Lab 7

Team members:

- Anoop Manikanta Subramani - 110127947
- Bindu Lokesh – 110126015

1.

```
Successfully tagged seed-router-image:latest
[07/04/24]seed@VM:~/.../router$ cd ..
[07/04/24]seed@VM:~/.../Labsetup$ docker-compose up -d
Starting seed-router        ... done
Starting hostA-10.9.0.5     ... done
Starting host2-192.168.60.6 ... done
Starting host1-192.168.60.5 ... done
Starting host3-192.168.60.7 ... done
[07/04/24]seed@VM:~/.../Labsetup$ dockps
feaf3aa3da4c  host1-192.168.60.5
bff32331c622  seed-router
d7511fb055f5  host2-192.168.60.6
df411014737b  host3-192.168.60.7
9765f668bbf1  hostA-10.9.0.5
[07/04/24]seed@VM:~/.../Labsetup$ docksh bff32331c622
root@bff32331c622:/# sudo iptables -P INPUT ACCEPT
bash: sudo: command not found
root@bff32331c622:/# iptables -P INPUT ACCEPT
root@bff32331c622:/# sudo iptables -P INPUT ACCEPT
bash: sudo: command not found
root@bff32331c622:/# iptables -P INPUT ACCEPT
root@bff32331c622:/# sudo iptables -P INPUT ACCEPT
bash: sudo: command not found
root@bff32331c622:/# iptables –P OUTPUT ACCEPT
Bad argument `–P'
Try `iptables -h' or 'iptables --help' for more information.
root@bff32331c622:/# sudo iptables -P INPUT ACCEPT
bash: sudo: command not found
root@bff32331c622:/# iptables -P INPUT ACCEPT
root@bff32331c622:/# iptables -P OUTPUT ACCEPT
root@bff32331c622:/# iptables -P FORWARD DROP
root@bff32331c622:/# iptables -P FORWARD ACCEPT
root@bff32331c622:/# iptables -P FORWARD ACCEPT
root@bff32331c622:/#
```

```
df411014737b  host3-192.168.60.7
9765f668bbf1  hostA-10.9.0.5
[07/04/24]seed@VM:~/.../Labsetup$ docksh d7511fb055f5
root@d7511fb055f5:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^C
--- 10.9.0.5 ping statistics ---
27 packets transmitted, 0 received, 100% packet loss, time 26622ms

root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.109 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.073 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.080 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.089 ms
64 bytes from 192.168.60.11: icmp_seq=6 ttl=64 time=0.146 ms
64 bytes from 192.168.60.11: icmp_seq=7 ttl=64 time=0.114 ms
c64 bytes from 192.168.60.11: icmp_seq=8 ttl=64 time=0.153 ms
^C
--- 192.168.60.11 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7146ms
rtt min/avg/max/mdev = 0.073/0.105/0.153/0.028 ms
root@d7511fb055f5:/# iptables -P FORWARD ACCEPT
root@d7511fb055f5:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^C
--- 10.9.0.5 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2037ms
```

```
root@d7511fb055f5:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=63 time=2.72 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.082 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.090 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.087 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=0.095 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=63 time=0.142 ms
^C
--- 10.9.0.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5084ms
rtt min/avg/max/mdev = 0.082/0.535/2.718/0.976 ms
root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.137 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.069 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.104 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.125 ms
c64 bytes from 192.168.60.11: icmp_seq=6 ttl=64 time=0.235 ms
^C
--- 192.168.60.11 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5121ms
rtt min/avg/max/mdev = 0.069/0.125/0.235/0.054 ms
root@d7511fb055f5:/#
```

Since the `FORWARD` policy is set to `DROP`, packets cannot pass through the router. Therefore, packets from `192.168.60.6` cannot reach `10.9.0.5` or `192.168.60.11`. By setting the `FORWARD` policy to `ACCEPT`, packets can now pass through the router, allowing `192.168.60.6` to successfully reach `10.9.0.5` and `192.168.60.11`.

2. Since the `INPUT` chain on `192.168.60.11` is configured to drop packets from `192.168.60.6` any incoming packets from `192.168.60.6` will be blocked.

```
root@bff32331c622:/# iptables -A INPUT -s 192.168.60.6 -j DROP
root@bff32331c622:/#

root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
```

Since the `OUTPUT` chain on `192.168.60.11` is configured to drop packets from `192.168.60.6`, any incoming packets from `192.168.60.6` will be blocked.

```
root@bff32331c622:/# iptables -A OUTPUT -d 10.9.0.5 -j DROP
root@bff32331c622:/#
root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
^C
--- 192.168.60.11 ping statistics ---
179 packets transmitted, 0 received, 100% packet loss, time 182345ms

root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
^C
--- 192.168.60.11 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 7158ms
```

