NFS HOWTO

Nicolai Langfeldt janl@math.uio.no

Version 1.0, 1 Ottobre 1999

COME installare NFS su client e server. Traduzione di kevin@arena.sci.univr.it.

Indice

1 Introduzione

1.1 Note legali

1.2 Liberatoria

Né Nicolai Langfeldt, né Ron Peters, né i relativi datori di lavoro o chiunque altro, si assumono la responsabilità di quanto può accadere se seguirete le istruzioni di questo documento. Se stabilite di seguire comunque le istruzioni, buona fortuna!

1.3 Commenti e critiche

Questo non sarà mai un documento finito, inviatemi una mail se incontrate problemi o successi, potreste contribuire a migliorare questo documento. Inviate denaro, commenti e/o domande a *janl@math.uio.no* oppure a *rpeters@hevanet.com*

nel caso di un server NFS relativo ai floppy. Se inviate una e-mail *accertatevi* che l'indirizzo di risposta sia corretto e funzionante. Ricevo molte mail e accertarmi di ogni indirizzo sarebbe troppo oneroso.

1.4 Altri argomenti

Se desiderate tradurre questo HOWTO fatemelo sapere così terrò traccia di quante lingue hanno pubblicato il documento :-)

Ringrazio Olaf Kirch che mi ha fornito spunti e suggerimenti per questo scritto :-)

1.5 Dediche

Questo HOWTO è dedicato ad Anne Line Norheim Langfeldt. Probabilmente non lo leggerà mai perché non è quel tipo di ragazza.

2 LEGGIMI

NFS (Network File System), il sistema di condivisione dei dischi via rete, ha tre caratteristiche importanti:

- Rende possibile la condivisione dei file in rete.
- Funziona abbastanza bene.
- Apre dei buchi di sicurezza che sono conosciuti dai cracker e facilmente utilizzabili per ottenere accesso (lettura, scrittura e cancellazione) a tutti i vostri file.

Parlerò di questi argomenti nel presente HOWTO. Accertatevi di leggere la sezione relativa alla sicurezza e sarete in grado di limitare al minimo la vulnerabilità e i rischi. La sezione sulla sicurezza contiene dei passi piuttosto tecnici e richiede alcune conoscenze di reti IP e i termini a esse relativi. Se non capite la terminologia utilizzata fate un passo indietro e leggete il Networking HOWTO oppure acquistate un libro relativo all'amministrazione di reti TCP/IP per familiarizzare con TCP/IP. È comunque una buona idea se amministrate macchine UNIX/Linux. Un ottimo libro al riguardo è *TCP/IP Network Administration* di Craig Hunt, pubblicato da O'Reilly & Associates, Inc. Dopo averlo letto e capito, avrete un maggior valore sul mercato del lavoro :)

Ci sono due sezioni per aiutarvi nella soluzione dei problemi di NFS, chiamate *Elenco di verifica di mount* e *FAQ*. Fate riferimento a esse se qualcosa non dovesse funzionare come previsto.

Il sito primario per nfsd per Linux 2.0 è `ftp.mathematik.th-darmstadt.de:/pub/linux/okir`, nel caso aveste bisogno di scaricarlo e compilarlo.

Per informazioni su NFS in relazione a Linux 2.2 fate riferimento a 10 (NFS in Linux 2.2).

3 Configurare un server NFS

3.1 Prerequisiti

Prima di continuare a leggere questo HOWTO dovete essere in grado di fare telnet tra due macchine che avete intenzione di configurare come client e server. Se non siete in grado di farlo, leggete il networking/NET-2 HOWTO per installare e configurare correttamente la rete.

3.2 Il primo passo

Prima di fare qualsiasi altra cosa, abbiamo bisogno di configurare un server NFS. Se fate parte di un dipartimento o università probabilmente ce ne saranno molti altri già configurati. Se avete accesso a tali server o state leggendo questo HOWTO per utilizzarli, non avete bisogno di leggere questa sezione e potete passare direttamente alla sezione 4 (Configurazione di un client NFS)

Se avete bisogno di installare NFS su una macchina che non abbia Linux, allora dovete leggere il manuale di sistema per scoprire come abilitare il servizio di NFS ed esportare i file tramite NFS. C'è una sezione apposita in questo HOWTO su come farlo su molti sistemi diversi. Dopo aver configurato tutto, potete passare alla sezione successiva. Oppure leggete altre parti di questa sezione poiché alcune cose che saranno dette potrebbero essere interessanti anche per altri sistemi, indipendentemente dal tipo di macchina che volete usare come server.

Se siete di fretta, fate riferimento alla sezione 10 (NFS in Linux 2.2) prima di continuare a leggere questo HOWTO.

Quelli di voi che continueranno a leggere, avranno bisogno di configurare alcuni programmi.

3.3 Il portmapper

Il portmapper su Linux può chiamarsi sia `portmap` sia `rpc.portmap`. La pagina man sul mio sistema dice che è un DARPA port to RPC program number mapper. Questo è il primo buco di sicurezza che aprite. La descrizione per chiudere alcuni di questi buchi è nella sezione 6 (Sicurezza e NFS), che vi consiglio di leggere con urgenza.

Avvio del portmapper. Lo si può fare in due modi: `portmap` oppure `rpc.portmap` e li dovreste trovare nella directory `/usr/sbin` (su alcune macchine si chiama rpcbind). Per ora lo potete avviare manualmente, ma è necessario che venga lanciato ogni volta che si riavviia la macchina e per questo motivo dovrete creare/modificare i vostri script rc. Gli script rc sono descritti in maggior dettaglio nella pagina man di init, in genere si trovano in `/etc/rc.d`, `/etc/init.d` oppure `/etc/rc.d/init.d`. Se c'è uno script che ha il nome simile a `inet` probabilmente è lo script giusto da modificare. Ciò che dovete scriverci va oltre lo scopo di questo HOWTO. Avviate portmap e controllate che esso sia correttamente partito con il comando `ps aux`. È partito? Bene.

Ancora una cosa. L'accesso remoto al portmapper è regolato dal contenuto dei file `/etc/hosts.allow` e `/etc/hosts.deny`. Se `rpcinfo -p` fallisce, ma l'esecuzione del portmapper continua, esaminate questi file. Verificate la sezione 6 (Sicurezza e NFS) per dettagli su questi file.

