Red Hat Linux 6.X as an Internet Gateway for a Home Network

Paul Ramsey <pramsey@refractions.net>

22 Giugno 2000

Un semplice tutorial su come configurare la distribuzione Red Hat 6 e relative varianti per operare come gateway per internet per una piccola rete domestica o in ufficio. Gli argomenti trattati comprendono masquerading, DNS, DHCP e sicurezza di base. Traduzione a cura di Alessio Ciregia, <alciregi(at)tin.it>, e revisione a cura di Daniele Masini, <d.masini(at)tiscali.it>.

Indice

1 Introduzione

Questa pagina contiene un semplice manuale per impostare Red Hat 6.X come gateway per Internet per una rete domestica o una piccola rete in ufficio. Le istruzioni sono molto semplificate: non sarà discusso nessun caso particolare e saranno fatte alcune considerazioni su quali indirizzi di rete saranno utilizzati. Le assunzioni principali sono:

- Si ha una connessione Cable o ADSL full time ad Internet.
- Si ha installato *Red Hat 6.x* su almeno uno dei propri computer. Si noti che queste indicazioni sono valide anche per le distribuzioni derivate da Red Hat, come *Mandrake 6.X* che è distribuita da MacMillan Publishing sotto una vari nomi.
- Il proprio computer Linux ha due schede di rete installare ed entrambe sono compatibili con Linux.
- Si ha un hub ethernet per collegare in rete più di un computer o un cavo incrociato se si vuole collegare in rete un solo computer.
- Si sa come modificare file di testo sulla propria macchina Linux.
- Si è in grado di effettuare il login sulla macchina Linux come `root`. Si è in grado di installare pacchetti RPM dal proprio CD-ROM Linux.

Se qualcuna di queste condizioni non è soddisfatta, allora questo documento probabilmente non è quello fa per il lettore.

Non c'è nulla di particolare che deve essere fatto durante il processo di installazione. Si scelga semplicemente un'installazione che abbia senso e la si porti a termine. Questo documento suggerisce di installare da zero ogni cosa che abbia a che fare con la rete, per evitare di fare supposizioni su cosa è stato installato o configurato durante l'installazione. Per essere sicuri che le cose funzionino e non ci sia confusione circa l'inserimento delle varie informazioni, tutte le configurazioni saranno fatte modificando direttamente i file di configurazione del sistema invece di utilizzare gli strumenti di configurazione grafici forniti da Red Hat. Da un lato, questo potrebbe risultare un po' più difficile, ma in questo modo le indicazioni saranno più facilmente applicabili a distribuzioni e situazioni differenti (per esempio, se X non funziona o si sta configurando un server senza monitor).

1.1 Versioni

L'ultima versione di questo documento dovrebbe sempre essere disponibile su *http://www.coastnet.com/˜pramsey/linux/homenet.html* per la versione HTML e su *http://www.coastnet.com/˜pramsey/linux/homenet.sgml* per la versione SGML.

- 21 Dicembre 1999: Prima versione.
- 2 Gennaio 2000: Aggiunti i suggerimenti di John Mellor sui collegamenti di rete esterni.

- 22 Gennaio 2000: Piccoli aggiornamenti a proposito di schede di rete identiche ed informazioni sull'IP aliasing da parte di Chris Lea.
- 16 Marzo 2000: Alcune informazioni sulla sicureza dei name server e sul supporto a Caldera da parte Nelson Gibbs.
- 22 Giugno 2000: Documentate le stranezze riguardanti la configurazione di Red Hat 6.2. Ulteriori informazioni sul PPPoE da parte Kerr First.

1.2 Copyright

2 Mettere insieme le cose

Nel caso in cui si stia usando o meno un hub, la topologia della propria rete sarà lievemente differente. Qui saranno trattate soltanto le connessioni con cablaggio RJ45 (quella cosa che somiglia al cavo telefonico) senza parlare di cavi coassiali. Col coassiale si possono collegare in rete più macchine senza aver bisogno di un hub, ma si deve avere attenzione a terminare i cavi e ad altre cose. Se si conosce già tutto sui collegamenti di rete, queste istruzioni risulteranno una ripetizione.

2.1 Con un hub

Se si ha un hub, la rete somiglierà a *questa* .

Si colleghi la scheda `eth0` della macchina Linux al modem o al terminale ADSL utilizzando il cavo fornito dal provider al momento dell'installazione (o uno che si sa che funzioni in questa configurazione). Questo è importante perché a volte è necessario collegare i modem con un cavo incrociato ed altre volte bisogna utilizzare un cavo dritto: quello che si deve usare è quello fornito dalla compagnia telefonica.

Si colleghi la scheda `eth1` della macchina Linux all'hub con un cavo dritto. Si colleghino tutti gli altri computer all'hub con cavi dritti.

2.2 Senza un hub

Se non si ha un hub, si può comunque connettere un computer alla macchina Linux, utilizzando un cavo incrociato. La topologia della rete somiglierà a *questa* .

Si colleghi la scheda `eth0` della macchina Linux al modem o al terminale ADSL utilizzando il cavo fornito dal provider. Si colleghi la scheda `eth1` della macchina Linux all'altro computer con un cavo incrociato.

2.3 Con una sola scheda di rete

Questa non è una configurazione consigliata (in questa configurazione la rete interna e quella esterna sono sulla stessa rete fisica e sono perciò teoricamente più suscettibili al cracking; in realtà, il rischio è probabilmente molto basso) ma *può* essere realizzata. Il percorso può variare.

