Administration Système

Projet: Mise en place d'une passerelle Samba

Vivien Boistuaud

Fabien Bidet

Ingénieurs 2000 UMLV

IR1 2005-2006

Table des matières

Introduction

L'objectif de ce document est de détailler l'installation, la configuration et l'exploitation d'un serveur de fichiers UNIX et d'une passerelle SAMBA permettant l'accès au serveur de fichier pour les clients Windows d'une entreprise.

Entre autre, il présente des scripts permettant d'effectuer les manipulations en question et la maintenance des serveurs. Il décrit également les processus de sécurisation des serveurs à adopter, ainsi que ceux qui pourraient être ajoutés optionnellement pour renforcer la sécurité.

Note : Lorsque cela n'est pas précisé, les commandes sont à saisire en utilisant les droits `root`.

1 Présentation du projet

1.1 Matériel requis

Pour ce projet, nous devons disposer, au minimum, du matériel suivant :

- 2 serveurs
  - Un disque système d'au moins 4 Go par serveur
  - 256 Mo de mémoire vive par serveur (512 Mo de swap)
  - Les serveurs doivent, autant que possible, disposer de matériel récent
  - Un lecteur de bande pour les sauvegardes du serveur de fichier
  - Un disque dur supplémentaire pour les données utilisateur, de taille suffisante pour y stocker les données de tous les utilisateurs (par exemple 100 Go).
- Au moins un client Windows
- Un client linux (optionnel)

Dans notre cas, les serveurs utilisaient était des Intel Pentium 3 et 4 (architecture ix86) et les clients testés étaient également des Pentium 3 et 4. La taille des partitions allouées aux systèmes étaient d'environ 4 Go.

1.2 Configuration requise

1.2.1 Pour les serveurs

Un système UNIX doit être préinstallé sur les disques systèmes des deux serveurs. Ces systèmes doivent être configurés et fonctionnels, notamment au niveau des interfaces réseaux, de la configuration IP et des tables de routage associées.

Les serveurs utilisés dans le cadre de ce projet fonctionnaient sous Debian GNU/Linux, et ils avaient la configuration IP suivante :

Type de serveur	Adresse IP / Masque	Nom de machine
Passerelle Samba	`10.0.0.1/24`	`alfred`
Serveur de fichiers	`10.0.0.2/24`	`batman`

1.2.2 Pour les clients

Un système d'exploitation doit être préinstallé sur chacun des clients et leur configuration réseau (dont IP) doit avoir été préalablement effectuée. Les clients Windows testés fonctionnaient sous Windows 2000 et XP, tandis que les clients UNIX testés fonctionnaient sous Debian GNU/Linux.

La configuration IP des clients était la suivante :

Type de serveur	Adresse IP / Masque	Nom de machine
Client Windows #1	`10.0.0.10/24`	`robin`
Client Unix #1	`10.0.0.11/24`	`vicky`

1.3 Choix technologiques et intérêt du projet

1.3.1 Le serveur de fichiers

Nous avons décidé de mettre en place un serveur NFS pour le stockage de données sur le réseau. Ce service permet de partager des fichiers entre plusieurs machines du même réseau. Les utilisateurs modifient les fichiers comme si c'était sur leur disque dur local : l'accès au partage se fait de manière transparente.

Les avantages sont que les utilisateurs peuvent partager des fichiers de manière simple, cela évite de stocker ses données en local, ce qui en cas de défaillance matérielle, peut être catastrophique, et les sauvegardes des données ne se font plus que sur le serveur NFS (plutôt que sur toutes les machines du réseau).

Par contre, l'inconvénient principal est que si le serveur est en panne, il est possible qu'un grand nombre d'utilisateurs, travaillant sur des fichiers contenus sur le serveur, ne puissent plus travailler. D'où l'intérêt, comme nous l'aborderons plus loin, d'implémenter une stratégie de tolérance aux pannes.

1.3.2 La passerelle Samba

Samba est le leader des logiciels libres permettant d'interfacer un système ou réseau UNIX avec des clients tournant sous les OS de la famille Microsoft Windows. Il existe très peu d'autres logiciels permettant d'assurer cet interfaçage.

Les clients Windows de la série NT (type NT4/2000/XP) supportent le NFS, et donc l'accès au serveur de fichiers que nous allons configurer, à l'aide d'un pilote additionnel fournit avec le système. Cependant, Microsoft a conçu ses OS pour être adaptable aux communications avec UNIX, mais pas pour pouvoir assurer une prise en charge totale.

C'est pourquoi Samba permet une bonne prise en charge des clients Windows en utilisant les protocoles propriétaires Microsoft. Il est donc conseillé d'opter pour une solution de type « passerelle samba » plutôt que d'utiliser un pilote NFS pour Windows.

De plus, samba permet beaucoup plus qu'une simple gestion des partages de fichier, ce qui sera abordé dans les sections 3 et 5 du présent rapport.

1.3.3 Les clients Windows

Afin de garantir la sécurité optimale d'accès aux fichiers via la passerelle Samba, il est conseillé d'utiliser un système Windows de la série NT (NT/2000/XP) et non un système de la série 9x car il est possible d'accéder à ces systèmes sans utiliser de mot de passe, ce qui représente une faille de sécurité majeure.

