Guida Pratica a iptables

Indice generale

1 Introduzione

Circa un anno fa, ho iniziato a studiare questo potente strumento che è netfilter, affascinato dalla sua essenzialità e dalla enorme versatilità che offre. Oltretutto sono un grande estimatore della riga di comando, poiché offre maggiore controllo rispetto ai click.

Dicevo, che quando mi sono avvicinato a netfilter subito mi è sembrato molto complicato e macchinoso, ma ora che ho potuto studiarlo a fondo, mi rendo conto che è un capolavoro ed è effettivamente molto semplice.

All'inizio ho anche cercato nei forum e in guide “non ufficiali”, pensando di trovare esempi validi come di solito mi è capitato in altri casi con software Open Source, ma dopo avere studiato a fondo i documenti ufficiali, ho visto che su questo argomento c'è molta confusione; si trovano diversi script contraddittori, mal strutturati e addirittura che minano la sicurezza degli host, e la cosa divertente è che chi li scrive spesso si spaccia per grande conoscitore! Non capite male, io non mi sento certo un guru del firewall, ma il mio intento in questa guida è fornire un po' di materiale che avrei voluto trovare io all'inizio della mia ricerca: cioè un po' di esempi semplici ed essenziali, senza tanti fronzoli al fine di fare acquisire bene la padronanza dello strumento.

Spero di avere raggiunto, anche solo in parte, questo obiettivo e di non avere fatto troppi errori di trascrizione di cui mi scuso in anticipo.

Ringrazio chiunque legga questa guida, anche soltanto l'indice.

2 Struttura di netfilter

2.1 Netfilter in Linux 2.4

Dal kernel 2.4, è stata inserita una infrastruttura per il manipolamento (mangling) dei pacchetti chiamata **netfilter,** completamente reimplementata rispetto alle versioni presenti nei kernel precedenti. Il discorso chiave di questo manipolamento è l'osservare l'intestazione dei pacchetti e il deciderne la sorte. Netfilter è parte integrante del kernel e **iptables** è il tool che è in grado di comunicare al kernel le regole per la gestione dei pacchetti.

2.2 Moduli di netfilter e operazioni preliminari

Netfilter ha diverse funzioni che possono essere abilitate inserendo i rispettivi moduli con il comando modprobe. I moduli principali sono **iptable_filter** e **ip_tables** ai quali si aggiungono altri moduli fondamentali per gestire correttamente il NAT, cioè **iptable_nat** e **ip_conntrack**.

Mentre iptable_nat è abbastanza ovvio che serve per gestire il NAT, il modulo ip_conntrack serve per il connection tracking, o tracciamento della connessione. Questo modulo registra in una tavola le connessioni attive (accettate). Se per quel tipo di connessione esiste la regola --state RELATED, ESTABLISHED -j ACCEPT, allora ogni connessione relativa alla prima viene accettata, senza essere valutata dalle regole che sono state applicate. In questo modo c'è un utilizzo minore di CPU.

Riassumendo le operazioni preliminari sono quindi sostanzialmente l'abilitazione del forwarding con il comando:

echo 1 >/proc/sys/net/ipv4/ip_forwarding

e l'inserimento dei moduli che servono per le nostre esigenze. Per quanto riguarda le prove che ho svolto, ho usato i seguenti moduli:

iptable_filter	*di base*
ip_tables	*di base*
iptable_nat	*per gestire il NAT/PAT*
ip_conntrack	*per il tracking delle connessioni*
ip_conntrack_ftp	*per il tracking delle connessioni ftp*
ip_nat_ftp	*per il NAT/PAT delle connessioni ftp*
ipt_mac	*per controllare il MAC ADDRESS nelle regole*
ipt_state	*per discriminare pacchetti in base allo stato della connessione*

2.3 Regole

L'insieme di restrizioni e di regole che compongono la politica di funzionamento del firewall, sono inserite in liste chiamate “catene” (**chains**), le quali vengono applicate ai pacchetti a seconda della provenienza o della destinazione. In ogni catena si esamina l'intestazione del pacchetto confrontandolo con le regole contenute e, nel caso se ne trovi una soddisfatta, la lettura all'interno della catena termina e si applica la tattica (**policy**) corrispondente. Se si arriva al fondo della catena viene applicata la tattica predefinita. Le

tattiche sono principalmente **DROP** (scarta) e **ACCEPT**, anche se se ne possono contare altre a seconda dei moduli che abbiamo caricato. Ad esempio, come vedremo ci sono policy per il NAT divise in SNAT e DNAT.

2.4 Comandi principali

Iptables supporta varie opzioni, questi sono i comandi principali:

-P : cambia la politica di una catena esistente:

```
iptables -P INPUT DROP
```

imposta la tattica base, per i pacchetti in ingresso diretti ai processi locali (esaminati quindi dalla catena INPUT), su DROP. Cioè se non si trovano regole che corrispondono al pacchetto in esame, questo viene scartato.

-A : appende una regola ad una catena

```
iptables -A INPUT -s 192.168.0.1 -j DROP
```

Aggiungi in coda alla catena input (-A INPUT) la regola per cui tutti i pacchetti con indirizzo sorgente 192.168.0.1 (-s 192.168.0.1) vengano scartati (-j DROP)

-D : cancella una regola da una catena

```
iptables -D OUTPUT 2
```

cancella la regole numero 2 dalla catena di output

```
iptables -D INPUT -s 192.168.0.1 -j DROP
```

cancella la regola che avevamo precedentemente impostato

-L : elenca le regole presenti in una catena (o in tutte le catene se non specificato)

```
iptables -L
```

elenca le regole delle catene principali (INPUT, OUTPUT, FORWARD)

iptables -L INPUT

elenca le regole della catena INPUT

iptables -t nat -L

elenca le regole presenti nei tre chain della tabella nat

-F : svuota le regole presenti in una catena

-p [protocollo] —-help : iptables ha un sistema di help che

può essere suddiviso in base ai protocolli. Ad esempio, se vogliamo sapere tutte le opzioni possibili su come filtrare il protocollo ICMP, è sufficiente scrivere:

iptables -p icmp —-help

Altri comandi sono:

-N : crea una nuova catena.

A volte è conveniente creare proprie catene in modo da rendere più semplice la gestione

-X : cancella una catena vuota

-I : inserisci una regola in una posizione specifica nella catena

-R : sostituisci un regola specifica nella catena con la nuova

2.5 Rendere le regole permanenti

E' bene ricordare che tutte le regole che si scrivono, non vengono memorizzate in modo permanente. Quindi al successivo riavvio è necessario reimpostarle. E' conveniente scriversi un proprio script

da eseguire in modo automatico all'avvio, e ricordarsi in questo script di azzerare sempre le regole precedentemente impostate in modo da ricominciare sempre da zero nel momento delle prove. Per esempio, per cancellare le regole dai chain principali (quelli del firewall) è sufficiente l'opzione -F, così per cancellare le regole di NAT basta specificare la tabella NAT (-t nat):

```
iptables -F

iptables -t nat -F
```

Un altro metodo per creare un script, è quello di impostare tutte le regole al nostro firewall e, una volta raggiunte le condizioni ottimali, eseguire il comando:

```
iptable_save > file_di_regole
```

Questo comando crea uno script, non shell ma comunque facilmente interpretabile, che può essere richiamato con il comando:

```
iptable_restore < file_di_regole
```

2.6 Debug

2.6.1 Script della shell

Poiché si è detto che le nostre regole saranno facilmente incapsulate in uno script shell, è necessario utilizzare i consueti metodi per controllare qualunque script. Iptables segnala sempre gli errori, il problema è capire a che punto dello script essi avvengono.

Io mi sono trovato bene a suddividere in sezioni il mio script, anche per pulizia di codice e di esecuzione, in cui ogni sezione incomincia con una stampa su video (**echo**) di che cosa si sta

facendo. Un altro ausilio piuttosto comodo è lanciare lo script con l'opzione -x, ad esempio:

```
bash -x /etc/firewall
```

Così facendo, bash stampa ogni riga dello script prima di eseguirla.

2.6.2 Analisi del comportamento: Packet Sniffing

Un'altra considerazione importante è che spesso non è facile capire cosa sta accadendo o quale regola da fastidio, quindi non sempre è sufficiente studiare i nostri script per capire cosa non va.

Uno strumento che per me è stato fondamentale è un analizzatore di rete o altrimenti detto “Sniffer”. Non c'è nulla di meglio che osservare i pacchetti che arrivano dai due lati del nostro router e vedere quali non passano, quali invece vengono instradati e con quali modifiche.

A questo scopo consiglio caldamente il software Ethereal di Gerald Combs, liberamente scaricabile dal sito www.ethereal.com.

3 Funzioni di iptables: NAT

3.1 Un po' di chiarimenti su NAT/PAT

Iptables offre diverse funzionalità, infatti oltre a controllare le regole per il firewall, può svolgere una funzione importantissima, quella di NAT/PAT.

Chiariamo subito che, quelle che vengono normalmente accorpate sotto la generalizzazione di NAT sono in realtà tre funzioni distinte: il NAT, il PAT e la combinazione delle due (NAT/PAT) e solitamente si fa riferimento al “NAT” accorpandole tutte, quindi ho voluto chiarire un po' le loro differenze.

3.1.1 NAT (Network Address Translation)

Il termine significa letteralmente “traduzione di indirizzi di rete” e la sua funzione la possiamo intuire con un esempio: supponiamo, come nella Figura 1, di avere tre host da proiettare in internet tramite il nostro Gateway configurato come NAT con due schede di rete di cui una si affaccia su internet e una sulla nostra rete privata (con indirizzo **192.168.0.1**).