Il kernel Linux include il supporto per l'"IP aliasing", che permette ad una scheda ethernet di funzionare simultaneamente con due indirizzi IP differenti. I tipi di kernel forniti con Red Hat e Mandrake comprendono di default il supporto per l'IP aliasing. Per impostare il gateway con una sola scheda di rete ethernet, in tutti i seguenti codici esemplificativi, si sostituisca semplicemente `eth1` con `eth0:0`.

Nella situazione con una singola scheda, non*è raccomandato utilizzare un server DHCP.*

Si colleghino tutte le macchine ed il modem o il terminale ADSL all'hub. Incrociare le dita e proseguire.

3 Configurare la rete

Bene, a questo punto Linux è già installato sul gateway. È possibile che una delle schede di rete sia già configurata, e che la connessione ad Internet sia già impostata. Comunque sia, ripercorriamo la configurazione dall'inizio, assumendo che non sia stato configurato niente.

Si effettui l'accesso come `root`. Tutte le istruzioni fornite in questo documento presumono che si sia riconosciuti dal sistema come `root`.

Il kernel di Linux si riferisce alle due schede ethernet come `eth0` e `eth1`, per cui è in questo modo che da ora in poi ci si riferirà ad esse. Il problema è riconoscerle. Ecco una maniera "semplice" per capirlo, che sicuramente funziona almeno il 50% delle volte: si metta il computer su un tavolo con la scheda madre orizzonatale ed il pannello posteriore di fronte (come si farebbe se si volesse aprirlo per farci qualche lavoro). La scheda più a sinistra è l'`eth0` – si può etichettarla in qualche modo. Adesso, si annoti su un foglio il costruttore ed il modello sia dell'`eth0` che dell'`eth1`.

Bene, vediamo se l'`eth0` e l'`eth1` sono state riconosciute automaticamente dal kernel. Si digiti `ifconfig eth0` e `ifconfig eth1`. In entrambi i casi, se il kernel ha riconosciuto la scheda, si dovrebbe vedere qualcosa tipo questo (considerando che i numeri e chissà cos'altro potrebbero essere differenti):

```
eth0   Link encap: Ethernet   HWaddr 00:60:67:4A:02:0A
       inet addr:0.0.0.0  Bcast:0.0.0.0  Mask:255.255.255.255
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:466 errors:0 dropped:0 overruns:0 frame:0
       TX packets:448 errors:0 dropped:0 overruns:0 carrier:0
       collisions:85 txqueuelen:100
       Interrupt:10 Base address:0xe400
```

Se il kernel non riconosce la scheda di rete si vedrà qualcosa tipo questo:

```
eth0: error fetching interface information: Device not found.
```

3.1 Configurare un driver di rete

Se entrambe le schede sono state riconosciute, si può saltare alla prossima sezione. Altrimenti, si legga questa sezione.

Quindi una o entrambe le schede non sono state riconosciute dal kernel. Questo non è un problema, davvero. Quello che stiamo per andare a fare è di dire al kernel più esplicitamente come trovare le schede. Ci sono un sacco di metodi e "trucchi", e non saranno trattati tutti. Si ricordi: quando le cose si fanno dure, i duri si rivolgono all' *Ethernet HOWTO* . Ecco dei consigli riassuntivi:

- *Si ha una scheda di rete PCI.* Probabilmente si è in una buona posizione, sempre che non sia talmente recente che non esistano ancora i relativi driver. È possibile raccogliere un grande quantità di informazioni sulle schede di rete (ed altre cose) andando a leggere in `/proc/pci`, annotandosi produttori e modelli.
- *Si ha una scheda di rete ISA.* Probabilmente si dovrà scoprire l'indirizzo di I/O di base e l'IRQ con cui opera la scheda. Si hanno i manuali, vero? Vero? Se non è così, potrebbe essere il momento di visitare il sito web del produttore e vedere se contiene qualche riferimento online. Oppure se si ha un vecchio dischetto DOS per la configurazione della scheda, si avvii il DOS e si guardi se c'è un programma di setup con cui si può leggere ed impostare l'indirizzo di I/O e l'IRQ.
- *Si ha una scheda ISA Plug'n'Play.* Prima si dovrà imparare come configurarla – leggere il *Plug'n'Play HOWTO* . Fortunatamente, una volta che la scheda sarà configurata si conoscerà esattamente sia l'indirizzo di I/O che l'IRQ.

Ora, poiché si conoscono sia la marca che il modello dell'`eth0` e dell'`eth1` si può visitare la *pagina di compatibilità* dell' *Ethernet HOWTO* e cercare la scheda. Si prenda nota del driver consigliato e di qualsiasi informazione a proposito di opzioni particolari che potrebbe richiedere la scheda.

È il momento di modificare un file di configurazione! Il file che modificheremo è `/etc/conf.modules`. Si apra il file nell'editor di testo preferito. Poiché ci sono così tante possibilità e combinazioni di cose che si possono trovare in questo file, sarà presentata come esempio la configurazione del mio gateway. Esso ha una scheda PCI 10/100Mb basata sul chip VIA Rhine, ed un clone di una scheda ISA 10Mb NE2000. Si usa quella a 100Mb per la rete interna e la scheda a 10Mb per la connessione esterna. Il file `/etc/conf.modules` è il seguente:

```
alias parport_lowlevel parport_pc
alias eth0 ne
options ne io=0x300 irq=10
alias eth1 via-rhine
```

Il file `conf.modules` è configurato come segue:

- La prima riga configura la porta parallela per la stampa. Probabilmente se ne avrà una simile anche nella popria configurazione. La si lasci pure così.
- La seconda riga (`alias eth0 ne`) dice al kernel di usare il driver `ne` per il dispositivo `eth0`.
- La terza riga (`options ne io=0x300 irq=10`) dice al driver `ne` a quale indirizzo I/O e a quale interrupt IRQ troverà la scheda ISA. Se si ha una scheda ISA probabilmente si dovrà usare questo tipo di direttiva: si sostituisca semplicemente il driver, l'indirizzo I/O e l'IRQ con le corrette informazioni riguardanti la propria scheda.