Le projet est orienté de façon à faire un minimum de modifications sur les clients pour qu'ils puissent se connecter : c'est à la passerelle UNIX de s'adapter aux protocoles et spécificités des systèmes Windows clients.

1.3.4 Les clients Unix

Il n'est pas indispensable de posséder des clients UNIX. Cependant, si vous souhaitez que des clients fonctionnant sous UNIX puissent accéder au serveur de fichier, il est déconseillé de les faire accéder directement au serveur NFS.

En effet, les deux serveurs ne partageant pas la même base de données utilisateurs, les logins et mots de passe des utilisateurs ne sont valides que sur la passerelle Samba. De plus, pour des raisons de sécurité, seul le serveur Samba sera autorisé à monter le système de fichiers NFS du serveur de données, en mode superutilisateur (`root`).

2 Le serveur de fichiers NFS

Remarque : L'ensemble des commandes de configuration du serveur de fichier doivent être effectués avec les privilèges de l'utilisateur `root`*, soit en se loguant en tant que* `root`*, soit en obtenant les droits superutilisateur à l'aide de la commande* `su`*.*

2.1 Pré-requis à l'installation d'un serveur de données

2.1.1 Installation d'un disque de données utilisateurs

Le serveur de données est donc un PC sous Debian GNU/Linux. Ce dernier allant contenir un grand nombre de fichiers, nous optons pour l'ajout d'un nouveau disque dur dédié aux données utilisateurs, qui seront partagées sur le réseau. Ce nouveau disque dur sera monté sous `/mnt/users` dans l'arborescence du système d'exploitation.

2.1.1.1 Dans le cas d'un disque unique (ou de disques isolés)

La démarche ci-dessous convient pour l'installation d'un disque unique, dans le cas où les besoins d'évolution d'espace disques seront pratiquement inexistants. C'est celle que nous avons adopté en raison des moyens qui étaient mis à notre disposition. Le choix de l'utilisation de disques extensibles (LVM) est cependant préférable (cf. section suivante).

Dans ce cas, il faut créer une partition à partir du disque dur et la formater en `ext3fs` (système de fichiers journalisé natif de Linux et dérivé d'`ext2fs`). Pour cela, exécutez la commande:

```
fdisk chemin_du_périphérique
```

Puis saisir les commandes `'n'` pour créer une partition, `'p'` pour indiquer qu'elle est primaire, `'1'` comme numéro de partition primaire, puis laisser les valeurs par défaut pour le premier et le dernier cylindre.

La partition étant désormais créé, on lui attribue un type à l'aide de la commande `'t'` en saisissant comme valeur `83`. Puis, on sauvegarde les informations sur la table des partitions à l'aide de la commande `'w'`.

Il ne reste plus qu'à formater la partition nouvellement créée en utilisant le système de fichier `ext3fs` à l'aide de la commande :

```
mkfs -t ext3 chemin_de_la_partition
```

Nous planifieront, dans la suite de ce projet, les opérations de maintenance associées à ce système de fichiers afin de garantir au maximum son intégrité.

Remarque : suivant que le disque est SCSI, SATA ou IDE, le chemin du périphérique peut être du type /dev/sdx (deux premiers cas) ou /dev/hdx (dernier cas). Si vous avez créé une seule partition comme décrit ci-dessus, le chemin de la partition devrait être le nom du périphérique suivit du chiffre 1.

2.1.1.2 Dans le cas de disques extensibles (LVM)

Pour pouvoir utiliser le LVM sous Debian GNU/Linux, le LVM étant une technologie disponible initialiement sur HP-UX, il est nécessaire d'installer les paquets de gestion du LVM à l'aide de la commande :

```
apt-get install lvm2 lvm-common
```

Notez que dans le cas d'une installation sur un noyau linux 2.6, il faudra installer le paquetage `lvm10` au lieu de `lvm2` et suivre les instructions fournies pour que celui-ci fonctionne.

Dans un premier temps, il faut partitionner le (ou les) disques qui vont faire partie du LVM. Pour cela, reportez vous à l'utilisation de la commande `fdisk` décrite précédemment. *Notez qu'en LVM, il n'est possible de gérer que des disques entiers.*

Une fois le(s) partition(s) créée(s), il faut créer le ou les volumes physiques LVM à l'aide de la commande (répétée une fois par partition) :

```
pvcreate chemin_de_la_partition
```

Les partitions étant désormais prêtes à l'utilisation en LVM, il faut désormais créer un groupe de volumes LVM qui contiendra les volumes physiques créés précédemment (une seule fois avec la première partition) :

```
vgcreate vguserdisk chemin_de_la_partition
```

Pour l'ajout de volumes supplémentaires au groupe de volume LVM créé (ici `vguserdisk`) il faut ensuite utiliser la commande :

```
vgextend vguserdisk chemin_de_la_partition
```

On peut vérifier la taille du groupe de volumes LVM à l'aide de la commande `vgdisplay`, ce qui permet également de connaître la taille exacte utilisable sur le LVM. Pour créer un volume logique, il faut ensuite utiliser la commande :

```
lvcreate –L taille –n lvuserdisk vguserdisk
```

Note : La taille est exprimée par défaut en mega, les suffixes K, M, G et T permettant de préciser une taille en Ko, Mo, Go et To respectivement.

