

Linux firewalling: IpTables in ambiente aziendale

Implementazione di un firewall aziendale mediante tecniche di packet filtering e network address translation

Gian Piero Borello - gianpiero.borello@infm.it

INFN-CNR (Istituto Nazionale per la Fisica della Materia - Consiglio Nazionale delle Ricerche)



LUGGE
Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



INFM - CNR (Istituto Nazionale per la Fisica della Materia - Consiglio Nazionale delle Ricerche)

Bersaglio appetibile per gli hacker (circa 50 attacchi al giorno)

- Ente di ricerca/universitario - “tanto” traffico di rete/facile nascondersi
- Siti “famosi” ospitati:
 - <http://www.infm.it> - Sito istituzionale
 - <http://www.festivalscienza.it> - Festival della Scienza
 - <http://www.iit.it> - Istituto Italiano di Tecnologia

Problemi “interni”

- LAN con circa 100 utenti (disattenzioni, sbagli, dolo,...)

Necessità di sicurezza ANCHE tramite politiche di firewalling robuste

Cosa deve fare un firewall in un ambiente aziendale ?

Un firewall è un dispositivo di rete che si interpone fra due o più reti il cui compito è quello di controllare i dati in transito fra le reti stesse. In ambito aziendale il firewall deve:

- Proteggere i computer (LAN) ed i server (DMZ) aziendali dagli attacchi esterni, rendendo però accessibili i servizi all'utenza esterna della rete.
- Impedire agli utenti della LAN operazioni non consentite (**importante**).
- Evitare che il traffico permesso possa provocare dei danni (a voi e ad organizzazioni esterne).
- Evitare il più possibile i danni che vi potrebbe arrecare un attaccante nel caso riuscisse ad impossessarsi di una delle vostre macchine o dei vostri server (**importante**).

Brevissima storia dei firewall Linux

IpTables è il firewall standard per i kernel 2.4.x e successivi, esso però è solo una parte di una infrastruttura inglobata nel kernel chiamata Netfilter e realizzata da Paul "Rusty" Russell e dal Netfilter Core Team a partire dal 1998.

Kernel 2.0.x - **Ipfwadm** (basato su ipfw di BSD)

Kernel 2.2.x - **IpChains**, concetto di catene di regole (access-list) a cui un pacchetto può essere indirizzato. Non è stateful.

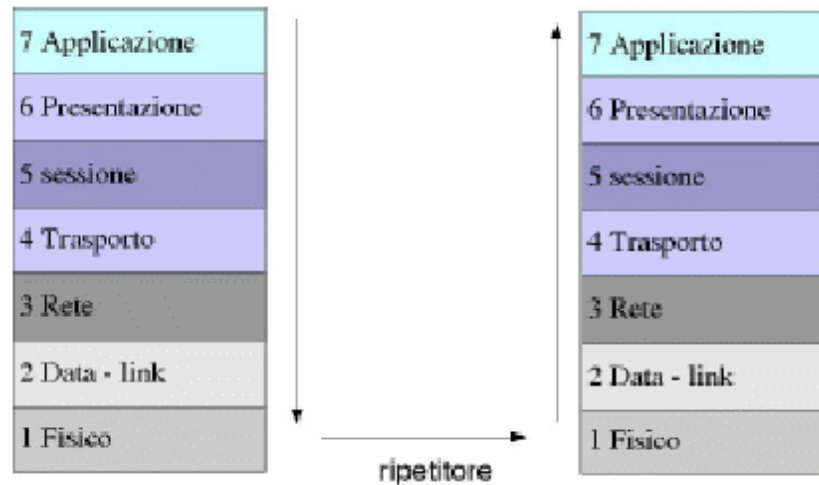
Kernel 2.4.x - **IpTables**, vi è un notevole salto in avanti introducendo nuove caratteristiche:

- Packet filtering di tipo stateful
- Diversi tipi di NAT (manipolazione dell'intestazione dei pacchetti)
- Packet mangling (manipolazione del payload dei pacchetti).

Cosa è un Packet filter stateful ? (1)

I packet filter (stateless/stateful) sono una particolare categoria di firewall in grado di analizzare l'**header** dei pacchetti di rete fino al livello 4 ISO-OSI (trasporto). Ciò significa che sono in grado di discriminare un pacchetto in base all'header di:

- livello 1 (Fisico) - interfaccia da cui proviene o è diretto il pacchetto
- livello 2 (Data-link) - in base agli indirizzi MAC sorgente e/o destinazione
- livello 3 (Rete) - in base agli indirizzi IP sorgente e/o destinazione
- livello 4 (Trasporto) - in base alla porta sorgente e/o destinazione

**LUGGE**

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO

Cosa è un Packet filter stateful ? (2)

Vantaggi:

- veloci e performanti anche con poca CPU
- analizzano i pacchetti indipendentemente dalla applicazione che li ha generati
- girano in kernel space

Svantaggi:

- NON possono controllare il payload del pacchetto a livello di application

I packet filter di tipo **stateful** sono in grado di basare le proprie politiche di filtraggio **anche** sullo stato corrente delle connessioni, spingendo quindi il proprio campo di azione al livello 5 ISO-OSI (sessione), correlando le diverse connessioni relative ad una stessa sessione grazie all'aiuto del sistema di tracciamento delle connessioni (*connection tracking conntrack*) e di moduli software dedicati alla gestione dei protocolli più complessi.

In pratica l'analisi di ogni pacchetto che transita risente anche dello "stato" e della storia che ha generato il pacchetto stesso.