- La quarta riga (`alias eth1 via-rhine`) dice al kernel di usare il driver via-rhine per l'`eth1`. Visto che la scheda `eth1` di esempio è una scheda PCI, non si ha bisogno di fornire informazioni riguardanti I/O e IRQ: il sottosistema PCI configura automaticamente il dispositivo.

Ci si assicuri di avere la voce `alias` nel file `conf.modules` per entrambe le proprie schede, e le righe corrette per le opzioni riguardanti tutte le schede ISA. Si potrebbero avere già delle righe in `conf.modules` per ogni scheda ethernet configurata durante l'installazione.

Quando il file `conf.modules` è stato modificato, si provi a digitare `ifconfig eth0` e `ifconfig eth1`. Si dovrebbe ricevere qualche errore se sono modificati gli indirizzi di I/O e gli IRQ senza un manuale del fabbricante.

3.1.1 Due schede di rete identiche

Dunque, si è stati molto molto astuti, comprando due schede di rete identiche per il proprio gateway Linux, ed ora non si riesce a farle funzionare insieme? Non bisogna preoccuparsi, per farle coesistere è solo una questione di usare la sintassi corretta nel file `/etc/conf.modules`. Supponendo che gli indirizzi e i numeri di IRQ siano già stati recuperati e presumendo che le schede in questione siano cloni NE2000 (una scelta comune) identici, il file `/etc/conf.modules` dovrebbe somigliare a questo:

```
alias eth0 ne
alias eth1 ne
options ne io=0x330,0x360 irq=7,9
```

Le opzioni di indirizzamento sono state scritte sulla stessa riga, ed il primo numero per ogni tipo di indirizzamento è relativo all'`eth0`, mentre il secondo numero è per l'`eth1`.

3.2 Configurare la rete interna

La "rete interna" è la rete su cui parlano tutti i computer di casa o dell'ufficio. La "rete esterna" è la spaventosa Internet dall'altro lato della macchina Linux. Nel complesso, la rete interna sarà completamente isolata dalla rete esterna madiante la macchina Linux, la quale opererà come un firewall con livello di sicurezza intermedio.

3.2.1 Il dispositivo di rete

Dal momento che i driver sono funzionanti ed il sistema è in grado di vedere sia l'`eth0` che l'`eth1` con il comando `ifconfig`, è il momento di configurare la rete casalinga interna. Si suppone che la rete interna sia collegata all'`eth1` e quella esterna all'`eth0`.

La rete interna sarà una rete privata e perciò avrà un indirizzo IP speciale riservato per il networking interno: `192.168.1.0`. Questa è una "rete privata di classe C", nel caso si vogliano impressionare i propri amici.

Prima di tutto dobbiamo assicurarci che la rete sia attivata. Si modifichi il file `/etc/sysconfig/network` e ci si assicuri che esistano le seguenti righe:

```
NETWORKING=yes
FORWARD_IPV4=yes
```

La prima riga dice a Linux che vogliamo che i dispositivi di rete vengano attivati al momento dell'avvio del sistema. La seconda riga dice a Linux di abilitare l'IP forwarding. Questo sarà richiesto quando andremo a configurare il masquerading nella Sezione 4.

Nota di Redhat 6.2: Per supportare correttamente l'IP forwarding e il masquerading, Red Hat 6.2 richiede delle modifiche al file `/etc/sysctl.conf`. Assicurarsi che le seguenti righe esistano e che siano impostate con i valori corretti:

```
net.ipv4.ip_forward = 1
net.ipv4.ip_always_defrag = 1
```

In Red Hat e nelle distribuzioni da essa derivate tutte le impostazioni delle interfacce di rete sono contenute in dei file che si trovano nella directory `/etc/sysconfig/network-scripts`. In questa directory, si crei il file `ifcfg-eth1` con al suo interno le seguenti righe:

```
DEVICE=eth1
IPADDR=192.168.1.1
ONBOOT=yes
```

Queste impostazioni dicono agli script di rete di configurare `eth1` al momento del boot e di assegnargli un particolare indirizzo IP. Attivare la propria rete con le nuove impostazioni per mezzo del seguente comando: `/etc/rc.d/init.d/network restart`

3.2.2 Il server DHCP

Un server DHCP configurerà automaticamente i dispositivi della rete casalinga interna con indirizzi IP. Questo è molto utile se si hanno dei computer portatili: si potranno semplicemente collegarli alla rete e saranno immediatamente configurati nel modo corretto. Se non si desidera impostare un server DHCP sulla rete interna, si può saltare questa sezione.

