Introduzione ai firewall con Linux

Mauro Barattin e **Oriano Chiaradia**

Belluno, 27 novembre 2004

Sommario

- Perché proteggersi
- Cosa proteggere con un firewall
- Approcci pratici alla sicurezza
- Definizione di firewall
- Breve introduzione tecnica
- Netfilter e Iptables
- Tabelle catene e regole
- Come si costruiscono le regole
- Filtraggio *steteless* e *statefull*
- Esempi applicativi
- Conclusioni

Perché è meglio proteggersi?

- Il nostro computer di casa, una volta connesso a Internet tramite un modem, è raggiungibile da ogni host del mondo! In mancanza di dispositivi di filtraggio, ogni servizio di rete attivo sul nostro computer (es. condivisioni windows...) può essere raggiungibile dall'esterno. Dal punto di vista della sicurezza, inoltre, ogni servizio esposto rappresenta una possibile vulnerabilità del nostro sistema e della nostra rete...
- L'offerta di connettività ADSL a basso costo ha permesso anche all'utenza casalinga di usufruire di linee "*always on*" ad alta velocità che, proprio perché sempre attive, necessitano di una maggiore protezione contro le intrusioni.

Cosa dobbiamo proteggere?

- **Dati**
  - Privacy
  - Integrità
  - Disponibilità
- **Risorse**
  - Integrità hardware e software
  - Tempo di calcolo/memoria
- **Reputazione**
  - Furto di identità (es. accesso a chiavi private)
  - Catena di attacchi (mascheramento provenienza originaria)
  - Presenza materiale indesiderato in archivi pubblici
  - Defacement

Tipologie di attacco

- **Intrusione**
  - Accesso non autorizzato
  - Scalata privilegi
  - Modifica dei sistemi (installazione di backdoors)
- **Denial of service**
  - Network flooding
  - Resource exhaustion
  - Disabilitazione/danneggiamento dei servizi
- **Furto di informazioni**
  - Accesso agli archivi
  - Monitoraggio del traffico di rete (sniffing)

Approcci alla sicurezza

- **Host level**
  - Sicurezza a livello di singola macchina
  - Ogni sistema è dotato di misure di protezione individuali
  - Non scalabile e di difficile gestione
  - Non adatto ad ambienti eterogenei per architetture e software
- **Network level**
  - Sicurezza a livello rete
  - Controllo di accesso concentrato (hardware/software dedicato)
  - Scalabile, gestibile, adattabile
  - Minore flessibilità

Cos'è un firewall?

Proviamo a dare qualche definizione...

- “Muro di fuoco”
- “Il firewall è un componente attivo che si interpone fra due o più reti (ad esempio una rete locale e una rete pubblica), configurato con un insieme di regole che determinano quali comunicazioni sono permesse e quali invece sono vietate”.
- “Il firewall è un componente hardware con due o più schede di rete sul quale viene installato un ambiente operativo che analizza e gestisce il traffico dei pacchetti in base alla configurazione fornita dall'amministratore di rete”.

...insomma, un firewall è un FILTRO!

Cosa fa e cosa non fa...

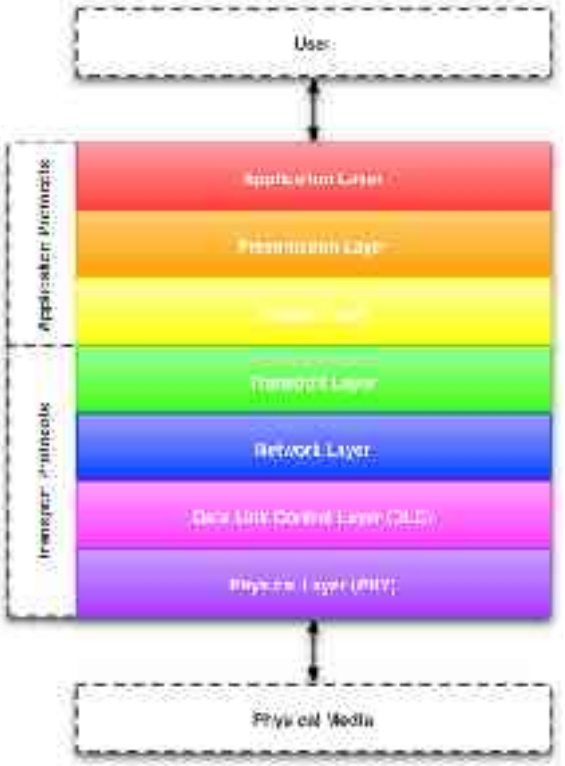
- **Cosa fa...**
  - Separazione fisica tra due o più reti
  - Restrizione sul traffico
  - Monitoraggio del traffico
  - Logging del traffico

- **Cosa non fa...**
  - Non protegge da virus
  - Non impedisce lo scaricamento di software infetto
  - Non esamina il *payload* dei pacchetti (i “dati”)
  - Non può fare ciò che non è stato previsto in partenza
  - Non protegge da tipi di attacchi non noti a priori
  - Non protegge da utenti legittimi
  - Non protegge in caso di punti di accesso non controllati

Packet filter vs. application proxy

I firewall si dividono in due categorie:

- **Packet filter**
  - Agisce a livello network/transport.
  - Limitate richieste hardware
- **Application proxy**
  - Agisce a livello application.
  - Applicazioni separate per ogni servizio
  - Richiede hardware più veloce
  - Poco adatti su reti ad alto traffico



Nel seguito ci occuperemo di packet filter...