On peut vérifier le bon déroulement de la création à l'aide de la commande `lvdisplay`. Elle permet de vérifier que la création s'est déroulée correctement et que le volume créé est accessible dans `/dev/vguserdisk/lvuserdisk`.

Si on souhaite agrandir la taille d'un volume logique, par exemple à la suite de l'ajout d'un disque au groupe LVM, il suffit d'utiliser la commande suivante :

```
lvextend –L nouvelle_taille /dev/vguserdisk/lvuserdisk
```

Il ne reste plus qu'à formater le volume logique nouvellement créé en utilisant le système de fichier `ext3fs` à l'aide de la commande :

```
mkfs -t ext3 chemin_de_la_partition
```

Cette solution est, certes, plus complexe que la première solution proposée, mais présente l'avantage de pouvoir facilement et rapidement étendre la capacité de stockage du serveur de données en fonction du besoin des utilisateurs et des évolutions du réseau et de l'entreprise.

Ces manipulations ont été testées avec succès à l'aide d'un ordinateur virtuel, en l'absence du matériel nécessaire à la mise en place d'un LVM utilisant plusieurs disques physiques.

2.1.2 Montage et configuration

Une fois que la partition est créée et formatée, il faut la monter manuellement dans un premier temps. Pour cela utilisez les commandes :

```
mkdir /mnt/userdisk/
```

```
mount –t chemin_de_la_partition /mnt/userdisk
```

Remarque : pour les volumes LVM, le chemin de la partition est le chemin du volume logique.

Pour que le montage s'effectue automatiquement au démarrage de la machine, il faut éditer le fichier `/etc/fstab` en y ajoutant la ligne suivante :

```
chemin_de_la_partition  /mnt/userdisk  ext3  defaults,errors=remount-ro   0  0
```

Remarque : Il ne faut pas mettre d'espaces mais des tabulations entre les différentes valeurs indiquées ci-dessus.

Il ne faut pas oublier d'attribuer les bons droits au dossier `/home/users` grâce à la commande :

```
chown root:root /mnt/userdisk
```

```
chmod 755 /mnt/userdisk
```

Ainsi, l'utilisateur `root` possèdera tous les droits de lecture et d'écriture sur le point de montage de la partition servant au stockage des documents utilisateurs. Les utilisateurs, quand à eux, auront les droits nécessaires pour lire et traverser le point de montage, ce qui sera suffisant pour qu'ils puissent accéder à leur répertoire personnel et aux répertoires partagés entre les utilisateurs.

2.2 Installation du démon NFS

Avant de commencer l'installation, il est nécessaire de charger le module `nfsd` si celui-ci n'est pas compilé dans le noyau (au besoin, vérifiez à l'aide de la commande `lsmod`). Cette opération se fait à l'aide de la commande :

```
modprobe nfsd
```

Pour que le chargement se fasse automatiquement aux prochains démarrages, il faut ajouter une nouvelle ligne au fichier `/etc/modules` et y placer le nom du module : `nfsd`.

Ensuite, pour utiliser nfs nous avons besoin d'installer `tcp-wrappers` pour le monitoring et le filtrage des requêtes entrantes, qui permettra d'assurer une certaine sécurité sur le serveur :

- Sur Debian : utilisez la commande `apt-get install tcpd`.
- Sur tout système Unix : récupérez une archive tar de l'application `tcp-wrappers` puis exécuter les commandes suivantes :

```
$ tar xzvf tcp_wrappers_xxx.tar.gz
$ cd tcp_wrappers_xxx
$ make REAL_DAEMON_DIR=/usr/sbin STYLE=-DPROCESS_OPTIONS linux
$ su
# make install
# exit
```

Après cela, il faut installer `nfs-utils`, qui contient le démon `nfs` :

- Sur Debian : utilisez la commande `apt-get install nfs-common nfs-user-server`
- Sur tout système Unix : téléchargez `nfs-utils` et saisissez les commandes :

```
$ tar xzf nfs-utils_xxx.tar.gz
$ cd nfs-utils_xxx
$ ./configure --prefix=/usr --enable-nfsv3
$ make
$ su
# make install
# exit
```

A présent le serveur NFS est prêt à être configuré pour devenir serveur de données.

2.3 Configuration du démon NFS

Pour configurer le serveur NFS, il faut éditer trois fichiers de configuration.

2.3.1 Configuration particulière à NFS

Le premier est le fichier `/etc/exports` qui contient les répertoires à mettre en partage NFS ainsi que les machines qui ont les droits d'accès (lecture ou lecture/écriture) sur ces répertoires. La syntaxe des lignes est la suivante :

`répertoire_partagé  host_autorisé_1`[ou `network/netmask`] (`option1`,`option2`) `host_autorisé_2`[ou `network/netmask`] (`option3`,`option4`).

Les options peuvent être '`ro`' pour read only (accès en lecture uniquement), '`rw`' pour read write (accès en lecture et écriture), `no_root_squash` (accès au répertoire en mode root).

