# Sommaire

# Installation d'une seedbox.

### Le concept:

- Un système: debian etch 4.0 & debian etch 5.0
- Un serveur web: lighttpd
- Un serveur sql: Mysql
- Un client: transmission
- Une interface web + client bittorrent: torrentflux-b4rt
- Pour rapatrier tes fichier du dédié vers ton pc on fera ça en SFTP avec filezilla.
- Un système sécurisé: un firewall, et un système de bannissement d'ip pour les petit malin qui tentent de se connecter à ton serveur via la méthode bruteforce (bannissement au bout de trois tentatives ratés).
- Un système qui se met à jour tout seul avec cron-apt.

### Les étapes

- Finalisation/Sécurisation/Optimisation de l'installation
- Configuration du firewall
- Installation de Lighttpd/Mysql/Xcache
- Installation de Transmission/torrentflux-b4rt

Copier/coller les commandes dans un terminal. Dans les cas ou la commande fait plusieurs lignes, copier TOUTES les lignes en même temps.

# Finalisation/Sécurisation/Optimisation de l'installation

### *Finalisation :*

Pour commencer tu te connecte en root à ton serveur.

```
ssh login@ip
```

On commence par désactiver la source CDROM(au cas ou si ce n'est déjà fait)

```
sed -i -e 's/^\(deb cdrom\)/#\1/' /etc/apt/sources.list
```

Modification de la configuration des locales:

```
echo "fr_FR ISO-8859-1
fr_FR.UTF-8 UTF-8
fr_FR.UTF-8@euro UTF-8
```

```
fr_FR@euro ISO-8859-15" > /etc/locale.gen
/bin/sed -i -e 's/^LANG=.*/LANG=fr_FR.UTF-8/' /etc/default/locale
/bin/sed -i -e 's/^LANG=.*/LANG=fr_FR/' /etc/environment
/usr/sbin/locale-gen
```

Configurations des dépôts contrib et non-free de Debian dans APT:

```
/bin/echo "# Debian Etch contrib
deb ftp://mir1.ovh.net/debian/ etch contrib
deb-src ftp://mir1.ovh.net/debian/ etch contrib

deb http://security.debian.org/ etch/updates contrib
deb-src http://security.debian.org/ etch/updates contrib" \
    | /usr/bin/tee /etc/apt/sources.list.d/etch-contrib.list
/bin/echo "# Debian Etch non-free
deb ftp://mir1.ovh.net/debian/ etch non-free
deb-src ftp://mir1.ovh.net/debian/ etch non-free

deb http://security.debian.org/ etch/updates non-free
deb-src http://security.debian.org/ etch/updates non-free" \
    | /usr/bin/tee /etc/apt/sources.list.d/etch-non-free.list
```

Une fois ceci fait, mettez à jour votre système d'exploitation:

```
/usr/bin/apt-get update
/usr/bin/apt-get upgrade
```

Nous activons la colorisation de la commande ls:

```
/bin/cp /etc/skel/.bashrc $HOME
/bin/sed -i -e 's/^# \(.*\(LS_OPTIONS\|dircolors\).*\)$/\1/' $HOME/.bashrc
```

Ensuite tu installes le support du temps internet pour garder ton système à l'heure:

```
apt-get install ntp
```

## *Sécurisation*

### Fail2ban

Tu vas installer Fail2ban, les logiciel qui s'occupe de bannir les IP qui tentent de se connecter par Brutforce

```
apt-get install fail2ban
```

Nous allons maintenant activer certaines configurations supplémentaires afin d'augmenter sa portée. En premier lieu, nous activons la protection contre les attaques en déni de service sur le SSH :

```
/bin/sed -i -e '/\[ssh-ddos\]/, /filter/ {0,/^enabled.*/ s//enabled = true/ }'
/etc/fail2ban/jail.conf
```

Nous activons aussi la protection du système d'authentification PAM (vu que ce mécanisme est présent un peu partout dans un système UNIX, c'est quelquechose de très pratique XD):

```
/bin/sed -i -e '/\[pam-generic\]/, /filter/ {0,/^enabled.*/ s//enabled =
true/ }' /etc/fail2ban/jail.conf
```