Il protocollo TCP/IP

I protocolli sono gli standard che specificano:

- Come avvengono i trasferimenti da una macchina ad un'altra
- Come sono rappresentati i dati
- Quali sono tecniche per la rivelazione d'errore
- Il meccanismo di *acknowledgment* per i pacchetti trasmessi

Tali protocolli rendono invisibile all'utente l'hardware sottostante durante una qualsiasi sessione di lavoro.

Per TCP/IP, il protocollo su cui si basa Internet, non si intende solo il protocollo di trasmissione **TCP** ed il protocollo di rete IP, ma una famiglia di protocolli comprendente anche l'**UDP**, l'**ICMP**, l'**ARP**, ...

TCP

(Transmission Control Protocol)

Il protocollo TCP si assume la responsabilità di instaurare (SYN) un collegamento tra due host, di rendere affidabile il trasferimento di dati e comandi tra essi (richiedendo, se necessario, la ritrasmissione di alcuni dati) ed infine di chiudere la connessione (FIN).

- Connection-oriented
- Affidabile (controllo di integrità e sequenzialità)
- Alto overhead
- Necessita lo stabilirsi di una connessione
- Ogni servizio identificato da una porta (si noti che il TCP usa la connessione, e non la porta, come sua fondamentale astrazione...)
- Usato dai servizi **HTTP**, **FTP**, **SMTP** ed altri

UDP

(Transmission Control Protocol)

Il protocollo UDP fornisce un servizio di recapito dei datagrammi connectionless ed inaffidabile (non prevede nessun meccanismo per il controllo dell'errore). Perciò i messaggi UDP possono essere persi, duplicati oppure arrivare fuori dall'ordine; inoltre i pacchetti possono arrivare più velocemente di quanto il ricevente sia in grado di processarli.

- Non connection-oriented
- Nessun controllo di integrità
- Basso overhead
- Servizi identificati da porte
- Usato da **DNS**, **NFS**, **NetBIOS**

ICMP

(Internet Control Message Protocol)

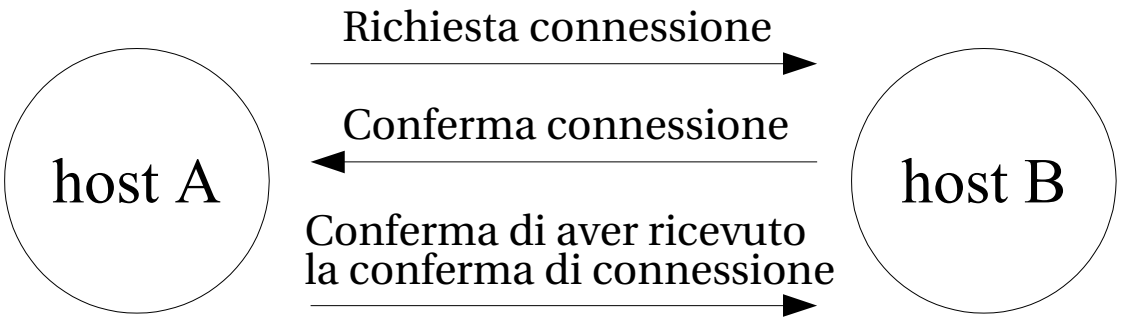
Il protocollo ICMP consente una comunicazione “straordinaria” tra routers ed hosts permettendo lo scambio di segnali di errore o di controllo (ad esempio i messaggi di redirect, che richiedono ad un host di cambiare la propria tabella di routing, e messaggi di echo request/echo reply, che l'host può usare per determinare se la destinazione può essere raggiunta).

- Echo request (type 8)
- Echo reply (type 0)
- Time exceeded (type 11)
- Destination unreachable (type 3)
- Redirect (type 5)

TCP Three-way handshake

Il three-way handshake assicura che la connessione tra due host A e B sia stata stabilita con successo.

- L'host A fa una richiesta di connessione verso l'host B (SYN).
- L'host B, ricevuta la richiesta di connessione, invia all'host A un datagramma di conferma (SYN + ACK).
- Ricevuto il datagramma di conferma, l'host A invia all'host B un datagramma che informa lo stesso host B che la conferma di connessione è arrivata allo host A (ACK).



Le porte del protocollo TCP/IP

Le porte sono il mezzo essenziale che permette ai protocolli TCP e UDP di gestire flussi multipli di dati attraverso una unica connessione fisica alla rete. Le porte sono indicate con un numero intero compreso tra 0 e 65.535 e possono essere assegnate sia al protocollo TCP che UDP.

