può essere utile

```
//comando per vedere le tabelle
sudo iptables -L
```

```
//comando più avanzato per vedere le tabelle
sudo iptables -L -n -v --line-numbers
```

- `L`: Elenca tutte le regole del firewall. `L` sta per "list".
- `n`: Visualizza gli indirizzi IP e le porte in formato numerico. Senza `n`, `iptables` proverebbe a risolvere gli indirizzi IP e i nomi delle porte in nomi di host e servizi, il che potrebbe rallentare l'output se ci sono molte regole.
- `v`: Fornisce un output dettagliato. `v` sta per "verbose" e aggiunge dettagli extra come il numero di pacchetti e byte che corrispondono a ciascuna regola.
- `-line-numbers`: Aggiunge numeri di linea all'inizio di ogni regola. Questo è utile per identificare le regole specifiche, soprattutto quando si vuole modificare o eliminare una regola specifica.

```
//regola per le connessioni già established o related
sudo iptables -A INPUT -j ACCEPT -m conntrack -ctstate ESTA
BLISHED,RELATED
```

- `A INPUT`: Questo parametro aggiunge (`A` sta per "append", cioè aggiungere) una regola alla catena `INPUT`. La catena `INPUT` gestisce i pacchetti in ingresso al sistema.

- `j ACCEPT`: Questa parte della regola specifica l'azione da intraprendere sui pacchetti che corrispondono alla regola. `j` sta per "jump", e `ACCEPT` indica che i pacchetti devono essere accettati e lasciati passare.
- `m conntrack`: Questo parametro specifica che viene utilizzato il modulo `conntrack`, che è un modulo di tracciamento delle connessioni. Questo modulo tiene traccia dello stato delle connessioni di rete.
- `-ctstate ESTABLISHED,RELATED`: Questo parametro specifica gli stati delle connessioni a cui si applica la regola. `ESTABLISHED` si riferisce ai pacchetti che fanno parte di una connessione già stabilita. `RELATED` si riferisce ai pacchetti che sono correlati a una connessione già stabilita, come un'ulteriore connessione FTP di dati correlata a una connessione di controllo FTP esistente.

```
//per cancellare una rule
sudo iptables -D INPUT 1
```

