par une cote inverse ` ou la mettre entre parenthèses précédées du signe $ ; elle sera remplacée par ce qu'elle écrit à l'écran dans la ligne de commande.
Imaginez par exemple que vous vouliez voir les informations sur le fichier exécutable de emacs.
[user@localhost user]$ ls -l `which emacs`
[user@localhost user]$ ls -l $(which emacs)

est ainsi équivalent à :
[user@localhost user]$ which emacs
/usr/bin/emacs
[user@localhost user]$ ls -l /usr/bin/emacs

Vous pouvez mixer les raccourcis vus précédemment :
[user@localhost user]$ emacs
[user@localhost user]$ ls -l `which !!`

### Remplacer un caractère par un autre : ^

Si vous souhaitez remplacer la première occurrence d'un caractère de la ligne de commande précédente par un autre, vous pouvez utiliser le symbole ^, comme ci-dessous :

[user@localhost user]$ lpcate i486-linux-libc5
lpcate : command not found
[user@localhost user]$ ^p^o
locate i486-linux-libc5

^p^o signifie : refait la même ligne de commande que précédemment, mais remplace le premier p par un o.

### Lancer un programme directement en tâche de fond : &

Il suffit de faire suivre la ligne de commande du symbole & :
[user@localhost user]$ cp -R /usr/doc /tmp &
[1] 7194
[user@localhost user]$ _

La commande est lancée en tâche de fond, c'est à dire qu'elle s'exécute, mais la main vous est rendue tout de suite. La fin de la commande est signifiée par un message :
[user@localhost user]$
[1]+ Done cp -R /usr/doc /tmp
[user@localhost user]$ _

### Lancer plusieurs programmes en même temps : &, &&, ||, ;

Vous avez plusieurs solutions :
prog1 ; prog2 lance prog1, puis prog2,
prog1 & prog2 lance prog1 en arrière-plan, puis immédiatement prog2 en avant-plan,
prog1 && prog2 lance prog1, puis prog2 seulement si prog1 n'a pas retourné d'erreur,
prog1 || prog2 lance prog1, puis prog2 seulement si prog1 A retourné une erreur.

# Sécuriser Linux

## I. Sécurisation de base

Même si la station Linux est réputée plus sécurisée que d'autres environnements de type Win32, il convient de procéder à quelques modifications pour que la sécurité de votre poste et de vos données soit optimum. Découvrez comment rendre encore plus " secure " le plus sûr des systèmes.

Nous allons voir les principes de base pour rendre son poste de travail GNU/Linux encore plus sécurisé. Notre démarche se basera sur le fait que vous possédez une distribution déjà installée et que celle-ci n'est pas compromise. En cas de doute, je ne saurais trop vous conseiller de faire une réinstallation complète de votre système, en ayant, bien sûr, auparavant sauvegardé vos données personnelles. Ceci étant dit, à vos claviers !

### Sécurité physique

Le BIOS (Basic Input/Output System). Une chose que vous pouvez faire (en dehors de Linux) pour rendre votre machine plus sûre est d'activer la protection par mot de passe de votre BIOS. En effet, quelque soit la politique de sécurité appliquée au niveau de votre système, il est possible à une personne physique de couper l'alimentation de votre machine et de redémarrer votre ordinateur à partir d'une disquette par exemple. Il est assez facile pour quelqu'un maîtrisant un minimum Linux de passer des paramètres de démarrage au noyau qui peuvent conduire à l'accès complet à vos données (voir LILO). En revanche, si vous avez mis en place une protection au niveau du BIOS, le redémarrage de la machine sera impossible sans votre mot de passe.