Enfin, nous redémarrons fail2ban pour prendre en compte les nouvelles configurations:

```
/etc/init.d/fail2ban restart
```

## Accès SSH

Pour améliorer la sécurité de ton serveur tu dois créer un compte utilisateur normale avec lequel tu te connecteras à celui-ci via PuttY (SSH)
Pour passer root afin de faire les opérations nécessaires, tu utiliseras la commande:

```
su
```

Avec cette commande tu rentre ton mot de passe root, et tu te retrouve... en root!
Tu peux donc travailler tranquillement, et on vas par la suite interdire la connection via ssh directement en root, pour des raisons de sécurité.

## Création de ton compte utilisateur

On installe un outil de génération de mot de passes:

```
/usr/bin/apt-get install apg
```

On vas créer un mot de passe sécurisé pour ton compte utilisateur avec la commande suivante:

```
/usr/bin/apg -q -a  0 -n 1 -M NCL
```

Bien noter ce mot de passe, c'est avec ce dernier que vous aller vous connecter à ssh une fois tout fini.

Nous créons un compte utilisateur sans privilèges (remplace "myuser" par le login de votre choix):

```
MY_USER='Nom_user_'
/usr/sbin/adduser $MY_USER
```

« myuser » est le login que vous utliserez pour vous connecter à SSH
Configure le serveur SSH de façon à ce qu'il n'accepte pas les connexions avec l'utilisateur root.
Utilise pour ce faire les lignes de commande suivantes:

```
/bin/sed -i -e 's/PermitRootLogin.*/PermitRootLogin no/' /etc/ssh/sshd_config
/etc/init.d/ssh restart
```

A partir de maintenant, tu te connecte à ton serveur avec putty en utilisant ton compte utilisateur, pour passer root et continuer la procédure tu tapes "su" comme indiqué plus haut.

## Recherche de Root Kits

Installation des logiciels de vérification de présence de root kits :

```
/usr/bin/apt-get install chkrootkit rkhunter libmd5-perl
```

Configuration de rkhunter en fonction du système:

```
SSH_ROOT_ALLOWED=no
TEST_ROOT_ALLOWED=$(/bin/grep -i "PermitRootLogin.*yes" /etc/ssh/sshd_config)
if [ -n "$TEST_ROOT_ALLOWED" ]; then
  SSH_ROOT_ALLOWED=yes
fi
/bin/sed -i -e 's|^[#]*\(ALLOWHIDDENDIR=/dev/.udev\)$|\1|' \
            -e 's|^[#]*\(ALLOWHIDDENDIR=/dev/.static\)$|\1|' \
            -e 's|^[#]*\(ALLOWHIDDENDIR=/dev/.initramfs\)$|\1|' \
            -e "s|^[#]*\\(ALLOW_SSH_ROOT_USER=\\).*$|\\1${SSH_ROOT_ALLOWED}|" \
    /etc/rkhunter.conf
```

La version de RkHunter présente dans Debian 5.0 Lenny permet de maintenir une base de signature des fichiers importants basée sur les informations fournies par le gestionnaire de paquets Debian. Cette fonctionnalité est importante car elle permet de détecter plus rapidement les modifications du contenu de certains logiciels (ps, ls, etc...). Pour activer cette fonctionnalité, utilisez la ligne de commande suivante :

```
/bin/sed -i -e 's|^[#]*\(HASH_FUNC=\).*$|\1md5sum|' \
            -e 's|^[#]*\(PKGMGR=\).*$|\1DPKG|' \
    /etc/rkhunter.conf
```

Puis tu mets à jour la base des menaces de RkHunter pour la première fois (par la suite elle est mise à jour chaque semaine):

```
/usr/bin/rkhunter --update
```

Si vous avez activé la fonctionnalité de vérification de la signature des fichiers, utilisez cette commande pour mettre à jour la base des signatures :

```
/usr/bin/rkhunter --propupdate
```