- `D`: Indica l'azione di "delete" (elimina). Questo parametro specifica che si desidera eliminare una regola.
- `INPUT`: La catena dalla quale si desidera eliminare la regola. In questo caso, la catena `INPUT`, che gestisce i pacchetti in ingresso al sistema.
- `1`: Il numero di linea della regola da eliminare. Questo numero si riferisce alla posizione della regola nell'elenco corrente delle regole della catena `INPUT`.

```
//comando per accettare pacchetti dall'esterno tramite icmp
sudo iptables -A INPUT -j ACCEPT -p icmp --icmp-type 8

//se si volesse usare tcp su una porta specifica (ex.la por
ta 22 per SSH)
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

- `p icmp`: Specifica che la regola si applica ai pacchetti ICMP (Internet Control Message Protocol).
- `-icmp-type 8`: Specifica il tipo di messaggio ICMP a cui si applica la regola. ICMP type 8 corrisponde ai messaggi di "echo request" (richieste di eco), che sono utilizzati dai comandi `ping` per verificare la raggiungibilità di un host.
- `p tcp`: Specifica che la regola si applica ai pacchetti TCP.
- `-dport 22`: Specifica la porta di destinazione (22 in questo caso). Questo significa che la regola si applica ai pacchetti destinati alla porta 22.

NB: per quale porta è necessario farlo sia con il protocollo tcp che con il protocollo udp.

Porta 22

- **Servizio**: SSH (Secure Shell)
- **Uso**: Utilizzata per accesso remoto sicuro e gestione di server attraverso una connessione cifrata.

Porta 53

- **Servizio**: DNS (Domain Name System)
- **Uso**: Utilizzata per la risoluzione dei nomi di dominio. I server DNS utilizzano questa porta per ricevere richieste di traduzione di nomi di dominio in indirizzi IP.

Porta 80

- **Servizio**: HTTP (HyperText Transfer Protocol)
- **Uso**: Utilizzata per il traffico web non cifrato. È la porta predefinita per i server web che servono pagine web non cifrate.

Porta 443

- **Servizio**: HTTPS (HTTP Secure)

- **Uso**: Utilizzata per il traffico web cifrato. È la porta predefinita per i server web che servono pagine web cifrate tramite SSL/TLS.

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

NB: La mancanza di regole esplicite per le porte 80 e 443 nella catena `INPUT` può essere attribuita al tracciamento dello stato `RELATED,ESTABLISHED`.

Ecco perché:

1. **Tracciamento dello Stato `RELATED,ESTABLISHED`**:
    - La regola con `ctstate RELATED,ESTABLISHED` nella catena `INPUT` consente effettivamente l'ingresso di pacchetti che fanno parte di connessioni stabilite o correlate. Ciò significa che se il tuo sistema inizia una connessione con un server web (ad esempio, facendo una richiesta HTTP), il traffico di ritorno da quel server (come le risposte alle tue richieste HTTP) è considerato parte di una connessione stabilita e sarà consentito da questa regola.
2. **Stabilimento della Connessione**:
    - Quando il tuo sistema effettua una richiesta HTTP o HTTPS a un server web, di solito passa attraverso la catena `OUTPUT`. Una volta stabilita la connessione, le risposte dal server web sono considerate parte della stessa connessione e sono consentite dalla regola `RELATED,ESTABLISHED` nella catena `INPUT`, senza la necessità di regole separate per le porte 80 e 443.
3. **Tracciamento Dinamico dello Stato**:
    - Con il firewall stateful, come quello fornito da `iptables`, una volta stabilita una connessione, il firewall tiene traccia di essa. Qualsiasi traffico successivo correlato a quella connessione (come le risposte da un server web dopo che sono state effettuate richieste in uscita) è automaticamente consentito, anche se non ci sono regole esplicite che consentono il traffico sulle porte 80 e 443 nella catena `INPUT`.

In sintesi, la mancanza di regole specifiche per le porte 80 e 443 nella catena `INPUT` non è un problema perché la regola `RELATED,ESTABLISHED` gestisce efficacemente il traffico in ingresso correlato a connessioni stabilite, il che include le risposte dai server web dopo che sono state effettuate richieste in uscita. Questo tracciamento dinamico semplifica la configurazione del firewall mantenendo la sicurezza.

NB: Se la default policy di una catena è impostata su "DROP", significa che tutti i pacchetti che NON corrispondono a nessuna regola esplicita saranno scartati. In questo caso, è fondamentale aggiungere regole per consentire il traffico in ingresso che si desidera far passare attraverso il firewall.

Se la default policy della catena `INPUT` è impostata su "DROP", devi aggiungere regole specifiche per consentire il traffico in ingresso che desideri permettere. Queste regole dovrebbero includere, almeno, il traffico necessario per stabilire e mantenere le connessioni di rete essenziali per il funzionamento del tuo sistema.

```
//comando che permette il trasferimento di pacchetti tra le interfacce lan      (interno) e wan (esterno) del firewall
sudo iptables -A FORWARD -j ACCEPT -i lan -o wan -m comment --comment "allow outbound traffic from lan"
```

- `i lan`: Specifica l'interfaccia di ingresso `lan`. Questa opzione consente di limitare l'applicazione della regola al traffico che entra nell'interfaccia `lan`.
- `o wan`: Specifica l'interfaccia di uscita `wan`. Questa opzione consente di limitare l'applicazione della regola al traffico che esce dall'interfaccia `wan`.
- `m comment --comment "allow outbound traffic from lan"`: Aggiunge un commento alla regola per scopi di documentazione o tracciabilità. Il commento non ha effetti sul comportamento della regola, ma fornisce informazioni utili sul suo scopo o sulla sua funzione.

