Linux NFS HOWTO

Nicolai Langfeldt `janl@linpro.no` v1.0, 1er octobre 1999

(30 novembre 1999. Adaptation française par Christophe Deleuze, `Christophe.Deleuze@lip6.fr`). HOWTO décrivant l'installation de serveurs et clients NFS.

Contents

1 Préambule

1.1 Blabla légal

1.2 Dénégation

Ni Nicolai Langfeldt, ni Ron Peters ni leurs employeurs, ni qui que ce soit, n'assume de responsabilité pour les conséquences que les instructions de ce document peuvent avoir. Si vous choisissez de suivre ces instructions, bonne chance !

1.3 Retour

Ce document ne sera jamais terminé, merci de m'envoyer par mail vos problèmes et réussites, cela pourra améliorer ce HOWTO. Envoyez argent, commentaires et questions à *janl@math.uio.no* , ou *rpeters@hevanet.com* pour ce qui concerne les serveurs NFS sur disquette. Si vous m'envoyez un mail, par pitié, *vérifiez* que l'adresse de retour soit correcte et fonctionne. Vous ne vous imaginez pas combien de mes réponses sont revenues à cause d'une adresse incorrecte.

1.4 Autre blabla

Si vous voulez traduire ce HOWTO merci de me le signaler, que je puisse savoir en quels langages j'ai été publié :-). [Ndt : c'est fait...]

Remerciements à Olaf Kirch pour m'avoir convaincu d'écrire ceci, puis fourni de bonnes suggestions.

Les commentaires sur la traduction sont à envoyer à Christophe Deleuze, Christophe.Deleuze@lip6.fr.

2 LISEZMOI.d_abord

NFS, le système de fichiers par réseau, a trois caractéristiques importantes :

- il permet le partage de fichiers sur un réseau ;
- il marche suffisamment bien ;
- il crée tout un tas de problèmes de sécurité bien connus des crackers qui peuvent facilement les exploiter pour obtenir l'accès (lecture, écriture et effacement) à tous vos fichiers.

Je parlerai de ces deux aspects dans ce HOWTO. Lisez bien la section sécurité et vous supprimerez quelques risques stupides. Ne dites pas que je ne vous ai pas prévenus. Les passages sur la sécurité sont parfois assez techniques et demandent quelques connaissances en réseau IP. Si vous ne connaissez pas les termes utilisés vous pouvez soit consulter le HOWTO réseau, improviser ou vous procurer un livre sur l'administration de réseau TCP/IP pour vous familiariser avec TCP/IP. C'est une bonne idée de toutes façons si vous administrez des machines UNIX/Linux. Un très bon livre sur le sujet est *TCP/IP Network Administration* par Craig Hunt, publié par O'Reilly & Associates, Inc. Et quand vous l'aurez lu et compris, vous vaudrez plus cher sur le marché du travail, vous ne pouvez qu'y gagner :-)

Il y a deux sections pour vous aider à régler vos problèmes NFS, la *Mount Checklist* et les *FAQs*. Jetez-y un oeil si quelque chose ne marche pas comme prévu.

Si vous voulez/avez besoin de le récupérer et compiler vous même, le site de référence pour le nfsd Linux 2.0 est *ftp.mathematik.th-darmstadt.de:/pub/linux/okir* .

À propos de NFS sous Linux 2.2 voir 10 (la section sur Linux 2.2).

3 Installer un serveur NFS

3.1 Conditions préalables

Avant de continuer à lire ce HOWTO, vous aurez besoin de pouvoir faire des telnet dans les deux sens entre les machines que vous utiliserez comme serveur et client. Si cela ne fonctionne pas, voyez le HOWTO réseau (NET-3) et configurez correctement le réseau.

3.2 Premiers pas

Avant de faire quoi que ce soit d'autre, il nous faut un serveur NFS installé. Si vous faites partie d'un département réseau d'une université ou autre, il y a probablement un grand nombre de serveurs NFS qui tournent déjà. Si votre but est d'utiliser un serveur déjà installé alors vous pouvez sauter à 4 (la section sur l'installation d'un client NFS).

Si vous devez installer un serveur sur une machine non Linux vous devrez lire les pages de manuel du système pour trouver comment configurer le serveur NFS et l'exportation des systèmes de fichiers par NFS. Ce HOWTO contient une section décrivant les manipulations nécessaires sur divers systèmes. Ceci fait, vous pourrez passer à la section suivante. Ou continuer à lire cette section vu que certaines des choses que je vais dire sont pertinentes quel que soit le type de machine que vous utilisez comme serveur.

Si vous utilisez Linux 2.2, voyez 10 (la section sur Linux 2.2) avant de passer à la suite.

Nous allons maintenant configurer tout un tas de programmes.

3.3 Le portmapper

Le portmapper de Linux est appelé soit `portmap` soit `rpc.portmap`. La page de manuel sur mon système dit que c'est un convertisseur de port DARPA vers numéro de programme RPC. C'est là que se trouve la première faille de sécurité. La gestion de ce problème est décrite à la section 6 (sur la sécurité), que, encore une fois, je vous invite très fortement à lire.

Lancez le portmapper. Il devrait être dans le répertoire `/usr/sbin` (sur quelques machines il est appelé rpcbind). Vous pouvez le lancer à la main pour cette fois mais il devra être lancé à chaque démarrage de la machine, il faudra donc créer ou éditer les scripts rc. Les scripts rc sont décrits dans la page de manuel init, ils sont généralement dans `/etc/rc.d`, `/etc/init.d` ou `/etc/rc.d/init.d`. S'il y a un script qui a un nom du genre `inet`, c'est probablement le script à éditer. Mais ce qu'il faut écrire ou faire sort du cadre de ce HOWTO. Lancez portmap, et vérifiez qu'il tourne avec `ps -aux`, puis `rpcinfo -p`. Il y est ? Benissimo.

Encore une chose. L'accès distant à votre portmapper est contrôlé par le contenu de vos fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Si `rcpinfo -p` échoue alors que le portmapper tourne, vérifiez ces fichiers. Voyez la section 6 (sur la sécurité) pour les détails concernant ces fichiers.

3.4 Mountd et nfsd

Les prochains programmes à lancer sont mountd et nfsd. Mais d'abord il faut éditer un autre fichier, `/etc/exports`. Disons que je veux que le système de fichiers `/mn/eris/local` qui est sur la machine `eris` soit disponible sur la machine `apollon`. Je l'indique dans `/etc/exports` sur eris :

```
/mn/eris/local  apollon(rw)
```

La ligne ci-dessus donne à `apollon` un accès en lecture/écriture sur `/mn/eris/local`. Au lieu de `rw` on pourrait mettre `ro` pour *read only* (lecture seule, c'est la valeur par défaut). D'autres options existent, et je parlerai de quelques unes liées à la sécurité plus loin. Elles sont toutes décrites dans la page de manuel `exports` qu'il faut lire au moins une fois dans sa vie. Il y a de meilleures façons de faire que de lister tous les hosts dans le fichier exports. Peuvent être autorisés à monter un système de fichiers NFS, des groupes réseaux (*net groups*) si vous utilisez NIS (ou NYS, auparavant connu sous le nom YP), des noms de domaines avec jokers et des sous réseaux IP. Mais il faudra vérifier qui peut obtenir un accès au serveur avec ce type d'autorisations groupées.

