Università di Verona, Facoltà di Scienze MM.FF.NN.

Insegnamento di Reti di Calcolatori

TCP e UDP, firewall e NAT

Davide Quaglia

Scopo di questa esercitazione è:
1) utilizzare Wireshark per studiare il traffico TCP e UDP;
2) imparare l'utilizzo del software per implementare regole di firewall e NAT su macchine Linux.

NOTA: considerevoli porzioni di questo documento sono tratte da Wikipedia che le mette a disposizione nel rispetto dei termini della GNU Free Documentation License.

1 Studio dell'evoluzione di una connessione TCP

Dopo aver configurato il sistema Linux come spiegato nella precedente esercitazione, si apra una shell e da lì si lanci Wireshark:

```
$> sudo wireshark
```

Si avvii una cattura sull'interfaccia esterna (eth1) e subito dopo si apra una seconda shell e si lanci:

```
$> telnet 209.85.227.99 80
```

Alla comparsa del promp "Escape character is ..." si scriva "ciao" e si batta INVIO.
Si blocchi la cattura e si imposti un filtro di visualizzazione "TCP and IP.addr == X.Y.Z.W" dove al posto delle lettere si mette l'indirizzo IP della propria interfaccia.

Selezionare un pacchetto TCP e prendere visione dei vari campi dell'header.

Analizzare, prendendo appunti su un foglio, l'andamento della sequenza di pacchetti TCP; si risponda alle seguenti domande:
1) Qual è il numero di sequenza iniziale in entrambe le direzioni ?
2) Si possono identificare le fasi di avvio e fine della connessione ?
3) Esiste una ben precisa relazione tra numeri di sequenza e numeri del campo acknowledge in pacchetti in viaggio in direzione opposta ?
4) Durante la connessione avviene l'aggiustamento della finestra di ricezione ? Chi manda l'aggiustamento ?

Avviare ora una nuova cattura e digitare da una shell il comando per lo scaricamento di un file tramite TCP

```
$> wget http://profs.sci.univr.it/~quaglia/temp/liv-fisico-2x.pdf
```

Si blocchi la cattura e si imposti un filtro di visualizzazione "TCP and IP.addr == X.Y.Z.W" dove al posto delle lettere si mette l'indirizzo IP della propria interfaccia. Si selezioni un pacchetto TCP che ha come sorgente il proprio IP. Si visualizzi l'andamento dei numeri di sequenza tramite la voce di menu "Statistics/TCP Stream Graph/Time-Sequence Graph (tcptrace)". Si selezioni un pacchetto TCP che ha come destinazione il proprio IP e si visualizzi nuovamente il grafico. Cosa si nota ?
Visualizzare, per entrambe le direzioni dei pacchetti, gli altri grafici che il menu "Statistics/TCP Stream Graph/" mette a disposizione.

2 Cattura di pacchetti UDP

Si avvii una cattura sull'interfaccia esterna (eth1) e subito dopo si apra una seconda shell e si lanci:

```
$> dig www.google.it
```

Si blocchi la cattura e si imposti un filtro di visualizzazione “UDP and IP.addr == X.Y.Z.W” dove al posto delle lettere si mette l'indirizzo IP della propria interfaccia.

Selezionare un pacchetto UDP e prendere visione dei vari campi dell'header. Che tipo di protocollo è citato nella colonna “Protocol” di Wireshark ?

3 Introduzione al sistema Netfilter del kernel di Linux

Netfilter è un componente del kernel del sistema operativo Linux, che permette l'intercettazione e manipolazione dei pacchetti che attraversano il calcolatore. Netfilter permette di realizzare alcune funzionalità di rete avanzate come la realizzazione di firewall basata sul filtraggio stateful dei pacchetti o configurazioni anche complesse di NAT, un sistema di sostituzione automatica degli indirizzi IP, tra cui la condivisione di un'unica connessione Internet tra diversi computer di una rete locale, o ancora la manipolazione dei pacchetti in transito.