Afin de ne pas être forcé d'exécuter cette commande après chaque utilisation d'apt-get, faites en sorte qu'elle soit exécutée automatiquement :

```
/bin/echo '// Update rkhunter file signatures databases after running dpkg.
DPkg::Post-Invoke {
  "if [ -x /usr/bin/rkhunter ]; then if [ $(/usr/bin/rkhunter --help | /bin/grep
"propupd" | /usr/bin/wc -l) -gt 0 ]; then /usr/bin/rkhunter --propupd; fi; fi";
};' | /usr/bin/tee /etc/apt/apt.conf.d/90rkhunter
```

Remarque: Pour connaître les informations que te renverra RkHunter chaque jour, tu pourras utiliser la commande:

```
/usr/bin/rkhunter --configfile /etc/rkhunter.conf --report-warnings-only --
checkall
```

Mais bon, le but est qu'après tout ça, tu n'aies plus à toucher à putty

Tu fais en sorte que chkrootkit s'exécute tous les jours:

```
/bin/sed -i -e 's/RUN_DAILY=.*/RUN_DAILY="true"/' /etc/chkrootkit.conf
```

## Redirection des emails du compte root

Un système Unix peut être assez bavard, et a tendance a envoyer tous les emails importants au compte Root. Il est très important de suivre ces emails. Pour ce faire, vous pouvez utiliser un lecteur d'e-mail en ligne de commande.... ou alors, rediriger les emails destinés au compte root de la machine vers votre email habituel.

Suivant la configuration du serveur SMTP de votre fournisseur de compte e-mail, les emails envoyés par votre machine peuvent être rejetés. Personnellement, je n'ai pas ces problèmes avec les comptes GMail.

En premier lieu, renseignez l'email que vous souhaitez utiliser :

```
ROOT_EMAIL=my-account@gmail.com
```

Et configurez la redirection des emails du compte Root vers cet email :

```
/bin/sed -i -e "s/^\\(root:\\).*\$/\\1 ${ROOT_EMAIL}/" \
         /etc/aliases
```

## Surveillance des logs

Tu vas installer logwatch pour surveiller ton système. Le rapport fourni par ce logiciel te renseigne sur les tentatives d'intrusions et les éventuels problèmes rencontrés par le système.

```
/usr/bin/apt-get install logwatch libdate-manip-perl
```

## Vérification des mises à jour

Installe cron-apt de façon à ce qu'il envoie un mail au root lorsque des mises à jour sont disponibles :

```
/usr/bin/apt-get -y install cron-apt
/bin/sed -i -e 's/^#[ \t]*\(MAILTO="root"\)/\1/' \
       -e '/^#[ \t]*\(MAILON="error"\)/a\
MAILON="upgrade"' \
    /etc/cron-apt/config
```

## Détection des intrusions

Nous installons snort pour surveiller les tentatives d'intrusion sur notre système. Vous recevrez alors un résumé quotidien des alertes de sécurité.

```
/usr/bin/apt-get install snort
```

## Protection contre les recherches de ports ouverts (port scanning)

PortSentry permet de se protéger contre les scanners de ports. En premier lieu, il vous l'installer :

```
DEBIAN_FRONTEND='noninteractive' apt-get install portsentry iptables
```

Une fois ceci fait, il est nécessaire de configurer proprement portsentry avant de l'activer. La première chose à faire est de faire en sorte que PortSentry ignore votre adresse IP. Cela vous évitera d'être banni de votre propre serveur. En premier lieu, il vous faut entrer votre adresse IP (fixe de préférence):
Si vous êtes identifé en SSH avec le compte root, il vous suffit d'utiliser la commande suivante :

```
PROTECTED_IP=$(/usr/bin/who --ips | /bin/grep root | /usr/bin/cut --
characters=40-)
/bin/echo "Votre adresse IP est : ${PROTECTED_IP}."
```

Dans tous les autres cas, rentrez votre adresse IP manuellement :

```
PROTECTED_IP=xx.xx.xx.xx
```