Note : ce fichier exports n'utilise pas la même syntaxe que d'autres Unix. Ce HOWTO contient une section sur la façon dont les autres Unix `export`ent leurs fichiers.

Maintenant nous sommes prêts à lancer mountd (ou peut-être s'appelle-t-il `rpc.mountd`), puis nfsd (qui s'appelle peut-être `rpc.nfsd`). Ils liront tous deux le fichier exports.

Si vous modifiez `/etc/exports`, vous devrez vous assurer que nfsd et mountd savent que le fichier a changé. La façon traditionnelle est de lancer `exportfs`. Beaucoup de distributions Linux n'ont pas le programme exportfs. Si c'est le cas sur votre machine, vous pouvez installer ce script :

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo Volumes NFS réexportés
```

Sauvez le dans `/usr/sbin/exportfs` par exemple, et n'oubliez pas de lui appliquer `chmod a+rx`. Désormais, chaque fois que vous changez votre fichier exports, lancez ensuite exportfs en root.

Maintenant, vérifiez que mountd et nfsd fonctionnent correctement. D'abord avec `rpcinfo -p`. Il devrait donner quelque chose du genre :

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    745  mountd
    100005    1   tcp    747  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

On voit que le portmapper a annoncé ses services, de même que mountd et nfsd.

Si vous obtenez : `rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused`, `RPC_PROG_NOT_REGISTERED` ou quelque chose du style c'est que le portmapper ne tourne pas. OU, vous avez quelques chose dans `/etc/hosts.{allow,deny}` qui interdit au portmapper de répondre, voyez 6 (la section sécurité) à ce propos. Si vous obtenez `No remote programs registered` alors soit le portmapper ne veut pas vous parler, soit quelque chose ne marche pas. Tuez nfsd, mountd et le portmapper et essayez de recommencer.

Après avoir vérifié que le portmapper rend compte des services vous pouvez vérifier aussi avec `ps`. Le portmapper continuera à afficher les services même si les programmes qui les offrent ont crashé. Il vaut donc mieux vérifier par ps si quelque chose ne marche pas.

Bien sur, vous devrez modifier vos fichiers systèmes rc pour lancer mountd et nfsd au démarrage de la même façon que le portmapper. Il y a de très fortes chances que les scripts existent déjà sur votre machine, vous aurez juste à décommenter les bonnes lignes ou les activer pour les bons *runlevels* (pardon niveaux d'exécution) d'init.

Quelques pages de manuel avec lesquelles vous devriez être familier : portmap, mountd, nfsd et exports.

Bon, si vous avez tout fait exactement comme j'ai dit vous êtes prêts à enchaîner sur le client NFS.

4 Installer un client NFS

Tout d'abord il faudra compiler un noyau avec le système de fichiers NFS, soit compilé dans le noyau, soit disponible sous forme de module. Si vous n'avez encore jamais compilé un noyau vous aurez peut être besoin de consulter le HOWTO du noyau. Si vous utilisez une distribution très cool (comme Chapeau Rouge) et que vous n'avez jamais trifouillé le noyau (pas toucher toucher) il y a des chances que NFS soit automagiquement disponible.

Vous pouvez maintenant, à l'invite (prompt) du root, entrer la commande `mount` appropriée et le système de fichiers apparaîtra. Continuons avec l'exemple de la section précédente, nous voulons monter `/mn/eris/local` depuis eris. La commande est :

```
mount -o rsize=1024, wsize=1024 eris:/mn/eris/local /mnt
```

Nous reviendrons plus tard sur les options rsize et wsize. Le système de fichiers est maintenant disponible sous /mnt et vous pouvez faire un cd sur lui, puis un ls et regarder les fichiers individuellement. Vous remarquerez que ce n'est pas aussi rapide qu'avec un système de fichiers local, mais beaucoup plus pratique que ftp. Si, au lieu de monter le système de fichiers, mount renvoie un message d'erreur comme `mount: eris:/mn/eris/local failed, reason given by server: Permission denied` alors le fichier exports est incorrect, ou vous avez oublié de lancer exportfs après avoir modifié le fichier exports. S'il dit `mount: clntudp_ipdate: RPC: Program not registered` cela signifie que nfsd ou mountd ne tourne pas sur le serveur, ou que vous avez le problème avec les fichiers `hosts.{allow,deny}` mentionné plus haut.

Pour vous débarrasser du système de fichiers, vous pouvez faire :

```
umount /mnt
```

Pour que le système monte automatiquement un système de fichiers NFS au démarrage, éditez `/etc/fstab` de la façon habituelle. Par exemple avec une ligne comme celle-ci :

```
# device       mountpoint    fs-type    options           dumps  sfckorder
...
eris:/mn/eris/local   /mnt   nfs     rsize=1024,wsize=1024   0    0
...
```

C'est presque tout ce qu'il y a à savoir. Vous pouvez jeter un coup d'oeil à la page de manuel `nfs`. Continuons plize.

4.1 Options de montage

Il y a trois comportements principaux des clients NFS en cas de chute du serveur qui sont spécifiés par les options de montage :

soft

Le client NFS renverra une erreur au processus concerné si après quelques essais le serveur NFS persiste à ne pas répondre. Si vous voulez utiliser cette option, vous devez vérifier que votre logiciel la gère correctement. Je ne recommande pas ce réglage, c'est un bon moyen de perdre des données et corrompre des fichiers. En particulier, n'utilisez pas ça pour les disques où sont stockés vos mails (si vous tenez à vos mails).

hard

Le client NFS réessaiera infiniment jusqu'à ce qu'il soit tué. Les opérations reprendront normalement si le serveur NFS se rétablit ou redémarre. Le client ne pourra pas être interrompu ou tué.

hard,intr

Comme hard, mais Ctrl-C tuera le processus bloqué. Dans quelques cas, notament un disque /usr/spool/mail monté par NFS cela ne changera rien car le shell ignore le Ctrl-C quand il teste si vous avez du mail. Je recommande cette option pour **tous** les systèmes de fichiers NFS, y compris le *spool* du mail.

Reprenons l'exemple précédent, votre entrée fstab est maintenant :

```
# device       mountpoint   fs-type    options            dumps  sfckorder
...
eris:/mn/eris/local   /mnt  nfs   rsize=1024,wsize=1024,hard,intr 0   0
...
```

4.2 Optimisation de NFS

Normalement, si les options rsize et wsize ne sont pas précisées, NFS écrira et lira par blocs de 4096 ou 8192 octets. Mais certaines combinaisons de noyau Linux et cartes réseau ne peuvent pas fonctionner avec ces valeurs, de plus, même si cela marche, cela peut ne pas être optimal du tout. Il nous faudra donc expérimenter et trouver les valeurs de rsize et wsize qui fonctionnent et donnent les transferts les plus rapides. Vous pouvez tester la vitesse obtenue avec différentes valeurs des options avec des commandes simples. La commande mount ci-dessus ayant été exécutée, si vous avez l'accès en écriture sur le disque vous pouvez tester les performances en écriture séquentielle :

