NFS HOWTO

Nicolai Langfeldt `janl@linpro.no`
Vertaler: *Reggy Ekkebus*, `reggy@zeelandnet.nl`

v1.0, 1 oktober 1999

HOWTO hoe stel je NFS servers en clienten in.

Inhoudsopgave

1 Voorwoord

1.1 Legaal materiaal

 De FAQ sectie is gebaseerd op een NFS FAQ gemaakt door Alan Cox. De Checklist sectie is gebaseerd op een mountprobleem checklist die gemaakt is door IBM Corporatie. De nfs-server-op-een-floppy sectie is geschreven door Ron Peters.

1.2 Disclaimer

Niet Nicolai Langfeldt, Ron Peters, noch hun werknemers of iemand anders, dragen verantwoordelijkheid voor wat er kan gebeuren als je de instructies in dit document opvolgt. Als je alle instructies zonder meer opvolgt, veel succes!

1.3 Feedback

Dit document is nooit klaar, dus hou me per mail op de hoogte van problemen en successen ter verbetering van deze HOWTO. Stuur geld, commentaar en/of vragen naar *janl@math.uio.no*, of, als het over NFS-server-op-een-floppy gaat, naar *rpeters@hevanet.com.* Als je een E-mailtje stuurt en je een wil antwoord laat dan wat simpele beleefdheid zien en *weet zeker* dat het return address klopt. Je hebt geen idee hoe veel antwoorden worden gebounced.

1.4 Ander materiaal

Als je deze HOWTO wilt vertalen laat me dat dan weten zodat ik bij kan houden in welke talen ik gepubliceerd ben :-).

Vervloekingen en Bedankjes naar Olaf Kirch die me zover kreeg dat ik deze HOWTO ging schrijven en me vervolgens goede suggesties gaf :-)

1.5 Opdracht

Deze HOWTO is opgedragen aan Anne Line Norheim Langfeldt, ondanks ze hem waarschijnlijk nooit zal lezen: zo'n meisje is ze niet. - Nicolai

2 LEESMIJ.eerst

NFS, het Netwerk File Systeem heeft drie belangrijke eigenschappen:

- Het maakt files delen over het netwerk mogelijk.
- Het werkt meestal goed genoeg.
- Het opent een vat aan veiligheidsrisico's die allemaal bekend zijn aan crackers, en die makkelijk te misbruiken zijn om toegang (lezen, schrijven en verwijderen) tot al je files te krijgen.

Ik zeg iets over beide onderwerpen in deze HOWTO. Lees in ieder geval de security sectie van de HOWTO en je zal minder kwetsbaar zijn voor domme veiligheidsrisico's. De passages over veiligheid zullen soms wat technisch zijn en vereisen kennis van IP netwerken en de gebruikte termen. Als je de termen niet herkent moet je terug naar af en de networking HOWTO lezen, ze overslaan of een boek over TCP/IP en netwerkadministratie lezen om je vertrouwd te maken met TCP/IP. Dat is so wie so een goed idee als je een UNIX/Linux machine beheert. Een erg goed boek hierover is *TCP/IP Network Administration* door Craig Hunt, gepubliceerd door O'Reilly & Associates, Inc. En als je dat gelezen en begrepen hebt heb je bovendien een hogere waarde op de arbeidsmarkt, dus je hebt niets te verliezen ;-)

Er zijn twee secties die je helpen met het oplossen van NFS problemen, te weten *Mount Checklist* en *FAQs*. Kijk daar in als iets niet werkt zoals het zou moeten.

Als je zelf de source wilt ophalen en compileren is de home-site voor de Linux 2.0 nfsd *ftp.mathematik.th-darmstadt.de:/pub/linux/okir*

Voor informatie over NFS onder 2.2 zie 10 (the Linux 2.2 section).

3 Een NFS server opzetten

3.1 Vereisten

Voordat je doorgaat met het lezen van de HOWTO moet je kunnen telnetten van de client naar de server en andersom. Als dat niet werkt moet je in de networking/NET-3 HOWTO kijken en je netwerk goed configureren.

3.2 Stap Een

Voordat we ook maar iets kunnen doen moeten we eerst een werkende NFS server hebben. Als je aangesloten bent op een afdelings- of universiteitsnetwerk zijn er meestal al NFS servers aanwezig. Als deze je toegang verschaffen, of als je deze HOWTO leest om tot een van ze toegang te krijgen dan kun je deze sectie over slaan en naar de sectie over 4 (hoe je een NFS client opzet) gaan.

Als je een NFS server onder een ander OS dan Linux wilt opzetten, lees dan de manuals om uit te vinden hoe je de NFS server opzet en hoe je het filesysteem exporteert met NFS. Er is een aparte sectie in deze HOWTO hoe je dat doet op vele andere systemen. Nadat je dat allemaal hebt uitgezocht kun je doorgaan met het lezen van de volgende sectie van deze HOWTO. Of je leest deze sectie verder omdat enkele van de dingen misschien hetzelfde zijn dan maakt de soort machine je gebruikt als server niet uit.

Als je linux 2.2 draait lees dan eerst 10 (de Linux 2.2 sectie) voor je doorgaat.

Degenen die nog steeds aan het lezen zijn moeten een aantal programma's installeren en configureren.

3.3 De portmapper

De portmapper in Linux heet `portmap` of `rpc.portmap`. De manpage op mijn systeem zegt dat het een "DARPA poort naar RPC programma nummer mapperïs. Het is de eerste security hole die je openzet als je deze HOWTO leest. Hoe je een van die holes weer kunt sluiten staat in 6 (de security sectie). Die ik je, alweer, vriendelijk doch dringend aanbeveel te lezen.

Start de portmapper. Dat is `portmap` of `rpc.portmap` en zou moeten staan in de `/usr/sbin` directory (op sommige machines heet hij rpcbind). In dit stadium kun je hem met de hand starten, maar hij moet elke keer als je opstart gestart worden, dus moet je een rc script maken/veranderen. Je rc scripts zijn nauwkeurig gedocumenteerd in de manpage `/etc/rc.d`, `/etc/init.d` of `/etc/rc.d/init.d`. Als er een script is dat zoiets als `inet` heet, is dit meestal het goede script om te veranderen. Maar wat te doen of te schrijven valt buiten het bereik van deze HOWTO. Start de portmapper, en kijk of hij het doet door `ps aux` te draaien en dan `rpcinfo -p`. Leeft hij? Goed.

Oh, een ding. Toegang tot je portmapper is geregeld door de inhoud van je `/etc/hosts.allow` en `/etc/hosts.deny` bestanden. Als `rpcinfo -p` faalt, maar je portmapper draait, kijk dan in deze files. Zie 6 (de security sectie) voor de details van deze files.