Prima di tutto si deve essere sicuri di avere installato il server DHCP. Per far questo si monti il CD di Linux e si installi l'RPM relativo al `dhcp`. Quindi si modifichi il file `/etc/dhcpd.conf` con le seguenti righe (e solo queste):

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.2 192.168.1.60;
  default-lease-time 86400;
  max-lease-time 86400;
  option routers 192.168.1.1;
  option ip-forwarding off;
  option broadcast-address 192.168.1.255;
  option subnet-mask 255.255.255.0;
}
```

Se si ha intenzione di impostare la macchina Linux per funzionare da caching domain name server, si inserisca la seguente opzione:

```
option domain-name-servers 192.168.1.1;
```

Se si conosce l'indirizzo del server DNS esterno e *non* si ha intenzione di usare la macchina Linux come server DNS, inserire la seguente opzione, dove x.x.x.x e y.y.y.y sono gli indirizzi IP dei server DNS esterni:

```
option domain-name-servers x.x.x.x, y.y.y.y;
```

Se si utilizzerà Samba come servizio di file sharing sulla macchina Linux per i computer Windows, si aggiungano le seguenti opzioni per usare la macchina Linux come server WINS e browsing server di default:

```
option netbios-name-servers 192.168.1.1;
option netbios-dd-server 192.168.1.1;
option netbios-node-type 8;
option netbios-scope "";
```

La configurazione di Samba e del WINS va al di là dello scopo di questo documento. Per informazioni a riguardo, si inizi col vedere il *SMB HOWTO* e si proceda da lì.

Ci sono ancora un po' di passi da fare. Quindi, si modifichi il file `/etc/rc.d/init.d/dhcpd` ricercando la seguente riga:

```
/sbin/route add -host 255.255.255.255 dev eth1
```

I client DHCP su cui gira Windows hanno bisogno di un particolare indirizzo di broadcast nelle risposte DHCP, e questo comando forza lo stack TCP/IP di Linux a produrlo. Se la riga precedente non c'è, *la si aggiunga*. Se invece *si trova* una riga analoga a quella precedente, ci si assicuri che il dispositivo a cui essa si riferisce sia `eth1`.

Il prossimo passo consiste nel modificare il file `/etc/rc.d/init.d/dhcpd` per usare l'`eth1` come il dispositivo di default. Sostituire la riga:

```
daemon /usr/sbin/dhcpd
```

con:

```
daemon /usr/sbin/dhcpd eth1
```

Adesso siamo pronti per avviare il server DHCP con il comando: `/etc/rc.d/init.d/dhcpd start`.

Quindi, dobbiamo accertarci che il server DHCP si avvii all'avvio del sistema. Alcuni pacchetti RPM del server DHCP non includono le direttive per garantire che il server parta ogni volta, così ci assicureremo che parta invocando il comando `chkconfig dhcpd on`.

Questo comando fa sì che RedHat aggiunga lo script di avvio del DHCP nelle directory dei vari runlevel sotto `/etc/rc.d`. Nei runlevel 3 e 5 (multiuser console e multiuser X) il server DHCP viene avviato. Nei runlevel 0, 1 e 6 (shutdown, single user e reboot) il server DHCP viene disattivato.

3.2.3 I computer client

Se si è impostato il DHCP, la configurazione dei computer client è molto semplice: si abiliti semplicemente la configurazione tramite DHCP. Per i computer Windows, questo vuol dire aprire il "Pannello di Controllo", quindi cliccare sull'opzione "Rete", cercare il protocollo "TCP/IP" e continuare con "Configura". Spuntare l'opzione che dice "Configura l'indirizzo TCP/IP automaticamente", applicare le modifiche e riavviare.

Prima di riavviare, si può digitare (sul gateway) il seguente comando: `tail -f /var/log/messages`. Questo visualizzerà il log di sistema di Linux continuamente. Se tutto è andato bene, quando il computer Windows verrà riavviato, si vedrà la sua richiesta di un indirizzo IP e la risposta da parte del server DHCP. Per interrompere il comando `tail -f`, premere Control-C.

Se non si è impostato il DHCP, la configurazione è comunque piuttosto semplice. Si apra l'opzione "Rete" dal "Pannello di Controllo", e si scelga di configurare il protocollo TCP/IP. Si può assegnare ai computer client qualsiasi indirizzo nella rete 192.168.1.0 eccetto 192.168.1.0 (l'indirizzo di rete), 192.168.1.255 (l'indirizzo di broadcast) o 192.168.1.1 (il server Linux). Si ricordi di non assegnare mai lo stesso indirizzo IP a due computer. Si imposti l'indirizzo del "Gateway" a 192.168.1.1, cosicché il traffico in uscita sia indirizzato attraverso il gateway Linux.

L' *IP Masquerading HOWTO* contiene informazioni molto dettagliate sulla configurazione dei client nella *sezione relativa alla configurazione* .

In generale, per configurare un computer client, si abilita la configurazione tramite DHCP, oppure si assegna manualmente un indirizzo della rete 192.168.1.X con il gateway 192.168.1.1. Si faccia in modo che anche il server DNS sia 192.168.1.1 se si è attivato un caching DNS server (v. più avanti) oppure si imposti il DNS con gli indirizzi forniti dal provider.

3.2.4 Il server DNS

Far funzionare la macchina Linux da caching DNS server migliorerà (leggermente) la velocità di navigazione, perché gli indirizzi DNS comunemente usati saranno salvati nella rete interna e non sarà necessario andarli a richiedere su Internet.

Se si è interessati a creare un DNS pienamente funzionante, c'è una grande quantità di cose (complesse) da imparare. È disponibile un *DNS HOWTO* , e il libro *DNS and BIND* è un buon riferimento cartaceo (ed anche molto completo).

Per fare in modo che le macchine client ottengano un vantaggio dal caching server, devono essere configurate per usare il gateway Linux come server DNS primario. Le direttive sul DHCP fornite nella sezione 3.2.2 sono un modo per farlo. Se i client sono configurati manualmente, si può cambiare la configurazione relativa al DNS negli stessi file usati per impostare l'indirizzo IP della macchina.

Per installare il server DNS, si installi l'RPM `bind`, quindi l'RPM `caching-nameserver`. A questo punto, è quasi tutto pronto.

Appena installato, il caching server funzionerà bene, ma se si conoscono gli indirizzi IP dei DNS del provider Internet si possono migliorare leggermente le prestazioni modificando il file `/etc/named.conf` e aggiungendo la riga seguente subito dopo quella contenente la direttiva `directory` (dove x.x.x.x e y.y.y.y sono gli indirizzi IP del server DNS primario e secondario):