```
time dd if=/dev/zero of=/mnt/testfile bs=16k count=4096
```

Ceci crée un fichier de 64 Mo ne contenant que des 0. Faites le quelques (5-10?) fois et prenez la moyenne des temps. C'est le temps 'elapsed' ou 'wall clock' qui est le plus intéressant. Ensuite vous pouvez tester les performances en lecture en relisant le fichier :

```
time dd if=/mnt/testfile of=/dev/null bs=16k
```

faites le quelques fois et prenez la moyenne. Puis démontez (`umount`) et remontez (`mount`) avec des valeurs plus grandes pour rsize et wsize. Il vaut mieux prendre des multiples de 1024, et probablement pas plus grand que 16384 octets, car les gros blocs ralentissent les accès aléatoires. Immédiatement après avoir `remount`é avec une taille supèrieure, placez vous (`cd`) dans le système de fichiers et faites des trucs comme ls, explorez un peu pour vérifier que tout est bien normal. Si la valeur de rsize/wsize est trop grande, les symptômes sont *vraiment* bizarres et pas évidents. Un symptôme typique est une liste de fichiers donnée par `ls` incomplète sans aucun message d'erreur. Ou la lecture de fichier qui échoue mystérieusement et sans message d'erreur. Après vous être assurés que les wsize/rsize choisis fonctionnent, vous pouvez faire les tests de rapidité. Différentes plateformes de serveur auront peut-être des tailles optimales différentes. SunOS et Solaris sont réputés pour être beaucoup plus rapides avec une taille de 4096 octets.

Les noyaux Linux récents (depuis 1.3) font parfois des lectures anticipées (*read ahead*) pour des rsizes supérieurs ou égaux à la taille de page de la machine. Sur les processeurs Untel la taille de la page est de 4096 octets. La lecture anticipée augmentera *sensiblement* les performances en lecture. Sur une machine Untel on devrait donc choisir un rsize de 4096 si c'est possible.

Un truc pour augmenter les performances d'écriture de NFS est d'invalider les écritures synchrones sur le serveur. Les spécifications de NFS disent que les requêtes d'écriture de NFS ne doivent pas être considérées comme terminées avant que les données ne soient sur un médium non versatile (normalement le disque). Ceci réduit les performances à l'écriture, les écritures asynchrones sont plus rapides. Le nfsd Linux ne fait pas d'écritures synchrones car l'implémentation du système de fichiers ne s'y prête pas, mais sur les serveurs non Linux vous pouvez augmenter les performances de cette façon dans votre fichier exports :

```
/dir    -async, access=linuxbox
```

ou quelque chose du genre. Référez vous à la page de manuel exports de la machine concernée. Notez que ceci augmente les risques de perte de données.

5 NFS sur les lignes à faible débit

Les lignes lentes (à faible débit) comprennent les modems, RNIS et aussi sans doute les autres connexions longue distance.

Cette section est basée sur la connaissance des protocoles utilisés mais pas sur des expérimentations. Faites moi savoir si vous essayez ceci ;-)

La première chose à retenir est que NFS est un protocole lent. Il a un grand *overhead* (sur-coût en bande passante). Utiliser NFS, c'est presque comme utiliser kermit pour transférer des fichiers. Il est *lent*. Presque tout est plus rapide que NFS. FTP est plus rapide. HTTP est plus rapide. rcp est plus rapide. ssh est plus rapide.

Vous voulez toujours l'essayer ? Ok.

Par défaut NFS est paramétré pour des lignes rapides et à faible latence. Si vous utilisez les paramètres par défaut sur des lignes à grande latence cela peut provoquer des erreurs, des annulations, des rétrécissements de fichiers, et des comportements bizarres.

La première chose à faire est de ne *pas* utiliser l'option de montage `soft`. Les temporisations retourneront des erreurs au logiciel, qui, dans l'immense majorité des cas, ne saura pas quoi en faire. C'est une bonne façon d'avoir des problèmes bizarres. Utilisez plutôt l'option de montage `hard`. Quand `hard` est actif les temporisations déclenchent des essais infinis au lieu d'annuler ce que le logiciel était en train de faire (quoi que ce soit). C'est ce que vous voulez. Vraiment.

La deuxième chose à faire est d'ajuster les options de montage `timeo` et `retrans`. Elles sont décrites dans la page de manuel nfs(5), en voici un extrait (version française) :

```
       timeo=n        La valeur,  en  dixiemes  de  secondes,  du
                      delai   avant  de  declencher  la  premiere
                      retransmission d'une RPC.   La  valeur  par
                      defaut  est 7/10 de seconde. Apres une pre
                      miere expiration, le delai  est  double  et
                      l'on recommence les retransmissions jusqu'a
                      ce que le delai atteigne la valeur maximale
                      de 60 secondes, ou que le nombre maximal de
```

```
                         retransmission soit depasse.  Il se produit
                         alors  une  erreur  d'expiration majeure de
                         delai.  Si le systeme est monte  "en  dur",
                         les  retransmissions  reprendront a nouveau
                         indefiniment.

                         On peut ameliorer les performances en  aug
                         mentant  le delai sur un  reseau charge, si
                         le serveur est un  peu  lent,  ou  si  l'on
                         traverse plusieurs routeurs ou passerelles.

          retrans=n      Le  nombre  d'expirations  mineures  et  de
                         retransmissions  qui  doivent  se  produire
                         avant de declencher une expiration majeure.
                         La  valeur  par  defaut  est  3 expirations
                         mineures.  Quand  une  erreur  d'expiration
                         majeure  se  produit,  soit l'operation est
                         abandonnee, soit  un  message  "server  not
                         responding" est affiche sur la console.
```

En d'autres mots : si une réponse n'est pas reçue avant la temporisation de 0,7 seconde (700 ms), le client NFS répétera la requête et doublera la temporisation à 1,4 seconde. Si la réponse n'arrive pas dans les 1,4 seconde, la requête est répétée à nouveau et la temporisation est doublée à 2,8 secondes.

La vitesse de la ligne peut être mesurée avec un ping ayant vos valeurs de rsize/wsize comme taille de paquet.