3.4 Mountd en nfsd

De volgende programma's die we draaiende moeten hebben zijn mountd en nfsd. Maar eerst veranderen we een andere file. `/etc/exports` deze keer. Ik wil het filesysteem in `/mn/eris/local` dat op de machine `eris` leeft beschikbaar stellen aan een machine genaamd `apollon`. In mijn `/etc/exports` op eris zet ik dan:

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

De hierboven gegeven regels geven apollon lees/schrijf toegang tot `/mn/eris/local`. In plaats van `rw` kan ik ook `ro` zeggen. ro staat voor read only (als je niets invult is de default read only). Er zijn andere opties die je mee kan geven, en ik zal enkele die te maken hebben met security later beschrijven. Ze zijn allemaal opgesomd in de `exports` man page die je minimaal een keer in je leven moet lezen. Er zijn ook betere manieren dan al je hosts in de exports file te zetten. Je kan bijvoorbeeld netgroepen gebruiken als je NIS (of NYS) (NIS was bekend als YP) draait en domeinwildcards en IP-subnets gebruikt als hosts die iets mogen mounten. Maar weet wie er toegang tot je systeem kan krijgen op de onbevoegde manieren als je zulke losse permissies gebruikt.

NB: Deze exports file heeft niet de zelfde syntax als andere Unixen. Er is een apparte sectie in deze HOWTO over andere Unixen en hun `exports` files.

Nu zijn we klaar om mountd te starten (of misschien heet dat `rpc.mountd`) en dan nfsd (deze kan `rpc.nfsd` heten). Zij lezen beide de exports file.

Als je `/etc/exports` verandert moet je zeker weten dat nfsd en mountd ook weten dat deze file veranderd is. De traditionele manier is `exportfs` draaien, maar veel Linux-distributies hebben geen exportfs programma. Als je geen exportfs hebt kun je in plaats daarvan dit script gebruiken:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exported file systems
```

Save het in, pak hem beet, `/usr/sbin/exportfs`, en vergeet het niet te `chmod a+rx`. Als je nu de exports file verandert, draai je daarna exportfs, als root.

Nu moet je checken of mountd en nfsd goed werken. Eerst met `rpcinfo -p`. Dat zou iets als dit moeten laten zien:

```
   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
```

Zoals je ziet heeft de portmapper aangekondigd dat hij in de lucht is, net als mountd en nfsd.

Als je `rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused`, `RPC_PROG_NOT_REGISTERED` of iets anders wat daar op lijkt krijgt, dan draait de portmapper niet. Of je hebt iets in `/etc/hosts.{allow,deny}` staan dat de portmapper antwoorden verbiedt, zie 6 (de security section) voor details van deze files. Als je krijgt `No remote programs registered.` dan wil de portmapper niet met je praten, of er is iets kapot. Kill nfsd, mountd en de portmapper en probeer het nog eens.

Behalve met rpcinfo of de portmapper draait kun je ook met ps kijken. De portmapper zal de services altijd blijven melden, zelfs als het programma dat ze aangeroepen heeft is gecrashed. Dus een ps check is slim als er iets is niet lijkt te werken.

Je moet natuurlijk de rc files van je systeem aanpassen om mountd en nfsd, net als de portmapper, als je boot te starten. Het kan goed dat de scripts al bestaan op je machine, je dan moet de kritieke sectie uncommenten of ze activeren in de goede runlevels.

Man pages die je nu moet kennen zijn: portmap, mountd, nfsd en exports.

Als je alles hebt gedaan zoals ik heb gezegd, kun je nu aan de NFS client sectie beginnen.

4 Een NFS client instellen

Eerst moet je een kernel hebben met het NFS file systeem erin gecompileerd of als module laadbaar. Dit is geconfigureerd voordat je de kernel compileerd. Als je nog nooit een kernel hebt gecompileerd moet je misschien in de Kernel HOWTO kijken en het uitvissen. Als je een hele coole distributie (zoals Red Hat) gebruikt en je hebt nooit gespeeld met de kernel of zijn modules (en dus geruineerd ;-), is nfs automagisch verkrijgbaar.

Je kan nu, op een root prompt, een geschikt commando intypen en het file systeem zal te voorschijn komen. Doorgaand met het voorbeeld in de vorige sectie willen we `/mn/eris/local` moeten van eris. Dat wordt gedaan met het commando:

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

(We komen terug op de rsize en wsize opties.) Het file systeem is nu verkrijgbaar onder `/mnt` en je kan daar naar `cd` en, en dan `ls` er in doen, en dan kijken naar de individuele files. Je zal merken dat het niet zo snel is als het lokale file systeem, maar meer makkelijker dan ftp. Als in plaats van het file systeem mounten mount zegt `mount: eris:/mn/eris/local failed, reason given by server: Permission denied`. Dan is de exports file fout, of je bent exportfs vergeten draaiten nadat je de exports file hebt geedit. Als het zegt `mount clntudp_create: RPC: Program not registered` betekend dat dat mountd en nfsd niet op de server draaien. Of je hebt het `hosts.{allow,deny}` probleem wat eerder is aangegeven.

Om van het filesysteem af te komen kun je het volgende doen:

```
umount /mnt
```

Om een nfs file systeem bij het booten te mounten moet je je `/etc/fstab` file aanpassen op de normale manier. Bijvoorbeeld een regel als deze is nodig:

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

Dat is alles wat er is, bijna alles. Lees verder.

4.1 Mount opties

Er zijn enkele opties die je misschien gelijk moet toevoegen. Ze bestuur de manier hoe een NFS client op een server crash of netwerk uitval reageert. Een van de coole dingen van NFS is dat NFS hier goed mee overweg kan. Als je de clients goed instelt. Er zijn twee duidelijke faal modes:

soft

De NFS client geeft een error aan het proces dat de file op het NFS gemounte file systeem gebruikt. Sommige programma's hanteren dit goed , de meeste niet. Ik kan deze setting niet aanbevelen, het is een recept voor slechte data en verloren data. Je moet dit zeker niet gebruiken voor mail disks —- als je mail waarde heeft.

hard

Het programma dat een file op een NFS gemount file systeem gebruikt zal hangen als de server crasht. Het process kan niet gekilld of geinterupt worden tenzij je `intr` specificeerd. Als de NFS server terug is zal het programma verder gaan waar het is gebleven. Ik beveel aan `hard,intr` te gebruiken op alle NFS gemounte file systemen.

Het vorige voorbeeld oppikken, dit is nu je fstab regel:

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

4.2 NFS optimaliseren

Normaal, als er geen rsize en wsize opties zijn gespecificeerd zal NFS lezen en schrijven met stukken van 4096 of 8192 byte. Sommige combinaties van Linux kernels en netwerk kaarten kunnen zulke grote blokken niet aan, en het hoeft niet optimaal te zijn. Dus we willen expirimenteren en een rsize en wsize vinden die werkt en zo snel mogelijk is. Je kan de snelheid van je opties testen met enkele simpele commando's. Geef het mount commando hierboven en kijk of je schrijf rechten hebt en doe deze test om de opeenvolgende schrijf snelheid te meten:

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

Dit maakt een file van 64MB met nul bytes (welke groot genoeg moet zijn zodat cashing niet een veel betekenend gedeelte is van de gemerkte snelheid, gebruik een grotere file als je veel geheugen hebt). Doe het een paar (5-10?) keer en kijk naar de gemiddelde tijd. Het is de 'elapsed' of 'wall clock' tijd die het meest interessant is in deze connectie. Dan kun je de lees snelheid meten door de file terug te lezen:

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

doe dat een paar keer en kijk naar het gemiddelde. Dan unmounten en opnieuw mounten met een grotere rsize en wsize. Dat moeten vemigvuldigingen van 1024 zijn, en niet groter dan 16384 byte sinds dat de maximale groote in NFS versie 2 is. Direct na het mounten met een grotere grote cd in het gemounte file systeem en doe dingen als ls, verken het fs een beetje ik kijk of alles is zoals het hoort. Als de rsize/wsize te groot is zijn de symptonen *erg* vreemd en niet 100% duidelijk. Een typisch symtoon is een incomplete file lijst als je 'ls' doet, en geen error bericht. Of files lezen faalt mysterieus zonder error bericht. Na het testen of de gegeven rsize/wsize werkt kun je de snelheids test opnieuw doen. Verschillende server platforms hebben waarschijnlijk een verschillende optimale grootte. SunOS en Solaris zijn naar men zegt veel sneller met blokken van 4096 byte dan met alle andere mogelijkheden.

Nieuwere Linux kernels (sinds 1.3) doen vooruitlezen voor rsizes hoger of gelijk aan de machine page grootte. Op Intel CPUs is de page grootte 4096 byte. Vooruit lezen zal veel betekenend cd NFS lees snelheid opvoeren. Dus op een Intel machine wil je 4096 byte rsize als dat mogelijk is.

Onthoud dat je de `/etc/fstab` update om de rsize/wsize in te vullen die je hebt gevonden.

Een truuk om de NFS schrijf snelheid te verhogen is door gelijktijdig schrijven op de server uit te zetten. De NFS specificatie verklaart dat NFS schrijf vragen niet beschouwd als af als de data die is geschreven op een niet veranderlijk medium (normaal de disk). Dit beperkt de schrijf snelheid een beetje, asynchroon schrijven zal NFS schrijven versnellen. De Linux nfsd heeft nooit gelijktijdig schrijven gedaan omdat de Linux file systeem implementatie dat niet toestaat, maar op niet linux servers kun je de snelheid vergroten op deze manier met dit in je exports file:

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

of zoiets dergelijks. Zie de exports man page op besproken machine. Onthoud dat dit het risico van data verlies verhoogd.

5 NFS over een trage verbinding

Trage lijnen zijn Modems, ISDN en mogelijk andere lange afstands connecties.

Deze sectie is gebaseerd op kennis over de gebruikte protocollen maar geen werkelijke experimenten. Laat het me weten als je dit probeert ;-)

Het eerste wat je moet weten is dat NFS een traag protocol is. Het heeft veel extra informatie nodig om het pakketje op z'n bestemming te krijgen. NFS gebruiken is bijna als kermit gebruiken om files te transporteren. Het is *traag*. Bijna alles is sneller dan NFS. FTP is sneller. HTTP is sneller. rcp is sneller. ssh is sneller.

Nog steeds vastbesloten om het te proberen? Ok.

De standaard parameters van NFS zijn voor snelle, korte tijds, lijnen. Als je deze standaard parameters gebruik over lange tijds, trage, lijnen kan dat er voor zorgen dat NFS problemen gaat aangeven, opdrachten opgeven , doen alsof files korter zijn dan dat ze eigenlijk zijn en NFS kan erg mysterieus doen.

Het eerste wat je moet doen is *niet* de `soft` mount optie gebruiken. Dit veroorzaakt timeouts die errors geven aan de software, welke, die de situatie waarschijnlijk niet goed hanteren. Dit is een goede manier om misterieuze fouten te krijgen. Gebruik daarom de `hard` mount optie. Als `hard` actief is zorgen timeouts voor eindeloos opnieuw proberen inplaats van annuleren. Dit is wat je wilt. Toch.

Het volgende dat gedaan moet is worden is goed instellen van de `timeo` en `retrans` mount opties. Er zijn beschrijvingen in de nfs(5) man page, maar hier is een kopie.

```
       timeo=n        De waarde in tienden van een seconde voor de
                            eerste hertransmissie na een RPC pauze. De
                            Standaard waarde is 7 tienden van een seconde.
                            Na de eerste pauze. Wordt de waarde verdubbeld
                      en bij elke keer dat het daarna nog faald wordt
                      de timeout verdubbeld totdat deze een waarden
                      van 60 seconden heeft bereikt of totdat het
                      aan hertransmissies is gehaald. De maximale
                      timeout is altijd 60 seconden. Een hogere timeout
                      waarde kan tot een snelheids verbetering lijden
                      als over een druk netwerk moet mounten, van een
                      langzame server moet mounten of als je door veel
                      routers of gateways door moet.

       retrans=n      Het aantal timeouts en hertransmisies die moeten
                      plaatsvinden. De standaard waarde is 3 timeouts
                      Als deze limiet gepaseerd is wordt het bericht
                      "server not responding" op de console weergegeven.
```

In andere woorden: Als er geen antwoord is binnen de 0.7 seconden (700ms) timeout zal de NFS client de aanvraag opnieuw doen en de tijd verdubbelen naar 1.4 seconden. Als er binnen die tijd geen antwoord komt wordt de aanvraag nog eens gedaan en de timeout wordt verdubbeld naar 2.8 seconden.

De snelheid van een lijn kan worden gemeten met ping met de zelfde pakket grootte als je rsize/wsize opties.