```
forwarders { x.x.x.x; y.y.y.y; };
```

Questa modifica fa in modo che il server DNS interno interroghi i server DNS del provider prima di attraversare Internet alla ricerca di un dato indirizzo. I server dei provider hanno spesso una cache di informazioni DNS molto ricca e possono offrire più velocemente una risposta rispetto a quanto possa fare il server DNS interno.

Il demone `named` ha avuto qualche problema di sicurezza negli ultimi 12 mesi, perciò è molto importante utilizzare l'ultima versione, e fare qualche cambiamento alle impostazioni di default per aumentare la sicurezza.

1. Si controlli la versione di `bind` e ci si assicuri che sia almeno la 8.2.2. Controllare l'ultima versione sul sito *Red Hat Updates* oppure *Mandrake Updates* .
2. Si restringa l'accesso al name server interno solo alla rete locale aggiungendo la riga `allow-query { 192.168.1/24; 127.0.0.1/32; };` nel file `/etc/named.conf` subito dopo la direttiva `forwarders`.
3. Si eviti di far girare il name server con i privilegi di `root`. Se il server girasse con tali privilegi, un exploit del server concederebbe all'intruso i privilegi di `root`. Facendo girare il server con i diritti di un utente con delle restrizioni, come `nobody`, si possono diminuire i rischi dovuti ad un eventuale exploit. Per far girare il name server come `nobody`, si modifichi il file `/etc/rc.d/init.d/named` cambiando la riga `daemon named` in `daemon named -u nobody -g nobody`.

Per assicurarsi che il server DNS parta al momento del boot, digitare: `chkconfig named on`. Di nuovo, questo assicura che il servizio venga attivato al boot per i runlevel usuali (3 e 5).

Bene, adesso si può avviare il server DNS: `/etc/rc.d/init.d/named start`

3.2.5 Verificare la rete interna

Finché non configuriamo la rete esterna, il servizio DNS non funzionerà (visto che deve comunicare con altri server DNS su Internet), ma possiamo provare la connettività interna di base con il programma `ping`.

Su uno dei computer client, si apra un terminale (MSDOS), e si digiti `ping 192.168.1.1`. Questo spedirà pacchetti verso la macchina Linux ad intervalli regolari, e la macchina Linux risponderà con altri i pacchetti. Se le cose stanno funzionando nel modo giusto, si dovrebbero vedere i tempi di ritorno dei pacchetti.

3.3 Configurare la rete esterna

Ora siamo pronti per configurare la rete esterna. Talvolta può essere difficoltoso, a seconda di quanto il provider supporta Linux. Se si hanno delle difficoltà, c'è un *ADSL mini-HOWTO* che descrive più dettagliatamente gli aspetti dell'ADSL. Se riesco a trovare un HOWTO sui modem, inserirò il link.

Il problema principale della maggior parte delle connessioni è quello di *ottenere un indirizzo IP*. Alcuni provider Internet assegnano indirizzi IP statici agli abbonati ADSL, e in questo caso la configurazione è semplice. Però, la maggior parte dei provider si sono spostati verso la configurazione dinamica mediante (indovinato) il DHCP. Questo vuol dire che il computer Linux fungerà da *server* DHCP sull'interfaccia `eth1`, e da *client* DHCP sull'interfaccia `eth0`.

Inoltre, molti provider hanno iniziato a fornire i loro servizi in modi specializzati e non standard presumendo che i loro clienti stiano usando Windows. Alcuni di questi casi saranno discussi alla fine della sezione 3.3.2.

3.3.1 Con un IP statico

Se il provider Internet fornisce un indirizzo IP statico, siamo in una botte di ferro. Prima di tutto, si crei un nuovo file di configurazione relativo ad un'interfaccia, `/etc/sysconfig/network-scripts/ifcfg-eth0` contenente le seguenti righe:

```
DEVICE=eth0
IPADDR=x.x.x.x
NETMASK=y.y.y.y
ONBOOT=yes
```

Sostituire x.x.x.x e y.y.y.y con i valori forniti dal provider Internet. Adesso si modifichi il file `/etc/resolv.conf` e si inseriscano le seguenti informazioni:

```
search dominio_del_provider
nameserver n.n.n.n
nameserver m.m.m.m
```

Il `dominio_del_provider` dovrebbe essere fornito dal provider Internet. Quindi si inserisca l'indirizzo IP del DNS primario e quello del DNS secondario al posto di n.n.n.n e m.m.m.m. Se la macchina Linux è stata impostata per funzionare da server DNS, si può aggiungere la riga `nameserver 127.0.0.1` prima delle altre direttive `nameserver`. Ciò farà sì che venga usato il caching server prima di inoltrare richieste all'esterno per richiedere informazioni concernenti il DNS.

3.3.2 Con il DHCP

Se il provider Internet usa la configurazione dinamica mediante DHCP, si deve creare un nuovo file di configurazione per la relativa interfaccia, `/etc/sysconfig/network-scripts/ifcfg-eth0` ed inserirvi le righe seguenti:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

Ci si assicuri che il demone dhcpcd client sia installato sul sistema: inserire il CD di Linux ed installa il pacchetto RPM `dhcpcd`.

È il momento di verificare la configurazione di rete del sistema. Si può usare semplicemente il comando `/etc/rc.d/init.d/network restart`. Quindi si verifichi la connessione esterna eseguendo `ping` verso un computer su Internet, tipo `www.yahoo.com`, e si controlli che qualcosa ritorni indietro.

3.3.3 Stranezze ed anomalie

La situazione reale può differire dalle semplicistiche situazioni qui descritte. Ecco qualche breve nota sulle varie difficoltà riscontrabili ed i link alle risorse più autorevoli per procedere alla loro risoluzione. Si ringrazia John Mellor per aver fornito i link e per avermi spinto ad aggiungere questa sezione.

PPP Over Ethernet (PPPoE) Molti provider ADSL (Bell Atlantic, per esempio) stanno attualmente insistendo che il loro nuovi utenti debbano utilizzare, per connettersi al servizio, il protocollo "PPP over Ethernet" (PPPoE). A tal fine, forniscono un programma client per Windows: non molto utile per gli utenti Linux. Fortunatamente, PPPoE è un protocollo semplice e sono in corso molti sforzi per il supporto sotto Linux.

- Il *Roaring Penguin PPPoE Client* è consigliato da Kerr First.
- *PPPoE on Linux for Bell Sympatico* `*http://www.panix.com/~dfoster/prog/linux/pppoe.html`
- PPPoE on Linux for Sympatico (*General Info*) (*Linux Info*)

Semplici trucchi per il DHCP Una delle abitudini preferite che hanno i provider è quella di legare il servizio ad un solo hostname, o comunque a una sola interfaccia di rete. Probabilmente questo è per prevenire che si possano collegare più computer alla stessa porta ethernet utilizzando un hub (naturalmente, usando Linux e il Masquerading si può ottenere lo stesso effetto con una maggiore sicurezza e la compagnia telefonica non ha modo di saperlo!).

Se il provider ci ha assegnato un hostname ed ha insistito affinché si impostasse la macchina Windows con questo nome, al fine di usare il loro servizio, allora bisogna assicursi che la macchina Linux invii questo hostname quando fa una richiesta al DHCP per ottenere un indirizzo IP.

Il client DHCP di Red Hat viene chiamato quando si imposta il `BOOTPROTO` con `dhcp` nel file di configurazione dell'interfaccia, ma viene chiamato senza riferimenti ad un hostname. Per chiamare il programma con un hostname, in Red Hat 6.1, si modifichi il file `/etc/sysconfig/network`, cambiando la riga:

```
HOSTNAME=
```

In modo che risulti come questa:

```
HOSTNAME=nome_assegnato_dal_provider
```

In alcune varianti di Red Hat potrebbe non funzionare. Se non funziona, controllare lo script `/sbin/ifup` e guardando se le chiamate a `dhcpcd` e `pump` includono il parametro `-h $HOSTNAME`. Se non lo fanno, lo si aggiunga, in modo che esse risultino analoghe a `/sbin/dhcpcd -i $DEVICE -h $HOSTNAME` e `/sbin/pump -i $DEVICE -h $HOSTNAME`.

Road Runner Il servizio Road Runner ha un processo di login particolare che deve essere eseguito prima che il server possa essere utilizzato. Fortunatamente è disponibile un dettagliato http://usmcug.usm.maine.edu/˜kpesce/rr}{Linux Road Runner HOWTO}.

3.3.4 Uno sguardo alle impostazioni di rete

Ora si ammiri il proprio lavoro. Si digiti `ifconfig` per vedere la configurazione di tutti i dispositivi di rete. Sul gateway dovrebbe risultare qualcosa di analogo a:

```
eth0  Link encap:Ethernet  HWaddr 00:60:67:4A:02:0A
      inet addr:24.65.182.43  Bcast:24.65.182.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
      RX packets:487167 errors:0 dropped:0 overruns:0 frame:0
      TX packets:467064 errors:0 dropped:0 overruns:0 carrier:0
      collisions:89 txqueuelen:100
      Interrupt:10 Base address:0xe400
eth1  Link encap:Ethernet  HWaddr 00:80:C8:D3:30:2C
      inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
      RX packets:284112 errors:0 dropped:0 overruns:0 frame:1
      TX packets:311533 errors:0 dropped:0 overruns:0 carrier:0
      collisions:37938 txqueuelen:100
      Interrupt:5 Base address:0xe800
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      RX packets:12598 errors:0 dropped:0 overruns:0 frame:0
      TX packets:12598 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

Si noti che l'interfaccia `eth0` ha un indirizzo IP esterno di fantasia, e l'`eth1` ha un indirizzo interno privato.