7- ECHO
9 - DISCARD
13 - DAYTIME
20 - FTP-DATA (*FTP data transfer*)
21- FTP (*File Transfer Protocol*)
22 - SSH (*Secure Shell*)
23 - TELNET
25 - SMTP (*Simple Mail Transfer Protocol*)
42 - WINS (*Windows Internet Naming Service*)
53 - DNS (*Domain Name Server*)
80 - HTTP (*Hyper Text Transfer Protocol*)
110 - POP3 (*Post Office Protocol 3*)
119 - NNTP (*Network News Transfer Protocol*)
135 - EPMAP (*DCE Endpoint Mapper*)
137 - NETBIOS-ns (*name service*)
138 - NETBIOS-dgm (*datagram service*)
139 - NETBIOS-ss (*session service*)
143 - IMAP (*Internet Message Access Protocol*)
161 - SNMP (*Simple Network Management Protocol*)
389 - LDAP (*Lightweight Directory Access Protocol*)
443- HTTPS (*Secure HTTP*)
445 - Microsoft-ds (*Microsoft Directory Service*)
465 – SMTPS (*Secure SMTP*)
995 – POP3S (*Secure POP3*)

Tipi di porte

Lo **IANA** (*Internet Assigned Numbers Authority*) è l'ente che ha tra i suoi scopi la standardizzazione delle porte e l'aggiornamento costante di un documento, chiamato ports-number, contenente l'elenco dei servizi registrati e delle relative porte utilizzate. Nel suddetto documento, lo spazio delle 65536 porte UDP e TCP è stato suddiviso in tre parti:

- **Well Known Ports (porte 0 – 1023)**: sono porte assegnate univocamente e sono riservate ai servizi server standard.
- **Registered Ports (porte 1024 – 49151)**: l'utilizzo di questo insieme di porte è libero nonostante contenga dei servizi registrati.
- **Dynamic and/or Private Ports (porte 49152 – 65535)**: nessun servizio è registrato in quest'area. Il suo utilizzo è libero.

Tipi di porte

Dal punto di vista della sicurezza riveste grande importanza lo stato di una porta vista dall'esterno della macchina. Sono posssibili tre casi:

- **Porta aperta (Open)**: un processo server è in ascolto sulla porta. E' possibile stabilire una connessione dall'esterno.
- **Porta chiusa (Closed)**: nessun processo è in attesa e la porta è quindi inutilizzata. Se un client cerca di connettersi, il sistema operativo manda un segnale di reset della comunicazione.
- **Porta filtrata (Filtered)**: un firewall, filtro, o un altro ostacolo di rete sta “coprendo” la porta impedendo di determinare se la porta è aperta. In questo caso nessun segnale di reset viene inviato al client e il sistema rimane semplicemente muto.

Come rilevare lo stato di una porta?

Strumenti che ci permettono di stabilire lo stato di una porta:

- **Analisi del sistema locale**: visualizzazione delle connessioni attive (*established*) e delle porte in ascolto (*listening*) permettendo quindi di individuare tutti i server in esecuzione sul computer in esame.

  ```
  netstat -a
  ```

- **Analisi da un sistema remoto**: visualizzazione delle porte in ascolto (*listening*) con l'ausilio di un "portscanner" (es. Nmap). Questo test risulta utile tutte le volte che la nostra macchina o la nostra rete non è direttamente esposta su internet ma è separata da questa da un firewall, un router od un generico gateway.

  ```
  nmap -v -sS -O www.sito.com
  ```

Reti interne

La RFC 1918 (*Request for comments*) specifica i range di indirizzi IP che possono essere usati per le reti private e che non vengono quindi utilizzati da Internet. Tali indirizzi sono:

- 10.0.0.0 → 10.254.254.254
- 172.16.0.0 → 172.31.254.254
- 192.168.0.0 → 192.168.254.254

I computer con questi indirizzi possono accedere a Internet solo attraverso server proxy o collegandosi ad un router che effettui il mascheramento delle connessioni uscenti (noto come "NAT").

Firewall di Linux

Dalla versione 2.4 il kernel si basa sull'infrastruttura **netfilter/iptables** che permette di configurare un firewall per fare:

- **Packet filtering** (stateless, statefull): regole di filtraggio dei pacchetti che il firewall origina, riceve o che transitano dal firewall.
- **Network Address Translation** (NAT): regole per alterare l'intestazione dei pacchetti, tenendo traccia delle manipolazioni fatte ed operando l'operazione inversa sui pacchetti di risposta.
- **Packet mangling**: modifica di alcuni flag dei pacchetti.