```
$ 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
```

De tijd hier is hoe lang het ping pakket er over deed om heen en terug te gaan naar lugulbanda. 15ms is best snel. Over een 28.000 bps lijn kun je iets verwachten als 4000-5000ms, en als de lijn belast is zal de tijd ook hoger zijn, gemakkelijk verdubbeld. Als deze tijd hoog is zeggen we dat er een 'high latency' is. Normaal, voor grotere pakketen en voor meer belastte lijnen zal de latency ook hoger zijn. Verhoog de `timeo` gepast voor je lijn en belasting. Als je ooit FTP wilt gebruiken samen met NFS moet je de ping tijden meten als je NFS aan het gebruiken bent om files te transporteren en verhoog `timeo` totdat deze gelijk is aan de latency van de lijn.

6 Security en NFS

Ik ben helemaal geen computer security expert. Maar ik heb een *klein* advies voor de security bewusten. Maar wees gewaarschuwd: Dit is bij lange na geen komplete lijst van de NFS gerelateerde problemen en als je denkt dat je veilig bent als je alles gelezen en geimplementeerd hebt heb je het mis.

Deze sectie is waarschijnlijk van geen belang als je op een *gesloten* netwerk zit met allemaal vertrouwde gebruikers en niemand die je niet vertrouwd kan op je netwerk inbellen, en het zou op geen manier verbonden met een ander netwerk waar je niet alle gebruikers vertrouwd. Vindt je dat ik parnoide klink? Dat ben ik helemaal niet. Dit is gewoon een *algemeen* veiligheids advies. En onthoud, de dingen die ik net heb gezegd is nog maar het begin. Een *veilige* site heeft een vlijtige en veel begrijpende admin die weet waar hij informatie kan vinden over aktuele en potentiele security problemen.

NFS heeft een basis probleem de client, zo niet anders om, vertrouwd de NFS server en omgekeert. Dit kan slecht zijn. Het betekend dat als het root account van de server is doorbroken kan het best simpel zijn om die van de client ook te doorbreken. En andersom. Er zijn een aantal kopie strategieen voor dit, waar we op terug komen.

Iets dat je moet lezen zijn de CERT adviezen over NFS, de meest tekst hieronder gaat over onderwerpen waar CERT adviezeurs over hebben geschreven. Zie *ftp.cert.org:/01-README* voor een lijst van CERT adviziezen. Hier zijn enkele NFS gerelateerde adviezen:

```
CA-91:21.SunOS.NFS.Jumbo.and.fsirand                        12/06/91
     Kwetsbaarheden betreffende Sun Microsystems, Inc. (Sun) Netwerk
     File systeem (NFS) en het fsirand programma. Deze kwetsbaarheden
     betreffen SunOS versies 4.1.1, 4.1 en 4.0.3 op alle architecturen.
     Patches zijn verkrijgbaar voor SunOS 4.1.1. Een eerste parch voor
     SunOS 4.1 is ook verkrijgbaar. Sun zal komplete patches beschikbaar
     stellen voor SunOS 4.1 en SunOS 4.0.3 later.
```

```
CA-94:15.NFS.Vulnerabilities                                   12/19/94
     Dit advies beschrijft veiligheid middelen om te bewaken tegen verschillende
     kwetsbaarheden in het Netwerk file systeem (NFS). De adviseur stond
     versteld van het stijgende aantal root doorbraken door indringers
     door het gebruik van tools om de kwetsbaarheden te misbruiken.

CA-96.08.pcnfsd                                                04/18/96
     Dit advies beschrijft de kwetsbaarheden in het pcnfsd programma (ook
     bekend als rpc.pcnfsd). Er zit een patch bij.
```

6.1 Client Veiligheid

Op de client kunnen we besluiten dat we de server niet te veel willen vertrouwen op een paar manieren dit doen we met mount opties. Bijvoorbeeld we kunnen suid programma's niet laten werken vanaf het NFS file systeem met de `nosuid` optie. Dit is een goed idee en je moet overwegen dit te gebruiken bij elke met NFS gemounte disk. Dit betekend dat de root gebruiker van de server geen suid-root programma kan maken op het file systeem en dan inlogd als een normale gebruiker op de client om dan het suid programma te gebruiken om ook root op de client te worden. We kunnen ook het uitvoeren van files op het NFS gemounte file systeem verbieden met de `noexec` optie. Maar dit is niet zo handig als de `nosuid` optie sinds omdat het normaal is dat een filesysteem nog een paar scriptjes of programmas inhoud die uitgevoerd moeten worden. Je vult deze opties in in de opties kolom, met de `rsize` en `wsize` optie, gescheiden door komma's.

6.2 Server security: nfsd

Op de server kunnen we beslissen dat we het root account op client niet vertrouwen. Dit kunnen we doen door het gebruik van de root_squash optie in exports:

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

Nu, als een gebruiker met UID 0 op de client probeert toegang (lezen, schrijven, verwijderen0 te krijgen op het filesysteem van de server veranderd de server het UID op de server naar het 'nobody' account. Wat betekend dat root op de client geen toegang krijgt tot files die alleen root op de server toegang tot kan krijgen. En je zou root_squash op alle geexporteerde filesystemen moeten gebruiken. Maar root op de client kan nog steeds elke gebruiker worden met het su commando en zo alsnog de file editten. Dit heeft een belangrijke betekenis: Alle belangrijke binaries en files moet van `root` zijn, en niet `bin` of een ander niet root account, sinds dat het enige account is dat root gebruiker op de client niet kan gebruiken. In de NFSd man page zijn er nog enkele andere squash opties zodat je iedereen kan wantrouwen die je wilt wantrouwen op de client. Je hebt ook de mogelijkheid om een bepaalt UID en GID bereik te squashen. Dit is beschreven in de Linux NFSd man page.

root_squash is eigenlijk standaard met Linux NFSd. Je kan root toegang geven tot je filesysteem met de `no_root_squash` optie.

Iets anders belangrijks is zeker te weten dat nfsd checkt om alle zijn aanvraag komt van een bevoorrechte poort. Als het aanvragen toestaat van elke oude poort op de client kan een gebruiker zonder speciale priveleges een programma verkregen van het internet makkelijk uitvoeren. Het praat tegen het nfs porotocol en zegt elke gebruiker te zijn die hij maar wil zijn. Griezelig. De linux nfsd checkt dit automatisch, bij een ander OS moet je dit zelf aanzetten. Dit is beschreven in de nfsd man page van je OS.

Iets anders. Exporteer nooit een file systeem naar 'localhost' of 127.0.0.1. Vertrouw me.

6.3 Server security: de portmapper

De basis portmapper, in combinatie met nfsd heeft een design probleem dat het mogelijk maakt om files te krijgen om de NFS server zonder enige priveleges. Gelukkig is de portmapper die de meeste Linux distributies gebruiken relatief veilig tegen deze aanval, en kan nog veiliger gemaakt worden door het configureren van toegangslijsten in twee files.

Niet alle Linux distributies zijn identiek gemaakt. Sommige schijnbare bij de tijdse distributies heben *geen* beveiligbare portmapper, zelfs niet vandaag de dag, vele jaren nadat de kwetsbaarheid werd ontdekt. Op z'n minst een distributie heeft de man page voor een beveiligbare portmapper maar de echte portmapper kan *niet* beveiligd worden. De makkelijkste manier om dit na te kijken is met het commando strings en kijk of hij de relevante files leest, `/etc/hosts.deny` en `/etc/hosts.allow`. Aannemend dat je portmapper `/usr/sbin/portmapper` is. Je kunt met dit commando kijk of je portmapper beveiligbaar is: `strings /usr/sbin/portmap | grep hosts`. Op mijn machine komt er dit:

```
/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
```

Eerst editten we `/etc/hosts.deny`. De volgende regel moet er in staan

```
portmap: ALL
```

wat toegang voor *iedereen* ontzegt. Nu portmapper gesloten is gaan we eerst even kijken of dat echt zo is met `rpcinfo -p`. rpcinfo zou als het goed is geen uitvoer moeten geven of misschien een fout bericht. Portmapper opnieuw opstarten zou *niet* nodig moeten zijn.

De portmapper afsluiten voor iedereen is een beetje drastisch, dus we openen hem weer door het veranderen van de `/etc/hosts.allow` file. Maar eerst moeten we uitvinden wat we erin gaan zetten. Het zijn normaal alle machines die toegang moeten hebben tot de portmapper. Normaal zijn er erg weinig machines die toegang moeten hebben voor elke reden. De portmapper onderhoud nfsd, mountd, ypbind/ypserv, pcnfsd, en de 'r' diensten zoals ruptime en rusers. Van deze zijn alleen nfsd, mountd, ypbind/ypserv en misschien pcnfsd van belang. Alle machines die daar toegang tot moeten hebben moeten dat ook hebben. Laten we zeggen dat het adres van je machine 129.240.233.254 is en dat hij leeft op het subnet 129.240.223.0 moet er toegang tot hebben (dat zijn termen die zijn geinteresseert door de networking HOWTO, ga terug en fris je geheugen op als dat nodig is). Dan schrijven we

```
portmap: 129.240.223.0/255.255.255.0
```

in `hosts.allow`. Dit is het zelfde netwerk adres als dat wat je geeft aan route en het subnet mask welke je aan ifconfig geeft. Voor het apparaat `eth0` op deze machine zou `ifconfig` dit moeten geven

```
...
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
...
```

en `netstat -rn` zou dit moeten geven