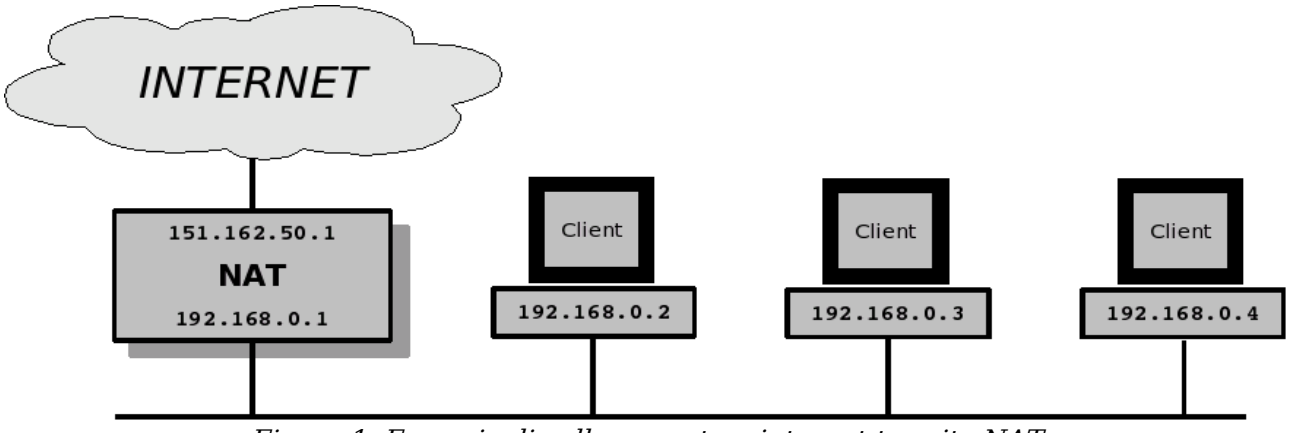

Figura 1: Esempio di collegamento a internet tramite NAT

Gli indirizzi degli host sono **192.168.0.2, 192.168.0.3** e **192.168.0.4**; tutti chiaramente indirizzi privati non adatti ad andare su internet.

La soluzione più semplice è avere un NAT che faccia da Gateway per la rete interna e che cambi gli indirizzi dei vari host in indirizzi pubblici (adatti ad internet). Ad esempio quando riceve un pacchetto da **192.168.0.2** diretto verso un host su internet (es. **google.it**), prima di eseguire il forward, cambia l'indirizzo IP sorgente in **151.162.50.2** (indirizzo IP pubblico) come si vede nella Figura 2.

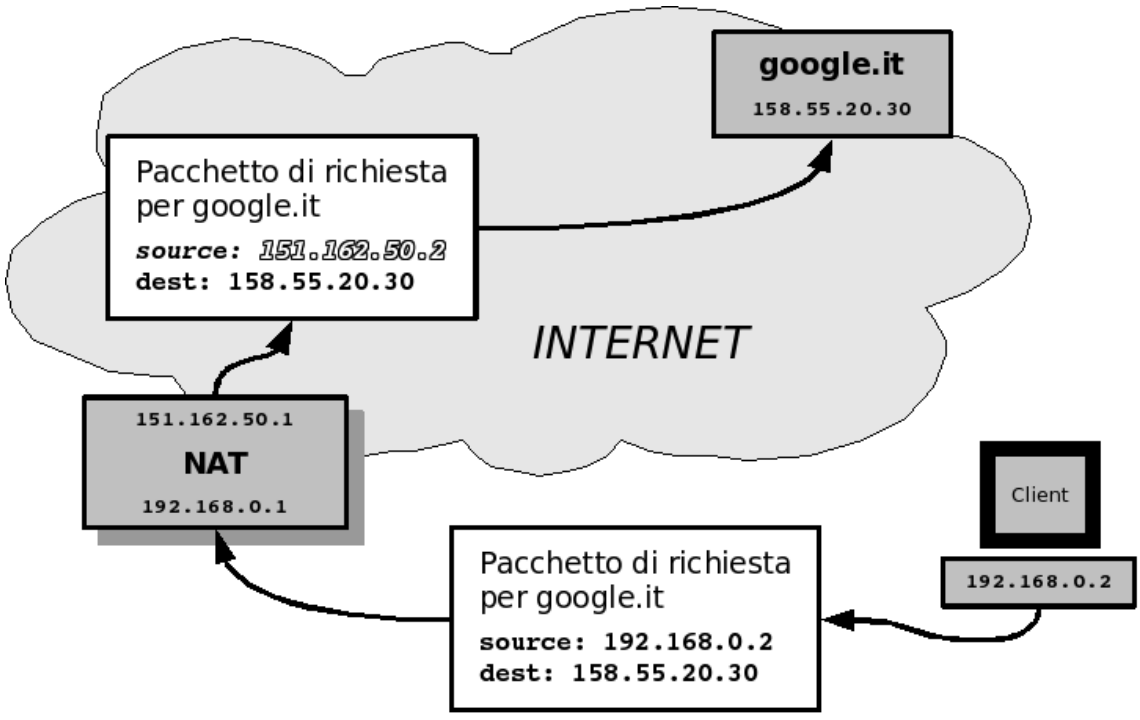

Figura 2: NAT - Modifiche dell'intestazione del pacchetto in uscita (RICHIESTA)

Generalizzando per ogni host, il lavoro che farà il NAT è la conversione secondo questa tabella:

$$192.168.0.2 \rightarrow 151.162.50.2$$

$$192.168.0.3 \rightarrow 151.162.50.3$$

$$192.168.0.4 \rightarrow 151.162.50.4$$

Allo stesso modo quando riceve dei pacchetti diretti ai tre indirizzi pubblici degli host, prima di rimandarli alla rete interna, ne cambia l'indirizzo IP di destinazione nell'adeguato indirizzo della rete interna:

$$151.162.50.2 \rightarrow 192.168.0.2$$

$$151.162.50.3 \rightarrow 192.168.0.3$$

$$151.162.50.4 \rightarrow 192.168.0.4$$

Per continuare il nostro esempio, la risposta dall'host google.it arriverà al nostro NAT con l'indirizzo di destinazione **151.162.50.2** che lui cambierà in **192.168.0.2** e proseguirà con il forward verso l'host della rete interna.

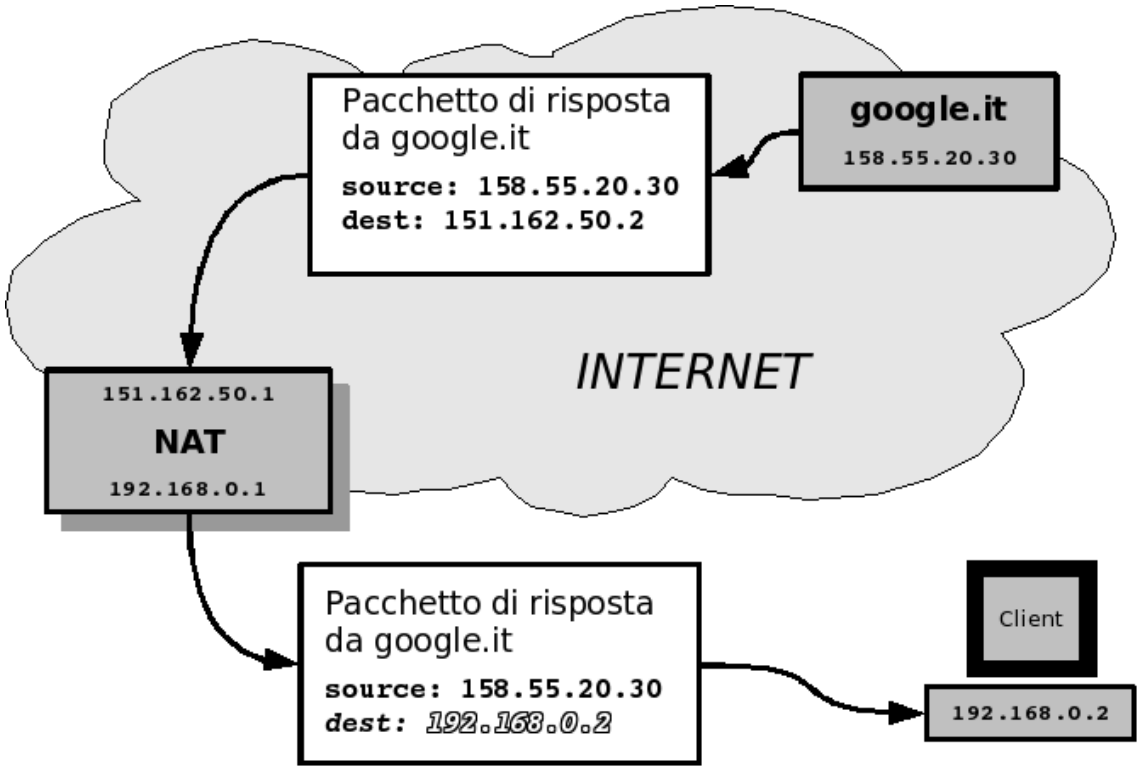

Figura 3: NAT - Modifiche dell'intestazione del pacchetto in ingresso (RISPOSTA)

E' chiaro che per mettere in pratica questa soluzione è necessario che il nostro ISP fornisca un numero di indirizzi IP pari al numero di computer nella nostra rete, il che non è conveniente.

3.1.2 PAT (Port Address Translator)

Di solito abbiamo a disposizione un solo indirizzo IP pubblico, spesso assegnato dinamicamente al nostro Router. Vogliamo quindi che i computer all'interno della nostra rete possano uscire su internet utilizzando questo indirizzo IP. Per fare ciò abbiamo bisogno che il Router mascheri i pacchetti uscenti come se provenissero da lui stesso e nello stesso tempo che sappia distinguere a chi inoltrare le risposte; usiamo il PAT, cioè “Traduttore di Porte di indirizzi”.

