Partage de données avec SAMBA

Samba est une solution logicielle d'interopérabilité avec Windows disponible sur les systèmes Linux et Unix. Le nom de Samba vient du protocole SMB : Server Message Block utilisé pour le partage de ressources sur les réseaux Microsoft. Il permet notamment le partage de fichiers ou d'imprimantes sur des serveurs Linux à destination de clients Windows. La suite logicielle Samba comprend aussi un client qui permet aux machines Linux d'accéder à des ressources partagées sur un serveur Windows.

1. Configuration générale

a. Les démons samba

Samba repose sur deux démons appelés **nmbd** et **smbd**. L'annonce des services et en général tout le fonctionnement NetBIOS over IP repose sur le démon **nmbd**. Les partages de fichiers et d'imprimantes eux-mêmes s'appuient sur le démon **smbd**.

Le script de gestion du service généralement présent dans les distributions lance ces deux démons à chacun de ses démarrages.

b. Les fichiers de configuration

Les démons samba trouvent leur configuration dans le fichier de configuration **smb.conf**, généralement dans le répertoire **/etc/samba**.

Le fichier de configuration est divisé en sections normalisées, chacune étant commencée par un titre entouré de crochets. Les paramètres de fonctionnement seront dans chacune des sections présentés sous la syntaxe **paramètre = valeur**.

Format synthétique de smb.conf

```
[section1]
 paramètre1 = valeur1
 paramètre2 = valeur2
[section2]
 paramètre3 = valeur3
 paramètre4 = valeur4
```

Il existe un outil fort utile appelé **testparm** qui valide le format d'un fichier de configuration samba. Il renvoie en outre un état épuré (sans les lignes de commentaire) de la configuration sur la sortie standard. Naturellement, cette sortie pourra être redirigée vers un fichier et générer un **smb.conf** lisible et de taille raisonnable. Il est à noter que la commande **testparm** ignore tout paramètre du fichier de configuration s'il est configuré à sa valeur par défaut. Ce comportement peut être modifié avec l'option **-v**. Toutes les options applicables sont alors affichées.

Exemple d'exploitation de testparm pour génération d'un fichier smb.conf simple

Cette méthode est souvent utilisée pour utiliser un fichier de configuration largement commenté, et un fichier réel de dimension raisonnable.

```
alpha:/etc/samba# mv smb.conf big.smb.conf
alpha:/etc/samba# wc -l big.smb.conf
326 big.smb.conf
alpha:/etc/samba# testparm big.smb.conf > smb.conf
Load smb config files from big.smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

alpha:/etc/samba# wc -l smb.conf
31 smb.conf
alpha:/etc/samba# testparm -v big.smb.conf > toutes-options.info.smb.conf
```

```
alpha:/etc/samba#
```

Les versions pré-installées de samba proposent toujours un fichier **smb.conf** pré-configuré. Si ce fichier peut constituer une bonne base de départ, sa taille (326 lignes chez Debian) risque d'impressionner les débutants, et on gagnera peut-être à réaliser un fichier de toutes pièces, avec uniquement les éléments dont on a explicitement besoin.

c. Configuration globale

Dans sa configuration la plus simple, une implémentation samba comprend un serveur qui héberge une ou plusieurs ressources. Certains paramètres concernent le fonctionnement global et l'identité de ce serveur et se retrouveront dans une section appelée **global** du fichier **smb.conf**.

Dans les exemples qui suivent, nous nous placerons dans la situation d'un serveur simple, hors domaine Windows, qui présente des partages à des clients Windows.

Éléments courants de la section [global] dans smb.conf

```
workgroup = groupe_de_travail
   server string = commentaire
   log file = /chemin/log.%m
   max log size = log_maxi
   security = user (defaut)
   encrypt passwords = true (defaut)
```

Section [global] du fichier smb.conf	
groupe_de_travail	Le nom du groupe de travail du serveur. Notez que ce paramètre désigne aussi le nom du domaine dans un fonctionnement en domaine.
commentaire	Commentaire associé au serveur. Visible par exemple dans le voisinage réseau des machines Windows.
log.%m	Définition du format standard des fichiers journaux.
log_maxi	Définition de la taille maximum des fichiers de journaux.
user	Facultatif car paramètre par défaut. Paramètre de sécurité qui oblige à une authentification avec un compte utilisateur.
encrypt passwords	Facultatif car paramètre par défaut. Nécessaire pour tous les clients modernes qui présenteront naturellement des mots de passe cryptés (depuis NT4SP3).