3. Iptables -L: This command lists all rules in the default filter table. You will see all current firewall rules organized by chain

```
root@bff32331c622:/# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  host2-192.168.60.6.net-192.168.60.0  anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             client1-10.9.0.5.net-10.9.0.0
```

iptables -L --line-number :- This command lists all rules with index numbers in each chain, which is useful for operations such as deletion. The rules will be displayed with an index number at the beginning of each line, indicating their position in the chain.

```
root@bff32331c622:/# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  host2-192.168.60.6.net-192.168.60.0  anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  anywhere             client1-10.9.0.5.net-10.9.0.0
```

4.

```
root@bff32331c622:/# iptables -L INPUT --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  host2-192.168.60.6.net-192.168.60.0  anywhere
root@bff32331c622:/# iptables -D INPUT 1
root@bff32331c622:/# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
root@bff32331c622:/# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
root@bff32331c622:/#
```

By listing the rules in the INPUT chain with index numbers and then deleting a specific rule using its index, you can effectively manage and update the iptables configuration on the

router. Verification through listing the INPUT chain again ensures that the rule has been successfully deleted.

5.

```
root@bff32331c622:/# iptables -t filter -F
root@bff32331c622:/# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@bff32331c622:/#
```

By using the iptables -t filter -F command, you can flush (delete) all rules in the specified table (default filter table). Verifying with iptables -L ensures that all rules have been successfully removed, and the table is empty.

6.

```
root@bff32331c622:/# iptables -P INPUT DROP
root@bff32331c622:/# iptables -A INPUT -p tcp --dport 23 -j ACCEPT
root@bff32331c622:/# iptables -F
root@bff32331c622:/# iptables -P INPUT ACCEPT
```

```
root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
^C
--- 192.168.60.11 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3078ms

root@d7511fb055f5:/# telnet 192.168.60.11
Trying 192.168.60.11...
Connected to 192.168.60.11.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
bff32331c622 login:
```

The ping will fail. The telnet connection will succeed. The default policy for the INPUT chain is set to DROP, so all incoming connections are blocked unless explicitly allowed. The rule to accept incoming telnet connections allows only telnet traffic, while other types of traffic, such as ICMP (used by ping), are blocked. By setting the default policy for the INPUT chain to DROP and adding a rule to allow telnet connections, we can block all incoming

connections except for telnet. This demonstrates how to selectively allow specific types of traffic while blocking others using iptables. After completing the task, flushing the rules and setting the default policy back to ACCEPT ensures the router returns to its original state.

7.

```
root@bff32331c622:/# iptables -A OUTPUT -p udp --dport 53 -d 8.8.8.8 -j DROP
root@bff32331c622:/# dig www.uwindsor.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.uwindsor.ca
;; global options: +cmd
;; connection timed out; no servers could be reached

root@bff32331c622:/# dig @8.8.8.8 www.uwindsor.ca

; <<>> DiG 9.16.1-Ubuntu <<>> @8.8.8.8 www.uwindsor.ca
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached

root@bff32331c622:/# iptables -D OUTPUT -p udp --dport 53 -d 8.8.8.8 -j DROP
root@bff32331c622:/# dig www.uwindsor.ca

; <<>> DiG 9.16.1-Ubuntu <<>> www.uwindsor.ca
;; global options: +cmd
;; connection timed out; no servers could be reached

root@bff32331c622:/# dig @8.8.8.8 www.uwindsor.ca

; <<>> DiG 9.16.1-Ubuntu <<>> @8.8.8.8 www.uwindsor.ca
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

By adding a rule to the OUTPUT chain to drop outgoing DNS requests to 8.8.8.8, we can effectively block DNS queries directed to the Google DNS server. Verifying the configuration with the dig command shows that general DNS lookups succeed while those explicitly using 8.8.8.8 fail. Deleting the rule ensures no impact on subsequent experiments.

8.

```
root@bff32331c622:/# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

By adding a rule to the INPUT chain to drop incoming ICMP echo requests, you can effectively block ping requests to the router. Verifying the configuration by pinging the router from another VM shows that no replies are received, confirming that the rule is working as intended.

```
root@d7511fb055f5:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
^C
--- 192.168.60.11 ping statistics ---
31 packets transmitted, 0 received, 100% packet loss, time 30732ms
```

9.

```
root@bff32331c622:/# iptables -P INPUT DROP
root@bff32331c622:/# iptables -A INPUT -p tcp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root@bff32331c622:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
d7511fb055f5 login:
```

This connection will succeed because outgoing connections are allowed, and the response packets from 192.168.60.6 will be allowed by the rule permitting related and established connections.

```
root@d7511fb055f5:/# telnet 192.168.60.11
Trying 192.168.60.11...
```

This connection will fail because the default policy for the INPUT chain is to drop all incoming connections, and there is no existing connection to which this request is related or established.