```
//comando per vedere la tabella nat
sudo iptables -L -t nat
```

```
//comando per effettuare il mascheramento (masquerading) de
l traffico in uscita dalla rete locale (lan) verso l'estern
o (wan)
sudo iptables -t nat -A POSTROUTING -o wan -j MASQUERADE -m
comment --comment "masquerade lan->wan"
```

- `t nat`: Specifica la tabella `nat`. Questa tabella è utilizzata per modificare l'indirizzamento dei pacchetti di rete (Network Address Translation, NAT).
- `A POSTROUTING`: Aggiunge (`A` sta per "append", cioè aggiungere) una regola alla catena `POSTROUTING`. Questa catena viene utilizzata per modificare i pacchetti dopo che sono stati instradati.
- `o wan`: Specifica l'interfaccia di uscita `wan`. Questa opzione consente di limitare l'applicazione della regola al traffico che esce dall'interfaccia `wan`.
- `j MASQUERADE`: Specifica l'azione di "masquerade" per il traffico corrispondente. Questo fa sì che l'indirizzo IP sorgente dei pacchetti venga modificato con l'indirizzo IP dell'interfaccia di uscita (`wan`), mascherando così la rete locale.
- `m comment --comment "masquerade lan->wan"`: Aggiunge un commento alla regola per scopi di documentazione o tracciabilità, fornendo informazioni sul suo scopo o sulla sua funzione.

NB: Se non viene specificata una tabella specifica con l'opzione `-t`, di default, `iptables` opera sulla tabella `filter`. La tabella `filter` è la tabella predefinita e viene utilizzata per il filtraggio dei pacchetti, inclusa l'accettazione, il rifiuto o il rilascio di pacchetti in base alle regole definite.

```
//comando che abilita l'inoltro del traffico IP all'interno
del kernel Linux
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Quando `ip_forward` è impostato su `1`, il sistema operativo sarà in grado di inoltrare i pacchetti IP da una interfaccia di rete a un'altra.

`sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"`: Questo comando utilizza `sh` per eseguire un comando specifico in una shell. L'`echo 1 > /proc/sys/net/ipv4/ip_forward` è il comando che scrive il valore `1` nel file `/proc/sys/net/ipv4/ip_forward`, abilitando così l'inoltro del traffico IP.

NB: Entrambi `ip a` e `ip route` sono comandi della suite di strumenti `iproute2` di Linux utilizzati per la configurazione e la gestione delle reti. Tuttavia, ciascun comando ha un obiettivo diverso:

1. **`ip a` (o `ip addr show`)**:
    - Questo comando viene utilizzato per visualizzare le informazioni sulle interfacce di rete attualmente configurate sul sistema, inclusi indirizzi IP, stati e altre informazioni correlate all'interfaccia. Ad esempio, puoi vedere l'indirizzo IP assegnato a un'interfaccia specifica con `ip a`. È utile per ottenere una panoramica delle interfacce di rete attive sul sistema.
2. **`ip route` (o `ip route show`)**:
    - Questo comando viene utilizzato per visualizzare la tabella di routing del kernel, che contiene le informazioni necessarie per instradare i pacchetti attraverso la rete. Mostra le rotte disponibili sul sistema, inclusi i gateway predefiniti e le rotte specifiche per reti particolari. È utile per comprendere come i pacchetti IP sono instradati attraverso la rete e per diagnosticare eventuali problemi di routing.