```
$ ping -s 8192 lugulbanda
PING lugulbanda.uio.no (129.240.222.99): 8192 data bytes
8200 bytes from 129.240.222.99: icmp_seq=0 ttl=64 time=15.2 ms
8200 bytes from 129.240.222.99: icmp_seq=1 ttl=64 time=15.9 ms
8200 bytes from 129.240.222.99: icmp_seq=2 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=3 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=4 ttl=64 time=15.0 ms

--- lugulbanda.uio.no ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14.9/15.1/15.9 ms
```

Le temps indiqué est celui que le paquet du ping a pris pour aller et revenir de lugulbanda. 15 ms, c'est assez rapide. Sur une ligne à 28 800 bps vous pouvez vous attendre à une valeur de l'ordre de 4000-5000 ms, et si la ligne est chargée ce temps sera encore plus élevé, facilement le double. En général, la latence augmente avec la taille des paquets et la charge de la ligne. Si vous comptez utiliser FTP et NFS en même temps il faudra mesurer les temps du ping pendant un transfert FTP et augmenter `timeo` en accord avec la latence de votre ligne.

6 NFS et la sécurité

Je ne suis en aucun cas un expert en sécurité informatique. Mais j'ai traîné dans le secteur et j'ai un *petit* conseil pour ceux qui se préoccupent de la sécurité. Mais attention. Ce n'est pas absolument pas une liste complète des problèmes liés à NFS et si vous pensez être en sécurité une fois que vous avez lu et mis en pratique tout ceci alors j'ai un pilier de pont (presque neuf) à vous vendre.

Cette section n'a probablement pas d'intérêt si vous êtes sur un réseau *fermé* où vous avez confiance en tous les utilisateurs, et que personne en qui vous n'avez pas confiance ne peut obtenir un accès sur les machines du réseau. I.e., il ne devrait y avoir aucun moyen de se connecter à votre réseau depuis l'extérieur et il ne devrait être relié à aucun autre réseau où vous n'avez pas confiance en tous les utilisateurs et en sa sécurité. Vous pensez que je suis parano ? Pas du tout. C'est un conseil de sécurité *de base*. Et rappelez-vous que c'est juste le commencement. Un site *sûr* nécessite un administrateur consciencieux et bien informé qui sait où trouver l'information sur les problèmes de sécurité existants ou potentiels.

Un problème de base de NFS est que le client, si on ne lui demande pas le contraire, fera confiance au serveur NFS et vice versa. Ça peut être mauvais. Cela signifie que si le compte root du serveur est cassé (*broken into*) il peut être très facile de casser le compte root du client. Et vice versa. Il y a quelques moyens de gérer ce problème sur lesquels nous reviendrons.

Les documents consultatifs (*advisories*) du CERT sur NFS sont une bonne source d'information, la plupart des problèmes (et des solutions) évoquées ci-dessous sont traités dans ces documents. Voyez *ftp.cert.org:/01-README* pour une liste à jour. En voici quelques-uns liés à NFS (il n'y a pas à ma connaissance de version française) :

```
CA-91:21.SunOS.NFS.Jumbo.and.fsirand                         12/06/91
     Vulnerabilities concerning Sun Microsystems, Inc. (Sun) Network
     File System (NFS) and the fsirand program.  These vulnerabilities
     affect SunOS versions 4.1.1, 4.1, and 4.0.3 on all architectures.
     Patches are available for SunOS 4.1.1.  An initial patch for SunOS
     4.1 NFS is also available. Sun will be providing complete patches
     for SunOS 4.1 and SunOS 4.0.3 at a later date.

CA-94:15.NFS.Vulnerabilities                                 12/19/94
     This advisory describes security measures to guard against several
     vulnerabilities in the Network File System (NFS). The advisory was
     prompted by an increase in root compromises by intruders using tools
     to exploit the vulnerabilities.

CA-96.08.pcnfsd                                              04/18/96
     This advisory describes a vulnerability in the pcnfsd program (also
     known as rpc.pcnfsd). A patch is included.
```

6.1 Sécurité du client

Du côté client il y a quelques options de mount qui permettent de ne pas faire trop confiance au serveur. L'option `nosuid` interdit le démarrage de programmes suid depuis le système de fichiers NFS. C'est une option à utiliser systématiquement, car elle empêche le root du serveur de créer un fichier suid sur le système de fichiers NFS, puis de se loger dans le client en utilisateur et de lancer le programme suid pour devenir root sur le client. Il est aussi possible d'interdire l'exécution des fichiers du système de fichiers NFS avec l'option `noexec`. Mais ceci est beaucoup moins utile que `nosuid` car le système de fichiers contiendra très probablement au moins *quelques* scripts ou programmes à exécuter. Ces options se rentrent dans la colonne d'options, avec `wsize` et `rsize`, séparées par des virgules.

6.2 Sécurité du serveur : nfsd

Du côté serveur on peut ne pas faire confiance au root du client, avec l'option `root_squash` (rembarrage du root :-) dans le fichier exports :

```
/mn/eris/local apollon(rw, root_squash)
```

Dans ce cas, si un utilisateur du client avec l'UID 0 essaye d'accéder (en lecture, écriture ou effacement) au système de fichiers, le serveur remplace l'UID par celui de l'utilisateur 'nobody' du serveur. Ceci signifie que l'utilisateur root du client ne peut accéder/modifier les fichiers du serveur que seul le root du serveur peut accéder/modifier. C'est bien, et vous aurez probablement à utiliser cette option sur tous les systèmes de fichiers que vous exportez. J'en entends un qui me dit : "Mais l'utilisateur root du client peut toujours utiliser 'su' pour devenir n'importe qui et accéder à ses fichiers !" Et là je réponds : "Oui, c'est comme ça, c'est Unix". Ceci a une conséquence importante : tous les fichiers et binaires importants devraient appartenir à `root`, et pas `bin` ou un compte autre que root, car le seul compte auquel le root du client ne peut pas accéder est le compte root du serveur. Plusieurs autres options permettant de ne pas faire confiance à qui ne vous plait pas sont énumérées dans la page de manuel nfsd. Il y a aussi des options pour rembarrer (*to squash*) des intervalles d'UID ou GID.

Il est important aussi de s'assurer que nfsd vérifie que toutes les requêtes viennent d'un port privilégié. S'il accepte les requêtes de n'importe quel port du client, un utilisateur quelconque peut exécuter un programme qu'il est facile de se procurer sur l'Internet. Il parle le protocole NFS et pourra prétendre être n'importe qui et être cru. Ça fait peur hein ? Le nfsd Linux effectue cette vérification par défaut, sur d'autres systèmes d'exploitation il faut la valider. Ça devrait être décrit dans les pages du manuel de ce système.

Autre chose. N'exportez jamais un système de fichiers vers 'localhost' ou 127.0.0.1. Croyez-moi.

6.3 Sécurité du serveur : le portmapper

Le portmapper de base, en combinaison avec nfsd présente un problème de conception qui rend possible de récupérer les fichiers d'un serveur NFS sans avoir aucun privilège. Heureusement le portmapper utilisé par la plupart des distributions Linux est relativement sûr vis à vis de cette attaque, et peut être sécurisé en configurant les listes d'accès au moyen de deux fichiers.

Toutes les distributions de Linux ne sont pas égales. Certaines apparemment à jour n'incluent *pas* un portmapper sur, même aujourd'hui, alors que le problème est connu depuis plusieurs années. Au moins une distribution contient même la page de manuel pour un portmapper sûr alors que le portmapper effectivement installé n'est *pas* sûr. Pour déterminer simplement si votre portmapper est le bon ou pas, lancez strings(1) et voyez s'il lit les fichiers appropriés `/etc/hosts.deny` et `/etc/hosts.allow`. Si votre portmapper est `/usr/sbin/portmap` exécutez la commande `strings /usr/sbin/portmap | grep hosts`. Sur ma machine cela donne :