Iptables è il programma che permette agli amministratori di sistema di configurare Netfilter, definendo le regole per i filtri di rete e il reindirizzamento NAT. Spesso con il termine Iptables ci si riferisce all'intera infrastruttura, incluso Netfilter.

Iptables è un componente standard di tutte le moderne distribuzioni di Linux. Esso fu introdotto nella versione principale del sistema operativo nel marzo del 2000, durante lo sviluppo della versione 2.4. Nella versione 2.2 si usava un sistema alternativo denominato ipchains, che a sua volta sostituì il sistema ipfwadm, usato nella versione 2.0.

Netfilter/Iptables trova applicazione sia in calcolatori che vengono usati come host (hanno una sola interfaccia di rete, e non inoltrano pacchetti da un'interfaccia ad un'altra) che per realizzare dei veri e propri router basati su Linux.

Con netfilter è possibile controllare il contenuto di ogni singolo pacchetto e definire le azioni da compiere in base alle sue caratteristiche. Ad esempio, si può definire una regola che impedisce la ricezione di pacchetti provenienti da un particolare indirizzo o che utilizzano una determinata porta per effettuare la connessione.

Il sistema Netfilter (Figura 2) è basato su regole raggruppate in *catene* (chain), a loro volta raggruppate in *tabelle* (tables). Ogni tabella definisce un tipo diverso di operazioni che è possibile effettuare sui pacchetti; ogni catena definisce come vengono trattati i pacchetti nelle diverse fasi della loro elaborazione.

Ogni regola è costituita da due parti: la *specifica delle caratteristiche* che un pacchetto deve avere affinché la regola stessa venga applicata (match) e una azione o *target*, che indica cosa fare quando il pacchetto rispetta le caratteristiche indicate. A ciascuna catena è anche associata una *politica di default*, che definisce come vengono trattati i pacchetti che non corrispondono ad alcuna regola.

Ogni pacchetto di rete che arriva o parte dal computer attraversa almeno una catena e ogni regola della catena controlla se il pacchetto ne rispetta la specifica. Se questo accade, il pacchetto seguirà il comportamento descritto nell'obiettivo della regola, e le regole successive della catena verranno

ignorate (a parte casi speciali). Se il pacchetto raggiunge la fine della catena senza essere processato da nessuna regola, la politica di default della catena determina cosa farne.

In ogni tabella esistono alcune catene predefinite, ma l'utente può crearne di nuove.

3.1 Le tabelle

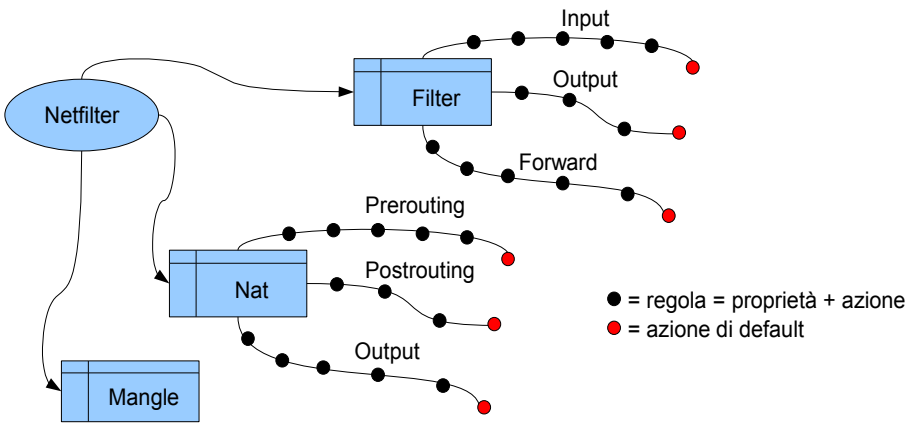

Figura 1. Tabelle, catene e regole.