In sintesi, `ip a` è utilizzato per visualizzare le informazioni sulle interfacce di rete, mentre `ip route` è utilizzato per visualizzare le informazioni sulla tabella di

routing del sistema. Entrambi sono comandi utili per la gestione e la diagnostica delle reti su sistemi Linux.

```
//comando che aggiunge una regola alla tabella nat per modi
ficare la            destinazione dei pacchetti TCP in arri
vo sulla porta 22 e destinati            all'indirizzo IP
1912.168.5.201. La regola riscriverà la destinazione dei
pacchetti in modo che siano inviati a 172.16.1.102 anziché
a 192.168.5.201.
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -d 19
2.168.5.201 -j DNAT --to-destination 172.16.1.102
```

- `A PREROUTING`: Aggiunge (`A` sta per "append", cioè aggiungere) una regola alla catena `PREROUTING`. Questa catena viene eseguita prima della decisione di instradamento e consente di modificare i pacchetti appena arrivati.
- `p tcp`: Specifica che la regola si applica ai pacchetti TCP.
- `-dport 22`: Specifica la porta di destinazione (porta 22 per SSH).
- `d 1912.168.5.201`: Specifica l'indirizzo IP di destinazione dei pacchetti. Tuttavia, c'è un errore nell'indirizzo IP, dovrebbe essere `192.168.5.201` anziché `1912.168.5.201`.
- `j DNAT`: Indica che la destinazione dei pacchetti corrispondenti deve essere modificata utilizzando la traduzione degli indirizzi di destinazione (DNAT).
- `-to-destination 172.16.1.102`: Specifica l'indirizzo IP di destinazione a cui i pacchetti saranno ridirezionati dopo la modifica.

Questa regola di DNAT viene utilizzata per ridirezionare il traffico in arrivo sulla porta 22 e destinato all'indirizzo IP 192.168.5.201 verso un altro indirizzo IP, 172.16.1.102. Potrebbe essere utile, ad esempio, quando si desidera instradare il traffico SSH verso un'altra macchina sulla rete.

NB: Una regola di `DNAT` come quella fornita è utile in diversi scenari di rete. Ecco alcuni casi d'uso comuni:

1. **Redirezione del traffico verso un server diverso**: Se vuoi instradare il traffico destinato a un certo indirizzo IP e porta verso un server diverso

sulla rete, puoi utilizzare una regola `DNAT`. Ad esempio, potresti voler ridirezionare il traffico SSH (porta 22) destinato a un server specifico verso un altro server.

2. **Bilanciamento del carico**: Puoi utilizzare la DNAT per distribuire il carico del traffico su più server. Ad esempio, puoi configurare un gruppo di server con lo stesso servizio e utilizzare una regola DNAT per distribuire equamente il traffico tra di essi.
3. **Sostituzione degli indirizzi IP**: In alcune situazioni, potresti dover modificare gli indirizzi IP di destinazione dei pacchetti per adattarli alla configurazione della rete. Ad esempio, quando una rete interna è migrata verso una nuova subnet, potresti utilizzare la DNAT per instradare il traffico verso i nuovi indirizzi IP.
4. **Accesso remoto a risorse interne**: Se vuoi consentire l'accesso remoto a risorse interne dietro un firewall, puoi utilizzare la DNAT per instradare il traffico esterno verso queste risorse. Tuttavia, assicurati di proteggere adeguatamente queste risorse per evitare accessi non autorizzati.

```
//comando che aggiunge una regola alla catena FORWARD del f
irewall che accetta  il traffico TCP inoltrato (forwarded)
destinato alla porta 22 e all'indirizzo IP di destinazione
172.16.1.102
sudo iptables -A FORWARD -p tcp --dport 22 -d 172.16.1.102
-j ACCEPT
```

- `A FORWARD`: Aggiunge (`A` sta per "append", cioè aggiungere) una regola alla catena `FORWARD`. Questa catena viene utilizzata per gestire il traffico inoltrato attraverso il firewall.
- `p tcp`: Specifica che la regola si applica ai pacchetti TCP.
- `-dport 22`: Specifica la porta di destinazione (porta 22 per SSH).
- `d 172.16.1.102`: Specifica l'indirizzo IP di destinazione dei pacchetti.
- `j ACCEPT`: Indica che i pacchetti corrispondenti devono essere accettati e inoltrati senza ulteriori controlli.