Cette démarche constitue donc la première étape pour la sécurisation de votre système. La première chose est de s'assurer qu'aucune autre personne ne pourra changer les paramètres du BIOS et que le mot de passe soit obligatoire au démarrage de la machine. Pour activer ces protections, il suffit simplement de se rendre dans les paramètres de configurations de votre BIOS. Il n'y a pas à proprement parler une technique qui permet d'y arriver tant le nombre de configuration est diversifié. Pour y accéder, il suffit d'appuyer sur la touche correspondante au message d'accueil inscrit au démarrage de votre machine :

```
DEL to enter SETUP,
SUPPR to enter SETUP,
INS to enter SETUP,
F1 to enter SETUP,
ALT+F1 for SETUP, ALT+F2 for FLASH,
```

jusqu'à l'apparition du menu de configuration de votre BIOS. La configuration des mots de passe pour le changement des données du BIOS ou le démarrage se font généralement dans les sous-menus : BIOS Setup, Security Setup ou General Setup. Une fois les mots de passe entrés, sauvegardez votre BIOS (touche F10) et rédémarrez. La protection par mot de passe est activée. Attention de ne pas les oublier ! Sans eux, vous n'aurez plus la possibilité de démarrer votre box ;).

### Processus d'amorçage

LILO (LInux LOader). LILO est un programme qui se charge de lancer le noyau Linux lorsque le BIOS a passé la main au système d'exploitation de la machine. Il est installé par défaut avec toutes les distributions récentes. Si vous souhaitez vérifier l'existence de LILO sur votre machine, maintenez la touche Majuscule enfoncée

lors du boot sur votre disque dur au démarrage de la machine. Si vous voyez apparaître cette invite : LILO boot: alors c'est que vous êtes potentiellement vulnérable au type d'attaque que nous décrivons ci-dessous. Dans le cas contraire, ne croyez pas que cela soit impossible et lisez tout de même ce qui suit.
Une des fonctionnalités offerte par LILO est la possibilité de passer des arguments au noyau Linux lors du démarrage de celui-ci. Cette option peut se révéler particulièrement utile si vous avez besoin de procéder à des opérations de maintenance ou s'il est impossible d'amorcer le système de manière normale (suite à des coupures de courant ou des pertes de données par exemple). Pour démarrer Linux en mode maintenance, il suffit d'indiquer l'argument " single " à l'invite boot :

```
LILO boot: linux single
```

Cet argument charge le noyau linux dans un mode d'administration, accessible uniquement au root. Le danger ici est que le fonctionnement normal du système, tel que vous l'aviez défini, n'est plus valable et qu'à la place il reste simplement une invite login root. Mais plus terrible encore est l'argument " init ". Celui-ci permet de charger un système linux sans aucune de ses fonctionnalités, services et paramètres. C'est-à-dire que n'importe qui passant cette argument se voit attribué un shell root sur la machine sans qu'aucun mot de passe ne lui ait été demandé :

```
LILO boot: linux init=/bin/bash
```

Il paraît donc évident que des mesures pour sécuriser LILO doivent être prises rapidement ! Je ne rentrerai pas dans une explication détaillée du fichier de configuration /etc/lilo.conf car ce n'est pas l'objectif de cet article. Je vous donne simplement les lignes à ajouter à celui-ci pour le rendre un peu plus sécurisé. Pour éditer et modifier ce fichier, vous devez être root (la commande su fait très bien l'affaire) :

```
[user@bdm ~]$ su
Password:
[root@bdm ~]# vi /etc/lilo.conf
```

Pour s'assurer que personne ne pourra passer d'arguments au noyau sans mot de passe, il suffit d'ajouter les deux lignes en gras dans le fichier de configuration :

```
# exemple de fichier de configuration
/etc/lilo.conf
boot=/dev/hda
map=/boot/map
default=linux
prompt
timeout=200
password="insérez_votre_mot_de_passe_ici"
restricted
image=/boot/vmlinuz
        label=linux
        root=/dev/hdb1
        initrd=/boot/initrd.img
        append="quiet devfs=mount hdd=ide-scsi"
        vga=normal
        read-only
other=/dev/hda1
        label=NT
        table=/dev/hda
```