```
/etc/hosts.allow
/etc/hosts.deny
@(#) hosts_ctl.c 1.4 94/12/28 17:42:27
@(#) hosts_access.c 1.20 96/02/11 17:01:27
```

Tout d'abord, éditons `/etc/hosts.deny`. Il devrait contenir la ligne :

```
portmap: ALL
```

qui refusera l'accès à *quiconque*. Maintenant qu'il est fermé, lancez `rpcinfo -p` pour vérifier qu'il lit et se conforme au contenu du fichier. rpcinfo ne devrait rien renvoyer, ou peut être un message d'erreur. Il ne devrait *pas* y avoir besoin de relancer le portmapper.

Fermer le portmapper à tous le monde est peut être un peu excessif, nous ré-ouvrons donc quelque peu l'accès en éditant le fichier `/etc/hosts.allow`. Mais il faut d'abord savoir ce qu'on va y mettre. À la base, il devrait contenir les noms de toutes les machines qui doivent avoir accès à votre portmapper. Sur le système Linux moyen il y a très peu de machines qui ont une bonne raison de demander cet accès. Le portmapper administre nfsd, mountd, ypbind/ypserv, pcnfsd et les services "en r" comme ruptime et rusers. Parmis ceux-ci, seuls nfsd, mountd, ypbind/ypserv et peut-être pcnfsd ont de l'importance. Toutes les machines qui ont besoin d'accéder à ces services sur votre machine devraient y être autorisées. Disons que votre machine est 129.240.223.254 et que tout ce qui vit sur le sous réseau 129.240.223.0 doit pouvoir y accéder (si ceci n'est pas clair pour vous, voyez le HOWTO réseau). On écrit :

```
portmap: 129.240.223.0/255.255.255.0
```

dans `hosts.allow`. C'est l'adresse de réseau que vous donnez aussi à la commande `route` et le masque de réseau que vous donnez à `ifconfig`. Pour le périférique `eth0` sur cette machine `ifconfig` devrait donner :

```
...
eth0      Link encap:10Mbps Ethernet  HWaddr 00:60:8C:96:D5:56
          inet addr:129.240.223.254  Bcast:129.240.223.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:360315 errors:0 dropped:0 overruns:0
          TX packets:179274 errors:0 dropped:0 overruns:0
          Interrupt:10 Base address:0x320
...
```

et `netstat -rn` devrait donner :