3.4 Mountd e nfsd

I programmi successivi che devono essere avviati sono mountd e nfsd. Ma prima dobbiamo modificare un altro file. Questa volta `/etc/exports`. Supponiamo che io voglia che il file system `/mn/eris/local`, che risiede su `eris`, possa essere disponibile anche sulla macchina chiamata `apollon`. Dobbiamo quindi mettere queste righe nel file `/etc/exports` di eris:

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

Le righe sopra indicate consentono l'accesso in lettura e scrittura a `/mn/eris/local`. Invece di `rw` potremmo mettere `ro` che indica l'accesso in sola lettura (se non si mette nulla, è di default a sola lettura). Ci sono altre opzioni che è possibile inserire e ne discuteremo alcune relative alla sicurezza più avanti. Le opzioni sono tutte elencate nella pagina man di `exports` che dovreste leggere almeno una volta nella vita. Ci sono modi migliori che elencare gli host nel file exports. Potete, per esempio, usare net groups se state utilizzando le NIS (o NYS), e potete sempre specificare domini interi oppure sottoreti IP come host autorizzati a montare qualcosa. Ma dovreste considerare che qualcuno non autorizzato potrebbe accedere al server se usate questo tipo di autorizzazioni.

Nota: la sintassi del file exports non è la stessa di altri Unix. C'è una sezione separata in questo HOWTO che riguarda il file `exports` di altri Unix.

Ora siamo pronti per lanciare il comando mountd (oppure può chiamarsi `rpc.mountd`), successivamente nfsd (che potrebbe chiamarsi `rpc.nfsd`). Entrambi leggono il file exports.

Se modificate il file `/etc/exports` accertatevi che nfsd e mountd sappiano che il file è stato cambiato. Il modo tradizionale consiste nel lanciare `exportfs`. Molte distribuzioni non hanno il programma exportfs, allora si può installare questo script sulla macchina:

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

Salvatelo chiamandolo `/usr/sbin/exportfs` e non dimenticate di modificare le autorizzazioni con il comando `chmod a+rx`. Ora, ogni volta che modificate il file exports, lanciate exportfs come root.

Dovreste quindi controllare che mountd e nfsd stiano girando correttamente. Prima con `rpcinfo -p`. Dovrebbe mostrare qualcosa di simile al seguente:

```
   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
```

Come si vede il portmapper ha avviato i propri servizi e così pure mountd e nfsd.

Se invece appare l'errore: `rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused`, `RPC_PROG_NOT_REGISTERED` o qualcosa di simile, significa che il portmapper non sta girando oppure c'è qualcosa nel file `/etc/hosts.{allow,deny}` che impedisce al portmapper di rispondere. Fate riferimento alla sezione 6 (Sicurezza e NFS) per dettagli su questi file. Se appare il messaggio

`No remote programs registered.` significa che il portmapper non vuole rispondere oppure qualcosa non funziona. Interrompete nfsd, mountd e il portmapper e riprovate la sequenza di avvio dall'inizio.

Dopo avere controllato che il portmapper riporti i servizi, si può controllare anche con ps. Il portmapper continuerà a riportare i servizi anche dopo che il programma che li ha utilizzati termina in maniera non corretta, per cui controllare con il ps può essere utile se sembra che qualcosa non funzioni.

Naturalmente, avrete bisogno di modificare i file rc per avviare mountd e nfsd e il portmapper quando si avvia la macchina. È probabile che gli script esistano già sulla macchina, dovrete solo togliere il commento dalle parti interessate oppure modificare il livello di init affinché queste vengano attivate.

Le pagine man che dovrebbero esservi familiari adesso, sono quelle di portmap, mountd, nfsd ed exports.

Bene, se avete fatto tutto esattamente come ho detto probabilmente è tutto pronto per iniziare a lavorare sul client NFS.

4 Configurazione di un client NFS

Prima di tutto avete bisogno di un kernel che abbia il supporto per NFS compilato staticamente oppure come modulo. Questo lo si configura prima di iniziare la compilazione. Se non avete mai compilato un kernel prima, leggete il Kernel HOWTO. Se state usando una buona distribuzione (come Red Hat [meglio Debian N.d.T]) e non avete mai messo mano al kernel o ai moduli (rovinandolo ;)), allora nfs dovrebbe essere già disponibile.

Ora, dal prompt di root, potete lanciare il comando appropriato e vedere il file system apparire. Continuando l'esempio della sezione precedente, vogliamo montare la partizione in `/mn/eris/local` da eris. Ciò è fatto con il comando:

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

(Torneremo successivamente sulle opzioni rsize e wsize). Il file system è ora disponibile sotto `/mnt`, potete fare `cd` in esso e con un `ls` vedere tutti i file che vi sono presenti. Noterete che non è così veloce come su un file system locale, ma molto più conveniente che usare ftp. Se, invece di montare il file system, il comando mount produce un errore come `mount: eris:/mn/eris/local failed, reason given by server: Permission denied`, significa che il file exports è errato oppure avete dimenticato di lanciare il comando exportfs dopo averlo modificato. Se invece dice `mount clntudp_create: RPC: Program not registered` significa che nfsd o mountd non sta girando sul server oppure c'è un problema nel file `hosts.{allow,deny}` di cui abbiamo parlato precedentemente.

Per rimuovere il filesystem, usare il comando:

```
umount /mnt
```

Per fare in modo che il sistema monti un file system al boot, occorre modificare il file `/etc/fstab`. Per il nostro esempio aggiungere la seguente riga:

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

Questo è tutto, più o meno.

4.1 Opzioni del comando mount

Ci sono alcune opzioni che dovreste considerare almeno una volta. Controllano il modo in cui i client NFS gestiscono i crash della rete o del server. Uno dei fattori positivi di NFS è che questi problemi vengono gestiti molto bene... se configurate i client in modo corretto. Ci sono due tipi di problemi:

soft

I client NFS sono responsabili di riportare l'errore al processo che sta accedendo a un file su un file system montato. Alcuni programmi gestiscono la segnalazione, altri no. Non raccomando l'uso di questo parametro, poiché è rivolto ai file corrotti e ai dati persi. È meglio non utilizzarlo in particolare per i dischi di posta — se ci tenete alla vostra posta.

hard

Il programma che cerca di accedere a un file su un file system NFS si bloccherà quando il server ha un crash. Il processo non potrà essere interrotto a meno che non si specifichi il parametro `intr`. Quando il server NFS torna in linea, il programma riprenderà a funzionare correttamente. Questo è probabilmente il funzionamento che si vorrebbe. Raccomando di usare `hard,intr` su tutti i file system montati via NFS.