Cette configuration permettra à toutes les images présentes dans le fichier d'être amorçables sans mot de passe à moins que des arguments n'aient été fournis au noyau. Si un utilisateur souhaitait passer des arguments au noyau, celui-ci devrait d'abord saisir le mot de passe associé. Pour enregistrer vos modifications et que celles-ci soient actives au prochain redémarrage, vous devez taper à l'invite de commande (toujours en root) : lilo. La nouvelle configuration est alors sauvegardée. Pour finir avec LILO, il convient de changer les permissions d'accès en lecture sur le fichier. Et oui, sinon tout le monde peut éditer celui-ci et y lire en clair votre mot de passe et passer à nouveau des paramètres au noyau. Pour cela, il suffit d'interdir la lecture ou l'écriture du fichier au non root :

```
chmod 600 /etc/lilo.conf
```

LILO est maintenant un peu plus sécurisé.

## Le fichier /etc/inittab.

Pour mémoire, inittab est le fichier de configuration responsable des services et fonctionnalités chargés lors du démarrage de Linux. Il existe deux modifications simples qui peuvent être apportées au fichier /etc/inittab pour rendre les paramètres par défaut d'init un peu moins vulnérables. La première est la modification du niveau d'exécution par défaut du système est la seconde et la suppression de la combinaison Ctrl+Alt+Suppr pour réinitialiser le système.
Typiquement, le niveau d'exécution par défaut pour les distribution disposant d'une interface graphique d'installation (Mandrake, Red Hat...) est de 5. Ce qui signifie que le système démarre automatiquement l'interface graphique au démarrage de la machine et vous propose une invite de login X. Cette fonctionnalité pose des problème de sécurité (voir X11). Il est conseillé d'utiliser un niveau d'exécution par défaut en mode console de niveau 3. Cela ne vous empêchera pas de lancer ultérieurement X en tapant à l'invite de commande : startx. Pour changer le niveau d'exécution, éditez votre fichier /etc/inittab en root. Trouvez la ligne correspondante à celle ci-dessous (normalement la première hors commentaires) et remplacez le 5 par 3 :

```
# Default runlevel.
# id:5:initdefault:
id:3:initdefault:
```

Pour désactiver la combinaison des trois touches, il suffit de mettre la ligne contenant " crtlaltdel " en commentaire en plaçant un # au début de la ligne comme dans l'exemple ci-dessous :

```
# Trap CTRL-ALT-DELETE
# ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

## Le fichier /etc/securetty

Une étape importante et souvent oubliée dans la sécurisation des postes de travail est le fichier /etc/securetty qui énumère les tty sur lesquels le login root peut arriver. Si vous utilisez Telnet par exemple pour l'administration de votre poste, il se peut que cela pose des problèmes. Il est conseillé de mettre en commentaire toutes les entrées de ce fichier et n'y laisser que un ou deux périphériques vc. Le format du fichier est simple et ne devrait pas vous poser de problème :

```
#tty1
#tty2
#tty3
#tty4
#tty5
#tty6
vc/1
vc/2
```

Ainsi toute les tentatives de connexions root distantes sur les tty seront rejettées. Ouf ;-). Maintenant que tout semble dans l'ordre, attaquons-nous au coeur de Linux : le système et les utilisateurs.

## Le système et les utilisateurs

Dans ce chapitre, je ne m'attarderai pas à expliquer le fonctionnement de init, des services et leurs configuration. Je vais simplement vous indiquer comment vérifier les processus qui sont lancés au démarrage de votre machine et comment les stopper. Nous verrons ensuite comment appliquer une politique de gestion des mots de passe des utilisateurs.

## Trouver et arrêter les services inutiles

Si vous avez bien suivi ce qui a été dit précédement, alors le niveau d'exécution par défaut de votre Linux devrait être de 3. C'est-à-dire en mode console, multi-utilisateur et avec le réseau activé. Pour neutraliser un service inutile, rien de plus simple. Nous allons déplacer son lien symbolique dans un répertoire temporaire. Généralement, les liens vers les scripts d'initialisation des services se trouvent dans le répertoire /etc/rc.d/. Par exemple, si vous souhaitez arrêter le serveur de police xfs du niveau d'exécution 5, il suffit de retirer le lien symbolique du répertoire qui y est lié :