Vous pouvez maintenant ajouter l'adresse IP à la liste des adresses IP ignorées par PortSentry :

```
/bin/echo "
# Ignoring root account owner IP:
${PROTECTED_IP}" \
    | /usr/bin/tee -a /etc/portsentry/portsentry.ignore.static
```

A présent, activez le bloquage des scans de ports TCP et UDP :

```
/bin/sed -i -e 's/^BLOCK_UDP=.*/BLOCK_UDP="1"/' \
           -e 's/^BLOCK_TCP=.*/BLOCK_TCP="1"/' \
         /etc/portsentry/portsentry.conf
```

Configurez PortSentry pour utiliser "iptables" plutôt que "route" pour bloquer les attaques :

```
/bin/sed -i -e 's/^KILL_ROUTE=.*$/#\0/' \
           -e '0,/^[\t #]*\(KILL_ROUTE=.*iptables[^\&]*\)$/s//\1/' \
         /etc/portsentry/portsentry.conf
```

Activez le mode de détection avancé de PortSentry :

```
/bin/sed -i -e 's/^TCP_MODE.*$/TCP_MODE="atcp"/' \
           -e 's/^UDP_MODE.*$/UDP_MODE="audp"/' \
         /etc/default/portsentry
```

Enfin, redémarrez PortSentry :

```
/etc/init.d/portsentry restart
```

## Configuration du firewall

Configuration du Firewall (iptables)

Bon, la je vais pas me prendre la tête non plus,

on va libérer les port 30000-35000 pour transmission, le port 22 pour ssh, le port 80 pour le serveur web, et on vas bloquer quelque organismes de surveillance.
Voici un script qui fait tout ça.

Tu copies/colles ce texte dans un fichier nommé /etc/init.d/firewall

```
nano /etc/init.d/firewall
```

Tu te retrouves dans l'éditeur de texte nano.

Tu copies/colles L'ensemble des règles, puis tu fais
(commandes clavier) [Control]+[o], [entrer], [Control]+[x]

pour sauvegarder le tout, et te retrouver en ligne de commande.

```
#!/bin/bash
echo Setting firewall rules...
#
#
#

###### Debut Initialisation ######

# Interdire toute connexion entrante
iptables -t filter -P INPUT
iptables -t filter -P FORWARD
echo - Interdire toute connexion entrante : [OK]

# Interdire toute connexion sortante
iptables -t filter -P OUTPUT
echo - Interdire toute connexion sortante : [OK]

# Vider les tables actuelles
iptables -t filter -F
iptables -t filter -X
echo - Vidage : [OK]

# Autoriser SSH
iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
echo - Autoriser SSH : [OK]

# Ne pas casser les connexions etablies
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
echo - Ne pas casser les connexions établies : [OK]

###### Fin Inialisation ######

##### Debut Regles ######

# Autoriser les requetes DNS, FTP, HTTP, NTP
iptables -t filter -A OUTPUT -p tcp --dport 21 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -p udp --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -p udp --dport 123 -j ACCEPT
echo - Autoriser les requetes DNS, FTP, HTTP, NTP : [OK]

# Autoriser loopback
iptables -t filter -A INPUT -i lo -j ACCEPT
iptables -t filter -A OUTPUT -o lo -j ACCEPT
echo - Autoriser loopback : [OK]

# Autoriser ping
iptables -t filter -A INPUT -p icmp -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -j ACCEPT
echo - Autoriser ping : [OK]

# HTTP
iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
#iptables -t filter -A INPUT -p tcp --dport 443 -j ACCEPT
#iptables -t filter -A INPUT -p tcp --dport 8443 -j ACCEPT
echo - Autoriser serveur Lighttpd : [OK]

# Torrentflux
iptables -A OUTPUT -o eth0 -p tcp --sport 30000:35000 -m state --state ! INVALID
-j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 30000:35000 -mstate --state ! INVALID -
j ACCEPT
echo - Autoriser torrentflux : [OK]

# VLC (streaming)
iptables -t filter -A INPUT -p tcp --dport 8080 -j ACCEPT
iptables -t filter -A OUTPUT -p tcp --dport 8080 -j ACCEPT
echo - Autoriser streming VLC : [OK]

# Bloquer le port 113-auth
iptables -A INPUT -p tcp --dport 113 -j REJECT
iptables -A OUTPUT -p tcp --dport 113 -j REJECT
echo - Bloquer le port 113-auth : [OK]

# Bloquer le port 111-sunrpc
iptables -A INPUT -p tcp --dport 111 -j REJECT
iptables -A OUTPUT -p tcp --dport 111 -j REJECT
echo - Bloquer le port 111-sunrpc : [OK]

# Bloquer TRIDENT MEDIA GUARD
iptables -I INPUT 1 -s  91.189.0.0/16 -j
iptables -I OUTPUT 1 -s  91.189.0.0/16 -j
#iptables -I INPUT 1 -s  91.189.0.0/32 -j
#iptables -I OUTPUT 1 -s  91.189.0.0/32 -j
#marche pas #iptables -I INPUT 1 -s 91.189.104.0-91.189.111.255  -j drop
echo - Bloquer TRIDENT MEDIA GUARD : [OK]
```

```
###### Fin Regles ######

echo Firewall mis a jour avec succes !
```