Come mostrato in Figura 1, esistono tre tabelle prestabilite, ognuna delle quali contiene delle catene predefinite. Esiste anche la possibilità di creare altre tabelle. L'amministratore può creare e cancellare le catene in qualsiasi tabella. Inizialmente, tutte le catene sono vuote e hanno una politica che permette a tutti i pacchetti di passare senza essere bloccati o alterati in alcun modo, esse vanno poi modificate a seconda delle proprie esigenze. Le tabelle predefinite sono le seguenti:

- **tabella filter**: è responsabile del filtraggio dei pacchetti, permette cioè di bloccarli o di farli passare (funzionalità di firewall). Ogni pacchetto passa attraverso la tabella filtro. Essa contiene le seguenti catene predefinite:
  - **catena INPUT**: tutti i pacchetti destinati al sistema passano attraverso questa catena.
  - **catena OUTPUT**: tutti i pacchetti creati dal sistema passano attraverso questa catena.
  - **catena FORWARD**: tutti i pacchetti che hanno come destinazione finale un altro sistema e che non sono stati generati dal sistema stesso, cioè tutti i pacchetti che vengono solamente instradati dal sistema, passano attraverso questa catena.

- **tabella nat**: questa tabella è responsabile dell'impostazione delle regole per la modifica degli indirizzi e porte dei pacchetti. Il primo pacchetto di una connessione passa attraverso questa tabella, e il risultato del passaggio del primo pacchetto determina come tutti gli altri pacchetti della stessa connessione verranno modificati. La tabella nat contiene le seguenti catene predefinite:
  - **catena PREROUTING**: passano attraverso questa catena i pacchetti in entrata, il passaggio avviene *prima* che la locale tabella di routing venga consultata per effettuare l'instradamento. Essa è usata per il NAT sulla destinazione o DNAT o PAT.
  - **catena POSTROUTING**: passano attraverso questa catena i pacchetti in uscita *dopo* che la locale tabella di routing sia stata consultata. Usata per il NAT sulla sorgente o NAT propriamente detto.
  - **catena OUTPUT**: permette un DNAT limitato sui pacchetti generati localmente.

- **tabella mangle**: questa tabella è responsabile delle modifiche alle opzioni dei pacchetti, come ad esempio quella che determina la qualità del servizio. Tutti i pacchetti passano attraverso questa tabella.

La posizione delle catene rispetto al percorso dei pacchetti è mostrato in Figura 2.

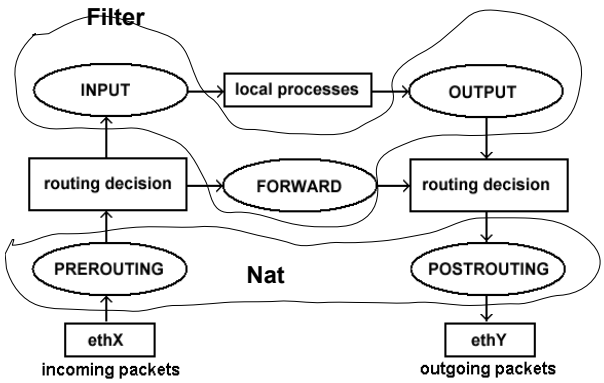

Figura 2. Catene e percorso dei pacchetti nel modulo di forwarding di Linux.

3.2 Regole: specifica delle caratteristiche

Questa parte di ciascuna regola serve a definire a quali pacchetti verrà applicata la regola. Una specifica di proprietà è costituita da un insieme di caratteristiche dei pacchetti che devono essere gestite dalla regola stessa. Le caratteristiche si possono unire mediante operatori logici.

