# Firewalling and Proxy Server HOWTO

Mark Grennan, mark@grennan.com v0.80, Feb. 26, 2000

Questo documento si propone di insegnare le basi dei sistemi firewall e di fornire alcuni dettagli sull'impostazione dei firewall per filtering e proxy su un sistema Linux. All'indirizzo *http://www.grennan.com/Firewall-HOWTO.html* è disponibile una versione HTML di questo documento.

## Indice

# 1 Introduzione

David Rudder ha scritto la versione originale di questo Firewall-HOWTO, oramai molte lune fa, e vorrei ancora ringraziarlo per avermi permesso di aggiornare il suo lavoro.

Vorrei inoltre ringraziare Ian Gough per la gentile assistenza prestata a codesto autore dislessico.

I firewall hanno guadagnato una grande popolarità come ultima novità in fatto di sicurezza in Internet. Come accade per tutte le cose che hanno successo, insieme alla fama sono iniziate le incomprensioni. Questo HOWTO spiegherà i concetti base su cosa sia un firewall e come impostarne uno.

Personalmente sto utilizzando il kernel 2.2.13 e la RedHat 6.1 per lo sviluppo di questo howto, e quindi gli esempi qui presenti sono basati su questa distribuzione. Se nella propria distribuzione si trovano delle differenze, invito ad informarmi via email così provvederò ad aggiornare il documento.

## 1.1 Commenti

Qualsiasi commento è più che benvenuto. **SIETE PREGATI DI COMUNICARE OGNI INESATTEZZA CONTENUTA IN QUESTO DOCUMENTO!!!** Sono un essere umano e come tale incline a commettere degli errori. Se ne riscontrate, renderli noti è nel mio interesse. Cercherò di rispondere a tutte le e-mail, ma poiché sono molto occupato, non offendetevi se non potrò farlo.

*Il mio indirizzo email è* **mark@grennan.com** `mailto:mark@grennan.com`

## 1.2 Liberatoria

**NON SONO RESPONSABILE DI EVENTUALI DANNI CAUSATI DA AZIONI INTRAPRESE BASANDOSI SUL CONTENUTO DI QUESTO DOCUMENTO**. Deve essere visto come un'introduzione alle modalità di funzionamento dei firewall e dei proxy server. Non sono, e non pretendo di esserlo, un esperto di sicurezza. ;-) Sono solo una persona che ha letto troppo e che ama i computer più di molta altra gente. Vi prego di capire che sto scrivendo questo documento per aiutare la gente ad informarsi sull'argomento, ma non sono pronto a scommettere la mia vita sull'accuratezza del suo contenuto.

## 1.3 Copyright (in inglese)



## 1.4 Perché ho scritto questo documento

Diversi anni fa, mentre lavoravo per lo Stato dell'Oklahoma come loro "Amministratore Internet" mi è stato chiesto di "mettere lo Stato su Internet", in pratica senza fondi (nota: al tempo non c'era un tale titolo, ero solamente un tipo che faceva tutto il lavoro). Il miglior modo affinché questa cosa si potesse realizzare era di usare quanto più software libero e hardware di recupero fosse possibile. Linux e una manciata di vecchi 486 era tutto quello di cui disponevo.

I firewall commerciali sono MOLTO sovrapprezzo e la documentazione sul loro funzionamento è considerata quasi top secret. Ho scoperto che crearmi da solo un firewall era praticamente impossibile.

Al mio lavoro successivo mi è stato chiesto di inserire un firewall. Linux aveva appena aggiunto il codice per l'implementazione, così ancora senza fondi, ho cominciato a realizzarne uno con Linux. Sei mesi dopo il mio firewall era a posto e questo documento è stato aggiornato.

## 1.5 Letture aggiuntive

- *The Linux Networking Overview HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/Networking-Overview-HOWTO.html`
- *The Ethernet HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html`
- *IPchains Firewalling made Easy!* `http://ipchains.nerdherd.org/`
- *Linux Network Address Translation* `http://www.linas.org/linux/load.html`
- *The Net-3 HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/NET-3-HOWTO.html`
- *The NET-PPP HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/PPP-HOWTO.html`
- *The easiest way to create Virtual Tunnels over TCP/IP networks* `http://vtun.netpedia.net/`

[ Altre URL sono da collocare qui ]

# 2 Capire i Firewall

Un firewall (parete tagliafuoco) è una struttura intesa ad impedire la diffusione del fuoco. Negli edifici i firewall sono i muri in mattone che dividono completamente le sezioni. In un auto un firewall è la parete metallica che separa l'abitacolo dal motore.

I firewall di Internet sono intesi per tenere le fiamme dell'inferno di Internet fuori dalla propria LAN privata, oppure per conservare i membri della propria LAN puri e casti negandogli l'accesso a tutte le tentazioni della diabolica Internet. ;-)

Il primo computer firewall era un host Unix non instradante con connessioni a due reti diverse. Una scheda di rete era connessa ad Internet e l'altra ad una LAN privata. Per raggiungere Internet dalla rete privata, si doveva effettuare il login sul server firewall (Unix). Poi si utilizzavano le risorse del sistema per accedere ad Internet. Per esempio, si poteva usare X-Windows per lanciare il browser Netscape nel sistema firewall e poi esportare il display sulla propria workstation. Il browser in funzione sul firewall aveva quindi accesso ad entrambe le reti.

Questo tipo di sistema dual homed (un sistema con due connessioni di rete) è una gran cosa se ci si può FIDARE DI TUTTI i propri utenti. Si potrebbe quindi semplicemente configurare un sistema Linux e dare un account a chiunque abbia bisogno di accedere ad Internet. Con questa impostazione, il solo computer nella propria rete privata che sa qualcosa del mondo esterno è il firewall. Nessuno può scaricare qualcosa sulla workstation personale, si deve prima scaricare un file sul firewall e poi scaricarlo dal firewall nella propria workstation.

NOTA IMPORTANTE: il 99% delle irruzioni iniziano guadagnando l'accesso a livello utente sul sistema da attaccare. Per questo motivo non raccomando questo tipo di firewall. Inoltre è molto limitante.

## 2.1 Politiche di Firewall

Non si creda che il firewall sia tutto quello di cui si ha bisogno. *Per prima cosa si decidano delle politiche.*

I firewall sono usati con due scopi:

1. tenere fuori la gente (worm / cracker).
2. tenere dentro la gente (dipendenti / bambini).

Quando ho iniziato a lavorare sui firewall mi sorpresi di scoprire che la compagnia per la quale lavoravo era più interessata a "spiare" i propri dipendenti che a tenere fuori dalla propria rete i cracker.

Almeno nel mio stato (Oklahoma) i datori di lavoro hanno il diritto di cominciare a controllare le chiamate telefoniche e l'attività Internet non appena informati i propri dipendenti.

Il Grande Fratello non è il governo. Grande Fratello = Grande Business.

Non mi si fraintenda. La gente dovrebbe lavorare, non giocare mentre è al lavoro. E sento che l'etica nel lavoro si sta sgretolando. Comunque, ho osservato anche che i tipi del management sono i primi ad abusare delle regole da loro imposte. Ho visto continui rimproveri diretti a lavoratori perché cercavano, per andare al lavoro, gli itinerari dei bus su Internet, quando poi gli stessi manager spendevano ore cercando ristoranti raffinati o nightclub per ricevere potenziali clienti.

La mia soluzione per questo tipo di abusi consiste nel pubblicare su una pagina Web, a disposizione di chiunque, i log del firewall.

L'affare sicurezza può essere spaventoso. Se amministri un firewall, guardati le spalle.

### 2.1.1 Come determinare una politica per la sicurezza

Ho consultato diversa documentazione di rilievo su come creare una politica per la sicurezza. Dopo anni di esperienza posso adesso affermare: non si creda ad una parola. Creare una politica è semplice.

1. determinare quali servizi si ha bisogno
2. determinare il gruppo di persone che si vuole servire
3. determinare a quali servizi ogni gruppo ha necessità di accedere
4. per ciascun gruppo descrivere come si potrebbe rendere sicuro il servizio
5. scrivere un'espressione che renda tutte le altre forme di accesso una violazione

La politica diventerà sempre più complicata nel tempo, però ora non si provi a coprire troppi aspetti. La si realizzi semplice e chiara.

## 2.2 Tipi di Firewall

Esistono due tipologie di firewall.

1. Firewall Filtranti - che bloccano i pacchetti di rete selezionati.
2. Proxy Server (talvolta detti firewall) - che fanno le connessioni di rete per voi.

### 2.2.1 Firewall Filtranti (Packet Filtering Firewall)

Il Packet Filtering è il tipo di firewall presente nel kernel Linux.

Un firewall filtrante funziona a livello di rete. È permesso ai dati di lasciare il sistema solo se lo permettono le regole del firewall. Come i pacchetti arrivano sono poi filtrati in base alle informazioni sul tipo, sull'indirizzo di provenienza, sull'indirizzo di destinazione e sulle porte contenute in ciascuno di essi.

Molti router di rete hanno l'abilità di effettuare alcuni servizi di firewall. Un firewall filtrante può essere pensato come un tipo particolare di router. Per questo motivo è necessaria una profonda conoscenza della struttura dei pacchetti IP per lavorarci.

Poiché sono analizzati e registrati pochissimi dati, i firewall filtranti occupano meno la CPU, e creano minor latenza nella propria rete.

I firewall filtranti non forniscono un controllo a livello di password. Gli utenti non possono identificarsi. La sola identità che un utente ha consiste nel numero IP assegnato alla sua macchina. Ciò può essere un problema se si intende usare DHCP (assegnazione dinamica dell'IP). Poiché le regole sono basate sui numeri IP, dovranno essere aggiustate non appena saranno assegnati nuovi numeri. Non saprei come automatizzare questo processo.

I firewall filtranti sono più trasparenti per gli utenti. L'utente non deve impostare regole nella sua applicazione per utilizzare Internet. Con la maggioranza dei proxy server questo non è vero.

### 2.2.2 Proxy Server

I Proxy sono principalmente usati per controllare, o monitorare, il traffico. Alcuni proxy di applicazioni possono fare la cache dei dati richiesti, ciò abbassa le richieste di banda e diminuisce il tempo d'accesso per il successivo utente che vuole accedere a quegli stessi dati. Inoltre fornisce un'evidenza inequivocabile su quanto è stato trasferito.

Esistono due tipi di proxy server.

1. Application Proxy (Proxy di Applicazione) - che fanno il lavoro al nostro posto.
2. Proxy SOCKS - che incrociano le comunicazioni.

#### 2.2.3 Application Proxy

Il miglior esempio è quello di una persona che effettua un telnet su un altro computer e poi da qui al resto del mondo. Solo con un proxy server di applicazione il processo è automatizzato. Non appena si fa telnet verso l'esterno il client per prima cosa vi manda al proxy. Il proxy poi si connette al server che si è richiesto (il mondo esterno) e restituisce i dati.

Poiché i proxy server gestiscono tutte le comunicazioni, possono registrare qualsiasi cosa vogliono (si vuole). Per i proxy HTTP (web) ciò può includere qualsiasi URL che si visita, per i proxy FTP qualsiasi file si scarica. Possono anche filtrare parole "inappropriate" dai siti che si visitano o controllare la presenza di virus.

Gli application proxy server possono autenticare gli utenti. Prima di effettuare una connessione verso l'esterno, il server può richiedere all'utente, per prima cosa, di effettuare il login. Ad un utilizzatore del web ciò comporterà la necessità di un login per ogni sito che desidera visitare.

