

# *Packet Filtering con iptables*

Cherubini Enrico – InfoSysNet Snc

<kevin@bestkevin.com>

Linux Day 2004 – 27/11/2004

(liberamente tratto dal documento di Scali  
Omar)

Rilasciato sotto licenza GPL

---

---

## *Packet Filtering con iptables*

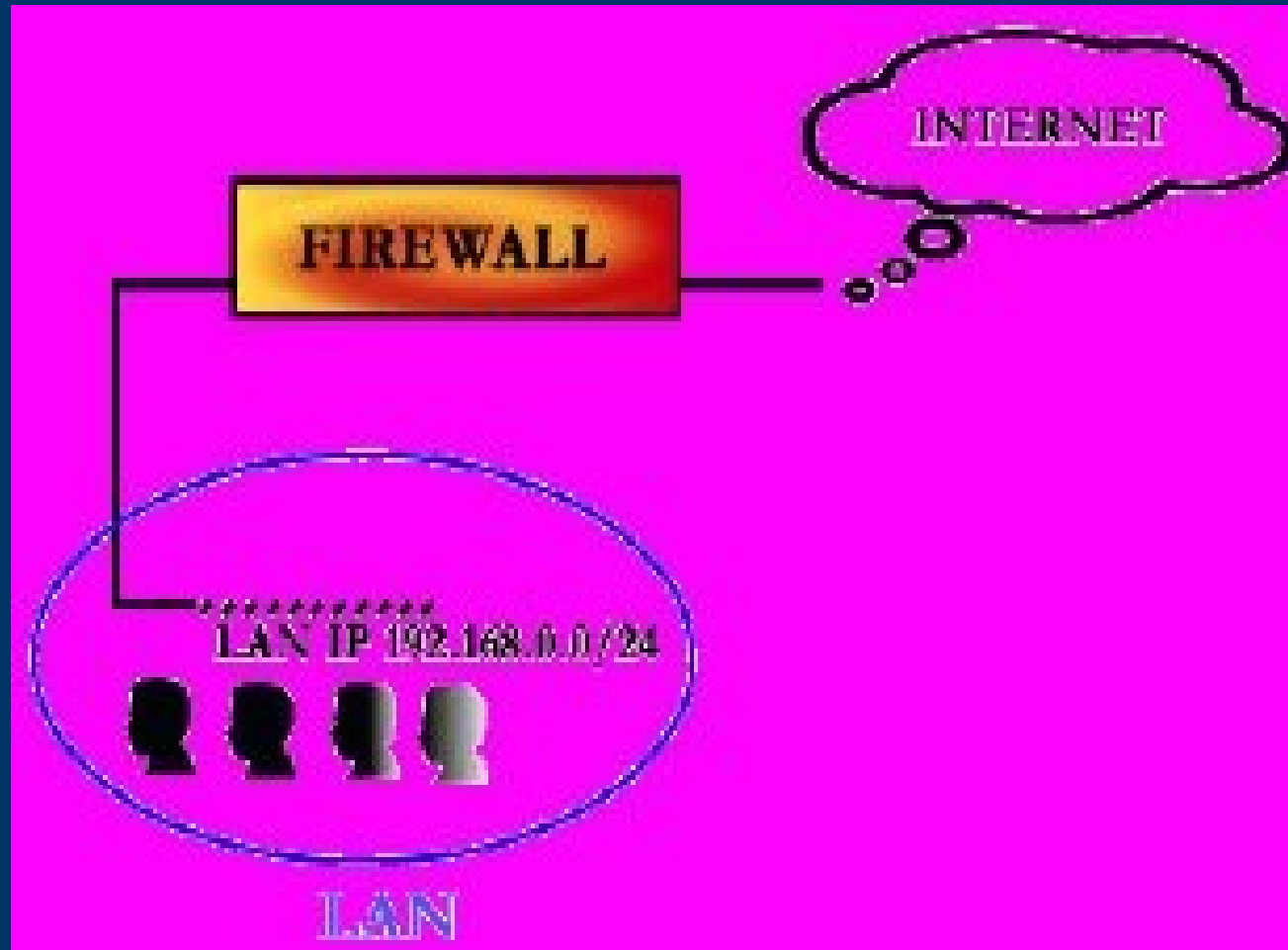
### Firewall

Un firewall in genere è composto da uno o più computer o dispositivi di rete che si interpongono tra le reti private e quelle esterne.

Il compito principale è quello di controllare i dati in transito tra le reti.

In particolare, un firewall viene utilizzato per stabilire quali servizi devono essere accessibili alla rete interna, come la posta e il web e quali possono essere resi disponibili ai visitatori dall'esterno.

