TP Réseaux et Systèmes d'Exploitation

Mise en place d'un serveur FTP
avec vsFTPd / pureFTPd sous netBSD 3.1

Compte-rendu

Master1 ISIS- 2007/2008 : Romain Matuszak, Romain Laisne, Clément Follet (tp08)

Table des matières

Installation de NetBSD 3.1:

A l'aide d'une disquette, on lance un système d'exploitation Linux quelconque pour pouvoir lire les informations contenues sur le CD d'installation de NetBSD, créé à partir de l'image iso téléchargée à l'adresse ftp://iso.NetBSD.org/pub/NetBSD/iso/.
Pour monter le Cdrom d'installation de netBSD :

```
> mount -t cd9660 /dev/cd0d /mnt/cd
```

On peut y voir la version de NetBSD dans le fichier Readme.files, et y trouver les images des disquettes de Boot (Boot1.fs et Boot2.fs).
On fait alors le transfert de ces fichiers sur deux disquettes séparées:

1ère disquette:	**dd if=/dev/cd0/boot1.fs of=/dev/rfd0a**	**// rfd0a=floppy**
2nde disquette:	**dd if=/dev/cd0/boot2.fs of=/dev/rfd0a**	

Ensuite, on 'reboot' le système en insérant la première puis la seconde disquette comme indiqué. On arrive dans le programme d'installation, on nous demande:
- de choisir la langue
- de modifier ou créer des partitions

Dans notre cas, nous avons attribué à netBSD la totalité du disque dur, sans créer de partition de swap (par manque de place). NetBSD reconnaît automatiquement la géométrie du disque dur. Si ce n'est pas le cas, il faut alors recopier les informations imprimées sur celui-ci lorsque netBSD le demande. Ces informations sont :
Nombre de cylindres, nombre de têtes, nombre de secteurs.
- Le « code d'ammorcage » de NetBSD qui permet de renseigner le MBR (Master Boot Record) du disque dur de la présence d'un nouvel OS (Operating System), et de pouvoir booter dessus de façon autonome.
- de choisir les composants à installer dans le cas d'une installation non complète (attention: prendre les composants Noyau, Base ET Système).

Une fois l'installation configurée, il reste à choisir le média à utiliser (en l'occurence lecteur de cdrom, mais il peut s'agir directement d'un lien ftp / http, d'un répertoire local, etc.)

Une fois NetBSD 3.1 installé, on configure la langue par défaut:

```
> wsconsctl -k -w encoding=fr
```

on vérifie que le système reconnaît bien tout le matériel:

```
> dmesg | more
```

Cette commande affiche page par page tous les périphériques reconnus par NetBSD, Dans le but de monter un serveur FTP, il nous faut posséder une carte ethernet compatible. Pour cela, il faut vérifier dans cette liste qu'un périphérique possédant une adresse mac est bien reconnu.
Dans notre cas, il s'agira du device « 3com 3C509B Etherlink III » nommé « ep0 ».

on vérifie l'état de la carte réseau :

```
> ifconfig ep0
```

celle-ci ne possède pour l'instant aucune adresse ip et ne pourra donc pas communiquer avec d'autres postes (ni avec elle-même).

on définit alors manuellement la configuration de la carte réseau:

```
> ifconfig ep0 inet 192.168.0.8 netmask 0xFFFFFF00
```

inet 192.168.0.8 pour lui donner une adresse IP locale,
netmask 0xFFFFFF00 définissant la partie fixe de l'adresse (0xFFFFFF) et le nombre de postes pouvant être mis sur le réseau (0x00), soit 254.