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Struttura di IpTables - Tabelle

IpTables basa la sua struttura principale su tre tabelle:

Funzione	Tabella	Catena	Scopo
Filtraggio pacchetti	filter	INPUT	Filtraggio traffico in ingresso
		OUTPUT	Filtraggio traffico in uscita
		FORWARD	Filtraggio traffico in transito
Manipolazione header – indirizzi/porte TCP/IP (NAT, PAT)	nat	PREROUTING	DNAT – Destination NAT
			PAT – Port Address Translation
		POSTROUTING	SNAT – Source NAT
			MASQUERADE
Manipolazione header – altri campi	mangle	INPUT	TOS – Type Of Service TTL – Time To Live (pochissimo usata)
		OUTPUT	
		FORWARD	
		PREROUTING	
		POSTROUTING	


LUGGE

Linux Users Group Genova

 ASSOCIAZIONE SENZA
SCOPO DI LUCRO

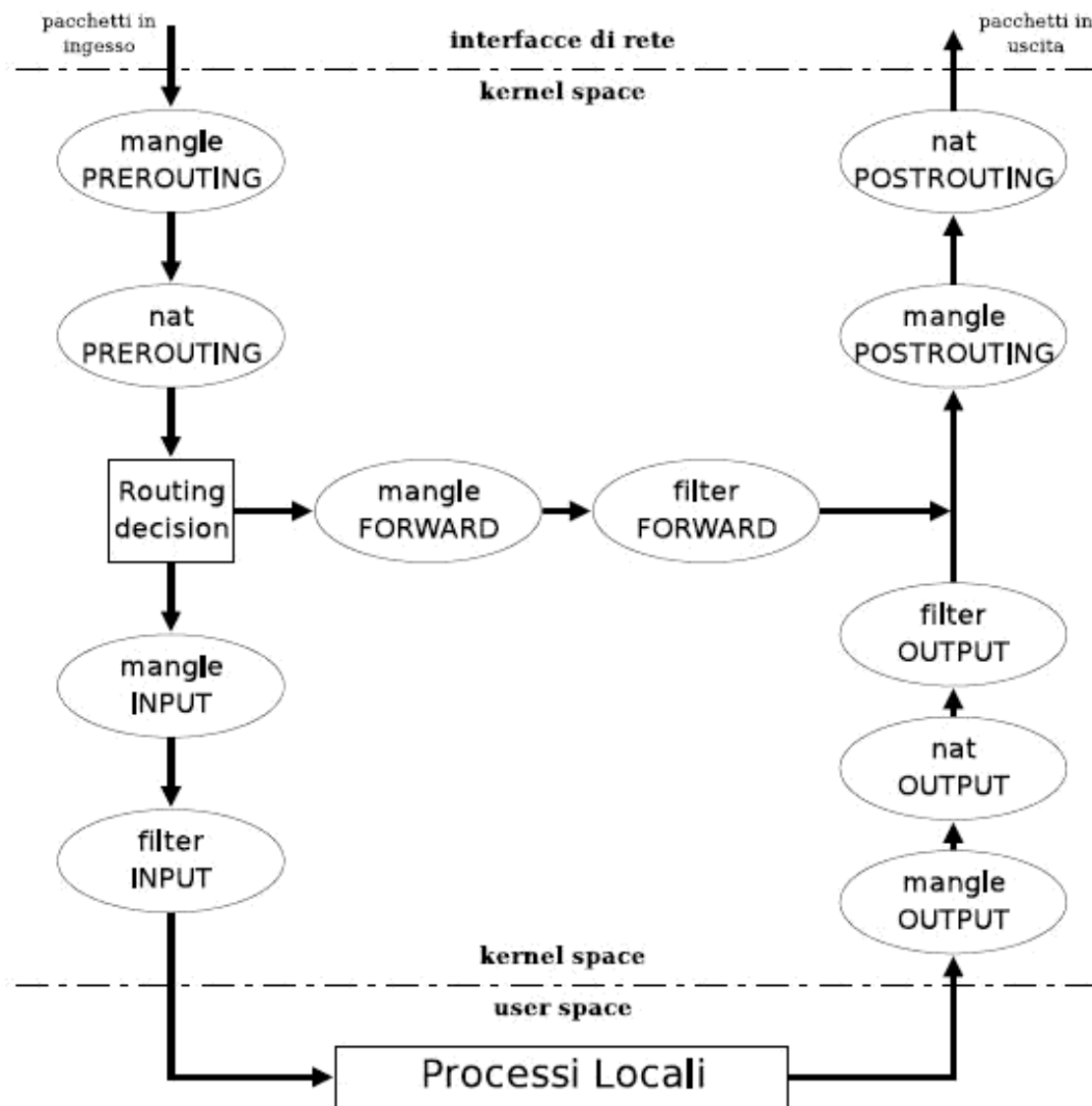

Struttura di IpTables - Catene

Una catena è una **collezione di regole**.

Ogni volta che giunge un pacchetto di rete in una catena, IpTables esamina la collezione di regole, di cui è composta, **una alla volta e nell'ordine in cui sono state scritte**. Due possibili scenari:

- Esiste una regola della catena che corrisponde al pacchetto in esame: Il destino del pacchetto è deciso dal **target della regola** (possibili target: ACCEPT, DROP, QUEUE, RETURN, MARK, MASQUERADE, REJECT, LOG, TOS, SNAT, DNAT)
- Nessuna regola della catena corrisponde al pacchetto in esame: Il destino del pacchetto è la **politica di default** della catena (politiche più usate: ACCEPT, DROP, REJECT).