```
Kernel routing table
Destination     Gateway         Genmask         Flags Metric Ref Use    Iface
...
129.240.223.0   0.0.0.0         255.255.255.0   U     0      0   174412 eth0
...
```

(Adresse réseau dans la première colonne)

Les fichiers `hosts.deny` et `hosts.allow` sont décrits dans les pages de manuel de mêmes noms.

IMPORTANT : ne *rien* mettre d'autre que des adresses IP (numériques) dans les lignes portmap de ces fichiers. Les *host name lookup* (recherche d'adresse IP (numérique) à partir de l'adresse alphanumérique ex. ftp.lip6.fr donne 132.227.77.2) peuvent indirectement déclencher une activité portmap qui déclenchera un *host name lookup* qui déclenchera...

Ceci fait, votre serveur devrait être un peu plus solide. Le dernier problème (mais oui !) est que quelqu'un casse le compte root (ou boute MS-DOS) sur une machine de confiance et utilise ce privilège pour envoyer des requêtes depuis un port sûr en se faisant passer pour n'importe quel utilisateur.

6.4 NFS et les coupent-feu (firewalls)

C'est une très bonne idée de bloquer les ports NFS et portmap dans votre routeur ou firewall. nfsd utilise le port 2049, que ce soit avec tcp ou udp. Le portmapper est au port 749 (tcp et udp) et mountd aux port 745 et 747 (tcp et udp). Vérifiez les ports avec la commande `rpcinfo -p`.

Si au contraire vous voulez que NFS traverse un firewall, il existe des options sur les nfsd et mountd récents pour leur spécifier le port à utiliser. Vous pouvez donc choisir un port qui ne soit pas bloqué par le firewall.

6.5 Résumé

Si vous configurez correctement votre installation portmapper/NFS avec hosts.allow/deny, root_squash, nosuid et les ports privilégiés, vous évitez beaucoup des bogues connues de NFS et pouvez presque vous sentir en sécurité au moins pour *ça*. Mais de toutes façons : quand un intrus obtient l'accès à votre réseau, il/elle peut faire apparaître des commandes bizarres dans votre .forward ou lire votre mail quand `/home` ou `/var/spool/mail` sont exportés. Pour la même raison, vous ne devriez jamais accéder à votre clé privée PGP par NFS. Ou au moins vous devez savoir quel est le risque. Et maintenant vous savez un peu.

NFS et le portmapper constituent un système complexe et il n'est donc pas totalement exclu que de nouvelles bogues soient découvertes, soit dans la conception soit dans l'implémentation que nous utilisons. Il pourrait même y avoir des défauts de sécurité connus, que quelqu'un utilise. Mais c'est la vie. Pour vous tenir au courant, vous devriez au moins lire les forums comp.os.linux.announce et comp.security.announce comme minimum absolu (en français, consultez fr.comp.os.linux.annonces).

7 "Checklist" mount

Cette section est basée sur la *mount checklist* (liste des problèmes liés à `mount`) de IBM Corp. Je les remercie de m'autoriser à l'utiliser dans ce HOWTO. Si vous avez un problème en montant un système de fichiers NFS, consultez cette liste avant de poster votre problème sur les niouzes. Chaque point décrit un type de problème et sa solution.

1. Mount répète `RPC: Program not registered`

   Le portmapper tourne ?

   Solution : lancez-le.

   mountd tourne ?

   Solution : lancez-le.

   nfsd tourne ?

   Solution : lancez-le.

   `/etc/hosts.deny` empêche le portmapper de répondre ?

   Solution : vous pouvez enlever la règle en question dans `hosts.deny` ou en ajouter une dans `hosts.allow` de façon que le portmapper soit autorisé à vous parler.

2. Système de fichier non exporté, ou non exporté au client en question.

   Solution : exportez le [Ndt : merci IBM !]

3. La résolution de noms ne s'accorde pas avec la liste des exports.

   e.g.: la liste des exports dit d'exporter vers `johnmad` mais le nom de `johnmad` est résolu en `johnmad.austin.ibm.com`. La permission de monter est refusée.

Solution : exportez vers les deux formes du nom.

Cela peut aussi arriver si le serveur a deux interfaces avec des noms différents et que les exports n'en spécifient qu'un.

Solution : exportez les deux interfaces.

Cela peut aussi se produire si le serveur ne peut pas faire un lookuphostbyname ou lookuphostbyaddr (ce sont des fonctions de bibliothèque) sur le client. Assurez-vous que le client peut faire `host` <`name`>; `host` <`ip_addr`>; et que les deux donnent la même machine.

Solution : mettez de l'ordre dans la résolution de noms.

4. Le système de fichiers a été monté après que NFS soit lancé (sur ce serveur). Dans ce cas le serveur exporte le point de montage sous-jacent, pas le système de fichiers.

   Solution : éteignez nfsd et relancez le.

   Note : les clients qui avaient monté le point de montage sous-jacent auront des problèmes pour y accéder après le redémarrage.

5. La date est très décalée sur une ou sur les deux machines (cela peut mettre la pagaille pour make)

   Solution : réglez correctement la date.

   L'auteur du HOWTO recommande d'utiliser NTP pour synchroniser les horloges. Vu qu'il y a des restrictions à l'exportation (au sens commercial !) de NTP aux É.U., vous devez vous procurer NTP pour Debian, Redhat ou Slakware depuis `ftp://ftp.hacktic.nl/pub/replay/pub/linux` ou un miroir.

6. Le serveur ne peut pas utiliser un mount d'un utilisateur qui est dans plus de 8 groupes.

   Solution : diminuez le nombre de groupes auxquels l'utilisateur appartient ou montez depuis un autre utilisateur.

8 FAQ

Voici la section FAQ. Elle est en partie basée sur une vieille FAQ NFS écrite par Alan Cox.

Si vous avez un problème pour monter un système de fichier, voyez si votre problème est décrit dans la section “Checklist mount”.

1. J'obtiens un tas d'erreurs “stale nfs handle” quand j'utilise Linux comme serveur NFS.

   Cela est dû à une bogue dans quelques vieilles versions de nfsd. Elle est corrigée à partir de nfs-server2.2beta16.

2. Quand j'essaye de monter le système de fichiers j'obtiens

   ```
   can't register with portmap: system error on send
   ```

   Vous utilisez probablement un système Caldera. Il y a une bogue dans les scripts rc. Contactez Caldera pour obtenir la solution.

3. Pourquoi ne puis-je pas exécuter un fichier après l'avoir copié sur le serveur NFS ?

   La raison est que nfsd cache les manipulations de fichiers pour des raisons de performances (rappelons qu'il fonctionne dans l'espace utilisateur). Ainsi, après une écriture le fichier peut ne pas être fermé tout de suite, et tant qu'il est ouvert le noyau ne vous autorisera pas à l'exécuter. Les nfsd plus récents que le printemps 95 [Ndt : hum...] ferment les fichiers ouverts après quelques secondes, les plus vieux pouvaient ne pas les relâcher avant plusieurs jours...

4. Mes fichiers NFS sont tous en lecture seule.

   Le serveur NFS Linux est par défaut en lecture seule. Voyez les sections “Mountd et nfsd” et “Exporter des systèmes de fichier” dans ce HOWTO et référez vous aux pages de manuel “exports” et “nfsd”. Vous devrez modifier `/etc/exports`.

5. Je monte depuis un serveur NFS Linux, `ls` marche et pourtant je ne peux pas lire ou écrire de fichiers.

   Sur les anciennes versions de Linux il faut monter un serveur NFS avec `rsize=1024, wsize=1024`.

6. Je monte depuis un serveur NFS Linux avec une taille de bloc comprise entre 3500 et 4000 et Linux crashe régulièrement.

   Bah alors ne le faites pas. Cela ne se produit pas avec les noyaux 2.0 et 2.2 ni, autant que je sache avec les 1.2.

7. Est-ce que Linux peut utiliser NFS sur TCP ?

   Non, pas pour le moment.

8. J'ai des tonnes d'erreurs bizarres en essayant de monter depuis un serveur Linux.

   Assurez-vous que vos utilisateurs sont dans 8 groupes au maximum. C'est une limitation des vieux serveurs.

9. Quand je redémarre ma machine elle se bloque parfois en essayant de démonter un serveur NFS bloqué (*hung*).

   Ne démontez **pas** les serveurs NFS en redémarrant ou arrêtant la machine, ça ne créera pas de problèmes si vous ne le faites pas. La commande est `umount -avt nonfs`.

10. Les clients NFS Linux sont très lents quand ils écrivent sur des systèmes Sun ou BSD.

    Normalement les écritures NFS sont synchrones (vous pouvez le désactiver si vous ne craignez pas de perdre des données). Les noyaux dérivés de BSD ont tendance à ne pas savoir travailler avec des petits blocs. Ainsi quand vous écrivez 4K de données depuis un client Linux dans des paquets de 1K, BSD fait ceci :

    ```
    lit une page de 4K
    traite 1K
    écrit 4K sur le disque
    lit une page de 4K
    traite 1K
    écrit 4K sur le disque
    ...
    ```

11. Quand je connecte de nombreux clients à un serveur NFS Linux, la performance diminue soudainement.

    Le protocole NFS utilise les paquets UDP fragmentés. Le noyau ne conserve qu'un nombre limité de fragments de paquets incomplets avant de commencer à jeter des paquets. En 2.2, ce paramètre est réglable à l'exécution au moyen du système de fichier /proc : `/proc/sys/net/ipv4/ipfrag_high_tresh` et `ipfrag_low_tresh`. En 2.0 ce sont des constantes définies à la compilation dans `.../linux/net/ipv4/ip_fragment.c`, `IPFRAG_HIGH_TRESH` et `IPFRAG_LOW_THRESH`. La signification des ces valeurs est que quand la mémoire consommée par les fragments UDP non réassemblés atteint “ipfrag_high_thresh” (en octets, 256K par défaut en 2.2.3 et 2.0.36) elle est ramenée à “ipfrag_low_tresh” d'un coup, en jetant des fragments. Ainsi, si la borne supérieure (high_tresh) est atteinte, la performance de votre serveur diminue drastiquement.

    256K est suffisant pour 30 clients. Si vous en avez 60, doublez la valeur. Et doublez aussi la borne inférieure (low_tresh).

12. J'utilise Linux 2.2 (ou suivant) avec knfsd et ma machine AIX, IRIX, Solaris, DEC-Unix, ... n'arrive pas à monter de volume.

    knfsd annonce qu'il implémente NFS version 3, alors que ce n'est pas vrai. Utilisez l'option qui permet de stopper ces annonces, ou mettez `"vers=2"` dans la liste d'options de montage de votre client.

13. Ma machine AIX 4 n'arrive pas à monter depuis mon serveur NFS Linux. Elle dit quelque chose du genre :

    ```
    mount: 1831-011 access denied for server:/dir
    mount: 1831-008 giving up on:
    server:/dir
    The file access permissions do not allow the specified action.
    ```

    AIX 4.2 utilise des ports réservés (<1024) pour NFS. AIX 4.2.1 et 4.3 peuvent utiliser d'autres ports, et essaient de monter par NFS3, NFS/TCP et finalement NFS/UDP.

    Ajouter

    ```
    nfso -o nfs_use_reserved_ports=1
    ```

    à la fin de `rc.tcpip` la forcera à utiliser les ports réservés (truc fourni par Brian Gorka).

9 Exporter un système de fichiers

Bien sur, la façon d'exporter les systèmes de fichiers par NFS n'est pas toujours la même sur toutes les plate-formes. Linux et Solaris 2 sont les plus déviants. Cette section liste de manière superficielle la façon de procéder sur la plupart des systèmes. Si votre système n'est pas traité ici, cherchez dans vos pages de manuel. Les mot-clés sont : nfsd, system administration tool, rc scripts, boot scripts, boot sequence, /etc/exports, exportfs. J'utiliserai le même exemple tout au long de cette section : comment exporter /mn/eris/local vers apollon en lecture/écriture.

9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX

Ces systèmes utilisent le format export traditionnel de Sun. Dans `/etc/exports`, écrivez :