#### 2.2.4 Proxy SOCKS

Un server SOCKS è molto simile ad una vecchia switch board. Semplicemente incrocia, attraverso il sistema, i cavi della propria connessione con un'altra connessione esterna.

La maggior parte dei server SOCKS funziona solamente con connessioni di tipo TCP e come i firewall filtranti non fornisco l'autenticazione degli utenti. Possono comunque registrare dove si è connesso ogni utente.

## 3 Architettura dei Firewall

Si può strutturare la propria rete in un sacco di modi per proteggere il proprio sistema usando un firewall.

Se si ha una connessione dedicata ad Internet attraverso un router, si potrebbe collocare il router direttamente nel proprio sistema firewall. Oppure si potrebbe passare attraverso un hub (concentratore) per fornire server a pieno accesso fuori dal proprio firewall.

### 3.1 Architettura Dial-up

Probabilmente si sta facendo uso di un servizio dialup, quale una linea ISDN. In questo caso si potrebbe utilizzare una terza scheda di rete per fornire un DMZ filtrato. Questo dà il pieno controllo sui propri servizi Internet e li separa dalla propria rete normale.

```
                 __________
  _/\__/\_      |          |          _______________
 |        |     | Sistema  |  (LAN)  |               |
/ Internet \----| Firewall |--(HUB)--| Workstation/s |
\_  _  _  _/    |__________|         |_______________|
```

```
  \/ \/ \/           |
                   (DMZ)
                   (HUB)
```

## 3.2 Architettura a Router singolo

Caso in cui è presente, tra voi ed Internet, un router o un modem. Se si possiede il router si potrebbero impostare alcune rigide regole di filtraggio, altrimenti se il router appartiene al proprio ISP, si potrebbe richiederne l'inserimento.

```
                 --------           ----------
  _/\__/\_      | Router |         |          |         _______________
 |        |     | oppure |  (DMZ)  | Sistema  |  (LAN)  |               |
/ Internet \----| Modem  |--(HUB)--| Firewall |--(HUB)--| Workstation/s |
\_  _  _  _/    |________|    |    |__________|         |_______________|
  \/ \/ \/                    |
                           (Server)
                           (esterno)
```

## 3.3 Server Proxy e Firewall

Se si vuole monitorare dove vanno gli utenti della propria rete e se la rete è piccola, si può integrare un server proxy nel firewall. Gli ISP qualche volta lo fanno per stilare una lista degli interessi dei propri clienti da poter rivendere ad agenzie di marketing.

```
                 ----------
  _/\__/\_      | Proxy /  |          _______________
 |        |     | Sistema  |  (LAN)  |               |
/ Internet \----| Firewall |--(HUB)--| Workstation/s |
\_  _  _  _/    |__________|         |_______________|
  \/ \/ \/
```

Si può collocare il server proxy nella propria LAN come si vuole. In questo caso il firewall dovrebbe però contenere delle regole che consentano la connessione ad Internet, e per i servizi che intende fornire, solo al server proxy. In questo modo gli utenti potranno accedere ad Internet solamente attraverso il proxy.

```
                 ----------
  _/\__/\_      |          |          _______________
 |        |     | Sistema  |  (LAN)  |               |
/ Internet \----| Firewall |--(HUB)--| Workstation/s |
\_  _  _  _/    |__________|    |    |_______________|
  \/ \/ \/                      |     ______________
                                |    |              |
                                +----| Server Proxy |
                                     |______________|
```

### 3.4 Configurazioni Internet ridondanti

Se si sta per avviare un servizio simile a YAHOO, o forse Slashdot, si potrebbe aver bisogno di organizzare il sistema utilizzando router ridondanti e firewall (consultare l'High Availability HowTo). Utilizzando tecniche DNS round-robin oppure load-balancing di server, si può creare un servizio 100% uptime.

```
  _/\__/\_                                     _/\__/\_
 |        |                                   |        |
/  ISP #1  \______                 (WAN)_____/ Partner  \
\_  _  _  _/      |                (HUB)     \_  _  _  _/
  \/ \/ \/        |               ___|____     \/ \/ \/
                __|___           |_______ |
  _/\__/\_     |_____ |         |Sistema ||          ______
 |        |   |      ||  (DMZ)  |Firewall||  (LAN)  |      |
/  ISP #2  \--|Router||--(HUB)--|  (VPN) ||--(HUB)--| WS/s |
\_  _  _  _/  |______|     |    |________|     |    |______|
  \/ \/ \/                 |         |         |     ______
         |            (Server)   (Server)      |    |      |
 ------  |            (esterno) (condivisi)    +----|Proxy |
| WS/s | |                                          |______|
| VPN  |-+
|______|
```

E' facile che la propria rete possa sfuggire di mano. Si controlli ogni connessione. E' sufficiente un utente con un modem per compromettere la LAN.

# 4 Impostazione del firewall Linux filtrante

### 4.1 Requisiti Hardware

I firewall filtranti non richiedono hardware particolare, sono solo un po' più complessi di un semplice router.

Tutto quello di cui si ha bisogno è:

1. un 486-DX66 con 32 mega di memoria
2. un disco fisso da 250 mega (meglio 500)
3. connessioni di rete (Schede LAN, Porte Seriali, Wireless?)
4. monitor e tastiera

Con alcuni sistemi utilizzando la console su porta seriale, si possono pure eliminare il monitor e la tastiera. Se si ha bisogno di un server proxy che gestisca parecchio traffico, si dovrebbe acquistare il sistema più potente possibile. Questo perché per ogni utente che si connette al sistema, sarà creato un nuovo processo. Se si hanno 50 o più utenti suppongo che si avrà bisogno di:

1. un Pentium II con 64 mega di memoria
2. un disco fisso da 2 giga per salvare tutti i log
3. due connessioni di rete

4. monitor e tastiera

Le connessioni di rete possono essere di qualsiasi tipo (schede NIC, ISDN e anche modem).

# 5 Requisiti Software

## 5.1 Selezionare un Kernel

Per creare un firewall filtrante non è necessario alcun software particolare. Basta Linux. Al momento sto utilizzando la RedHat 6.1.

La base del firewall in Linux è stata modificata diverse volte. Se si sta utilizzando un kernel vecchio (1.0.x o precedente) ci si procuri una nuova copia. Queste versioni usano ipfwadm (**http://www.xos.nl/linux/ipfwadm/**) e non sono più supportate.

Se si sta utilizzando la 2.2.13 o più recente si starà facendo uso di ipchains, sviluppato da **http://www.rustcorp.com/linux/ipchains/**.

Se invece si utilizza il nuovo kernel 2.4 esiste una nuova utility per il firewall con diverse nuove caratteristiche. Scriverò presto qualcosa a riguardo.

## 5.2 Selezionare un server proxy

Se si vuole impostare un server proxy si avrà bisogno di uno dei seguenti pacchetti:

1. Squid
2. The TIS Firewall Toolkit (FWTK)
3. SOCKS

Squid è un pacchetto eccellente ed è in grado di sfruttare il proxy trasparente di Linux. Descriverò di seguito come impostare questo server.

Al momento della stesura di questo documento, *Network Associates* `http://www.networkassociates.com/` e Trusted Information System (TIS) si sono uniti, quindi si visitino i loro siti per ottenere informazioni sui cambiamenti. Nel frattempo, il toolkit dovrebbe essere ancora disponibile, ***http://www.tis.com/research/software/*** `http://www.tis.com/research/software/` .

Trusted Information System ha reso disponibile una collezione di programmi realizzati per facilitare la gestione del firewall. Con questi strumenti è possibile impostare un daemon per ogni servizio (WWW, telnet, ecc.) che si intende utilizzare.

# 6 Preparare il sistema Linux

Si installi il sistema Linux nel modo più compatto possibile. La mia installazione presentava una configurazione di tipo server quindi ho provveduto a disabilitare i servizi non necessari nel file /etc/inetd.conf. Per una maggior sicurezza si dovrebbero disinstallare i servizi inutili.

Siccome la maggior parte delle distribuzioni non fornisce in genere un kernel efficiente per le proprie esigenze, sarà necessario compilarne uno proprio. Questa operazione sarebbe meglio effettuarla su un computer diverso dal firewall. Se si installa sul firewall un compilatore C e le utilità, ci si ricordi di rimuoverle dopo aver completato la configurazione del kernel.

## 6.1 Compilazione del Kernel

Si parta con un'installazione minima della propria distribuzione Linux. Meno software si installa e meno buchi, backdoor e/o bug ci saranno ad introdurre problemi di sicurezza nel proprio server.

Si prenda un kernel stabile. Io sto utilizzando, per il mio sistema, la 2.2.13. Quindi questa documentazione è basata sulle sue impostazioni.

È necessario ricompilare il kernel di Linux con le opzioni appropriate. Se non si è mai ricompilato il proprio kernel prima d'ora, si dovrebbe leggere il Kernel HOWTO, l'Ethernet HOWTO e il NET-2 HOWTO.

Qui ci sono le impostazioni di rete che so per certo come funzionanti. Ne ho marcate alcune con un ?. Se si è intenzionati ad utilizzarle, le si abilitino pure. Per modificare le impostazioni del mio kernel utilizzo make menuconfig.

```
<*> Packet socket
[ ] Kernel/User netlink socket
[*] Network firewalls
[ ] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
[ ] IP: multicasting
[*] IP: advanced router
[ ] IP: kernel level autoconfiguration
[*] IP: firewalling
[?] IP: always defragment (required for masquerading)
[?] IP: transparent proxy support
[?] IP: masquerading
--- Protocol-specific masquerading support will be built as modules.
[?] IP: ICMP masquerading
--- Protocol-specific masquerading support will be built as modules.
[ ] IP: masquerading special modules support
[*] IP: optimize as router not host
< > IP: tunneling
< > IP: GRE tunnels over IP
[?] IP: aliasing support
[*] IP: TCP syncookie support (not enabled per default)
--- (it is safe to leave these untouched)
< > IP: Reverse ARP
[*] IP: Allow large windows (not recommended if <16Mb of memory)
< > The IPv6 protocol (EXPERIMENTAL)
---
< > The IPX protocol
< > Appletalk DDP
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] Bridging (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
< > WAN router
[ ] Fast switching (read help!)
[ ] Forwarding between high speed interfaces
[ ] PU is too slow to handle full bandwidth
```

```
QoS and/or fair queueing  --->
```

Dopo aver completato tutte le impostazioni sarà necessario ricompilare, reinstallare il kernel e riavviare. Io utilizzo il comando:

make dep;make clean;make bzlilo;make modules;make modules_install;init 6 per fare tutto in un colpo solo.

## 6.2 Configurazione di due schede di rete

Se il proprio computer possiede due schede di rete, molto probabilmente sarà necessario inserire un'istruzione append nel proprio file /etc/lilo.conf, per specificare l'IRQ e l'indirizzo di entrambe le schede. L'istruzione presente nel mio lilo è la seguente:

```
append="ether=12,0x300,eth0 ether=15,0x340,eth1"
```

## 6.3 Configurazione degli indirizzi di rete

Siamo arrivati finalmente alla parte più divertente dell'impostazione. Non sto per addentrarmi ad approfondire come impostare una LAN, per risolvere eventuali problemi si legga il Networking-HOWTO.