Possibilità di creare catene user-defined



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Cenni sulla sintassi di IpTables - Generale (1)

La sintassi generale del comando IpTables è la seguente:

```
iptables [-t <tabella>] -<Comando> <Catena> <Regole> -j <target>
```

Tramite il flag -t si indica la tabella alla quale ci stiamo riferendo, se viene omesso ci si riferisce a "filter". Possibili valori:

- filter
- nat
- mangle

Cenni sulla sintassi di IpTables - Comandi (2)

Comandi sulle catene:

- Crea una nuova catena -N
- Cancella una catena vuota -X
- Cambia la default policy ad una catena -P
- Elenca le regole in una catena -L
- Cancella (tutte) le regole di una catena -F
- Azzera i contatori di una catena -Z

Comandi sulle regole all'interno di una catena:

- Appende una regola alla catena -A
- Inserisce una nuova regola in una data posizione -I
- Rimpiazza una regola in una data posizione -R
- Elimina una regola -D

Cenni sulla sintassi di IpTables - Regole (3)

IpTables può **filtrare** i pacchetti in base a molti parametri fra cui:

- ip protocol *-p protocol*
- source/destination IP address *-s src_port -d dest_port*
- source/destination TCP port *-sport src_port -dport dest_port*
- input/output interface *-i interf -o interf*
- in base ai flag del protocollo TCP *--tcp-flags SYN, ACK, FIN*
- mac address *--m mac --mac-source xx:yy:hh:jj:kk:ww*
- negazioni delle precedenti voci (es *! -p tcp* significa tutti i protocolli a parte quello TCP)
- in base allo stato della connessione (stateful) *-m state --state state_type*

Cenni sulla sintassi di IpTables - Target (4)

Tramite il flag -j viene decisa infine la sorte dei pacchetti che corrispondono alla regola (matching). I principali (e più usati) target sono:

- ACCEPT - i pacchetti che corrispondono alla regola vengono accettati
- DROP - i pacchetti che corrispondono alla regola vengono distrutti senza risposta all'host mittente
- REJECT - i pacchetti che corrispondono alla regola vengono rifiutati avvisando dell'azione l'host mittente
- LOG - le intestazioni del pacchetto che corrisponde alla regola vengono inserite in un file di log. In genere in /var/log/kern.log

Esempi di regole di filtering con IpTables

```
iptables [-t filter] -P FORWARD ACCEPT
```

la politica di default della catena FORWARD di filter è ACCEPT

```
iptables [-t filter] -A OUTPUT -dport 80 -j ACCEPT
```

faccio uscire il traffico http

```
iptables [-t filter] -A INPUT -s 193.205.152.1 -j DROP
```

non accetto l'ingresso del traffico proveniente da 193.205.152.1

```
iptables [-t filter] -A FORWARD -dport 22 -j LOG
```

log del traffico SSH che attraversa la macchina



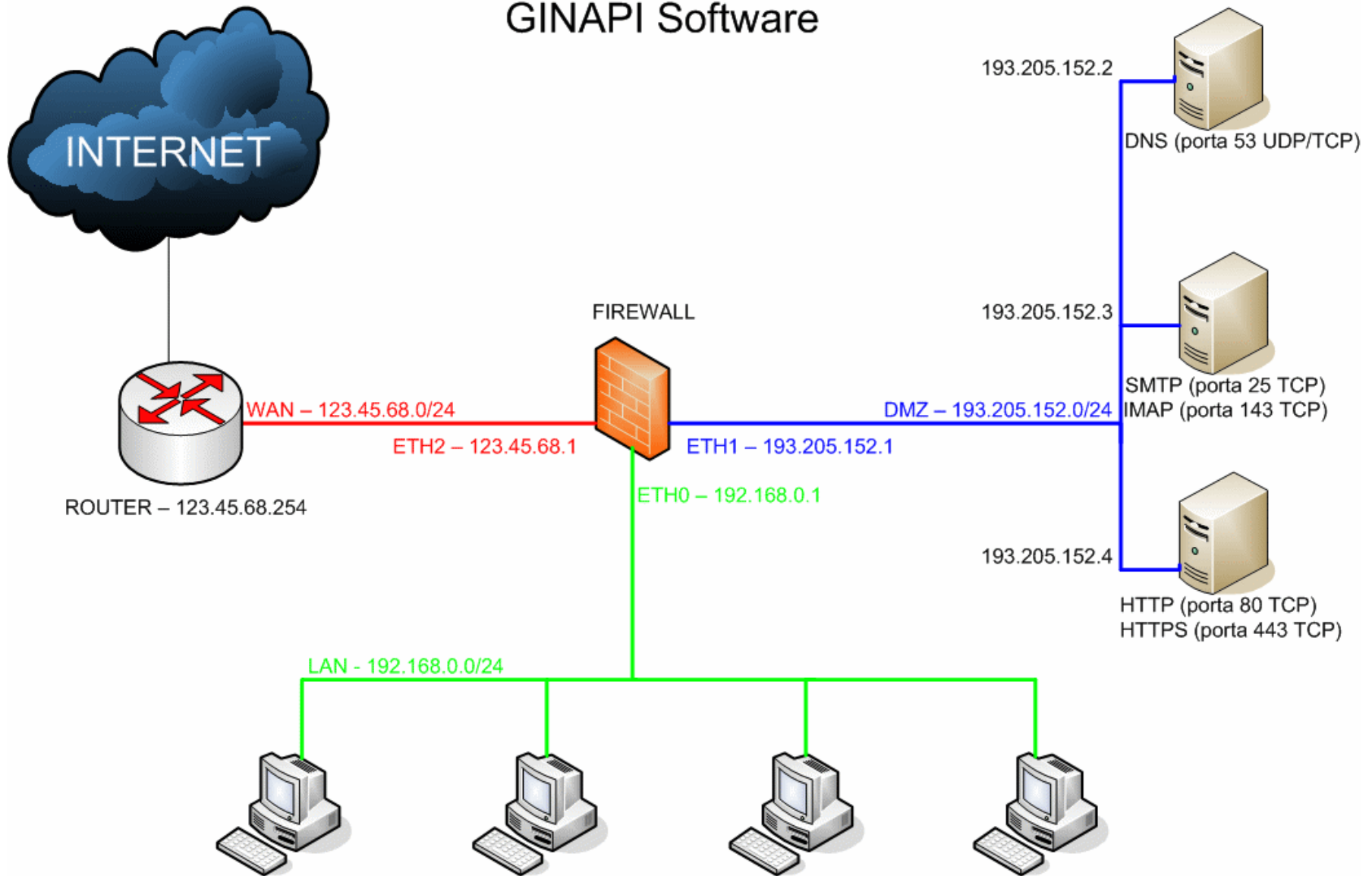
LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