Netfilter è il sistema di filtraggio dei pacchetti

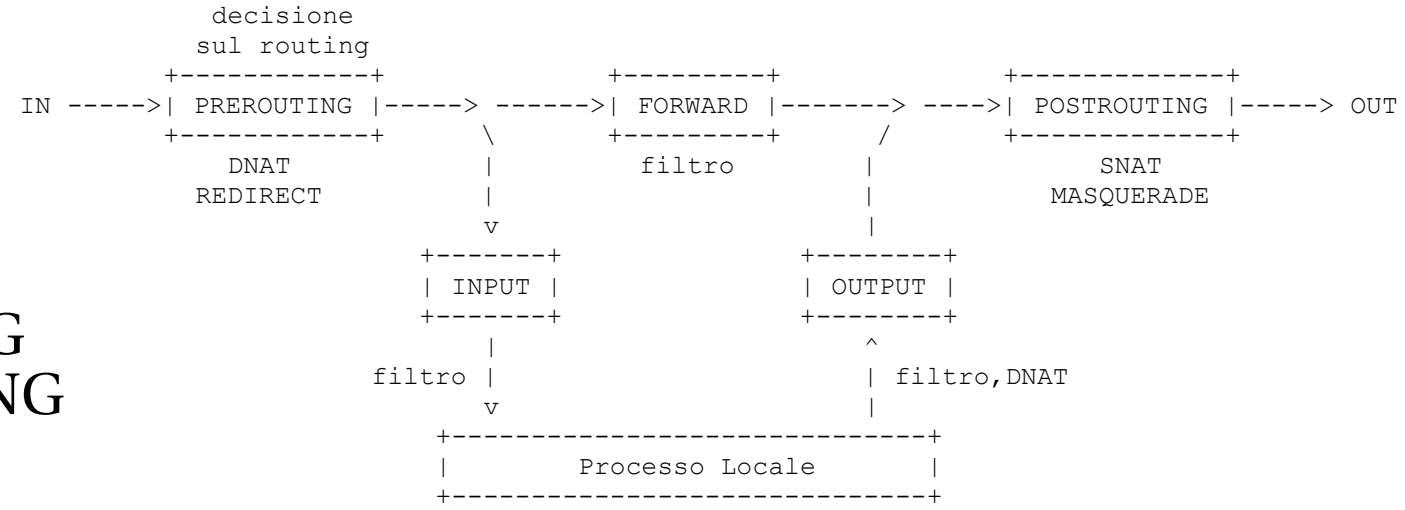
Iptables è il tool per creare le regole

Tabelle e catene

- Si basa sui concetti di **tabelle**, **catene** (*chains*) e **regole** (*rules*).
- Una tabella è formata da catene e una catena da regole.
- Una catena è un insieme di regole concatenate fra loro che vanno ad agire sulle intestazioni dei pacchetti per verificarne la corrispondenza.
- Per ciascuna catena, Iptables numera le regole in ordine crescente partendo dalla regola numero 1.

Tabelle e catene

- **filter**
  - INPUT
  - OUTPUT
  - FORWARD
- **nat**
  - PREROUTING
  - POSTROUTING
  - OUTPUT
- **mangle**
  - PREROUTING
  - INPUT
  - FORWARD
  - OUTPUT
  - POSTROUTING



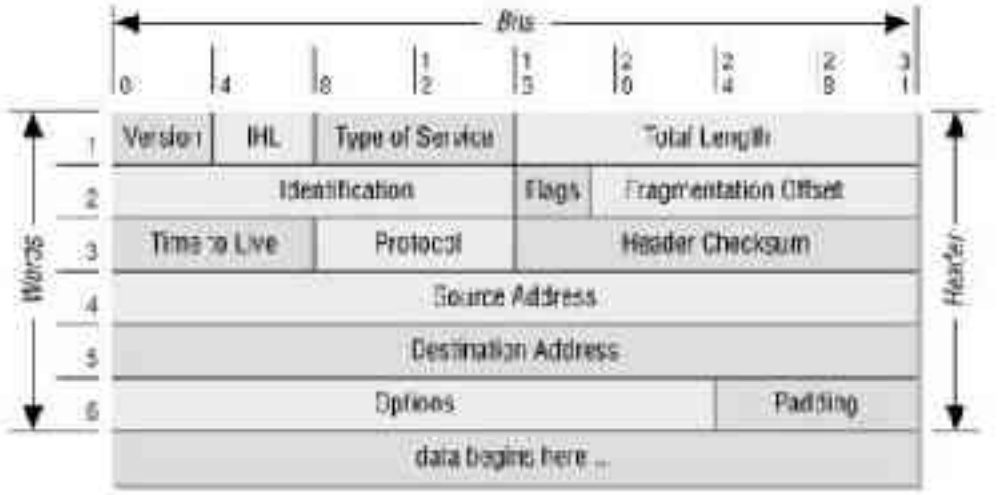
Le catene

- **INPUT**: contiene le regole di filtraggio per i pacchetti indirizzati al firewall stesso.
- **OUTPUT**: contiene le regole di filtraggio per i pacchetti generati da un processo interno al firewall e destinati verso un host esterno.
- **FORWARD**: contiene le regole di filtraggio per i pacchetti non destinati al firewall, secondo la tabella di routing.

