connecter avec leur identifiant et mot de passe habituels (option local_enable) pour accéder à leurs fichiers. Toutefois, cette configuration convient parfaitement pour l'usage de Falcot SA.

11.4. Serveur de fichiers NFS

NFS (*Network File System*) est un protocole qui permet d'accéder à un système de fichiers à distance par le réseau, pris en charge par tous les systèmes Unix. Pour Windows, il faudra employer Samba.

NFS est un outil fort utile mais il ne faut jamais oublier ses limitations, surtout en termes de sécurité : toutes les données circulent en clair sur le réseau (un *sniffer* peut donc les intercepter) ; le serveur restreint les accès en fonction de l'adresse IP du client, ce qui le rend vulnérable au *spoofing* (usurpation d'adresses IP) ; enfin, si une machine est autorisée à accéder à un système de fichiers NFS mal configuré, son utilisateur root peut accéder à tous les fichiers du partage (n'appartenant pas à root) puisque le serveur NFS utilise l'identifiant utilisateur que le client lui a communiqué (sans aucune vérification possible ; le protocole est ainsi conçu depuis le début).

DOCUMENTATION

NFS howto

Malgré son grand âge, le NFS howto contient de nombreuses informations intéressantes, notamment des méthodes pour optimiser les performances de NFS. On y découvre aussi un moyen de sécuriser les transferts NFS à l'aide d'un tunnel SSH (mais cette technique ne permet pas d'employer `lockd`).

➨ http://nfs.sourceforge.net/nfs-howto/

11.4.1. Sécuriser NFS (au mieux)

Étant donné que NFS fait confiance aux informations reçues par le réseau, il convient de s'assurer que seules les machines autorisées à l'employer peuvent se connecter aux différents serveurs RPC qui lui permettent de fonctionner. Le pare-feu doit donc prohiber le *spoofing* pour qu'une machine extérieure ne puisse pas se faire passer pour une machine intérieure et les différents ports employés doivent être restreints aux machines devant accéder aux partages NFS.

B.A.-BA

RPC

RPC (*Remote Procedure Call*, ou appel de procédure distante) est un standard Unix pour des services distants. NFS est un service RPC.

Les services RPC s'enregistrent dans un annuaire, le *portmapper*. Un client désireux d'effectuer une requête NFS s'adresse au *portmapper* (port 111 en TCP ou UDP) et lui demande où se trouve le serveur NFS. On lui répond généralement en indiquant le port 2 049 (port par défaut pour NFS). Tous les services RPC ne disposent pas nécessairement d'un port fixe.

D'autres services RPC sont nécessaires au fonctionnement optimal de NFS, notamment `rpc.mountd`, `rpc.statd` et `lockd`. Malheureusement, ils emploient par défaut un port aléatoire affecté par le `portmapper` et il est donc difficile de filtrer le trafic qui leur est destiné. Les administrateurs de Falcot SA ont trouvé une solution à ce problème.

Les deux premiers services cités sont implémentés par des programmes utilisateur, démarrés respectivement par `/etc/init.d/nfs-kernel-server` et `/etc/init.d/nfs-common`. Ils disposent d'options pour forcer le choix des ports employés. Pour employer systématiquement les options adéquates, il faut modifier les fichiers `/etc/default/nfs-kernel-server` et `/etc/default/nfs-common`.

Ex. 11.22 *Fichier* `/etc/default/nfs-kernel-server`

```
# Number of servers to start up
RPCNFSDCOUNT=8

# Runtime priority of server (see nice(1))
RPCNFSDPRIORITY=0

# Options for rpc.mountd.
# If you have a port-based firewall, you might want to set up
# a fixed port here using the --port option. For more information,
# see rpc.mountd(8) or http://wiki.debian.org/SecuringNFS
# To disable NFSv4 on the server, specify '--no-nfs-version 4' here
RPCMOUNTDOPTS="--manage-gids --port 2048"

# Do you want to start the svcgssd daemon? It is only required for Kerberos
# exports. Valid alternatives are "yes" and "no"; the default is "no".
NEED_SVCGSSD=

# Options for rpc.svcgssd.
RPCSVCGSSDOPTS=
```