GINAPI Software



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Con 3 reti (LAN, WAN e DMZ) il firewall ha a che fare con 8 diversi tipi di traffico:

- Traffico in ingresso al firewall (tabella filter, catena INPUT)
- Traffico in uscita dal firewall (tabella filter, catena OUTPUT)

- Traffico LAN -> DMZ (tabella filter, catena FORWARD)
- Traffico LAN -> WAN (tabella filter, catena FORWARD)
- Traffico WAN -> DMZ (tabella filter, catena FORWARD)
- Traffico WAN -> LAN (tabella filter, catena FORWARD)
- Traffico DMZ -> LAN (tabella filter, catena FORWARD)
- Traffico DMZ -> WAN (tabella filter, catena FORWARD)

Il Security manager della ditta ha deciso che:

- Il firewall deve essere “trasparente” (si lavora solo in console)
- LAN -> DMZ DNS, SMTP, IMAP interni
- LAN -> WAN HTTP esterni
- WAN -> DMZ DNS, HTTP, HTTPS, SMTP interni
- WAN -> LAN no traffico - (a parte pacchetti risposta LAN -> WAN)
- DMZ -> LAN no traffico - (a parte pacchetti risposta LAN -> DMZ)
- DMZ -> WAN SMTP, DNS esterni + pacchetti risposta WAN -> DMZ
- Controlli antispoofing sulle schede di rete
- Tutto il resto del traffico DEVE essere bloccato

Politica di default delle catene

Ad ogni catena presente nelle tabelle di IpTables **deve** essere associata una policy di default la quale definisce il comportamento predefinito per i pacchetti di rete che non corrispondano a nessuna regola definita nella catena. Le policy più usate sono ACCEPT e DROP.

Due delle richieste erano:

“Tutto il resto del traffico DEVE essere bloccato”

“il firewall deve essere “trasparente””

La policy di default di tutte le catene sarà DROP (rifiuta il traffico)

```
iptables -t filter -P INPUT DROP
```

```
iptables -t filter -P OUTPUT DROP
```

```
iptables -t filter -P FORWARD DROP
```

Uno degli approcci più semplici nella gestione delle regole di firewalling è dirottare il traffico della catena FORWARD in diverse catene user-defined da gestire separatamente (soluzione divide-et-impera)

- Si creano 6 nuove catene (comando: `iptables -t filter -N`)
- Si dirotta, in funzione delle interfacce ethernet di provenienza e di destinazione del pacchetto (**importante**), parte del traffico della catena FORWARD, considerando la nuova catena come un target della regola
- Si associa a ciascuna delle catene una lista di regole di sicurezza.

Es traffico WAN -> LAN (schede di rete del FW eth2 -> eth0)

```
iptables -t filter -N wan_lan
```

```
iptables -t filter -A FORWARD -i eth2 -o eth0 -j wan_lan
```

```
iptables -t filter -A wan_lan filtering_rules.....
```

Traffico LAN -> DMZ

Creo la catena lan_dmz e dirotto parte del traffico di FORWARD

```
iptables -t filter -N lan_dmz
```

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -j lan_dmz
```

La richiesta era: “DNS, SMTP, IMAP interni”

```
iptables -t filter -A lan_dmz -p udp -d 193.205.152.2 --dport 53 -j ACCEPT
```

```
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.2 --dport 53 -j ACCEPT
```

```
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.3 --dport 25 -j ACCEPT
```

```
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.3 --dport 143 -j ACCEPT
```



Traffico LAN -> WAN

Creo la catena lan_wan e dirotto parte del traffico di FORWARD

```
iptables -t filter -N lan_wan
```

```
iptables -t filter -A FORWARD -i eth0 -o eth2 -j lan_wan
```

La richiesta era: “HTTP esterni”

```
iptables -t filter -A lan_wan -p tcp --dport 80 -j ACCEPT
```



Traffico WAN -> DMZ

Creo la catena wan_dmz e dirotto parte del traffico di FORWARD

```
iptables -t filter -N wan_dmz
```

```
iptables -t filter -A FORWARD -i eth2 -o eth1 -j wan_dmz
```

La richiesta era: “DNS, HTTP, HTTPS, SMTP interni”

```
iptables -t filter -A wan_dmz -p udp -d 193.205.152.2 --dport 53 -j ACCEPT
```

