Introduction à Samba

Guillaume Valadon (doug@sieste.org)

25 juin 2000

Table des matières

1 Qu'est-ce que Samba ? 2
- 1.1 Les débuts . . . 3
- 1.2 Concepts du protocole SMB . . . 5
- 1.3 Browsing . . . 5
- 1.4 Authentification . . . 6

2 Installation 7

3 Présentation du fichier smb.conf 9

4 Première configuration 11

5 Exemple d'utilisation 13
- 5.1 Création des partages . . . 13
- 5.2 Impression . . . 15
- 5.3 Sous-réseaux . . . 16
- 5.4 Paramètres remarquables . . . 16

6 Les commandes fournies par Samba 18
- 6.1 smbclient . . . 18
- 6.2 nmblookup . . . 19
- 6.3 smbwrapper & smbmount/smbumount . . . 20
- 6.4 smbstatus . . . 21
- 6.5 testparm . . . 22

7 Outils de configuration de Samba 23

8 Conclusion 24

Bibliographie 24

Chapitre 1

Qu'est-ce que Samba ?

D'après le petit Larousse, il s'agit d'une danse brésilienne à deux temps, mais ce n'est pas de cela dont je vais parler. Samba est un ensemble d'applications utilisant le protocole SMB (Server Message Block), appelé aussi CIFS (Common Internet File System). Disponible sur de nombreux systèmes d'exploitation dont, parmis les plus connus :

AIX
Amiga
FreeBSD
HP-UX
Irix
NetBSD
OS/2
QNX
Solaris/SUNOS

Malgré la quarantaine d'OS supportés, Linux et SUNOS ont été les mieux testés.

Samba fait rentrer les machines UNIX dans le monde Windows, en leur permettant aussi bien de fournir des partages de fichiers et d'imprimantes, que d'y accèder. Il est maintenu par un groupe de développeurs, la Samba Team, et est distribué sous License GPL, ce qui est un énorme avantage en comparaison des prix des licenses de Windows NT Server. Ainsi, il n'y a plus besoin de se préoccuper des couteuses licences par poste ou par siège, un seul serveur Samba gère autant de connexions et de postes que l'on souhaite.

Le mot *partage* est souvent employé dans un réseau de machines Windows. Il désigne le répertoire, ou l'imprimante auquel on accéde via le réseau. Sous Windows, les partages d'une machine sont visibles grâce au *Voisinage Réseau*, et on peut facilement les associer à une lettre de lecteur classique ce qui rend transparentes leurs utilisations.

1.1 Les débuts

Le protocole NetBIOS fit sont apparition en 1984, puis vint NetBEUI en 1985, tous deux développés par IBM. Leur fonctionnement sur TCP/IP est décrit par les RFC[1] 1001 (concepts) et 1002 (protocole). SMB fut quant à lui développé en partenariat entre Microsoft et Intel en 1987, et se place au dessus de NetBIOS et NetBEUI. Il s'agit d'un protocole orienté connexion travaillant sur un mode question réponse, suivant une architecture client/serveur. Plus clairement, le client pose une question, le serveur y répond.

NetBIOS sur TCP/IP offre trois services distincts : la gestion des noms (enregistrement, et résolution), un échange de données en mode connecté et un autre en mode non connecté.

Le système de gestion des noms permet de faire la relation entre les adresses IP des machines, et leurs noms NetBIOS. Ceux-ci ne sont pas suffixés par un domaine comme dans le monde IP, et sont limités à quinze caractères. On peut utiliser les caractères suivants a-z, A-Z, 0-9, ! @ # % ^ & ( ) - ' { } ~. Cependant l'utilisation du "." n'est pas recommandée car cela peut poser des problèmes dans les futures versions de NetBIOS sur TCP/IP.

