Lezione 10: Packet Filtering: Netfilter, Firewall, NAT

Claudio Ardagna, Patrizio Tufarolo – Università degli Studi di Milano

Insegnamento di Laboratorio di Reti di Calcolatori

Introduzione

- In questa lezione affronteremo argomenti che abbracciano i livelli più alti dello stack ISO/OSI, dal livello 3 (livello di rete) al livello 7 (livello applicativo)
- Tratteremo l'argomento del packet filtering, andando a studiare in particolar modo l'implementazione del modulo «netfilter» di Linux, configurabile tramite il software IPTables, che fornisce funzionalità di filtraggio, logging e manipolazione dei pacchetti
- Ci concentreremo infine sulle funzionalità di NAT (Network Address Translation)

Terminologia - 1

- ISO/OSI (Open System Interconnection)
  - Standard de iure che organizza l'architettura di una rete di calcolatori in una struttura composta da 7 livelli (stack di rete)
- Livello di rete
  - Livello dello stack ISO/OSI che permette di interconnettere reti eterogenee. Riceve dei *segmenti* dal soprastante livello di trasporto e produce dei *pacchetti* che verranno passati al livello datalink, sottostante
- Livello di trasporto
  - Livello dello stack ISO/OSI che permette il trasporto di informazioni in unità chiamate *segmenti*. Il suo compito è quello di fornire un meccanismo di trasporto delle informazioni affidabile, per il corretto funzionamento del livello di sessione

Terminologia - 2

- Livello applicativo
  - Livello 7 dello Stack ISO/OSI, all'interno del quale sono collocate applicazioni e servizi di rete
- Router
  - Dispositivo per l'interconnessione di reti a livello 3 che stabilisce un percorso logico di comunicazione costituito da *hop* e costruito sulla base di una *tabella di instradamento*
- NAT
  - Network Address Translation: tecnica che consiste nel manipolare i pacchetti a Livello 3 modificando gli indirizzi IP sorgente e destinazione del pacchetto, memorizzando opportunamente le traduzioni effettuate in una tabella

Terminologia - 3

- D-NAT
  - Destination Nat, l'indirizzo destinazione del pacchetto IP viene sostituito con un altro indirizzo destinazione
- S-NAT
  - Source NAT, l'indirizzo sorgente del pacchetto IP viene sostituito con un altro indirizzo sorgente
- Masquerading o NAT Dinamico
  - Caso particolare di Source NAT in cui le connessioni generate da un insieme di host, vengono presentate come provenienti da un unico indirizzo IP
  - Vengono modificate anche le porte sorgente e destinazione
  - Internet non è una rete point-to-point!
- NAT Statico o NAT 1 a 1 – Caso particolare di Masquerading nel quale un host mantiene l'indirizzo privato e l'indirizzo pubblico viene totalmente mappato su di lui

Introduzione a netfilter

- Netfilter è il componente del kernel Linux che permette l'intercettazione e la manipolazione di pacchetti
- Implementa funzionalità di rete avanzate come il filtraggio stateful del traffico di rete e la NAT
- Può essere esteso con moduli del kernel, per implementare ulteriori funzionalità di inspection e manipolazione dei pacchetti
- È gestibile tramite i comandi *iptables* (per IPv4) e *ip6tables* (per IPv6)
- Supporta la deep packet inspection, per fare analisi sul protocollo anche a livello applicativo (*l7_filters*)

Netfilter – funzionamento

- Il funzionamento di netfilter è incentrato sull'utilizzo di tabelle. Queste sono implementate a livello kernel. Netfilter ha 4 tabelle (filter, nat, mangle e raw).
- Ogni tabella contiene delle *chain* (catene), che sono delle vere proprie Access Control List e contengono a loro volta delle *rules* (regole). È possibile aggiungere in user-space delle chain per dare un ordine logico alle regole.
- Ogni regola è divisa in due parti:
  - Filtro – proprietà che un pacchetto deve avere affinché la regola sia valida
  - Target – azione da compiere nel caso il pacchetto corrisponda alle proprietà impostate nel filtro (matching)