## Packet Filtering con iptables



## *Packet Filtering con iptables*

### Tipologie di firewall:

#### Packet filtering

Tale tecnica controlla ogni singolo pacchetto che transita da una rete all'altra utilizzando le informazioni contenute all'interno del suo header, accettandolo o rifiutandolo secondo le politiche espresse.

---

---

## *Packet Filtering con iptables*

### Application gateway

In tale situazione l'applicazione chiamata proxy si occupa di autorizzare e inoltrare i pacchetti in transito tramite l'impostazione di coppie account/password e di conseguenza applicare le regole impostate.

---

---

## *Packet Filtering con iptables*

### Packet inspection

Questo tipo di tecnica esegue ulteriori e approfonditi controlli sul pacchetto in transito, tecnica utilizzata nei sistemi IDS (Intrusion Detection System).



## *Packet Filtering con iptables*

### Netfilter

IpTables è il firewall standard per i kernel 2.4 e successivi, esso però è solo una parte di una infrastruttura inglobata nel kernel chiamata Netfilter realizzata da Rusty Russel ([www.netfilter.org](http://www.netfilter.org)).

---

---

## *Packet Filtering con iptables*

Tale nuova struttura ci consente di scrivere appositi moduli per gestire il filtraggio o la manipolazione dei pacchetti e di caricarli solamente quando necessario all'interno del kernel.

---

---

## *Packet Filtering con iptables*

Questa logica di filtraggio basata su un prerouting permette una maggiore efficienza nell'applicazione delle policy, in questo modo ogni pacchetto attraversa una sola catena di filtraggio contrariamente a quanto accade nelle precedenti versioni del kernel.

---

---

## *Packet Filtering con iptables*

### Gestione logica

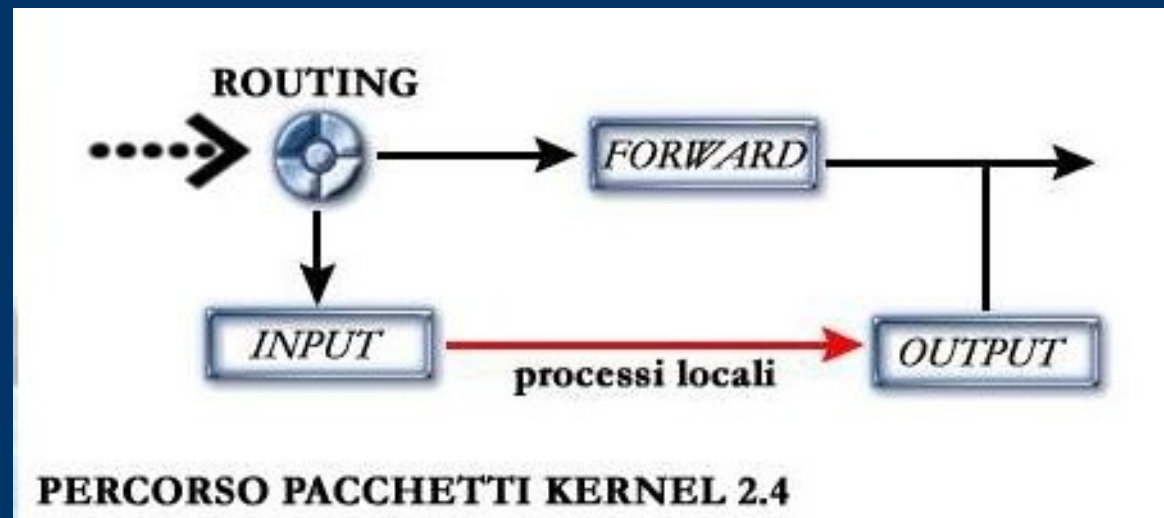
Netfilter tratta i pacchetti in maniera differente rispetto alle precedenti versioni del kernel, quando uno di essi arriva dalla rete viene prima di tutto sottoposto ad un processo di routing e quindi se destinato o proveniente dalla rete locale e diretto verso l'esterno è sottoposto solamente al filtraggio della catena FORWARD.

---

---

## *Packet Filtering con iptables*

La catena INPUT agisce solamente per tutti i pacchetti destinati esclusivamente al firewall, mentre la catena OUTPUT si occupa di filtrare esclusivamente quelli generati



## *Packet Filtering con iptables*

### Gestione logica

Le catene principali accettano di default tutti i pacchetti in transito.