Un seizième caractère permet de déterminer le type de ressources associé au nom. Parmi les plus utilisées, on a :

Valeur	Description
00	Nom NetBIOS
03	Service de messagerie
1b	Domain Master Browser
20	Serveur de fichier (et d'impression)

Ce dernier octet permet aussi de connaître le type de ressources des groupes :

Valeur	Description
1c	Nom du domaine
1d	Nom du Master Browser
1e	Nom du group utilisé dans l'élection du master browser

Ainsi en tapant :

```
c :\>nbtstat -a quicky

        NetBIOS Remote Machine Name Table

   Name               Type         Status
---------------------------------------------
QUICKY          <00>            UNIQUE     Registered
SMB             <00>            GROUP      Registered
QUICKY          <03>            UNIQUE     Registered
```

[1] On peut les récuperer sur `ftp ://ftp.ripe.net/rfc`

QUICKY	<20>	UNIQUE	Registered
SMB	<1E>	GROUP	Registered
BENZ	<03>	UNIQUE	Registered
SMB	<1D>	UNIQUE	Registered
..__MSBROWSE__	<01>	GROUP	Registered

MAC Address = 00-A0-65-00-02-EF

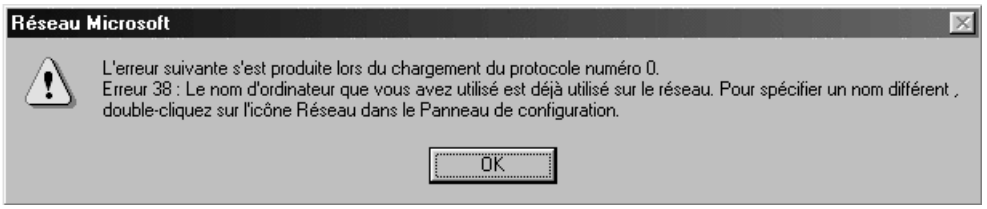
On peut voir que la machine quicky peut recevoir des messages pour l'utilisateur benz, et fait partie du groupe SMB.

Au niveau du système d'exploitation un partage est simplement identifié grâce à des adresses UNC (Universal Naming Convention) :

```
//machine/partage
```

A partir de l'UNC, la machine va pouvoir trouver l'IP de la machine et effectuer la connexion au partage.

Lors du démarrage, une machine va annoncer son nom grâce à un broadcast, si aucune autre machine ne possède le même, la machine le conserve, sinon comme on peut le voir ici, le système d'exploitation demande à ce qu'il soit changé.



Ce mécanisme d'enregistrement du nom peut également être effectué par un serveur NBNS (NetBIOS Name Server) - l'implémentation de ce système par Microsoft s'appele WINS (Windows Internet Name Service). Les machines s'adresseront alors à ce serveur pour savoir si un nom est disponible ou pas. La résolution des noms NetBIOS (le passage du nom à l'IP) peut quant à elle être effectué soit par broadcast soit par l'intermédiaire d'un serveur WINS.

Les serveurs WINS permettent d'éliminer les broadcasts, on va ainsi pouvoir utiliser SMB à travers des routeurs.

On obtient différents types d'enregistrement et de résolution de noms appelés :

b-node : broacast

p-node : serveur WINS

h-node : serveur WINS, si celui-ci ne marche pas, on se sert du broadcast

1.2 Concepts du protocole SMB

Le protocle SMB fonctionne au dessus de NetBIOS, fonctionant lui-même sur TCP/IP. Samba n'implémentant que SMB sur TCP/IP, les échanges de données sont effectués en mode connecté.

Le tableau 1.1 nous montre où se situe SMB par rapport au modèle OSI :

TAB. 1.1 – Pile de protocoles

<table>
<tr><td colspan="3">SMB</td></tr>
<tr><td>NetBIOS</td><td rowspan="2">NetBEUI</td><td>NetBIOS</td></tr>
<tr><td rowspan="2">IPX</td><td>TCP & UDP</td></tr>
<tr><td>IP</td></tr>
<tr><td colspan="2">802.2,
802.3,802.5</td><td>Ethernet</td></tr>
</table>

Il existe de nombreuses variantes de SMB, parmi les opérations minimales qu'elles doivent pouvoir effectuer sur des fichiers et imprimantes partagés, citons :

connection et déconnexion

ouverture et fermeture

lecture et écriture

1.3 Browsing

Pour connaître tous les serveurs d'un domaine, une machine s'adressera au *master browser* qui lui renverra une *browse list*, c'est ce que l'on appelle *browsing* en anglais. Cette liste a été construite à partir des *server annoucements* et *domain annoucements* faits par tous les serveurs et les *domain master browsers*[2] sur le réseau. Ceci permet un gain de temps et de ressources au niveau des machines clientes qui n'auront pas à reconstruire la *browse list* au fur et à mesure des changements.