Netfilter – tabella filter

- Tabella delle regole di filtraggio dei pacchetti. Permette di scegliere quali bloccare e quali far passare.
- Ha 3 chain di base:
  - Input – tutti i pacchetti in arrivo destinati al sistema passano per questa catena
  - Forward – tutti i pacchetti in arrivo (non generati dal sistema stesso) destinati ad un altro sistema passano per questa catena
    - Ciò è possibile se il sistema è un Router, ovvero ha il flag di ip forwarding abilitato
  - Output – tutti i pacchetti generati dal sistema passano per questa catena

Netfilter – tabella nat

- Tabella delle regole di traduzione degli indirizzi.
  Nel caso il protocollo sia session-oriented, solo il primo pacchetto di una sessione passa per questa tabella, la decisione presa vale per tutti gli altri pacchetti appartenenti alla stessa sessione.
- Ha 3 chain di base:
  - Prerouting – in questa catena passano i pacchetti in entrata, prima che venga presa la decisione di instradamento. È usata per fare DNAT
  - Postrouting – in questa catena passano i pacchetti in uscita, dopo che è stata presa la decisione di instradamento. È usata per fare SNAT
  - Output – permette di effettuare DNAT sui pacchetti generati localmente

Netfilter – tabella mangle

- Permette di fare modifiche alle opzioni dei pacchetti e di applicare politiche avanzate (es. QoS)
- Ha le seguenti chain:
  - Prerouting – Esamina tutti i pacchetti in entrata nel sistema, prima che venga consultata la tabella di routing
  - Input – Esamina tutti i pacchetti in entrata nel sistema destinati al sistema stesso
  - Forward – Esamina tutti i pacchetti in entrata nel sistema ma destinati a un altro sistema
  - Output – Esamina tutti i pacchetti generati dal sistema
  - Postrouting – Esamina tutti i pacchetti, dopo che è stata effettuata la decisione di routing e prima di inoltrare effettivamente il pacchetto al sistema destinatario

Netfilter – tabella raw

- Permette di evitare il tracciamento della connessione, qualora si desideri avere un filtraggio stateless.
- Ha due sole chain:
  - Prerouting
  - Output