Riprendendo l'esempio precedente, modifichiamo la linea dell'fstab:

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

4.2 Ottimizzare NFS

Normalmente, se non vengono usate le opzioni rsize e wsize, NFS leggerà e scriverà blocchi di 4096 o 8192 byte. Alcune combinazioni di kernel di Linux e schede di rete possono non essere in grado di gestire blocchi così grandi o potrebbero non esserlo comunque in maniera ottimale. Quindi dobbiamo provare a sperimentare varie dimensioni per determinare quali siano i parametri che funzionano e garantiscono le migliori prestazioni. Si può provare l'influenza delle opzioni sulla velocità con alcuni semplici comandi. Se avete montato la partizione come visto precedentemente e avete i diritti di scrittura, potete provare con il seguente test di scrittura sequenziale:

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

In questo modo si crea un file di 64 MB di zeri (grande abbastanza per fare in modo che l'uso della cache non sia significativo sulle prestazioni; usate una dimensione maggiore se avete molta memoria disponibile). Lanciatelo alcune volte (5-10?) e calcolate il tempo medio. Il tempo da tenere maggiormente in considerazione è quello indicato con 'elapsed' oppure 'wall clock'. Potete quindi verificare le prestazioni in lettura:

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

Fatelo alcune volte e calcolate la media dei tempi. Quindi smontate e rimontate la partizione nuovamente ma con rsize e wsize maggiori. Dovrebbero essere sempre multipli di 1024 e non essere più grandi di 16384, che è la dimensione massima ammessa da NFS versione 2. Dopo averla montata nuovamente, fate un cd nel file system montato e provate qualche semplice comando tipo ls, esplorate il file system per verificare se tutto funziona correttamente. I sintomi di rsize/wsize troppo grandi, sono *molto* strani e per nulla ovvi. Un sintomo tipico è un elenco incompleto di file quando viene fatto un 'ls', senza alcun messaggio di errore. Oppure la lettura dei file fallisce miseramente senza messaggi di errore. Dopo avere stabilito che le dimensioni di rsize e wsize funzionano, potete provare di nuovo a effettuare i controlli. Server diversi hanno dimensioni ottimali diverse. SunOS e Solaris hanno la reputazione di andare molto più veloci con blocchi di 4096 byte.

I kernel di Linux più recenti (dal 1.3), eseguono un read-ahead per rsize di dimensioni maggiori o uguali alla dimensione della pagina della macchina. Sui processori Intel, la dimensione della pagina è di 4096 byte. Poiché l'uso del read-ahead aumenta *significativamente* le prestazioni in lettura di NFS, raccomando di impostare a 4096 le dimensioni di rsize.

Ricordatevi di modificare `/etc/fstab` per riflettere le dimensioni di rsize/wsize che avete trovato essere le migliori.

Un trucco per accelerare le prestazioni in scrittura di NFS è quello di disabilitare la scrittura in sincronia sul server. Le specifiche di NFS controllano che le richieste di scrittura sul server non siano considerate terminate finché i dati non siano memorizzati su un supporto non volatile (il disco). Questo limita le prestazioni in scrittura, per cui disabilitando questa caratteristica si otterrà un incremento delle prestazioni. Il nfsd di Linux non ha più fatto scritture sincrone da quando non lo fa nemmeno il file system stesso. Su macchine non Linux, è possibile migliorare le prestazioni modificando in questo modo il file /etc/exports:

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

o in modo simile. Fate riferimento alla pagina man relativa al file exports della macchina in questione. Da notare che ciò aumenta la possibilità di perdita di dati.

5 NFS su linee lente

Le linee lente includono Modem, ISDN e tutte le altre connessioni su lunga distanza.

Questa sezione si basa sulla conoscenza dei protocolli usati, ma non su prove reali poiché non ho modo di verificarli. Fatemi sapere le vostre esperienze se avete la possibilità di provare ;-)

La prima cosa da ricordare è che NFS è un protocollo lento. Ha un grosso overhead di sistema. Usare NFS è come usare il kermit per trasferire i file. È veramente *lento*. Quasi tutto è più veloce di NFS. FTP, HTTP, rcp e ssh sono più veloci.

Siete ancora convinti di volerlo provare? OK.

I parametri predefiniti di NFS sono per linee veloci e con bassa latenza. Se usate questi parametri su linee ad alta latenza si potrebbero verificare errori, operazioni non portate a termine, file che risultano essere più corti di quanto siano in realtà e altri fatti misteriosi.

La prima cosa da fare è di *non* usare l'opzione per il mount `soft`. Questo potrebbe fare in modo che i timeout restituiscano errori alle applicazioni, che potrebbero non gestirli correttamente. Quella appena descritta potrebbe essere la causa di misteriosi fallimenti. Usate invece l'opzione `hard`. Quando l'opzione `hard` è attiva, i timeout generano infiniti tentativi invece di terminare l'operazione che il software voleva effettuare. Ed è ciò di cui avete bisogno.

La prossima cosa da fare è ingannare le opzioni `timeo` e `retrans`. Sono descritte nella pagina man nfs(5), che è qui riportata [tradotta NdT]:

```
       timeo=n        Il valore, in decimi  di  secondo  prima di
                      tentare  una  ritrasmissione  dopo  un  RPC
                      timeout.  Il  valore  di  default  è  di  7
```

	decimi di secondo. Dopo il primo timeout, il timeout viene raddoppiato dopo ogni successivo timeout fino a un massino di 60 secondi oppure nel caso avvenga un timeout maggiore. Inoltre, se il filesystem è montato in modo hard, ogni nuovo timeout avrà come valore di partenza, il doppio del valore di partenza della sequenza di timeout precedente, che si raddoppia a ogni ritrasmissione. Il massimo timeout rimane di 60 secondi. Le migliori prestazioni si raggiungono incrementando il valore di timeout quando si monta un disco su una rete lenta, su un server lento oppure attraverso router e gateway.
retrans=n	Il numero di timeout minori e ritrasmissioni che si devono verificare prima che si verifichi un timeout maggiore. Il valore di default è di 3 timeout. Quando si verifica un timeout maggiore, viene bloccata l'operazione sul file e sulla console appare "server not responding".

In altre parole: se non viene ricevuta una risposta entro il timeout di 0,7 secondi (700ms) il client NFS ripeterà la richiesta raddoppiando il timeout a 1,4 secondi. Se non si riceve risposta entro 1,4 secondi la richiesta viene ripetuta ancora e il timeout viene raddoppiato ancora a 2,8 secondi.