Une élection a lieu entre les machines du réseau et permet de déterminer laquelle sera *master browser*. Cette tâche ne pouvant être effectuée par toutes les machines pour des raisons de puissance, on peut demander à Samba de le devenir en falsifiant l'élection. Il mentira en annonçant un numéro de système d'exploitation supérieur à celui de NT Server (0x20) par exemple. Cela est utile dans le cas de l'utilisation de Samba à travers des réseaux routés où l'on va avoir besoin de synchroniser les *master browser* des différents sous-réseaux.

Comme nous l'avons déja vu, la gestion des noms peut-être assurée par un serveur WINS. Lorsqu'une machine arrive sur le réseau, elle envoie son nom NetBIOS, son IP, et son groupe de travail au serveur WINS. Ces informations vont être par la suite automatiquement mises à jour. Il n'y a pas d'élection la

[2] *master browser* d'autres domaines du réseau

fonction *serveur WINS* est statique, elle est déterminée au moment de la mise en place de Samba.

1.4 Authentification

Un réseau de machines Windows comprend deux méthodes d'identification des utilisateurs, par :

groupe de travail (workgroup) : l'authentification est distribuée à l'ensemble des machines du réseau, chaque machine gère sa propre base d'utilisateurs

domaine : l'authentification est assurée par une seule machine pour l'ensemble des partages du réseau

Chapitre 2

Installation

Les sources Samba sont disponibles sur le serveur ftp.samba.org et ses miroirs. Il convient tout d'abord de les décompresser par

```
tar xzvf samba-x.x.x.tar.gz
```

Le répertoire crée s'appelle **samba-x.x.x** et contient de nombreux documents et exemples.

Rendons nous dans **samba-x.x.x/source** où nous allons lancer l'installation par

```
./configure
make
make install
```

Sous un système Linux 2.2.x ou supérieur, il est conseillé d'utiliser :

```
./configure --smbmount
```

Samba est désormais installé dans le répertoire **/usr/local/samba** suivant l'arborescence suivante :

bin/ contient les serveurs `smbd` et `nmbd` ainsi que d'autres utilitaires

lib/ contient les fichiers smb.conf et lmhost

man/ les manuels des commandes fournies par Samba

swat/ l'outil d'administration de Samba

var/ logs d'activité des démons, et informations nécessaires à leur fonctionnement

private/ contient entre autres le fichier smbpasswd

`smbd` fournit les partages et les services d'impression aux clients Windows, `nmbd` assure la résolution des noms et la fonction de serveur WINS. En plus de ces serveurs, de nombreux utilitaires sont fournis, citons par exemple `smbclient` qui offre une façon simple d'accéder aux serveurs SMB en ligne de commande à la manière de ftp, `smbwrapper` permettant d'utiliser des partages comme un

système de fichier classique, et `nmblookup` grâce auquel on retrouve l'adresse IP d'une machine à partir de son nom NetBIOS.

Chapitre 3

Présentation du fichier smb.conf

Le fichier smb.conf est le fichier de configuration de Samba. Toutes les sections (sauf `[global]`) représentent un partage, et sont définies par un mot entre crochets (ex : `[toto]`). Plus de deux cents paramètres permettent de définir leur comportement, et sont de la forme `nom = valeur`, où casse et nombre d'espaces ne sont pas pris en compte.

Ce fichier étant lu de haut en bas, une section s'arrête là où une autre commence, des paramètres de la section `[global]` peuvent être recouverts par ceux d'une autre section. Remarquons également que certains paramètres ne peuvent apparaître que dans `[global]`. Les lignes commençant par # ou ; seront interprétées comme des commentaires.

Il existe trois sections spéciales : `[global]`, `[homes]`, et `[printers]`.