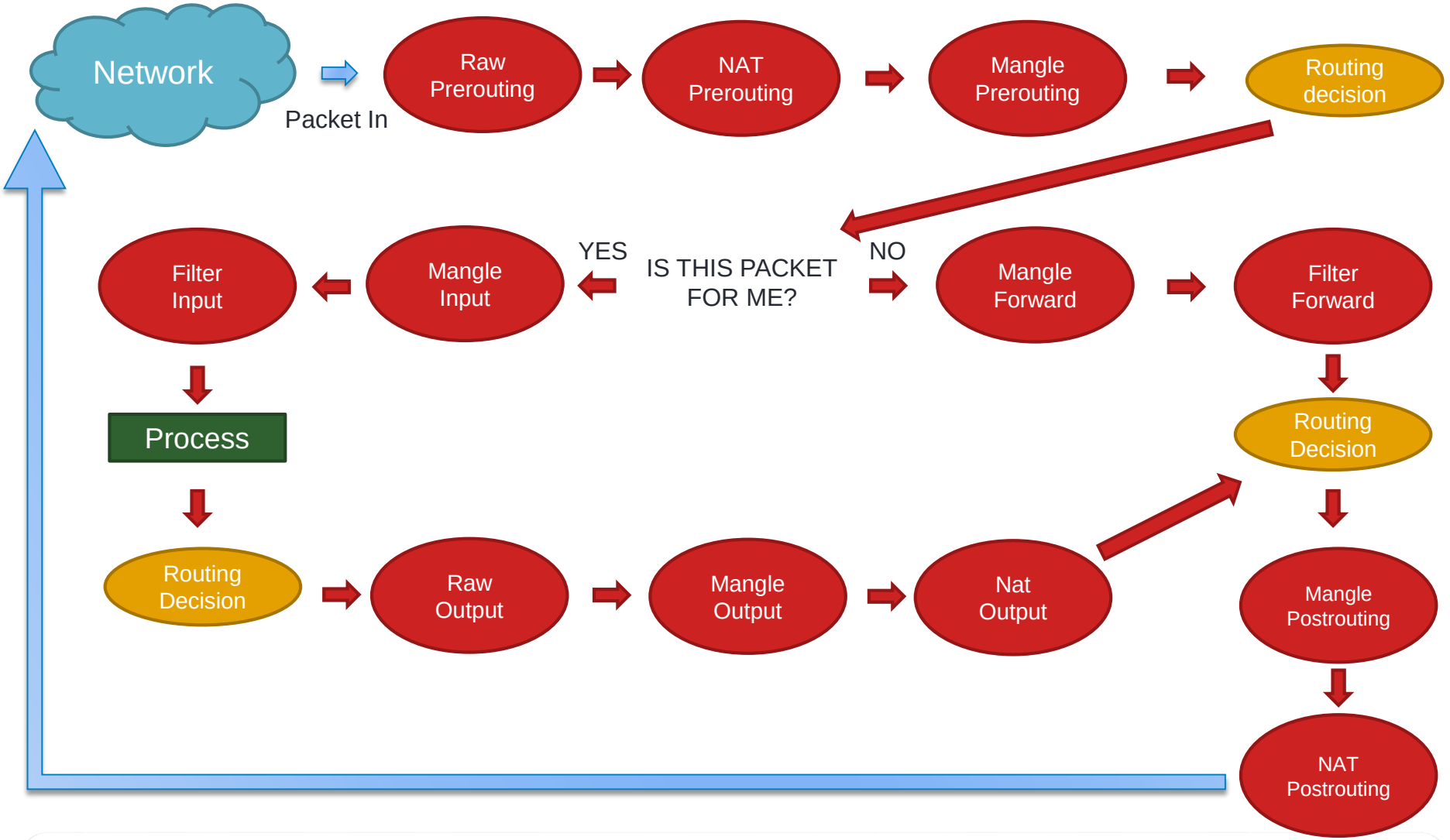
Tabelle Netfilter: esempio tabella filter

```
Chain INPUT (policy DROP)
target     prot opt source               destination
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
DROP       all  -f  0.0.0.0/0            0.0.0.0/0
DROP       all  -f  0.0.0.0/0            0.0.0.0/0
DROP       all  --  [redacted]           0.0.0.0/0
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
DROP       tcp  --  [redacted]           0.0.0.0/0           tcp dpt:80
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  [redacted]           [redacted]          icmptype 8
ACCEPT     icmp --  [redacted]           [redacted]          icmptype 8
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:25 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:22 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:80 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:443 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:465 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:587 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:993 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:995 state NEW,RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            [redacted]          tcp dpt:3389 state NEW,RELATED,ESTABLISHED
ACCEPT     all  --  0.0.0.0/0            [redacted]          state RELATED,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            [redacted]
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
```

Flusso del pacchetto in un sistema Linux



Netfilter - targets

- I targets (obiettivi) sono le azioni da compiere su un pacchetto.
- Un obiettivo può essere:
  - Una chain – per far gestire il pacchetto a una catena specifica, definita manualmente
  - Uno degli obiettivi predefiniti
    - ACCEPT → lascia passare il pacchetto
    - DROP, REJECT → scarta/rifiuta il pacchetto
    - QUEUE → mette il pacchetto in una coda, che può essere dedicata a una specifica applicazione
    - RETURN → ha lo stesso effetto di raggiungere la fine di una chain, agisce ricorsivamente come una chiamata a funzione
    - LOG → logga il pacchetto sul demone di logging di sistema (es. syslog)
    - DNAT → esegue il Destination NAT
    - SNAT → esegue il Source NAT
    - MASQUERADE → esegue il Masquerading
  - Un obiettivo definito da un'estensione
    - Esempio: NFLOG, netfilter log, logging avanzato tramite interfaccia di rete

Commandistica IPTables

- La commandistica IPTables è reperibile, come al solito, nel relativo manuale (*man iptables*)
- Ricordiamo, in ogni caso, alcuni comandi utili:
  - `iptables -l [-t tabella]`
    - Restituisce la lista delle regole, divise per chain, per la tabella specificata (opzionale). Se non è specificata alcuna tabella, restituisce la lista delle regole per la tabella filter.
      - Es: `iptables -l -t nat`
      - `iptables -l -t [tabella] --line-numbers`
        stampa anche i numeri di linea
  - `iptables -v`
    - Modalità *verbose*, restituisce l'output con maggiore dettaglio (ad es. indicando i contatori dei match per una data regola/chain)
  - `iptables -n`
    - Come in quasi ogni comando Unix relativo al networking, l'argomento «n» viene utilizzato per evitare la risoluzione degli indirizzi/numero di porta in nomi