- **PREROUTING**: contiene le direttive che devono essere applicate prima del processo di routing. A questo livello si applica il Destination NAT (DNAT).
- **POSTROUTING**: Direttive che devono essere aplicate dopo il processo di routing. A questo livello nel quale vengono applicate le direttive di Source NAT (SNAT).

Filtraggio... in base a cosa?

- **Livello fisico**
  - Interfaccia di rete
- **Livello data link**
  - MAC sorgente
- **Livello rete**
  - Indirizzo IP sorgente
  - Indirizzo IP destinazione
- **Livello trasporto**
  - Tipo di protocollo (TCP, UDP, ICMP, ...)
  - Porta TCP/UDP sorgente
  - Porta TCP/UDP destinazione
  - Tipo di messaggio ICMP

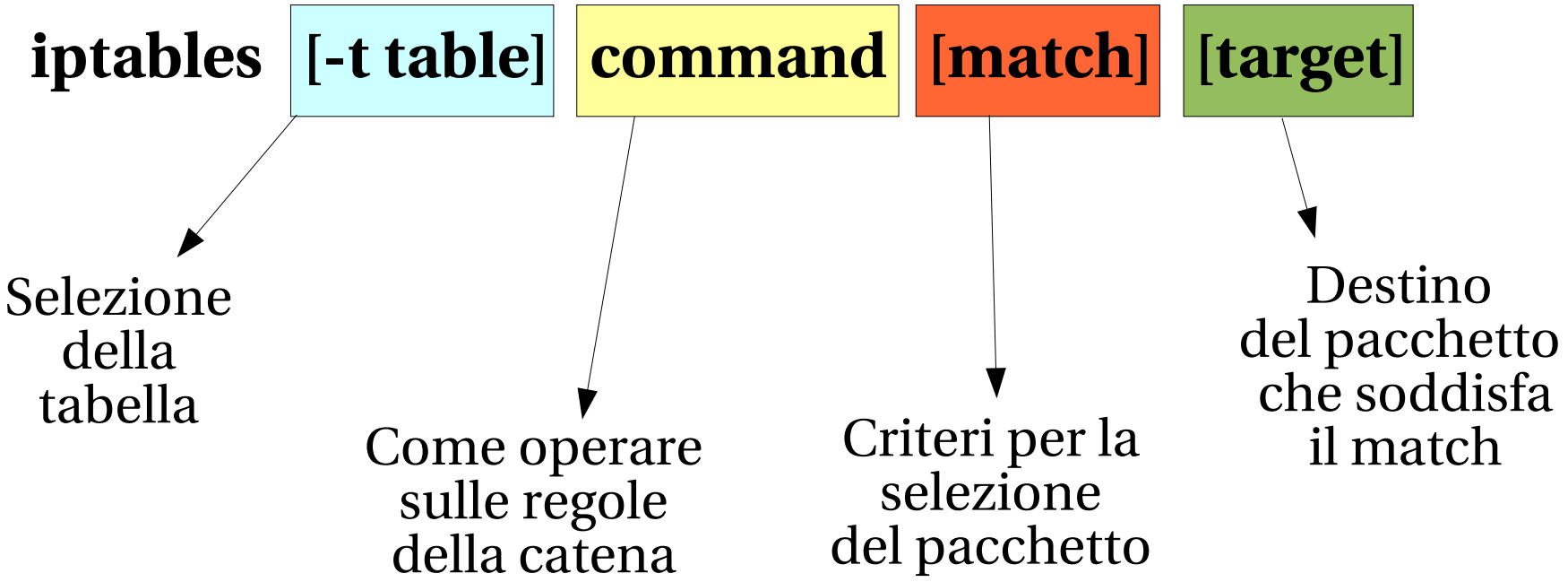


Alcune di queste informazioni della propria macchina posso essere ottenute dall'output dei seguenti comandi:

- ip
- ifconfig
- route
- arp

Come si costruisce una regola

Iptables accetta la sintassi seguente:



Command

I comandi di Iptables che agiscono sulle chain sono i seguenti:

-A | --append	Aggiunge una nuova regola in coda. precedenti.
-I | --insert	Inserisce una nuova regola in posizione...
-R | --replace	Sostituisce una regola con una nuova.
-D | --delete	Elimina una regola.
-L | --list	Visualizza le informazioni sulle chain.
-N | --new-chain	Crea una nuova chain.
-X | --delete-chain	Elimina una chain solo se vuota.
-P | --policy	Cambia la policy per una chain.
-F | --flush	Elimina tutte le regole sulle chain.
-Z | --zero	Azzera i byte e i contatori in tutte le chain.
-E | --rename-chain	Cambia nome ad una chain.

Match generiche

La match generica agisce su ogni tipo di protocollo che il kernel è capace di controllare. Non è richiesto nessun comando o opzione per controllare la match.