```
[root@bdm ~]# cd /etc/rc.d/rc5.d
(se rendre au répertoire des scripts de niveau
5)
[root@bdm rc5.d]# mkdir temporaire
(créer un dossier nommé temporaire)
[root@bdm rc5.d]# mv S90xfs temporaire
(déplacer le script S90xfs dans le répertoire
temporaire)
```

Le nom des fichiers est à adapter à votre système, mais le principe est le même. Dès lors, que vous relancerez Linux, celui-ci ne chargera pas le service pour le niveau d'exécution choisi. Pour réactiver le service, redéplacer le script vers son répertoire original :

```
[root@bdm ~]# cd /etc/rc.d/rc5.d/temporaire
[root@bdm temporaire]# mv S90xfs ../
```

Connaître tous les services utiles sur sa machine n'est pas une mince affaire et dans ce domaine pas de magie. Il faut tester et retester, encore et encore, jusqu'à avoir une configuration qui vous convienne avec les services souhaités.

## Shadow est-il activé ?

Une bonne chose à faire est de vérifier que les mots de passe shadow /etc/shadow sont bien installés sur votre système. Normalement, toutes les distribution récentes l'incluent automatiquement. En effet, sans cette précaution, n'importe qui serait à même de récupérer votre fichier de mot de passe /etc/passwd et d'utiliser un cracker brute force (du type : John The Ripper), afin de récupérer vos logins et compte root. Shadow est un fichier lisible uniquement par root qui contient votre password de manière cryptée. Pour vérifier la présence de ce fichier sur votre Linux, tapez à l'invite de commande sous votre compte utilisateur (non root) :

```
[user@bdm ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Si vous obtenez un autre message que celui-ci ou qu'il n'y a pas de fichier, alors c'est que shadow n'est installé sur votre système ou pas correctement. Pour remédier à ce défaut de sécurité, installer le package shadow depuis le cd de votre distribution ou de son site FTP.

## Politique de gestion des mots de passe utilisateurs

Si vous lisez The Hackademy Journal, vous aurez peut-être remarqué l'article sur " Comment choisir un bon password " paru dans le numéro THJ 02. Une bonne solution pour la sécurité est d'imposer aux utilisateurs un changement de mot de passe périodique afin de couper l'herbe sous le pied aux crackers (casser un mot de passe crypté prend du temps, donc si les changements ont lieu avant que le cracker ait eu le temps de casser le précédent, tout le travail fourni par celui-ci sera inutile). Pour automatiser cette tâche, il existe une commande très utile : chage. Pour en savoir plus : man chage. Nous vous donnons ci-dessous un exemple d'automatisation :

```
chage -m 0 -M 60 -d 0 -I 0 -E 0 -W 3 brotha
```

Avec cette commande, l'utilisateur brotha devra changer son mot de passe à la prochaine connexion sur le système. Celui-ci devra le changer également tous les 60 jours et sera avertit de l'expiration 3 jours avant que le changement soit obligatoire.

## Sécurisez votre accès au réseau !

Dans le cadre de la sécurité d'un poste de travail GNU/Linux connecté à un réseau local ou Internet, c'est l'étape obligatoire. Vous possédez un connexion haut débit par ADSL ou câble ? Imaginez que votre machine est une voiture ! Si vous ne prennez pas les précautions nécessaires, c'est un peu comme si vous la laissiez bien garée devant chez vous tous les soirs, portières ouvertes et clés sur le contact... A bon entendeur.

## Le demon Internet inetd, votre pire cauchemar

Dans un grand nombre de distribution (ex : Mandrake 9.x), le service inetd a été remplacé par xinetd. Néanmoins, certaines l'utilisent encore et ce point ne devrait pas être évité (ex : Debian). Sur les machines de type Unix, la plupart des services réseaux sont associés à un demon spécifique qui a pour mission de prendre en charge les requêtes de connexions distantes. A chaque service son demon associé. Par exemple, les requêtes Telnet seront prises en charges par in.telnetd, les requêtes FTP par in.ftpd, etc.