Commandistica IPTables – 2

- Policy di default per una chain:
  - `iptables [-t tabella] -P <chain> <target>`
- Flush delle regole inserite
  - `iptables [-t tabella] –F`
- Inserire una regola in una chain (in testa o in una determinata posizione)
  - `iptables [-t tabella] –I <chain> [posizione] <filtro> -j <target>`
- Appendere una regola a una chain (in coda)
  - `iptables [-t tabella] -A <chain> <filtro> -j  <target>`
- Rimuovere una regola da una chain
  - `iptables [-t tabella] –D <chain> <numerolinea>`
  - `iptables [-t tabella] –D <chain> <filtro> -j <target>`

Scrivere una regola per netfilter

- Ogni regola ha un filtro e un target
- Il filtro è generalmente composto da questi flag, specificati e combinati opportunamente a seconda del significato della regola
  - `-s <host/rete sorgente>`
  - `-d <host/rete destinazione>`
  - `-i <interfaccia>`
  - `-p <protocollo>` → Protocollo di livello 4 - ICMP, TCP, UDP
    - `--sport <porta sorgente>` → solo per protocollo TCP o UDP
    - `--dport <porta destinazione>` → solo per protocollo TCP o UDP
  - `-m <match>` → modulo che testa il matching di una specifica proprietà
- Il target è preceduto dall'argomento `-j` (jump) e può essere uno di quelli visti precedentemente

Esercizio 1 – Filtraggio di una porta

- Realizzare una topologia IMUNES composta da:
  - PC ←→ Router ←→ Host
- Sull'host mettersi in ascolto sulla porta 8080 TCP con netcat
  - `nc -l -p 8080`
- Aprire netcat sul PC verso l'host e scrivere qualcosa
  - `nc <iphost> 8080`
- Scrivere una regola iptables sull'host che blocchi il traffico sulla porta TCP 8080, verificare che netcat smetta di funzionare
- Dopo aver fatto il flush della regola sull'host, scrivere una regola iptables sul router, scegliendo opportunamente tabella e chain, che blocchi il traffico in transito sulla porta TCP 8080 diretto all'host, verificare con netcat

Esercizio 1 - Soluzione

- Regola 1 – SULL'HOST:
  - `iptables –t filter -I INPUT -p tcp --dport 8080 -j DROP`
- Flush delle regole
  - `iptables –F`
- Regola 2:
  - `iptables –t filter –I FORWARD -p tcp --dport 8080 -j DROP`

Scenario di NAT - DNAT

140.150.160.170

159.149.70.100

172.25.27.21

Effettua una connessione verso 159.149.70.100:8080

Tramite D-NAT il router permette di contattare 172.25.27.21:80 cambiando l'indirizzo e la porta di destinazione del pacchetto

