Anish Laddha
Professor Avinash Kak
ECE404
February 19 2024

Homework 9

#1: Flush and delete all previously defined rules and chains

sudo iptables -F
sudo iptables -t nat -F
sudo iptables -X

-F in first command flushes all the chains in the default filter table
The -t nat -F in the second command flushes all the chains in the nat table
The -X deletes all the the user specified chains in the default table

#2: Write a rule that only accepts packets that originate from f1.com

sudo iptables -A INPUT -s 67.199.248.13 -j ACCEPT

The -A INPUT means that this rule will be appended to the input chain, and everything after is the rule specification. In the rule, -s specifies that the source address is the f1.com address, and the -j specifies what to do when a packet of this kind is found, in this case accept.

#3: For all outgoing packets, change their source IP address to your own machine's IP address

sudo iptables -t nat -A POSTROUTING -j MASQUERADE

The -t nat specifies that this rule will apply to the nat table, the -A POSTROUTING specifies that it will impact the POSTROUTING chain, and the -j MASQUERADE means that for any table meeting these previous requirements (in post routing/nat), it will masquerade their IP as its own.

#4: Write a rule to protect yourself against indiscriminate and nonstop scanning of ports on your machine.

sudo iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST SYN -m limit --limit 1/s -j ACCEPT

The -A means that it will impact the Forward chain, the -p tcp says that it will search the tcp protocol part of the packet, and the --tcp-flags specifies we will search SYN,ACK,FIN,RST packets to check if the SYN bit is set. If it is, the -m limit --limit 1/s says that we will limit this to 1 packet/s, and the -j ACCEPT will only accept 1 packet/second

#5: Write a rule to protect yourself from a SYN-flood Attack by limiting the number of incoming 'new connection' requests to 1 per second once your machine has reached 500 requests.

sudo iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 500 -m state --state NEW -j ACCEPT

For all incoming packets (-A input), we will search the TCP protocol part of the packet (-p tcp) for syn packets, and then limit them. The limit-burst 500 will allow 500 packets to come in, before throttling it down to the aforementioned limit. The state part specifies that these are new packets, and the -j Accepts indicates that we should accept all packets that follow these rules.

#6: Write a rule to allow full loopback access on your machine i.e. access using localhost
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT

The -A specifies the input chain, the -i lo specifies the interface via which a packet was received was loopback, and to accept all these packets. The next one is the same, except this is a rule on the output chain (i.e. allow these packets to go out).

#7: Write a port forwarding rule that routes all traffic arriving on port 8888 to port 25565.
sudo iptables -t nat -A PREROUTING -p tcp --dport 8888 -j DNAT --to-destination :25565

In the nat table, this rule applies to the prerouting chain. If the destination port of a tcp packet is 8888 (-p tcp --dport 8888) then the -j DNAT (jump) will rewrite the destination address (or port in this case) to 25565.

#8: Write a rule that only allows outgoing ssh connections to engineering.purdue.edu.
sudo iptables -A OUTPUT -p tcp --dport 22 -d engineering.purdue.edu -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -p tcp --sport 22 -s engineering.purdue.edu -m state --state ESTABLISHED,RELATED -j ACCEPT

For incoming and outgoing traffic (-A IN/OUTPUT), from ssh i.e. port 22 (-p tcp --d/sport 22), we accept packets from established connections (-m state --state ESTABLISHED,RELATED), but we also allow outgoing packets for new connections. (-m state --state NEW,ESTABLISHED)

#9: Drop any other packets if they are not caught by the above rules.
sudo iptables -A INPUT -j DROP
sudo iptables -A OUTPUT -j DROP

These specify that for the INPUT/OUTPUT chain, to drop all packets not satisfying the rules

Tables on next page

Final Filter Table:

```
anishladdha@anishladdha-QEMU-Virtual-Machine:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 500 state NEW
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  128.46.104.20        anywhere             tcp spt:ssh state RELATED,ESTABLISHED
DROP       all  --  anywhere             anywhere
ACCEPT     all  --  67.199.248.13        anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 5

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             128.46.104.20        tcp dpt:ssh state NEW,ESTABLISHED
DROP       all  --  anywhere             anywhere
```

Final Nat Table:

```
anishladdha@anishladdha-QEMU-Virtual-Machine:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere             tcp dpt:8888 to::25565

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
```