```
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.2 --dport 53 -j ACCEPT
```

```
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.4 --dport 80 -j ACCEPT
```

```
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.4 --dport 443 -j ACCEPT
```

```
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.3 --dport 25 -j ACCEPT
```



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Traffico WAN -> LAN (1)

Creo la catena wan_lan e dirotto parte del traffico di FORWARD

```
iptables -t filter -N wan_lan
```

```
iptables -t filter -A FORWARD -i eth2 -o eth0 -j wan_lan
```

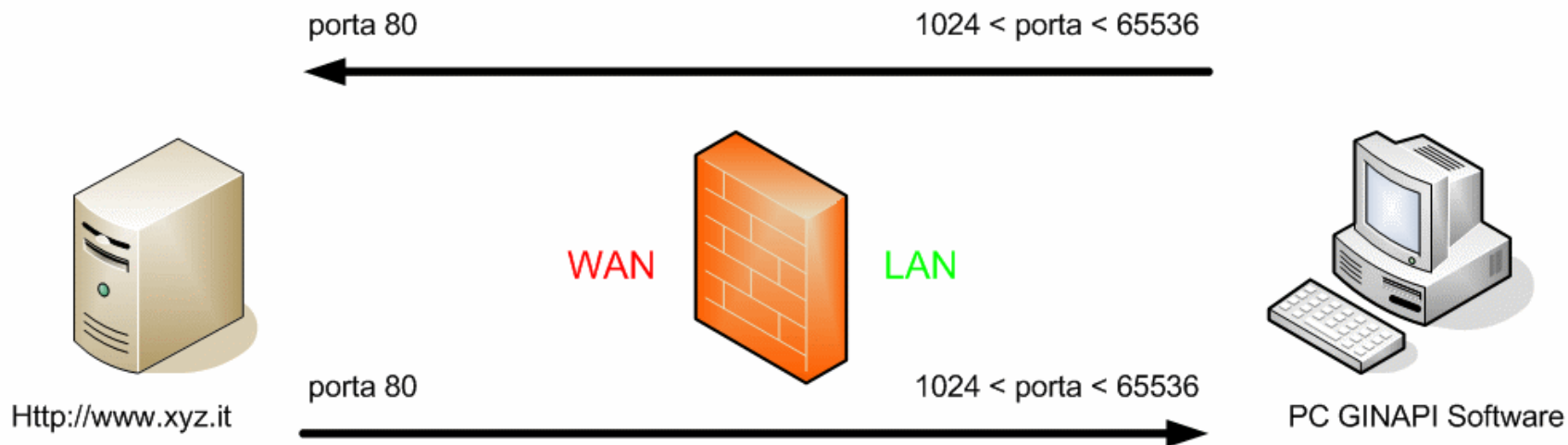
La richiesta era: “no traffico - (**a parte pacchetti risposta LAN -> WAN**)”

A causa della struttura del protocollo TCP/IP non posso sapere a priori quale porta TCP (compresa fra 1024 e 65535 - porte non privilegiate, effimere) la macchina della rete LAN ha usato per iniziare la connessione TCP.

Problema: Quali porte TCP lascio aperte ? Tutte quelle maggiori di 1023 ?

Un dipendente della GINAPI Software digita [Http://www.xyz.it](http://www.xyz.it)

LAN -> WAN: `iptables -t filter -A lan_wan -p tcp --dport 80 -j ACCEPT`



WAN -> LAN – Sol.stateless: `iptables -t filter -A wan_lan -p tcp --dports 1024:65535 -j ACCEPT`

Problemi di sicurezza: Troppe porte sempre aperte; Accetto anche connessioni ORIGINATE dall'esterno

WAN -> LAN – Sol.stateful: `iptables -t filter -A wan_lan -m state --state ESTABLISHED, RELATED -j ACCEPT`

Accetto solo i pacchetti di connessioni GIA' stabilite o relazionate a connessioni GIA' stabilite (es. ICMP)

Connection tracking -> IpTables stateful

Il connection tracking è il meccanismo che permette ad IpTables di tenere traccia delle connessioni stabilite indicando se un pacchetto in transito appartenga o meno (o sia relativo) ad una di esse.

Tale meccanismo, durante una connessione di rete, provvede a registrare le informazioni più importanti della connessione stessa; con queste informazioni è quindi possibile classificare i pacchetti in base al tipo di connessione:

- **NEW** - Pacchetti che stabiliscono una nuova connessione
- **ESTABLISHED** - Pacchetti di una connessione GIA' esistente
- **RELATED** - Pacchetti di una connessione GIA' esistente anche se non ne fanno parte direttamente (es. ICMP di errore)
- **INVALID** - Pacchetti "sospetti" non appartenenti a nessuna connessione in corso (in generale da rifiutare)



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Traffico WAN -> LAN (2)

Creo la catena wan_lan e dirotto parte del traffico di FORWARD

```
iptables -t filter -N wan_lan
```

```
iptables -t filter -A FORWARD -i eth2 -o eth0 -j wan_lan
```

Faccio passare SOLO i pacchetti di rete di connessioni GIA' stabilite oppure i pacchetti rete di connessioni attinenti a connessioni GIA' stabilite

```
iptables -t filter -A wan_lan -m state --state ESTABLISHED, RELATED -j ACCEPT
```



Traffico DMZ -> LAN

Creo la catena lan_dmz e dirotto parte del traffico di FORWARD

```
iptables -t filter -N dmz_lan
```

```
iptables -t filter -A FORWARD -i eth1 -o eth0 -j dmz_lan
```

La richiesta era: “no traffico - (a parte pacchetti risposta LAN -> DMZ)”