La section `[global]` définit le fonctionnement général de Samba (`smbd` et `nmbd`). On peut y définir sa politique d'accès, les interfaces qu'il doit écouter, la façon dont il va trouver les adresses IP associées à un nom NetBIOS particulier, etc ...

Les sections `[homes]` et `[printers]` sont particulières. Elles vont respectivement engendrer la création à la volée du partage personnel de l'utilisateur se connectant au serveur Samba d'après /etc/passwd, et la création des imprimantes que sert le serveur d'après le fichier /etc/printcap. Elles permettent donc de ne pas créer explicitement une entrée par utilisateur et par imprimante dans le fichier de configuration.

Ainsi partager **/tmp** reviendrait à déclarer une section `[tmp]` dans smb.conf, et à fournir un paramètre de type `path` afin que le partage pointe vers un répertoire existant sur le système local :

```
[tmp]
    path = /tmp
```

Pour y permettre l'écriture, on ajoutera simplement :

```
writeable = Yes
```

Il existe un ensemble de variables que l'on peut placer dans toutes les directives qui acceptent des chaînes de caractères en argument.
Voici les plus utiles :

%u : l'utilisateur courant

%g : le groupe de %u

%U : l'utilisateur demandé (celui que le client fournit, peut être différent de %u si on utilise 'force user')

%H : le répertoire de connexion de %u

%S : le partage en cours

%m : le nom Netbios de la machine cliente

%L : le nom Netbios du serveur

%I : l'adresse IP du client

%v : la version de Samba

Elles permettent de définir des contextes particuliers dans les partages, ainsi on pourrait loguer les événements par client et non plus globalement avec le paramètre suivant dans la section `[global]`

```
log file = /usr/local/samba/var/smbd.%m
```

De nombreux paramètres seront utilisés avec leurs valeurs par défaut :

```
[homes]
    writeable = Yes
```

Ces deux lignes dans smb.conf permettent de créer un partage en écriture des répertoires de connexion des utilisateurs de la machine. Le nom NetBIOS du serveur SMB sera le nom de la machine dans le monde IP, et le path aura pour valeur le répertoire de connexion de l'utilisateur.

La relecture du fichier smb.conf est effectuée toutes les soixante secondes par `smbd` et `nmbd`, on peut également la forcer en leur envoyant un signal HUP. Les modifications ne sont pourtant pas prises en compte immédiatement afin de pas arrêter les connexions en cours.

Lorsque qu'une machine cliente cherche à utiliser un *partage*, celui-ci est recherché dans smb.conf en commençant par `[partage]`, `[homes]`, puis `[printers]`. Si aucun partage n'est trouvé, le partage par défaut est utilisé, sinon une erreur est renvoyée.

Chapitre 4

Première configuration

Créeons simplement le fichier smb.conf suivant :

```
[global]
    workgroup = SMB
    netbios name = SAMBA

[homes]
    comment = Repertoire personnel de %u
    path = /home/%u
    read only = No # équivalent à writable = yes
    browseable = No
```

On s'aperçoit que depuis un poste Windows l'accès à la machine nous est refusé, le mot de passe n'étant pas bon. Ceci est dû au comportement par défaut de celui-ci : il s'attend à recevoir des mots de passe sans aucun chiffrement. Windows NT (à partir du SP3), Windows 95 (mis à jour avec VRDRUPD.EXE), et Windows 98 utilisent quant à eux des mots de passe chiffrés.

Pour résoudre ce problème, deux solutions s'offrent à nous.

La première consiste à modifier la base de registre de Windows pour qu'il ne chiffre plus les mots de passe mais ceci est assez hasardeux et peu souhaitable si l'on cherche un tant soit peu à sécuriser les accès.

La seconde est celle qui doit être retenue, et consiste à demander poliment à Samba d'utiliser des mots de passe chiffrés par l'ajout de la ligne suivante dans smb.conf :

```
encrypt passwords = yes
```

Le chiffrement utilisé est celui du protocole SMB et non plus celui du monde UNIX, il convient donc de créer un fichier d'utilisateur supplémentaire appelé smbpassd qui ne sert qu'à la connexion au serveur.