`-s | --source`	Specifica l'indirizzo IP sorgente.
`-d | --destination`	Specifica l'indirizzo IP destinazione.
`-p | --protocol`	Lista di protocolli separata dal carattere ",".
`-i | --in-interface`	Interfaccia di rete di ingresso.
`-o | --out-interface`	Interfaccia di rete di uscita.

Le regole nelle match possono venire anticipate dal carattere "!".
La sua funzione è di negare la regola.

Match TCP, UDP, ICMP

Match per il protocollo TCP:

- `--tcp-flags` → Filtra un flag specifico del TCP.
- `--syn` → Pacchetti con flag SYN attivato.
- `--sport | --source-port` → Porta di origine.
- `--dport | --destination-port` → Porta di destinazione.

Match per il protocollo UDP:

- `--sport | --source-port` → Porta di origine.
- `--dport | --destination-port` → Porta di destinazione.

Match per il protocollo ICMP:

- `--icmp-type` → Tipo di pacchetto.

Match esplicite

Match **MAC** (sono caricate con l'opzione "`-m mac`"):

`--mac-source` → Numero MAC dell'indirizzo sorgente.

Match **multiporta** (sono caricate con l'opzione "`-m multiport`"):

`--source-port` → Lista di porte sorgente separate da ",".
`--destination-port` → Lista di porte di destinazione separate da ",".
`--port` → Lista porte sia in ingresso che in uscita.

Match state

Queste match garantiscono l'accesso alle informazione sul “tracking” delle connessioni. Sono caricate con l'opzione “-m state”.

`--state` → Stato della connessione.
Indica lo stato che deve essere confrontato della connessione: INVALID, ESTABLISHED, NEW e RELATED.

Approcci al packet filtering

- **Stateless filtering**
  - La decisione sull'instradamento o meno di un pacchetto è presa esclusivamente sulla base della sola informazione contenuta nei campi header del pacchetto stesso
    - Indirizzo di provenienza/destinazione
    - Protocollo di trasporto
    - Informazioni addizionali

- **Statefull inspection**
  - La decisione è presa sulla base dell'informazione contenuta nel pacchetto e del traffico precedente, di cui è mantenuta memoria

Iptables è in grado di eseguire la statefull inspection...

Stateful inspection

Un pacchetto (TCP, UDP, ICMP) può essere in uno degli stati:

- **NEW**: crea una nuova connessione
- **ESTABLISHED**: appartiene a una connessione esistente
- **RELATED**: relativo, ma non parte di una connessione esistente (es. ICMP di errore)
- **INVALID**: non identificato o errore non relativo ad alcuna connessione

I target

Con “target” si intende il destino dato ai pacchetti da una regola.
I target sono molti e possono essere estesi tramite *patch*, i più comunemente utilizzati sono:

- **ACCEPT**: il pacchetto viene accettato.
- **DROP**: il pacchetto viene scartato, senza segnalazione.
- **REJECT**: simile a DROP, con segnalazione all'indirizzo sorgente.
- **LOG**: il pacchetto viene memorizzato nel file di log.
- **SNAT**: cambia l'indirizzo sorgente del pacchetto.
- **DNAT**: cambia l'indirizzo di destinazione del pacchetto.
- **MASQUERADE**: simile a DNAT...
- **ULOG**: trasmette i pacchetti loggati in user-space.

Esempi

Elimina tutte le regole impostate nella tabella “filter” (implicito):

```
# iptables -F
```

Elimina tutte le regole impostate nella tabella “nat”:

```
# iptables -t nat -F
```

Elimina tutte le catene definite dall'utente nella tabella “filter”:

```
# iptables -X
```

Per “policy” si intende la politica adottata in una catena.
Ogni catena ne ha una di default che ha come valore ACCEPT...

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

Esempi...

Accetta in ingresso le connessioni verso la porta ssh del firewall:

```
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Si impedisce l'ingresso a ogni pacchetto non proviene da 192.168.0.*

```
# iptables -A INPUT -s ! 192.168.0.0/24 -j DROP
```

Non permette il passaggio di connessioni verso la porta telnet:

```
# iptables -A FORWARD -i eth0 -o eth1 -p tcp  \
          --dport telnet  -j DROP
```

Accetta i pacchetti che appartengono ad una connessione nota o relativi ad essa (es. ICMP error, ftp data, ...)

```
# iptables -A INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT
```

Esempi

Logga tutte le connessioni dell'host 10.0.0.3 verso la porta 80 locale.

```
# iptables -A INPUT -s 10.0.0.3 -p tcp –dport 80 \
          -j LOG --log-prefix "Dropped (http): "
```

Rigetta tutte le connessioni TCP.