```
iptables -t filter -A dmz_lan -m state --state ESTABLISHED, RELATED -j ACCEPT
```



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Traffico DMZ -> WAN

Creo la catena dmz_wan e dirotto parte del traffico di FORWARD

```
iptables -t filter -N dmz_wan
```

```
iptables -t filter -A FORWARD -i eth1 -o eth2 -j dmz_wan
```

La richiesta era: “SMTP, DNS esterni + pacchetti risposta WAN -> DMZ”

```
iptables -t filter -A dmz_wan -p tcp --dport 25 -j ACCEPT
```

```
iptables -t filter -A dmz_wan -p udp --dport 53 -j ACCEPT
```

```
iptables -t filter -A dmz_wan -p tcp --dport 53 -j ACCEPT
```

```
iptables -t filter -A dmz_wan -m state --state ESTABLISHED, RELATED -j ACCEPT
```



Antispoofing

E' buona norma implementare, mediante iptables, l'antispoofing (**come prime regole di ogni catena**).

Le tre reti della GINAPI Software sono: LAN 192.168.0.0/24, DMZ 193.205.152.0/24, WAN 123.45.68.0/24

```
iptables -t filter -A lan_dmz -s ! 192.168.0.0/24 -j DROP
```

```
iptables -t filter -A lan_wan -s ! 192.168.0.0/24 -j DROP
```

```
iptables -t filter -A dmz_wan -s ! 193.205.152.0/24 -j DROP
```

```
iptables -t filter -A dmz_lan -s ! 193.205.152.0/24 -j DROP
```

```
iptables -t filter -A wan_dmz -s 192.168.0.0/24 -j DROP
```

```
iptables -t filter -A wan_dmz -s 193.205.152.0/24 -j DROP
```

```
iptables -t filter -A wan_lan -s 192.168.0.0/24 -j DROP
```

```
iptables -t filter -A wan_lan -s 193.205.152.0/24 -j DROP
```



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



IP forwarding

A questo punto, una volta configurato la parte di sicurezza (filtering) è necessario attivare l'opzione di IP forwarding nel kernel Linux della macchina che farà da firewall.

Tramite l'IP forwarding il firewall si comporterà da router instradando correttamente, secondo la tabella di routing, il pacchetti di rete fra le sottoreti di cui e' composta la GINAPI Software (LAN, WAN, DMZ)