Scenario di NAT - SNAT

192.168.0.2

192.168.0.1

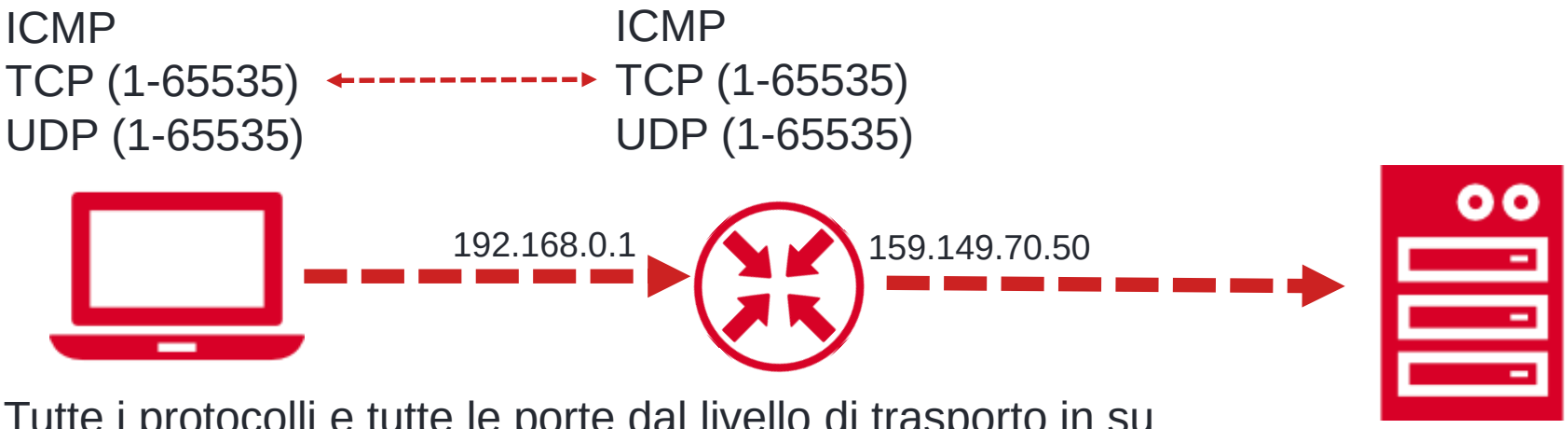
159.149.70.50

8.8.8.8

Il client ha un indirizzo IP privato, è dietro a un router e non può essere contattato dall'esterno (le risposte non possono tornare indietro, perché il router non ha una rotta verso 192.168.0.2)

Tramite SNAT il router imposta il suo indirizzo sorgente. Quando il server 8.8.8.8 riceverà il pacchetto vedrà 159.149.70.50 come sorgente. Tramite la tabella di NAT, il router recapiterà le risposte opportunamente.

Scenario di NAT – NAT 1-1



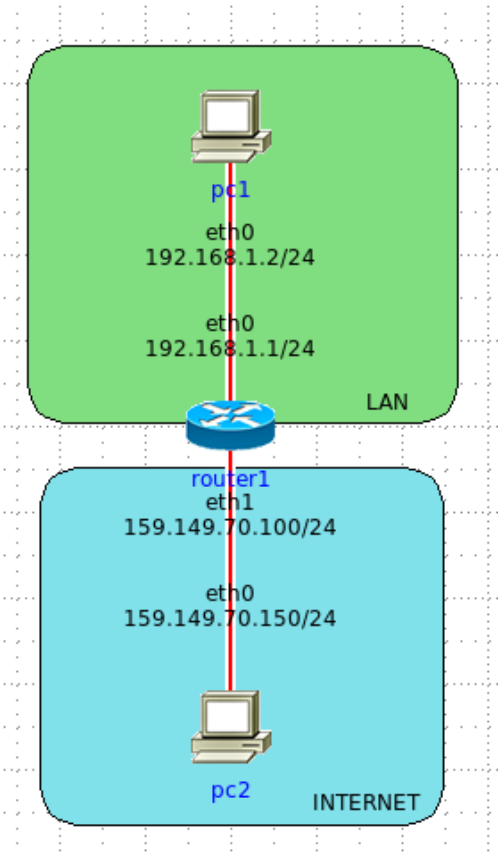
Tutte i protocolli e tutte le porte dal livello di trasporto in su del computer sono mappate sul router. Il computer appare come connesso direttamente a internet. Egli esce su internet con SNAT (o MASQUERADING), e tutte le porte sono mappate su di lui tramite DNAT.

Esercizio 2 – NAT 1 a 1

- Avendo a disposizione un IP pubblico su internet, assegnato al vostro router, avete deciso di mapparne tutte le porte di tutti i protocolli sul vostro PC, tramite NAT 1 a 1.
- Simulate la situazione con IMUNES e configurate il NAT in modo opportuno.
- Avviare tcpdump sul router, e sul PC. Effettuare un Ping verso il pc su internet