### **INPUT**

Utilizzabile per tutti i pacchetti destinati esclusivamente alla macchina firewall e quindi elaborati dai processi locali.



## *Packet Filtering con iptables*

### **FORWARD**

Pacchetti destinati ad una delle macchine della rete locale (LAN) o provenienti da essa e dirette all'esterno e non al firewall.

### **OUTPUT**

Pacchetti generati dalla macchina firewall diretti all'esterno.

---

---

## *Packet Filtering con iptables*

### Utilizzo di iptables

Il binario iptables consente di intervenire e modificare le catene di regole secondo le nostre necessità, tramite questo comando possediamo un completo controllo sul firewall della nostra linux box, illustriamo ora un esempio che ci permette di chiarire la sintassi utilizzata:

---

---

## *Packet Filtering con iptables*

```
# iptables -t filter -A FORWARD -p tcp -i eth0 -o eth1 --dport ssh -j DROP
```

Questa regola imposta il firewall in modo da scartare ogni tentativo di accesso ssh all'interno della nostra rete locale, analizziamo di seguito le varie opzioni utilizzate:

---

---

## *Packet Filtering con iptables*

### tabella (-t)

L'infrastruttura netfilter ha introdotto tre tipi di tabelle che facilitano le impostazioni delle regole, la prima (filter) contiene le tabelle preesistenti ed è utilizzata per impostare le regole di filtraggio, la seconda (nat) è utilizzata per le regole che riguardano il masquerading mentre l'ultima (mangle) viene utilizzata per la manipolazione dei pacchetti come il marcammento e i bit TOS.

## *Packet Filtering con iptables*

```
# iptables -t filter -A FORWARD -p tcp -i  
eth0 -o eth1 --dport ssh -j DROP
```

### **catena (-A)**

Una catena consiste in una policy che stabilisce quali pacchetti devono essere accettati e quali no, tale opzione permette di aggiungere una determinata regola alle catene elencate, sono disponibili anche altre opzioni che permettono di creare (-N) o cancellare (-D) le catene.

---

---

## *Packet Filtering con iptables*

### protocollo (-p)

IPTables permette di applicare le proprie regole in base ad un determinato tipo di protocollo, sono disponibili i comuni tcp, udp, icmp.

### interfaccia (-i -o)

Il filtraggio dei pacchetti può avvenire in base al nome dell'interfaccia di ingresso (-i) oppure di uscita (-o), molto utile quando si dispone di più interfacce di rete sulla stessa macchina.

---

---

## *Packet Filtering con iptables*

```
# iptables -t filter -A FORWARD -p tcp -i  
eth0 -o eth1 --dport ssh -j DROP
```

*porta sorgente e/o destinazione (--sport --  
dport)*

Queste due opzioni permettono di stabilire la porta (tramite il numero oppure il nome) di destinazione (--dport) o di sorgente ( --sport) da utilizzare all'interno della nostra policy.

---

---

## *Packet Filtering con iptables*

### regole (-j oppure -jump)

Questa opzione ci permette di definire, attraverso una delle parole chiave, il destino (chiamato obiettivo) del pacchetto che soddisfa la regola indicata



## *Packet Filtering con iptables*

Ecco alcune opzioni utilizzabili:

**ACCEPT** Accetta i pacchetti che soddisfano la regola indicata

**DROP** Scarta il pacchetto (sostituto di DENY su ipchains)

**REJECT** Scarta il pacchetto e avvisa tramite il messaggio (port unreachable)

**QUEUE** Accoda i pacchetti per una successiva elaborazione



## *Packet Filtering con iptables*

**MIRROR** Rispedisce il pacchetto alla macchina che lo ha generato

**LOG** Registrazione dei pacchetti, obiettivo dotato di ulteriori opzioni

**MASQUERADE** Utilizzato per impostare il NAT da ip dinamico



## *Packet Filtering con iptables*

### Esempi di utilizzo

```
# iptables -A INPUT -s 192.168.200.0/24 -d  
0/0 -i eth0 -j DROP
```

Questa regola blocca tutto il traffico in ingresso destinato al sistema locale proveniente dalla rete 192.168.200.\* attraverso l'interfaccia eth0.

---

---

## *Packet Filtering con iptables*

```
# iptables -A OUTPUT -d www.msn.it -p  
tcp -j DROP
```

Questa regola ci permette di bloccare tutto il traffico tcp generato dal sistema locale che risulta destinato alla sito [www.msn.it](http://www.msn.it)

## *Packet Filtering con iptables*

```
# iptables -A INPUT -p icmp -s !  
192.168.0.0/16 -icmp-type 8 -j DROP
```

Questa regola ci permette invece di bloccare i pacchetti ICMP tipo 8 (echo\_request) provenienti da indirizzi estranei alla rete 192.168.\*.\* e destinati sempre alla rete stessa.