2. Partage de répertoire

Le partage de répertoire se fait par l'ajout d'une section dans le fichier **smb.conf**.

Format type d'une section partage dans smb.conf

```
[nom_partage]
   comment = commentaire
   path = chemin
   readonly = lecture_seule
   browseable = yes
```

Déclaration de partage dans smb.conf.	
nom_partage	Le nom sous lequel le partage sera vu par les machines Windows.
commentaire	Facultatif. Définition du commentaire associé au partage.

chemin	Définition du chemin du répertoire à partager. Le répertoire doit exister dans le système de fichier Linux.
lecture_seule	Définition de l'accès au partage en lecture seule ou en lecture-écriture. *lecture_seule* aura la valeur yes ou no selon la configuration choisie. Notez que ce paramètre s'applique au partage et que l'accès reste soumis aux permissions du système de fichiers Linux.
browseable	Gestion de la visibilité du partage depuis les clients.

Si on regarde l'ensemble des paramètres possibles pour le fichier **smb.conf**, on peut légitimement être impressionné par leur quantité. Il faut savoir que nombre de paramètres fonctionnels peuvent être exprimés de plusieurs façons. Prenons l'exemple de l'accès à un partage en lecture seule vu dans la déclaration des partages. Les propositions suivantes sont toutes équivalentes :

readonly = yes

readonly = true

writable = no

writable = false

writeable = no

writeable = false

3. Gestion des identités

a. Algorithmes de hachage et stockage des mots de passe

Sur la très grande majorité des systèmes d'exploitation et applications, les mots de passe ne sont pas stockés en clair au sein du système. Les mots de passe des comptes sont cryptés et c'est la version cryptée qui est seule stockée. Le mot de passe en clair est oublié aussitôt qu'il a été créé.

Quand un utilisateur se connecte et tape ses éléments d'identification, le mot de passe est aussitôt crypté, et cette version fraîchement cryptée du mot de passe est comparée avec la version cryptée stockée dans la base de comptes du système. Ainsi, le mot de passe ne circule jamais en clair sur le réseau.

Les algorithmes employés pour crypter le mot de passe appartiennent à la famille des algorithmes de hachage. Ils ont un fonctionnement un peu particulier en ce sens qu'ils permettent de crypter, mais jamais de décrypter des données : ils sont à sens unique, et de ce fait un peu à part dans le monde de la cryptographie. Ce mode de fonctionnement explique pourquoi quand un utilisateur perd son mot de passe, on peut lui en réaffecter un, mais pas lui dire quel était le mot de passe oublié. La seule information stockée est la version cryptée du mot de passe, et elle est par hypothèse indéchiffrable.

Les algorithmes de hachages les plus courants s'appellent MD4, MD5 et SHA1. Ils sont utilisés pour stocker les mots de passe, les opérations de signature numérique ou les contrôles d'intégrité.

b. Authentification auprès des serveurs Samba

Un serveur Linux avec la suite logicielle Samba installée utilise nativement les comptes du système pour les authentifications Samba. Ainsi, toute connexion de la part d'un client se fait avec un compte hébergé par le système Linux. Cette situation risque toutefois de poser un problème. Le client Windows va présenter un mot de passe crypté par l'algorithme de hachage natif des systèmes Windows MD4 : Message Digest 4, alors que les mots de passe des systèmes Linux exploitent l'algorithme MD5 : Message Digest 5. Le mot de passe crypté présenté par le client Windows ne sera donc pas le même que celui stocké dans le fichier **/etc/shadow** du système Linux et l'authentification sera donc impossible, même si le mot de passe en clair est le même.

Pour que les clients Windows puissent s'authentifier après de systèmes Linux, il faut donc que ces systèmes hébergent une version du mot de passe cryptée en MD4 en plus du mot de passe natif Linux crypté en MD5. Ces deux mots de passe sont gérés indépendamment et peuvent même être différents.

c. Génération des mots de passe MD4

La commande spécifique **smbpasswd** permet la création d'un mot de passe MD4 pour un compte Linux existant. Ce mot de passe est stocké à part, généralement dans un fichier **/etc/samba/smbpasswd**.

