Luke Canfield ECN Login: lcanfiel Due Date: 3/28/2024

HW09 ECE 40400

```
# Rule #1
sudo iptables -F
sudo iptables -X
```

The “sudo iptables -F” command deletes all the rules in the built-in chains of the filter table. After execution the filter table will have no rules, but the chains ‘INPUT’, ‘OUTPUT’ and ‘FORWARD’ will still exist. The “sudo iptables -X” command deletes all user defined chains which are sequences of rules within an iptables that can be referenced by name. Sudo allows us to execute commands with superuser privileges to perform tasks that require elevated permission. iptables is used to configure the Linux kernel packet filtering ruleset.

```
# Rule #2
sudo iptables -A INPUT -s 67.199.248.12 -j ACCEPT
```

-A INPUT specifies that this rule should be appended to the INPUT chain which is used to filter incoming packets. -s 67.199.248.12 specifies the source IP address that the rule will match against. 67.199.248.12 is the IP address of f1.com which I found using nslookup.io online. -j specifies the action to be taken if a packet matches the rule and here case it’s set to ‘ACCEPT’

```
# Rule #3
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

-t specifies the table that the rule should be added to and here it’s set to nat (Network Address Translation) table. This table is responsible for modifying network packets. -A POSTROUTING appends the rule to the end of the ‘POSTROUTING’ chain which is traversed by packets after they have been routed. -o eth0 specifies the outgoing network interface. -j MASQUERADE is the target of the rule and is a special form of NAT that changes the source IP address of outgoing packets to the local machine’s IP address

```
# Rule #4
sudo iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
sudo iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
```

First command drops TCP packets where all flags are turned off. This usually indicates a malformed or suspicious packet so by dropping these packets we’re mitigating potential attacks. Similarly with the second command, we drop TCP packets where all flags are turned on as it suggests that the packet may be malicious and potentially part of an attack such as a constant scanning of ports. -p tcp targets TCP packets and –tcp-flags allows us to specify TCP flags to match on.

```
# Rule #5
sudo iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 500 -j
ACCEPT
```

-A INPUT specifies the rule to be added to the ‘INPUT’ chain. -p tcp specifies the rules apply to TCP packets. –syn matches packets with the SYN flag set which is the first step in 3-way handshake used to establish a connection. The -m limit –limit 1/s –limit-burst 500 uses the “limit” match extension to limit the rate of matching packets to 1 packet per second with a burst of up to 500 packets. This ensures that the system only accepts new incoming TCP connection requests at a rate of 1 per second with a burst of up to 500 packets.

```
# Rule #6
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
```

The first command allows incoming packets on the loopback interface (‘lo’) and accepts them. ‘lo’ is a virtual network interface that the operating system uses to communicate with itself. The second command allows outgoing packets on the loopback interface and accepts them. -A OUTPUT specifies the rule to the ‘OUTPUT’ chain.

```
# Rule #7
sudo iptables -t nat -A PREROUTING -p tcp --dport 8888 -j DNAT --to-destination
:25565
```

Sudo, iptables, -t nat, and -p tcp all function as described in the above sections. -A PREROUTING specifies the rule to the ‘PREROUTING’ chain which is part of the NAT table and is used to modify the packets as soon as they arrive before any routing decisions are made. –dport specifies the destination port of the packets to which the rule applies to which is 8888 in this case. -j DNAT specifies the target of the rule which is Destination Network Address Translation. When a packet matches this rule it will be modified according to the –to-destination option which specifies the new destination address and a port to which the packets should be forwarded which is 25565.

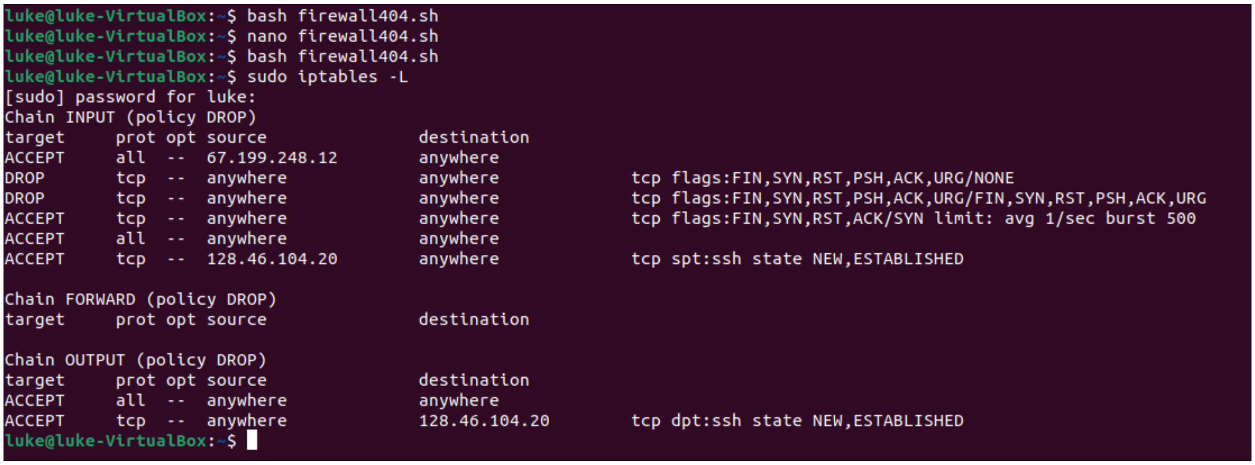
```
# Rule #8
sudo iptables -A OUTPUT -p tcp --dport 22 -d 128.46.104.20 -m state --state
NEW,ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -p tcp --sport 22 -s 128.46.104.20 -m state --state
NEW,ESTABLISHED -j ACCEPT
```

Sudo, iptables, -A OUTPUT, -p tcp, -j accept function as specified in previous sections. –dport 22 matches packets with a destination port of 22. -d 128.46.104.20 matches packets with the specified IP address which is equivalent to engineering.purdue.edu. This was found using nslookup.io. -m state –state NEW,ESTABLISHED matches packets that are in a NEW or ESTABLISHED connection state ensuring that only new outgoing SSH connections are allowed and incoming responses from established connections are accepted. The second command is the exact same expect –sport 22 matches packets with a source port of 22 ensuring that connections are allowed and responses from established connections are accepted.

```
# Rule #9
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT DROP
```

The first command sets the default policy for the INPUT chain to DROP. Any packets that do not explicitly match any defined rules will be dropped denying them access to the system. The second command sets the default policy for the FORWARD chain to DROP. It’s used for packets that are being routed through the system to another destination so any packets not specifically allowed to be forwarded will be dropped. The third command sets the default policy for the OUTPUT chain to DROP. Any output packets that do not explicitly match any of the defined rules will be dropped restricting the system from initiating communication.

Screenshot of iptables



```
luke@luke-VirtualBox:~$ bash firewall404.sh
luke@luke-VirtualBox:~$ nano firewall404.sh
luke@luke-VirtualBox:~$ bash firewall404.sh
luke@luke-VirtualBox:~$ sudo iptables -L
[sudo] password for luke:
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  67.199.248.12        anywhere
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,PSH,ACK,URG
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 500
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  128.46.104.20        anywhere             tcp spt:ssh state NEW,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination

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