Pour notre serveur, nous souhaitons que les utilisateurs puissent accéder en lecture et écriture `/mnt/userdisk` par l'intermédiaire de la passerelle SAMBA (`alfred`). Pour autoriser la machine `10.0.0.1` (serveur Samba) à accéder au partage `/mnt/userdisk` en lecture et écriture, et en mode `root`, il faut insérer la ligne suivante dans le fichier `/etc/exports` :

```
/mnt/userdisk 10.0.0.1(rw,no_root_squash)
```

2.3.2 Sécurisation du serveur à l'aide de tcp-wrappers

Les deux autres fichiers sont `/etc/hosts.allow` et `/etc/hosts.deny`. Le premier indique les machines qui sont autorisées et le deuxième indique les machines qui doivent être refusées d'accès. Pour les deux fichiers, la syntaxe des lignes est la suivante :

```
service: host [ou network/netmask] , host [ou network/netmask]
```

Les différents services possibles (liés à NFS) sont `portmap` (démon assignant les ports de manière dynamique pour les services RPC = Remote Procedure Call), `lockd` (processus qui permet aux clients NFS de verrouiller des fichiers sur le serveur), `mountd` (processus qui reçoit la requête de montage d'un client et vérifie que le système de fichiers demandé est bien monté), `rquotad` (processus qui fournit des informations sur les quotas utilisateur s'appliquant sur aux utilisateurs distants), et `statd` (processus qui avertit les clients NFS lorsqu'un serveur est redémarré).

Pour notre serveur nous autorisons le serveur SAMBA à utiliser tous les services, pour cela nous ajoutons, dans le fichier `/etc/hosts.allow`, la ligne suivante :

ALL : 10.0.0.1

A présent il est possible d'accéder au partage `/mnt/userdisk` du serveur NFS à partir du serveur Samba (voir « partie 3. Configuration client » pour le montage du dossier en local sur un client).

2.4 Sauvegardes des fichiers utilisateurs

Étant donné que la plupart des données des utilisateurs seront stockés sur le serveur, il est nécessaire d'effectuer des sauvegardes régulières sur un périphérique de stockage extérieur (un lecteur de bande par exemple) en cas de « crash » du disque dur.

Admettons qu'un lecteur de bande soit disponible sous `/dev/lecteurBande`. Nous allons créer une archive du dossier `/mnt/userdisk` à l'aide de la commande `cpio –oc` et la placer sur le lecteur de bande.

Pour que la sauvegarde soit périodique, nous utilisons le démon `crond`. Nous éditons le fichier `/etc/crontab` de la façon suivante :

```
0 0     *  *   5  root   cpio –oc </mnt/userdisk > /dev/lecteurBande
```

La sauvegarde du dossier `/home/users` sur le lecteur de bande s'effectuera tous les vendredi à 0 :00.

Pour que la sauvegarde s'effectue tous les soirs à 0 :00, par exemple, il suffit de modifier la partie « `0 0  *  *  5` » par « `0 0  *  *  *` ». Cela dépend de la criticité des données stockées et des exigences de l'entreprise en matière de sécurité des données.

Si un jour il faut changer le disque dur (contenant les données) sur le serveur pour défaillance matériel. Il suffira de restaurer l'archive pour récupérer toutes les données sauvegardées auparavant. Pour cela il faudra taper : `cpio -iudmc < /dev/lecteurBande`. Les fichiers seront restaurer exactement à l'endroit d'où ils ont été sauvegardés (dans notre cas dans le dossier `/mnt/userdisk`).

D'autres stratégies, non explicitées ici, peuvent permettre une meilleure sécurisation des données et une tolérance aux pannes. Pour cela, reportez-vous à la section 5 du présent rapport.

3 La passerelle SAMBA

La passerelle SAMBA est à la fois cliente du serveur de fichiers NFS et passerelle pour les clients utilisant le protocole NetBIOS (dans notre cas, NetBIOS est encapsulé dans le réseau IP).

3.1 Configuration en tant que client NFS

Pour que le serveur Samba puisse se connecter au serveur de fichier en tant que client NFS, il est nécessaire de charger le module `nfs` dans le noyau du système d'exploitation, si ce n'est déjà fait. Pour cela, il suffit d'exécuter la commande :

```
modprobe nfs
```

Pour que le chargement se fasse automatiquement aux prochains démarrages, il faut ajouter une nouvelle ligne au fichier `/etc/modules` et y placer le nom du module : `nfs`.

Il faut ensuite installer le paquetage contenant les applications et pilotes communs aux serveurs et clients nfs, nommé (sous Debian) `nfs-common`. Cette installation se fait à l'aide de la commande :

```
apt-get install nfs-common
```

Il faut ensuite créer un répertoire pour que l'arborescence NFS puisse être montée. On choisit pour cela le répertoire `/mnt/userdata` sur le serveur Samba. La création se fait à l'aide de :

```
mkdir /mnt/userdata
chmod 755 /mnt/userdata
chown root:root /home/userdata
```

Le montage de la racine NFS se fait ensuite, dans un premier temps manuellement, à l'aide de la commande :

```
mount –t nfs 10.0.0.2:/mnt/userdisk /mnt/userdata
```

Comme le serveur Samba est accrédité pour monter l'arborescence NFS en tant que `root`, il a tous les droits sur les fichiers de l'arborescence, et les droits d'accès sont fonction des droits de chaque utilisateur sur le serveur Samba.