Puis tu rend le script exécutable

```
chmod +x /etc/init.d/firewall
```

Ensuite tu exécutes ce script et tu re-lanças fail2ban

```
/etc/init.d/firewall
/etc/init.d/fail2ban restart
```

# Installation de Lighttpd/Mysql/Xcache

## *Installation de Lighttpd et PHP5*

Installation des paquets nécessaires :

```
/usr/bin/apt-get install lighttpd libterm-readline-gnu-perl php5-cgi
```

Création et modification d'une configuration pour PHP5 en mode Fast CGI :

```
/bin/cp /etc/lighttpd/conf-available/10-fastcgi.conf /etc/lighttpd/conf-
available/10-fastcgi-php5.conf

/bin/sed -i -e 's/php4/php5/g' /etc/lighttpd/conf-available/10-fastcgi-php5.conf
```

Activation de la configuration ainsi créée :

```
/usr/sbin/lighty-enable-mod fastcgi-php5
```

On recharge lighttpd pour que la nouvelle configuration soit prise en compte:

```
/etc/init.d/lighttpd force-reload
```

## *Installation de Xcache*

(pour gagner encore un peu plus en performance)

Installation des dépendances nécessaires à la compilation du logiciel:

```
/usr/bin/apt-get install php5-dev make
```

Numéro de la version de XCache que l'on va installer:

```
VERSION=1.2.2
```

Téléchargement des sources de XCache:

```
/usr/bin/wget http://xcache.lighttpd.net/pub/Releases/$VERSION/xcache-
$VERSION.tar.gz \
    --output-document=/tmp/xcache-$VERSION.tar.gz
```

Décompression de l'archive obtenue:

```
/bin/tar --directory=/tmp -xzf /tmp/xcache-$VERSION.tar.gz
```

On se place dans le dossier créé:

```
cd /tmp/xcache-$VERSION
```

Compilation:

```
/usr/bin/phpize --clean
/usr/bin/phpize
./configure --enable-xcache
/usr/bin/make
/usr/bin/make install
```

Configuration de PHP pour utiliser ce module:

```
/bin/cp /tmp/xcache-$VERSION/xcache.ini /etc/php5/conf.d/xcache.ini
/bin/sed -i -e 's/^zend_extension_ts.*/; \0/' \
            -e 's/^\(zend_extension =\).*/\1
\/usr\/lib\/php5\/20060613+lfs\/xcache.so/' \
         /etc/php5/conf.d/xcache.ini
```

Configuration de la quantité de mémoire utilisée pour le cache (ici 64Mo, à toi de choisir):

```
/bin/sed -i -e 's/^\(xcache\.size[ ]*=\).*/\1 64M/' \
            -e 's/^\(xcache\.var_size[ ]*=\).*/\1 64M/' \
         /etc/php5/conf.d/xcache.ini
```