Ceci est réalisé grâce à la commande smbpasswd :

```
$ smbpasswd -a toto
New SMB password :
Retype new SMB password :
Added user toto.
Password changed for user toto.
```

Ce nouveau fichier ne se substitue pas à /etc/password. C'est toujours grâce à lui que l'accès aux fichiers est réalisé : l'écriture et la lecture de fichiers contenus dans un partage sont dépendants des droits UNIX sur ceux-ci. C'est pour cela que l'utilisateur doit exister dans les deux fichiers.

Un problème peut cependant apparaitre lorsque l'on essaye de créer un compte UNIX avec un nom d'utilisateur Windows trop long (en général plus de 8 caractères). Il faut dans ce cas utiliser le paramètre `username map` de la section `[global]` qui prend en argument un nom de fichier contenant la relation nom UNIX <=> nom Windows.

Dans username.map on aura par exemple :

```
root = administrateur
```

Logons nous en tant que toto avec le mot de pass toto sur une station Windows 9x, et tapons :

```
C :\>net use t : \\smb\toto
Exécution achevée.
```

Le disque T : est desormais associé au répertoire personnel de l'utilisateur toto.

Chapitre 5

Exemple d'utilisation

On va étudier les paramètres intéressants de Samba à partir d'un cas réel. Prenons le cas d'une entreprise possédant trois services :

direction

technique

comptabilité

On va créer les trois partages correspondants ainsi que les répertoires personnels des utilisateurs de ces services.

5.1 Création des partages

On crée donc trois groupes UNIX avec `groupadd` : direction, compta et technique. Puis on crée les utilisateurs de ces 3 groupes par :

`adduser -d /shares/`*`groupe/utilisateur`* `-g` *`groupe utilisateur`*

```
[global]
    workgroup = entreprise
    netbios name = direction
    netbios aliases = compta technique
    server string = Serveur Samba
    interfaces = 127.0.0.1/24 192.168.0.1/24
    bind interfaces only = Yes
    encrypt passwords = Yes
    include = smb.conf.%L
```

`netbios aliases` permet à smbd d'avoir plusieurs noms NetBIOS

`interfaces` interfaces que Samba va écouter

`bind interfaces only` limite les connexions aux interfaces spécifiées dans `interfaces`

`include` est de loin le paramètre le plus intéressant de samba, il fonctionne comme le #include en C et permet d'insérer des smb.conf spécifiques par exemple au serveur demandé par l'intérmédiaire des variables %.

Ici on demande à Samba d'insérer un smb.conf relatif au serveur sur lequel on veut se connecter. Dans **/usr/local/samba/lib/**, on a donc :
smb.conf.compta

```
[homes]
    comment = Repertoire personnel de %u
    path = /shares/compta/%u
    read only = No
    browseable = No
    valid users = +compta
[compta]
    path = /shares/compta/
    browseable = Yes
    read only = No
    valid users = +compta
```

On fait de même pour direction et technique.

On a donc trois serveurs dans le groupe de travail entreprise, possédant chacun des partages dont l'accès est limité aux utilisateurs de ces services grâce à `valid users`. Le + devant le nom compta signifie ici le groupe compta : tous les utilisateurs de ce groupe auront ainsi accès à `[homes]`.

L'entreprise grandissant, on décide d'embaucher une secrétaire, celle-ci n'aura accès qu'au partage secretariat des 3 sections.

On crée l'utilisateur et un groupe secret .

On ajoute au fichier smb.conf.compta :

```
[secretariat]
    path = /shares/secretariat/%L
    browseable = Yes
    valid users = +compta secret
    read only = yes
    write list = secret
```

On fait de même pour direction et technique.

Seule la secrétaire pourra écrire dans `[secretariat]`, les utilisateurs du service pourront seulement y lire. `write list` permet de spécifier la liste des utilisateurs pouvant écrire dans le partage.

A ce stade, il convient de constater que les permissions Samba s'appliquent avant celles du système d'exploitation. Ainsi, si l'on a un partage avec `read only`

= No de positionné mais que l'utilisateur ne possède pas les droits d'écriture sur le répertoire associé, celui-ci ne pourra pas y écrire.

Il faut donc qu'il y ait cohésion entre les droits accordés par Samba et par l'OS.