L'obiettivo è di fornire il sistema firewall di due connessioni di rete, una diretta verso Internet (versante non sicuro) e una verso la LAN (versante sicuro).

Ad ogni modo, ci sono un paio di decisioni da prendere.

1. Si vogliono utilizzare per la propria rete locale indirizzi IP reali o qualcosa di differente ?
2. L'indirizzo IP è statico oppure è assegnato dall'ISP ?

Dal momento che non si desidera che Internet abbia accesso ad alcuna parte della rete privata, non è necessario utilizzare degli "indirizzi reali". Si potrebbero prendere degli indirizzi qualsiasi per la propria rete LAN, ma ciò non è raccomandato. Se i dati dovessero essere instradati fuori dalla LAN, potrebbero giungere ad una porta di un altro sistema.

Esistono comunque molti intervalli di indirizzi internet riservati appositamente per le reti private.

Tra questi c'è 192.168.1.xxx che utilizzeremo nei nostri esempi.

Sarà inoltre necessario utilizzare l'IP masquerading. Con questo procedimento il firewall farà proseguire i pacchetti e, per farli viaggiare in Internet, li modificherà utilizzando un indirizzo IP "REALE".

Utilizzando questi indirizzi IP non-instradabili la propria rete sarà più sicura. I router internet non invieranno pacchetti con questi indirizzi.

A questo punto potrebbe essere utile leggere l'IP Masquerading HOWTO.

```
        24.94.1.123  __________    192.168.1.1
   _/\__/\_        \ |          | /           _______________
  |        |        \| Sistema  |/           |               |
 / Internet \--------| Firewall |------------| Workstation/s |
 \_  _  _  _/        |__________|            |_______________|
   \/ \/ \/
```

E' necessario possedere un indirizzo IP "reale" da assegnare alla scheda di rete corrispondente ad Internet. Questo indirizzo può essere assegnato in modo permanente (indirizzo IP statico) oppure al momento della connessione alla rete attraverso il processo PPP.

Si dovranno quindi assegnare gli indirizzi IP alla rete locale, ad esempio, 192.168.1.1 alla scheda di rete corrispondente alla LAN. Questo sarà il gateway di default. A tutte le altre macchine nella rete protetta (LAN) si potrà assegnare un numero appartenente all'intervallo 192.168.1.xxx (da 192.168.1.2 a 192.168.1.254).

Io utilizzo la RedHat Linux. Per configurare la rete al momento del boot ho aggiunto un file ifcfg-eth1 nella directory /etc/sysconfig/network-scripts. Sempre in questa directory si potrebbero trovare anche ifcfg-ppp0 o ifcfg-tr0. Questi file 'ifcfg-' sono utilizzati dalla RedHat per configurare e abilitare i dispositivi di rete al boot. Il nome è assegnato in base al tipo di connessione.

Segue, come esempio, ifcfg-eth1 (seconda scheda ethernet):

```
DEVICE=eth1
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
GATEWAY=24.94.1.123
ONBOOT=yes
```

Se si sta per utilizzare una connessione dialup si dia un'occhiata ai file ifcfg-ppp0 e chat-ppp0. Questi file controllano la connessione PPP.

Questo file ifcfg potrebbe essere simile a:

```
DEVICE="ppp0"
ONBOOT="yes"
USERCTL="no"
MODEMPORT="/dev/modem"
LINESPEED="115200"
PERSIST="yes"
DEFABORT="yes"
DEBUG="yes"
INITSTRING="ATZ"
DEFROUTE="yes"
HARDFLOWCTL="yes"
ESCAPECHARS="no"
PPPOPTIONS=""
PAPNAME="LoginID"
REMIP=""
NETMASK=""
IPADDR=""
MRU=""
MTU=""
DISCONNECTTIMEOUT=""
RETRYTIMEOUT="5"
BOOTPROTO="none"
```

## 6.4 Verifica del funzionamento della rete

Si cominci impartendo i comandi ifconfig e route. Se si possiedono due schede di rete l'output di ifconfig dovrebbe essere simile al seguente:

```
#ifconfig
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:1620 errors:0 dropped:0 overruns:0
          TX packets:1620 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0

eth0      Link encap:10Mbps Ethernet  HWaddr 00:00:09:85:AC:55
          inet addr:24.94.1.123 Bcast:24.94.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1000 errors:0 dropped:0 overruns:0
          TX packets:1100 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0
          Interrupt:12 Base address:0x310

eth1      Link encap:10Mbps Ethernet  HWaddr 00:00:09:80:1E:D7
          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:1110 errors:0 dropped:0 overruns:0
          TX packets:1111 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0
          Interrupt:15 Base address:0x350
```

mentre la propria tabella di instradamento dovrebbe essere:

```
#route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
24.94.1.0       *               255.255.255.0   U     1500   0       15 eth0
192.168.1.0     *               255.255.255.0   U     1500   0        0 eth1
127.0.0.0       *               255.0.0.0       U     3584   0        2 lo
default         24.94.1.123     *               UG    1500   0       72 eth0
```

**Nota:** 24.94.1.0 è il lato Internet di questo firewall e 192.168.1.0 è il lato privato.

Ora è necessario assicurarsi che ogni computer della propria LAN sia in grado di effettuare dei ping verso l'indirizzo interno del sistema firewall (192.168.1.1 in questo caso). Se non è possibile, si passi nuovamente al NET-2 HOWTO e si lavori ancora un po' sulla rete.

Dopodiché, dal firewall, si provi ad effettuare un ping ad un sistema presente in Internet. Per i miei test utilizzo www.internic.net. Se non funziona, provare con un server del proprio ISP. Se ancora non funziona allora parte della propria connessione ad Internet è errata. Si dovrebbe essere in grado di connettersi ovunque in Internet dal proprio firewall. Si dia un'occhiata all'impostazione del proprio gateway di default. Se si sta utilizzando una connessione dialup si faccia un doppio controllo della user ID e della password. Quindi si rilegga il net-2 HOWTO, e si riprovi.

Ora si provi ad effettuare un ping verso l'indirizzo esterno del firewall (24.94.1.123) utilizzando un computer della propria LAN. Non dovrebbe funzionare. Se funziona allora il masquerading o l'IP Forwarding sono abilitati, oppure si ha già qualche filtro di pacchetti impostato. Li si disabilitino e si riprovi. E' necessario assicurarsi che il filtraggio sia a posto.

Per i kernel più recenti della 2.1.102 si può impartire il comando:

```
echo "0" > /proc/sys/net/ipv4/ip_forward
```

Se si sta utilizzando un vecchio kernel (PERCHÉ ?) sarà necessario ricompilare il kernel con il forwarding disabilitato (meglio passare ad un kernel più recente).

Si provi ad effettuare di nuovo un ping verso l'indirizzo esterno del firewall (24.94.1.123), non dovrebbe funzionare.

Ora si abiliti l'IP forwarding e/o il masquerading, si dovrebbe essere in grado, da qualsiasi sistema della LAN, di effettuare dei ping ovunque in Internet:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

**NOTA IMPORTANTE:** Se si utilizzano indirizzi IP "REALI" per la propria LAN (non 192.168.1.*) e non si riescono ad effettuare i ping verso internet, ma solo verso il lato internet del firewall, allora ci si assicuri che il proprio ISP instradi i pacchetti per gli indirizzi della propria rete privata.

Un test di questo problema può essere effettuato se si ha qualcuno in Internet (diciamo un amico che utilizza un provider locale) attraverso l'utilizzo del traceroute verso la propria rete. Se il tracciamento si blocca con il router del proprio provider allora significa che il router non inoltra il proprio traffico.

Funziona ? Splendido. La fase più ardua è compiuta. :-)

## 6.5 Sicurezza del Firewall

Un firewall non è di nessuna utilità se lasciato aperto agli attacchi. Un "malintenzionato" potrebbe ottenere l'accesso ad un servizio non del firewall e modificarlo secondo le proprie esigenze.

Date un'occhiata al file /etc/inetd.conf, utilizzato per configurare inetd, noto anche come "super server". Esso controlla parecchi demoni che attiva nel momento in cui sono richiesti da pacchetti che arrivano diretti verso porte "note" ("well known port").

Si possono disabilitare echo, discard, daytime, chargen, ftp, gopher, shell, login, exec, talk, ntalk, pop-2, pop-3, netstat, systat, tftp, bootp, finger, cfinger, time, swat e linuxconfig se presente.

Per disabilitare un servizio, si aggiunga un # come primo carattere della linea corrispondente al servizio. Una volta concluso, si invii un SIG-HUP al processo impartendo "**kill -HUP <pid>**", dove <pid> è il numero del processo di inetd. Questo comando obbliga inetd a rileggere il suo file di configurazione (inetd.conf) e a riavviarsi senza dover spegnere il sistema.

Si faccia un test effettuando un telnet alla porta 15 (netstat) del firewall, se si ottiene dell'output allora non si è disabilitato questo servizio.

telnet localhost 19

Si può inoltre creare il file /etc/nologin. Si inserisca qualche linea di testo tipo (FILA VIA!). Quando questo file esiste, login non consentirà agli utenti di connettersi. Essi vedranno il contenuto di questo file ed il loro login sarà rifiutato. Solo root potrà accedere.

Se l'utente è root allora il login deve avvenire su una delle tty elencate in /etc/securetty. Fallimenti saranno registrati con l'apposita funzione del syslog. Con presenti entrambi questi controlli l'unico modo per accedere al firewall è attraverso la console e solo come root.

MAI e poi MAI effettuare un telnet verso un sistema e accedere come ROOT. Se è necessario accedere in remoto come root utilizzare SSH (Secure Shell). Si potrebbe perfino disabilitare il telnet.

Se si è davvero paranoici allora si dovrebbe utilizzare lids (Linux Intrusion Detect System), una patch di scoperta delle intrusioni per il kernel Linux; può proteggere file importanti evitando che siano modificati.

Quando è presente, nessuno (incluso root) può cambiare i file, le directory o sotto-directory protette. Per modificare i file protetti è necessario riavviare il sistema con l'impostazione security=1 nel LILO (riavvio in single user mode).

# 7 Impostazione del filtraggio IP (IPFWADM)

Se si sta utilizzando un kernel 2.1.102 o più recente passare alla sezione riguardante IPCHAINS.

Nei vecchi kernel l'IP Forwarding è abilitato per default. Per questo motivo la propria rete dovrebbe cominciare vietando l'accesso a chiunque e cancellando tutte le regole impostate l'ultima volta. Questo frammento di script dovrebbe essere collocato nello script di startup della propria rete (/etc/rc.d/init.d/network).

```
  #
  # Imposta IP packet Accounting e Forwarding
  #
  #   Forwarding
  #
  # Di default NEGA tutti i servizi
  ipfwadm -F -p deny
  # Cancella tutti i comandi
  ipfwadm -F -f
  ipfwadm -I -f
  ipfwadm -O -f
```

Ora abbiamo il firewall definitivo. Nulla può passare.