---

---

## *Packet Filtering con iptables*

```
# iptables -A INPUT -s www.microsoft.com  
-p tcp --dport ftp -j DROP
```

Questo esempio invece ci permette di bloccare i tentativi di connessione al nostro server ftp dalla rete

## *Packet Filtering con iptables*

### NAT (Network Address Translation)

Dopo aver preso confidenza con la sintassi è necessario, per andare avanti, illustrare il funzionamento del NAT. Il suo utilizzo fornisce una serie di vantaggi, i vari ISP forniscono un solo indirizzo ip durante il collegamento, tramite il NAT invece è possibile collegare una serie di client (come una rete locale) e utilizzare un solo collegamento ad internet passante per il firewall.

---

---

## *Packet Filtering con iptables*

Esistono due tipologie di NAT:

Il **Source NAT** si utilizza quando si vuole alterare l'indirizzo sorgente del primo pacchetto ossia si decide di cambiare la provenienza della connessione.

SNAT viene sempre applicato in fase di post-routing, prima che il pacchetto venga immesso in rete, il mascheramento (masquerading) è una forma specializzata di SNAT.

---

---

## *Packet Filtering con iptables*

Il **Destination NAT** viene utilizzato quando si vuole alterare l'indirizzo di destinazione del primo pacchetto ovvero viene cambiata la destinazione della connessione. DNAT viene sempre effettuato in fase di pre-routing, appena il pacchetto arriva dalla rete.

Il port forwarding, il load sharing e il proxy trasparente sono tutte forme avanzate di DNAT

---

---

## *Packet Filtering con iptables*

### Connection Tracking

Ora introduciamo un nuovo interessante modulo chiamato `ip_conntrack`, oltre ad essere una novità assoluta esso ci permette di tenere traccia delle connessioni stabilite indicandoci se un pacchetto in transito appartiene o è relativo ad essa, tale tecnica viene chiamata "stateful firewall".

---

---

## *Packet Filtering con iptables*

Tale meccanismo durante una connessione al sistema provvede ad annotare le informazioni più importanti e inoltre è in grado di stabilire a quale categoria di connessione appartiene, di seguito un elenco:

**NEW** Pacchetti che stabiliscono una nuova connessione

**ESTABLISHED** Pacchetti appartenenti ad una connessione esistente

---

---

## *Packet Filtering con iptables*

**RELATED** Pacchetti relativi ad una connessione esistente anche se non ne fanno parte direttamente, come i pacchetti icmp di errore.

**INVALID** Pacchetti sospetti non appartenenti a nessuna connessione, in generale essi sono da rifiutare

---

---

## *Packet Filtering con iptables*

### Connection Tracking

Per l'utilizzo di tale caratteristica è necessario indicare l'opzione ``-m state --state categoria'` come nell'esempio seguente:

```
# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
```

Questa regola accetta tutti i pacchetti in ingresso appartenenti o relativi ad una connessione nota.

---

---

## *Packet Filtering con iptables*

```
# iptables -A FORWARD -d 192.168.0.0/16  
-m state --state INVALID -j DROP
```

Tramite questa specifica indichiamo al firewall di eliminare il pacchetto destinato agli indirizzi 192.168.\*.\* quando questi non sono identificabili con sessioni

---

---

## *Packet Filtering con iptables*

### Sistema di logging

Il sistema di logging deve essere richiamato attraverso l'opzione `-j LOG`, questa modifica ha consentito di introdurre nuove opzioni che ci permettono di memorizzare i log in file differenti con l'aggiunta di righe commentate, ecco le due principali opzioni comunemente utilizzate:

---

---

## *Packet Filtering con iptables*

--log-level

Possiamo specificare un nome di livello tra quelli disponibili nel sistema syslogd (debug, info, notice, warning, err, crit, alert), il risultato sarà scritto nel file di log associato.

---

---

## *Packet Filtering con iptables*

--log-prefix

Tale opzione ci permette di inserire una stringa nel file di log in modo da identificare con precisione l'evento registrato.

Vediamo subito un esempio per chiarire meglio la sintassi utilizzata:

```
# iptables -A INPUT -p tcp --dport telnet --  
syn -j LOG --log-level info --log-prefix  
"Telnet Request"
```

---

---

# *Packet Filtering con iptables*

Arrivederci e grazie.

Enrico Cherubini

<kevin@bestkevin.com>

---

---