Ex. 11.23 *Fichier* `/etc/default/nfs-common`

```
# If you do not set values for the NEED_ options, they will be attempted
# autodetected; this should be sufficient for most people. Valid alternatives
# for the NEED_ options are "yes" and "no".

# Do you want to start the statd daemon? It is not needed for NFSv4.
NEED_STATD=

# Options for rpc.statd.
```

```
#   Should rpc.statd listen on a specific port? This is especially useful
#   when you have a port-based firewall. To use a fixed port, set this
#   this variable to a statd argument like: "--port 4000 --outgoing-port 4001".
#   For more information, see rpc.statd(8) or http://wiki.debian.org/SecuringNFS
STATDOPTS="--port 2046 --outgoing-port 2047"

# Do you want to start the idmapd daemon? It is only needed for NFSv4.
NEED_IDMAPD=

# Do you want to start the gssd daemon? It is required for Kerberos mounts.
NEED_GSSD=
```

Après ces modifications et un redémarrage des services, `rpc.mountd` emploie le port 2 048 ; `rpc.statd` écoute le port 2 046 et utilise le port 2 047 pour les connections sortantes.

Le service `lockd` est géré par un *thread* (processus léger) noyau, fonctionnalité compilée sous forme de module dans les noyaux Debian. Le module dispose également de deux options pour choisir systématiquement le même port : nlm_udpport et nlm_tcpport. Pour employer ces options automatiquement, il faut créer un fichier `/etc/modprobe.d/lockd` comme dans l'exemple ci-dessous.

Ex. 11.24 *Fichier* `/etc/modprobe.d/lockd`

```
options lockd nlm_udpport=2045 nlm_tcpport=2045
```

Avec tous ces paramétrages, il est maintenant possible de contrôler plus finement les accès au service NFS grâce à un pare-feu. Ce sont les ports 111 et 2 045 à 2 049 (en UDP et en TCP) qui doivent faire l'objet d'attentions particulières.

11.4.2. Serveur NFS

Le serveur NFS est intégré au noyau Linux ; Debian le compile dans ses noyaux sous forme de module. Pour l'activer automatiquement à chaque démarrage, il faut installer le paquet *nfs-kernel-server*, qui contient les scripts d'initialisation adéquats.

Le fichier de configuration du serveur NFS, `/etc/exports`, donne les répertoires exportés à l'extérieur. À chaque partage NFS sont associées des machines qui ont le droit d'y accéder. Un certain nombre d'options permettent de dicter quelques règles d'accès. Le format de ce fichier est très simple :

```
/repertoire/a/partager machine1(option1,option2,...) machine2(...) ...
```

Chaque machine est identifiée par son nom DNS ou son adresse IP. Il est aussi possible de spécifier un ensemble de machines en employant la syntaxe *.falcot.com ou en décrivant une plage complète d'adresses IP (exemples : 192.168.0.0/255.255.255.0, 192.168.0.0/24).

Par défaut, un partage n'est accessible qu'en lecture seule (option ro comme *read only*). L'option rw (comme *read-write*) donne un accès en lecture/écriture. Les clients NFS doivent se connecter depuis un port réservé à root (c'est-à-dire inférieur à 1 024) à moins que l'option insecure (pas sûr) n'ait été employée (l'option secure — sûr — est implicite en l'absence de insecure, mais on peut quand même la mentionner).

Le serveur ne répond à une requête NFS que lorsque l'opération sur disque a été complétée (option sync). L'option async (asynchrone) désactive cette fonctionnalité et améliore quelque peu les performances, au détriment de la fiabilité puisqu'il subsiste alors un risque de perte de données en cas de *crash* du serveur (des données acquittées par le serveur NFS n'auront pas été sauvegardées sur le disque avant le *crash*). La valeur par défaut de cette option ayant changé récemment (par rapport à l'historique de NFS), il est recommandé de toujours mentionner explicitement l'option souhaitée.