```
/mn/eris/local -rw=apollon
```

La documentation complète se trouve dans la page de manuel `exports`. Après avoir édité le fichier, lancez `exportfs -av` pour exporter les systèmes de fichiers.

La rigueur de la syntaxe demandée par exportfs varie. Sur certains systèmes vous verrez que la ligne précédente peut être :

```
/mn/eris/local apollon
```

ou même quelque chose de dégénéré comme :

```
/mn/eris/local rw=apollon
```

Je recommande d'utiliser la syntaxe stricte. Il se peut que la prochaine version de `exportfs` soit plus exigeante vis à vis de la syntaxe et ne fonctionne plus.

9.2 Solaris 2

Sun ont complètement réinventé la roue quand ils ont fait Solaris 2, et donc c'est complètement différent des autres systèmes. Il faut éditer le fichier `/etc/dfs/dfstab` et y placer les commandes de partage (*share*) documentées dans la page de manuel `share(1M)`, comme ceci :

```
share -o rw=apollon -d "Eris Local" /mn/eris/local
```

Lancez ensuite le programme `shareall` pour exporter les systèmes de fichiers.

10 NFS sous Linux 2.2

Au moment où j'écris, la version courante du noyau est 2.2.12 et utiliser NFS peut être assez pénible. Ou pas. J'ignore ce qu'il en sera pour Linux 2.4.

La grosse nouveauté dans Linux 2.2 c'est le support d'un serveur nfs dans le noyau, appelé knfsd 2.2. Ce type d'implémentation a des avantages, principalement la rapidité, une machine Linux 2.2 avec knfsd est un serveur NFS respectable. Vous pouvez cependant toujours utiliser l'ancien nfsd avec Linux 2.2, et cela présente quelques avantages aussi, dont la simplicité.

Si vous utilisez un paquetage noyau source ou binaire fabriqué par quelqu'un comme RedHat (6.0 et suivantes), SuSE (6.1 et suivantes il me semble) ou un autre intégrateur de système professionnel ils auront probablement intégré complètement “knfsd” et vous n'avez pas de soucis à vous faire, cela marchera. Pour l'essentiel. Jusqu'à ce que vous vouliez compiler un noyau vous même. Si vous utilisez un noyau 2.2 standard (au moins jusqu'à 2.2.12) knfsd ne fonctionnera pas.

Pour le faire fonctionner vous même il vous faut le paquetage knfsd de H.J. Lu. C'est un ensemble de patchs avec les utilitaires requis pour 2.2 que Lu maintient bénévolement. Récupérez le depuis votre miroir de noyau local, le site maître est *ftp.kernel.org:/pub/linux/devel/gcc/* . **Ce n'est pas destiné au grand public**. Si vous trouvez que c'est trop compliqué, n'insistez pas et attentez qu'un paquetage noyau soit disponible auprès de votre intégrateur (Redhat, SuSE...).

Ne m'envoyez pas de question à ce sujet, je ne peux pas vous aider, je n'ai aucun serveur basé sur knfsd qui tourne. Si vous trouvez des erreurs ou omissions dans la documentation, écrivez-moi et je corrigerai ce HOWTO.

Toujours là ? Ok. H.J. Lu annonce les nouvelles versions de son paquetage sur la liste de diffusion linux-kernel, où il passe d'autres choses liées à NFS dans Linux 2.2. Lisez-la.

10.1 Le client

Le client est presque simple. Afin que les verrous (*locks*) marchent correctement il faut que `statd` (du paquetage knfsd) soit compilé, installé et lancé depuis vos scripts de démarrage. Statd a besoin d'un répertoire appelé `/var/lib/nfs` qu'il vous faudra créer avant de le lancer (sans quoi il se termine immédiatement sans message d'erreur).

Une fois que statd tourne vous pouvez utiliser le programme `testlk` (dans `tools/locktest`) pour tester si un verrou sur un fichier d'un volume monté par NFS fonctionne. Ça devrait. S'il affiche *No locks available*, statd ne fonctionne pas.

En fait, vous pouvez aussi vous passer des verrous (ce que je ne recommande pas) en mettant `"nolock"` dans la liste des options de montage.

Autant que je sache, c'est tout ce qu'il faut pour faire fonctionner correctement le client.

Ah, si vous avez un serveur NFS Alpha ou Sparc vous verrez que le client nfs de Linux 2.2 est vraiment de la merde. Les débits sont extrêmement faibles, bien pire qu'avec Linux 2.0. Bien sur on peut corriger le problème. Les noyaux 2.2 d'Alan Cox (un petit peu plus expérimentaux que ceux de Linus) incluent un patch pour améliorer la performance du client 2.2 avec un serveur Alpha ou Sparc. Si vous voulez utiliser les noyaux d'Alan Cox, vous devriez lire la liste de diffusion linux-kernel, et si c'est le cas vous savez où les trouver. Le site de référence est `http://www.uio.no/~trondmy/src/` , au cas où vous voudriez essayer de l'appliquer à un noyau 2.2 standard. Ce patch ne sera probablement pas intégré dans Linux 2.4, car il demande trop de changements dans le noyau pour être accepté dans le cycle de développement actuel. Attendez Linux 2.5.

`trondmy` propose des patchs pour utiliser NFS version 3 avec Linux, et qui permettent aussi d'utiliser TCP comme mécanisme de transport au lieu d'UDP. NFSv3 est très bien pour des réseaux grande distance ou avec des taux de pertes non nuls, ou des temps de latence élevés.

Si vous utilisez ces patchs, il vous faut lire linux-kernel, car de sales bugs, qui mangent vos fichiers, sont parfois découverts. Alors **soyez prudent**.

10.2 Le serveur

Le serveur NFS de Linux 2.2 et suivants est appelé `"knfsd"`. Il est difficile à configurer. Il faudra vous débrouiller tout seul ou utiliser ce que SuSE, RedHat et autres fournissent dans leurs paquetages 2.2. Désolé, mais vous pouvez toujours utiliser l'ancien nfsd. Il est lent mais facile à installer.

11 Serveur NFS sur une disquette

Cette section a été écrite par Ron Peters, *rpeters@hevanet.com* . Elle explique comment installer un serveur NFS en démarrant depuis une disquette. L'objectif initial était de partager par NFS un cédérom d'une autre machine pour installer Linux sur une machine sans lecteur de cédérom.

11.1 Introduction

Ce document a pour but d'aider ceux qui auront le même problème que moi récemment. J'installais un serveur Linux sur une machine sans lecteur de cédérom et sans moyen d'en installer un, à part peut être un SCSI externe. Ce genre de situations sera sans doute de plus en plus rare et ce document perdra donc de son intérêt, mais j'aurais bien aimé l'avoir quand j'essayais d'installer ma machine.

Vu que la machine n'avait pas de lecteur de cédérom, j'ai pensé installer un serveur NFS pour Win95 afin de partager le lecteur de cédérom juste le temps d'installer ma machine et de la mettre sur le réseau. Je n'ai trouvé que deux produits (je ne citerai pas les noms mais l'un est un freeware et l'autre avait une licence limitée à 14 jours), l'un ne marcha pas "clés en main" et l'autre n'était pas capable de gérer les conventions de nommage Linux suffisamment bien pour mener à bien l'installation.

J'ai donc décidé d'essayer de redémarrer ma machine Win95 sous Linux avec les disquettes boot/root et d'utiliser une disquette supplémentaire pour installer un serveur NFS.

Cela a été remarquablement simple, la procédure est en fait probablement plus simple que de lire cette introduction. Cependant, je pense qu'il est intéressant de rassembler les information nécessaires dans ce document.

11.2 Attentes

J'ai utilisé les disquettes boot/root fournies dans une des distributions de Slakware (InfoMagic developpers distributions). Le noyau utilisé sur les disquettes était un 2.0.34, et les programmes du serveur NFS venaient d'un serveur pour 2.0.30. J'ai toujours utilisé la méthode d'installation Slakware, non pas qu'elle soit plus facile ou meilleure ou pire, mais simplement qu'elle m'est familière et que je n'ai jamais pris le temps d'apprendre à en utiliser une autre.

Je ne pense pas qu'il puisse y avoir beaucoup de problèmes liés à la version du système. Je recommanderais simplement d'utiliser un système relativement récent, ce qui devrait être le cas si vous utilisez les disquettes boot/root de la distribution à installer.

11.3 Matériel nécessaire

- Une machine et une disquette de boot supportant le réseau. La machine devant jouer le rôle du serveur NFS doit avoir une carte réseau reconnue pendant le démarrage. Pour plus d'informations, voir le HOWTO sur le réseau (NET4-HOWTO).
- Une deuxième disquette contenant rpc.portmap, rpc.mountd et rpc.nfsd. Vous pouvez trouver facilement ces fichiers par un ftpsearch sur le ouèbe.
- Un cd (par exemple) Slakware (ou autre).

11.4 Configuration du serveur

11.4.1 Démarrer le serveur NFS temporaire

Démarrez la machine qui sera serveur NFS depuis la disquette de démarrage et assurez-vous que la carte réseau est reconnue, de même que le lecteur de cédérom. Dans la suite je suppose que la carte réseau en question est eth0.

11.4.2 Monter la disquette et le cédérom

Une fois que le système est démarré, vous n'avez plus besoin des disquette boot/root, le système étant complètement installé en disque mémoire. Remplacez la disquette root par la disquette supplémentaire, et montez la :