Si crei il file /etc/rc.d/rc.firewall. Questo script dovrebbe consentire traffico email, Web e DNS. ;-)

```
#! /bin/sh
#
# rc.firewall
#
# Libreria funzioni.
. /etc/rc.d/init.d/functions

# avvia configurazione.
. /etc/sysconfig/network

# Controlliamo che la rete sia presente
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi
case "$1" in
  start)
  echo -n "Starting Firewall Services: "
  # Inoltro delle email al proprio server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25
  # Consenti connessione email verso server email esterni
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.10 25 -D 0.0.0.0/0 1024:65535
  # Consenti accesso web al proprio web server
```

```
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.11 80
  # Consenti connessioni Web verso server Web esterni
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.* 80 -D 0.0.0.0/0 1024:65535
  # Consenti traffico DNS
  /sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 192.1.2.0/24
  ;;
  stop)
  echo -n "Stooping Firewall Services: "
  ipfwadm -F -p deny
  ;;
  status)
  echo -n "Now do you show firewall stats?"
  ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: firewall {start|stop|status|restart|reload}"
        exit 1
esac
```

NOTA: in questo esempio l'email server è presente all'indirizzo 192.1.2.10 dove dovrebbe essere in grado di inviare e riceve sulla porta 25.

Il server web è presente all'indirizzo 192.1.2.11. Consentiamo a chiunque appartenente alla LAN di accedere ai server web e DNS esterni.

Ciò non è perfettamente sicuro, in quanto la porta 80 non dovrebbe essere usata come porta web, un hacker acuto potrebbe utilizzarla per creare una rete privata virtuale (VPN) attraverso il firewall. Per raggirare questa situazione è necessario impostare un proxy web, e consentire solo il proxy attraverso il firewall. Gli utenti della rete LAN devono passare attraverso il proxy per accedere ai server web esterni.

Si potrebbe essere interessati a tenere conto del traffico che passa attraverso il firewall. Il seguente script conta tutti i pacchetti. E' possibile aggiungere una o due linee per contare i pacchetti diretti solo verso un determinato sistema.

```
  # Cancella le regole correnti di "conteggio"
  ipfwadm -A -f
  # Conteggio
  /sbin/ipfwadm -A -f
  /sbin/ipfwadm -A out -i -S 192.1.2.0/24 -D 0.0.0.0/0
  /sbin/ipfwadm -A out -i -S 0.0.0.0/0    -D 192.1.2.0/24
  /sbin/ipfwadm -A in  -i -S 192.1.2.0/24 -D 0.0.0.0/0
  /sbin/ipfwadm -A in  -i -S 0.0.0.0/0    -D 192.1.2.0/24
```

Se tutto ciò di cui si ha bisogno è un firewall filtrante allora ci si può fermare qui. Si effettuino dei test e se la si goda.

# 8 Impostazione del filtraggio IP (IPCHAINS)

Linux ipchains è la riscrittura del codice IPv4 di filtraggio di Linux e di ipfwadm, il quale a sua volta è una riscrittura di ipfw della BSD, credo. È richiesto per l'amministrazione dei filtri dei pacchetti IP nei kernel Linux versione 2.1.102 e successivi.

Il vecchio codice non gestisce i frammenti, ha contatori 32-bit (almeno su Intel), non consente la specificazione di protocolli che non siano TCP, UDP o ICMP, non può effettuare grossi cambiamenti atomicamente, non è possibile specificare regole inverse, ha alcune bizzarrie, e può essere difficile da utilizzare (incline a far commettere errori all'utente). Così almeno secondo l'autore.

Non mi accingo ad approfondire come gestire un firewall con IPChains in quanto esiste un'ECCELLENTE!! HOWTO all'indirizzo

*http://www.rustcorp.com/linux/ipchains/HOWTO.html* `http://www.rustcorp.com/linux/ipchains/HOWTO.html` .

Si lavora con le catene per nome. Si parte con tre catene predefinite input, output e forward che non possono essere rimosse. Si possono anche creare proprie catene. Le regole, da questo sistema, possono essere aggiunte e cancellate.

Le operazioni per operare su intere catene sono:

1. Crea una nuova catena (-N).
2. Cancella una catena vuota (-X).
3. Cambia la tattica di una delle catene predefinite (-P).
4. Elenca le regole di una delle catene (-L).
5. Svuota una catena delle sue regole (-F).
6. Azzera i contatori dei pacchetti e dei byte di tutte le regole di una catena (-Z).

Esistono inoltre diversi modi per manipolare le regole di una catena:

1. Appendi una nuova regola ad una catena (-A).
2. Inserisci una nuova regola in una determinata posizione della catena (-I).
3. Sostituisci una regola presente in una determinata posizione di una catena (-R).
4. Cancella una regola presente in una determinata posizione della catena (-D).
5. Cancella la prima regola di una catena (-D).

Ci sono anche alcune operazioni riguardanti il masquerading, incluse in ipchains in quanto è una buona collocazione:

1. Elenca le connessioni correntemente mascherate (-M -L).
2. Imposta i timeout del masquerading (-M -S).

Esistono alcuni problemi di tempo riguardanti la modifica delle regole del firewall.

Se non si presta particolare attenzione, si potrebbero far passare dei pacchetti mentre si è nel bel mezzo delle modifiche.

```
# ipchains -I input 1 -j DENY
# ipchains -I output 1 -j DENY
# ipchains -I forward 1 -j DENY
```

... effettua le modifiche ...

```
# ipchains -D input 1
# ipchains -D output 1
# ipchains -D forward 1
#
```

Questa soluzione scarta tutti i pacchetti per tutta la durata delle modifiche.

Segue un duplicato, adattato per IPChains, delle regole precedenti.

```
#!/bin/sh
#
# rc.firewall
#
## Svuota tutto, cominciamo da zero
  /sbin/ipchains -F input
  /sbin/ipchains -F output
  /sbin/ipchains -F forward

## Redirezione, necessaria per il proxy HTTP trasparente
  #$IPCHAINS  -A input -p tcp -s 192.1.2.0/24 -d 0.0.0.0/0 80 -j REDIRECT 8080

## Creazione della propria catena
  /sbin/ipchains -N my-chain
  # Consentiamo alle email di giungere al server
  /sbin/ipchains -A my-chain -s 0.0.0.0/0 smtp -d 192.1.2.10 1024:-j ACCEPT
  # Consentiamo connessioni email verso server email esterni
  /sbin/ipchains -A my-chain -s 192.1.2.10 -d 0.0.0.0/0 smtp -j ACCEPT
  # Consentiamo connessioni Web dirette al proprio server Web
  /sbin/ipchains -A my-chain -s 0.0.0.0/0 www -d 192.1.2.11 1024: -j ACCEPT
  # Consentiamo connessioni Web dirette verso server Web esterni
  /sbin/ipchains -A my-chain -s 192.1.2.0/24 1024: -d 0.0.0.0/0 www -j ACCEPT
  # Consentiamo traffico DNS
  /sbin/ipchains -A my-chain -p UDP -s 0.0.0.0/0 dns -d 192.1.2.0/24 -j ACCEPT

## Se si utilizza il masquerading
  # Non mascherare il traffico interno-interno
  /sbin/ipchains -A forward -s 192.1.2.0/24 -d 192.1.2.0/24 -j ACCEPT
  # Non mascherare direttamente l'interfaccia esterna
  /sbin/ipchains -A forward -s 24.94.1.0/24 -d 0.0.0.0/0 -j ACCEPT
  # Maschera tutti gli IP interni diretti all'esterno
  /sbin/ipchains -A forward -s 192.1.2.0/24 -d 0.0.0.0/0 -j MASQ

## Scarta qualsiasi altra cosa
  /sbin/ipchains -P my-chain input DENY
```

Non ci si fermi qui. Questo non è un gran firewall e sono sicuro che si vorranno fornire altri servizi. Ancora, si legga l'IPCHAINS-HOWTO.

# 9 Installare un proxy trasparente SQUID

Il proxy squid è disponibile all'indirizzo ***http://squid.nlanr.net/*** `http://squid.nlanr.net/` .

Gli sviluppatori di SQUID mettono a disposizione pacchetti per RedHat e Debian. Se possibile utilizzare uno di questi.

# 10 Installare il proxy server TIS

## 10.1 Reperire il software

TIS FWTK è disponibile all'indirizzo ***http://www.tis.com/research/software/*** `http://www.tis.com/research/software/` .

**Non fate il mio stesso errore.** Quando si prelevano i file da TIS, SI LEGGA IL README. TIS fwtk è racchiuso in una directory nascosta del loro server.

TIS richiede che venga letto il loro contratto all'indirizzo *http://www.tis.com/research/software/fwtk_readme.html* `http://www.tis.com/research/software/fwtk_readme.html` e che sia **inviata, per apprendere il nome della directory nascosta un'email all'indirizzo *fwtk-request@tislabs.com*** `mailto:fwtk-request@tislabs.com` con presente, nel corpo del messaggio, la sola parola **accepted**

Nessun argomento è richiesto nell'oggetto. Il loro sistema provvederà a inviare il nome della directory (buona per 12 ore) dove poter prelevare il sorgente.

Al momento della stesura, la versione corrente di FWTK è la 2.1.

Ciò che ora rimane da fare è la configurazione del firewall.

## 10.2 Compilare TIS FWTK

La compilazione della versione 2.1 di FWTK è molto più semplice rispetto a qualsiasi altra versione precedente.

SPIEGA QUI!!!

Ora impartire **make**.

## 10.3 Installare TIS FWTK

Digitare **make install**.

La directory di default dell'installazione è /usr/local/etc. E' possibile cambiarla (non l'ho fatto) con una più sicura. Ho scelto di cambiare l'accesso a questa directory con 'chmod 700'.

## 10.4 Configurare TIS FWTK

Ora comincia il vero divertimento. E' necessario imparare il sistema per chiamare questi nuovi servizi e creare le tabelle per controllarli. Non ho intenzione qui di riscrivere il manuale di TIS FWTK, voglio solo mostrare le impostazioni che ho trovato funzionanti e spiegare i problemi che ho incontrato e come li ho risolti.

Esistono tre file che riguardano questi controlli:

- /etc/services
  - Segnala al sistema a quali porte il servizio è presente
- /etc/inetd.conf
  - Segnala a inetd quali programmi richiamare quando qualcuno bussa ad una determinata porta
- /usr/local/etc/netperm-table
  - Specifica chi può accedere e chi no ai servizi

Perché FWTK funzioni, è necessario modificare questi file da principio. Modificare i file dei servizi senza che siano impostati correttamente i file inetd.conf o netperm-table può rendere il proprio sistema inaccessibile.

### 10.4.1 Il file netperm-table

Questo file controlla chi può accedere ai servizi di TIS FWTK. Il traffico si può considerare diretto ad entrambi i lati del firewall. Le persone all'esterno della propria rete dovrebbero identificarsi prima di poter guadagnare l'accesso, mentre agli utenti della rete locale dovrebbe essere consentito di passare.

In questo modo le persone possono identificarsi; il firewall utilizza un programma **authsrv** per mantenere un database delle user ID e delle password. La sezione della netperm-table riguardante l'autenticazione controlla dove è collocato il database e chi vi può accedere.

Ho avuto alcuni problemi nel chiudere l'accesso a questo servizio. Nota infatti la presenza nella linea permit-hosts del carattere * usato per consentire l'accesso a chiunque. L'impostazione corretta di questa linea, se vi dovesse funzionare, è " `authsrv: permit-hosts localhost`.

```
#
# Tabella di configurazione del proxy
#
# server di autenticazione e regole clienti
authsrv:      database /usr/local/etc/fw-authdb
authsrv:      permit-hosts *
authsrv:      badsleep 1200
authsrv:      nobogus true
# Applicazioni client che utilizzano il server authentication
*:            authserver 127.0.0.1 114
```

Per inizializzare il database e creare il record user administrative, effettuare un su root e avviare **./authsrv** nella directory /var/local/etc. Qui è presente una semplice sezione.