Dès lors il convient de s'assurer que ces demandes de connexions soient bien gérées par le système. Le fichier de configuration de inetd : /etc/inetd.conf est assez simple à comprendre. Il sera lu par le demon à chaque fois que celui-ci est lancé ou redémarré. Pour désactiver un service, il suffit simplement de mettre un # au début de la ligne pour la passer en commentaire. Dans l'exemple ci-dessous, nous avons gardé la possibilité d'accepter les connexions FTP, en refusant les autres :

```
ftp     stream  tcp  nowait  root  /usr/sbin/tcpd
in.ftpd -l -a
#telnet  stream  tcp  nowait  root  /usr/sbin/tcpd
in.telnetd
#shell   stream  tcp  nowait  root
/usr/sbin/tcpd in.rshd
#talk    stream  tcp  nowait  root
/usr/sbin/tcpd in.talkd
```

Comme vous pourrez le constater votre fichier devrait être plus important que celui-ci. En cas de doute sur tel ou tel service, il est conseillé de placer la ligne en commentaire. Pour réactiver le service, il suffira d'enlever le # et de redémarrer le demon. Pour activer ces modifications sans relancer votre machine : killall -HUP inetd

## Comment modifier les ports par défaut des services ?

Nous n'allons pas revoir ici les principes de bases du fonctionnement du réseau. Sachez simplement pour mémoire qu'il existe pour chaque service réseau un port qui lui est associé. Il existe sur Linux un fichier de configuration qui se charge de faire cette relation : /etc/services. Voici un extrait de ce fichier :

```
ftp-data    20/tcp
ftp         21/udp
ssh         22/tcp
ssh         22/udp
telnet      23/tcp
smtp        25/tcp
time        37/tcp
```

Pour changer le port par défaut en écoute (port par défaut qui attend la connexion), il suffit de modifier l'information directement dans le fichier /etc/services. Par exemple, si on souhaite activer le service telnet pour une administration distante sur le port 9356 :

```
ftp-data    20/tcp
ftp         21/udp
ssh         22/tcp
ssh         22/udp
telnet      9356/tcp
smtp        25/tcp
time        37/tcp
```

C'est tout. Veillez à spécifier des numéros de port qui soient relativement élevé en gardant à l'esprit que les 1024 premiers sont réservés et que le dernier est 65535. Désormais pour se connecter par telnet sur la machine, il faudra préciser explicitement le port à contacter, soit :

```
telnet nom_ou_ip_de_la_machine 9356
```

## L'arme fatale : TCP wrappers

TCP wappers est inclus depuis pas mal de temps dans les distributions Linux et fournit un niveau de sécurité essentiel pour les services gérés par inetd et pour le serveur ssh. Comme vous avez pu le voir dans l'exemple de fichier inetd ci-dessus, le demon appelé pour la plupart des services n'est pas le demon correspondant, mais /usr.sbin/tcpd le_nom_du_demon. Cela signifie qu'avant d'autoriser le demon à prendre en charge la connexion, celle-ci sera filtrée par le demon de TCP wappers : tcpd. Il existe principalement deux fichiers de configuration qui gèrent TCP wappers : /etc/hosts.deny qui se charge du refus de connexion et /etc/hosts.allow qui les autorise. Si vous souhaitez mettre en place une sécurité forte sur votre machine et que vous n'avez pas besoin de fournir des services à des machines distantes, alors il existe un moyen simple de refuser toutes les demandes de connexions. En root, éditez le fichier /etc/hosts.deny et placez l'unique ligne suivante :

```
ALL : ALL
```

Sachez aussi que le fichier /etc/hosts.allow ne devrait rien contenir. Les règles fixées dans celui-ci étant prioritaire à celles de /etc/hosts.deny. Pour vérifier la validité des règles mises en place, il existe une commande simple : tcpdchk.

