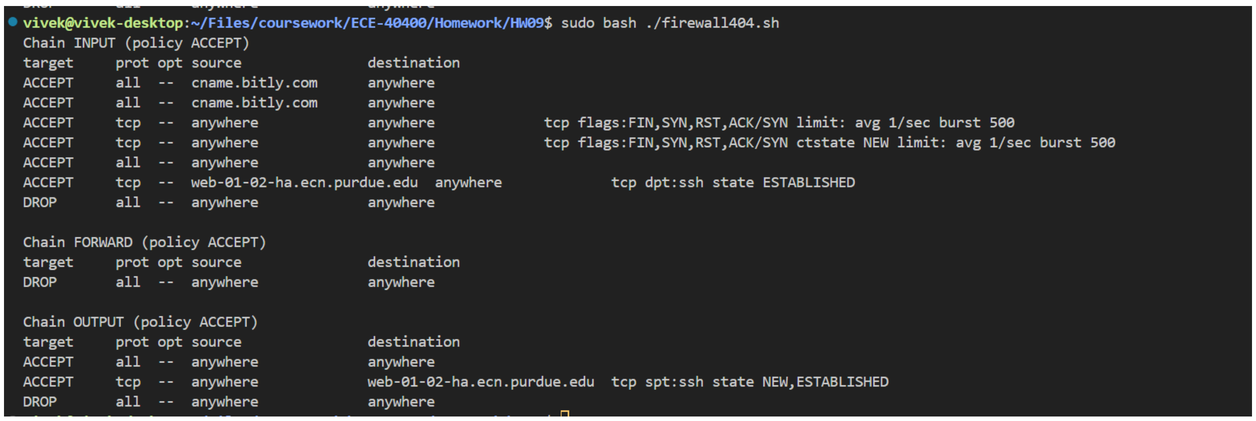
hw09_Panchagnula_Raghava

# Homework Number: HW09

# Name: Raghava Vivekananda Panchagnula

# ECN Login: rpanchag

# Due Date: 3/28/2024



```
vivek@vivek-desktop:~/Files/coursework/ECE-40400/Homework/HW09$ sudo bash ./firewall404.sh
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  cname.bitly.com      anywhere
ACCEPT     all  --  cname.bitly.com      anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 500
ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN ctstate NEW limit: avg 1/sec burst 500
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  web-01-02-ha.ecn.purdue.edu  anywhere             tcp dpt:ssh state ESTABLISHED
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             web-01-02-ha.ecn.purdue.edu  tcp spt:ssh state NEW,ESTABLISHED
DROP       all  --  anywhere             anywhere
```

Iptables –L output. It was added to the bash script for ease of use on my part, will be commenting out that line for the actual submission.

- *Flush and delete all previously defined rules and chains*

```
sudo iptables -t nat -F

sudo iptables -t mangle -F

sudo iptables -t filter -F

sudo iptables -t raw -X

sudo iptables -F

sudo iptables -X
```

This command was written to flush and delete all previously defined rules, tables, and chains. The first four clear each of the tables listed respectively. The fifth command clears the default table and all the rules defined in it. The last rule clears all user defined in the default table.

- Write a rule that only accepts packets that originate from f1.com.

```
sudo iptables -A INPUT -p tcp -s f1.com -j ACCEPT
```

This command adds a rule to the input chain. It specifies that for all incoming packets with a TCP protocol from the source domain f1.com, the firewall should accept these packets.

- *For all outgoing packets, change their source IP address to your own machine's IP address (Hint: Refer to the MASQUERADE target in the nat table)*

```
sudo iptables -t nat -A POSTROUTING -o wlo1 -j MASQUERADE
```

This command operates on the nat table and appends a rule to the postrouting chain where all packets are sent after processing before finally being sent out. This targets the wlo1 which is the outbound interface, in my case it refers to the wifi card on my computer, and it uses masquerade as the target of the rule. This means that their source address will be replaced with the address of wlo1 for every single outbound packet.

- *Write a rule to protect yourself against indiscriminate and nonstop*

scanning of ports on your machine.

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

This is a command that that takes all incoming tcp packets with the syn flag active, and limits them to 1 packet per second with a maximum burst limit of 500, which means that in a short time, after 500 packets, the rate limit starts to take hold. This is important because in day-to-day computing, dozens of packets may be sent out in short succession, and so to reduce impact on normal computing, there needs to be a certain number of packets allowed freely before imposing the limit.

- *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 conntrack --ctstate NEW -m limit --
limit 1/s --limit-burst 500 -j ACCEPT
```

This command is similar to the above, except it adds a condition to check that the packets are new initiating connection packets as well for more specificity in protecting against a SYN flood attack.

- *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
```

These commands append a rule to the input and output table, with the same input/output interfaces, which is the lo interface, to accept the packets incoming our outgoing from this. This essentially allows all connections from and to the loopback interface, which is the virtual network interface that allows communication between processes running on the same host.

- *Write a port forwarding rule that routes all traffic arriving on port 8888 to port 25565. Make sure you specify the correct table and chain. Subsequently, the target for the rule should be DNAT.*

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

This command takes any TCP traffic destined for port 8888, by adding a rule to the nat table in the prerouting chain, and uses the target DNAT, which is the destination network address translation, and changes the destination port of the packet to 25565.

- *Write a rule that only allows outgoing ssh connections to Engineering.purdue.edu.*

```
    sudo iptables -A INPUT -p tcp --dport 22 -s engineering.purdue.edu -m state
--state ESTABLISHED -j ACCEPT

    sudo iptables -A OUTPUT -p tcp --sport 22 --dst engineering.purdue.edu -m
state -state NEW,ESTABLISHED -j ACCEPT
```

There are two commands for this requirement. The first is for the incoming ssh packets. Since the requirement was to only allow outgoing ssh packets, this rule only allows established tcp packets on port 22 from engineering.purdue.edu. The second rule accepts established and new connections to be sent as outputs.

- *Drop any other packets*

```
sudo iptables -A INPUT -j DROP

sudo iptables -A OUTPUT -j DROP

sudo iptables -A FORWARD -j DROP
```

This set of commands drops all other packets on the input, output, and forward chains.