Cette configuration est cependant temporaire et disparaitra au prochain redémarrage de la machine.
Pour conserver ces paramètres, il faut créer un fichier « d'autoconfiguration ».
Pour cela, il suffit d'ajouter la ligne précedente au fichier ifconfig.ep0 contenu dans /etc :

```
> echo "ifconfig ep0 inet 192.168.0.8 netmask 0xFFFFFF00" > /etc/ifconfig.ep0
```

De même on peut donner un nom réseau à la machine, en ajoutant ce nom au fichier /etc/myname :

```
> echo "tp08" > /etc/myname
```

Désormais, nous pouvons redémarrer la machine et remarquer avec la commande **ifconfig ep0** que la carte réseau est bien configurée.

La commande **ping tp08** nous permettra de vérifier que toutes les couches sont bien traversées lors de l'envoi d'un « paquet », et donc que la connexion logiciel-matériel se fait correctement.

Reste maintenant à installer puis configurer le logiciel FTP Vsftpd.

Installation de Vsftpd:

Il faut tout d'abord télécharger le fichier vsftpd-2.0.4nb1.tgz (ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD-3.1/i386/All/vsftpd-2.0.4nb1.tgz).

La carte mère utilisée pour notre poste serveur ne disposant pas de port USB, il existe 2 solutions pour transférer le fichier d'installation sur le disque dur de la machine :

a) graver le fichier sur un cdrom

Il faut monter le Cdrom après l'avoir inséré, dans un répertoire que l'on aura pris soin de créer :

```
> mkdir /mnt/cdrom
> mount -t cd9660 /dev/cd0d /mnt/cdrom
```

Le fichier est alors accessible dans le dossier /mnt/cdrom .

b) le transférer par FTP

C'est la solution que nous avons utilisée car nous ne disposions pas de graveur de Cdrom.
Sur le PC possédant le fichier d'installation, installer un serveur FTP simple et gratuit (sous windows par exemple, filezilla server), et le configurer pour créer un compte (login, motdepasse) qui aura les autorisations d'accès en lecture du fichier vsftpd-2.0.4nb1.tgz. Il est aussi nécessaire d'attribuer une adresse IP fixe à ce PC (exemple 192.168.0.100).

Il suffit alors de connecter les 2 machines par un cable réseau croisé, étant donné que la carte réseau de la machine netBSD a été auparavant configurée, il suffit d'accéder à l'autre machine par FTP :

```
> cd /tmp
> ftp 192.168.0.100
(entrer login, motdepasse)
> BIN
> GET vsftpd-2.0.4nb1.tgz
```

Le fichier d'installation est alors enregistré dans le répertoire /tmp.

NB : Si la machine possède un ou plusieurs ports usb, il suffit de brancher la clé et de la monter avec cette commande :

```
> mkdir /mnt/usb
> mount -t msdos /dev/sd0a /mnt/usb
```

puis extraire et installer le paquetage :

```
> pkg_add vsftpd-2.0.4nb1.tgz
```

Le paquetage pkgsrc de NetBSD comprend les fichiers d'installation dans net/vsftpd mais nous n'avions pris que les composantes les plus importantes lors de l'installation de NetBSD, manque de place. Dans ce cas, un simple **pkg_add -r vsftpd** aurait suffit ou une installation manuelle avec la commande **cd /usr/ports/ftp/vsftpd/ && make install clean** .

Configuration de Vsftpd:

Maintenant que vsftpd est installé, on peut l'executer de cette manière :

```
> /usr/pkg/libexec/vsftpd
```

Pour rendre le serveur FTP autonome, il serait intéressant que celui-ci exécute automatiquement vsftpd lorsqu'il reçoit une commande de type FTP.
Pour cela, il faut éditer le fichier **inetd.conf** :

```
> vi /etc/inetd.conf
```

Pour y ajouter la ligne : **ftp stream tcp nowait root /usr/pkg/libexec/vsftpd vsftpd**
(cf annexe pour les commandes de base de vi).