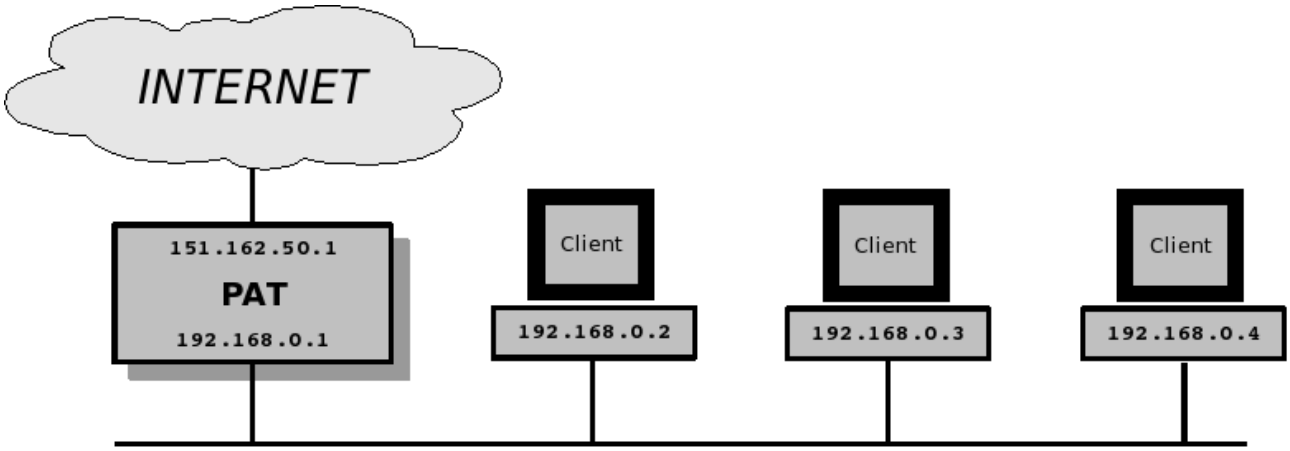

Figura 4: Esempio di collegamento a internet tramite PAT

Osservando la rappresentazione dell'infrastruttura (Figura 4) notiamo che è assolutamente analoga alla precedente. Ci sono N computer che devono riuscire a mandare pacchetti su internet ma ora abbiamo a disposizione un solo indirizzo IP pubblico, cioè quello sull'interfaccia esterna del nostro Gateway PAT (**151.162.50.1**). Ovviamente per eseguire il forward di pacchetti su internet, il nostro Router, procederà ad una sostituzione dell'indirizzo sorgente privato con il proprio indirizzo pubblico, ma in più dovrà essere in grado di riconoscere a chi sono destinate le risposte, (poiché tutte avranno il medesimo indirizzo di destinazione). Per raggiungere tale obiettivo, è necessario modificare anche le porte sorgenti.

Ad esempio, se il client **192.168.0.2** vuole mandare un

pacchetto di richiesta http all'host google.it, creerà un pacchetto con il proprio IP sorgente, l'indirizzo IP di destinazione di google.it e in più specificherà una propria porta sorgente (ad esempio la **2000**) e una porta destinazione (**80**, che corrisponde al servizio http).

Il Gateway PAT, prima di eseguire il forward, cambierà l'indirizzo sorgente nel proprio indirizzo IP pubblico (**151.162.50.1**) e cambierà la porta sorgente con una propria di cui terrà traccia (ad esempio la **4001**).

Ora supponiamo che anche l'host **192.168.0.3** faccia una richiesta http a **google.it**. Il PAT cambierà sempre l'indirizzo sorgente con il proprio IP pubblico, ma mapperà la porta sorgente scelta dal client (supponiamo sempre la **2000**) ad un'altra libera e ne terrà comunque traccia (supponiamo che scelga sia **4002**).

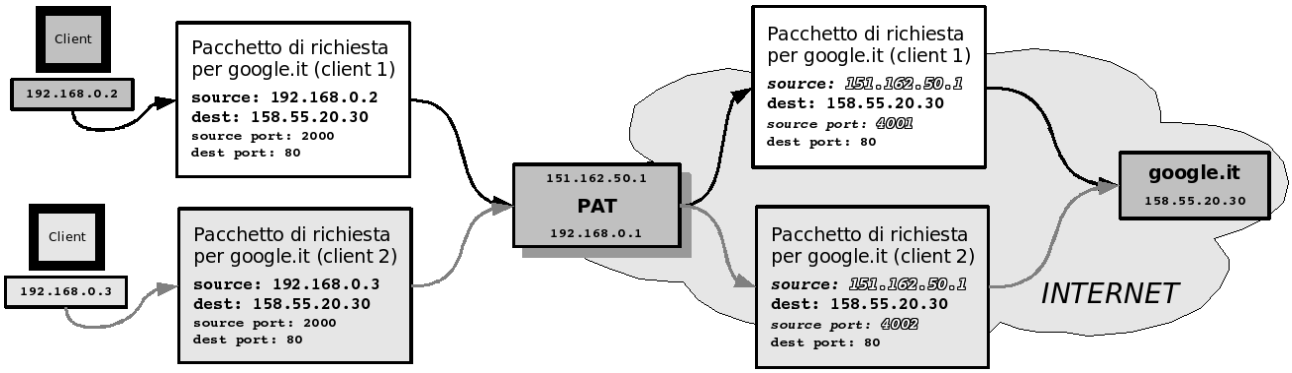

Figura 5: PAT - Modifiche delle intestazioni dei pacchetti in uscita (RICHIESTA)

Ora quando **google.it** risponderà alla prima richiesta userà come porta destinazione la porta sorgente ricevuta con il primo pacchetto di richiesta (**4001)** e risponderà alla seconda richiesta verso la porta **4002**. La differenza di porta destinazione dei pacchetti di risposta permettono al PAT di distinguere in modo univoco i due flussi di dati.

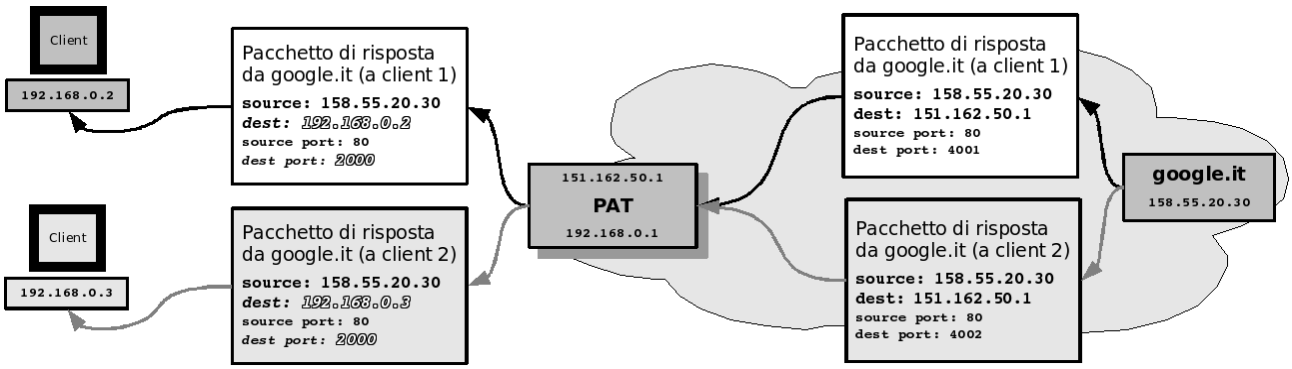

Figura 6: PAT - Modifiche delle intestazioni dei pacchetti in entrata (RISPOSTA)

3.1.3 NAT/PAT

Nella realtà, quasi sempre si incontrano casi in cui si utilizza un misto delle due tecniche. Ad esempio è facile che un'azienda acquisti dal proprio ISP un numero limitato di indirizzi IP pubblici ma che abbia un numero assai maggiore di computer da collegare su internet.

Il compito del Router, ipotizzando che ce ne sia uno solo, è quello di sfruttare gli N indirizzi pubblici con la tecnica NAT ma, terminati questi, continua a proiettare i pacchetti da e verso internet con la tecnica PAT nattando in modo appropriato per bilanciare il traffico tra gli IP pubblici disponibili.

Nel proseguire questo testo comunque, anch'io generalizzerò spesso il NAT/PAT con l'abbreviazione NAT.

3.2 Filosofia del NAT/PAT su iptables

Netfilter ha una tabella separata per gestire le regole per il NAT/PAT e, per modificare tali regole dovremo sempre farvi riferimento tramite l'opzione **`-t nat`**. Come possiamo osservare dalla Figura 7 ci sono tre chains: PREROUTING, POSTROUTING e OUTPUT.

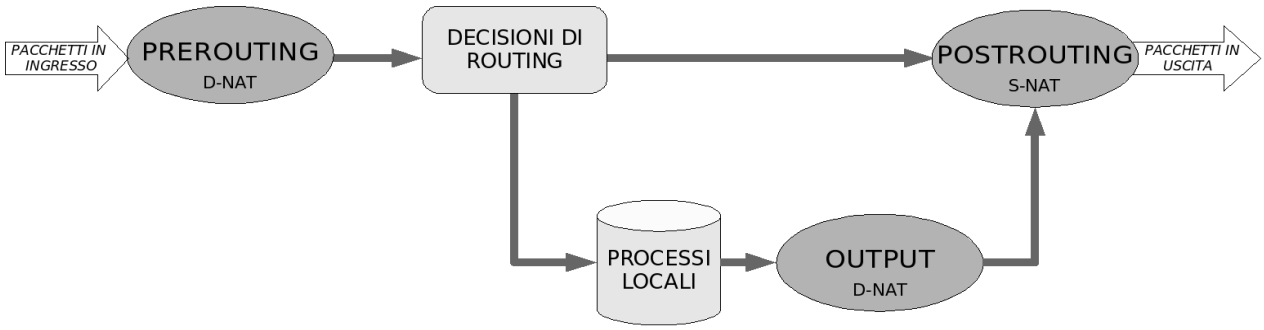

Figura 7: Schema chain del NAT

I pacchetti in ingresso vengono sottoposti alla catena PREROUTING appena vengono ricevuti, da qualunque interfaccia di rete provengano, anche prima di essere sottoposti alle regole di firewall ed alle decisioni di routing. Poiché è in questo chain che avviene il DNAT (Destination NAT), cioè la variazione dell'indirizzo destinazione del pacchetto ed eseguirla prima di tutti i successivi controlli fa sì che sia il firewall che le decisioni di routing vedano l'intestazione del pacchetto con l'indirizzo di destinazione reale.