Esercizio 2 – Soluzione

Topologia



Comandi

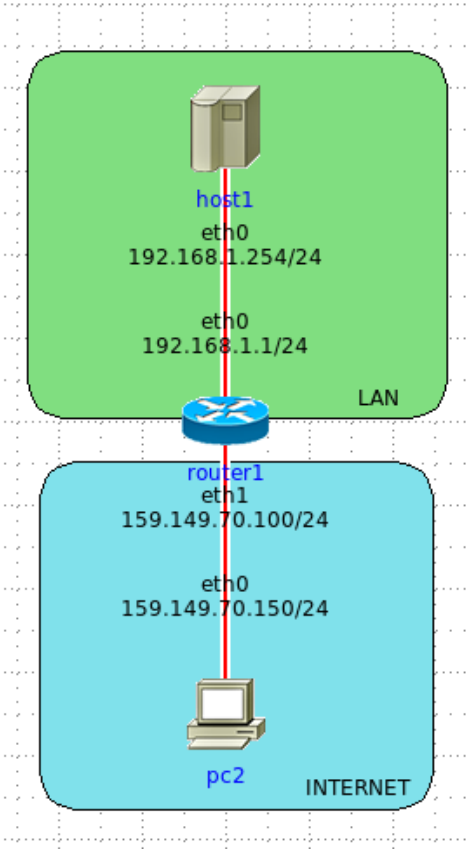
- PC1
  - `ip addr add 192.168.1.2/24 dev eth0`
  - `ip route add 0.0.0.0/0 via \ 192.168.1.1`
- Router1
  - `ip addr add 192.168.1.1/24 dev eth0`
  - `ip addr add 159.149.70.100/24 dev \ eth1`
- PC2
  - `ip addr add 159.149.70.150/24 dev \ eth0`
- Regole di NAT su Router1
  - `iptables -t nat -i POSTROUTING –j \ SNAT --to-source 159.149.70.100`
  - `iptables -t nat -i PREROUTING –j \ DNAT --to-destination 192.168.1.2`

Esercizio 3 – Portmapping

- Nella rete di casa vostra, avete un router e un web server che ospita il vostro blog. Il web server è in ascolto sulla porta 80 (`service lighttpd start`)
- Il web server è in grado di comunicare con la rete internet tramite il suo default gateway, ovvero il router
- Il web server, ovviamente, non è direttamente accessibile sulla rete internet. Gli host appartenenti ad internet NON HANNO la rotta per raggiungere il pc
- La subnet che state utilizzando è la 192.168.1.0/24, il gateway è all'indirizzo 192.168.1.1, il computer ospitante il blog è all'indirizzo 192.168.1.254, assegnato staticamente
- Simulate questa situazione su IMUNES, scegliendo IP non appartenenti a pool privati per simulare il computer sulla rete internet, avendo l'accorgimento di rendere possibile la comunicazione tra il router e il computer su internet
- Configurate il router in modo che effettui SNAT sostituendo l'indirizzo sorgente dei pacchetti generati esclusivamente dal webserver con il suo indirizzo IP
- Rendete il webserver accessibile sull'IP del router tramite DNAT, alla porta 8080
- Provate a effettuare una richiesta con CURL verso la porta 8080 del router
  - `curl http://IP_PUBBLICO_ROUTER:8080`
- Provate a pingare il PC su internet dal web server, e osservate, sul router, quello che avviene con il NAT