Si dia un'occhiata alla tabella di routing digitando il comando `route`. Sul gateway dovrebbe risultare qualcosa di analogo a:

```
Kernel IP routing table
Destination     Gateway      Genmask         Flags Metric Ref Use Iface
255.255.255.255 *            255.255.255.255 UH    0      0     0 eth1
192.168.1.0     *            255.255.255.0   U     0      0     0 eth1
24.65.182.0     *            255.255.255.0   U     0      0     0 eth0
127.0.0.0       *            255.0.0.0       U     0      0     0 lo
default         24.65.182.1  0.0.0.0         UG    0      0     0 eth0
```

Quindi, abbiamo verificato che la rete esterna è configurata, la rete interna è configurata, il dispositivo locale è configurato, l'indirizzo speciale di broadcast 255.255.255.255 è configurato, e la route di default è configurata per puntare al gateway di default del provider. Perfetto!

Ora si hanno la rete interna e quella esterna. Tutto quello che rimane da fare consiste nell'aprire la porta fra le due. Quindi, per prima cosa dobbiamo assicurarci che nessun "mostro" possa entrare dall'esterno.

3.4 Sicurezza

Uno degli inconvenienti nello stare connessi a Internet in modo permanente tramite l'ADSL o via modem è che il computer è esposto a potenziali minacce alla sicurezza per 24 ore al giorno, 7 giorni alla settimana. Utilizzare Linux come gateway riduce i rischi, perché esso nasconde gli altri computer: per quanto riguarda Internet, solo la macchina Linux è disponibile per le connessioni. Questo vuol dire che la rete interna è sicura quanto la macchina Linux, così a questo punto saranno firniti dei consigli basilari per rendere la macchina Linux più sicura.

Per prima cosa, si devono chiudere fuori tutti i tipi cattivi. Per fare questo, si modifichi il file `/etc/hosts.deny` in modo che somigli a questo:

```
#
# hosts.deny  This file describes the names of the hosts which are
#             *not* allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
#            The portmap line is redundant, but it is left to remind you that
#        the new secure portmap uses hosts.deny and hosts.allow. In particular
#             you should know that NFS uses portmap!
ALL: ALL
```

Questo indica ai "TCP wrappers" – che controllano il 95% delle connessioni in entrata – di negare tutte le connessioni provenienti da qualsiasi computer. Questa è una regola piuttosto buona! Ma, ma ti impedirà anche di accedere alla macchina Linux dall'interno della rete casalinga, che è una seccatura, così faremo un'eccezione: si modifichi il file `/etc/hosts.allow` in modo che somigli a questo:

```
#
# hosts.allow  This file describes the names of the hosts which are
#              allowed to use the local INET services, as decided
#              by the '/usr/sbin/tcpd' server.
#
ALL: 127.0.0.1
ALL: 192.168.1.
```

Questo indica ai "TCP wrappers" di permettere connessioni a tutti i servizi dal dispositivo di loopback (127.0.0.1) e dalla rete casalinga (192.168.1.).

Adesso sono stati chiusi fuori i mostri, con un potente lucchetto. Se si vogliono attivare barriere e sistemi di allarme, si dovrà essere un po' più sofisticati. Il *Security HOWTO* è un buon posto per iiziare se si desidera imparare qualcosa di più su come mettere in sicurezza la macchina Linux.

4 Configurare il Masquerading

Tutto a posto! I preliminari sono finiti, adesso siamo dove inizia la magia. L'IP masquerading è uno dei servizi veramente magici che Linux fornisce. Ci sono prodotti commerciali per Windows che fanno la stessa cosa, ma non in modo così efficace: un vecchio 386 può felicemente fornire servizi di IP masquerading per un intero ufficio di medie dimensioni, ma non riesce a far girare Windows 95, anche solo col pacchetto di masquerading (come supplemento, ho letto su alcune recenti riviste che Windows 2000 supporterà la "condivisione delle connessioni" senza nessun software aggiuntivo – questo suona come se le compagnie che vendevano software di condivisione delle connessioni siano state "abbracciate ed estese" dalla Microsoft – comunque, non consiglio di provare Windows 2000 su un 386).

Linux ha funzionalità di firewall estremamente versatili, ed adesso le andremo ad utilizzare nella maniera più semplice e cruda. Se si desidera imparare come implementare un firewall da esperto, si dovrebbe leggere sia il *Firewalling HOWTO* per capire la teoria e l' *IPChains HOWTO* per le istruzioni sul nuovo strumento di firewall `ipchains` che viene fornito col kernel 2.2.X (e per estensione con Red Hat 6.X). È anche disponibile un *IP Masquerading HOWTO* molto buono che contiene molti più dettagli riguardo ai trucchi sul masquerading.

Una volta che la rete interna e quella esterna sono operative, configurare un semplice masquerading è molto facile. Si modifichi il file `/etc/rc.d/rc.local` aggiungendo le seguenti righe:

```
# 1) Svuota le tabelle delle regole.
/sbin/ipchains -F input
/sbin/ipchains -F forward
/sbin/ipchains -F output
# 2) Imposta i tempi del MASQ e consente i pacchetti in entrata per la configurazione DHCP.
/sbin/ipchains -M -S 7200 10 60
/sbin/ipchains -A input -j ACCEPT -i eth0 -s 0/0 68 -d 0/0 67 -p udp
# 3) Nega tutti i pacchetti di forwarding tranne quelli della rete interna.
#    Maschera questi ultimi.
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -s 192.168.1.0/24 -j MASQ
# 4) Carica i moduli di forwarding per i servizi speciali.
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
```

Le ultime due righe caricano i moduli del kernel che permettono all'FTP ed a RealAudio di funzionare sui computer della rete interna. Ci sono altri moduli per i servizi speciali che possono essere aggiunti se se ne ha bisogno:

- CUSeeMe (`/sbin/modprobe ip_masq_cuseeme`)
- Internet Relay Chat (`/sbin/modprobe ip_masq_irc`)
- Quake (`/sbin/modprobe ip_masq_quake`)
- VDOLive (`/sbin/modprobe ip_masq_vdolive`)

Ora siamo pronti per provare il masquerading! Si lanci lo script `rc.local` con il comando `/etc/rc.d/rc.local` e siamo pronti a partire! Ci si sieda davanti ad uno degli altri computer e si provi a navigare nel web. Con un po' di fortuna, ogni cosa funzionerà.