Syntaxe de la commande smbpasswd pour affecter un mot de passe

```
smbpasswd -a nom_compte
```

Commande smbpasswd : options et paramètres	
-a	Facultatif. Nécessaire si le compte ne dispose pas encore de mot de passe samba.
nom_compte	Le compte Linux auquel il faut affecter un mot de passe samba.

d. Synchronisation avec les mots de passe Linux

Il est possible de demander à synchroniser les mots de passe samba avec les mots de passe du système Linux. Attention, comme expliqué précédemment, les mots de passe sont encryptés dans les deux systèmes avec deux algorithmes de hachage différents, par définition irréversibles. La synchronisation ne peut donc se faire qu'au moment où le mot de passe est saisi en clair lors de l'utilisation de la commande **smbpasswd**. Le mot de passe en clair est alors encrypté deux fois avec les deux algorithmes différents, et les deux bases de compte sont modifiées. Cette synchronisation est activée par une directive dans le fichier **smb.conf**.

Activation de la synchronisation de mots de passe dans smb.conf

```
unix password sync = yes
```

e. Suppression ou désactivation d'un compte samba

On peut souhaiter interrompre pour un utilisateur l'accès aux ressources partagées sur un serveur samba. La commande **smbpasswd** permet de supprimer, désactiver ou de réactiver le compte samba, indépendamment du compte Linux associé.

Commande smbpasswd pour désactiver un compte samba

```
smbpasswd -d nom_compte
```

Commande smbpasswd pour réactiver un compte samba

```
smbpasswd -e nom_compte
```

Commande smbpasswd pour supprimer un compte samba

```
smbpasswd -x nom_compte
```

Où *nom_compte* représente le compte utilisateur samba à manipuler. Il est à noter que les opérations sur les comptes samba n'ont aucune incidence sur le compte Linux correspondant.

4. Le client Samba

Le client samba permet d'accéder à un partage d'une machine Windows ou Samba depuis un client Linux. Il permet éventuellement à un client Linux de se connecter à un serveur Samba Linux, mais on l'envisagera plutôt pour accéder à des données sur un partage Windows depuis une machine Linux. Les deux commandes principales du client samba sont **smbclient** et **smbmount**.

a. Exploitation ponctuelle de ressources avec smbclient

On utilise essentiellement **smbclient** pour obtenir des informations sur les ressources partagées hébergées par un serveur SMB.

Utilisation de smbclient pour récupérer des informations sur un serveur smb

```
smbclient -L adresse_serveur -U nom_utilisateur
```

smbclient pour affichage des partages : paramètres	
adresse_serveur	L'adresse IP du serveur dont on veut observer les ressources.
nom_utilisateur	Indique le nom de l'utilisateur qui fait la requête auprès du serveur. Doit être un compte existant et valide sur le serveur.

On pourra également utiliser la commande **smbclient** de façon interactive en se connectant à une ressource partagée et en accédant à un shell qui permettra de réaliser des opérations sur les fichiers.

Utilisations de smbclient en mode interactif

```
smbclient \\\\adresse_serveur\\partage -U nom_utilisateur
smbclient //adresse_serveur/partage -U nom_utilisateur
```

Où *partage* représente le nom du partage hébergé par le serveur. Les multiples antislash sont nécessaires même s'ils obligent à une syntaxe un peu curieuse. En fait, il s'agit d'un chemin UNC : *Uniform Naming Convention*, utilisé pour désigner une ressource dans les environnements Windows. Un chemin UNC est composé du nom du serveur précédé de deux antislashs, puis du chemin de la ressource, séparé par un slash à chaque niveau. Or, il se trouve que dans les environnements Linux, l'antislash est un caractère réservé qui indique que le shell ne doit pas interpréter le caractère suivant. Pour écrire un véritable antislash, il faut donc le faire précéder d'un premier antislash qui indique que le deuxième doit être considéré comme un antislash naturel. Une alternative plus légère consiste à redresser les slashs et à utiliser les slashs droits. Les deux syntaxes sont admises.