Pour que le système de fichier NFS soit monté automatiquement au prochain démarrage du serveur Samba, il est nécessaire d'ajouter la ligne suivante dans le fichier `/etc/fstab` :

```
10.0.0.2:/mnt/userdisk      /mnt/userdata       nfs    defaults     0      0
```

Le client NFS est désormais fonctionnel et le système de fichier monté sous `/mnt/userdata` est accessible exactement comme n'importe quel répertoire local.

3.2 Installation des démons Samba

Samba utilise deux démons nommés `nmbd` et `smbd`. Pour les installer sous Debian, il suffit de saisir la commande suivante :

```
apt-get install samba samba-common
```

Lors de l'installation nous sont demandées diverses informations. A la question concernant le groupe de travail / Domaine auquel doit appartenir le serveur Samba, répondez simple `Workgroup` pour le moment, car nous effectueront la configuration manuellement plus tard.

Lors de la question concernant le chiffrement des mots de passe, répondez oui, ceci étant la solution la plus sécurisée. Les contraintes techniques qui en découlent seront expliquées plus loin dans ce rapport. Concernant la question sur les serveurs WINS fournit par DHCP, cela dépend du réseau auquel votre passerelle s'intègre. Généralement, seuls les serveurs Windows sont capables

de fournir les adresses des serveurs de noms WINS par DHCP. Dans la majorité des cas, il convient de répondre non.

Enfin, le programme d'installation demande si le serveur doit être exécuté en tant que démon ou dépendre d'`inetd`. Etant donné que notre passerelle Samba est ici supposée une charge assez conséquente, il est fortement recommandé de l'exécuter en tant que démon. L'option inetd n'est tolérable qu'en cas d'utilisation de Samba pour un faible nombre de requêtes, très étalées dans le temps.

Enfin, il faut accepter la création de la base de données des mots de passe cryptés pour Windows. Celle-ci permet des communications sécurisées lors des processus d'authentification entre les clients et la passerelle, ce qui est essentiel.

Pour démarrer, arrêter, ou redémarrer le serveur Samba, il suffit d'exécuter le script `/etc/init.d/samba` avec une des directives `start`, `stop` ou `restart`.

L'installation étant terminée (de la version 3.0.14a sous debian), et les démons étant automatiquement configurés pour se lancer au démarrage du système, il faut maintenant configurer le serveur Samba.

3.3 Configuration de Samba

Le fichier de configuration de samba se trouve, par défaut sous Debian, dans le répertoire `/etc/samba` et se nomme `smb.conf`.

3.3.1 Configuration globale

La section `[global]` du fichier de configuration définit un certain nombre d'informations de configuration globales du serveur Samba. Les principaux que nous avons utilisés sont :

```
   netbios name = IG2K-NFS-SMB
   workgroup = ig2k
   server string = Passerelle Samba-NFS d'entreprise (%h Samba version %v)
  security = user
   obey pam restrictions = yes
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\sUNIX\spassword:* %n\n
*Retype\snew\sUNIX\spassword:* %n\n .
   syslog = 0
   log file = /var/log/samba/log.%m
   max log size = 1000
   dns proxy = No
   panic action = /usr/share/samba/panic-action %d
   invalid users = root
```

Les directives ont les significations suivantes :

- `netbios name` : permet de définir un nom de machine NetBIOS (protocole de communication pour les échanges de fichier Windows) différent du nom de machine UNIX.
- `workgroup` : permet de fixer le nom du groupe de travail auquel est rattaché le serveur (en général, ce doit être le même groupe de travail que les stations Windows du réseau mais pas obligatoirement)
- `server string` : permet de définir une description approfondie du serveur (son rôle, sa version, sa localisation…). Les variables %h et %v seront respectivement remplacées par le nom de machine UNIX et par la version du serveur Samba.
- `security` : permet de définir si la sécurité est gérée au niveau utilisateur ou au niveau ressource. Il est conseillé d'opter pour une sécurité `user` car elle forcera tous les utilisateurs à s'authentifier avant de pouvoir accéder aux ressources partagées.

- `obey pam restriction` : permet de supporter les mots de passe encryptés Windows.
- `passwd program` : permet de définir le logiciel de changement de mots de passe UNIX ainsi que la façon de l'utiliser (%u pour le login de l'utilisateur dont on souhaite changer le mot de passe).
- `passwd chat` : permet de préciser les messages envoyés par l'application de changement de mots de passes, ainsi que les répondre à fournir, %n désignant le nouveau mot de passe.
- `syslog` : permet de définir si l'application Samba logue les évènements associés dans le log system. Ici nous préférons loguer les évènements dans un fichier séparé.
- `log file` : permet de définir le fichier de log personnalisé. Si la variable %m est utilisée, un fichier de log sera créé par client. Ici nous avons choisis de loguer les évènements indépendamment pour chaque client.
- `max log size` : taille maximale de chaque log en Ko.
- `dns proxy` : permet de définir si les noms de serveurs WINS doivent être obtenus par DHCP.
- `panic action` : permet de définir une action à entreprendre en cas d'erreur interne du serveur Samba.
- `invalid users` : permet de définir les utilisateurs qui n'ont pas le droit de se loger sur le système par NetBIOS. Pour des raisons de sécurité, il est fortement conseille de désactiver le compte `root`, ce que nous avons fait.

