ESERCIZIO 11 GIUGNO 2021

L'esercitazione di iptables consiste nel creare una serie di regole per un singolo host che potrebbe avere più interfacce e funzionare come router.

Scrivete nel file di testo i comandi che si devono impartire per ottenere i risultati richiesti. Si consiglia di leggere prima tutta la lista per determinare correttamente i requisiti. Le regole saranno testate per verificarne la correttezza. Le regole devono essere applicate nell'ordine in cui vengono proposte.

Si inizi garantendo che le catene siano tutte vuote.

Utilizziamo -A per inserirle una dopo l'altra

Consentire qualsiasi traffico sull'interfaccia di loopback

- **iptables -A INPUT -i lo -j ACCEPT**
- **iptables -A OUTPUT -o lo -j ACCEPT**

Consentire il traffico delle connessioni HTTP entranti

- **iptables -A INPUT -p tcp --dport 80 -j ACCEPT**
- **iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT**

Consentire connessioni SSH uscenti verso la rete host-only 192.168.56.0/24

- **iptables -A OUTPUT -p tcp -d 192.168.56.0/24 --dport 22 -j ACCEPT**
- **iptables -A INPUT -p tcp -s 192.168.56.0/24 --sport 22 -m state --state ESTABLISHED -j ACCEPT**

Bloccare l'inoltro del traffico proveniente dalla rete host-only verso altre destinazioni

- **iptables -A FORWARD -s 192.168.56.0/24 ! -d 192.168.56.0/24 -j DROP**
  (il punto esclamativo nega, quindi droppa le cose che NON hanno quella destionazione)

Consentire la risoluzione dei nomi DNS

- **iptables -A OUTPUT -p udp --dport 53 -j ACCEPT**
- **iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT**

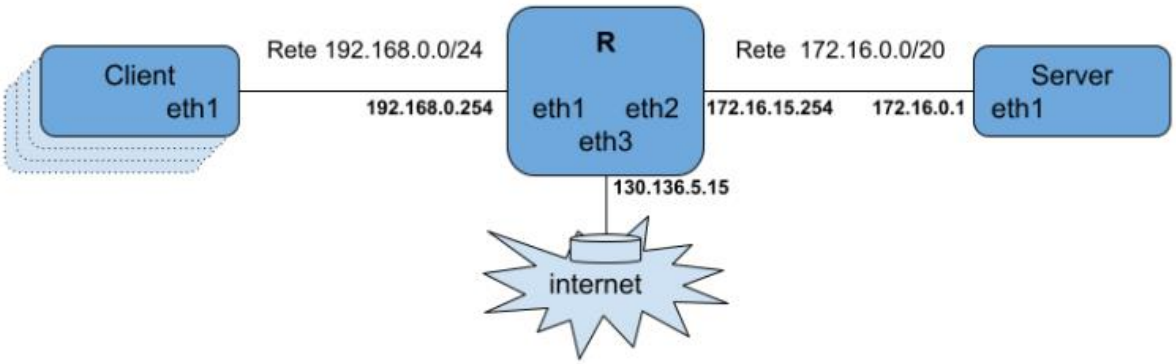
Infine bloccare tutto il traffico non elencato nei punti 2, 3, 5

- **iptables -P INPUT DROP**
- **iptables -P OUTPUT DROP**
- **iptables -P FORWARD ACCEPT**

DA NOTARE: per il FORWARD deve valere la regola di default accept perché altrimenti non avrebbe senso la regola terzultima regola che applica DROP ad una azione specifica.

ESERCIZIO 13 SETTEMBRE 2023

Facendo riferimento allo schema di rete sopra riportato, si definiscano regole di filtraggio che consentano il traffico come sotto specificato; qualsiasi altro pacchetto deve essere scartato.
NOTA: le regole devono essere installate su ogni host coinvolto nel flusso di traffico specificato.