I pacchetti in uscita passano, in ultima analisi, attraverso la catena POSTROUTING appena prima di essere inviati sulla rete (a qualunque scheda di rete), subito dopo le regole di firewall. Questo per il solito motivo di integrazione tra firewall e NAT, infatti è qui che avviene il cosiddetto SNAT (Source NAT), cioè la variazione dell'indirizzo sorgente del pacchetto, ed eseguirla per ultimo rende visibile, al firewall e al routing, l'indirizzo sorgente effettivo del

pacchetto.

In fine, i pacchetti generati localmente, in uscita, hanno un ulteriore catena prima di essere sottoposti al chain POSTROUTING, che si chiama semplicemente OUTPUT. Anche qui si può svolgere il DNAT.

3.3 NAT in pratica

3.3.1 NAT semplice

Per fare il NAT semplice, come nell'esempio descritto in precedenza (vedi 3.1.1, Figura 1) dobbiamo fare in modo che il router spedisca i pacchetti ricevuti dagli host interni verso l'esterno modificandone l'indirizzo sorgente; per l'host con IP 192.168.0.2 si farà:

```
iptables -t nat -A POSTROUTING -s 192.168.0.2 \
-j SNAT --to-source 151.162.50.2
```

Appendi nella catena POSTROUTING (-A POSTROUTING) della tabella nat (-t nat) la regola per cui tutti i pacchetti con IP sorgente 192.168.0.2 (-s 192.168.0.2) avranno come politica il source nat (-j SNAT) a cui si cambierà l'indirizzo sorgente con 151.162.50.2 (--to-source 151.162.50.2)

Come opzione in più si può specificare la scheda di rete di uscita e, supponendo che quella su internet sia eth1, si aggiungerà in coda

```
-o eth1
```

che sta ad indicare che si usa eth1 come interfaccia di output.

Questo comando è da ripetere anche per gli altri host della rete, con gli opportuni IP per soddisfare le scelte di NAT descritte nell'esempio introduttivo.

A questo punto i pacchetti escono, ma è necessario che anche

dall'esterno si possano raggiungere gli host all'interno della rete, quindi per ogni host serve la regola speculare:

```
iptables -t nat -A PREROUTING -d 151.162.50.2 \
-j DNAT --to 192.168.0.2 [-i eth1]
```

Appendi nella lista PREROUTING (-A PREROUTING) della tabella nat (-t nat) la regola per cui tutti i pacchetti con destinazione 151.162.50.2 (-d 151.162.50.2) [provenienti dall'interfaccia input eth1 (-i eth1)] abbiano come politica la sostituzione della destinazione (-j DNAT) con la nuova destinazione 192.168.0.2 (--to 192.168.0.2)

Anche in questo caso, la specifica dell'interfaccia di ingresso è opzionale, come è indicato dalle parentesi quadre.

3.3.2 PAT e MASQUERANDING

Per quanto riguarda il PAT, basta agire nella catena POSTROUTING per tutta la rete con un comando solo. Non è necessario anche agire al contrario in quanto netfilter terrà traccia delle connessioni richieste dall'interno verso l'esterno e gestirà correttamente i flussi di risposta. In più così un host su internet non può interpellare un host della rete interna direttamente ma può solo rispondergli e questo fornisce già una maggiore sicurezza. Il comando necessario è, in caso di IP pubblico statico (come nell'esempio del paragrafo 3.1.2 Figura 4):

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 \
-j SNAT --to-source 151.162.50.1 [-o eth1]
```

Appendi nella catena POSTROUTING (-A POSTROUTING) della tabella nat (-t nat) la regola per cui tutti i pacchetti provenienti dalla rete interna (-s 192.168.0.0/24) usino la politica del source nat (-j SNAT) e cambino l'indirizzo sorgente in 151.162.50.1 (--to-source 151.162.50.1) [verso la periferica di output eth1 (-o eth1)]

Spesso però l'indirizzo IP pubblico (nel nostro caso il 151.162.50.1) è assegnato dinamicamente, come accade con le connessioni dial-up, quindi cambia di connessione in connessione; in questo caso c'è una politica che si adegua a questa situazione che si chiama MASQUERADE. Quindi si userà, al posto del precedente comando:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 \
-j MASQUERADE [-o eth1]
```

Appendi alla catena POSTROUTING (-A POSTROUTING) della tabella NAT (-t nat) la regola per cui i pacchetti provenienti dalla rete interna (-s 192.168.0.0/24) [per l'interfaccia di output eth1 (-o eth1)] vengano "mascherati" (-j MASQUERADE)

Nota *che sarebbe possibile anche specificare singolarmente gli IP da instradare in internet (come fatto nel NAT, dove però eravamo obbligati) invece che specificare tutta la rete, ma questo sarebbe già un filtraggio che è un compito da adibire preferibilmente al firewall, sia per pulizia delle regole che per comodità.*

3.3.3 NAT/PAT

Per il caso del NAT/PAT, nel quale supponiamo di avere una rete più grossa da gestire con ad esempio tre indirizzi pubblici, la sintassi è molto simile al PAT, solo che in questo caso c'è più di una possibilità per il mapping degli indirizzi sorgenti. Se ad esempio abbiamo una serie di indirizzi pubblici contigui (es: 130.120.32.7, 130.120.32.8 e 130.120.32.9) possiamo usare la regola:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24
-j SNAT ––to 130.120.32.7–130.120.32.9 [-o eth1]
```

Se invece gli indirizzi fossero completamente diversi, come ad esempio: 130.120.32.7, 130.120.32.25 e 130.120.32.50, la regola

sarebbe:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 \
-j SNAT --to 130.120.32.7 --to 130.120.32.25 \
--to 130.120.32.50 [-o eth1]
```