```
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
...
```

(Netwerk adres in de eerste kolom).

De `hosts.deny` en `hosts.allow` files worden bescreven in de gelijknamige man pagina's.

BELANGRIJK: Zet *niets* anders dan *IP NUMMER* in de portmap regels van deze files. Host naam lookups kunnen inderict lijden tot portmap activiteit veroorzaken welke hostnaam lookups opwekt wat er indirect portmap activiteit veroorzaakt welke hostnaam lookups opwekt....

Het bovenstaande zou je server vaster moeten maken. Het enige overblijvende probleem (Ja, juist) is iemand die root verbreekt (of MS-DOS opstart) op een vertrouwde machine en het recht gebruikt om vragen te sturen van de veilige poort als elke gebruik willen ze zijn.

6.4 NFS en firewalls

Het is een erg goed idee om je nfs en portmap poorten te firewallen in je router of firewall. De nfsd opereerd op poort 2049, beiden udp en tcp protocollen. De portmapper op poort 111, tcp en udp, en mountd op port 745 en 747, tcp en udp. Normaal. Moet je de poorten checken met het `rpcinfo -p` commando.

Als je wilt dat NFS door de firewall gaat zijn er opties in de nieuwere NFSds en mountds om ze een specifieke (niet standaard) poort te laten gebruiken welke geopend kan worden in de firewall.

6.5 Samenvatting

Als je de hosts.allow/deny, root_squash, nosuid en privileged port gebruikt in de portmapper/nfs software vermijd je veel van de nu bekende bugs in nfs en kan bijna veilig voelen over *dat* ten minste. Maar nog steeds, na dat alles: Als een indringer toegang heeft tot je netwerk, s/he kan hij vreemde commandos laten verscheinen in je `.forward` of leest je mail als `/home` of `/var/spool/mail` is geexporteerd met NFS. Om de zelfde reden, moet je nooit je PGP prive sleutel benaderen over nfs. Of op z'n minst zou je het risico moeten kennen. En je weet er een beetje van.

NFS en de portmapper maken een complex subsysteem en daarom is het helemaal niet vreemd dat nieuwe bugs worden gevonden, in het basis design of in de implementatie die wij gebruiken. Er zijn misschien zelfs gaten die jij kent, welke iemand anders misbruikt. Maar dat is leven. Om op de hoogte te blijven van zulke dingen lees dan op z'n minst de nieuwsgroepen *comp.os.linux.announce* en *comp.security.announce* als een minimum.

7 Mount lijst

Deze sectie is gebaseerd op IBM Corp.s "NFS mount problem checklist". Veel dank aan hun voor het verkrijgbaar maken voor deze HOWTO. Als je een probleem hebt met het mounten van een NFS filesysteem kijk dan eerst in deze lijst voordat je je probleem post. Elk punt beschrijft een fout en de oplossing.

1. Mount blijft zeggen`RPC: Program not registered`
   Draait portmapper?
   Oplossing: Start het.
   Draait mount?
   Oplossing: Start het.
   Draait nfsd?
   Oplossing: Start het.
   Mag de portmapper geen antwoord geven door `/etc/hosts.deny`?
   Oplossing: Verwijder de regel in `hosts.deny` of voeg een regel toe aan `hosts.allow` zodat de portmapper tegen je mag praten.

2. File systeem niet geëxporteerd, of niet geëxporteerd aan de client in kwestie.
   Oplossing: Exporteer het.

3. Naam resolutie komt niet overeen met de exports lijst.
   e.g.: export lijst zegt exporteer aan `johnmad` maar `johnmad` z'n naam is beslist als `johnmad.austin.ibm.com`. worden de mount rechten afgewezen.
   Oplossing: Exporteer aan beide vormen van de naam.
   Het kan ook gebeuren als de client twee apparaten heeft met verschillende namen voor elke van de twee en de file noemt er maar een.
   Oplossing: exporteer beide apparaten.
   Dit kan ook gebeuren als de server niet lookuphostbyname of lookuphostbyaddr kan doen (dit zijn bibliotheek functies) op de client. Weet zeker dat de client `host` <`name`>; `host` <`ip_addr`>; kan doen en dat beiden de zelfde machine laten zien.
   Oplossing: Maak de naam resolutie goed.

4. Het file systeem is gemount nadat NFS was gestart (op die server). In dat geval exporteert de server het onderliggende mount punt, niet het gemounte filesysteem.
   Oplossing: Sluit NFSd af en start het opnieuw op.
   Notitie: De clienten die het onderliggende mount punt hebben gemount krijgen problemen na het herstarten.

5. De datum is helemaal verkeerd op een van de machines (dit kan make in de war sturen)
   Oplossing: Zorg dat de datum goed staat.
   De HOWTO auteur raad NTP aan om de klokken gelijk te zetten. Sinds er export beperkingen zitten op NTP in de VS moet je NTP voor Debian, Red Hat of Slackware halen van

   `ftp://ftp.hacktic.nl/pub/replay/pub/linux` of een mirror.

6. De server kan geen mount accepteren van een gebruiker die in meer dan 8 groepen zit.
   Oplossing: verlaag het aantal groepen waar een gebruiker in zit of mount via een andere gebruiker.

8 FAQ's

Dit is de FAQ sectie. Het is gedeeltelijk gebaseerd op een oude NFS FAQ geschreven door Alan Cox.

Als je een probleem hebt met het mounten van een filesysteem kijk dan of je probleem is beschreven in de "Mount Checklist" sectie.

1. Ik krijg veel "stale nfs handle" fouten als ik Linux gebruik als een nfs server.

   Dit wordt veroorzaakt door een bug in sommig oudere nfsd versies. Het is verbeterd in nfs-server2.2beta16 en later.

2. Als ik een file systeem probeer te mounten krijg ik

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

   Je gebruikt waarschijnlijk een Caldera systeem. Er is een bug in de rc scripts. Neem contact op met Caldera om een fix te krijgen.

3. Waarom kan ik een file niet uitvoeren na het te kopieeren op de NFS server?

   De reden is dat nfsd open file handles cached voor snelheids redenen (onthoud, het draait in gebruikers ruimte). Zolang nfsd een file open heeft (als in het geval na erin te schrijven), staat de kernel je niet toe het uit te voeren. nfsds nieuwer dan lente 95 laten open files vrij na een paar seconden, oudere houden ze tot zelfs een paar dagen.

4. Mijn NFS files zijn allemaal read only

   De Linux NFS server staat standaard op alleen lezen. Lees de sectie over "mountd en nfsd" en "filesystemen exporteren" in deze HOWTO, en kijk in de "exports" en "nfsd" man paginas. Je moet de `/etc/exports` veranderen.

5. Ik mount van een Linux NFS server en als `ls` werkt kan ik geen files lezen en schrijven.

6. Op oudere versies van Linux moet NFS server met `rsize=1024,wsize=1024` mounten.

7. Ik mount van een Linux NFS server met een block size van tussen 3500-4000 en het crasht de linux computer geregeld

   Doe het dan niet. Dit gebeurt niet met 2.0 en 2.2 kernels. En zo ver ik me herinner is er ook geen probleem met 1.2.

8. Kan Linux NFS doen over TCP

9. Nee, momenteel niet

10. Ik krijg ontzettend veel vreemde fouten als ik een machine probeer te mounten vanaf een Linux box.

    Weet zeker dat je gebruikers in 8 groepen of minder zitten. Oudere servers willen dit.

11. Als ik mijn machine reboot hangt hij soms als ik een gecrashte NFS server unmount.

    Unmount NFS servers **niet** als je reboot of halt, sla ze gewoon over, het kan geen kwaad als je ze niet unmount. Het commando is `umount -avt nonfs`.

12. Linux NFS clients zijn erg traag als ze schrijven naar SUN en BSD systemen.

    NFS schrijft is normaal synchroon (je kan dit uitzetten als het niet uitmaakt dat soms je data verliest). BSD afgeleide kernels neigen niet instaat te zijn te werken met kleine blokken. Dus als 4K data van een Linux box doet BSD dit

    ```
    lees 4K page
    verander 1K
    schrijf 4K terug naar de schijf
    lees 4K page
    verander 1K
    schrijf 4K page terug naar de schijf
    etc..
    ```