La velocità di una linea può essere misurata con un ping con le dimensioni del pacchetto e di rsize/wsize uguali.

```
$ 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
```

Il tempo qui è quanto impiega il pacchetto del ping ad andare avanti e indietro da lugulbanda. 15 ms è abbastanza veloce. Su una linea a 28.800 bps ci si può aspettare qualcosa come 4000-5000ms, e se la linea è molto carica questo tempo sarà ancora più alto, anche doppio. Quando il tempo è elevato, si dice che la linea ha elevata latenza. Generalmente per pacchetti più grandi e per linee cariche, il tempo tende ad aumentare. Aumentate il parametro `timeo` per adattarlo alla velocità della linea e al carico. E poiché la latenza aumenta se si usa la linea per altre operazioni, se volete provare FTP e NFS allo stesso momento, provate a misurare i tempi del ping mentre usate NFS e FTP per trasferire i file e aumentare `timeo` perché corrisponda alla latenza della linea.

6 Sicurezza e NFS

Non sono un esperto di sicurezza, ma sono *abbastanza* coscio dei problemi di sicurezza. Ma attenzione: questo non è certo un elenco completo dei problemi legati a NFS e se pensate di essere sicuri una volta che avrete letto e implementato ciò che è scritto qui, avrei giusto giusto un ponte da vendervi.

Questa sezione non è probabilmente di utilità se vi trovate in un una rete chiusa, dove conoscete gli utenti e nessuno che non conoscete può accedere alle macchine della rete. Ovvero, non ci dovrebbero essere modi per entrare nella rete in dialin (via modem) e non dovrebbe essere collegata ad altre reti di cui non conoscete gli utenti. Pensate che io sia paranoico? Non lo sono per nulla. Questo è solo un aiuto di *base* sulla sicurezza. E ricordate, le cose che dico qui sono solo l'inizio. Un sito *sicuro* ha bisogno di un amministratore diligente che sappia dove trovare informazioni e tutti i problemi relativi alla sicurezza.

NFS ha un problema di sicurezza di base per cui il client, se non specificato altrimenti, si fida del server NFS e viceversa. Questo può essere negativo. Significa che se l'account di root viene compromesso sul server, viene compromesso anche quello di tutti i client e viceversa. Ci sono alcune strategie di copia per questo, sulle quali torneremo poi.

Dovreste leggere gli avvisi del CERT su NFS, molto di ciò che è qui scritto deriva dai messaggi scritti dal CERT. Vedere *ftp.cert.org/01-README*

per un elenco aggiornato degli avvisi del CERT. Ecco qui alcuni avvisi relativi a NFS.

```
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 Sicurezza del client

Sul client possiamo decidere di non volere dare troppa fiducia al server in un paio di modi con delle opzioni del mount. Per esempio possiamo vietare l'uso di programmi SUID su partizioni NFS con l'opzione `nosuid`. Questa è una buona idea e dovreste considerarne l'uso con tutti i dischi che montate via NFS. Significa che gli utenti root del server non possono fare programmi SUID-root sul file system, entrare come utenti normali sui client e usare i programmi SUID per diventare root anche sui client. Possiamo anche vietare l'esecuzione di file sulla partizione montata usando l'opzione `noexec`, ma è sicuramente meno pratico di `nosuid`, poiché è naturale pensare che una partizione debba avere dei file eseguibili o degli script. Potete inserire queste opzioni nella colonna opzioni con `rsize` e `wsize` separati da virgole.

6.2 Sicurezza del server: nfsd

Sul server possiamo decidere che non vogliamo dare fiducia all'account root dei client. Possiamo farlo usando l'opzione root_squash nel file exports:

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

Ora, se l'utente con UID 0 sul client cerca di accedere (lettura, scrittura, cancellazione) al file system, il server sostituisce l'UID con quello dell'utente 'nobody' del server. Ciò significa che il root dei client non può accedere o modificare i file del server che solo root può accedere o modificare. Ciò rappresenta un fatto positivo e probabilmente dovreste usare `root_squash` su tutti i file system che esporte. Potrebbe sorgere il dubbio che l'utente root dei client può usare il comando 'su' per diventare un altro utente e accedere e cambiare i file di quell'utente!. La risposta è affermativa. Questo è ciò che avviene e quello che deve avvenire su Unix e NFS e ha un'importante conseguenza: i file importanti devono essere di proprietà di `root` e non di `bin` o altri utenti non root, poiché il solo utente a cui l'utente root dei client non può accedere è l'account root del server. Nella pagina man di NFSd ci sono molte altre opzioni squash e potete decidere di non dare fiducia a qualsiasi utente dei client. È possibile anche applicare lo squash a gruppi di UID o GID. Tutto ciò è descritto nella pagina man di NFSd.

root_squash è realtà un'opzione di default con Linux NFSd, per garantire accesso come root ai filesystem, utilizza l'opzione `no_root_squash`.

Un'altra cosa importante è assicurarsi che nfsd controlli che tutte le richieste provengano da una porta privilegiata. Se accetta richieste da qualsiasi porta, un utente senza privilegi particolari potrebbe lanciare un programma facilmente ottenibile su Internet che comunica con il server nfs e gli fa credere di essere un utente qualsiasi. L'nfsd di Linux fa questo controllo per default, ma su altri sistemi operativi dovrete effettuare questo controllo per contro vostro, che dovrebbe essere descritto nella pagina man di nfsd del sistema operativo usato.

Un'altra cosa: mai esportare un filesystem a 'localhost' o 127.0.0.1. Credetemi.

6.3 Sicurezza del server: il portmapper

Il portmapper di base ha problemi se usato con nfsd che rendono possibile ottenere file dal server NFS senza alcun privilegio. Fortunatamente il portmapper che viene distribuito con Linux è relativamente sicuro contro questo attacco e può essere reso maggiormente sicuro configurando l'elenco degli accessi in due file.

Non tutte le distribuzioni di Linux sono uguali. Alcune distribuzioni apparentemente aggiornate *non* includono un portmapper sicuro nemmeno oggi, a distanza di molti anni da quando il problema è stato reso noto. Almeno una distribuzione contiene la pagina man per un portmapper sicuro, ma il portmapper reale *non* è sicuro. Un modo facile per controllare se il vostro portmapper è valido oppure no, è quello di lanciare il comando strings(1) per verificare se il portmapper legge i file `/etc/hosts.deny` e `/etc/hosts.allow`, importanti per gestire la sicurezza. Posto che il portmapper è `/usr/sbin/portmap` si può controllarlo con il seguente comando: `strings /usr/sbin/portmap | grep hosts`. Sulla mia macchina il risultato è:

```
/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
```

Per prima cosa modifichiamo il file `/etc/hosts.deny`. Dovrebbe contenere la riga:

```
portmap: ALL
```

che nega l'accesso a *chiunque*. Poiché l'accesso è chiuso, provate il comando `rpcinfo -p` per verificare se il vostro portmapper legge e obbedisce realmente a questo file. rpcinfo non dovrebbe dare alcun risultato oppure un messaggio di errore. Non dovrebbe essere necessario riavviare il portmapper.

Chiudere il portmapper a chiunque è un po' troppo drastico, quindi riapriamolo modificando il file `/etc/hosts.allow`, ma prima cerchiamo di capire che cosa inserirci. Il file dovrebbe semplicemente avere un elenco di tutte le macchine a cui si vuole garantire accesso al portmapper. Ci sono comunque pochissimi casi di macchine che hanno bisogno di un accesso totale per qualsiasi ragione. Il portmapper amministra nfsd, mountd, ypbind/ypserv, pcnfsd e i servizi 'r', come ruptime e rusers. Di questi, solo nfsd, mountd, ypbind/ypserv e forse pcnfsd possono avere qualche conseguenza. Tutte le macchine che richiedono accessi ai servizi della vostra macchina dovrebbero essere autorizzate a farlo. Supponiamo che l'indirizzo della vostra macchina sia 129.240.223.254 e che si trovi in una sottorete 129.240.223.0 che deve poter accedere ai servizi della macchina (questa è la terminologia introdotta da Networking HOWTO: se non vi ricordate rileggetelo). Scriviamo quindi:

```
portmap: 129.240.223.0/255.255.255.0
```

in `hosts.allow`. Questo è quanto viene scritto anche come subnet mask in ifconfig. Per il dispositivo `eth0` di questa macchina, `ifconfig` dovrebbe essere:

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

e `netstat -rn` dovrebbe mostrare:

```
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
...
```

(l'indirizzo di rete è nella prima colonna).

I file `hosts.deny` e `hosts.allow` sono descritti nelle pagine man con lo stesso nome.

IMPORTANTE: *non* inserite *nient'altro* che *NUMERI IP* nelle linee del portmapper di questi file. La risoluzione dei nomi può indirettamente causare attività del portmapper che può causare attività del portmapper che può indirettamente causare attività del portmapper...

Ciò che abbiamo visto rende il server più chiuso. Il solo problema che ora rimane (eh già :)) è che qualcuno riesca a corrompere la shell di root (o che riesca a far partire la macchina con un floppy MS-DOS) su una macchina affidabile e, usando questo privilegio, spedisca da una porta sicura richieste come utente qualsiasi.

6.4 NFS e firewall

È una buona cosa proteggere con un firewall le porte di nfs e del portmapper sul vostro router o firewall. L'nfsd funziona sulla porta 2049, con i protocolli udp e tcp. Il portmapper lavora in genere sulla porta 111, sia tcp sia udp e mountd sulle porte 745 e 747, sia tcp sia udp. Controllate le porte usate con il comando `rpcinfo -p`.

Se invece NFS deve attraversare un firewall, ci sono delle opzioni su nfsd e mountd più recenti per usare porte non standard che possono essere tenute aperte attraverso il firewall.

6.5 Riassunto

Se usate hosts.allow/deny, root_squash, nosuid e l'opzione per le porte privilegiate nei programmi portmapper/nfs, evitate la maggior parte dei bug conosciuti di nfs e potete essere *abbastanza* sicuri. Comunque ci sono altri problemi: se qualcuno ha accesso alla rete può far apparire strani comandi nel vostro `.forward` o leggere la vostra posta se le directory `/home` o `/var/spool/mail` sono esportate via NFS. Per la stessa ragione, non dovreste mai lasciare le vostre chiavi private di PGP su un disco esportato via NFS. O almeno ora sapete i rischi che correte.

NFS e il portmapper formano un sottosistema complesso e non è improbabile che nuovi bug vengano scoperti, sia nella progettazione sia nell'implementazione. Ci possono sempre essere buchi di cui qualcuno sta abusando. Ma questa è la vita. Per tenervi aggiornati su questo genere di problemi dovreste leggere alcuni newsgroup come *comp.os.linux.announce* e *comp.security.announce* .

7 Elenco di verifica di mount

Questa sezione è basata sull'elenco di verifica dei problemi di NFS mount di IBM Corp. I miei ringraziamenti a loro per averlo reso disponibile per questo HOWTO. Se rilevate alcuni problemi montando un file system via NFS, controllate questo elenco prima di inviare il vostro problema. Ogni elemento descrive un problema e come risolverlo.

1. Il mount continua a indicare `RPC: Programma non registrato`

   Il portmapper è attivo?

   Soluzione: avviarlo.

   Mountd è attivo?

   Soluzione: avviarlo.

   Nfsd è attivo?

   Soluzione: avviarlo.

   Il portmapper non può rispondere a causa di `/etc/hosts.deny`?

   Soluzione: rimuovere la regola in `hosts.deny` o aggiungere una regola a `hosts.allow` per fare in modo che il portmapper possa comunicare con voi.

2. Il file system non viene esportato o non viene esportato al client in questione:

   Soluzione: esportarlo.

3. La risoluzione del nome non corrisponde all'elenco di exports

   Esempio: la lista di export dice di esportare a `johnmad` ma il nome di `johnmad` è risolto in `johnmad.austin.ibm.com` e quindi il mount viene negato.

Soluzione: inserire nell'export entrambi i nomi.

Può anche accadere se il client ha due interfacce con nomi diversi e l'exports ne specifica una sola.

Soluzione: inserire nell'exports entrambe le interfacce.

Può anche accadere se il server non riesce a eseguire le funzioni lookuphostbyname o lookuphostbyaddr sul client. Accertatevi che il client possa fare host <name>; host <ip_addr>; e che entrambe mostrino la stessa macchina.

Soluzione: controllare la risoluzione dei nomi.

4. Il file system è stato montato dopo che l'NFS è partito (sul server). In questo caso il server sta esportando la directory sottostante NFS e non il filesystem esportato.

   Soluzione: arrestare e riavviare NFSd.

   Nota: I client che abbiano già montato il filesystem avranno problemi dopo il riavvio del server.

5. La data è sbagliata su una o entrambe le macchine (ciò può dare problemi usando il comando make)

   Soluzione: correggere la data.

   L'autore dell'HOWTO raccomanda di usare NTP per sincronizzare gli orologi. Poiché ci sono delle restrizioni su NTP negli USA, occorre prelevare le versioni per Debian, Red Hat o Slackware da ftp://ftp.hacktic.nl/pub/replay/pub/linux o qualche mirror.

6. Il server non accetta il mount da un utente che è in più di 8 gruppi.

   Soluzione: diminuire il numero di gruppi cui appartiene o usa un altro utente.

8 FAQ

Questa è la sezione delle FAQ. È basata su una vecchia FAQ di NFS di Alan Cox.

Se si verificano problemi montando un filesystem, verificate se il vostro problema è descritto nella sezione Elenco di verifica di mount.

1. Ho un gran numero di errori 'stale nfs handlè quando uso Linux come server NFS

   Ciò è causato da un bug in alcune vecchie versioni di nfsd. È stato corretto a partire da nfs-server2.2beta16

2. Quando provo a montare un filesystem ottengo

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

   Probabilmente state usando un sistema basato sulla distribuzione Caldera. C'è un bug negli script rc. Contattate Caldera per ottenere la versione corretta.

3. Perché non posso eseguire un file dopo averlo copiato sul server NFS?

   Il fatto è che nfsd tiene una cache dei file aperti per motivi di prestazioni (ricordate, gira in un ambiente utente). Mentre nfsd ha un file aperto (come nel caso di una scrittura), il kernel non ne consente l'esecuzione. Le versioni di nfsd più recenti di spring 95 rilasciano i file aperti dopo qualche secondo, versioni più vecchie possono impiegare anni...

4. I miei file su NFS sono in sola lettura

   Il default per il server NFS di Linux è di montare i filesystem in sola lettura. Leggete le sezioni Mountd e nsfd ed Esportare filesystem in questo HOWTO e fate riferimento alle pagine man di exports e nfsd. Avrete bisogno di modificare `/etc/exports`.

5. Monto una partizione da un server Linux e, mentre il comando ls funziona, non riesco a leggere o scrivere i file.

   Su versioni più vecchie di Linux occorre lanciare il server NFS con `rsize=1024,wsize=1024`.

6. Monto un server NFS Linux con la dimensione dei blocchi tra 3500-4000 e regolarmente smette di rispondere.

   Allora semplicemente non fatelo. Questo non si verifica con i kernel 2.0 e 2.2. Per quanto mi ricordo non dovrebbero esserci problemi nemmeno con la versione 1.2.

7. Può Linux gestire NFS via TCP?

   Al momento no.

8. Ottengo numerosi strani errori se provo a montare una macchina usando Linux.

   Accertatevi che l'utente sia in 8 gruppi o meno. Server più vecchi lo richiedono.

9. Quando riavvio la mia macchina a volte smette di rispondere provando a smontare un NFS server che non risponde.

   Non smontate server NFS quando riavviate. Ignorateli, non causano problemi se non li si smonta. Il comando è `umount -avt nonfs`.

10. I client NFS Linux sono molto lenti quando scrivono su sistemi Sun o BSD.

    Normalmente NFS scrive i dati in modo sincrono (è possibile disabilitare questa modalità lo si desidera, ma si rischia di perdere dei dati). Funzionano peggio i kernel derivati da BSD, che tendono a non essere in grado di lavorare in piccoli blocchi, quindi quando si scrivono 4K di dati da una macchina Linux in pacchetti da 1K, BSD fa questo:

    ```
    lettura di una pagina da 4K
    modifica di 1K
    scrittura di 4K sul disco fisico
    lettura di una pagina da 4K
    modifica di 1K
    scrittura di 4K sul disco fisico
    ecc..
    ```

11. Quando connetto molti client a un server NFS Linux le prestazioni calano improvvisamente.

    Il protocollo NFS usa pacchetti UDP frammentati. Il kernel ha un limite sulla quantità di frammenti di pacchetti incompleti di cui può disporre prima di eliminare i pacchetti. Nella versione 2.2 il protocollo è adattabile in runtime attraverso il filesystem /proc: `/proc/sys/net/ipv4/ipfrag_high_thresh` e `ipfrag_low_thresh`. Nella versione 2.0 queste sono le costanti di runtime definite in `.../linux/net/ipv4/ip_fragment.c`, `IPFRAG_HIGH_THRESH` e `IPFRAG_LOW_THRESH`. Il significato di questi valori è costituito dal fatto che quando l'occupazione della memoria dei frammenti UDP non assemblati raggiunge il limite "ipfrag_high_thresh" in byte (256K per default in 2.2.3 e 2.0.36) viene ridotta a "ipfrag_low_tresh" improvvisamente. Questa operazione viene effettuata eliminando i frammenti. È simile alla perdita di pacchetti e se il limite più elevato viene raggiunto, le prestazioni del server calano notevolmente.

    256K è sufficiente per un numero pari a 30 client. Se ne avete 60, raddoppiatelo. E raddoppiate anche il limite inferiore.

12. Sto usando Linux 2.2 (o versione successiva) con knfsd e non riesco a fare in modo che la mia macchina AIX, IRIX, Solaris, DEC-Unix, ... lo monti.

Knfsd indica che viene implementata la versione 3 di NFS. Ma questo non avviene. Esiste un'opzione per evitare che annunci questa versione. Usatela. Oppure potete immettere `vers=2` nell'elenco di opzioni di montaggio sui client.

13. La mia macchina AIX 4 non può montare il mio server NFS Linux. Viene visualizzato