Pour permettre à nos utilisateurs de sauvegarder temporairement leurs données on crée dans smb.conf :

```
[tmp]
    comment = Partage temporaire
    path = /tmp
    guest ok = Yes
```

`guest ok` permet à tout le monde de se connecter à ce partage, aucun mot de passe ne sera demandé.

On décide maintenant de partager un lecteur de cdrom on ajoute `[cdrom]` dans le smb.conf :

```
[cdrom]
    comment = Lecteur de cede
    path = /mnt/cdrom
    root preexec = mount /mnt/cdrom
    root postexec = umount /mnt/cdrom
```

Les paramètres `preexec` et `postexec` permettent d'executer des commandes à la connexion et à la déconnexion au partage. Le préfixe root indique qu'elles seront lancées sous le super utilisateur.

5.2 Impression

Notre entreprise veut maintenant se doter d'une imprimante.

L'impression d'un document nécessite trois étapes :

ouverture du fichier

écriture

fermeture

A ce stade, `smbd` se charge de l'impression.

Pour créer une imprimante via Samba, il suffit de créer le partage suivant dans smb.conf :

```
[imprimante]
    printable = yes
    printer = lp
```

```
path = /var/spool/lpd
```

Une fois le fichier écrit dans **/var/spool/lpd** celui sera imprimé grâce à `lp`.

Comme déja vu lors de la présentation de smb.conf, on peut grâce à `[printers]`, créer des imprimantes à la volée selon /etc/printcap. Le paramètre `load printers = yes` dans `[global]` permet de les rendre visibles.

5.3 Sous-réseaux

La société s'agrandit et est désormais présente dans deux lieux géographiques différents reliés entre eux grâce à l'Internet. SMB utilisant le broadcast pour la résolution des noms et l'annoncement des machines sur le réseau, on ne va pas pouvoir l'utiliser tel quel dans cet environnement routé.

On va donc utiliser des serveurs WINS, cependant passer sur tous les postes clients pour configurer ce système est fastidieux, nous allons donc demander à Samba d'enregistrer les annoncements et de répondre aux requêtes WINS grâce à `wins proxy`. On va aussi configurer les deux `smbd` pour être `domain master browser` et `local master browser`.

Ajoutons les paramètres suivants dans la section `[global]` du sous-réseau A :

```
domain master browser = Yes
remote browse sync = IP du serveur Samba sur le sous-réseau B
wins support = Yes
wins server = IP du serveur Samba sur le sous-réseau B
```

Et dans pour le sous-réseau B :

```
local master = Yes
prefered master = Yes
os level = 99
wins support = Yes
wins server = IP du seveur Samba sur le sous-réseau A
```

`remote browse sync` va permettre la synchornisation de la browse list en faisant appel au master browser du sous-réseau B

`os level` demande à `smbd` de mentir et ainsi devenir `master browser` du sous-réseau B

5.4 Paramètres remarquables

Un certain nombre de paramètres n'ont pas été abordés dans les sections précédentes mais demeurent néanmoins intéressants.

paramètre	signification
available	active ou désactive un partage
copy	clone le service donné en arguments dans le service courant
create mask/mode	permissions UNIX avec lesquelles sera crée un fichier
allow/deny hosts	machines autorisées ou non à utiliser les partages
message command	commande à lancer à la reception d'un message

Si l'on a installé Samba en utilisant `./configure --with-syslog`, on peut grâce aux paramètres `syslog *` enregistrer tous les messages en passant par `syslogd`.

Les options `printer driver *` permettent d'automatiser la configuration de l'imprimante sur les postes Windows.

Chapitre 6

Les commandes fournies par Samba

Ce chapitre est une présentation des commandes offertes avec Samba qui sont les plus utiles.

6.1 smbclient

Elle permet d'accèder à des partages en ligne de commande comme avec un client ftp classique. On peut également s'en servir pour envoyer des messages, lister les partages d'une machine, ou bien imprimer un fichier.