Redémarrage du serveur web

```
/etc/init.d/lighttpd force-reload
```

### *Installation du serveur SQL MySQL :*

```
apt-get install mysql-server
```

Création du mot de passe root de MySQL :

```
MYSQL_PWD=`apg -q -a  0 -n 1 -M NCL`
```

Affichage du nouveau mot de passe :

```
echo "Votre mot de passe pour l'utilisateur root de MySQL sera : '$MYSQL_PWD'."
```

Attention: Notez la valeur affichée et conservez la précieusement !

Mise en place du nouveau mot de passe :

```
mysqladmin -u root password "$MYSQL_PWD"
```

Le serveur MySQL est maintenant prêt à être utilisé.

### Création de la base de données

Création de la base de données avec un utilisateur capable de l'administrer.

Configuration du nom de la base de données :

```
MYSQL_DB=torrentflux
```

Création de la base de données :

```
echo "CREATE DATABASE IF NOT EXISTS $MYSQL_DB DEFAULT CHARACTER SET utf8 DEFAULT
COLLATE utf8_unicode_ci" \
    | mysql --user=root --password
```

Remarque: Le mot de passe demandé est celui de l'utilisateur root de MySQL.

Configuration de l'utilisateur habilité à administrer la base de données :

```
MYSQL_USERNAME="`echo $MYSQL_DB|tr '[A-Z]' '[a-z]'`"
MYSQL_USERPWD=`apg -q -a  0 -n 1 -M NCL`
```

Affichage de l'identifiant et du mot de passe de l'utilisateur pour la base de données :

```
echo "L'utilisateur habilité pour la base de donnée '$MYSQL_DB' est
'$MYSQL_USERNAME' avec le mot de passe '$MYSQL_USERPWD'."
```

Attention: Informations affichées à noter et à conserver précieusement !
Elles vous seront demandés lors de la configuration des bases MySQL de torrentflux.

Création de l'utilisateur et réglage de ses habilitations.

```
echo "GRANT ALL PRIVILEGES ON $MYSQL_DB.*
        TO $MYSQL_USERNAME@localhost
        IDENTIFIED BY '$MYSQL_USERPWD';" | mysql --user=root --password
```

## Installation de Transmission/torrentflux-b4rt

Pour commencer, installation des extensions PHP et autres logiciels nécéssaires au bon fonctionnement de Torrentflux

```
/usr/bin/apt-get install bzip2 php5-mysql php5-gd php5-cli unrar grep python \
    net-tools mawk wget unzip cksfv vlc-nox uudeview apg python-psyco python-
crypto \
    libxml-simple-perl libxml-dom-perl libdbd-mysql-perl libdigest-sha1-perl \
    bittorrent bittornado
```

Nous installons maintenant les paquets qui ne sont présents que sur Debian 4.0 Etch :

```
if [ "$(/bin/cat /etc/debian_version)" = "4.0" ]; then
  /usr/bin/apt-get install cksfv python-psyco
fi
```

La commande cksfv étant nécessaire au bon fonctionnement de Torrenflux-B4rt, il est nécessaire de le récupérer directement dans les dépôts de Debian 4.0 Etch

```
if [ "$(/bin/cat /etc/debian_version)" = "5.0" ]; then
  /bin/echo "deb ftp://ftp.debian.org/debian/ etch main" \
      | /usr/bin/tee /etc/apt/sources.list.d/etch-main.list
  /usr/bin/apt-get update
  /usr/bin/apt-get install -y cksfv
  /bin/rm /etc/apt/sources.list.d/etch-main.list
  /usr/bin/apt-get update
fi
```

### *Installation de torrentflux-b4rt*

Sélection de la version de Torrentflux-b4rt a installer. Cette version est encore BETA car l'interface HTML n'est pas parfaite (selon les développeurs) :D) :

```
VERSION=1.0-beta2
```