```
mount /dev/fd0 /floppy
```

Ceci fonctionne pour une disquette avec un système de fichiers ext2. J'imagine que la disquette pourrait utiliser un système de fichiers MSDOS mais je n'ai pas essayé. Je suppose que cela serait plus simple que de faire une image disque. Dans ce cas, il faudrait utiliser `mount -t msdos ...etc`.

Montez le cédérom :

```
mount -t iso9660 /dev/hdc /cdrom
```

J'ai utilisé les périphériques disquette et cédérom, on peut en utiliser d'autres selon ce que l'on veut faire. Les points de montage /floppy et /cdrom doivent exister sur l'image de la disquette root. Si ce n'est pas le cas, créez-les, ou bien vous pouvez utiliser n'importe quels autres points de montage.

11.4.3 Configurer le réseau sur le serveur provisoire

Il faut maintenant configurer le serveur NFS et le réseau. Il n'y a que quelques commandes à lancer, et quelques informations qu'il vous faudra rassembler auparavant (je donne ici des valeurs d'exemple) :

IPADDR:172.16.5.100 #L'adresse du serveur temporaire.

NETMASK:255.255.255.0 #Le masque de réseau (netmask).

BROADCAST:172.16.5.255 #L'adresse de diffusion sur le réseau

ETHNETWORK:172.16.5.0 #L'adresse réseau

GATEWAY:172.16.5.251 #Nécessaire seulement si vous avez une passerelle. Si c'est le cas, vous le savez. La plupart des réseau "à la maison" n'en ont pas.

Les commandes pour se connecter au réseau (utiliser les valeurs données ci-dessus) :

`ifconfig eth0 inet IPADDR arp netmask NETMASK broadcast BROADCAST`

`route add -net ETHNETWORK netmask NETMASK eth0`

Celle-ci uniquement si vous avez une passerelle et que vous devrez la traverser :

`route add default gw GATEWAY netmask 0.0.0.0 eth0`

Si tout va bien, vous êtes maintenant sur le réseau et devriez pouvoir faire des `ping` sur les autres machines.

11.4.4 Configurer le volume NFS

Choisissez le répertoire à partager. Dans mon exemple, c'était `/cdrom/slakware`. Placez-le dans le fichier `/etc/exports` :

`echo "/cdrom/slakware" > /etc/exports`

11.5 Lancer le serveur NFS

Allez dans `/floppy/usr/bin` et lancez :

`./rpc.portmap`

`./rpc.mountd`

`./rpc.nfsd`

11.5.1 Prêt, commencez l'installation

Normalement, le répertoire `/cdrom/slakware` est maintenant partageable. Démarrez votre machine (celle à installer) depuis les disquettes boot/root (j'ai utilisé les mêmes qui ont servi à démarrer le serveur) et commencez l'installation.

Quand il faut choisir le média source à utiliser, choisissez "serveur NFS". Il vous demandera l'adresse IP du serveur, qui est celle que vous avez appelé IPADDR pour le serveur. Il vous faut aussi donner le répertoire à monter, qui est celui que vous avez indiqué dans le fichier `/etc/exports` du serveur.

Le volume NFS devrait maintenant être monté, surveillez l'apparition de messages d'erreur. Si tout va bien, continuez l'installation.

11.6 Problèmes

11.6.1 Rien ici pour l'instant

Je n'ai rien à dire à ce sujet pour le moment. Peut être si des gens utilisent cette procédure, on aura des choses à ajouter.

11.7 À faire

11.7.1 Disquette DOS

Voir si la disquette supplémentaire peut être au format DOS.

11.7.2 Commandes RPC

Vérifiez l'ordre dans lequel lancer les commandes rpc.* et si toutes sont nécessaires.

12 PC-NFS

Vous ne voulez pas utiliser PC-NFS, mais plutôt samba.

Samba est bien meilleur que PC-NFS, il fonctionne avec "Windows3 for Workgroups" et les versions suivantes de Windows. Il est plus rapide et plus sur. Utilisez plutôt samba.