Universidade Federal do Ceará

Disciplina: Segurança
Professor: Marcos Dantas Ortiz
Aluno: Rafael da Silva Gonçalves

Exercícios – Iptables

1) Explicar o que as seguintes regras fazem:

a) iptables -t filter -A INPUT -s 192.168.0.0/24 -i eth1 -j ACCEP

Essa regra permite que os pacotes originados da rede 192.168.0.0/24 e chegando pela interface eth1 sejam aceitos pelo sistema.

b) iptables -A INPUT -j LOG --log-prefix "FW INPUT"

Essa regra adiciona uma entrada na cadeia de entrada do iptables que registra um log para todos os pacotes recebidos pelo sistema.Essa é uma técnica comum para monitorar o t ráfego de entrada em um firewall.

c) iptables -I FORWARD -s 192.168.0.0/24 -d www.facebook.com -j DROP

Portanto, com este comando, qualquer tráfego originado da rede 192.168.0.0/24 e direcionado para www.facebook.com será bloqueado pelo firewall.

d) iptables -A OUTPUT -o lo -j ACCEPT

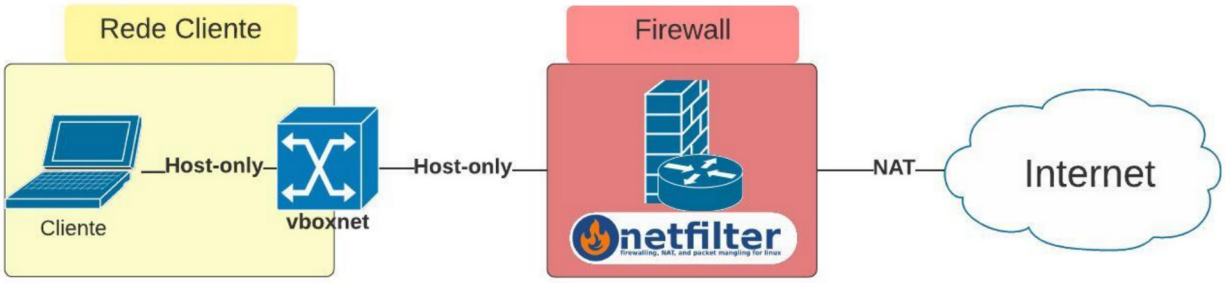
Com esse comando, qualquer tráfego de saída gerado pelo sistema e que passar pela interface de loopback será permitido e não será bloqueado pelo firewall. Isso é útil para permitir a comunicação interna entre processos e serviços no próprio sistema.

e) iptables -D FORWARD -s 192.168.13.0/24 -d www.google.com -j REJECT

Excluir uma regra da cadeia FORWARD que corresponda a pacotes originados da sub-rede 192.168.13.0/24 e destinados a www.google.com.

2) Criar regras necessárias para implantar as seguintes políticas de segurança no Firewall, utilizando cenário de rede virtualizado (virtualbox) abaixo:

Obs.: adicione nas respostas as regras e os prints dos tráfegos (quando possível)



a) Por padrão, o Firewall deve descartar todos os pacotes

Política DROP - INPUT, OUTPUT e FORWARD

```
rafael@rafael:~$ sudo iptables --policy INPUT DROP
rafael@rafael:~$ sudo iptables --policy OUTPUT DROP
rafael@rafael:~$ sudo iptables --policy FORWARD DROP
rafael@rafael:~$
```

```
rafael@rafael:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9217ms
```

b) Permitir o tráfego loopback no host do firewall.