Téléchargement des sources:

```
/usr/bin/wget http://download.berlios.de/tf-b4rt/torrentflux-
b4rt_$VERSION.tar.bz2  \
    --output-document=/tmp/torrentflux-b4rt_$VERSION.tar.bz2
```

Décompression de l'archive:

```
/bin/tar --directory /tmp -xjf /tmp/torrentflux-b4rt_$VERSION.tar.bz2
```

Déplacement du dossier ainsi créé vers son emplacement définitif:

```
/bin/mv /tmp/torrentflux-b4rt_$VERSION /opt/torrentflux
```

Création d'un lien symbolique de la partie Web de Torrentflux vers un emplacement accessible à l'aide du serveur HTTP :

```
/bin/ln -s /opt/torrentflux/html /var/www/torrentflux
```

Il faut rendre inscriptible par le serveur le dossier de configuration:

```
/bin/chown -R www-data:www-data /var/www/torrentflux/inc/config/
```

Création du dossier destiné à recevoir les téléchargements de Torrentflux:

```
/bin/mkdir --parent /home/$MY_USER/torrentflux
```

Il faut également le rendre inscriptible par le serveur HTTP :

```
/bin/chown -R www-data:www-data /home/$MY_USER/torrentflux
```

Afin de pouvoir détecter les ports utilisés, TorrentFlux-B4rt utilise la commande netstat. Pour que cette commande fonctionne correctement, elle doit être lancée avec le compte root. Pour permettre cela, nous allons configurer sudo pour que www-data puisse lancer cette commande sans avoir à saisir un mot de passe :

```
/bin/sed -i -e '/Cmnd alias/a\
Cmnd_Alias TFB4RT_NETSTAT = /bin/netstat' \
       -e '/User privilege/a\
www-data ALL = NOPASSWD: TFB4RT_NETSTAT' \
    /etc/sudoers
```

Nous créons maintenant le script shell qui permet à TorrentFlux-B4rt d'utiliser cette configuration :

```
/bin/mkdir /opt/torrentflux/bin
/bin/echo '#!/bin/bash
# Call netstat using sudo.
/usr/bin/sudo /bin/netstat $@' | /usr/bin/tee /opt/torrentflux/bin/netstat
/bin/chmod +x /opt/torrentflux/bin/netstat
```

### *Installation du client Bittorrent Transmission*

Choix de la version de Transmission à télécharger:

```
VERSION=1.06
```

Lancement du téléchargement:

```
/usr/bin/wget http://download.m0k.org/transmission/files/transmission-
$VERSION.tar.bz2 \
    --output-document=/tmp/transmission-$VERSION.tar.bz2
```

Décompression du fichier obtenu:

```
/bin/tar --directory /tmp -xjf /tmp/transmission-$VERSION.tar.bz2
```

Décompression du patch nécessaire au bon fonctionnement avec TorrentFlux :

```
/bin/tar --directory /tmp -xjf
/opt/torrentflux/clients/transmission/Transmission-1.06_tfCLI-svn3356.tar.bz2
```

Mise en place du patch:

```
/bin/cp /tmp/Transmission-1.06_tfCLI-svn3356/cli/transmissioncli.c
/tmp/transmission-$VERSION/cli/transmissioncli.c
```

Installation des dépendances pour la compilation:

```
/usr/bin/apt-get install make gcc libc6-dev pkg-config libssl-dev
```

Déplacement dans le dossier des sources:

```
cd /tmp/transmission-$VERSION
```

Lancement de la configuration:

```
./configure --disable-gtk
```

Compilation:

```
/usr/bin/makev
```

Installation:

```
/usr/bin/make install
```

Désinstallation des logiciels nécessaires à la compilation:

```
/usr/bin/apt-get --purge remove make gcc libc6-dev libssl-dev pkg-config
```

## Installation / Configuration de Torrentflux

Redémarage du serveur http

```
/etc/init.d/lighttpd force-reload
```

### *Configuration de Torrentflux via l'URL:*

```
http://Ton_ip/torrentflux/setup.php
```

Suivez alors les différentes étapes de configuration :

1. Base de données : Désactivez la création de la base de données (en décochant la case "Create Database").
2. Configuration du serveur : Entrez /home/$MY_USER/torrenflux
comme emplacement du User Download Path
3. Vérification des dépendances logicielles : Normalement, tout devrait bien se passer.