Esercizio 2 – Soluzione

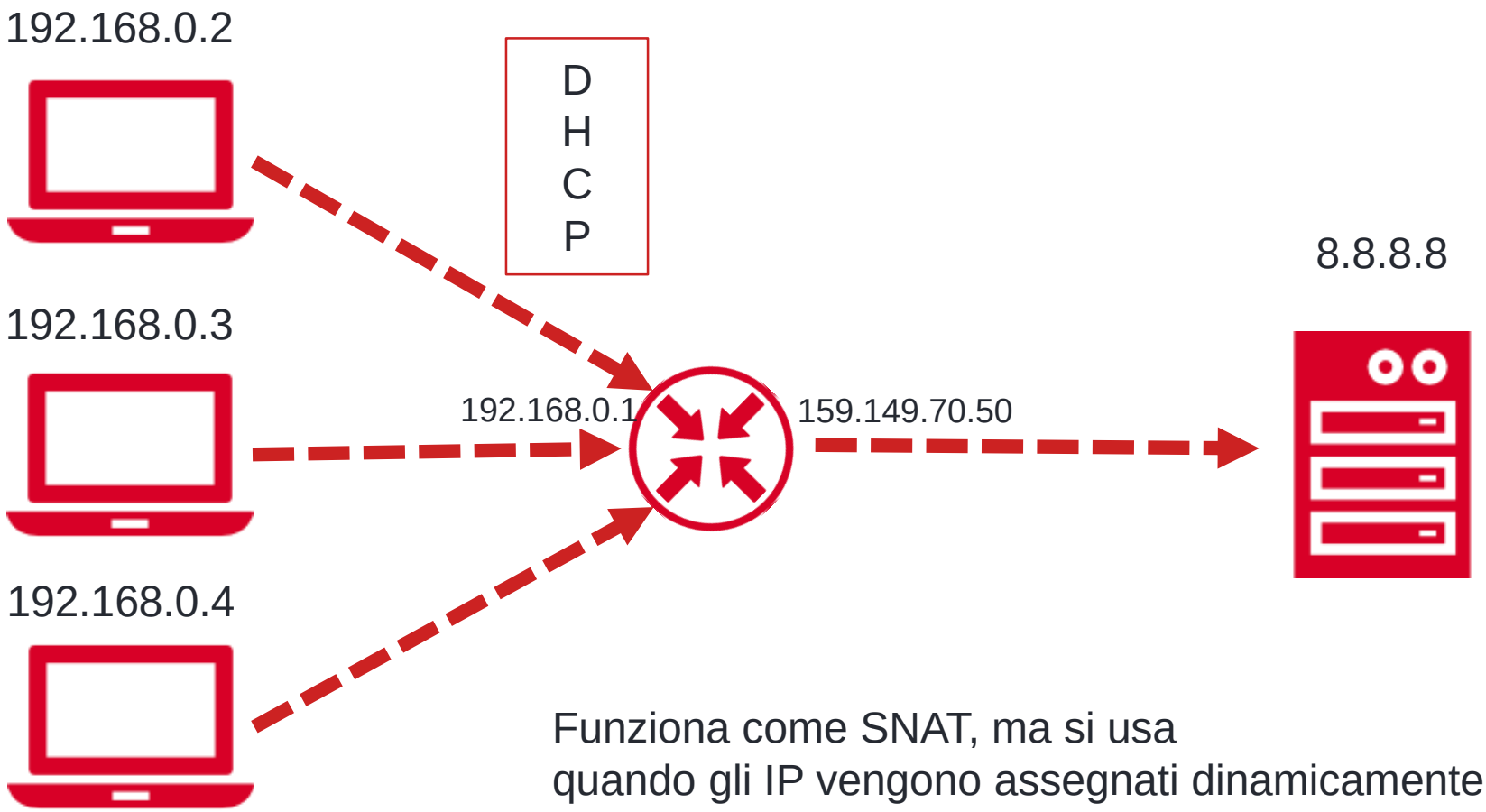
Topologia



Comandi

- PC1
  - `ip addr add 192.168.1.254/24 dev eth0`
  - `ip route add 0.0.0.0/0 via \ 192.168.1.1`
- Router1
  - `ip addr add 192.168.1.1/24 dev eth0`
  - `ip addr add 159.149.70.100/24 dev \ eth1`
- PC2
  - `ip addr add 159.149.70.150/24 dev \ eth0`
- Regole di NAT su Router1
  - `iptables -t nat -i POSTROUTING -j \ SNAT -s 192.168.1.254 --to-source 159.149.70.100`
  - `iptables -t nat -i PREROUTING -p tcp \ -d 159.149.70.100 --dport 8080 \ -j DNAT \ --to-destination 192.168.1.254:80`