3.3.2 Configuration d'un partage de fichiers

3.3.2.1 Structure générale

La structure général d'un partage ressemble à celle-ci :

```
[partage]
   comment = commentaire du partage
   path = /chemin/vers/répertoire/partagé
   valid users = @groupe_autorisé1 utilisateur_autorisé1 @groupe2 user2
   writeable = [yes | no] #ecriture autorisee
   read only = [yes | no] #lecture seule
   create mask = xxxx # masque UNIX appliqué pour creation de fichiers
   directory mask = xxxx # masque UNIX appliqué pour accès aux répertoires
   force group = groupname # nom du groupe sous lequel les utilisateurs
                           # agissent s'ils sont membres de plusieurs
   write list = user1 @group1 user2 @group2 #liste des personnes autorisées
                                            #à écrire dans le dossier
   read list = @group3 user3 #liste des personnes autorisées en lecture
   browseable = [yes | no]   #visible par tous les utilisateurs même ceux
                             #non autorisés
```

3.3.2.2 Le partage spécial `homes`

Le partage `[homes]` a un statut particulier car il est générique pour tous les utilisateurs. En effet, il permet de créer un partage automatique, portant comme nom le login de l'utilisateur, et lui permettant d'accéder à son dossier `home` UNIX.

Dans la section 3.4 est abordé l'ajout d'utilisateur, la création de son espace personnel, ainsi que la gestion des quotas d'espace disque, qui permettront à l'utilisateur de ne pas saturer l'espace disque.

La configuration que nous avons adoptée pour les répertoires `homes` est la suivante :

```
[homes]
   comment = Repertoire Personnel
   read only = no
   writeable = yes
```

```
   browseable = no
   create mask = 0600
   directory mask = 0700
```

Cette configuration permet de n'afficher qu'à l'utilisateur concerné son répertoire, lui donner les droits d'accès en écriture. Cependant, par mesure de sécurité et pour empêcher l'introduction de programmes malveillants sur le serveur, nous avons mis le masque de création des fichiers à 600. Pour les répertoires, 700 semble une bonne valeur, afin d'autoriser leur traversée.

3.3.2.3 Ajout d'un nouveau partage

Dans notre cas, il s'agit principalement de permettre à chaque utilisateur Windows de stocker ses propres fichiers sur son espace alloué sur le serveur de fichiers. Cependant, il est plus que probable que les utilisateurs souhaitent pouvoir partager des fichiers au sein d'un même service ou d'un groupe d'utilisateurs, d'un projet particulier…

C'est pourquoi il est indispensable d'être capable de mettre en place de nouveaux partages répondant aux demandes des utilisateurs. Nous pouvons identifier 2 types principaux de demandes, qui sont détaillés ci-après.

3.3.2.3.1 Partage réservé aux utilisateurs d'un même groupe

Le groupe peut être dans ce cas un service au sein d'une entreprise ou un groupe travaillant sur un projet commun par exemple. Dans ce cas, on considère que les utilisateurs font partis ont un groupe UNIX en commun, qui leur est propre.

Dans ce cas, on va masquer aux utilisateurs non autorisés l'affichage du répertoire (`browseable = no`) et on va autoriser les utilisateurs de ce groupe commun (par exemple `compta`) et focer les utilisateurs à utiliser ce groupe lorsqu'ils travaillent dans ce partage (force group).

Nous obtenons la section suivante à ajouter dans `smb.conf` (et à personnaliser) :

```
[Comptabilite]
   comment = Fichiers communs du service comptabilite
   path = /chemin/a/personnaliser/cf/section/suivante
   read only = no
   writeable = yes
   create mask = 0660
   directory mask = 0770
   force group = compta
   valid users = @compta
```

Ainsi, tous les utilisateurs du groupe UNIX compta pourront voir ce partage, s'y connecter, y écrire et partager des documents au sein du service.

3.3.2.3.2 Partage de documents avec plusieurs groupes

Dans ce cas, il y a une personne ou un groupe de personne (par exemple le service de communication de l'entreprise) qui souhaite mettre à disposition de tous des documents. Dans ce cas, il faut que les membres du groupe `communication` puissent mettre des fichiers dans le partage et qu'un certain nombre d'autres groupes puisse y accéder.

Dans ce cas, il convient de donner les autorisations suivantes :

Nous obtenons la section suivante à ajouter dans smb.conf (et à personnaliser) :

```
[CommInt]
   comment = Communication Interne de documents
   path = /chemin/a/personnaliser/cf/section/suivante
```

```
create mask = 0660
directory mask = 0770
force group = communication
write list = @communication
read list = @group1 @group2 …
```

A l'aide des deux configurations types décrites ci-dessus, l'administrateur de la passerelle devrait pouvoir contenter toute demande d'utilisateur, dans les limites matérielles fixées par l'espace disque, bien entendu.

3.4 Gestion des utilisateurs et groupes UNIX

Cependant, les configurations décrites ci-dessus ne seront pleinement fonctionnelles que si les comptes, groupes et droits UNIX correspondant ont été correctement configurés. En effet, en plus d'appliquer les restrictions décrites dans le fichier `smb.conf`, samba limite les droits en fonction des droits de chaque groupe et utilisateur.