1. **i Client sulla rete privata 192.168.0.0/24 devono poter interrogare DNS e servizi di sincronizzazione NTP in Internet (porte UDP 53 e 1233)**
2. **il servizio SMTP (porta 25 tcp) del Server collocato sulla rete privata 172.16.0.0/20 deve essere raggiungibile da qualsiasi host di Internet**
3. **il servizio LDAP (porta 389 tcp) del Router deve essere raggiungibile dal Server**

SUL CLIENT

- iptables -F INPUT
- iptables -F OUTPUT
- iptables -F FORWARD
- iptables -I INPUT -i lo -j ACCEPT
- iptables -I OUTPUT -o lo -j ACCEPT

1. iptables -A INPUT -p udp -i eth1 --sport 53 -m state --state ESTABLISHED -j ACCEPT
   iptables -A OUTPUT -p udp -o eth1 --dport 53 -j ACCEPT
   iptables -A INPUT -p udp -i eth1 --sport 1233 -m state --state ESTABLISHED -j ACCEPT
   iptables -A OUTPUT -p udp -o eth1 --dport 1233 -j ACCEPT

- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -P FORWARD DROP

SUL SERVER

- iptables -F INPUT
- iptables -F OUTPUT
- iptables -F FORWARD
- iptables -I INPUT -i lo -j ACCEPT
- iptables -I OUTPUT -o lo -j ACCEPT

2. per semplicità accettiamo tutte le sorgenti ma a rigore andrebbero escluse le reti private
   iptables -A INPUT -p tcp -i eth1 -d 172.16.0.1 --dport 25 -j ACCEPT
   iptables -A OUTPUT -p tcp -o eth1 -s 172.16.0.1 --sport 25 -m state --state ESTABLISHED -j ACCEPT
3. iptables -A OUTPUT -p tcp -o eth1 -s 172.16.0.1 -d 172.16.15.254 --dport 389 -j ACCEPT
   iptables -A INPUT -p tcp -i eth1 -s 172.16.15.254 -d 172.16.0.1 --sport 389 -m state --state ESTABLISHED -j ACCEPT

- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -P FORWARD DROP

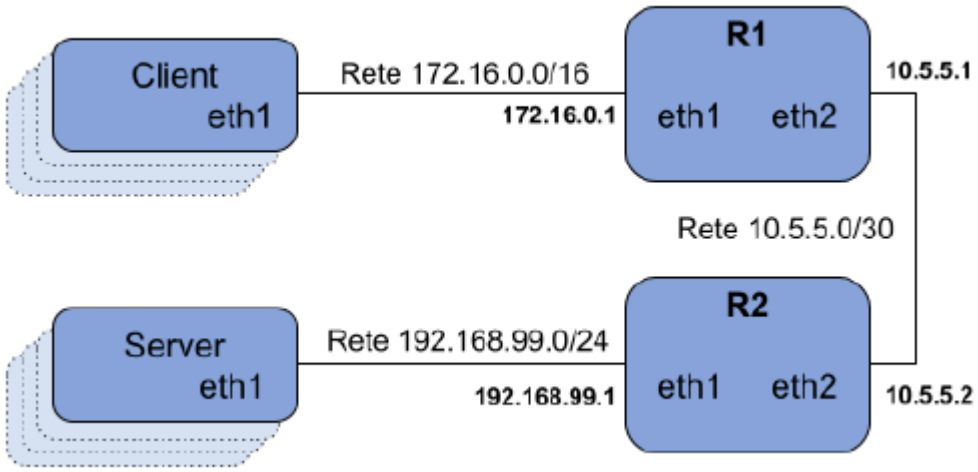
SUL ROUTER

- iptables -t nat -F
- iptables -F INPUT
- iptables -F OUTPUT
- iptables -F FORWARD
- iptables -I INPUT -i lo -j ACCEPT
- iptables -I OUTPUT -o lo -j ACCEPT

1. iptables -A FORWARD -p udp -i eth1 -o eth3 -s 192.168.0.0/16 --dport 53 -j ACCEPT
   iptables -A FORWARD -p udp -i eth3 -o eth1 -d 192.168.0.0/16 --sport 53 -m state --state ESTABLISHED -j ACCEPT
   iptables -A FORWARD -p udp -i eth1 -o eth3 -s 192.168.0.0/16 --dport 1233 -j ACCEPT
   iptables -A FORWARD -p udp -i eth3 -o eth1 -d 192.168.0.0/16 --sport 1233 -m state --state ESTABLISHED -j ACCEPT
   iptables -t nat -A POSTROUTING -p tcp -i eth1 -s 192.168.0.0/24 -o eth3 -j SNAT --to-source 130.136.5.15
2. **iptables -t nat -A PREROUTING -i eth3 -o eth1 -s 130.136.5.15 -p tcp --dport 25 -j DNAT --to-destination 172.16.0.1**
   iptables -A FORWARD -p tcp -i eth3 -o eth2 -d 172.16.0.1 --dport 25 -j ACCEPT
   iptables -A FORWARD -p tcp -i eth2 -o eth3 -s 172.16.0.1 --sport 25 -m state --state ESTABLISHED -j ACCEPT
3. iptables -A INPUT -p tcp -i eth2 -s 172.16.0.1 -d 172.16.15.254 --dport 389 -j ACCEPT
   iptables -A OUTPUT -p tcp -o eth2 -s 172.16.15.254 -d 172.16.0.1 --sport 389 -m state --state ESTABLISHED -j ACCEPT

- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -P FORWARD DROP

ESERCIZIO 27 GIUGNO 2023



1. **i Client devono poter accedere via HTTPS (porta TCP 443) a qualunque server; il reale indirizzo dei client deve essere nascosto sia a R2 che ai server**

Client

- iptables -A OUTPUT -d 192.168.99.0/24 -o eth1 -p tcp --dport 443 -j ACCEPT
- iptables -A INPUT -s 192.168.99.0/24 -i eth1 -p tcp --sport 443 -m state ESTABLISHED -j ACCEPT

R1

- iptables -A FORWARD -i eth1 -s 172.16.0.0/16 -o eth2 -d 192.168.99.0/24 -p tcp --dport 443 -j ACCEPT
- iptables -A FORWARD -o eth1 -s 192.168.99.0/24 -i eth2 -d 172.16.0.0/16 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
  #Cambiamo l'indirizo sorgernte così da nascondere gli indirizzi dei client a R2 e servers sostituendolo con quello di R1 visibile su eth2
- iptables -t nat -I POSTROUTING -s 172.16.0.0/16 -d 192.168.99.0/24 -p tcp --dport 443 -j SNAT --to-source 10.5.5.1

R2 (R2 vede arrivare i pacchetti da R1, idem per il ritorno: destinate a 10.5.5.1)

- iptables -A FORWARD -i eth2 -s 10.5.5.1 -o eth1 -d 192.168.99.0/24 -p tcp --dport 443 -j ACCEPT
- iptables -A FORWARD -o eth2 -s 192.168.99.0/24 -i eth1 -d 10.5.5.1 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

Server

- iptables -A INPUT -s 10.5.5.1 -i eth1 -p tcp --dport 443 -j ACCEPT
- iptables -A OUTPUT -d 10.5.5.1 -i eth1 -p tcp --sport 443 -m state ESTABLISHED -j ACCEPT

2. **il Server 192.168.99.2 deve poter accedere via SSH (porta TCP 22) ai due Router**
   anche se la macchina si chiama Server, si comporta da client SSH, in quanto deve poter accedere ai due router

   qui faccio uscire le connnessioni ssh verso i due router...
   - iptables -A OUTPUT -d 192.168.99.1 -o eth1 -p tcp --dport 22 -j ACCEPT
     in uscita devo poter accedere a 10.5.5.2 che è il primo router
     iptables -A OUTPUT -d 10.5.5.1 -o eth1 -p tcp --dport 22 -j ACCEPT
     e devo poter accedere a 10.5.5.1
     ...qui devo controllare che possano entrare
   - iptables -A INPUT -s 192.168.99.1 -i eth1 -p tcp --dport 22 -m state --state ESTABLISHED -j ACCEPT
   - iptables -A INPUT -s 10.5.5.1 -i eth1 -p tcp --dport 22 -m state --state ESTABLISHED -j ACCEPT

   R2
   - iptables -A INPUT -s 192.168.99.2 -i eth1 -p tcp --dport 22 -j ACCEPT
   - iptables -A OUTPUT -d 192.168.99.2 -o eth1 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

   forward per poi arrivare a R1

   - iptables -A FORWARD -s 192.168.99.2 -p tcp -i eth1 -o eth2 --dport 22 -d 10.5.5.1 -j ACCEPT
   - iptables -A FORWARD -d 192.168.99.2 -p tcp -i eth2 -o eth1 --sport 22 -s 10.5.5.1 -m state –state ESTABLISHED -j ACCEPT

   R1

   - iptables -A INPUT -s 192.168.99.2 -p tcp -i eth2 --dport 22 -j ACCEPT
     in input accetto le connessioni che arrivano dal server
   - iptables -A OUTPUT -d 192.168.99.2 -p tcp -o eth2 --sport 22 -m state –state ESTABLISHED -j ACCEPT

3. **il Client 172.16.0.2 deve poter accedere via SSH al server 192.168.99.2 utilizzando R1 e R2 come "Jump Host" nel modello SSH Tunneling**