Testar com ping (ICMP) para o próprio host – 127.0.01

```
rafael@rafael:~$ sudo iptables -A INPUT -i lo -j ACCEPT
rafael@rafael:~$ sudo iptables -A OUTPUT -o lo -j ACCEPT
```

```
rafael@rafael:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.194 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.070 ms
```

c) Permitir o acesso remoto via SSH ao Firewall.

ssh user@ip_do_firewall

```
rafael@rafael:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
rafael@rafael:~$ sudo iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

d) Permitir a realização de ping ao Firewall (ICMP echo(8) e ICMP echo reply(0)).

ping ip_do_firewall

```
rafael@rafael:~$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
rafael@rafael:~$ sudo iptables -A OUTPUT -p icmp --icmp-type 0 -j ACCEPT
```

```
clienterafael@clienterafael:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.887 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.04 ms
^C
--- 192.168.56.103 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.887/0.984/1.036/0.069 ms
```

e) Permitir que o host do firewall faça requisições DNS.

Testar com nslookup

```
rafael@rafael:~$ sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
rafael@rafael:~$ sudo iptables -A INPUT -p udp --sport 53 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
rafael@rafael:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.251.135.68
Name:   www.google.com
Address: 2800:3f0:4004:802::2004
```

f) Permitir que a **rede cliente** faça requisições HTTP.

Testar com wget

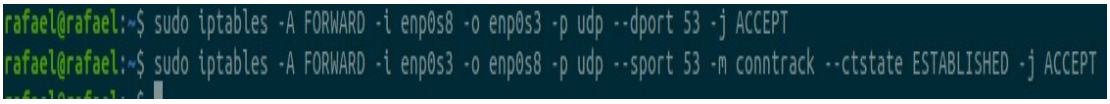
```
rafael@rafael:~$ sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -p udp --dport 53 -j ACCEPT
rafael@rafael:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -p udp --sport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT
rafael@rafael:~$ sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -p tcp --dport 80 -j ACCEPT
rafael@rafael:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j ACCEPT
clienterafael@clienterafael:~$ wget http://www.idecan.org.br/
--2023-06-18 19:50:51--  http://www.idecan.org.br/
Resolving www.idecan.org.br (www.idecan.org.br)... 191.238.223.80
Connecting to www.idecan.org.br (www.idecan.org.br)|191.238.223.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33083 (32K) [text/html]
Saving to: 'index.html'

index.html                      100%[=============================================================================>]  32,31K  --.-KB/s    in 0,06s

2023-06-18 19:50:51 (567 KB/s) - 'index.html' saved [33083/33083]
```

g) Permitir que a **rede cliente** faça requisições HTTPS.

Testar com wget

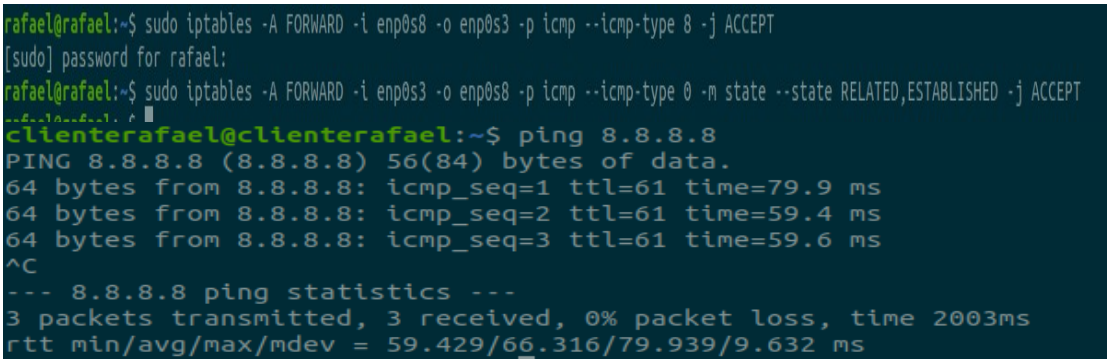
h) Permitir que a **rede cliente** faça requisições DNS.

Testar com nslookup



I) Permitir a realização de ping pela **rede cliente** (ICMP echo(8) e ICMP echo reply(0)).

ping ip_externo



Bom Trabalho!