Alcune delle opzioni più importanti per specificare una regola sono le seguenti:

```
-p [!] protocollo
--protocol [!] protocollo
```

Specifica il nome del protocollo dei pacchetti. Se avanti al nome del protocollo è presente il carattere '!' si specificano tutti i pacchetti che non usano quel protocollo. Alcuni nomi di protocollo molto usati sono ip, icmp, udp e tcp.

```
-s [!] sorgente[/prefisso]
--source [!] sorgente[/prefisso]
```

Specifica i pacchetti provenienti da una determinata sorgente. La sorgente può essere specificata con un indirizzo IP, un indirizzo IP con associato un prefisso di rete, o un nome del DNS. Se '!' precede l'indirizzo, si specificano tutti i pacchetti che non provengono da quell'indirizzo

```
-d [!] destinazione[/prefisso]
--destination [!] destinazione[/prefisso]
```

Specifica i pacchetti che hanno una determinata destinazione. La specifica dell'indirizzo di destinazione segue regole analoghe alla specifica dell'indirizzo sorgente.

```
-i, --in-interface [!] [nome]
```

Nome opzionale dell'interfaccia (es. eth0,eth1) attraverso la quale il pacchetto è ricevuto (per pacchetti che transitano attraverso le catene INPUT, FORWARD and PREROUTING). Se '!' precede il nome la regola è invertita, cioè si specificano tutti i pacchetti che non provengono da quell'interfaccia. Se il nome dell'interfaccia finisce con '+' tutte le interfacce che cominciano col nome specifcato prima del '+' sono valide. Se il parametro non è specificato allora viene assunto come valore predefinito la stringa '+', cioè tutte le interfacce.

```
-o, --out-interface [!] [nome]
```

Nome opzionale dell'interfaccia (es. eth0,eth1) attraverso la quale il pacchetto sarà inviato (per pacchetti che transitano attraverso le catene FORWARD, OUTPUT e POSTROUTING). Se '!' precede il nome la regola è invertita, cioè si specificano tutti i pacchetti che non provengono da quell'interfaccia. Se il nome dell'interfaccia finisce con '+' tutte le interfacce che cominciano col nome specifcato prima del '+' sono valide. Se il parametro non è specificato allora viene assunto come valore predefinito la stringa '+', cioè tutte le interfacce.

3.3 Regole: target

Il target di una regola è l'azione da compiere se un pacchetto rispetta le proprietà della regola, e viene specificato con la seguente opzione:

```
-j obiettivo
--jump obiettivo
```

L'azione può essere:
- il salto ad una catena definita dall'utente
- una delle azioni predefinite (ACCEPT, DROP, QUEUE, o RETURN)
- una delle azioni aggiuntive, come ad esempio REJECT o LOG.

Quando l'obiettivo è il nome di una catena definita dall'utente, il pacchetto viene fatto passare per quella catena. Se il pacchetto non viene consumato da nessuna regola della nuova catena, in quanto non rispetta la specifica di nessuna delle sue regole, esso ritorna alla catena di partenza.

Gli obiettivi predefiniti principali sono:

ACCEPT — accetta
Questa azione comporta che netfilter accetterà il pacchetto. Il risultato pratico di questa accettazione dipende da quale catena sta processando il pacchetto. Per esempio, un pacchetto che è accettato dalla catena INPUT può essere ricevuto dal sistema, un pacchetto accettato dalla catena OUTPUT può essere inoltrato dal sistema, e un pacchetto accettato dalla catena FORWARD potrà essere smistato dal sistema a un'altra destinazione, un pacchetto “accettato” in una catena della tabella NAT non subirà alterazioni.

DROP — scarta
Questa azione determina che il pacchetto venga scartato senza effettuare ulteriori operazioni su di esso. Il pacchetto scomparirà senza che alcuna indicazione del fatto che sia stato scartato venga fornita all'applicazione o al sistema che ha inviato il pacchetto. Il mittente del pacchetto vedrà semplicemente scadere il tempo a disposizione per la comunicazione, e non potrà distinguere tra il caso in cui il pacchetto è stato ricevuto e poi scartato e il caso in cui il pacchetto non è mai stato ricevuto. Questo comportamento aumenta la sicurezza di un sistema in quanto un potenziale nemico non potrà neppure determinare se il sistema esiste effettivamente.