Si legga la documentazione di FWTK per imparare come aggiungere utenti e gruppi.

```
#
# authsrv
authsrv# list
authsrv# adduser admin "Auth DB admin"
```

```
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# list
Report for users in database
user   group  longname           ok?    proto   last
------ ------ ------------------ -----  ------  -----
admin         Auth DB admin      ena    passw   never
authsrv# display admin
Report for user admin (Auth DB admin)
Authentication protocol: password
Flags: WIZARD
authsrv# ^D
EOT
#
```

I controlli del gateway telnet (tn-gw) sono i primi da impostare.

Nell'esempio, autorizzo gli host presenti all'interno della rete privata a passare senza autenticarsi (permit-hosts 19961.2.* -passok). Ogni altro utente invece dovrà inserire il proprio user ID e la password per utilizzare il proxy (permit-hosts * -auth).

Inoltre, consento ad un altro sistema (196.1.2.202) di accedere direttamente al firewall senza passare attraverso il firewall stesso. Le due righe inetacl-in.telnetd servono a definire questo. Più avanti sarà spiegato come queste righe sono richiamate.

Il timeout Telnet dovrebbe essere mantenuto breve.

```
# regole del gateway telnet:
tn-gw:              denial-msg      /usr/local/etc/tn-deny.txt
tn-gw:              welcome-msg     /usr/local/etc/tn-welcome.txt
tn-gw:              help-msg        /usr/local/etc/tn-help.txt
tn-gw:              timeout 90
tn-gw:              permit-hosts 192.1.2.* -passok -xok
tn-gw:              permit-hosts * -auth
# Solo l'amministratore può effettuare telnet direttamente al Firewall
# tramite la Porta 24
netacl-in.telnetd: permit-hosts 192.1.2.202 -exec /usr/sbin/in.telnetd
```

I comandi r funzionano allo stesso modo del telnet.

```
# rlogin gateway rules:
# regole gateway rlogin:
rlogin-gw:    denial-msg      /usr/local/etc/rlogin-deny.txt
rlogin-gw:    welcome-msg     /usr/local/etc/rlogin-welcome.txt
rlogin-gw:    help-msg        /usr/local/etc/rlogin-help.txt
```

```
rlogin-gw:    timeout 90
rlogin-gw:    permit-hosts 192.1.2.* -passok -xok
rlogin-gw:    permit-hosts * -auth -xok
# Solo l'Amministratore può eseguire direttamente il telnet al Firewall
netacl-rlogind: permit-hosts 192.1.2.202 -exec /usr/libexec/rlogind -a
```

È consigliabile che nessuno possa accedere direttamente al firewall, incluso l'accesso in FTP. Pertanto, non si metta un server FTP sul proprio firewall.

Inoltre, la riga permit-hosts consente a chiunque all'interno della rete protetta di accedere liberamente ad Internet, mentre tutti gli altri devono autenticarsi. Ai miei controlli sono stati aggiunti il log di ogni file inviato e ricevuto (-log { retr stor }).

Il timeout ftp controlla quanto tempo ci vuole per far cadere una cattiva connessione come pure quanto a lungo può rimanere aperta una connessione senza attività.

```
# regole gateway ftp:
ftp-gw:           denial-msg     /usr/local/etc/ftp-deny.txt
ftp-gw:           welcome-msg    /usr/local/etc/ftp-welcome.txt
ftp-gw:           help-msg       /usr/local/etc/ftp-help.txt
ftp-gw:           timeout 300
ftp-gw:           permit-hosts 192.1.2.* -log { retr stor }
ftp-gw:           permit-hosts * -authall -log { retr stor }
```

I web, gopher e browser basati su ftp sono stravolti dall'http-gw. Le prime due righe creano una directory dove poter memorizzare i documenti ftp e web esattamente come passano attraverso il firewall. Ho reso questi file di proprietà di root e li ho collocati in una directory accessibile solo dal root.

La connessione Web dovrebbe essere tenuta breve. Viene inoltre effettuato un controllo sul tempo di attesa di un utente su una cattiva connessione.

```
# regole gateway www e gopher:
http-gw:      userid         root
http-gw:      directory      /jail
http-gw:      timeout 90
http-gw:      default-httpd  www.afs.net
http-gw:      hosts          192.1.2.* -log { read write ftp }
http-gw:      deny-hosts     *
```

ssl-gw è di fatto un semplice gateway passatutto. Prestategli attenzione. In questo esempio consento a tutti all'interno della rete protetta di connettersi a qualsiasi server al di fuori della rete, fatta eccezione per gli indirizzi 127.0.0.* e 192.1.1.*, e solo sulle porte da 443 a 563. Le porte da 443 a 563 sono conosciute come porte SSL.

```
# regole gateway ssl:
ssl-gw:        timeout 300
ssl-gw:        hosts          192.1.2.* -dest { !127.0.0.* !192.1.1.* *:443:563 }
ssl-gw:        deny-hosts     *
```

Segue un esempio di come utilizzare il plug-gw per consentire connessioni ad un server news. Nell'esempio, si abilitano tutti gli utenti all'interno della rete privata a connettersi ad un solo sistema e solo alla sua porta news.

La seconda riga consente al server news di ripassare i dati alla rete protetta.

Dal momento che la maggior parte dei client si aspettano di restare connessi mentre gli utenti leggono le news, il timeout per un server di news dovrebbe essere lungo.

```
# NetNews Pluged gateway
plug-gw:        timeout 3600
plug-gw: port nntp 192.1.2.* -plug-to 24.94.1.22 -port nntp
plug-gw: port nntp 24.94.1.22 -plug-to 192.1.2.* -port nntp
```

Il gateway finger è semplice. Chiunque dall'interno della rete protetta deve prima di tutto eseguire il login e solo dopo può ottenere l'abilitazione a utilizzare il programma finger sul firewall. Tutti gli altri invece ricevono semplicemente un messaggio.

```
# Abilitazione del servizio finger
netacl-fingerd: permit-hosts 192.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

Non ho impostato i servizi Mail e X-windows pertanto non aggiungo degli esempi in merito. Se qualcuno possiede un esempio funzionante è pregato di inviarmi un'email.

### 10.4.2 The /etc/services file

Qui è dove tutto comincia. Quando un client si connette al firewall lo fa su una porta conosciuta (minore di 1024). Ad esempio telnet si connette sulla porta 23. Il demone inetd sente questa connessione e cerca il nome di questo servizio nel file /etc/services. Quindi, richiama il programma assegnato al nome nel file /etc/inetd.conf.

Alcuni dei servizi che stiamo creando non si trovano normalmente nel file /etc/services. È possibile assegnare alcuni di essi ad una porta qualsiasi. Ad esempio, ho assegnato la porta corrispondente al telnet dell'amministratore (telnet-a) alla porta 24. Volendo, lo si può assegnare alla porta 23. Affinché l'amministratore (ossia voi stessi) possa connettersi direttamente al firewall è necessario eseguire il telnet alla porta 24 e non alla 23, e se il file netperm-table viene impostato, come ho fatto io, sarà possibile farlo solamente da un sistema all'interno della propria rete protetta.

```
telnet-a        24/tcp
ftp-gw          21/tcp              # questo nome è cambiato
auth            113/tcp   ident    # verifica dell'utente
ssl-gw          443/tcp
```

# 11 Il proxy server SOCKS

## 11.1 Impostare il server proxy

Il proxy server SOCKS è disponibile all'indirizzo **http://www.socks.nec.com/**.

Decomprimere e decompattare (untar) i file all'interno di una directory del proprio sistema, e seguire le istruzioni su come effettuare il make. Personalmente ho avuto un paio di problemi in quest'ultimo caso. Assicurarsi che i Makefile siano corretti.

È importante notare che il proxy server deve essere aggiunto nel file /etc/inetd.conf. È necessario quindi aggiungere la riga:

```
socks  stream  tcp  nowait  nobody  /usr/local/etc/sockd  sockd
```

per segnalare al server di entrare in esecuzione quando richiesto.

## 11.2 Configurare il Proxy Server

Il programma SOCKS ha bisogno di due file di configurazione separati. Il primo per specificare gli accessi autorizzati, il secondo per instradare le richieste al proxy server appropriato. Il file di accesso dovrebbe essere localizzato sul server. Il file di instradamento dovrebbe trovarsi su ogni macchina Unix. I computer DOS e presumibilmente i Macintosh effettueranno il proprio instradamento.

### 11.2.1 Il file di accesso

Con socks 4.2c Beta, il file di accesso è denominato "sockd.conf". Dovrebbe contenere 2 righe, una riga permit e una riga deny. Ogni riga conterrà tre campi:

- L'identificatore (permit/deny)
- L'indirizzo IP
- Il modificatore di indirizzo

L'identificatore è di tipo permit oppure deny. È consigliabile avere sia la riga permit, sia la riga deny.

L'indirizzo IP è un indirizzo a 4 byte come nella tipica notazione IP. Ossia, 192.168.1.0.

Anche il modificatore di indirizzo è un tipico indirizzo IP rappresentato da un numero di 4 byte e funziona come una netmask. Supponiamo che questo numero sia a 32 bit (serie di 1 o di 0). Se il bit è un 1, il bit corrispondente dell'indirizzo che si sta controllando deve essere uguale al bit corrispondente presente nel campo dell'indirizzo IP.

Ad esempio, se la riga è:

```
permit 192.168.1.23 255.255.255.255
```

saranno ammessi solo gli indirizzi IP i cui bit corrispondono a 192.168.1.23, ossia, solo 192.168.1.3. La riga:

```
permit 192.168.1.0 255.255.255.0
```

ammetterà ogni numero all'interno del gruppo da 192.168.1.0 a 192.168.1.255, ossia l'intero dominio di Classe C. Non si dovrebbe invece avere la riga:

```
permit 192.168.1.0 0.0.0.0
```

dal momento che ammetterà qualsiasi indirizzo, indistintamente.

Pertanto, prima di tutto si abilitino tutti gli indirizzi che si vogliono abilitare e si neghino i restanti. Per ammettere tutti coloro che sono presenti nel dominio 192.168.1.xxx, le righe:

```
permit 192.168.1.0 255.255.255.0
deny 0.0.0.0 0.0.0.0
```

funzioneranno correttamente. Notare il primo "0.0.0.0" della riga deny. Con un modificatore 0.0.0.0, il campo dell'indirizzo IP non è rilevante. Tutti zero è la norma perché è semplice digitarli.

È consentita più di una voce per ciascun tipo.

È anche possibile fornire o negare gli accessi a degli utenti specifici, tramite l'autenticazione ident. Non tutti i sistemi supportano ident, tra cui Trumpet Winsock, pertanto questo argomento non sarà trattato in questa sede. La documentazione a corredo di socks è del tutto adeguata riguardo questo argomento.

### 11.2.2 Il file di instradamento

Il file di instradamento in SOCKS è infelicemente chiamato "socks.conf". L'"infelicemente" è dovuto al fatto che appare molto simile a quello del file di accesso, pertanto potrebbe essere facile confonderli.

Il file di instradamento informa il client SOCKS quando utilizzare socks e quando non farlo. Ad esempio, nella nostra rete, 192.168.1.3 non avrà bisogno di utilizzare socks per parlare con 192.168.1.1, il firewall, in quanto possiede una connessione diretta via Ethernet e definisce automaticamente 127.0.0.1, ossia il loopback. Naturalmente non è necessario SOCKS per parlare con se stessi. Esistono tre voci:

- deny
- direct
- sockd

Deny specifica a SOCKS quando respingere una richiesta. Questa voce possiede gli stessi tre campi già descritti per sockd.conf: identificatore, indirizzo e modificatore. Generalmente, dal momento che questo viene gestito anche da sockd.conf, il file di accesso, il campo del modificatore è impostato a 0.0.0.0. Se si vuole precludere se stessi dal chiamare qualsiasi posto, può essere fatto in questo punto.

La voce direct specifica gli indirizzi per i quali non deve essere utilizzato socks. Si tratta degli indirizzi che possono essere raggiunti senza il proxy server. Ancora una volta, abbiamo i tre campi: identificatore, indirizzo e modificatore. Nel nostro esempio corrisponderebbe a:

```
direct 192.168.1.0 255.255.255.0
```

che permette di andare direttamente ovunque nella nostra rete protetta.

La voce sockd infine specifica al computer quale host ha in esecuzione il demone server socks. La sintassi è:

```
sockd @=<serverlist> <IP address> <modifier>
```

Si noti la voce @=. Questa permette di impostare gli indirizzi IP di una lista di proxy server. Nel nostro esempio, viene utilizzato un unico proxy server. Tuttavia è possibile averne molti per consentire un carico maggiore e per avere a disposizione una ridondanza in caso di errore.

I campi di indirizzo IP e di modificatore funzionano esattamente come negli altri esempi. Attraverso questi è possibile specificare dove devono andare gli indirizzi.

### 11.2.3 DNS presente dietro il Firewall

L'impostazione del Domain Name Service da dietro un firewall è un compito relativamente semplice. Prima è necessario impostare il DNS sulla macchina firewall e poi configurare ogni macchina dietro al firewall in modo che possa utilizzare questo DNS.

## 11.3 Lavorare con un Proxy Server

### 11.3.1 Unix

Per fare in modo che le proprie applicazioni funzionino con il proxy server, dovranno essere "SOCKettizzate". Saranno necessarie due diverse tipologie di telnet, una per la comunicazione diretta e una per la comunicazione tramite il proxy server. SOCKS fornisce delle istruzioni su come SOCKettizzare un programma, come pure un paio di programmi pre-SOCKettizzati. Se si utilizza una versione SOCKettizzata per andare direttamente da qualche parte, SOCKS si commuterà automaticamente nella versione diretta. Per questo motivo si vorranno rinominare tutti i programmi sulla rete protetta e sostituirli con i programmi SOCKettizzati. "Finger" diventerà "finger.orig", "telnet" diventerà "telnet.orig" ecc. Bisognerà inoltre informare SOCKS di ognuno di questi cambiamenti tramite il file include/socks.h.

Alcuni programmi saranno in grado di gestire l'instradamento e la SOCKettizzazione per conto loro. Netscape è uno di questi. È possibile utilizzare un proxy server sotto Netscape inserendo l'indirizzo del server (192.168.1.1 nel nostro caso) nel campo SOCKS sotto i Proxy. Ciascuna applicazione avrà bisogno di almeno qualche modifica, indipendentemente da come gestisce un proxy server.

### 11.3.2 MS Windows con Trumpet Winsock

Trumpet Winsock viene già distribuito con il supporto intrinseco per i server proxy. Nel menu di "setup", si inseriscano l'indirizzo IP del server, e gli indirizzi di tutti i computer raggiungibili direttamente. Trumpet provvederà poi a gestire tutti i pacchetti in uscita.

### 11.3.3 Come far funzionare il Proxy Server con i pacchetti UDP

Il pacchetto SOCKS funziona solamente con i pacchetti TCP, non con quelli UDP. Questa caratteristica lo rende leggermente meno utile. Molti programmi interessanti, come talk e Archie, utilizzano UDP. Esiste un pacchetto studiato per essere usato come proxy server per pacchetti UDP denominato UDPrelay, di Tom Fitzgerald <fitz@wang.com>. Sfortunatamente, nel momento in cui viene scritto questo documento, non è compatibile con Linux.

## 11.4 Svantaggi dei Proxy Server

Il proxy server è soprattutto un `dispositivo di sicurezza`. Un suo utilizzo per aumentare l'accesso ad internet con limitati indirizzi IP causerà molti svantaggi. Un proxy server consentirà un maggior accesso dall'interno della rete protetta verso l'esterno, ma manterrà l'interno completamente inaccessibile dall'esterno. Ciò implica l'impossibilità di avere connessioni server, talk o archie oppure mail dirette verso i computer presenti all'interno. Questi svantaggi potrebbero sembrare irrilevanti, ma bisogna pensare ad essi in questi termini:

- Un report su cui state lavorando sul vostro computer è stato lasciato all'interno della rete protetta con firewall. Vi trovate a casa, e decidete di lavorarci ancora un po'. Ma questo non è possibile. Non potete raggiungere il vostro computer perché si trova al di là del firewall. Per prima cosa cercherete di connettervi al `firewall`, ma dal momento che tutti hanno un accesso proxy server, nessuno avrà impostato un account per voi.
- Vostra figlia frequenta il college. Volete inviarle un'email. Avete alcuni affari personali di cui parlare, e preferireste poter ricevere la posta direttamente sulla vostra macchina. Avete completa fiducia nell'amministratore del sistema, tuttavia si tratta pur sempre di posta privata.

- L'incapacità di utilizzare i pacchetti UDP rappresenta un grande svantaggio dei proxy server. Immagino che il supporto UDP sarà disponibile a breve.

FTP provoca un altro problema con i proxy server. Quando si riceve o si esegue un comando `ls`, il server FTP apre un socket sulla macchina client e inoltre la utilizza per inviare le informazioni. Un proxy server non permetterà queste operazioni, pertanto FTP non funzionerà molto bene.

Inoltre, i proxy server sono lenti. A causa del sovraccarico maggiore, quasi ogni altro mezzo per ottenere questo accesso sarà più veloce.

Sostanzialmente, se si possiedono gli indirizzi IP, e non ci si preoccupa della sicurezza, è meglio non utilizzare un firewall e/o i proxy server. Se non si possiedono gli indirizzi IP, e non ci si preoccupa della sicurezza, si potrebbe pensare di utilizzare un emulatore IP, come Term, Slirp o TIA. Term è disponibile su `ftp://sunsite.unc.edu`, Slirp su `ftp://blitzen.canberra.edu.au/pub/slirp`, e TIA su `marketplace.com`. Questi pacchetti saranno più veloci, consentiranno connessioni migliori e forniranno un livello maggiore di accesso alla rete interna da internet. I proxy server sono ottimi nel caso di reti con molti host che richiedono di connettersi alla rete esterna al volo, con una sola impostazione e con poco lavoro successivo.

# 12 Configurazioni avanzate

Esiste un'altra configurazione di cui vorrei parlare prima di concludere questo documento. Quelle che ho già descritto probabilmente saranno sufficienti per la maggior parte delle persone. Tuttavia, ho intenzione di mostrare una configurazione più avanzata in grado di chiarire alcune questioni. Se avete domande relative a quanto è stato descritto finora, o se siete semplicemente interessati alla versatilità dei proxy server e dei firewall, continuate la lettura.

## 12.1 Una rete ampia con enfasi sulla sicurezza

Supponiamo, ad esempio, di voler mettere in rete il nostro sito. Si possiedono 50 computer e una sottorete di 32 (5 bit) numeri IP. Servono diversi livelli di accesso all'interno della rete. Pertanto, è necessario proteggere certe parti della rete dal resto.

I livelli sono:

1. Il livello esterno. Si tratta del livello disponibile a tutti. È il luogo dove si cercano nuovi volontari.
2. **Truppa**. Questo è il livello di persone che hanno superato il livello esterno. È il luogo dove vengono istruiti sul governo diabolico e su come fabbricare le bombe.
3. **Mercenari**. Questo è il livello dove sono tenuti i piani *veri*. In questo livello sono memorizzate tutte le informazioni su come il governo del terzo mondo ha intenzione di conquistare il mondo, i piani che coinvolgono Newt Gingrich, Oklahoma City, i prodotti di minor importanza e cosa è immagazzinato veramente negli hangar dell'area 51.

### 12.1.1 Impostazione della rete

I numeri IP sono arrangiati in modo tale che:

- Un numero utilizzato sia 192.168.1.255, che rappresenta l'indirizzo di broadcast non utilizzabile.
- 23 dei 32 indirizzi IP siano allocati a 23 macchine che saranno poi accessibili da Internet.

- Un IP extra sia assegnato ad una Linux box della rete.
- Un IP extra sia assegnato ad un'altra Linux box della rete.
- Due IP siano assegnati al router.
- Quattro siano lasciati liberi, ma ad essi siano comunque assegnati nomi di dominio quali paul, ringo, john, e george, tanto per confondere un po' le idee.
- Le reti protette abbiano entrambe gli indirizzi 192.168.1.xxx.

Quindi, vengono costruite due reti separate, ognuna localizzata in stanze diverse. L'instradamento può avvenire tramite Ethernet a infrarossi in modo che sia completamente invisibile alle postazioni esterne. Fortunatamente, l'ethernet a infrarossi funziona esattamente come l'ethernet normale.

Queste reti sono entrambe connesse ad una delle box Linux tramite un indirizzo IP extra.

Esiste un file server che connette le due reti protette. Questo perché i piani per conquistare il mondo coinvolgono alcune delle Truppe di livello più alto. Il file server mantiene l'indirizzo 192.168.1.17 della rete della Truppa e l'indirizzo 192.168.1.23 della rete dei Mercenari. Deve avere due indirizzi IP differenti poiché possiede due diverse schede Ethernet. L'IP Forwarding è disabilitato.

Il forwarding IP è disabilitato anche su entrambe le Linux box. Il router non inoltrerà pacchetti destinati a 192.168.1.xxx se non gli sarà richiesto esplicitamente di farlo, pertanto Internet non sarà in grado di entrare. La ragione per cui si è scelto di disabilitare l'IP Forwarding è che in questo modo i pacchetti provenienti dalla rete della Truppa non saranno in grado di raggiungere la rete dei Mercenari, e viceversa.

Il server NFS può essere impostato in modo da offrire file diversi a reti diverse. Questo può tornare utile, e un piccolo trucco con i link simbolici può fare in modo che i file comuni possano essere condivisi con chiunque. L'utilizzo di questa impostazione e di un'altra scheda ethernet può offrire questo unico file server a tutte e tre le reti.

### 12.1.2 Impostazione del Proxy

Ora, dal momento che tutti e tre i livelli vogliono essere in grado di monitorare la rete per i propri scopi, tutti e tre hanno bisogno di un accesso alla rete. La rete esterna è connessa direttamente ad internet, pertanto in questo caso non dobbiamo preoccuparci dei proxy server. Le reti dei Mercenari e della Truppa si trovano al di là del firewall, pertanto è necessario impostare dei proxy server.

Entrambe le reti hanno un'impostazione molto simile. Ad entrambe vengono assegnati gli stessi indirizzi IP. Inserirò un paio di parametri, solo per rendere le cose più interessanti.

1. Nessuno può utilizzare il file server per l'accesso ad Internet. Questo esporrebbe il file server a virus ed altri problemi, ed è quindi molto importante e assolutamente da evitare.
2. Non verrà consentito l'accesso della Truppa al World Wide Web, dal momento che sono in addestramento, questo potere di recupero di informazioni potrebbe essere pericoloso.

Pertanto, il file sockd.conf sulla box Linux della Truppa conterrà la riga:

```
deny 192.168.1.17 255.255.255.255
```

e sulla macchina Mercenaria:

```
deny 192.168.1.23 255.255.255.255
```

Inoltre, la box Linux della Truppa avrà anche la seguente linea:

```
deny 0.0.0.0 0.0.0.0 eq 80
```

che indica di negare l'accesso a tutte le macchine che cercano di accedere alla porta uguale (eq) a 80, la porta http. Questa continuerà a consentire tutti gli altri servizi, nega solo l'accesso al Web.

Quindi, entrambi i file conterranno:

```
permit 192.168.1.0 255.255.255.0
```

per consentire a tutti i computer sulla rete 192.168.1.xxx di utilizzare questo proxy server, fatta eccezione per quelli ai quali l'accesso è già stato negato (cioè, il file server e l'accesso al Web per la rete Truppa).

Il file sockd.conf della Truppa avrà il seguente formato:

```
deny 192.168.1.17 255.255.255.255
deny 0.0.0.0 0.0.0.0 eq 80
permit 192.168.1.0 255.255.255.0
```

mentre il file Mercenario avrà:

```
deny 192.168.1.23 255.255.255.255
permit 192.168.1.0 255.255.255.0
```

Questo dovrebbe configurare tutto correttamente. Ogni rete è isolata di conseguenza, con l'appropriato numero di interazioni. Tutti dovrebbero essere felici.

# 13 Semplificare la gestione

## 13.1 Tool per il firewall

Esistono molti pacchetti software che sono in grado di rendere più semplice la gestione del firewall.

Si presti però attenzione, non si utilizzino questi tool almeno che non se ne possa fare a meno. Questi script inducono a commettere errori con la stessa facilità con cui permettono di gestire le regole.

Sia le interfacce grafiche sia quelle basate sul web sono state sviluppate per operare con le regole di filtraggio di Linux. Alcune compagnie hanno perfino creato dei firewall commerciali basati su Linux, utilizzando proprie box con proprio codice di gestione (bello).

In realtà non sono un fan delle GUI. Comunque, ho utilizzato firewall con interfacce GUI per un po' di tempo e ho constatato che in effetti aiutano a fornire un comodo report di tutte le regole, per una rapida occhiata.

gfcc (GTK+ Firewall Control Center) è un'applicazione GTK+, basata su ipchains, che consente di controllare le tattiche e le regole del firewall di Linux. Si vada all'indirizzo *http://icarus.autostock.co.kr* `http://icarus.autostock.co.kr/` e ci si procuri la propria copia. Questo è un tool davvero buono.

Sono disponibili parecchi script per impostare un firewall, uno script davvero completo è presente all'indirizzo *http://www.jasmine.org.uk/~simon/bookshelf/papers/instant-firewall/instant-firewall.html* `http://www.jasmine.org.uk/~simon/bookshelf/papers/instant-firewall/instant-firewall.html` . Un altro script ben fatto è disponibile all'indirizzo *http://www.pointman.org/* `http://www.pointman.org/` .

Kfirewall è un'interfaccia GUI per ipchains o ipfwadm (dipende dal proprio kernel) *http://megaman.ypsilonia.net/kfirewall/* `http://megaman.ypsilonia.net/kfirewall/` .

FCT è un tool di configurazione di firewall basato su HTML. Comprende generazione automatica di script per comandi di filtraggio IP (ipfwadm) su firewall con interfacce multiple e per qualsiasi servizio internet *http://www.fen.baynet.de/~ft114/FCT/firewall.htm* `http://www.fen.baynet.de/~ft114/FCT/firewall.htm` .

### 13.2 Tool generici

Webmin è un pacchetto di amministrazione generale del sistema. Non è di aiuto per gestire le regole del firewall ma è comunque utile per abilitare e disabilitare i demoni e i processi. Questo programma è VERAMENTE buono, spero che J. Cameron includa un modulo per IPCHAINS.

Se siete un ISP, vi interesserà sapere qualcosa a riguardo di IPFA (IP Firewall Accounting)

*http://www.soaring-bird.com/ipfa/* `http://www.soaring-bird.com/ipfa/` . Consente di gestire log per mese/giorno/minuto, inoltre ha una GUI per l'amministrazione basata sul Web.

## 14 Raggirare un firewall proxy

Giusto per rovinare la giornata, e per far tenere i piedi per terra circa la sicurezza, descriverò quanto sia facile raggirare un firewall proxy.

Ora dopo aver fatto tutto quanto descritto in questo documento si ha un server ed una rete veramente sicuri. Si possiede un DMZ e nessuno può accedere alla rete, inoltre si registrano tutte le connessioni effettuate verso il mondo esterno. Tutti gli utenti passano per il proxy e nessuno può accedere ad Internet direttamente.

Uno degli utenti, con una connessione dedicata propria, viene però a conoscenza di

*httptunnel* `http://www.nocrew.org/software/httptunnel.html` . httptunnel crea un canale dati virtuale bidirezionale incapsulato in richieste HTTP. Se desiderato le richieste HTTP possono essere inviate attraverso il proxy HTTP.

Oppure, nel loro sistema, gli utenti potrebbero installare una Virtual Private Network (vpn). Visitare: *http://sunsite.auc.dk/vpnd/* `http://sunsite.auc.dk/vpnd/`

O ancora questo utente potrebbe aggiungere un modem al sistema NT e accedere all'instradamento.

Infine sulla workstation, nella LAN privata, potrebbe cambiare il gateway di default in modo che punti al nuovo instradamento verso Internet.

Ora, dalla workstation, si può andare ovunque. L'unica cosa che l'amministratore del firewall potrebbe notare è che qualcuno si sta connettendo con un accesso DNS molto lungo.

Siete pronti a conquistare il mondo!

## 15 APPENDICE A - Script di esempio

### 15.1 Script RC usando GFCC

```
#!/bin/bash
#
# Firewall Script - Versione 0.9.1
#
# chkconfig: 2345 09 99
# description: script firewall per i kernel 2.2.x
```

```
# Da impostare per i test:
# set -x
#
# NOTE:
#
#  Questo script è stato scritto per la RedHat 6.1 o più recente.
#
#  Prestare attenzione per quanto riguarda l'offerta di servizi pubblici quali web o ftp server.
#
# INSTALLAZIONE:
#  1. si collochi questo file nella directory /etc/rc.d/init.d  (si deve essere root..)
#     lo si chiami con qualcosa come "firewall"    :-)
#     lo si renda di proprietà del root -->  "chown root.root (filename)"
#     lo si renda eseguibile -->  "chmod 755 (filename)"
#  2. si utilizzi GFCC per creare le regole per il firewall ed esportale in un file
#     con nome /etc/gfcc/rules/firewall.rule.sh
#
#  3. si aggiunga il firewall alla struttura init della RH --> "chkconfig --add (filename)"
#     Al successivo boot del router, tutto dovrebbe sistemarsi automagicamente!
#     si dorma pure tranquilli la notte sapendo di essere *MENO* vulnerabile di prima...
#
# NOTE SULLE VERSIONI:
#   30 Jan, 2000 - Modificato per lo script GFCC
#   11 Dec, 1999 - aggiornamento di Mark Grennan <mark@grennan.com>
#   20 July, 1999 - scrittura iniziale - Anthony Ball <tony@LinuxSIG.org>

################################################

# Libreria funzioni.
. /etc/rc.d/init.d/functions

# Configurazione rete.
. /etc/sysconfig/network

# Controlla che la rete sia presente.
[ ${NETWORKING} = "no" ] && exit 0

# Controlla cosa è stato richiesto
case "$1" in

  start)
        # Inizia a fornire gli accessi
        action "Starting firewall: " /bin/true
        /etc/gfcc/rules/firewall.rule.sh
        echo
        ;;

  stop)
        action "Stoping firewall: " /bin/true
```

```
        echo 0 > /proc/sys/net/ipv4/ip_forward
        /sbin/ipchains -F input
        /sbin/ipchains -F output
        /sbin/ipchains -F forward

        echo
        ;;

  restart)
        action "Restarting firewall: " /bin/true
        $0 stop
        $0 start

        echo
        ;;

  status)
        # Visualizza elenco di tutte le regole
        /sbin/ipchains -L
        ;;

  test)
        action "Test Mode firewall: " /bin/true
        /sbin/ipchains -F input
        /sbin/ipchains -F output
        /sbin/ipchains -F forward
        echo 1 > /proc/sys/net/ipv4/ip_forward
        /sbin/ipchains -A input -j ACCEPT
        /sbin/ipchains -A output -j ACCEPT
        /sbin/ipchains -P forward DENY
        /sbin/ipchains -A forward -i $PUBLIC -j MASQ

        echo
        ;;

  *)
        echo "Usage: $0 {start|stop|restart|status|test}"
        exit 1

esac
```

## 15.2 GFCC script

Questo script è stato generato da Graphical Firewall program (GFCC). Questo non è l'insieme delle regole in funzione, è l'insieme delle regole esportate.

```
#!/bin/sh
# Generato da Gtk+ firewall control center
```

```
IPCHAINS=/sbin/ipchains


localnet="192.168.1.0/24"
firewallhost="192.168.1.1/32"
localhost="172.0.0.0/8"
DNS1="24.94.163.119/32"
DNS2="24.94.163.124/32"
Broadcast="255.255.255.255/32"
Multicast="224.0.0.0/8"
Any="0.0.0.0/0"
mail_grennan_com="192.168.1.1/32"
mark_grennan_com="192.168.1.3/32"

$IPCHAINS -P input DENY
$IPCHAINS -P forward ACCEPT
$IPCHAINS -P output ACCEPT

$IPCHAINS -F
$IPCHAINS -X

# regole catena input
$IPCHAINS -A input -s $Any -d $Broadcast -j DENY
$IPCHAINS -A input -p udp -s $Any -d $Any netbios-ns -j DENY
$IPCHAINS -A input -p tcp -s $Any -d $Any netbios-ns -j DENY
$IPCHAINS -A input -p udp -s $Any -d $Any netbios-dgm -j DENY
$IPCHAINS -A input -p tcp -s $Any -d $Any netbios-dgm -j DENY
$IPCHAINS -A input -p udp -s $Any -d $Any bootps -j DENY
$IPCHAINS -A input -p udp -s $Any -d $Any bootpc -j DENY
$IPCHAINS -A input -s $Multicast -d $Any -j DENY
$IPCHAINS -A input -s $localhost -d $Any -i lo -j ACCEPT
$IPCHAINS -A input -s $localnet -d $Any -i eth1 -j ACCEPT
$IPCHAINS -A input -s $localnet -d $Broadcast -i eth1 -j ACCEPT
$IPCHAINS -A input -p icmp -s $Any -d $Any -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any -j ACCEPT ! -y
$IPCHAINS -A input -p udp -s $DNS1 domain -d $Any 1023:65535 -j ACCEPT
$IPCHAINS -A input -p udp -s $DNS2 domain -d $Any 1023:65535 -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any ssh -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any telnet -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any smtp -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any pop-3 -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any auth -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any www -j ACCEPT
$IPCHAINS -A input -p tcp -s $Any -d $Any ftp -j ACCEPT
$IPCHAINS -A input -s $Any -d $Any -j DENY -l

# regole catena forward
$IPCHAINS -A forward -s $localnet -d $Any -j MASQ

# regole catena output
```

### 15.3 Script RC senza GFCC

Questo è l'insieme delle regole scritte di mio pugno per il firewall. Non si è utilizzato GFCC.