Questa regola `ACCEPT` viene utilizzata per consentire il passaggio del traffico TCP destinato alla porta 22 e all'indirizzo IP 172.16.1.102 attraverso il firewall. Questo può essere utile, ad esempio, se hai un server SSH con indirizzo IP 172.16.1.102 e desideri consentire ai client di connettersi a questo server attraverso il firewall.

```
//comandi che permettono di tracciare i tentativi di connes
sione SSH e di       bloccare quelli che superano una certa
soglia (6 tentativi in 60 secondi) per   evitare attacchi d
i forza bruta
sudo iptables -I INPUT 4 -p tcp --dport 22 -m conntrack --c
tstate NEW -m recent --name ssh-list --set -m comment --com
ment "track new ssh attempts"

sudo iptables -I INPUT 5 -p tcp --dport 22 -m conntrack --c
tstate NEW -m recent --name ssh-list --update --seconds 60
--hitcount 6 -j DROP -m comment --comment "drop excessive s
sh attempts"
```

- `I INPUT 4`: Inserisce (`I`) la regola nella catena `INPUT` al quarto posto. Questo significa che questa regola verrà valutata prima di altre regole esistenti.
- `p tcp --dport 22`: Specifica che la regola si applica solo al traffico TCP destinato alla porta SSH (porta 22).
- `m conntrack --ctstate NEW`: Applica la regola solo ai pacchetti che sono nuove connessioni.
- `m recent --name ssh-list --set`: Utilizza il modulo `recent` per tenere traccia dei pacchetti di nuova connessione che corrispondono a questa regola e li aggiunge a una lista chiamata `ssh-list`.

- `I INPUT 5`: Inserisce la seconda regola nella catena `INPUT` al quinto posto.
- `p tcp --dport 22`: Specifica che la regola si applica solo al traffico TCP destinato alla porta SSH (porta 22).

- `m conntrack --ctstate NEW`: Applica la regola solo ai pacchetti che sono nuove connessioni.
- `m recent --name ssh-list --update --seconds 60 --hitcount 6`: Utilizza il modulo `recent` per verificare se ci sono più di 6 nuovi tentativi di connessione SSH nella lista `ssh-list` negli ultimi 60 secondi.
- `j DROP`: Se il numero di tentativi di connessione supera la soglia definita, la regola instraderà i pacchetti al target `DROP`, cioè li scarterà.

```
//comando che configura una regola nella tabella nat per il reindirizzamento del traffico HTTP (porta 80) proveniente dalla rete locale (sottorete 172.16.1.0/24) verso un proxy HTTP trasparente
sudo iptables -t nat -I PREROUTING 3 -p tcp --dport 80 -m comment --comment "transparent http proxy" -s 172.16.1.0/24 -j DNAT --to-destination 172.16.1.1:8888
```

- `sudo iptables -t nat -I PREROUTING 3`: Inserisce (`I`) la regola nella catena `PREROUTING` della tabella `nat` al terzo posto. La catena `PREROUTING` viene valutata prima che avvenga la traduzione dell'indirizzo di destinazione.
- `p tcp --dport 80`: Specifica che la regola si applica solo al traffico TCP destinato alla porta 80, che è la porta standard per HTTP.
- `m comment --comment "transparent http proxy"`: Aggiunge un commento alla regola per descriverne lo scopo.
- `s 172.16.1.0/24`: Specifica che la regola si applica solo al traffico proveniente dalla sottorete 172.16.1.0/24 (rete locale).
- `j DNAT --to-destination 172.16.1.1:8888`: Indica che il traffico corrispondente deve essere destinato a un proxy HTTP trasparente. Il proxy HTTP trasparente è configurato per ascoltare sulla porta 8888. L'opzione `DNAT` (Destination NAT) consente di modificare l'indirizzo di destinazione dei pacchetti in modo che vengano inviati al proxy HTTP invece del server originale.