REJECT — rifiuta
Questa azione ha lo stesso effetto di DROP con l'eccezione che viene spedito un pacchetto di errore ICMP al mittente del pacchetto. Esso è principalmente utilizzato nelle catene INPUT o FORWARD della tabella filtro. Un pacchetto di errore può indicare esplicitamente che il pacchetto è stato filtrato.

LOG — annota
Con questa azione il pacchetto viene annotato, cioè la ricezione del pacchetto viene annotata inviando un messaggio sul SysLog. Questo obiettivo può essere utile per permettere all'amministratore di sapere

quali pacchetti vengono filtrati o allo sviluppatore per controllare il corretto funzionamento del sistema.

DNAT
Questa azione comporta la riscrittura dell'indirizzo di destinazione del pacchetto, per permettere il NAT sulla destinazione. Questo obiettivo è valido esclusivamente nelle catene OUTPUT e PREROUTING della tabella NAT. La decisione effettuata sul primo pacchetto verrà ripetuta per tutti i pacchetti della connessione, e i pacchetti di risposta avranno l'indirizzo sorgente originario.

SNAT
Questa azione comporta la riscrittura dell'indirizzo del mittente del pacchetto, per permettere il NAT sulla sorgente. Questo obiettivo è valido solo nella catena POSTROUTING della tabella NAT e il suo risultato è ripetuto per tutti i pacchetti della stessa connessione.

MASQUERADE — maschera
Questa è una forma speciale di SNAT per indirizzi IP dinamici, come quelli forniti da molti Internet Service Provider per i loro utenti.

3.4 Monitoraggio delle connessioni

Una delle funzionalità più importanti offerte da Netfilter è la possibilità di identificare i pacchetti facenti parte di una stessa connessione (stateful packet filtering). Questo permette di creare delle regole basate sulla relazione che un pacchetto ha nei confronti della connessione a cui appartiene. Il NAT si basa su queste informazioni per tradurre allo stesso modo gli indirizzi dei pacchetti di una stessa connessione, e iptables usa queste informazioni per realizzare firewall avanzati.

Netfilter assegna ad ogni pacchetto uno dei seguenti stati:

- NEW (NUOVO), il pacchetto inizia una nuova connessione;
- ESTABLISHED (STABILITO), il pacchetto fa parte di una connessione già stabilita;
- RELATED (IN RELAZIONE), il pacchetto ha qualche relazione con un'altra connessione già stabilita;
- INVALID (INVALIDO), il pacchetto non fa parte di alcuna connessione e non è possibile crearne.

Un caso comune è che il primo pacchetto TCP visto dal firewall viene classificato come NEW, la risposta viene classificata come ESTABLISHED e un messaggio di errore, ad esempio un errore ICMP, come RELATED. Un errore ICMP che non appartiene a nessuna connessione può essere classificato come INVALID.

Netfilter può utilizzare l'informazione sullo stato dei pacchetti per creare filtri più potenti e più semplici da definire. Per esempio, una regola può lasciar passare i pacchetti NEW solo dall'interno del firewall verso la rete esterna, e i pacchetti RELATED e ESTABLISHED in entrambe le direzioni. Questo permette di inviare delle risposte a connessioni create dall'interno, ma non permette di creare nuove connessioni dall'esterno, il che aumenta la sicurezza di un sistema perché gli attacchi che vengono dall'esterno non sono in grado di stabilire connessioni (Figura 3).