Scenario di NAT - Masquerading

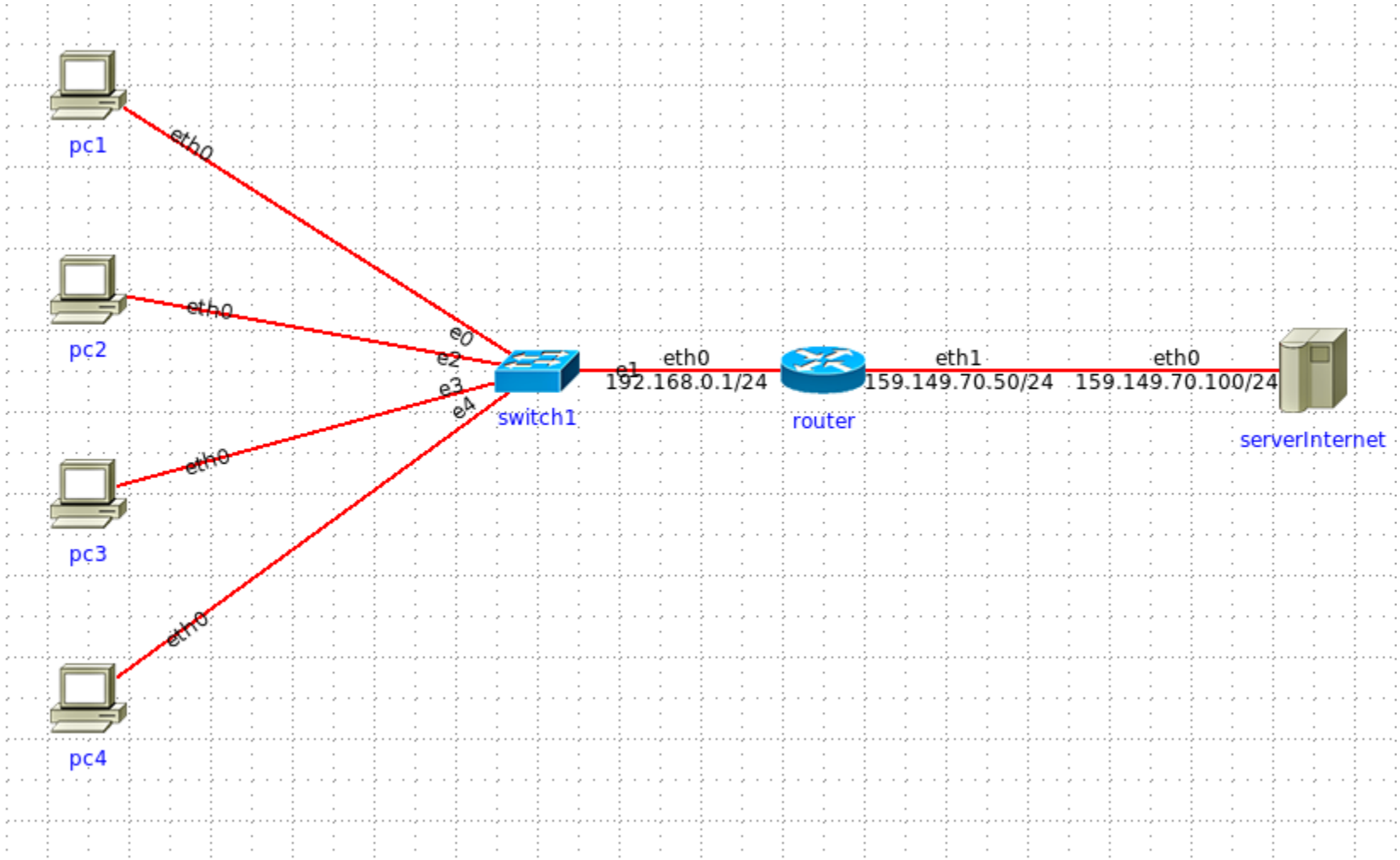


Funziona come SNAT, ma si usa
quando gli IP vengono assegnati dinamicamente

Esercizio 4 – Masquerading – Per casa

- Nella vostra rete di casa, il router è anche un DHCP server che assegna dinamicamente gli indirizzi ai dispositivi connessi, specificando – tramite DHCP option – se stesso come default gateway
- I dispositivi, tuttavia, devono potersi connettere a Internet
  Teoricamente parlando, non c'è nessun problema sul traffico in uscita: hanno una rotta…
- Come abbiamo ribadito più volte, però, i pc su internet non hanno una rotta verso i vostri dispositivi
- Simulate la situazione con IMUNES, e configurate il masquerading in modo opportuno

Esercizio 4 – Masquerading – Soluzione



Esercizio 4 – Masquerading – Soluzione

DHCP

```
cat << EOF > /etc/dhcp/dhcpd.conf
default-lease-time: 500;
max-lease-time: 7200;
subnet 192.168.0.0 netmask
255.255.255.0 {
   range 192.168.0.2 192.168.0.254;
   option routers 192.168.0.1;
}
EOF
```

Regola iptables

```
iptables -t nat -i POSTROUTING \
-s 192.168.0.0/24 -j MASQUERADE
```