Puis, relancer inetd :

```
> ps aux | grep inetd
> kill xxx
(xxx est le 2ème nombre sur la ligne correspondant à « inetd -l »
```

Il faut, en plus de cela, configurer vsftpd pour qu'il puisse être exécuté en arrière-plan par **inetd**.

```
> vi /usr/pkg/etc/vsftpd.conf
```

Et verifier que la ligne suivante est présente, avec la valeur « NO » : **listen=NO**
NB : Les « **#** » en début de ligne font office de commentaire, donc tout ce qui suit n'est pas pris en compte par vsftpd. Supprimer le **#** devant la ligne si besoin.

Désormais, si vsftp n'est pas déjà lancé, il s'exécutera automatiquement lors de l'envoi vers la machine d'une commande de type FTP.

Création et gestion des utilisateurs :

Désormais, il faut gérer les utilisateurs autorisés à se connecter en mode « privé », c'est à dire avec les droits d'accès suivants :

- lecture / écriture de données
- création / modification de répertoires

Pour ajouter un utilisateur FTP, il faut le créer avec la commande « useradd » et lui configurer son dossier « home ». De plus l'on va attribuer cet utilisateur au groupe « ftp ».

```
> groupadd ftp
> useradd -g ftp -d /home/ftp/prive toto
> passwd toto
(mot_de_passe)
> mkdir /home/ftp/prive
> chown root.ftp /home/ftp/prive
```

Refaire cette étape avec tous les comptes utilisateurs à créer.

L'utilisateur toto a désormais accès au contenu du répertoire /home/ftp/prive.
Il doit bien entendu lui être impossible de « remonter » vers les répertoires système.
Pour cela, il nous faut configurer le fichier **vsftpd.conf**.

```
> vi /usr/pkg/etc/vsftpd/vsftpd.conf
```

Voici les valeurs essentielles à inscrire dans le fichier de configuration (ou à modifier) :

```
anonymous_enable=YES    #autoriser les connexions anonymes
local_enable=YES    #autoriser les connexions locales
anon_upload_enable=NO    #refuser l'upload pour les anonymes
write_enable=YES    #autoriser l'écriture (en général)
chroot_local_users=YES    #empecher les utilisateurs de remonter
ftpd_banner=message_d'accueil
listen=NO    #activer inetd
non_anon_password=YES    #ne pas demander de mot de passe pour les anonymes
anon_upload_enable=NO    #désactive l'upload pour les anonymes
anon_mkdir_write_enable=NO    #désactive la création de répertoire pour les anonymes
```

```
anon_other_write_enable=NO     #désactive toute autre forme d'écriture pour les anonymes
```

Notre configuration nous permet de répondre aux besoins essentiels du serveur.
Pour améliorer la sécurité, il est préférable de créer une liste d'utilisateurs bannis automatiquement, tels que « root », « daemon », « nobody », qui peuvent être à l'origine de conflits.

```
> vi /usr/pkg/etc/vsftpd/vsftpd.user_list
Entrer ces utilisateurs (1 par ligne, liste non exhaustive) :
root, apache, nobody, www, bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator.
```

Il faut bien sûr paramétrer vsfptd pour lui indiquer cette liste d'utilisateurs bannis :

```
> echo "userlist_enable=YES" >> /usr/pkg/etc/vsftpd/vsftpd.conf
> echo "userlist_deny=YES" >> /usr/pkg/etc/vsftpd/vsftpd.conf
> echo "userlist_file=/usr/pkg/etc/vsftpd/vsftpd.user_list" >> /usr/pkg/etc/vsftpd/vsftpd.conf
```

Désormais ces utilisateurs seront automatiquement bannis, sans même leur demander de mot de passe.

Il reste désormais à appliquer les droits corrects au dossier « home » du FTP :

```
> mkdir /home/ftp/prive
> chown root.ftp /home/ftp/prive
> chmod 770 /home/ftp/prive
(770 = rwxrwx---)
```

L'accès anonyme se fait par l'utilisateur **ftp** dont il faut configurer le dossier home :
On fera appartenir cet utilisateur à un groupe **anonyme** dont on configure les droits:

```
> addgroup anonyme
> usermod -g anonyme ftp
> mkdir /home/ftp/anonyme
> chown root.anonyme /home/ftp/anonyme
> chmod 755 /home/ftp/anonyme
(755 = rwxr-xr-x)
```

Vsftpd est configuré pour que l'utilisateur **ftp** puisse se connecter sans mot de passe, et il lui est impossible de créer/modifier des fichiers/répertoires.