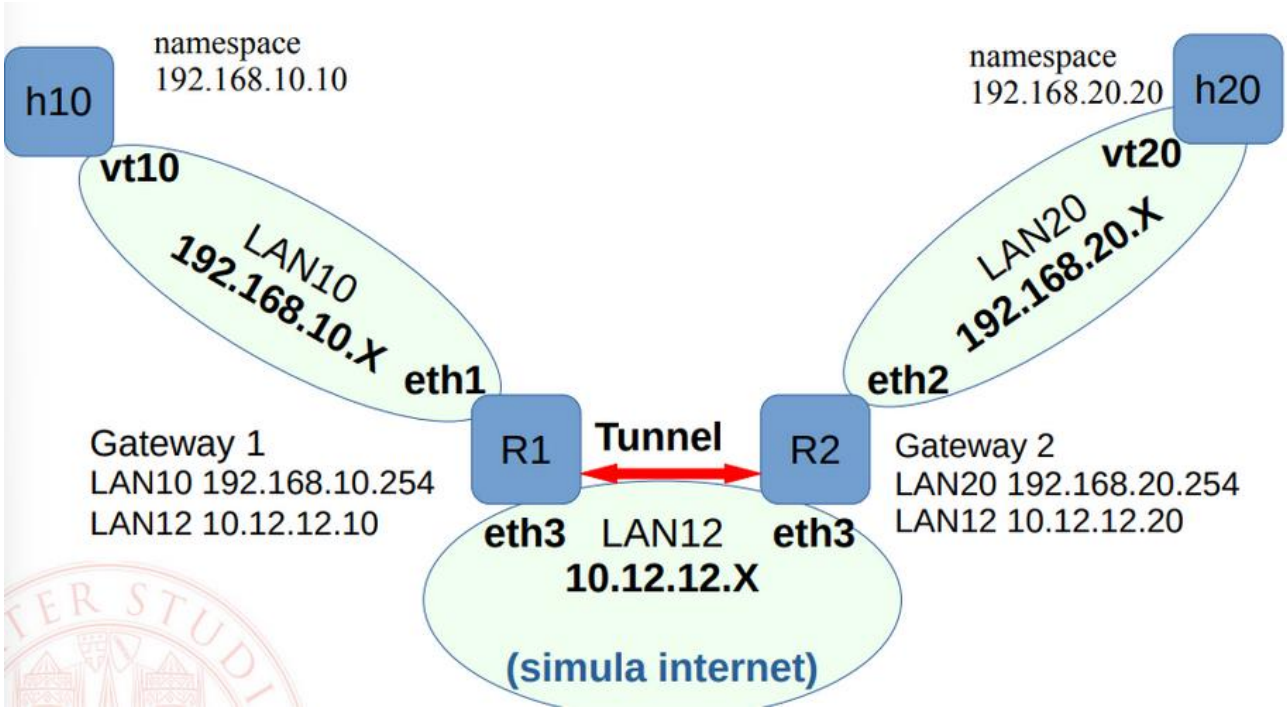
ESERCIZIO 8 FEBBRAIO 2024

Riprende un esercizio di Suricata, con il tracciato dato abbiamo visto un attacco di bruteforcing su http, dobbiamo scrivere regole per consentire tutti gli altri tipi di traffico presenti. Immaginiamo di dover installare tale configurazione su packet filter di un router.

```
# flush per partire puliti
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# router = server dhcp per le tre reti
iptables -I INPUT -p udp --sport 68 --dport 67 -j ACCEPT
iptables -I OUTPUT -p udp --dport 68 --sport 67 -m state --state ESTABLISHED -j
ACCEPT


# router = server dns per rete .3
iptables -I INPUT -p udp -s 172.23.3.187 -d 172.23.3.1 --dport 53 -j ACCEPT
iptables -I OUTPUT -p udp -d 172.23.3.187 -s 172.23.3.1 --sport 53 -m state --state
ESTABLISHED -j ACCEPT

# host su rete 3 = server NTP per client su rete 2
iptables -I FORWARD -p udp -s 172.22.2.159 -d 172.23.3.187 --dport 123 --sport 123
-j ACCEPT
iptables -I FORWARD -p udp -d 172.22.2.159 -s 172.23.3.187 --dport 123 --sport 123
-m state --state ESTABLISHED -j ACCEPT

# host su rete 3 = server ssh per client su rete 2
iptables -I FORWARD -p tcp -s 172.22.2.159 -d 172.23.3.187 --dport 22 -j ACCEPT
iptables -I FORWARD -p tcp -d 172.22.2.159 -s 172.23.3.187 --sport 22 -m state --
state ESTABLISHED -j ACCEPT

# host su rete 2 = server POP per client su rete 1
iptables -I FORWARD -p tcp -s 172.21.1.118 -d 172.22.2.159 --dport 110 -j ACCEPT
iptables -I FORWARD -p tcp -d 172.21.1.118 -s 172.22.2.159 --sport 110 -m state --
state ESTABLISHED -j ACCEPT

# host su rete 2 = server IMAPS per client su rete 1
iptables -I FORWARD -p tcp -s 172.21.1.118 -d 172.22.2.159 --dport 993 -j ACCEPT
```

```
iptables -I FORWARD -p tcp -d 172.21.1.118 -s 172.22.2.159 --sport 993 -m state --
state ESTABLISHED -j ACCEPT

# detfault deny tranne loopback (e per il contesto virtualbox, eth0)
iptables -I INPUT -i lo -j ACCEPT
iptables -I OUTPUT -o lo -j ACCEPT
iptables -I INPUT -i eth0 -j ACCEPT
iptables -I OUTPUT -o eth0 -j ACCEPT

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

ESERCIZIO SUL GITHUB DEL BRO



1. **scrivere il procedimento per riuscire a inserire, in FORWARD di R1 e R2, una regola di log nella posizione specifica subito prima delle regole DROP del traffico sulla porta 22, in modo da registrare i pacchetti che non vengono accettati dalle regole iniziali (che consentono le connessioni ssh da H10 a H20**