Lister les partages :

```
$ smbclient -L //samba -N
added interface ip=192.168.0.2 bcast=192.168.0.255 nmask=255.255.255.0
Domain=[SMB] OS=[Unix] Server=[Samba 2.0.6]

        Sharename    Type    Comment
        ---------    -----   --------
        tmp          Disk    Partage temporaire
        cdrom        Disk    Lecteur de cede
        IPC$         IPC     IPC Service (Serveur Samba)

        Server               Comment
        ---------            --------
        SAMBA                Serveur Samba

        Workgroup            Master
        ---------            --------
        SMB
```

-N : smbclient ne demandera pas de mot de passe

Lister les partages sur une machine distante en utilisant un serveur WINS particulier :

```
$ smbclient -s pouet.conf -L toto -N
```

où pouet.conf contient :

```
[global]
    wins server = 192.168.0.1
    name resolve order = wins
```

Envoyer un message :

```
$ smbclient -M machine
test d'envoi de message
```

Options remarquables :

-R *ordre des moyens de résolution de noms* : utiliser seulement ces types de résolutions

-I *IP destination* : se connecter à cette IP

-U *nom d'utilisateur* : utiliser cet utilisateur

-D *répertoire* : commencer depuis ce répertoire

6.2 nmblookup

Cette commande permet de trouver l'IP d'une machine à partir de son nom

```
$ nmblookup samba
querying samba on 192.168.0.255
192.168.0.1 samba<00>
```

options remarquables :

-M : rechercher le master browser

-S : lister les ressources après que l'Ip ait été retrouvée

-a *adresse IP* : lister les ressources

-R :

-B *adresse IP de broadcast* : effectuer la résolution de nom via cette IP

-U *adresse IP du serveur WINS* : idem mais vers un serveur WINS

```
$ nmblookup -A 192.168.0.1
Looking up status of 192.168.0.1
received 12 names
       SAMBA           <00> -               M <ACTIVE>
       SAMBA           <03> -               M <ACTIVE>
```

```
       SAMBA           <20> -                M <ACTIVE>
       COMPTA          <00> -                M <ACTIVE>
       COMPTA          <03> -                M <ACTIVE>
       COMPTA          <20> -                M <ACTIVE>
       SMB             <00> - <GROUP>        M <ACTIVE>
       SMB             <1b> -                M <ACTIVE>
       SMB             <1e> - <GROUP>        M <ACTIVE>
       TECHNIQUE       <00> -                M <ACTIVE>
       TECHNIQUE       <03> -                M <ACTIVE>
       TECHNIQUE       <20> -                M <ACTIVE>
num_good_sends=0 num_good_receives=0

la ligne :
       SAMBA           <00> -                M <ACTIVE>
```

Nous donne le nom NetBIOS associée à cette IP

6.3 smbwrapper & smbmount/smbumount

Ces commandes permettent de rendre transparente l'utilisation d'un partage. En effet, elles donnent l'impression que celui-ci fait partie du système de fichier.

Pout utiliser smbwrapper, tapons :

```
$ smbsh
Username : doug
Password :
```

Puis pour lister le contenu du répertoire personnel de l'utilisateur :

```
$ ls /smb/samba/doug
```

Où **/smb** n'existe pas sur le système, Samba étant le nom de la machine sur laquelle on souhaite accéder au partage doug.

Toutes les commandes habituelles sur les fichiers deviennent valides cependant, on ne peut pas éxecuter de programmes depuis le partage.

La solution de choix sur un système Linux 2.2.x est l'utilisation du module smbfs, on va ainsi pouvoir monter et démonter des partages comme on l'aurait fait avec des disques durs.

```
# smbmount //samba/benz /home/doug/samba/ -o username=benz,uid=doug,gid=doug
```

On peut voir qu'avec l'option -o on peut passer un certain nombre de paramètres à smbmount, en lui demandant notamment de donner les droits sur le point de montage ainsi crée à tel ou tel utilisateur.

```
# mount
/dev/sdb1 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/sdb6 on /home type ext2 (rw)
/dev/sdb7 on /softs type ext2 (rw)
/dev/scd0 on /mnt/cdrom type iso9660 (ro)
//samba/benz on /home/doug/samba type smbfs (0)
```

Comme on peut le voir ici, le partage est bien considéré comme faisant partie du système de fichier.