Le serveur FTP est maintenant configuré, sécurisé et prêt à l'emploi!

Aller plus loin

La méthode de base pour la gestion des utilisateurs est de les créer, un par un, par la commande **useradd**. Mais dans ce cas, les utilisateurs du FTP sont enregistrés aussi sur le système (netBSD), et les risques qu'un utilisateur puisse accéder aux fichiers système ne sont pas nuls.
La solution est de créer un utilisateur « virtuel » unique, tous les utilisateur réels venant se connecter sur ce dernier. Il n'y a ainsi qu'un seul « user » à créer dans le système et la gestion de la sécurité est beaucoup plus aisée.
C'est cette méthode que nous allons développer par la suite, bien que nous n'ayons pas réussi à la mettre en place sur notre machine avec vsftpd.

Tout d'abord nous allons créer une base de données contenant les login / mots de passe de chaque utilisateur autorisé à se connecter au serveur. Cette base de données sera cryptée pour une sécurité maximale.

Installation du logiciel db :

Le package « berkeley db » pour netBSD 3,1 se trouve ici :
ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD-3.1/i386/All/db3-3.11.2nb3.tgz
Pour l'installer, transférer le fichier .tgz (par clé usb, Cdrom ou FTP, comme vu auparavant) puis taper :

```
> pkg_add db3-3.11.2nb3.tgz
```

Une fois installé, créer un fichier texte contenant, ligne par ligne, les login et mot de passe de chaque utilisateur.

```
> vi /tmp/logins.txt
login1
motdepasse1
login2
motdepasse2
login3
```

```
motdepasse3
etc.
```

executer db sur ce fichier texte pour générer la base de données d'utilisateurs puis appliquer des permissions restreintes à ce fichier :

```
> db_load -T -t hash -f /tmp/logins.txt /etc/vsftpd_login.db
> chmod 600 /etc/vsftpd_login.db
```

Installation de PAM :

Il faut désormais installer « pam » (Pluggable Authentication Modules), disponible ici :
ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD-3.1/i386/All/PAM-0.77nb5.tgz
et son module d'authentification dbm :
ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD-3.1/i386/All/pam-dbm-0.2nb1.tgz

Installer ces deux modules par la commande **pkg_add** .

il faut configurer le module pam :

```
> vi /etc/pam.d/vsftpd
(y ajouter : )
auth required /usr/pkg/lib/security/pam_userdb.so db=/etc/vsftpd_login.db
account required /usr/pkg/lib/security/pam_userdb.so db=/etc/vsftpd_login.db
```

Ajouter (ou modifer) ces options dans vsftpd.conf :

```
guest_enable=YES
guest_username=virtualftp
pam_service_name=vsftpd
```

Création de l'utilisateur virtuel :

Enfin, créer l'utilisateur **virtualftp** et son dossier correspondant.

```
> useradd -d /home/ftp/virtual
> mkdir /home/ftp/virtual
> chown virtual.wheel /home/ftp/virtual
```

```
> chmod 700 /home/ftp/virtual
```

La configuration est normalement correcte pour une utilisation de vsftpd avec utilisateurs virtuels. Cependant nous ne réussissons en aucun cas à nous authentifier auprès du serveur lors de la connexion.
Il semblerait (après plusieurs recherches sur internet) que la gestion des utilisateurs virtuels sous netBSD ne soit pas aisée...
Le problème peut venir :

- du fichier .db généré par db_load, et qui n'est pas correctement reconnu
- du module d'authentification pam-dbm qui n'est pas adapté à la base de données .db

Dans le cas d'un besoin impératif de sécurité, il est donc préférable de se rabattre sur un logiciel FTP gérant plus facilement les utilisateurs virtuels.
Nous allons donc, pour cela, remplacer **vsftpd** par un autre logiciel, **pure-FTPd**.