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

Rete privata/problema traffico LAN -> WAN

La rete LAN è privata (192.168.0.0/24). Tale gruppo di indirizzi IP è riservato a macchine **NON** connesse direttamente ad Internet (RFC 1918).

I router “*dovrebbero*” distruggere i pacchetti privati.

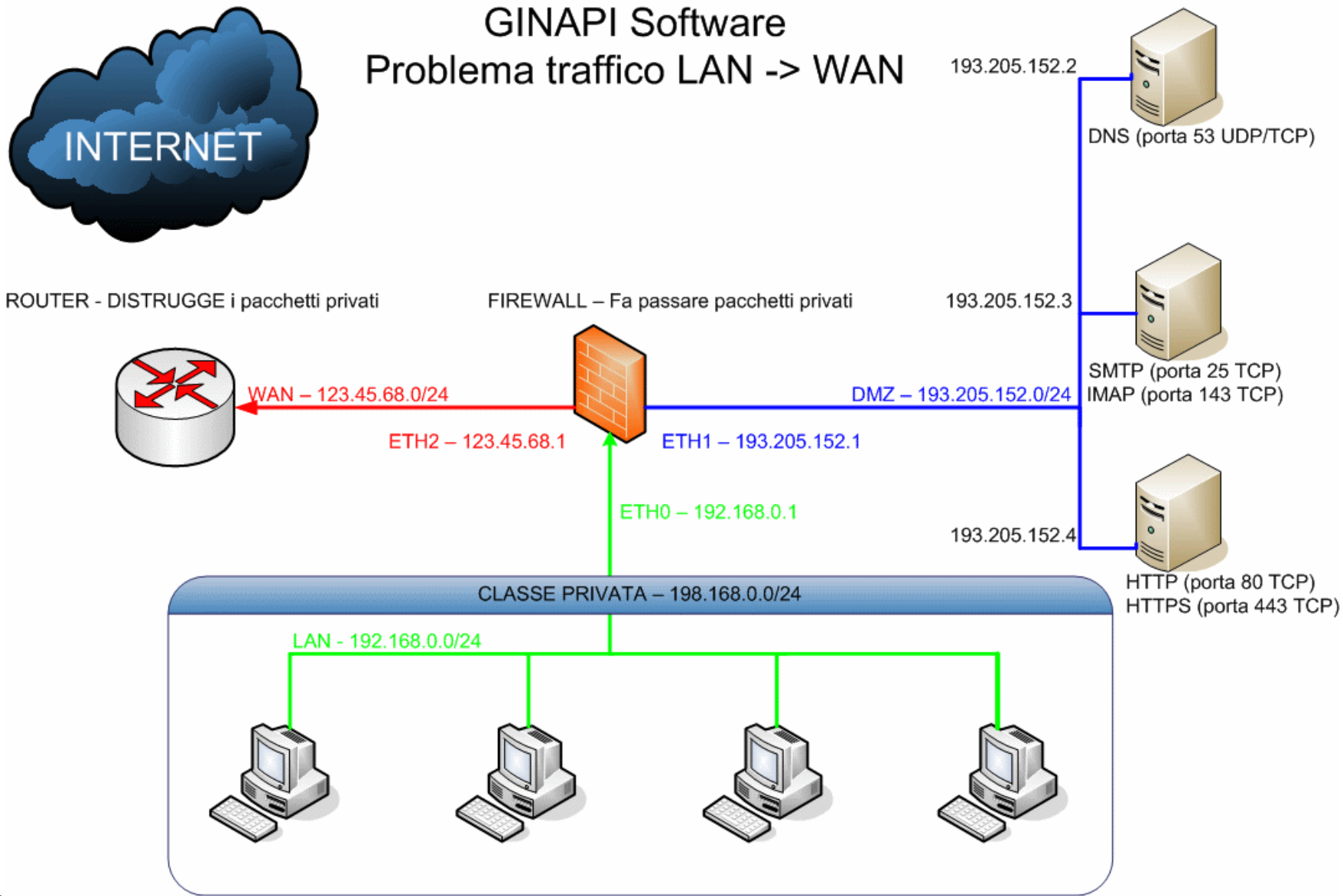
Vantaggi uso indirizzi IP privati:

- Non sono raggiungibili dall'esterno e quindi più sicuri
- Non servono tanti IP pubblici (costosi) quante sono le macchine possedute

Svantaggi uso indirizzi IP privati:

Problema: come possono navigare in internet dalle macchine LAN ?

GINAPI Software Problema traffico LAN -> WAN



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Network Address Translation/IP-Mascherading (1)

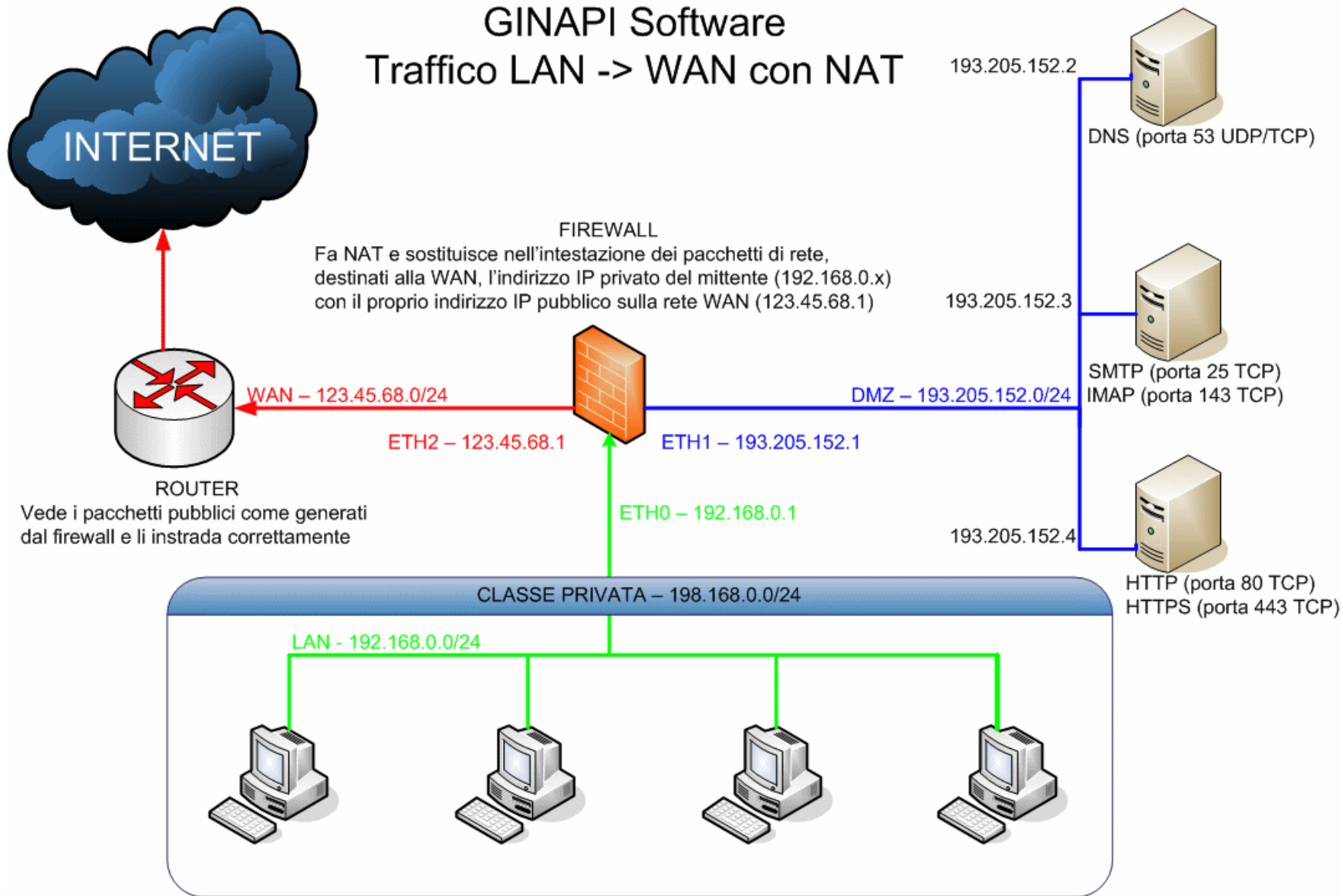
Meccanismo che consente di manipolare le intestazioni dei pacchetti IP in transito (dal firewall).

- Destination NAT (DNAT) modifica dell'indirizzo IP di destinazione del pacchetto (tabella nat, catena PREROUTING)
- Source NAT (SNAT) modifica dell'indirizzo IP sorgente del pacchetto (tabella nat, catena POSTROUTING - **funziona solo con IP statici**)
- **MASQUERADE** consente di cambiare l'indirizzo IP sorgente di un pacchetto IP **con quello della macchina su cui gira l'IpTables** (tabella nat, catena POSTROUTING - **funziona sia con IP statici che dinamici**)
- REDIRECT modifica la **porta** di destinazione di un pacchetto IP (tabella nat, catena PREROUTING)

Network Address Translation/IP-Mascherading (2)

Uso la tabella nat di IpTables e la sua catena predefinita POSTROUTING per mascherare (NAT) il **solo** traffico proveniente dalla rete LAN (192.168.0.x) e destinato ad internet (WAN) via scheda eth2

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth2 -j MASQUERADE
```