Infine è possibile combinare le due cose, ad esempio se avessimo come IP pubblici 130.120.32.7, 130.120.32.25, 130.120.32.26 e 130.120.32.27 si userebbe la regola:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 \
-j SNAT --to 130.120.32.7 \
--to 130.120.32.25-130.120.32.27 [-o eth1]
```

Nota *che la selezione degli indirizzi IP con cui proiettare i computer in internet avviene automaticamente scegliendo l'indirizzo IP con il minor numero di collegamenti attivi: questo garantisce il bilanciamento del carico (load balancing).*

3.3.4 Redirezione

Un altro esempio di Destination NAT è la redirezione. Partiamo dalla situazione descritta per il PAT del paragrafo 3.1.2 Figura 4 e immaginiamo di avere un server http all'interno della rete il cui indirizzo IP è 192.168.0.55. Questo server è in attesa di richieste sulla porta standard, cioè l'80 e serve gli host di tutta la rete interna.

Se vogliamo che questo server fornisca il suo servizio anche su internet, la soluzione più semplice è che il nostro Gateway PAT, detentore dell'indirizzo IP pubblico 151.162.50.1, fornisca egli stesso il servizio. In parole povere, vogliamo che da internet, si possa mettere come indirizzo nel proprio browser

http://151.162.50.1. Per fare ciò è necessario attivare la REDIREZIONE, e fare in modo che il nostro Gateway riconosca tutte le richieste TCP a lui rivolte (sulla porta 80) e le inoltri all'interno della rete al nostro server http (192.168.0.55); il comando è:

**iptables -t nat -A PREROUTING -p tcp --dport 80 **
-i eth1 -j DNAT —-to 192.168.0.55:80

Appendi alla catena PREROUTING (-A PREROUTING) della tabella NAT (-t nat) la regola per cui tutto i pacchetti TCP (-p tcp), destinati alla porta 80 (--dport 80) e provenienti dall'interfaccia esterna (-i eth1) siano aggiornati cambiando l'indirizzo di destinazione (-j DNAT) in 192.168.0.55 porta 80 (--to 192.168.0.55:80)

E supponendo di avere un altro server http all'interno della rete con IP 192.168.0.58 (sempre in attesa sulla porta 80) da raggiungere da internet, questa volta sulla porta 10080 per distinguerlo dall'altro (lo interpelleremo quindi con l'indirizzo http://151.162.50.1:10080):

**iptables -t nat -A PREROUTING -p tcp **
--dport 10080 -i eth1 -j DNAT —-to 192.168.0.58:80

Appendi alla catena PREROUTING (-A PREROUTING) della tabella NAT (-t nat) la regola per cui tutto i pacchetti TCP (-p tcp), destinati alla porta 10080 (--dport 10080) e provenienti dall'interfaccia esterna (-i eth1) siano aggiornati cambiando l'indirizzo di destinazione (-j DNAT) in 192.168.0.58 porta 80 (--to 192.168.0.58:80)

4 Funzioni di iptables: firewall

4.1 Filosofia del firewall su iptables

Come osserviamo nella Figura 8, il firewall è strutturato in tre liste di regole, ovvero tre catene (chain), che sono INPUT, OUTPUT e FORWARD.

I pacchetti in ingresso diretti ai processi della macchina stessa, vengono sottoposti alla catena INPUT, quelli in uscita (sempre generati dalla macchina in esame) vengono sottoposti alla catena OUTPUT e, tutti gli altri pacchetti, cioè quelli solamente in transito sulla macchina che li instrada, vengono analizzati dalle regole della lista FORWARD.

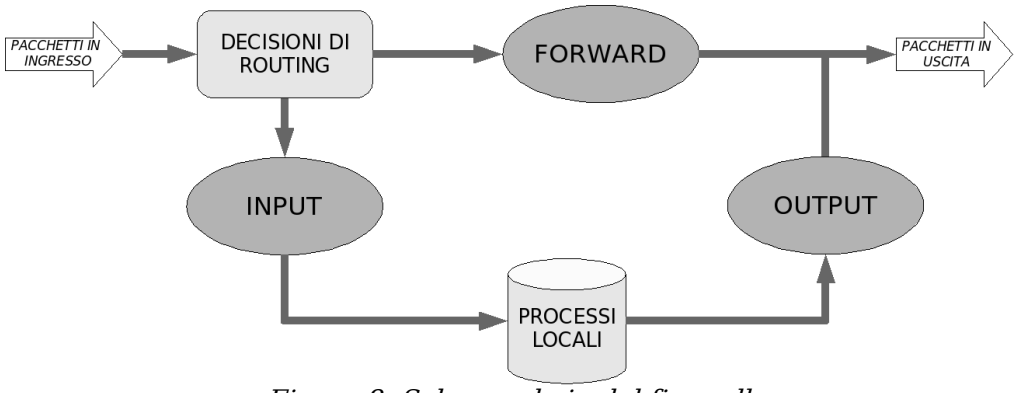

Figura 8: Schema chain del firewall

4.2 Integrazione di firewall e NAT

Come è già stato anticipato, il firewall e il NAT sono strutturati per funzionare perfettamente insieme e, per essere più specifici, è raro dovere modificare una regola del NAT a causa del firewall o viceversa. Infatti le catene del NAT sono disposte in modo che il firewall veda i pacchetti con le destinazioni reali, e quindi non

abbiamo nulla di cui preoccuparci. Per chiarezza comunque consiglio di dare un'occhiata alla Figura 9 che mostra uno schema riassuntivo delle catene del firewall e del NAT.

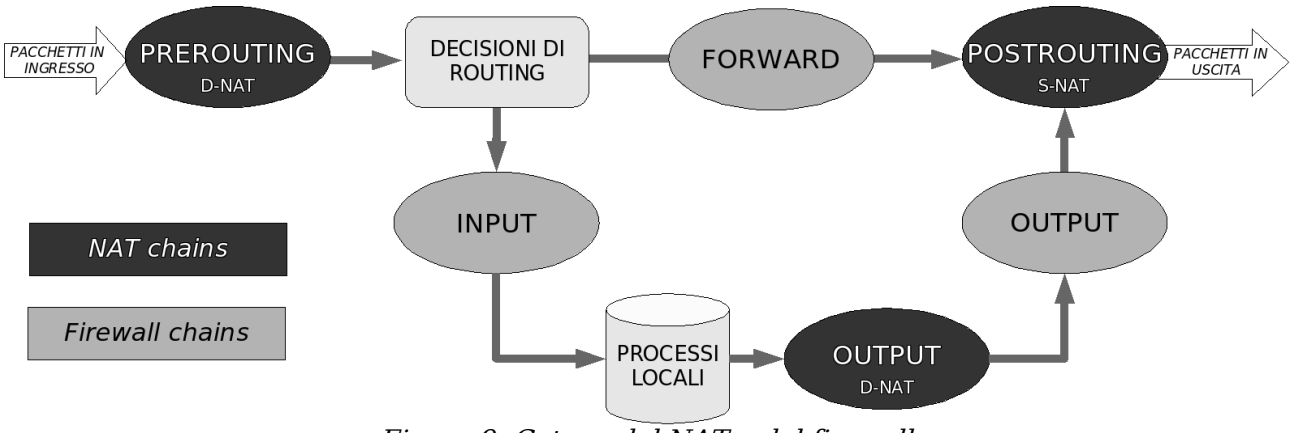

Figura 9: Catene del NAT e del firewall

Notiamo che ci sono 2 catene di OUTPUT, ma non c'è da fare confusione, la prima è quella del NAT (nello specifico Destination NAT) e la si raggiunge specificando la tabella nat (**`-t nat -A OUTPUT`**), la seconda è quella del firewall. Tutte le regole di filtraggio si inseriscono ovviamente in quella del firewall, senza specificare tabelle (es: **`iptables -P OUTPUT ACCEPT`**)

Se volete una spiegazione più completa su tutti i chains, consiglio il testo *Iptables Tutorial* di Oskar Andreasson che contiene uno schema molto approfondito.

4.3 Cosa e come filtrare

Purtroppo non ci sono regole fisse per sapere cosa filtrare, anche perchè più aumenta la sicurezza, più aumentano le restrizioni. La parola d'ordine è dunque “compromesso”; in ogni modo, gli approcci sono sostanzialmente due:

il primo è di lasciare passare tutto il traffico tranne quello che è esplicitamente pericoloso. Quindi si possono difendere le porte

sensibili riservate (le prime 1024), poi bloccare i pacchetti portatori di attacchi specifici (DoS, Ping of death, ecc);

il secondo approccio è quello di bloccare tutto il traffico tranne quello che è esplicitamente necessario.

Ovviamente il primo approccio non è certo il più sicuro, ma può essere versatile per una rete casalinga o non particolarmente esposta ad attacchi. Il secondo metodo è preferenziale, anzi fondamentale se è necessaria la massima sicurezza.

Per ognuno dei due sistemi, il mio consiglio è comunque di blindare almeno il firewall togliendogli ogni accesso su internet. In poche parole di scartare tutto il traffico in INPUT e in OUTPUT. Comunque, poiché come ho accennato questo testo non vuole essere una lezione di tattiche di sicurezza ma solo una guida per l'uso di iptables, farò diversi esempi di regole che possono essere spunti per crearsi le proprie.

4.4 Firewall in pratica

4.4.1 Infrastruttura in esame

Prima di tutto definiamo la rete a cui si farà riferimento in questo testo: la situazione è piuttosto comune, cioè una rete locale (LAN) che si affaccia a internet tramite un Router/Gateway al quale però, viene anteposta una macchina Linux che svolge il compito di Firewall che andremo, man mano, a configurare tramite **`iptables`** (vedi Figura 10).

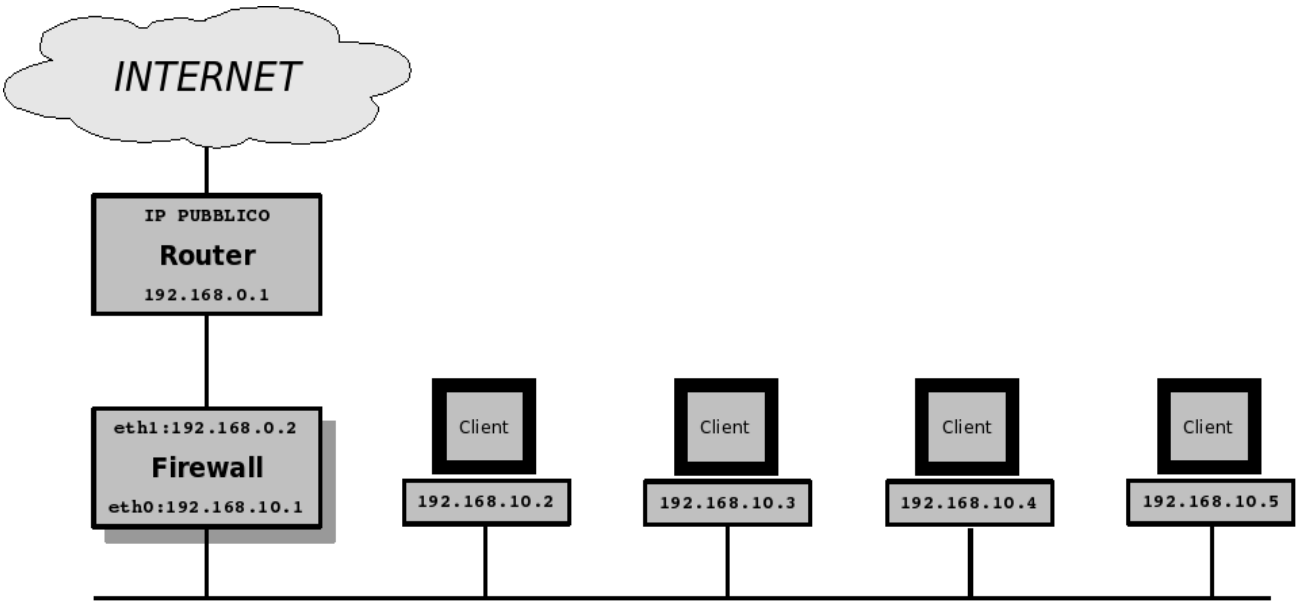

Figura 10: infrastruttura in esame

La rete interna è una classica rete di classe C **(192.168.10.x/255.255.255.0)** con quattro client collegati ad un Router tramite il nostro Firewall che si affaccia alla rete interna con l'IP **192.168.10.1** (interfaccia **eth0**) e con l'esterno con l'IP **192.168.0.2**. L'indirizzo pubblico del Router non ha importanza in quanto non ci interessa per il nostro lavoro. Ci basta sapere che il traffico che riceve verrà nattato in internet.

Inoltre, sono partito dal presupposto di non poter cambiare la tabella di routing del Router quindi il nostro Firewall farà anche da PAT in modo da avere maggior beneficio a livello di sicurezza.

Per chiarezza espongo la routing table del Firewall:

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.10.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG    1      0        0 eth1
```

Nota *che le spiegazioni che seguono si adattano perfettamente*

alla situazione abbastanza comune in cui il Router è un così detto "modem ethernet" (ADSL o di qualunque altro tipo).