Utilisateurs Virtuels : Installation de pure-FTPd

Tout d'abord, supprimons toute trace des précédents packages installés.

```
> pkg_delete vsftpd PAM pam-dbm db3
```

Ensuite, installons pure-FTPd (ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD-3.1/i386/All/pure-ftpd-1.0.21.tgz)

```
> pkg_add pure-ftpd
(si demandé : )
> cp /usr/pkg/share/examples/rc.d/pure_ftpd /etc/rc.d/pure_ftpd
> echo "pure_ftpd"=YES >> /etc/rc.conf
```

Et redéfinissons le nouveau **inetd** (sans oublier de le relancer)

```
> vi /etc/inetd.conf
ftp stream tcp nowait root /usr/pkg/sbin/pure-ftpd pure-ftpd -l puredb:/usr/pkg/etc/pureftpd.pdb
> kill xxx && /usr/sbin/inetd -l
(xxx = numéro du processus inetd)
```

L'option -l est un lien vers la base de données cryptée des utilisateurs, nous la définirons plus tard.
Redéfinissons un nouvel utilisateur virtuel appartenant au groupe **ftpgroup** :

```
> groupadd ftpgroup
> useradd -g ftpgroup -d /dev/null -s /etc ftpuser
> chown ftpuser.ftpgroup /home/virtualftp
> chmod 700 /home/virtualftp
```

L'avantage de pure-ftpd est qu'il est d'origine développé pour gérer les utilisateurs virtuels.
Il contient en effet un utilitaire **pure-pw** très puissant et très modulable, qui permet de génerer une base de données d'utilisateurs virtuels.
Créons un utilisateur « toto » virtuel, qui viendra se connecter sur l'utilisateur système « ftpuser » dont on va restreindre les droits :

```
> mkdir /usr/pkg/etc
> pure-pw useradd toto -u ftpuser -d /home/virtualftp
(entrer le mot de passe de l'utilisateur)
```

Cette commande a créé un fichier /usr/pkg/etc/pureftpd.passwd qui regroupe les informations de chaque utilisateur, mais ce n'est pas le fichier base de données!
Il faut générer ce dernier par la commande :

```
> pure-pw mkdb
```

L'utilisateur anonyme est par défaut géré par l'utilisateur système « **ftp** », il faut donc définir les droits de ce dernier:

```
> mkdir /home/anonyme
> usermod -g ftpgroup -d /home/anonyme
> chown anonyme.ftpgroup /home/anonyme
> chmod 500 /home/anonyme
```

L'utilisateur anonyme a pour login « anonymous » et aucun mot de passe.

Voilà notre serveur configuré et sécurisé au maximum.

Sous netBSD, il semble donc préférable d'utiliser pureFTPd plutôt que vsftpd, car pour ce dernier nous ne savons toujours pas s'il est possible (sous netbsd) d'utiliser les utilisateurs virtuels, dont le niveau de sécurité en fait une option indispensable pour tout serveur FTP professionnel.

Annexe - Les commandes de base de vi

Raccourcis	Description
<i>	insère du texte avant le curseur
<d><w>	supprime un mot
<d><d>	supprime une ligne
<x>	supprime le caractère sous le curseur
<k>	monte d'une ligne
<j>	descend d'une ligne
<h>	se déplace d'un espace vers la gauche
<f><x>	déplace le curseur sur la première occurence de x
<F><x>	déplace le curseur sur la dernière occurence de x
<ctrl-f>	défile d'un écran vers le bas
<ctrl-b>	défile d'un écran vers le haut
:q	quitte VI
:q!	quitte sans sauvegarder le fichier
:wq	sauvegarde les modifications sur disque et quitte VI