13. Als ik veel clienten verbind met mijn Linux NFS server daalt de snelheid ineens.

    Het NFS protocol gebruikt gefragmenteerde UDP pakketen. De kernel heeft een limiet van hoeveel fragmenten of incomplete pakketen het kan hebben voordat hij er weg gaat gooien. In 2.2 is dit looptijd veranderbaar via het /proc/ filesysteem: `/proc/sys/net/ipv4/ipfrag_high_thresh` en `ipfrag_low_thresh`. In 2.0 voor het compileren te veranderen in `.../linux/net/ipv4/ip_fragment.c`, `IPFRAG_HIGH_THRESH` en `IPFRAG_LOW_TRESH`. De betekenis van deze waarden is dat wanneer het geheugen gebruikt van niet samengestelde UDP fragmenten de “ipfrag_high_thresh” in bytes bereikt (256K standaard in 2.2.3 en 2.0.36) wordt het terug gebracht naar “ipfrag low tresh” in een keer. Dit wordt gedaan door het weggooien van fragmenten. Dit ziet er bijna uit als pakket verlies, en als de hoge drempel is bereikt zakt de snelheid van je server.

    256K is genoeg voor minder dan 30 clienten. Als je er 60 hebt, verdubbel het dan. En verdubbel dan ook de lage drempel.

14. Ik gebruik Linux 2.2 (of later) met knfsd en ik mijn AIX, IRIX, Solaris, DEC-Unix, ... machine niet zover krijgen om hem te mounten.

    Knfsd geeft aan dat het NFS versie 3 gebruikt. Dat doet hij niet. Er is een optie dit uit te zetten. Gebruik deze. Of je kan "vers=2"bij de clienten in de mount opties lijst zetten.

15. Mijn AIX 4 machine kan mijn Linux NFS server niet mounten. Zegt hij.

    ```
    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.
    ```

    of iets als dit.

    AIX 4.2 used reserved ports (<t1024) for NFS. AIX 4.2.1 and 4.3 are not constrained to reserved ports. Also, AIX 4.2.1 and 4.3 try to mount using NFS3, then NFS/TCP, then fiannly NFS/UDP.

    Dit

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

    aan het einde van rc.tcpip toevoegen forceert hem gereserveerde poorten te gebruiken opnieuw. (Deze tip is gegeven door Brian Gorka)

9 Filesystemen exporteren

De manier om filesystemen te exporteren met NFS is niet compleet het zelfde met verschillende platformen natuurlijk. In dit geval zijn Linux en Solaris 2 de afwijkende. Deze sectie laat de oppervlakkige manier zien hoe je het moet doen op de meeste systemen. Als je systeem niet word besproken moet je de man pagina’s van je OS bekijken. Sleutelwoorden zijn nfsd, system administration tool, rc scripts, boot scripts, boot sequence, /etc/exports, exportfs. Ik gebruik een voorbeeld in deze sectie: Hoe /mn/erix/local te exporteren aan apollon lezen/schrijven.

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

Deze Osen zijn traditioneel Sun exporteer formaat. In `/etc/exports` schrijf:

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

De complete documentatie is in de `exports` man pagina. Na het veranderen van de file draai `exportfs -av` om de filesystemen te exporteren.

Hoe strikt het exportfs commando is over de syntax variaties. Op verschillende Osen zul je uitvinden dat de vorig ingevoerde regels lezen:

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

of zelfs afgetakelt als dit:

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

Ik raad je aan formeel te zijn. Je riskeert dat de volgende versie van `exportfs` veel strickter is en dan stopt ineens alles met werken.

9.2 Solaris 2

Sun heeft compleet opnieuw het wiel uitgevonden toen ze Solaris 2 deden. Dus dit is helemaal anders dan bij alle andere Osen. Wat je doet is de `/etc/dfs/dfstab` file editten. Daarin zet je share commando's als gedocumenteerd in de `share` (1M) man pagina. Als dit:

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

Na het editten daarvan draai je het programma `shareall` om het filesysteem te exporteren.

10 NFS onder Linux 2.2

Als ik dit schrijf is Linux 2.2.12 de aktuele kernel versie en om NFS te gebruiken eronder kan het een beetje een dagelijkse taak zijn. Of niet.

Wat de status van NFS in Linux 2.4 zal zijn weet ik niet.

Het grote nieuwe ding in Linux 2.2 is ondersteuning voor in de kernel nfs server deamon, genoemd knfsd in 2.2. Deze manier van implementatie van nfsd heeft enkele voordelen, de belangrijkste is de snelheid. Een Linux 2.2 machine met knfsd is een respectabele nfs server. Je kan nog steeds de oude nfsd gebruiken met Linux 2.2 en er zijn nog enkele voordelen van dit te gebruiken, het is simpeler.

Als je een kernel source of binary package gebruikt gemaakt door iemand zoals RedHat (6.0 en later), SuSE (6.1 of later, geloof ik) of een ander professioneel systeem integrator hebben ze waarschijnlijk volle "knfsd"functionaliteit geintergreert in hun kernel en je moet je geen zorgen maken, het werkt. Meestal. Totdat je zelf een kernel compileert. Als je een standaard Linux 2.2 kernel gebruik (vanaf 2.2.12) zal knfsd breken.

Om dit zelf voor elkaar te krijgen heb je het H.J. Lus knfsd package nodig. Dit is een collectie patches, en de benodigde tools voor 2.2 die Lu onderhoud in zijn vrij tijd. Je kan het halen van je lokale kernel mirror, de standaard site is *ftp.kernel.org:/pub/linux/devel/gcc/*. **Dit is niet bedoeld voor algemene consumptie**. Als je dit pakket verwarrend vind probeer het dan alsjeblieft niet zelf. Wacht tot er een package komt van je favoriete systeem intergrator (Red Hat, SuSE of ...) komt.

Zend me ook geen vragen over dit, ik kan je niet helpen. Ik heb geen knfsd gebaseerde servers draaien. Als je fouten of verzuimingen vindt in dit document, schrijf me dan dan verander ik de HOWTO en breng hem opnieuw uit.

Nog steeds aan het lezen? Ok. H.J.Lu post nieuwe versies van dit pakket op de linux-kernel mailing lijst. Andere onderwerpen die te maken hebben met NFS in 2.2 worden daar ook gepost. Lees het.

Er is een interressant ding op te merken over het knfsd package. Het zegt dat het NFS versie 3 ondersteund. Dat doet het niet. Er is een optie om 'm te laten stoppen dit te laten zeggen, of op de clients kun je "vers=2öpgeven in de mount opties lijst.

10.1 De client

De client is bijna simpel. Om goede locking te krijgen moet je statd (van het knfsd package) gecompileerd krijgen, geinstalleerd en gestart uit je opstart scripts. Doe dat. Statd heeft een directory nodig genoemd /var/lig/nfs om te functioneren anders stopt hij er mee zonder een fout bericht, dus die dir moet je maken voordat je het draait.

Als statd draait kun je het testlk programma gebruiken (in tools/locktest om te testten of het locken van een file op een NFS gemount filesysteem werkt. Het moet. Als het print *No locks available* werkt statd niet.

Eigenlijk kun je locken geheel vermijden (niet dat ik dat aanraad), door het geven van de "nolocköptie in de mount optie lijst.

Zo ver ik weet is dit alles wat je aan de praat moet krijgen op de client.

Oh, als je een Sparc of Alpha NFS server hebt zul je uitvinden dat de nfsclient in Linux 2.2 absoluut kl*te is. De transfer snelheid van en naar de server is zo slecht ... dat je het je niet kan voorstellen. Het is veel slechter dan onder Linux 2.0. Veel. Maar hier is natuurlijk een fix voor. De Alan Cox series van 2.2 kernels (die iets meer expirementeel is dan de normale 2.2 kernels van Linus) hebben een patch om Linux 2.2 het goed te laten doen met Alpha en Sparc servers. Als je de Alan Cox versie van de 2.2 kernels wilt gebruiken moet je de linux-kernel mailing list lezen en als je weet waar je patch kan vinden. Hier is de home page van deze patch <http://www.uio.no/~trondmy/src/>, in het geval dat je het wilt proberen op een gewone 2.2 kernel. Deze patch zal misschien niet in de 2.4 kernel zitten, omdat het te veel veranderingen aan de kernel vereist om geaccepteerd te worden in de ontwikkelings cyclus. Wacht op Linux 2.5.