4.4.2 Regole preliminari

Definiamo ora delle regole preliminari che farò a meno di ripetere nei due esempi di firewall che seguono ma che sono comuni a entrambi. Prima di tutto impostiamo la regola per fare in modo che il Firewall faccia da PAT (vedi 3.3):

```
iptables -t nat -A POSTROUTING -s 192.168.10.0/24\
 -j SNAT —-to-source 192.168.0.2 -o eth1
```

Nota *che le politiche di base per i chain della tabella nat devono essere impostate su ACCEPT. Questa è l'impostazione di default quindi non è necessario intervenire.*

Ora impostiamo le politiche di base che, per un buon firewall devono essere impostate tutte su DROP:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

In più, poiché normalmente i firewall sono macchine dedicate e generalmente prive di monitor e tastiera, vediamo di costruire una regola perchè possa essere raggiunta tramite una connessione ssh dal computer dell'amministratore (che supponiamo sia l'ultimo **192.168.10.5**).

A questo proposito specifichiamo un po' meglio il funzionamento del concetto di "stato di connessione" che serve a distinguere le connessioni (in base ai flag dei pacchetti). Infatti una connessione tra due host, è caratterizzata da diversi pacchetti che vengono

suddivisi sostanzialmente in due tipi diversi: un pacchetto iniziale che inizia la connessione, definito di tipo **NEW** e tutti gli altri che si riferiscono a quella connessione definiti **ESTABLISHED**. Se non vogliamo che la nostra macchina accetti connessioni in ingresso è sufficiente scartare i pacchetti di tipo NEW in ingresso. Esistono tuttavia un altro tipo di pacchetti, cioè quelli relativi a connessioni esistenti che sono definiti di tipo **RELATED**. Per capire quali sono basta pensare al protocollo FTP che utilizza una porta principale ed una seconda porta di supporto.

Poiché la nostra macchina scarta automaticamente tutti i pacchetti, basta accettare in INPUT i pacchetti di tipo NEW ed ESTABLISHED e permettere in OUTPUT i pacchetti di tipo ESTABLISHED:

```
iptables -A INPUT -p tcp -i eth0 -s 192.168.10.5 \
-d 192.168.10.1 --dport 22 -m state \
--state NEW,ESTABLISHED -j ACCEPT
```

Appendi alla lista INPUT (-A INPUT) la regola per cui i pacchetti TCP (-p tcp) provenienti dall'interfaccia eth0 (-i eth0), con indirizzo sorgente 192.168.10.5 (-s 192.168.10.5), con indirizzo destinazione 192.168.10.1 (-d 192.168.10.1), destinati alla porta 22 (--dport 22) e con stato della connessione NEW e ESTABLISHED (-m state –-state NEW,ESTABLISHED), vengano accettati (-j ACCEPT).

```
iptables -A OUTPUT -p tcp -o eth0 -s 192.168.10.1\
-d 192.168.10.5 --sport 22 -m state \
--state ESTABLISHED -j ACCEPT
```

Appendi alla lista OUTPUT (-A OUTPUT) la regola per cui i pacchetti TCP (-p tcp), in uscita verso l'interfaccia eth0 (-o eth0), con indirizzo sorgente 192.168.10.1 (-s 192.168.10.1), con indirizzo destinazione 192.168.10.5

(-d 192.168.10.5), con porta sorgente 22 (--sport 22) e con stato della connessione ESTABLISHED (-m state –--state ESTABLISHED) abbiano come politica ACCEPT (-j ACCEPT)

Una sicurezza in più è tenere traccia nei log di sistema quando avviene questa connessione. Per fare questo usiamo una nuova politica che è chiamata LOG. Ricordate che abbiamo detto che come un pacchetto soddisfa una certa regola le altre non sono più testate? Questo non avviene con i pacchetti con tattica LOG che proseguono nella catena. Se così non fosse il pacchetto verrebbe solo loggato ma non servirebbe alla connessione.

Quindi avverto che

LE REGOLE DI LOG DEVONO ESSERE MESSE PRIMA

della corrispettiva regola che filtra il pacchetto!

Nel nostro caso, inseriamo la nostra regola di log al primo posto nella catena input:

```
iptables -I INPUT 1 -p tcp -i eth0 \
-s 192.168.10.5  -d 192.168.10.1 --dport 22 \
-m state --state NEW \
-j LOG --log-prefix "Connessione SSH:"
```

Inserisci come regola 1 nella catena INPUT (-I INPUT 1) la regola per cui i pacchetti TCP (-p tcp), provenienti da eth0 (-i eth0), con sorgente 192.168.10.5 (-s 192.168.10.5), con destinazione 192.168.10.1 (-d 192.168.10.1), con porta destinazione 22 (--dport 22) e di tipo NEW (-m state –-state NEW) vengano salvati nei log di sistema (-j LOG) con l'intestazione “Connessione SSH:” (--log-prefix “Connessione SSH”)

Se pensiamo che un utente troppo zelante possa cambiare il proprio IP con quello della macchina dell'amministratore (ad esempio quando questa è spenta), possiamo restringere

ulteriormente i controlli da fare al nostro pacchetto in ingresso controllando l'indirizzo MAC. Supponendo che l'indirizzo MAC della scheda di rete del computer dell'amministratore sia 00:10:00:10:14:A0, è sufficiente modificare il comando in questo modo:

```
iptables -A INPUT -p tcp -i eth0 -s 192.168.10.5 \
-m mac —-mac-source 00:10:00:10:14:A0\
-d 192.168.10.1 --dport 22 -m state \
--state NEW,ESTABLISHED -j ACCEPT
```

Sintesi della configurazione:

1. Abbiamo abilitato il NAT
2. Abbiamo impostato su DROP le tattiche principali
3. Abbiamo consentito il collegamento amministrativo ssh

4.4.3 Primo esempio di Firewall

Ora che il nostro Firewall è protetto, possiamo creare qualche regola per fare navigare i computer della rete interna. Questo primo approccio vuole essere indirizzato ad un firewall con pochissime limitazioni in modo da essere molto versatile; ovviamente questo non può fornire il massimo della sicurezza ma è ad un livello ragionevole se non si hanno particolari esigenze.

Nota bene che il nostro firewall ora non può fare ancora nulla, solo essere raggiunto dal computer di amministrazione in ssh, infatti nonostante il PAT sia abilitato, la catena FORWARD non ha regole e quindi vale la politica di default che abbiamo impostato su

DROP. Le mie intenzioni sono fare andare i computer della rete interna su internet, senza lasciare che il Firewall ne abbia la possibilità, ecco perché agiremo solo nella catena FORWARD che, come abbiamo visto, si riferisce al traffico da instradare a host diversi da quello in esame.

Un tipo di regola che toglie già un po' di problemi, è quella di bloccare ogni tentativo di connessione verso i computer interni e di permetterli invece in uscita: useremo l'ausilio dell'estensione state permettendo il forwarding di pacchetti riferiti a connessioni nuove (NEW), stabilite (ESTABLISHED) o relative ad altre (RELATED) generati dai nostri host, mentre dall'esterno permetteremo il forwarding solo di pacchetti di connessioni stabilite e loro relativi (ESTABLISHED e RELATED).

Quindi, per ogni computer che decidiamo di mandare su internet useremo queste due regole:

```
iptables -A FORWARD -s 192.168.10.3 -i eth0 \
-m state ––state NEW,RELATED,ESTABLISHED -j ACCEPT
```

Appendi al chain FORWARD (-A FORWARD) la regola per cui ogni pacchetto il cui indirizzo sorgente è 192.168.10.3 (-s 192.168.10.3), proveniente dalla rete interna (-i eth0) e con stato della connessione uguale a NEW, ESTABLISHED o RELATED (-m state –-state NEW,RELATED,ESTABLISHED) vengano accettati (-j ACCEPT).

```
iptables -A FORWARD -d 192.168.10.3 -i eth1 \
-m state ––state RELATED,ESTABLISHED -j ACCEPT
```

Appendi al chain FORWARD (-A FORWARD) la regola per cui ogni pacchetto il cui indirizzo destinazione è 192.168.10.3 (-s 192.168.10.3), proveniente dalla rete esterna (-i eth1) e con stato della connessione uguale a ESTABLISHED o RELATED (-m state –-state RELATED,ESTABLISHED) sia accettato (-j ACCEPT).

Conviene mettere tutto in un ciclo for per abilitare tutti gli host da autorizzare:

```
AUTORIZ="192.168.10.2 192.168.10.3 192.168.10.4"
for i in $AUTORIZ
do
    iptables -A FORWARD -s $i -i eth0 -m state \
      —-state NEW,RELATED,ESTABLISHED -j ACCEPT
    iptables -A FORWARD -d $i -i eth1 -m state \
      —-state RELATED,ESTABLISHED -j ACCEPT
done
```

Ovviamente se volessimo autorizzare tutta la rete in un colpo solo, basta mettere come source e destination tutta la rete, cioè:

```
iptables -A FORWARD -s 192.168.10.0/24 -i eth0 \
-m state —-state NEW,RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.10.0/24 -i eth1 \
-m state —-state RELATED,ESTABLISHED -j ACCEPT
```

Come regole potremmo anche fermarci qui, anche perchè dall'esterno siamo protetti da nuove connessioni, ping e da loro relative vulnerabilità; dall'interno il nostro firewall non può comunicare che con il computer dell'amministratore quindi siamo abbastanza tranquilli.