On démontera le système de fichier de la façon suivante :

```
# smbumount /home/doug/samba/
```

Il est intéressant de positionner le bit setuid à un sur smbmount et smbumount pour permettre aux utilisateurs d'accéder aux partages.

En utilisant smbfs, on arrive à éxecuter des programmes résidant sur le partage sur notre machine.
smbtar est un script utilisant smbclient et permettant de sauvegarder des données stockées sur un système distant.

Ainsi :

```
$ smbtar -s samba -u benz -x benz -p mot de passe -t home.tar
```

Avec cette commande, j'ai sauvegardé le contenu du partage `-p` sur la machine `samba` dans le fichier `-t`. Il est à déplorer que le mot de passe ne soit pas demandé s'il n'est pas fourni en ligne de commande.

Le SMB HOWTO[1] montre comment effectuer périodiquement des sauvegardes de partages disants.

6.4 smbstatus

smbstatus permet de connaitre les connexions en cours avec le serveur `smbd`.

```
$ smbstatus
Samba version 2.0.6
Service  uid     gid     pid     machine
----------------------------------------------------------------------------------------
benz     benz    benz    343     quicky (192.168.0.2)    Fri Apr  7 16 :39 :52 2000

No locked files
```

[1] Disponible sur `http://www.freenix.org/linux`

```
Share mode memory usage (bytes) :
   1048464(99%) free + 56(0%) used + 56(0%) overhead = 1048576(100%) total
```

6.5 testparm

La commande que l'on se doit d'utiliser après chaque modification manuelle de smb.conf, car elle indique les fautes dans celui-ci.

Chapitre 7

Outils de configuration de Samba

Avant de les utiliser, il convient de bien connaître la syntaxe de smb.conf, et donc de bien avoir lu le fichier man de smb.conf.

Ksamba, et GnoSamba fonctionnent sous X-Window, SmbEdit sous Windows. Ils permettent de créer en quelques clics de souris des partages et de leur attribuer simplement un certains nombre de paramètres. Comanche, l'outil de configuration d'Apache permet depuis peu de paramétrer Samba.

SWAT et Webmin sont particulièrement intéressants car ils permettent la configuration à distance de Samba via un navigateur web. On notera que SWAT ne devient réellement utilisable que dans le contexte d'une connexion sécurisée par SSL.

Enfin, l'outil le plus utile est gsmb qui permet de faire la relation entre les fichiers /etc/passwd et smbpasswd à travers une interface très pratique.

Se les procurer :

Ksamba : `http://www.kneschke.de/projekte/ksamba`

GnoSamba : `http://se.samba.org/samba/GUI/`

Smbedit : `http://se.samba.org/samba/smbedit/intro.htm`

SWAT : fourni avec les sources de Samba

Webmin : `http://www.webmin.com/webmin/`

gsmb : `http://savage.iut-blagnac.fr/projets/developpement/gsmb/`

Comanche : `htpp://www.comanche.org`

Chapitre 8

Conclusion

A partir de ce document, on va pouvoir commencer à utiliser pleinement Samba. Cependant certains points n'ont pas été abordés et semblent prometteurs comme la fonction de Contrôleur Principal de Domaine, le partage de mots de passe via LDAP, et les connexions sécurisées via SSL.

Samba est simple d'utilisation, très documenté, et plus important encore bénéficie d'une équipe de développement très active. C'est donc un outil de choix pour remplacer des machines NT dans un environnement Windows, d'autant que la fonction PDC semble de mieux en mieux supportée.

Bibliographie

[1] Richard Sharpe, Gerald Carter
Sams Teach Yourself Samba in 24 Hours
Sams, 1999

[2] Robert Eckstein, David Collier-Brown, Peter Kelly
Using Samba
O'Reilly, 1999

[3] Richard Sharpe *Just what is SMB ?*
`http://anu.samba.org/cifs/docs/what-is-smb.html`
1999

[4] *RFC 1001 & 1002*
`ftp://ftp.ripe.net`
1987

[5] `http://www.cifs.com`

[6] `news.protocol.smb`

[7] Les documents de **samba-x.x.x/docs/**

[8] *SAMBA HOWTO*
Disponible sur `http://www.freenix.org/linux`