trondmy heeft ook patches om Linux NFS versie 3 te laten gebruiken, dit stelt je ook in staat om tcp te gebruiken als transport mechanisme in plaats van UDP. NFSv3 is erg goed voor grote netwerken en andere netwerken waar het pakket verlies niet nul is of waar de wachttijden groot zijn.

De reden waarom je de linux-kernel mailing lijst moet lezen om deze patches te gebruiken is omdat voordat er slechte bugs worden ontdekt die je files op eten. Dus wees gewaarschuwd.

10.2 De server

De nfs server deamon onder Linux 2.2 en later is "knfsd"genoemd. Het is tricky om het in te stellen. Je moet dit allemaal zelf uitzoeken, of gebruik SuSE, Red Hat of een andere die 2.2 kernel pakketjes maakt. Je kan nog steeds Linux 2.2 gebruiken. Het is langzaam maar makkelijk in te stellen.

11 NFS server op een floppy

Deze sectie is geschreven door Ron Peters, *rpeters@hevanet.com* Het legt uit hoe je een NFS server instelt als je van een floppy boot. Het was eigenlijk bedoeld om instaat te zijn een cdrom van een niet-linux/unix machine over NFS te exporteren om op een andere machine die geen CDRom heeft Linux te installeren.

11.1 Introductie

Dit document is gemaakt voor de mensen die in dezelfde problemen raken als waar ik laatst ook in kwam. Ik was een Linux server aan het bouwen op een machine die geen CDROM had en ook geen mogelijkheid had er een CDROM in te steken alleen een externe SCSI of zoiets. Zulke machines komen natuurlijk niet zoveel meer voor. Daarom kan het zijn dat dit deel van de howto niet zoveel waard is. Alhoewel ik het wel waardeerde toen ik op die machine Linux ging installeren.

Omdat mijn machine geen cdrom drive had, dacht ik ik moet een NFS server voor Windows 95 vinden om zo mijn cdrom lang genoeg te exporteren om mijn computer te installeren en op mijn netwerk te krijgen. Van de twee producten die ik heb gevonden (Ik noem geen namen maar een was freeware en de andere had 14 dag gelimiteerde licentie), een werkte niet op mijn computer en de andere kon geen Linux namen aan.

Ik heb dan geprobeert mijn Win95 machine op te starten met een boot/root set van diskettes om dan een suplementaire diskette te gebruiken om een NFS server in te stellen.

Dit was redelijk simpel en de procedure is waarschijnlijk simpeler dan het lezen van deze intro.

11.2 Verwachtingen

Dit document is gemaakt door het gebruik van de boot/root diskettes van een van de aktuele ontwikkelings distributie van Slackware. Ik gebruikte kernel versi 2.0.34 voor de boot/root diskettes, maar de NFS server programma's zijn gepakt van een 2.0.30 server. Ik heb altijd de Slackware installatie methode gebruikt, niet omdat het makkelijker, better of slechter is, maar gewoon omdat ik het gewend ben en nog geen tijd heb gehad om een andere methode te proberen.

Ik geloof niet dat er veel problemen gaan zijn als je dit gebruikt met een andere versie van het OS. Ik raad aan iets redelijk recent te gebruiken. Omdat een boot/root set wordt gebruikt voor de installatie is het niet zo moeilijk om die te maken.

11.3 Benodigdheden

- Systeem met netwerk ondersteuning en een opstart diskette. Het systeem dat NFS server wordt moet een netwerk kaart hebben die wordt herkend tijdens het opstarten. Meer informatie over dit kan worden gevonden in de Networking HOWTO.
- Een tweede floppy met rpc.portmap, rpc.mountd en rpc.nfsd. Deze files kunnen gemakkelijk gevonden worden op het web.
- Slackware of een andere distributie (moet op cd zijn).

11.4 Server instellingen

11.4.1 Boot de NFS server

Boot het NFS server systeem van de opstart diskette en zorg dat de netwerk kaart wordt herkent. Het is ook nodig dat de CDROM wordt herkend. Ik gebruik eth0 als een voorbeeld netwerk kaart.

11.4.2 Mount de diskette en cdrom

Zodra het systeem is opgestart zijn de boot/root diskettes niet meer nodig. Het systeem is geheel opgeslagen in het RAM geheugen.

Verwissel de root diskette met de aanvullende diskette. Mount de diskette:

mount /dev/fd0 /floppy

Dit veronderstelt dat er een ext2 file systeem type aanwezig is op de floppy. Ik denk dat de aanvullende diskette ook een DOS diskette kan zijn, maar dat heb ik nog niet geprobeert. In dit geval zou het mount -t msdos ...etc moeten zijn. Dit moet eigenlijk in de todo sectie.

Mount de cdrom:

mount -t iso9660 /dev/hdc /cdrom

De diskette en cdrom zijn de apparaten die ik heb gebruikt. Deze kunnen anders zijn afhankelijk van je bedoelingen. De mount punten /floppy en /cdrom bestaan op de root diskette dus die kunnen worden gebruikt. Als ze niet bestaan maak ze dan aan of neem een ander mount punt.

11.4.3 Het netwerk instellen op de NFS server

Dit is waar de NFS server wordt ingesteld om het het netwerk te betreden. Er moeten maar een paar commandos worden gestart. Er zijn een paar stukjes informatie die je nodig gaat hebben voordat je de commandos kan uitvoeren (de waarden zijn voorbeelden):

IPADDR:172.16.5.100 #Dit is het adres van de NFS server

NETMASK:255.255.255.0 #Dit is het netmask

BROADCAST:172.16.5.255 #Alleen het laatste nummer (255) is anders dan bij IPADDR

ETHNETWORK:172.16.5.0 #Het zelfde verhaal, alleen is het laatste nummer nu een 0

GATEWAY:172.16.5.251 #Alleen nodig als je een gateway hebt. Je weet het waarschijnlijk wel. De meest thuis netwerken hebben geen gateway.

De commandos om op het netwerk te komen. Zet de waarden van hierboven erin:

ifconfig eth0 inet IPADDR arp netmask NETMASK broadcast BROADCAST

route add -net ETHNETWORK netmask NETMASK eth0

Gebruik het volgende commando alleen als je een gateway hebt en je er door moet:

route add default gw GATEWAY netmask 0.0.0.0 eth0

Als alles goed gaat, zou je nu op het netwerk moeten zitten en instaat moeten zijn om de andere computers te pingen.

11.4.4 De NFS share instellen

Bepaal de directory die je wilt share met NFS. In het geval van mijn voorbeeld is dat de /cdrom/slakware directory. Zet de directory in de /etc/exports file:

echo /cdrom/slakware"> /etc/exports

11.5 Start de NFS server

Ga naar /floppy/usr/sbin en voer de volgende commando's uit:

./rpc.portmap

./rpc.mountd

./rpc.nfsd

11.5.1 Klaar, start de installatie

Dit zou de /cdrom/slakware"directory moeten exporteren in de /etc/exports file. Als dit is gedaan, kun je de machine die moet worden geinstalleerd opstarten van de boot/root diskettes (Ik gebruikte de zelfde als waar ik de NFS server mee heb geboot) en start de installatie.

Als je klaar bent om de bron lokatie aan te geven, kies de NFS server optie. Het vraagt om het ip adres van de server. Geef het het adres dat je gebruikte als IPADDR voor de NFS server. Het vraagt ook naar de directory die moet worden gemount. Dit is de directory die je in /etc/exports op de NFS server hebt gezet.

Het systeem mount dan de NFS server. Kijk uit naar elk fout bericht. Alles zou goed moeten gaan en je kan verder met de installatie.

11.6 Problemen oplossen

11.6.1 Nog niets

Ik heb nog geen informatie om problemen op te lossen. Als mensen deze procedure gebruiken, zullen er meer tips en hints zijn.

11.7 Te Doen

11.7.1 DOS Disk

Schrijf een Dos diskette uit voor de suplimentary disk.

11.7.2 rpc commandos

Een specifieke volgorde van het draaien van rpc.* commando's uitvinden en of alle of maar een paar moeten gedraaid worden.

12 PC-NFS

Je wil PC-NFS niet draaien. Je wil samba draaien.

Samba is veel beter dan PC-NFS en het werkt met Windows 3 voor Workgroups en latere versies van Windows. Het is sneller en nog veiliger ook. Gebruik het. Echt.