Se proprio vogliamo proteggere un po' di più la rete interna, con conseguente discriminazione di alcune applicazioni, possiamo

bloccare il traffico UDP che è portatore delle maggiori vulnerabilità ed è difficilmente controllabile. Dobbiamo però tenere presente che il servizio DNS spesso è UDP (porta 53), quindi se i nameserver sono sull'extranet, potrebbe essere saggio bloccare tutti i pacchetti DNS al di fuori di quelli destinati e provenienti dal nostro server DNS.

Quindi innanzi tutto blocchiamo il traffico DNS; ricordo che è importante l'ordine perchè se questa regola fosse messa in coda non verrebbe mai raggiunta, quindi sarebbe come se non ci fosse. La mettiamo quindi in testa:

```
iptables -I FORWARD 1 -p udp -j DROP
```

Inserisci nella posizione 1 della tabella FORWARD (-I FORWARD 1) la regola per cui tutti i pacchetti UDP (-p udp) vengano scartati (-j DROP)

E subito dopo autorizziamo gli altri IP a colloquiare con il DNS il cui IP supponiamo sia 192.168.0.11, sempre preparando un ciclo (anche in questo caso dobbiamo fare attenzione che le regole siano prima di quella precedente, quindi le mettiamo all'inizio):

```
for i in $AUTORIZ
do
  iptables -I FORWARD 1 -p udp -s $i \
  -d 192.168.0.11 --dport 53 -j ACCEPT
```

Inserisci in testa alla catena FORWARD (-I FORWARD 1) la regola per cui tutti i pacchetti UDP (-p udp), provenienti dall'IP attuale (-s $i), destinati al server DNS 192.168.0.11 (-d 192.168.0.11), porta 53 (---dport 53) siano accettati (-j ACCEPT)

```
iptables -I FORWARD 1 -p udp -s 192.168.0.11 \
-d $i --sport 53 -j ACCEPT
```

Inserisci in testa alla catena FORWARD (-I FORWARD 1) la regola per cui tutti i pacchetti UDP (-p udp), provenienti dal server DNS 192.168.0.11

(-s 192.168.0.11), destinati all'IP attuale (-d $i), porta sorgente 53 (--sport 53) siano accettati (-j ACCEPT)

done

Per una listato completo d'esempio riamando all'appendice A.1

4.4.4 Secondo esempio di Firewall

Ora ci apprestiamo a configurare un firewall che permetta effettivamente solo alcuni servizi e quindi maggiormente sicuro ma anche più lungo da gestire.

La rete in esame è sempre quella di Figura 10 e consideriamo sempre le regole preliminari del paragrafo 4.4.2. In più per semplicità prepariamo le regole solo per un host (**192.168.10.4**), poiché a questo punto spero che il meccanismo per autorizzare tutti in blocco o la preparazione di semplici cicli in shell sia chiaro. Focalizziamoci quindi nei diversi tipi di traffico.

Prima di tutto permettiamo il dialogo con il server DNS, ipotizzando che sia all'esterno della rete, ricordando che il traffico può essere sia UDP che TCP:

```
iptables -A FORWARD -p udp -s 192.168.10.4 \
-d <IP server DNS> --dport 53 -j ACCEPT

iptables -A FORWARD -p tcp -s 192.168.10.4 \
-d <IP server DNS> --dport 53 -j ACCEPT

iptables -A FORWARD -p udp -s <IP server DNS> \
-d 192.168.10.4 --sport 53 -j ACCEPT

iptables -A FORWARD -p tcp -s <IP server DNS> \
-d 192.168.10.4 --sport 53 -j ACCEPT
```

Se è possibile è meglio che i server DNS siano all'interno della

rete o che sia il firewall stesso, infatti un tipo di attacco è basato proprio sul mandare risposte DNS mendaci per fare collegare i propri host ad un server malintenzionato.

Ora abilitiamo tutti i pacchetti in uscita e in risposta relativi al servizio HTTP (porta 80 per il server):

```
iptables -A FORWARD -s 192.168.10.4 -i eth0 \
-p tcp --dport 80 -m state \
--state NEW,RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.10.4 -o eth0 \
-p tcp --sport 80 -m state \
--state RELATED,ESTABLISHED -j ACCEPT
```

Come si vede, il primo comando abilita tutti i pacchetti con sorgente l'host da autorizzare ma che siano strettamente da inviare alla porta 80 di qualunque server, e il secondo abilita le risposte che devono avere porta sorgente 80 e non devono essere relativi a nuove connessioni.

Ora possiamo continuare con la stessa tecnica per abilitare altri servizi, ad esempio le transazioni HTTPS (porta 443):

```
iptables -A FORWARD -s 192.168.10.4 -i eth0 \
-p tcp --dport 443 -m state \
--state NEW,RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.10.4 -o eth0 \
-p tcp --sport 443 -m state \
--state RELATED,ESTABLISHED -j ACCEPT
```

Consideriamo l'ipotesi che sia necessario anche guardare la posta tramite il protocollo IMAP, apriamo la porta corrispondente (143):

```
iptables -A FORWARD -s 192.168.10.4 -i eth0 \
-p tcp --dport 443 -m state \
--state NEW,RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -d 192.168.10.4 -o eth0 \
-p tcp --sport 443 -m state \
--state RELATED,ESTABLISHED -j ACCEPT
```

Un altra cosa che possiamo abilitare, è la possibilità di eseguire il comando ping bloccando però di essere pingati, quindi permettiamo i pacchetti echo request (8) verso l'esterno e solo quelli di risposta verso di noi (0 echo reply):

```
iptables -A FORWARD -i eth0 -s 192.168.10.4 \
-p icmp -m icmp ––icmp-type 8 -j ACCEPT

iptables -A FORWARD -i eth1 -d 192.168.10.4 \
-p icmp -m icmp ––icmp-type 0 -j ACCEPT
```

Il sistema visto è molto comodo quando si vuole filtrare molto permettendo solo pochi servizi fondamentali. In più, con un accurato script, possiamo facilmente decidere i servizi che può usare ognuno dei computer della rete (vedi appendice A.2)

4.4.5 Comandi per evitare attacchi specifici

Ora prendiamo in esame alcune regole che possono essere utili in caso sia necessario permettere del traffico notoriamente soggetto a vulnerabilità. L'obiettivo è fermare o limitare i pacchetti più critici. Tenete conto che le regole devono essere messe nella catena opportuna e nell'ordine opportuno per funzionare, quindi se voglio proteggere il computer in esame saranno messe nella catena INPUT e se devo difendere la rete interna vanno messe nella

catena FORWARD. Solitamente se l'obiettivo è scartare (DROP) di solito vanno messe all'inizio del chain.

4.4.5.1 Estensione limit

Un modulo molto utile per prevenire certi tipi di attacchi è **limit**, che serve per aggiungere una restrizione temporale. Facciamo un esempio: vogliamo loggare tutti i tentativi di ping verso la nostra macchina, ma questo sarebbe già di per sé una vulnerabilità poiché si può saturare la memoria di massa di LOG. Allora possiamo limitare il LOG di questo tipo di pacchetti a due volte al minuto:

```
iptables -A INPUT -p icmp -m icmp —-icmp-type 8\
 -m limit --limit  2/minute —-limit-burst 6 \
-j LOG
```

Aggiungi in coda alla catena INPUT la regola per cui tutti i pacchetti ICMP Echo Request (-m icmp –-icmp-type 8) vengano loggati (-j LOG) con una frequenza massima di 2 al minuto (-m limit –-limit 2/minute) con eccezione dei primi 6 (--limit-burst 6) che possono sforare.

In questo modo, vengono loggati, la prima volta, 6 pacchetti oltre i quali, se non è trascorso un minuto, se ne loggano solo 2 al minuto. Si può aggiungere limiti basati su secondi (/second o /s), minuti, ore o giorni. Il burst (o raffica) standard è di 5 e di solito non è necessario cambiarlo.

Integriamo il comando sul LOG dell'esempio del primo firewall limitando la connessione ssh a 1 volta al minuto:

```
iptables -I INPUT 1 -p tcp -i eth0 \
-s 192.168.10.5  -d 192.168.10.1 --dport 22 \
-m state --state NEW \
```

```
-m limit —-limit 1/minute \
-j LOG --log-prefix "Connessione SSH:"
```

4.4.5.2 Protezione dal Syn-Flood

Per proteggersi da questo tipo di attacco (che satura la memoria della vittima con un flusso enorme di richieste di connessione), basta limitare temporalmente tutte le connessioni in ingresso che si accettano. Per evitare rischi è sufficiente limitare a 1 al secondo. Esplicitamente:

```
iptables -I INPUT -p tcp —-syn -m limit \
—-limit 1/s -j ACCEPT
```

Aggiungi la regola all'inizio della catena INPUT (-I INPUT) per cui sia accettino (-j ACCEPT) connessioni TCP (-p tcp) con il SYN flag settato (--syn) una volta al secondo (-m limit -–limit 1/s)

A titolo d'esempio, ci difendiamo da un possibile syn-flooding nella connessione ssh permessa nel primo firewall:

```
iptables -I INPUT 1 -p tcp -i eth0 \
-s 192.168.10.5  -d 192.168.10.1 --dport 22 \
-m state --state NEW -m limit —-limit 1/s \
-j ACCEPT
```

4.4.5.3 Protezione da Port Scanner

In modo analogo al Syn-flooding, ci difendiamo dai pacchetti con il flag RST settato limitandoli ad 1 al secondo:

```
iptables -A INPUT -p tcp \
—-tcp-flags SYN,ACK,FIN,RST RST \
-m limit —-limit 1/s -j ACCEPT
```

Inserisci in coda alla catena INPUT (-A INPUT) una regola per cui tutti i

pacchetti TCP (-p tcp) che, controllati i flag SYN, ACK, FIN e RST, abbiano RST impostato (–-tcp-flags SYN,ACK,FIN,RST RST), siano accettati (-j ACCEPT) con un limite di 1 al secondo (-m limit –-limit 1/s).