```
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.
```

o qualcosa di simile.

AIX 4.2 ha utilizzato porte riservate (<1024) per NFS. AIX 4.2.1 e 4.3 non sono limitati a porte riservate. Inoltre, AIX 4.2.1 e 4.3 cercano di effettuare il montaggio mediante NFS3, quindi NFS/TCP, infine NFS/UDP.

L'aggiunta di

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

alla fine di `rc.tcpip` forzerà l'utilizzo di porte riservate. (Questo suggerimento è stato fornito da Brian Gorka)

9 Esportare filesystem

Il modo di esportare i filesystem attraverso NFS non è completamente coerente tra le varie piattaforme. In questo caso Linux e Solaris 2 sono le eccezioni. Questa sezione propone un breve elenco di modi di effettuare l'esportazione sui vari sistemi. Se il vostro sistema non è riportato dovete controllare sulle pagine del manuale. Provate a cercare parole come: nfsd, system administration tool, rc scripts, boot scripts, boot sequence, /etc/exports, exportfs. Userò un esempio durante questa sezione: come esportare /mn/eris/local ad apollon in lettura e scrittura.

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

Questi sistemi operativi usano il tradizionale formato di Sun per il file export. In `/etc/exports` scrivete:

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

La documentazione completa si trova nella pagina man di `exports`. Dopo avere modificato il file, lanciate `exportfs -av` per esportare i filesystem.

Quanto sono limitate le varie versioni di exportfs relative alle variazioni di sintassi? Su alcuni sistemi troverete che le righe precedenti vengono lette come:

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

oppure possono degenerare in:

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

Consiglio di essere formali, altrimenti si rischia che la versione successiva di `exportfs` sia più sensibile e che quindi non funzioni più nulla.

9.2 Solaris 2

Sun ha completamente reinventato la ruota quando fecero Solaris 2. Quindi la loro sintassi è completamente diversa da quella di tutti gli altri. Ciò che dovete fare è modificare il file `/etc/dfs/dfstab`. Questo file deve contenere i comandi share, come descritto nella pagina man share(1M). Ad esempio:

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

Dopo avere modificato il file, lanciate il programma `shareall` per esportare i filesystem.

10 NFS in Linux 2.2

Mentre scrivo questo HOWTO, Linux 2.2.12 è la versione del kernel corrente e utilizzare NFS può rivelarsi un po' complesso.

Cosa accade se lo stato di NFS in Linux 2.4 è i unknown.

La nuova caratteristica importante di Linux 2.2 è il supporto per un demon del server nfs in-kernel, noto come knfsd in 2.2. Questo modo di implementare nfsd ha alcuni vantaggi, quello principale è la velocità. Una macchina Linux 2.2 con knfsd è un server nfs rispettabile. Si può continuare a utilizzare l'nfsd precedente con Linux 2.2. Questo comporta alcuni vantaggi, principalmente in fatto di semplicità.

Se utilizzate un kernel sorgente o un pacchetto binario creato da RedHat (6.0 o successiva), SuSE (6.1 o successiva) o altri integratori di sistemi professionali, hanno probabilmente integrato tutte le funzionalità di knfsd nel relativo kernel e non è necessario preoccuparsi di nulla, poiché funzionerà certamente nella maggior parte dei casi. Fino al momento in cui non compilate un vostro kernel. Se usate un kernel di stock Linux 2.2 (almeno fino a 2.2.12) knfsd verrà interrotto.

Per utilizzare questo kernel è necessario disporre del pacchetto knfsd di H.J. Lus. Si tratta di una raccolta di patch e delle utility necessarie per la versione 2.2 che Lu sta gestendo a tempo perso. Potete disporne dal mirror del kernel locale, il sito principale è *ftp.kernel.org:/pub/linux/devel/gcc/* . **Non è indicato per uso generale**. Se trovate che questo pacchetto può creare dubbi non fatelo da soli. Attendete un pacchetto del kernel dell'integratore del sistema preferito (ad esempio, Red Hat, SuSE o ...).

Inoltre, non inviatemi domande relative a questo argomento, non sono in grado di aiutarvi. Non dispongo di alcun server knfsd. Se trovate errori o omissioni in questa documentazione, scrivetemi e rivedrò questo HOWTO e ne distribuirò una nuova versione.

State ancora leggendo? Ok. H.J.Lu pubblica argomenti relativi alle nuove versioni di questo pacchetto sulla mailing list del kernel Linux. Altri problemi relativi a NFS nella versione 2.2 vengono pubblicati in questa mailing list. Leggeteli.

È importante notare una cosa in relazione al pacchetto knfsd. Annuncia che supporta la versione 3 di NFS. Tuttavia, non la supporta. Esiste un'opzione che si può utilizzare per evitare la presenza dell'annuncio di NFS3, oppure si può specificare `vers=2` nell'elenco di opzioni di mount sui client.

10.1 Il client

Il client è alquanto semplice. Per bloccarlo è necessario avere `statd` (del pacchetto knfsd) compilato, installato e avviato dagli script di boot. Fatelo. Statd ha bisogno di una directory `/var/lib/nfs` per funzionare, altrimenti verrà terminato senza messaggi di errore. La directory deve essere creata prima di eseguirlo.

Dopo l'esecuzione di statd si può usare il programma `testlk` in `tools/locktest` per verificare se il blocco di un file in un filesystem NFS funziona. Dovrebbe essere così. Se viene visualizzato *No locks available*, statd non funziona.

In realtà, si può anche evitare il blocco completo (non sto consigliando di farlo), immettendo "`nolock`" nell'elenco di opzioni di mount.

Per quanto ne so, questo è ciò che è necessario per far funzionare il client.

Se avete un server Sparc o Alpha NFS scoprirete che il client NFS in Linux 2.2 avrà prestazioni peggiori. Le velocità di trasferimento da e verso il server sono così tragiche che ... non potete nemmeno immaginarle. È molto peggio che in Linux 2.0. Esiste tuttavia una soluzione. La serie di kernel 2.2 di Alan Cox (che è leggermente più sperimentale dei normali kernel 2.2 di Linus) include una patch per fare in modo che Linux 2.2 funzioni quando usato con i server Alpha e Sparc. Se volete usare i kernel 2.2 di Alan Cox dovreste leggere la mailing list relativa ai kernel Linux e potrete scoprire dove si trova la patch. La home page di questa patch è `http://www.uio.no/~trondmy/src/` , nel caso vogliate provare ad applicarla a un kernel 2.2 di produzione. Questa patch non sarà probabilmente disponibile in Linux 2.4, perché richiede troppe modifiche nel kernel per essere accettata nel ciclo di sviluppo corrente. Aspettate Linux 2.5.