3.4.1 Construction de l'arborescence des répertoires

Comme nous l'avons vu dans les sections précédentes, les fichiers utilisateurs seront stockés dans `/mnt/userdata/`. Afin d'avoir une arborescence aussi classée que possible, nous allons créer les répertoires `shared` et `users`. Ceux-ci seront créés à l'aide des commandes :

```
mkdir /mnt/userdata/shared

chown root:root /mnt/userdata/shared

chmod 755 /mnt/userdata/shared

mkdir /mnt/userdata/users

chown root:root /mnt/userdata/users

chmod 755 /mnt/userdata/users
```

Nous créons également un répertoire spécifique permettant de sauvegarder les fichiers de configuration du serveur samba. Celui-ci s'appellera `saves` et est créé comme suit :

```
mkdir /mnt/userdata/saves

chown root:root /mnt/userdata/saves

chmod 700 /mnt/userdata/saves
```

Ainsi, nous stockeront dans shared les répertoires partagés entre plusieurs utilisateur, et ils porteront le nom du partage Samba auquel ils correspondent. Dans users, nous placeront les répertoires propres à chaque utilisateur, en nommant le répertoire du nom du login de l'utilisateur.

3.4.2 Procédure de création d'un groupe

Pour créer un groupe, nous avons utilisé la commande groupadd comme suit :

```
groupadd nom_du_groupe
```

Pour ajouter une personne à ce groupe, s'il s'agit d'un groupe secondaire, il suffit d'utiliser `vigr` et d'ajouter le login de l'utilisateur à la suite du dernier : ou, s'il y a déjà d'autres utilisateurs dans le groupe, de laisser un espace et ajouter le login en fin de ligne.

Lorsque vous effectuez ce type de modification, il faut penser à vérifier que le format du fichier de groupes n'est pas corrompu grâce à la commande `grpck`.

S'il s'agit du groupe principal d'un utilisateur, il faut modifier le fichier `/etc/passwd` grâce à `vipw`. Dans ce cas, il faut indiquer le numéro du groupe (que vous pouvez trouver grâce à `vigr`) et l'indiquer dans le quatrième champ de la ligne correspondant à l'utilisateur.

Si vous souhaitez affecter un groupe lors de la création d'un compte (ce qui est obligatoire) alors vous devrez suivre la procédure décrite dans la section suivante.

Par mesure de simplicité et afin de pouvoir regrouper tous les utilisateurs Samba dans un groupe commun pour pouvoir donner des droits de consultation globaux, nous avons mis tous les utilisateurs dans un nouveau groupe `sambausers`, puis les autres groupes d'appartenance seront définis dans le fichier `group`. Ceci évite, entre autre, de modifier manuellement le fichier `passwd`.

De plus, si le groupe possède un répertoire partagé, il faut modifier le fichier `smb.conf` en conséquence (cf. ci-dessus) et donner les droits d'accès conséquents.

Dans le cas d'un répertoire partagé uniquement entre membres du groupe :

```
mkdir /mnt/userdata/shared/sharename

chown root:groupname /mnt/userdata/shared/sharename

chmod 770 /mnt/userdata/shared/sharename
```

Dans le cas d'un repertoire partagé entre un groupe pour l'écriture et tout le monde pour la lecture, seuls changent les droits d'accès:

```
chmod 775 /mnt/userdata/shared/sharename
```

3.4.3 Procédure d'ajout d'un utilisateur

Pour ajouter un utilisateur, il faut tout d'abord ajouter son compte au système UNIX comme suit :

```
useradd -g sambausers -G othergroup1,othergroup2 -d
/mnt/userdata/users/sonlogin -c "Nom complet de la personne" -p "mot2passe"
sonlogin
```

Une fois l'utilisateur ajouté et son mot de passe définit, il faut synchroniser le mot de passe Samba avec le mot de passe UNIX. Pour cela, il faut utiliser l'utilitaire `smbpasswd`.

```
smbpasswd -a sonlogin
```

Nous avons ensuite dû saisir deux fois le mot de passe de l'utilisateur pour que celui-ci soit enregistré. Le switch `-a` ne doit être utilisé que lors de l'ajout d'un utilisateur. En cas de modification d'un mot de passe, il doit être omis.

Une fois les mots de passe configurés, pour que l'utilisateur puisse avoir accès à son répertoire home, il faut le créer. Cela s'effectue à l'aide des commandes :

```
mkdir /mnt/userdata/users/sonlogin

chown sonlogin:sambausers /mnt/userdata/users/sonlogin

chmod 700 /mnt/userdata/users/sonlogin
```

De cette façon, au niveau des droits UNIX, l'utilisateur aura les pleins pouvoirs sur son espace de stockage et personne d'autre ne sera autorisé à y accéder.

Remarque : Dans la mesure où les deux serveurs n'utilisent pas de serveur NIS commun, les numéros de compte utilisateurs du serveur Samba ne correspondront pas aux numéros du serveur NFS.

C'est pourquoi les utilisateurs doivent toujours accéder au NFS via la passerelle Samba, afin que leurs droits et restrictions soient toujours respectés.