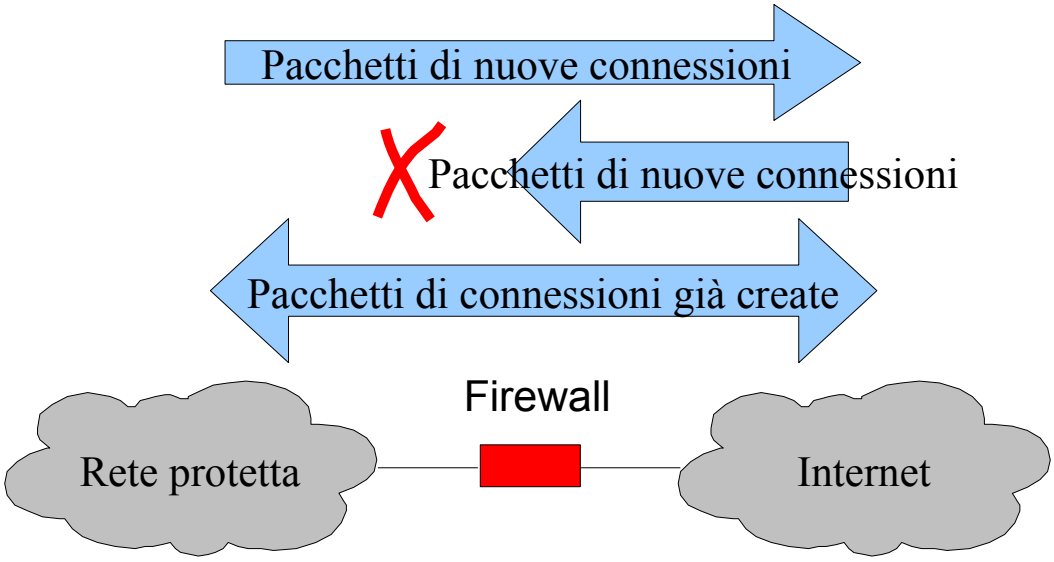

Figura 3. Blocco delle connessioni TCP.

4 Il software IPTables

Iptables è un'applicazione che permette agli amministratori di configurare le tabelle, le catene e le regole di Netfilter. Dato che iptables modifica il funzionamento del sistema operativo, per essere eseguito è necessario entrare nel sistema come utente amministratore, che nei sistemi di tipo Unix è l'utente root, il quale ha i permessi per compiere qualsiasi tipo di operazione. Sulla maggior parte dei sistemi Linux, iptables è installato come `/usr/sbin/iptables`. La lista completa delle funzionalità del comando è consultabile nella relativa documentazione, che può essere visualizzata con il comando `man iptables`.

Per fare delle prove si consiglia di aprire una nuova shell che abbia direttamente le credenziali di root:

```
sudo /bin/bash
```

4.1 Opzioni generali

Tutte le forme di invocazione di Iptables supportano le seguenti opzioni:

```
-t tabella
```

applica il comando alla tabella specificata. Quando questa opzione è omessa, il comando si applica alla tabella filter.

```
-v
```

fornisce una quantità maggiore di informazioni.

```
--line-numbers
```

nell'elenco delle regole, aggiunge i numeri di riga all'inizio di ogni regola.

4.2 Invocazione di Iptables

Le operazioni messe a disposizione dal programma Iptables sono invocabili specificando una serie di

parametri sulla riga di comando. I termini tra parentesi graffe, {...|...|...}, sono necessari, ma solo uno di essi può essere inserito in una singola invocazione del comando. I termini tra parentesi quandre, [...], sono opzionali.

```
iptables { -L | --list | -F | --flush | -Z | --zero } [ nome_catena ] [ opzioni ]
```

Queste opzioni permettono di ottenere la lista delle regole di una catena, con -L o --list, cancellare tutte le regole di una catena, con -F o --flush, e azzerare il contatore di byte e pacchetti di una catena, con l'opzione -Z o --zero. Se nessuna catena è specificata, l'operazione è eseguita su tutte le catene. Ad esempio, per elencare tutte le regole della tabella filter, si usa il comando:

```
iptables -L
```

Per cancellare tutte le catene si usa il comando:

```
iptables -F
```

```
iptables { -A | --append | -D | --delete } nome_catena nome_regola [ opzioni ]
```