```
# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```

Visualizza le regole nella tabella “filter”.

```
# iptables -A INPUT -m mac --mac-source  00:0A:5E:20:10:FE \
          -p tcp --dport pop3 -j ACCEPT
```

Visualizza le regole nella catena di INPUT.

```
# iptables -L
# iptables -L INPUT -v -n --line-numbers
```

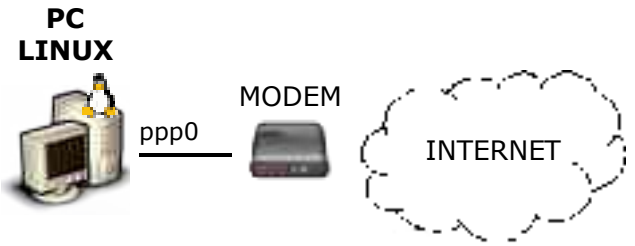
Approcci al packet filtering

Esistono due approcci ortogonali per la politica di sicurezza:

- **Default Permit Stance**: si vietano soltanto alcuni servizi particolari, partendo da un sistema aperto.
- **Default Deny Stance**: si vieta qualunque cosa per default e si controllano direttamente i servizi che si vogliono abilitare. In altre parole, tutto ciò che non è esplicitamente permesso e proibito!

La nostra analisi si basa sul primo approccio.

PC di casa



IP DINAMICO

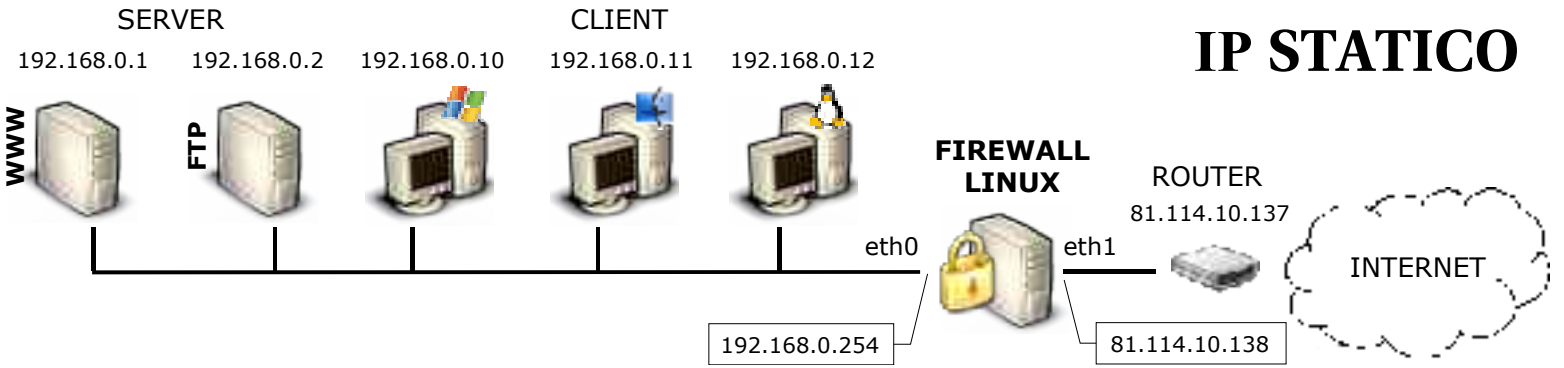
```
# iptables -F
# iptables -P INPUT DROP
# iptables -P OUTPUT ACCEPT
# iptables -A INPUT -i ppp0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Tipi di NAT

- **DNAT** : utilizzato per indirizzare l'indirizzo IP di destinazione verso un altro nodo, appartenente ad una zona protetta dal firewall.
- **SNAT**: utilizzato per riscrivere l'indirizzo IP sorgente del pacchetto. L'utilizzo principale si ha quando più computer condividono la stessa connessione Internet con un unico IP statico assegnato al firewall o al router. Senza questa traduzione la LAN non potrebbe non potrebbe comunicare con l'esterno in quanto, generalmente, usa indirizzi IP privati e non pubblici.
- **MASQUERADE**: equivalente a SNAT, tuttavia non richiede di specificare un numero IP sorgente in quanto questo viene calcolato ogni volta. L'utilizzo principale si ha quando più condividono la stessa connessione Internet con un unico Ip dinamico assegnato al firewall o al router.

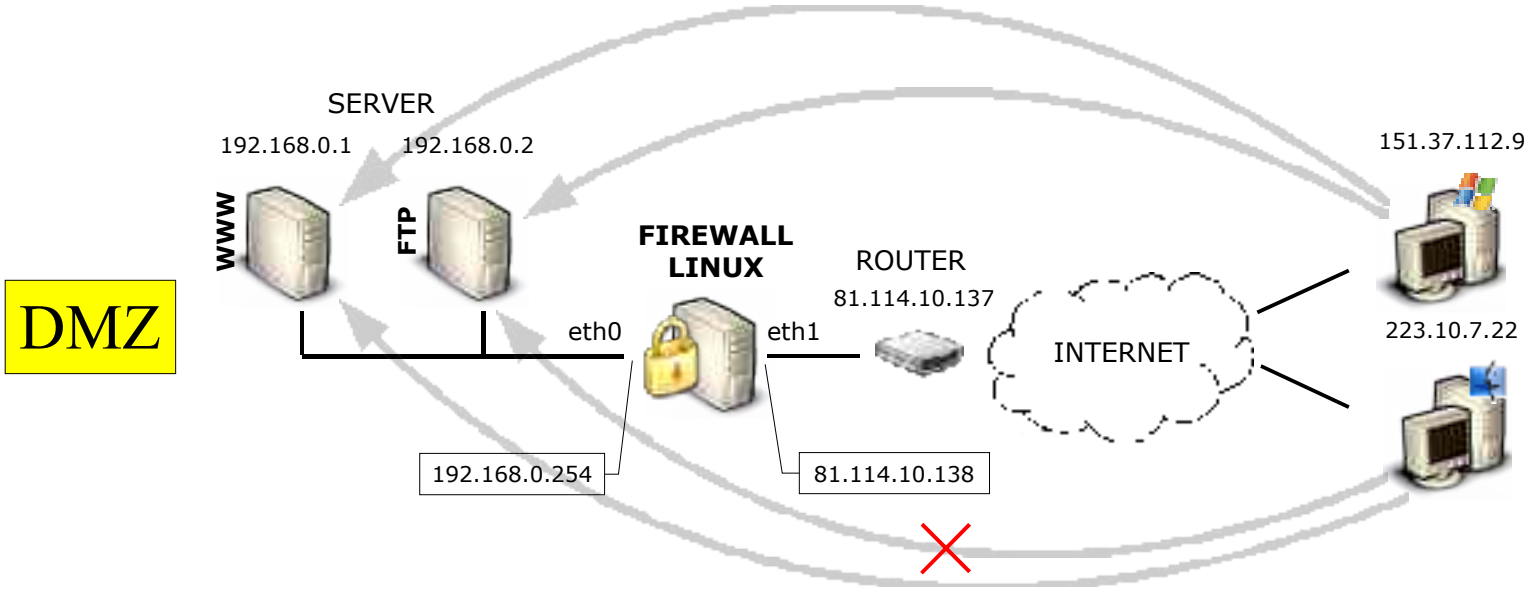
.

LAN aziendale