```
#!/bin/bash
#
# Firewall Script - Versione 0.9.0

# chkconfig: 2345 09 99
# description: script firewall per kernel 2.2.x

# Da impostare per i test:
# set -x

#
# NOTE:
#
#  Questo script è stato realizzato per essere utilizzato con la RedHat 6.0 o versioni più recenti.
#
#  Questo script dovrebbe funzionare con la maggior parte dei router, dial-up e modem.
#  E' stato scritto per le distribuzioni RedHat.
#
#  Prestare attenzione nel caso si desideri offrire servizi pubblici come web o ftp server.
#
# INSTALLAZIONE:
#  1. Questo file è pensato per un sistema RedHat. Dovrebbe
#     funzionare, forse senza apportare modifiche, anche su altre distro, ma
#     ancora...chi lo sa ?!!? Queste istruzioni riguardano i sistemi RedHat.
#
#  2. si collochi questo file in /etc/rc.d/init.d  (si dovrà essere l'utente root..)
#     e lo si denomini con qualcosa come "firewall"    :-)
#     lo si renda di proprietà del root -->  "chown root.root <filename>"
#     lo si renda eseguibile -->  "chmod 755 <filename>"
#
#  3. Si impostino i valori in base alla propria rete, alle interfacce interne, e
#     ai server DNS.
#     Si rimuovano i commenti alle linee presenti più avanti per abilitare
#     i servizi opzionali verso l'interno,
#     ci si assicuri che il proprio NIC interno sia "eth0" (altrimenti si cambi il valore
#     presente più avanti).
#     lo si provi -->  "/etc/rc.d/init.d/<filename> start"
#     si visualizzi un elenco delle regole -->  "ipchains -L -n"
#     si sistemi tutto ciò che non va...  :-)
#
#  4. Si aggiunga il firewall alla struttura init della RH --> "chkconfig --add <filename>"
#     Al boot successivo del root tutto dovrebbe impostarsi automagicamente!
#     si dorma pure tranquilli la notte sapendo di essere *MENO* vulnerabile di prima...
#
# NOTE SULLE VERSIONI:
#   20 July, 1999 - scrittura iniziale - Anthony Ball <tony@LinuxSIG.org>
```

```
#   11 Dec, 1999 - aggiornamenti di Mark Grennan <mark@grennan.com>
#

################################################
#  Sostituire i valori sottostanti con quelli
#  della propria rete locale.

PRIVATENET=xxx.xxx.xxx.xxx/xx

PUBLIC=ppp0
PRIVATE=eth0

# i propri server dns
DNS1=xxx.xxx.xxx.xxx
DNS2=xxx.xxx.xxx.xxx

################################################

# alcuni pratici valori generici da usare
ANY=0.0.0.0/0
ALLONES=255.255.255.255

# Libreria funzioni.
. /etc/rc.d/init.d/functions

# Configurazione rete.
. /etc/sysconfig/network

# Controlliamo che la rete sia presente.
[ ${NETWORKING} = "no" ] && exit 0

# Vediamo cosa è stato richiesto.
case "$1" in

  start)
        # Inizia a fornire gli accessi
        action "Starting firewall: " /bin/true

        ##
        ## Setup
        ##
        # Ripulisci tutte le liste
        /sbin/ipchains -F input
        /sbin/ipchains -F output
        /sbin/ipchains -F forward

        # Blocca qualsiasi cosa
        /sbin/ipchains -I input 1 -j DENY

        # imposta la tattica a deny (per default è ACCEPT)
```

```
/sbin/ipchains -P input DENY
/sbin/ipchains -P output ACCEPT
/sbin/ipchains -P forward ACCEPT

# Abilitiamo il packet forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

##
## Installazione dei moduli
##
# Inserire il modulo ftp attivo. Questo permetterà ftp non-passivo verso
# le macchine della rete locale (ma non verso il router in quanto non mascherato).
if ! ( /sbin/lsmod | /bin/grep masq_ftp > /dev/null ); then
    /sbin/insmod ip_masq_ftp
fi

##
## Alcune caratteristiche riguardanti la sicurezza
##
# Abilitare il Source Address Verification su tutte le interfacce
# presenti e future per ottenere la protezione dallo spoof.
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then
    for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
        echo 1 > $f
    done
else
    echo
        echo "PROBLEMI NELL'ABILITARE LA PROTEZIONE DALL'IP SPOOFING.  FARE ATTENZIONE. "
    echo
fi

# scarta bcasts sulle interfacce restanti
/sbin/ipchains -A input -d 0.0.0.0 -j DENY
/sbin/ipchains -A input -d 255.255.255.255 -j DENY

# scarta i seguenti senza loggarli in quanto tendono ad essere molti...
/sbin/ipchains -A input -p udp -d $ANY 137 -j DENY   # NetBIOS su IP
/sbin/ipchains -A input -p tcp -d $ANY 137 -j DENY   #   ""
/sbin/ipchains -A input -p udp -d $ANY 138 -j DENY   #   ""
/sbin/ipchains -A input -p tcp -d $ANY 138 -j DENY   #   ""
/sbin/ipchains -A input -p udp -d $ANY 67 -j DENY    # bootp
/sbin/ipchains -A input -p udp -d $ANY 68 -j DENY    #   ""
/sbin/ipchains -A input -s 224.0.0.0/8 -j DENY       # indirizzi Multicast

##
## Permetti alla rete locale di uscire
##
# accetta tutti i pacchetti sull'interfaccia loopback
/sbin/ipchains -A input -i lo -j ACCEPT
```

```
# accetta tutti i pacchetti provenienti dalle interfacce interne "fidate"
/sbin/ipchains -A input -i $PRIVATE -s $PRIVATENET -d $ANY -j ACCEPT
/sbin/ipchains -A input -i $PRIVATE -d $ALLONES -j ACCEPT

##
## Permetti accessi dall'esterno ai servizi del firewall (se si vuole osare)
##
# accetta pacchetti ICMP
/sbin/ipchains -A input -p icmp -j ACCEPT
# accetta pacchetti TCP
/sbin/ipchains -A input -p tcp ! -y -j ACCEPT

# permetti richieste DNS (al firewall)
/sbin/ipchains -A input -p udp -s $DNS1 domain -d $ANY 1023: -j ACCEPT
/sbin/ipchains -A input -p udp -s $DNS2 domain -d $ANY 1023: -j ACCEPT
# o (IDEA MIGLIORE) esegui un server DNS caching sul router e usa le
# seguenti due linee al loro posto ...
# /sbin/ipchains -A input -p udp -s $DNS1 domain -d $ANY domain -j ACCEPT
# /sbin/ipchains -A input -p udp -s $DNS2 domain -d $ANY domain -j ACCEPT

# rimuovi il commento dalla seguente linea per accettare richieste ssh
/sbin/ipchains -A input -p tcp -d $ANY 22 -j ACCEPT

# rimuovi il commento dalla seguente linea per accettare richieste telnet (PESSIMA IDEA!!)
/sbin/ipchains -A input -p tcp -d $ANY telnet -j ACCEPT

# rimuovi il commento dalla seguente linea per permettere NTP (network time protocol) verso il router
# /sbin/ipchains -A input -p udp -d $ANY ntp -j ACCEPT

# rimuovi il commento dalla seguente linea per permettere SMTP (non per i client mail - solo server)
/sbin/ipchains -A input -p tcp -d $ANY smtp -j ACCEPT

# rimuovi il commento per permettere POP3 (per client mail)
/sbin/ipchains -A input -p tcp -d $ANY 110 -j ACCEPT

# rimuovi il commento dalla seguente linea per inviare mail o effettuare ftp
/sbin/ipchains -A input -p tcp -d $ANY auth -j ACCEPT

# rimuovi il commento alla seguente linea per permettere HTTP (solo se si sta eseguendo un web server sul router)
/sbin/ipchains -A input -p tcp -d $ANY http -j ACCEPT

# rimuovi il commento alla seguente linea per accettare richieste FTP
/sbin/ipchains -A input -p tcp -d $ANY ftp -j ACCEPT

##
## Masquerading
##
# maschera i pacchetti provenienti dalla rete locale
/sbin/ipchains -A forward -s $PRIVATENET -d $ANY -j MASQ
```

```
        ##
        ## scarta QUALSIASI altra cosa e registrala in /var/log/messages
        ##
        /sbin/ipchains -A input -l -j DENY

        # Rimuovi il blocco
        /sbin/ipchains -D input 1

        ;;

  stop)
        action "Stoping firewall: " /bin/true
        echo 0 > /proc/sys/net/ipv4/ip_forward
        /sbin/ipchains -F input
        /sbin/ipchains -F output
        /sbin/ipchains -F forward

        echo
        ;;

  restart)
        action "Restarting firewall: " /bin/true
        $0 stop
        $0 start

        echo
        ;;

  status)
        # Elenca le impostazioni
        /sbin/ipchains -L
        ;;

  test)
        ##
        ##     E' alquanto semplice
        ##    (Non è AFFATTO sicuro)
        action "WARNING Test Firewall: " /bin/true
        /sbin/ipchains -F input
        /sbin/ipchains -F output
        /sbin/ipchains -F forward
        echo 1 > /proc/sys/net/ipv4/ip_forward
        /sbin/ipchains -A input -j ACCEPT
        /sbin/ipchains -A output -j ACCEPT
        /sbin/ipchains -P forward DENY
        /sbin/ipchains -A forward -i $PUBLIC -j MASQ

        echo
        ;;
```

```
  *)
        echo "Usage: $0 {start|stop|restart|status|test}"
        exit 1

esac


esac
```

# 16 APPENDICE B - Script RC VPN per la RedHat

```
#!/bin/sh
#
# vpnd            Questo script basato sulla shell provvederà ad avviare e
#                 a chiudere vpnd (Virtual Privage Network connections).
#
# chkconfig: - 96 96
# description: vpnd
#

# Libreria funzioni.
. /etc/rc.d/init.d/functions

# Configurazione rete.
. /etc/sysconfig/network

# Controlla che la rete sia presente.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/sbin/vpnd ] || exit 0

[ -f /etc/vpnd.conf ] || exit 0

RETVAL=0

# Vediamo cosa è stato richiesto.
case "$1" in
  start)
        # Avvia i daemon.
        echo -n "Starting vpnd: "
        daemon vpnd
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/vpnd
        echo
        ;;
  stop)
        # Ferma i daemon.
        echo -n "Shutting down vpnd: "
```

```
        killproc vpnd
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/vpnd
        echo
        ;;
  restart)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: vpnd {start|stop|restart}"
        exit 1
esac

exit $RETVAL
```

# 17 Nota sulla traduzione

La traduzione originale di questo HOWTO è opera della Apogeo `http://www.apogeonline.com/` , per il libro **Linux HowTo (La bibbia di Linux)** realizzato in collaborazione con il Pluto. L'Apogeo ha gentilmente concesso questa ed altre traduzioni ad ILDP per la sua diffusione elettronica.

Conversione in SGML e correzione a cura di Giovanni Bortolozzo *bortopluto.linux.it* , cui vanno segnalati pure eventuali errori, incongruenze ecc.

Aggiornamento alla versione 0.80 ad opera di Marco Masetti *marcomas@libero.it* .