5 Problemi

Ci sono un po' di problemi e un po' di cose che possono andare male usando un documento semplice come questo, perché ci sono diversi casi particolari. La maggior parte di essi riguardano la configurazione degli apparati della rete interna e di quella esterna. Cercherò di rispondere alle persone che hanno dei problemi: segnalatemi cosa è andato male e aggiungerò dei link in questo documento, così le persone che hanno problemi riguardanti casi particolari possono avere una traccia per un aiuto. Contattatemi pure all'indirizzo *pramsey@refractions.net* .

5.1 ICQ non funziona

Alcune parti di ICQ funzionano bene con il masquerading. Altre parti non funzionano affatto. C'è un *modulo ICQ beta* , anche se in fase di sviluppo, che risolve alcuni dei problemi (ma non tutti) nel far girare ICQ con il masquerading. Il file `README` nel codice sorgente della distribuzione descrive come compilare il modulo. Una volta compilato ed installato, si digiti `/sbin/modprobe ip_masq_icq`.

5.2 Se si ha Caldera 2.X e non Red Hat 6.X

Bene, per prima cosa congratulazioni per la controtendenza! Seconda cosa, Nelson Gibbs (ngibbs@pacbell.net) invia buone notizie, perché molte di queste istruzioni saranno valide per Caldera. Ci sono alcuni cambiamenti importanti da tenere in conto, comunque:

1. Una dichiarazione `GATEWAY=xxx.xxx.xxx.xxx` nei file `/etc/sysconfig/network-scripts/ifcfg-eth0` ed `eth1` per l'interfaccia (l'interfaccia locale usa un indirizzo IP remoto e l'interfaccia remota usa l'IP del gateway del provider).
2. Ci si assicuri che lo script `/etc/sysconfig/daemons/dhcpd` segnali `ROUTE_DEVICE` come `eth1` e *non* `eth0`.
3. `/etc/dhcpd.conf` richiede una dichiarazione di sottorete per entrambe le interfacce (non sono completamente sicuro perché ho fatto la seconda dichiarazione: subnet 216.102.154.201 netmask 255.255.255.255 { } senza nessun'altra opzione ed il server DHCP ascolta e spedisce sulla eth0 e la eth1). Il server DHCP dà errore se è specificata una sola sottorete.
4. *Non* aggiungere la route all'host `255.255.255.255`, lo script `/etc/rc.d/init.d/dhcpd` che usa Caldera ha già risolto il problema. *Assicurarsi* di cambiare a `eth1` tutti i riferimenti all'`eth0` contenuti nello script.

5.3 Si vuole che una delle macchine interne funzioni come server Web

Una cosa da nulla! Comunque, *si ha bisogno di un indirizzo IP statico* affinché questa semplice serie di istruzioni funzioni. Nel caso di indirizzo IP dinamico, si avrà bisogno di qualche script in più per assicurarci che l'indirizzo IP sia aggiornato nei comandi di port forwarding quando l'indirizzo cambia.

Si ricordi che fare il forwarding di una porta esterna su una macchina interna rende la macchina "interna" meno "interna" di prima, ma può essere fatto in maniera trasparente e con una degradazione delle prestazioni praticamente nulla. Uno degli effetti collaterali del codice dell'IP masquerading nel kernel di Linux è l'abilità di fare alcuni giochetti divertenti con i pacchetti quando questi raggiungono lo strato network, e `ipmasqadm` ne trae vantaggio.

Per alcune ragioni `ipmasqadm` non è fornito con tutte le varianti di Red Hat e di Mandrake, quindi si dovrà probabilmente scaricarlo dal *sito web* del manutentore – esiste un *RPM* ed anche il codice sorgente.

Una volta ottenuto l'RPM, lo si installi e si aggiungano le seguenti righe al file `/etc/rc.d/rc.local`:

```
/usr/sbin/ipmasqadm portfw -f
/usr/sbin/ipmasqadm portfw -a -P tcp -L x.x.x.x 80 -R 192.168.1.x 80
```

Il primo comando svuota le regole del port forwarding ed il secondo comando aggiunge il forward dalla porta 80 dell'interfaccia esterna alla porta 80 della macchina interna che fa da web server. Si noti che l'indirizzo IP statico esterno va al posto di x.x.x.x e l'indirizzo IP dell'interfaccia della macchina interna va al posto di 192.168.1.x.

Adesso le richieste esterne per la porta 80 saranno redirette in modo trasparente alla porta 80 della macchina interna. Non si può testare questo meccanismo effettuando un telnet o connettendosi alla porta 80 del gateway da una delle macchine interne: il port forwarder serve solo le richieste che arrivano sull'interfaccia *esterna*.