Une fois cette commande exécutée et après avoir tapé le mot de passe de l'utilisateur, on est dans un shell spécifique **smbclient** qui permet de réaliser des opérations sur les fichiers. Les principaux usages seront évidemment de récupérer ou d'envoyer des fichiers vers le partage. On peut se déplacer dans l'arborescence avec la commande **cd**, puis les deux commandes essentielles seront **get** pour récupérer des fichiers, et **put** pour envoyer des fichiers vers le partage.

Exemple d'utilisation de smbclient en mode interactif

L'utilitaire smbclient présente un jeu de commandes semblable à celui des clients FTP.

```
alpha:~# smbclient \\\\192.168.0.1\\data -U toto
Enter toto's password:
Domain=[WSERVEUR] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> ls
  .                                   D        0  Wed Feb  3 19:28:33 2010
  ..                                  D        0  Wed Feb  3 19:28:33 2010
  deux                                D        0  Wed Feb  3 18:50:05 2010
  un                                  D        0  Wed Feb  3 19:28:38 2010

    40915 blocks of size 262144. 34718 blocks available
smb: \> cd un
smb: \un\> ls
  .                                   D        0  Wed Feb  3 19:28:38 2010
  ..                                  D        0  Wed Feb  3 19:28:38 2010
  fichier.txt                         A       27  Wed Feb  3 19:15:49 2010
  truc.bmp                            A        0  Wed Feb  3 18:46:44 2010

    40915 blocks of size 262144. 34718 blocks available
smb: \un\> get fichier.txt
getting file \un\fichier.txt of size 27 as fichier.txt (2,0 kb/s) (average 2,0 kb/s)
smb: \un\> exit
alpha:~# ls
fichier.txt
alpha:~#
```

b. Montage d'un partage smb avec smbmount

Si **smbclient** permet un accès ponctuel à des partages, il existe un moyen plus confortable d'exploiter des répertoires partagés depuis un client Linux : le montage d'un partage sur la station Linux.

La commande **smbmount** permet de réaliser le montage d'un partage SMB sur un répertoire local comme on peut le faire d'un filesystem local ou d'un partage NFS.

Syntaxes de la commande smbmount

```
smbmount \\\\adresse_serveur\\partage point_montage -o user=nom_utilisateur
```

```
smbmount //adresse_serveur/partage point_montage -o user=nom_utilisateur
```

smbmount : options et paramètres	
adresse_serveur	L'adresse IP du serveur dont on veut accéder au partage.
partage	Le nom du partage hébergé par le serveur.
point_montage	Le répertoire existant sur lequel sera monté le partage.
nom_utilisateur	Le nom de l'utilisateur qui fera la requête auprès du serveur. Doit être un compte existant et valide sur le serveur.

Il existe une alternative à cette syntaxe, c'est de réaliser le montage par la commande **mount** en appelant **smbmount** en tant que sous-programme. Cette syntaxe présente l'avantage d'uniformiser toutes les opérations de montage, et donc de ne retenir qu'une syntaxe générique.

Syntaxe de la commande mount pour partage smb

```
mount -t smbfs -o username=nom_utilisateur //adresse_serveur/partage point_montage
```

L'option **-t smbfs** provoque l'appel du sous-programme **smbmount** pour réaliser le montage, mais à partir d'une syntaxe quasi-standard pour réaliser le montage.

c. Montage d'un partage CIFS

Pour répondre aux besoins d'ouverture du protocole, SMB s'est normalisé, a évolué et s'appelle désormais CIFS : *Common Internet File System*. La suite logicielle Samba désigne désormais son client et les éléments logiciels sous ce nom. Les habitudes ayant la peau dure, l'usage de la dénomination SMB perdure encore largement.

Selon les versions de samba employées, on peut n'utiliser que smb, cifs seul ou smb et cifs indifféremment. La tendance est à la disparition de smb au profit de cifs.

Syntaxe de la commande mount pour partage cifs

```
mount -t cifs -o username=nom_utilisateur //adresse_serveur/partage point_montage
```

Il est possible de vérifier côté serveur quels sont les clients connectés. La commande **smbstatus** permet d'afficher les connexions smb actives.