   droppiamo i pacchetti sulla porta 22
   iptables -I FORWARD -p tcp --dport 22 -j DROP
   iptables -I FORWARD -p tcp --sport 22 -j DROP
   consentiamo la comunicazione ssh
   iptables -I FORWARD -p tcp -s 192.168.10.10 -d 192.168.20.20 --sport 22 -j ACCEPT

iptables -I FORWARD -p tcp -s 192.168.10.20 -d 192.168.20.10 --dport 22 -m state --state ESTABLISHED -j ACCEPT
ora dobbiamo **loggare I pacchetti che sono stati droppati**, dobbiamo quindi metterla nella posizione giusta, subito prima del DROP sulla porta 22. Facciamo quindi `iptables -t FORWARD -nL --line-numbers` vediamo che va messa in posizione 3
iptables -I FORWARD 3 -j LOG --log-prefix "Pacchetto ssh droppato"

2. **rimuovere da H20 la regola di routing che utilizza R2 come default gateway, e inserire su R2 una regola di NAT tale per cui il traffico ssh in arrivo attraverso la vpn e diretto a H20 venga mascherato perché appaia proveniente da R2 stesso, impostando come indirizzo sorgente 192.168.20.254**

   per **rimuovere il gateway**
   `ip netns exec h20 ip route del default via 192.168.20.254`
   mettiamo **una regola di nat** per fare finta che il traffico ssh in arrivo dalla vpn diretto a H20 deve sembrare che proviene da R2 stesso. Mettiamo postrouting perché facciamo l'operazione dopo che il pacchetto è stato già controllato dal firewall
   iptables -t nat -I POSTROUTING -s 192.168.10.0/24 -d 192.168.20.0/24 -j SNAT --to-source 192.168.20.254

3. inserire su H10 e H20 regole per imporre vincoli su ssh analoghi a quelli inseriti sui gateway: H10 si deve solamente poter collegare via ssh a "H20" (ricordate che in realtà avendo fatto i NAT si collegherà a R1), e H20 deve solo poter ricevere connessioni ssh da "host1" (che sembreranno provenire da R2)

   NON CI STA SOLUZIONE

Le virgolette ci ricordano che sono in azione i NAT... host1 e host2 avranno visione di una realtà di rete falsata, ma l'effetto finale è che si deve fare quanto di meglio possibile per ottenere l'obiettivo richiesto. Cosa non è possibile distinguere?

COME FUNZIONANO I NAT

Esempio : abbiamo architettura host – router – internet
Nel traffico internet -> host router deve cambiare indirizzo di destinazione facendo quindi DNAT –to-destination [indirizzo dell'host]
Nel traffico host->internet router deve cambiare indirizzo sorgente facendo quindi SNAT –to-source [indirizzo del router]