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Mettiamo i pezzi insieme (1)

```
#  
# Attivazione ROUTING fra schede di rete  
#  
echo 1 > /proc/sys/net/ipv4/ip_forward  
#  
#=====  
#  
# Tabella Filter  
#  
#=====  
#  
# Politiche di default delle catene della tabella filter  
#  
iptables -t filter -P INPUT DROP  
iptables -t filter -P OUTPUT DROP  
iptables -t filter -P FORWARD DROP  
#  
# Catene INPUT e OUTPUT  
# NON hanno regole perche' non si vuole traffico da/per firewall  
#
```



Mettiamo i pezzi insieme (2)

```
#  
# Catena FORWARD  
#  
# Genero le 6 catene user-defined  
#  
iptables -t filter -N lan_dmz  
iptables -t filter -N lan_wan  
iptables -t filter -N wan_dmz  
iptables -t filter -N wan_lan  
iptables -t filter -N dmz_lan  
iptables -t filter -N dmz_wan  
#  
# In base alle schede di rete sorgente e destinazione  
# dirotto il traffico FORWARD alle 6 catene appena create  
#  
iptables -t filter -A FORWARD -i eth0 -o eth1 -j lan_dmz  
iptables -t filter -A FORWARD -i eth0 -o eth2 -j lan_wan  
iptables -t filter -A FORWARD -i eth2 -o eth1 -j wan_dmz  
iptables -t filter -A FORWARD -i eth2 -o eth0 -j wan_lan  
iptables -t filter -A FORWARD -i eth1 -o eth0 -j dmz_lan  
iptables -t filter -A FORWARD -i eth1 -o eth2 -j dmz_wan
```



Mettiamo i pezzi insieme (3)

```
#  
# Catena lan_dmz (antispoofing e regole)  
#  
iptables -t filter -A lan_dmz -s ! 192.168.0.0/24 -j DROP  
iptables -t filter -A lan_dmz -p udp -d 193.205.152.2 --dport 53 -j ACCEPT  
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.2 --dport 53 -j ACCEPT  
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.3 --dport 25 -j ACCEPT  
iptables -t filter -A lan_dmz -p tcp -d 193.205.152.3 --dport 143 -j ACCEPT  
#  
# Catena lan_wan (antispoofing e regole)  
#  
iptables -t filter -A lan_wan -s ! 192.168.0.0/24 -j DROP  
iptables -t filter -A lan_wan -p tcp --dport 80 -j ACCEPT
```

Mettiamo i pezzi insieme (4)

```
#  
# Catena wan_dmz (antispoofing e regole)  
#  
iptables -t filter -A wan_dmz -s 192.168.0.0/24 -j DROP  
  
iptables -t filter -A wan_dmz -s 193.205.152.0/24 -j DROP  
iptables -t filter -A wan_dmz -p udp -d 193.205.152.2 --dport 53 -j ACCEPT  
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.2 --dport 53 -j ACCEPT  
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.4 --dport 80 -j ACCEPT  
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.4 --dport 443 -j ACCEPT  
iptables -t filter -A wan_dmz -p tcp -d 193.205.152.3 --dport 25 -j ACCEPT  
#  
# Catena wan_lan (antispoofing e regole)  
#  
iptables -t filter -A wan_lan -s 192.168.0.0/24 -j DROP  
  
iptables -t filter -A wan_lan -s 193.205.152.0/24 -j DROP  
iptables -t filter -A wan_lan -m state --state ESTABLISHED, RELATED -j ACCEPT
```



Mettiamo i pezzi insieme (5)

```
#
# Catena dmz_lan (antispoofing e regole)
#
iptables -t filter -A dmz_lan -s ! 193.205.152.0/24 -j DROP
iptables -t filter -A dmz_lan -m state --state ESTABLISHED, RELATED -j ACCEPT
#
# Catena dmz_wan (antispoofing e regole)
#
iptables -t filter -A dmz_wan -s ! 193.205.152.0/24 -j DROP
iptables -t filter -A dmz_wan -p tcp --dport 25 -j ACCEPT
iptables -t filter -A dmz_wan -p udp --dport 53 -j ACCEPT
iptables -t filter -A dmz_wan -p tcp --dport 53 -j ACCEPT
iptables -t filter -A dmz_wan -m state --state ESTABLISHED, RELATED -j ACCEPT
#
#=====
#
# Tabella Nat
#
#=====
#
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth2 -j MASQUERADE
```



LUGGE

Linux Users Group Genova

ASSOCIAZIONE SENZA
SCOPO DI LUCRO



Link

- <http://www.netfilter.org>
sito gruppo Netfilter/IpTables
- <http://www.commedia.it/ccontavalli/docs-it/iptables/iptables4dummies>
ottimo tutorial sulla configurazione di IpTables