4.4.5.4 Protezione dai Ping of death

Anche i pacchetti echo request del ping sono sfruttati per attacchi basati sul flooding, quindi li limitiamo con il solito comando limit:

```
iptables -A INPUT -p icmp -m icmp —-icmp-type 8\
 -m limit --limit  1/s -j ACCEPT
```

4.4.5.5 Altri filtri contro i DoS

E' buona norma filtrare i pacchetti multicast usando il modulo d'estensione **pkttype**:

```
iptables -A INPUT -m pkttype —pkt-type multicast\
 -j DROP
```

Come anche filtrare i pacchetti che non hanno superato il sanity check con l'estensione **unclean**:

```
iptables -A INPUT -m unclean -j DROP
```

Infine c'è un tipo di attacco DoS basato sul mandare pacchetti echo request di lunghezza esagerata che possono mandare in crisi l'host. Ecco un possibile filtro:

```
iptables -I INPUT -p icmp -m icmp —-icmp-type 8\
 -m length —-length 128:65535 -j DROP
```

A.1 Listato primo Firewall

```
#IP autorizzati ad uscire su internet
AUTORIZ=”192.168.10.2 192.168.10.3 192.168.10.4”

#puliamo i chain dalle regole
iptables -F
iptables -t nat -F

#impostiamo policy di base
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#prepariamo il nat
iptables -t nat -A POSTROUTING -s 192.168.10.0/24\
 -j SNAT —-to-source 192.168.0.2 -o eth1

#connessione ssh: LOG nuove connesioni
iptables -A INPUT -p tcp -i eth0 \
 -s 192.168.10.5  -d 192.168.10.1 --dport 22 \
 -m state --state NEW \
 -j LOG --log-prefix "Connessione SSH:"

#connessione ssh
iptables -A INPUT -p tcp -i eth0 -s 192.168.10.5 \
 -d 192.168.10.1 --dport 22 -m state \
 --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp -o eth0 -s 192.168.10.1\
 -d 192.168.10.5 --sport 22 -m state \
 --state ESTABLISHED -j ACCEPT
```

```
#traffico UDP permesso
for i in $AUTORIZ
do
  iptables -A FORWARD -p udp -s $i \
   -d 192.168.0.11 --dport 53 -j ACCEPT

  iptables -A FORWARD -p udp -s 192.168.0.11 \
   -d $i --sport 53 -j ACCEPT
done

#blocchiamo tutto il rimanente traffico UDP
iptables -A FORWARD -p udp -j DROP

#regole per eseguire connessioni e ricevere
#risposte
for i in $AUTORIZ
do
    iptables -A FORWARD -s $i -i eth0 -m state \
      —-state NEW,RELATED,ESTABLISHED -j ACCEPT
    iptables -A FORWARD -d $i -i eth1 -m state \
      —-state RELATED,ESTABLISHED -j ACCEPT
done
```

A.2 Listato secondo Firewall

In questo esempio si suppone che, della nostra rete, solo 192.168.10.2 e .3 possano collegarsi a tutti i servizi offerti, mentre 192.168.10.4 può solo vedere pagine HTTP. In più solo 192.168.10.5 può eseguire ping.

```
#IP autorizzati a raggiungere il server DNS

DNS="192.168.10.2 192.168.10.3 192.168.10.4"

#IP autorizzati a collegarsi a servizi HTTP
HTTP="192.168.10.2 192.168.10.3 192.168.10.4"

#IP autorizzati a collegarsi a servizi HTTPS
HTTPS="192.168.10.2 192.168.10.3"

#IP autorizzati a collegarsi a server di posta
#IMAP
IMAP="192.168.10.2 192.168.10.3"

#IP autorizzati a PINGARE l'extranet
PING="192.168.10.5"


#puliamo i chain dalle regole
iptables -F
iptables -t nat -F


#impostiamo policy di base
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

```
#prepariamo il nat
iptables -t nat -A POSTROUTING -s 192.168.10.0/24\
 -j SNAT —-to-source 192.168.0.2 -o eth1

#DNS
for i in $DNS
do
  iptables -A FORWARD -p udp -s $i \
   -d <IP server DNS> --dport 53 -j ACCEPT

  iptables -A FORWARD -p udp -s <IP server DNS> \
   -d $i --sport 53 -j ACCEPT

  iptables -A FORWARD -p tcp -s $i \
   -d <IP server DNS> --dport 53 -j ACCEPT

  iptables -A FORWARD -p tcp -s <IP server DNS> \
   -d $i --sport 53 -j ACCEPT
done
```

```
#HTTP
for i in $HTTP
do
  iptables -A FORWARD -s $i -i eth0 \
   -p tcp --dport 80 -m state \
   --state NEW,RELATED,ESTABLISHED -j ACCEPT

  iptables -A FORWARD -d $i -o eth0 \
   -p tcp --sport 80 -m state \
   --state RELATED,ESTABLISHED -j ACCEPT
done

#HTTPS
for i in $HTTPS
do
  iptables -A FORWARD -s $i -i eth0 \
   -p tcp --dport 443 -m state \
   --state NEW,RELATED,ESTABLISHED -j ACCEPT

  iptables -A FORWARD -d $i -o eth0 \
   -p tcp --sport 443 -m state \
   --state RELATED,ESTABLISHED -j ACCEPT
done
```

```
#IMAP
for i in $IMAP
do
  iptables -A FORWARD -s $i -i eth0 \
   -p tcp --dport 143 -m state \
   --state NEW,RELATED,ESTABLISHED -j ACCEPT

  iptables -A FORWARD -d $i -o eth0 \
   -p tcp --sport 143 -m state \
   --state RELATED,ESTABLISHED -j ACCEPT
done

#PING
for i in $PING
do
  iptables -A FORWARD -i eth0 -s $i -p icmp \
   -m icmp —-icmp-type 8 -j ACCEPT

  iptables -A FORWARD -i eth1 -d $i -p icmp \
   -m icmp —-icmp-type 0 -j ACCEPT
done
```

B. Glossario

Ho creato questo glossario focalizzando l'attenzione ai termini usati in netfilter, però per comodità ho inserito anche le sigle più comuni con la semplice forma estesa.

Catena: altrimenti detta chain, è un contenitore per le regole sull'analisi dei pacchetti. Ci sono sei catene principali. Quelle del firewall sono: INPUT, OUTPUT, FORWARD. Quelle del NAT sono: PREROUTING, OUTPUT, POSTROUTING

Chain: *vedi Catena*

DNS: *Domain Name Server*

DoS: *Denial of Service*

Extranet: rete esterna, cioè al di fuori del firewall

FTP: *File Transfer Protocol*

HTTP: *Hyper Text Transfer Protocol*

HTTPS: *Secure HTTP*

ICMP: *Internet Control Management Protocol*

IP: *Internet Protocol* (spesso si riferisce all'indirizzo IP)

ISP: *Internet Service Provider*, o più comunemente Provider. E' un ente che fornisce un collegamento a internet, e con esso indirizzi IP pubblici (dinamici o statici).

LAN: *Local Area Network*

MAC: *Media Access Control* Indirizzo fisico della scheda di rete

Nameserver: server DNS

NAT: *Network Address Translator* E' un dispositivo che fa da Gateway per una rete locale convertendo gli indirizzi IP degli host della rete privata in indirizzi pubblici in modo che questi host possano scambiare dati da/a internet con indirizzi pubblici, che devono essere comunque assegnati dall'ISP

NAT/PAT: *Network Address Translator / Port Address translator*

PAT: *Port Address Translator* E' un dispositivo che fa da Gateway ad una rete locale e permette agli host al suo interno di uscire su internet mascherandone l'indirizzo modificandone anche le porte sorgenti

Policy: Politica o target della regola. Ad esempio -J ACCEPT

Politica: *vedi policy*

SSH: *Secure Shell*

Target: Obiettivo della regola. *Vedi policy*

TCP: *Terminal Control Protocol*

UDP: *User Datagram Protocol*

C. Indice delle illustrazioni

D. Guide e fonti

Premetto che le fonti che ho usato sono diverse, comunque ho elencato i testi fondamentali che ho studiato e consiglio caldamente. Consiglio di diffidare da script trovati su internet da improvvisati saggi della sicurezza, poiché nella maggior parte dei casi erano script assurdi. E' ovvio che non basta che lo script giri per essere sicuri di un corretto funzionamento del firewall! Consiglio, quindi prima di usare uno di questi script in modo definitivo, di fare una bella scansione di porte in fase di test.

Il sito ufficiale di netfilter, dove si trovano, oltre alla documentazione, anche le INFORMAZIONI SULLE LICENZE è: http://www.netfilter.org

Testi specifici:

Linux 2.4 NAT HOWTO di Rusty Russell

Linux 2.4 Packet Filtering HOWTO di Rusty Russell

Iptables Tutorial di Oskar Andreasson

Appunti di Informatica Libera di Daniele Giacomini

Testi di supporto:

Linux Security HOWTO di Kevin Fenzi e Dave Wreski

Aggiornamenti di questo documento:

http://digilander.libero.it/lnx_sx *http://sandrolnx.altervista.org*

NOTA1: Netfilter / iptables è un software Open Source rilasciato sotto i termini di licenza GPL (http://www.gnu.org/copyleft/gpl.html). Per informazioni consultare la pagina ufficiale www.netfilter.org

NOTA2: Una vecchia versione di questo documento è disponibile come monografia (sempre dell'autore Sandro Cuciz) presso il Politecnico di Torino

GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed

under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to

text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in

quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other

section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the

various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright

holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.