3.4.4 Gestion des quotas utilisateurs

Afin d'éviter aux utilisateurs de saturer les espaces disques qui leur sont alloués, il est possible d'implémenter une stratégie de quota. Pour cela, nous avons installé l'outil quota sous debian à l'aide de la commande :

```
apt-get quota
```

Pour que celui-ci fonctionne, il faut également lancer quota_v2 grâce à `modprobe quota_v2`, puis rajouter une entrée `quota_v2` dans le fichier `/etc/modules`.

Il faut ensuite s'intéresser à la commande `setquota`, ce que nous n'avons pas eu le temps de faire mais qui pourrait s'avérer utile en cas de déploiement in-situ.

3.5 Sauvegardes et sécurité

Dans la mesure où nous ne disposons pas d'un équipement nous permettant de faire du RAID (ni d'assez de disque pour faire un RAID logiciel), nous programmons une entrée dans la `crontab` afin que le système sauvegarde automatiquement sa configuration tous les vendredi matin à une heure.

Pour ce faire, nous ajoutons la ligne suivante dans la `crontab` :

```
0 0     *  *   6  root   rm –rf /mnt/userddata/saves/etc.cpio
0 1     *  *   4  root   cpio –oc < /mnt/etc > /mnt/userdata/saves/etc.cpio
```

4 Les clients

4.1 Accès depuis un client Windows

L'accès depuis un client Windows peut se faire depuis l'explorateur réseau, la procédure étant plus rapide si les clients Windows appartiennent au même groupe de travail que la passerelle Samba.

En ligne de commande, il est possible de lister les ressources partagées par un serveur Samba à l'aide de la commande :

net view \\alfred\ /domain:workgroup \\nomnetbiosserveur

On peut ensuite monter un dossier en tant que disque réseau à l'aide de la commande :

```
net use [lettre_lecteur]: /Domain :workgroup \\nomnetbios\ressource [passwd]
/U:loginutilisateur
```

Si le login d'utilisateur est le nom de l'utilisateur Windows courant, alors le mot de passe de l'utilisateur courant sera envoyé et il n'y aura pas besoin de le préciser dans la ligne de commande. Dans les autres cas, il faudra le préciser.

On peut donc ainsi programmer le montage automatique d'un disque réseau en plaçant cette ligne de commande dans un fichier C:\samba.bat ou C:\samba.cmd (Windows 2000/XP) qui pourra être lancé au démarrage en créant une clé 'sambamount' dans la base de registre sous la clé `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` et y placer en valeur C:\nomdufichier.ext.

5 Améliorations et tolérance aux pannes

Afin d'améliorer la sécurité des serveurs et d'implémenter une tolérance de pannes, il pourrait être intéressant de s'intéresser aux éléments suivants.

Tout d'abord, l'arrêt de tous les services réseaux autres que le NFS sur le serveur de données, à l'exception du ssh qui doit, dans ce cas, être configuré pour n'autoriser lui-même les connexions que de l'administrateur réseau, qui possèdera une clé lui permettant de s'authentifier.

Ensuite, il n'est pas forcément nécessaire de désactiver le ssh sur le server Samba, dans la mesure où les droits utilisateurs seront correctement gérés. Mais il pourrait être intéressant d'effectuer la même limitation que celle citée ci-dessus pour éviter de surcharger le serveur d'autres requêtes que des requêtes NetBIOS.

De plus, il serait intéressant de mettre en place un système RAID 0 pour les disques systèmes des deux serveurs afin de garantir une disponibilité maximale et un temps de rétablissement rapide en cas de perte d'un disque dur.

Le disque de données utilisateurs pourrait également être monté en RAID0 mais cela n'est pas forcément indispensable. La fréquence des sauvegardes pourrait en revanche être augmentée, en utilisant une sauvegarde incrémentale afin de ne pas surcharger les lecteurs de bandes et de ne pas consommer trop d'espace de stockage physique pour les bandes sauvegardées (armoire, entrepôts…).

De plus, au niveau du serveur Samba, les fonctionnalités du démon vont beaucoup plus loin que celles que nous avons exploitées : elles permettent aussi le partage d'imprimante en réseau, ainsi que la gestion d'un annuaire LDAP afin de simuler un Domaine Microsoft Windows équipé d'un ActiveDirectory. Il y a donc de nombreuses autres options du démon auquel on aurait pu s'intéresser mais qui dépassaient le cadre de ce projet d'administration système.

Enfin, concernant le serveur NFS, il est possible de réaliser une liaison NFS over SSL et ainsi sécuriser le traffic de données sur le réseau. Ceci est surtout important si les utilisateurs ont la possibilité de se loger sur certaines machines connectées au réseau en tant qu'administrateur, car ils pourraient utiliser des applications spécifiques (comme ethereal) pour récupérer les flux réseaux transitant entre la passerelle Samba et le serveur de fichiers.

Conclusion

Ce projet nous a beaucoup appris sur le fonctionnement de Samba et de NFS et nous a permis de voir à quel point il est facile de déployer rapidement un service compatible entre les ordinateurs Windows et les stations Unix. Samba permet de simuler de nombreuses fonctionnalités d'un serveur Windows Server tout en bénéficiant de la stabilité du système Linux.

Ce projet pourrait nous être utile à l'avenir pour déployer, à moindre coût, une solution de compatibilité entre Windows et Linux en entreprise.