Pour ne pas donner un accès root au système de fichiers à n'importe quel client NFS, toutes les requêtes provenant d'un utilisateur root sont transformées en requêtes provenant de l'utilisateur nobody. Cette option (root_squash) est activée par défaut ; l'option inverse no_root_squash ne doit être employée qu'avec parcimonie étant donné les risques qu'elle comporte. Les options anonuid=*uid* et anongid=*gid* permettent d'employer un autre utilisateur écran à la place des UID/GID 65 534 (qui correspondent à l'utilisateur nobody et au groupe nogroup).

D'autres options existent encore, que vous découvrirez dans la page de manuel `exports(5)`.

ATTENTION

Première installation

Le script `/etc/init.d/nfs-kernel-server` ne démarre rien si le fichier `/etc/exports` ne prévoit aucun partage NFS. C'est pourquoi il faut démarrer le serveur NFS juste après avoir rempli ce fichier pour la première fois :

```
# /etc/init.d/nfs-kernel-server start
```

11.4.3. Client NFS

Comme tous les systèmes de fichiers, il est nécessaire de le monter pour l'intégrer dans l'arborescence du système. Étant donné qu'il s'agit d'un système de fichiers un peu particulier, il a fallu adapter la syntaxe habituelle de la commande `mount` et le format du fichier `/etc/fstab`.

Ex. 11.25 *Montage manuel avec la commande* `mount`

```
# mount -t nfs -o rw,nosuid arrakis.interne.falcot.com:/srv/partage /partage
```

Ex. 11.26 *Entrée NFS dans le fichier* `/etc/fstab`

```
arrakis.interne.falcot.com:/srv/partage /partage nfs rw,nosuid 0 0
```

L'entrée ci-dessus monte automatiquement à chaque démarrage le répertoire NFS `/srv/partage/` présent sur le serveur arrakis dans le répertoire local `/partage/`. L'accès demandé est en lecture/écriture (paramètre rw). L'option nosuid est une mesure de protection qui supprime tout bit setuid ou setgid présent sur les programmes contenus dans les partage NFS. Si le répertoire NFS est dédié au stockage de documents, il est recommandé d'employer de plus l'option noexec qui empêche l'exécution de programmes par NFS.

La page de manuel `nfs(5)` détaille toutes les options possibles.

11.5. Partage Windows avec Samba

Samba est une suite d'outils qui permettent de gérer le protocole SMB (aussi appelé « CIFS ») sous Linux. Ce dernier est employé par Windows pour accéder aux partages réseau et aux imprimantes partagées.

Samba sait également jouer le rôle de contrôleur de domaine Windows. C'est un outil extraordinaire pour assurer une cohabitation parfaite entre les serveurs sous Linux et les machines de bureautique encore sous Windows.

OUTIL

Administrer Samba avec SWAT

SWAT (*Samba Web Administration Tool*, outil d'administration web de Samba) est une interface web permettant de configurer le service Samba. Le paquet Debian *swat* n'activant pas l'interface de configuration par défaut, il faut le faire manuellement en exécutant `update-inetd --enable swat`.

SWAT est alors accessible à l'URL `http://localhost:901`. Pour y accéder, il faut employer le compte root (et le mot de passe administrateur habituel). Attention cependant, SWAT réécrit le fichier `smb.conf` à sa manière pensez donc à en faire une copie préalable si vous ne faites qu'essayer cet outil.

SWAT, très agréable à utiliser, dispose d'un assistant qui permet de définir le rôle du serveur en trois questions. Il est ensuite possible de configurer toutes les options globales ainsi que celles de tous les partages. On peut bien entendu créer de nouveaux partages. Chaque option est accompagnée d'un lien qui renvoie à la documentation correspondante.

Malheureusement, plus personne de maintient SWAT de manière active et il sera vraisemblablement supprimé de la prochaine version stable de Debian (*Jessie*).