Con queste opzioni si può aggiungere o cancellare una regola dalla catena specificata. Le opzioni -A e --append permettono di aggiungere, -D e --delete di cancellare. Ad esempio, per aggiungere una regola alla catena INPUT nella tabella filter per scartare tutti i pacchetti UDP, si può usare il seguente comando:

```
iptables -A INPUT -p udp -j DROP
```

La tabella filtro è la tabella di default, non è quindi necessario specificarla con l'opzione -t. Per cancellare la regola aggiunta dal comando precedente, si può usare il seguente comando:

```
iptables -D INPUT -p udp -j DROP
```

Questo comando in realtà cancella la prima regola della catena INPUT che ha la specifica "-p udp -j DROP". Se vi sono altre regole con la stessa specifica, solo la prima verrà cancellata.

```
iptables { -R | --replace | -I | --insert } nome_catena numero_regola nuova_regola
[ opzioni ]
```

Con questo comando è possibile sostituire, con -R o --replace, una regola esistente o inserire, con -I o --insert, una nuova regola nella catena specificata. Ad esempio, per sostiture la quarta regola della catena INPUT con una regola che scarta tutti i pacchetti ICMP, si può usare il seguente comando:

```
iptables -R INPUT 4 -p icmp -j DROP
```

Per inserire una nuova regola nella seconda posizione della catena OUTPUT per scartare tutto il traffico TCP sulla porta 80, si può usare il comando:

```
iptables -I OUTPUT 2 -p tcp --dport 80 -j DROP
```

```
iptables { -D | --delete } nome_catena numero_regola [ opzioni ]
```

Questa opzione permette di cancellare una regola della catena specificata, indicandone la posizione all'interno della catena. Le regole sono numerate a partire da 1. Per esempio, per cancellare la terza regola dalla catena FORWARD, si usa il comando:

```
iptables -D FORWARD 3
```

```
iptables { -N | --new-chain } nome_catena
iptables { -X | --delete-chain } [ nome_catena ]
```

Queste opzioni permettono di creare, con -N o --new-chain, e cancellare, con -X o --delete-chain, catene definite dall'utente. Nel caso della cancellazione, se nessuna catena è specificata, tutte le catene definite dall'utente verranno cancellate. Non è possibile cancellare le catene predefinite, come le catene INPUT o OUTPUT della tabelle filtro.

```
iptables { -P | --policy } nome_catena nome_politica
```

Questo comando è usato per assegnare una politica di default ad una catena. Ad esempio, per impostare la politica DROP per la catena INPUT della tabella filter si usa il comando:

```
iptables -P INPUT DROP
```

```
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to 1.2.3.0
```

Questo comando è usato per fare il source natting dei pacchetti uscenti dall'interfaccia eth1. L'indirizzo originale

Esercizi

1. Impostare a DROP la politica di default della catena INPUT della tabella Filter e ad ACCEPT la politica di default della catena OUTPUT della tabella Filter. Fare un ping alla propria macchina Windows e vedere l'effetto. Rifare l'esperimento catturando i pacchetti con Wireshark (si puo' impostare “icmp” nel filtro di visualizzazione); cosa si nota ?
2. Impostare ad ACCEPT la politica di default della catena INPUT della tabella Filter e ad DROP la politica di default della catena OUTPUT della tabella Filter. Fare un ping alla propria macchina Windows e vedere l'effetto. Rifare l'esperimento catturando i pacchetti con Wireshark (si puo' impostare “icmp” nel filtro di visualizzazione); cosa si nota ?
3. Rimettere tutte le catene in ACCEPT. Impostare un source natting usando 130.192.16.19. Fare un ping alla propria macchina Windows catturando i pacchetti con Wireshark (si puo' impostare “icmp” nel filtro di visualizzazione); cosa si nota nell'header IP dei pacchetti ICMP catturati ?