Network Security

Homework 1

Name: Mahdi Aghajanian

---

1)

In general:

Allow: This action allows the packet to pass through the firewall and reach its destination.

Deny: This action blocks the packet from passing through the firewall.

Drop: Similar to the deny action, the drop action discards the packet without sending any response to the sender.

Reject: Unlike the drop action, the reject action sends a response to the sender indicating that the packet has been rejected.

Redirect: The redirect action forwards the packet to a different destination or interface.

Log: This action logs information about the packet, including its source, destination, and other relevant details.

Modify: The modify action allows for the modification of certain packet attributes, such as the source or destination IP address, port numbers, or protocol.

In Ubuntu:

ACCEPT: This action allows the packet to pass through the firewall without any restrictions.

DROP: When this action is applied, the packet is silently discarded without any response.

REJECT: This action is similar to DROP, but it sends a response back to the sender indicating that the packet has been rejected.

LOG: When this action is applied, the packet is logged in the system's log files.

RETURN: This action is used to exit the current chain and return to the previous chain in the firewall rules.

---

2)

The output from Nmap includes a list of scanned targets, along with additional information depending on the options used. One important piece of information is the "interesting ports table," which provides details about the ports on each target. This table includes the port number, protocol, service name, and state of each port.

The state of a port can be one of the following:

Open: This means that there is an application actively listening on the port, indicating that it is accessible and accepting connections.

Filtered: This state indicates that a firewall, filter, or other network obstacle is blocking the port, preventing Nmap from determining whether it is open or closed.

Closed: Closed ports have no application listening on them, but they could potentially open up at any time.

Unfiltered: This state indicates that the port is accessible, but Nmap is unable to determine whether it is open or closed due to factors such as packet filtering or lack of response from the target.

In addition to the interesting ports table, Nmap can provide further information about the scanned targets. This includes reverse DNS names, operating system guesses, device types, and MAC addresses.

---

3)

Masquerading is a network address translation (NAT) technique used in the filtering table of a firewall to connect a private network with the Internet. It allows hosts in a private network to use private IP addresses and communicate with the Internet using a single public IP address assigned to the router or gateway.

- Masquerading in the Filtering Table：

Masquerading is implemented in the "nat" table of the Linux kernel's netfilter framework.

The "nat" table contains rules that define changes to the source and target addresses of packets.

Masquerading is a special case of NAT that enables the translation of private IP addresses to a public IP address when packets leave the private network and go to the Internet.

It allows the router or gateway to act as an intermediary, hiding the private IP addresses of the internal network from the external network.

- Problems Masquerading Can Solve:

IP Address Conservation: Masquerading allows multiple hosts in a private network to share a single public IP address, which helps conserve public IP addresses.

Internet Access for Private Network: Masquerading enables hosts in a private network to access the Internet using a single public IP address assigned to the router or gateway.

Security and Privacy: Masquerading provides a level of security by hiding the private IP addresses of the internal network from the external network, making it harder for potential attackers to directly target individual hosts.

Simplified Network Configuration: With masquerading, hosts in the private network can use private IP addresses without requiring public IP addresses for each host. This simplifies network configuration and management.

---

4)

Port forwarding works by redirecting incoming network traffic from a specific port on the router's public IP address to a specific port on an internal device or service.

Answer in details:

Configuration:

The user configures the router to forward incoming traffic on a specific port to a designated IP address and port within the private network.

This configuration is typically done through the router's web interface or configuration settings.

Incoming Connection:

When an external device or service initiates a connection to the router's public IP address on the specified port, the router receives the incoming network traffic.

Port Forwarding:

The router checks its port forwarding configuration to determine where to forward the incoming traffic.

It modifies the destination IP address and port of the incoming traffic to match the internal IP address and port specified in the port forwarding rule.

Delivery to Internal Device:

The router forwards the modified network traffic to the designated internal device or service within the private network.

The internal device or service receives the network traffic as if it were directly connected to the external device or service.

Problem that port forwarding solve for us:

Remote Access:

Port forwarding allows users to access devices or services within their private network from outside locations.

Hosting Services:

Port forwarding enables hosting of various services, such as web servers, game servers, or FTP servers, from within a private network.

It allows external users to connect to these services using the public IP address and forwarded port.

Multi-Device Connectivity:

Port forwarding allows multiple devices within a private network to provide services on the same port.

By forwarding different external ports to different internal IP addresses and ports, multiple devices can offer services simultaneously.

Peer-to-Peer Applications:

Port forwarding is essential for peer-to-peer applications, such as torrent clients or video conferencing software.

It enables direct communication between peers by bypassing network address translation (NAT) limitations.

5)

A) iptables -A INPUT -p tcp --dport 23 -s 192.168.0.0/16 -j DROP

```
C:\Users\Padidar> telnet 192.168.83.128
Connecting To 192.168.83.128...Could not open connection to the host, on port 23: Connect failed
```

B) iptables -A INPUT -p tcp --syn -m limit --limit 20/minute --limit-burst 20 -j ACCEPT

iptables -A INPUT -p tcp --syn -j DROP

```
root@mahdi-virtual-machine:/home/mahdi/Desktop# hping3 -c 40 -i u100000 --rand-source -S 192.168.83.128
HPING 192.168.83.128 (ens33 192.168.83.128): S set, 40 headers + 0 data bytes
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.8 ms
len=40 ip=192.168.83.128 ttl=1 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=3.1 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=7.1 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=3.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=6.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=2.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=5.6 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=7 win=0 rtt=0.6 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=4.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=7.8 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=3.1 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=6.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=1.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=4.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=4.1 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=7.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=2.0 ms
len=40 ip=192.168.83.128 ttl=1 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=6.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=1.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=4.0 ms
len=40 ip=192.168.83.128 ttl=64 DF id=0 sport=0 flags=RA seq=30 win=0 rtt=4.7 ms

--- 192.168.83.128 hping statistic ---
40 packets transmitted, 21 packets received, 48% packet loss
round-trip min/avg/max = 0.6/4.3/7.8 ms
```

C) iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to-destenation 192.168.179.150:80

D) iptables -t nat -A POSTROUTING -s 192.168.179.178 -o eth0 -j SNAT --to-source 10.1.1.1

```
root@mahdi-virtual-machine:/home/mahdi/Desktop# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
```

icmp

No.	Time	Source	Destination	Protocol	Length	Info
10	2.289580354	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (no response found!)
16	3.320432326	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (no response found!)
18	4.344151816	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (no response found!)
20	5.368735980	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (no response found!)
22	6.392134869	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (no response found!)
26	7.416499167	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=6/1536, ttl=64 (no response found!)
28	8.440361988	10.1.1.1	192.168.1.5	ICMP	98	Echo (ping) request id=0x0002, seq=7/1792, ttl=64 (no response found!)

E) iptables -P OUTPUT DROP

D) iptables -A FORWARD -d 192.168.179.120 -p tcp --dport 21 -j DROP