4. Suppression du fichier setup.php : Attention, cette étape déclenche une erreur. C'est tout à fait normal, ne vous en inquiétez pas.

Une fois arrivé à l'étape de la suppression du fichier setup.php, la configuration est terminée, revenez à la ligne de commande, et supprimez le fichier setup.php:

```
/bin/rm /var/www/torrentflux/setup.php
```

Et rendez non modifiable par le serveur la configuration de Torrentflux:

```
/bin/chown -R www-data:www-data /var/www/torrentflux/inc/config/
```

Préparation d'un couple identifiant / mot de passe pour le compte administrateur:

```
TORRENTFLUX_PWD=`apg -q -a  0 -n 1 -M NCL`
echo "Votre mot de passe pour le compte 'admin' est '$TORRENTFLUX_PWD'."
```

Bien noter ce mot de passe.

Aller à la page d'indentification Torrentflux.
Le premier couple identifiant / mot de passe saisi devient votre identifiant d'administration. Vous pouvez saisir vos identifiants habituels, ou ceux proposés par la commande ci-dessus :

```
     http://VOTRE_SERVEUR(ip ou adresse)/torrentflux/
```

### *Configuration*

Une fois votre identifiant saisi, vous vous voyez présenter une page de configuration.
Transfer

Dans cette page, vous allez régler les paramètres de transfert de fichiers. Personnellement, j'utilise les réglages suivants:

* Default BitTorrent Client : Transmission
* Torrent Metainfo Client : btshowmetainfo.py
* Port Range (B+T+M) : Réglez ici les ports de connection à BitTorrent (30000-35000).
* Default Percentage When Seeding Should Stop (B+T+M+A) : 0 % : 0 pour rester indéfiniment en seed.
* Wget : Enable Passive FTP : True.
* Enable Nzbperl : All users : Autorisez tous les utilisateurs a effectuer des téléchargements depuis Usenet. (je ne m'etant pas sur cette option, ce n'est pas le sujet de ce tuto)
* Use Subdirectories : Use Usenet group name.

Webapp

Dans cette page, vous réglez les paramètres de l'interface. Je modifie les valeurs suivantes:

* Select Authentication Type: Form-Auth + Cookie : C'est une valeur qui simplifie la vie :D.
* Default language: French

* Enable template cache: False : L'activer semble causer quelques légers problèmes.

Index

Dans cette page, vous réglez les paramètres de mise à jour automatique de l'affichage. Voici ceux que j'utilise:

* AJAX update: True : Mise à jour en AJAX de la page.
* Default Torrent Search Engine: mininova
* Default Sort Order: Name - Ascending : Tri des torrents par nom (c'est le plus facile pour s'y retrouver).

Dir
Dans cette page, vous réglez les paramètres des dossiers de téléchargement de Torrentflux.

* Public Write : True : Très pratique pour effacer vos torrents depuis le shell.

Users

Dans cette page, vous pouvez créer vos utilisateur normaux. N'utilisez l'administrateur que pour l'administration, pas pour le téléchargement.

Prochaine étape: installer SSL (https), mettre en place un serveur vlc pour le streaming ;)

Voila, àprès ça, lorsque que vous téléchargerez le dernier blockbuster dès son upload sur guiks, le ratio du torrent sera supérieure à 1 bien avant que vous ayez vous même fini de le télécharger.

```
Bon DL ;D

Tutoriel fortement inspiré du blog de Lone Wolf merci à lui.

http://howto.landure.fr/

Ce tuto peut être copié, réutilisé, modifié, redistribué tant que l'on cite mon
nom(0cl0ck), GuiKs, et les sources que j'ai utilisé.
```