```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# iptables -F
# iptables -t nat -F
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT ACCEPT
# iptables -A INPUT -i eth0 -s 192.168.0.0/24 -p tcp –dport ssh -j ACCEPT
# iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A FORWARD -i eth0 -o eth1 -s 192.168.0.0/24 -j ACCEPT
# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE
```

LAN aziendale + servizi pubblici




```
...
# iptables -t nat -A PREROUTING -i eth1 -d 81.114.10.138 \
          -p tcp --dport http -j DNAT -to 192.168.0.1

# iptables -t nat -A PREROUTING -i eth1 -s 151.37.112.9 -d 81.114.10.138 \
          -p tcp --dport ftp -j DNAT -to 192.168.0.2
```

Le connessioni vanno permesse anche sulla catena FORWARD

Conclusioni

- Il firewall è un elemento essenziale per raggiungere un elevato livello di sicurezza su una rete locale, ma non basta installare un sistema di firewall per avere la certezza dell inviolabilità della propria rete.
- Un firewall ha bisogno di costante controllo e mantenimento del software che lo costituisce. E' essenziale tenere aggiornato il proprio firewall con tutte le *patch* di sicurezza per mettersi al riparo da eventuali falle che potrebbero permettere ad un malintenzionato di penetrare nel sistema. Proprio per ridurre questa eventualità si consiglia di disabilitare tutti i servizi non strettamente necessari.
- Esistono varie tipologie di firewall ognuna delle quali può avere pregi e punti deboli.
- La realizzazione di un architettura di firewall è spesso non banale da realizzare e non esiste una ricetta generale per risolvere i problemi che possono nascere nella configurazione, installazione e mantenimento di un firewall. Ogni architettura di firewall è quindi specifica alla tipologia particolare di LAN a cui viene applicata.
- Il modello di firewall che si può implementare dipende molto dalle politiche di sicurezza che devono essere decise a priori ed applicate con coerenza.

Bibliografia

- W. R. Stevens, “TCP/IP Illustrated”, Addison Wesley, 1994
- C. Hunt, “TCP/IP Network Administration”, O'Reilly, 1995
- R. Russell - “Packet filtering HOWTO”
- E. Bruni - “Linux Firewall”, 2002
- C. Contavalli - “Iptables for Fun”, 2003
- M. Lotto, “Network Packet Filtering”, 2003
- G. Bianchini - “Sistemi firewall”, 2002
- F. Bucciarelli - “Filtraggio del traffico IP in linux”, 2003
- R. Veraldi – “Firewalls”, 2000
- V. Vecchione - “Netfilter: scacco matto all'intruso!”, 2002
- Yan Raber, “Le porte del protocollo TCP/IP”
- Umberto Zanatta, “linuxDidattica: la Rete”
- Sito web Netfilter (http://www.netfilter.org)
- Manpage di iptables