## Une bonne idée : le firewalling avec ipchains/iptables

En matière de firewalling et de filtrage de paquets, ipchains (noyau 2.2) et aujourd'hui iptables (noyau 2.4) fournissent un moyen efficace pour mettre en place une sécurité forte sur un réseau ou sur un hôte. Néanmoins, pour des personnes ne maî-

trisant pas parfaitement le fonctionnement de TCP/IP et de ses différents protocoles, ils peuvent sembler un peu durs à aborder. Pour vous en convaincre, je vous laisser consulter la page man : man iptables. Heureusement pour nous, le monde est bien fait et dans les domaines de l'open source rien n'est impossible. Je vous propose donc d'installer GuardDog (1) qui se chargera à votre place d'éditer et d'enregistrer vos règles pour ipchains/iptables suivant la version de votre kernel. Sachez tout de même que celui-ci ne fonctionne qu'avec KDE 2 et un noyau 2.2 mais que KDE 3 et un noyau 2.4 sont vivement recommandés. La configuration est très simple et ne devrait pas vous poser de difficultés particulières.
Une fois installé, vous devez lancer le programme en root et effectuer la première configuration. Par défaut, la zone Internet (c'est-à-dire le monde extérieur) et la zone locale (votre machine) sont déjà configurées et l'ensemble des règles en font un système complètement fermé. C'est donc à vous d'ouvrir les portes de votre machine au monde extérieur. Il vous suffit de sélectionner les services que vous souhaitez laisser entrer sur votre machine, comme HTTP, HTTPS, FTP ou POP3 et sélectionner les services qui doivent se connecter vers l'Internet comme SMTP par exemple.

## X Window, votre meilleur ennemi

Une fois que vous aurez mis en place toutes ces modifications, normalement tous les ports de votre machine devraient être fermés sauf peut-être un : le port 6000. Celui-ci correspond à X Window qui fonctionne comme un serveur et permet les connexions distantes. Pour vérifier ce point, vous pouvez lancer nmap (2) sur votre machine et tester les ports ouverts :

```
[brotha@brotha etc]$ nmap -p 1-65535 localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on bdm (192.168.1.2):
(The 65530 ports scanned but not shown below are
in state: closed)
Port       State       Service
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up)
scanned in 4 seconds
```

Il apparaît donc clairement que X attend les demandes de connexion sur le port 6000. Il faut y remédier, surtout qu'il existe des exploits public de déni de sevice sur celui-ci. Pour le fermer définitivement, il suffit d'éditer en root le fichier /usr/X11R6/bin/startx et d'ajouter l'argument -nolisten TCP dans la ligne serverargs :

```
serverargs="-nolisten TCP"
```

## Conclusion

Toutes les techniques mentionnées dans cette partie sont assez connues et s'adressent en priorité aux personnes ayant installé une distribution Linux dans le but de tester et découvrir le monde du libre, pour des stations de travail ne fournissant aucun service sur le réseau. Ce qui représente la majorité des cas pour des postes bureautiques et/ou multimédia connectés à Internet chez les particuliers. Ces mesures ne représentent en aucun cas une solution de sécurisation complète, mais peuvent être le point de départ pour une utilisation un peu plus secure de sa machine, en évitant de la laisser ouverte (trop facilement) aux nombreux scripts-kiddies qui peuplent aujourd'hui le réseau.

## Ressources

Sécurité sous Linux – Editions CampusPress
(1) GuardDog : http://www.simonzone.com/software/guarddog/
(2) Nmap : http://www.insecure.org/nmap/

# Offre spéciale d'abonnement

# LINUXSCHOOL Magazine

## LE NOUVEAU MAGAZINE 100% LINUX

**1 an**

**de pur linux (6numéros)**

**24 euros !!!**

A découper ou à recopier et à envoyer accompagné de votre réglement de 24 euros à l'ordre de LPN à LINUXSCHOOL Magazine15 rue Chevreul - 94700 MAISONS-ALFORT

NOM : _____ PRENOM : _____

ADRESSE : _____

CODE POSTAL : _____ VILLE : _____

# L'intérêt est toujours conservateur

**Errico Malatesta**
***Militant anarchiste italien***

**Les codes-barres vous enchaînent**