Anche `trondmy` dispone di patch per fare in modo che Linux usi la versione 3 di NFS, elementi che consentiranno inoltre di usare il protocollo tcp come meccanismo di trasporto invece di UDP. NFSv3 è ottimo per reti long-haul e altre reti in cui la perdita dei pacchetti non è uguale a zero o la latenza è elevata.

Il motivo per cui dovreste leggere la mailing list relativa ai kernel Linux per usare queste patch consiste nel fatto che talvolta si sono verificati grossi errori. Bug che si nutrono dei vostri file. Quindi fate **attenzione**.

10.2 Il server

Il demon del server nfs in Linux 2.2 e nelle versioni successive è noto come "`knfsd`". È insidioso da installare. Dovete arrangiarvi da soli o fare riferimento ai pacchetti di kernel 2.2 rilasciati da SuSE, Red Hat e altri. Spiacente. Potete ancora usare il vecchio nsfd anche in Linux 2.2. È lento ma semplice da installare.

11 Server NFS su un floppy

Questa sezione è stata scritta da Ron Peters, *rpeters@hevanet.com* Spiega come installare un server NFS al momento dell'avvio da un floppy. Era inizialmente progettata per fare in modo che NFS condividesse un cdrom da un'altra macchina non Linux/UNIX per installare Linux su una macchina che non disponeva di cdrom.

11.1 Introduzione

Questo documento è stato creato per coloro che sperimenteranno lo stesso problema che ho avuto di recente. Stavo creando un server Linux su una macchina che non aveva un cdrom e non aveva, né ha alcuna possibilità di aggiungervene uno a eccezione di un dispositivo esterno SCSI o simile. Ora che sta diventando sempre meno probabile che si debba installare un server su una macchina come questa, questo documento potrebbe rivelarsi poco utile. Tuttavia, lo avrei apprezzato al momento della creazione della mia macchina.

Dato che la mia macchina non aveva un'unità cdrom, pensavo di trovare un server NFS per Win95 e condividere il cdrom per il tempo sufficiente a installare il sistema e utilizzarlo in rete. Dei due prodotti che ho trovato (non dirò i nomi, ma uno era freeware e l'altro aveva una licenza limitata di 15 giorni), uno non funzionava e l'altro non poteva gestire sufficientemente bene la convenzione di denominazione di Linux per poter completare l'installazione.

Ho quindi cercato di avviare la mia macchina Win95 con il set di dischi boot/root e di usare un floppy supplementare per installare il server NFS.

È stato molto semplice e la procedura è probabilmente più semplice della lettura di questa introduzione, ma credo che inserire l'intera procedura in un posto solo avrà più valore.

11.2 Aspettative

Questo documento deriva dall'uso di dischi boot/root di una delle correnti distribuzioni di sviluppo InfoMagic di Slackware. Ho usato la versione 2.0.34 del kernel per i dischi boot/root, ma i programmi del server NFS derivano da un server 2.0.30. Ho già usato il metodo di installazione Slackware, non perché sia più semplice o migliore, ma perché mi trovo più a mio agio e non ho dedicato altro tempo a cercare un altro metodo.

Non credo che ci saranno molti problemi di utilizzo con questo documento in relazione alla versione del sistema operativo. Consiglierei di usare qualcosa di abbastanza recente. Poiché è probabile che verrà utilizzato per l'installazione, sarà probabilmente usato un set di dischi boot/root corrente.

Le dimensioni possono variare.

11.3 Requisiti

- Sistema di rete e disco di avvio. Il sistema che rappresenterà il server NFS deve avere una scheda di rete e deve essere riconosciuto durante il processo di boot. Ulteriori informazioni sono disponibili in Networking HOWTO.
- Floppy secondario che contiene rpc.portmap, rpc.mountd e rpc.nfsd. Questi file possono essere facilmente disponibili da un ftpsearch sul Web.
- Supporto di origine Slackware (o altro) (si presume che sia il cd).

11.4 Installazione del server

11.4.1 Boot del server NFS temporaneo

Avviate il sistema del server NFS dal floppy di boot e assicuratevi che la scheda di rete venga riconosciuta. È anche necessario che il CDROM venga riconosciuto. Userò eth0 come scheda di rete di esempio.

11.4.2 Montaggio del floppy e del cdrom

Dopo avere avviato il sistema, i dischi boot/root non sono più necessari. Il sistema è completamente contenuto nella RAM.

Sostituite il floppy root con il disco supplementare. Montate il floppy:

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

Questo presume che il floppy sia abbia un file system di tipo ext2. Immagino che il disco supplementare possa essere un floppy DOS contenente dei file, ma non l'ho ancora provato. Penso che sarà più semplice di un'immagine di un disco. In questo caso, sarebbe `mount -t msdos ...etc`. Questo deve probabilmente essere inserito nella sezione "Operazioni da effettuare".

Montaggio del cdrom:

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

I dispositivi di floppy e cdrom sono quelli che ho usato. Potrebbero essere diversi in base all'applicazione. I punti di mount /floppy e /cdrom sono presenti nell'immagine del disco floppy perché possano essere usati. In caso contrario, createli o utilizzate i punti di mount che desiderate.

11.4.3 Installazione della rete sul server temporaneo.

In questa sezione il server NSF temporaneo viene impostato per comunicare con la rete. Ci sono pochi comandi da eseguire. Sono necessarie ancora alcune informazioni prima di eseguire i comandi (i valori sono esempi):

IPADDR:172.16.5.100 #Questo è l'indirizzo del server temporaneo.

NETMASK:255.255.255.0 #Questa è la netmask.

BROADCAST:172.16.5.255 #L'ultimo numero (255) è significativo di IPADDR.

ETHNETWORK:172.16.5.0 #Ancora una volta, leggermente diverso da IPADDR.

GATEWAY:172.16.5.251 #Necessario solo se avete un gateway. Probabilmente lo sapete già, ma la maggior parte delle reti provate non ha un gateway.

I comandi per far funzionare la rete. Inserite i valori elencati sopra:

```
ifconfig eth0 inet IPADDR arp netmask NETMASK broadcast BROADCAST
```

```
route add -net ETHNETWORK netmask NETMASK eth0
```

Usate il comando seguente solo se avete un gateway e dovete attraversarlo:

```
route add default gw GATEWAY netmask 0.0.0.0 eth0
```

Se tutto funziona, vi trovate in rete e potraete eseguire un ping degli altri nodi.

11.4.4 Installazione della condivisione NFS.

Determinate la directory che desiderate diventi la condivisione NFS. Nel caso del mio esempio, ho usato la directory /cdrom/slakware. Immettete questa directory nel file /etc/exports:

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

11.5 Esecuzione del server NFS

Andate a /floppy/usr/sbin ed eseguite:

```
./rpc.portmap
```

```
./rpc.mountd
```

```
./rpc.nfsd
```

11.5.1 Completato, iniziate l'installazione.

Questo dovrebbe condividere la directory "/cdrom/slakware" nel file /etc/exports. Al termine, potete avviare la macchina perché sia installata da floppy boot/root (ne ho usati alcuni con cui ho avviato il server NFS) e iniziare l'installazione.

Quando siete pronti a scegliere la collocazione dell'origine dei supporti, scegliete l'opzione relativa al server NFS. Ti verrà richiesto l'indirizzo IP del server. Immettete l'indirizzo usato come IPADDR per il server. Vi verrà anche richiesto il montaggio della directory. Si tratta della directory che avete collocato in /etc/exports sul server NFS.

Il sistema provvederà quindi a montare NFS sul server. Attenzione a eventuali messaggi di errore. Tutto dovrebbe essere completo. Puoi continuare l'installazione.

11.6 Risoluzione dei problemi

11.6.1 Non ancora disponibile.

Non dispongo ancora di informazioni sulla risoluzione dei problemi. Forse mentre utilizzerete questa procedura, ci saranno più suggerimenti disponibili.

11.7 Operazioni da effettuare

11.7.1 Disco DOS.

Create un disco DOS per il floppy supplementare.

11.7.2 Comandi rpc.

Create un ordine specifico dell'esecuzione dei comandi rpc.* e se è necessario eseguire solo alcuni o tutti i comandi.

12 PC-NFS

Non dovreste usare PC-NFS, ma samba.

Samba è migliore di PC-NFS e funziona con Windows 3 for Workgroups e versioni successive di Windows. È più veloce e più sicuro. Usatelo.