Firewall Assignment

Task 1:

We have set-up a network of three VMs using Vbox. Here are the three VMs and the commands run on them:

Host 1:
Public interface (eth0): 172.24.1.1
This is the host which is considered to be on the public internet. This is created using the Bridged networking mode in Vbox. It gets an IP address on the host's network.

Firewall (Host 2):
Public interface (eth0): 172.24.1.2 (example)
Internal Network (eth1): 10.10.10.1
This acts like a router. It has two interfaces: one connected to host through bridge network - eth0, and another on an 'internal network' (another VBox networking mode) - eth1. The firewall port forwards all the connections from it to Host 3 on the internal network. The following iptables rule:

1. `sudo nano /etc/sysctl.conf` and write `net.ipv4.ip_forward=1`
2. sudo iptables -t nat -A PREROUTING -i eth0 -j DNAT --to-destination 10.10.10.2. This will DNAT all the traffic from firewall to host 2.
3. sudo iptables -t nat -A POSTROUTING -o eth1 -d 10.10.10.2 -j SNAT --to-source 10.10.10.1. This is required so that Host 2 replies back to Firewall instead of to the original ip.

Host 3:
Internal Network (eth0): 10.10.10.2
This is the VM where our server will be running. It is not publicly accessible and can be only accessed through the router/firewall.

We have a working router setup now. Host 1 can reach Host 3 by accessing Firewall's public IP. We run our custom firewall on Host 2 using steps discussed in Task 2.

Task 2:

We have used NFQUEUE for forwarding all the incoming packets to the host to the firewall code.
This rule

```
iptables -I INPUT -j NFQUEUE --queue-num 1
```

Is ran to forward all packets from iptables INPUT policy to the NetFilter Queue 1.

In firewall.py, when a packet enters the queue, a callback function is called which validates the packet and performs the required action. First we check if packet has valid headers.
If packet is valid, then rules are iterated over. When a rule succeeds remaining rules are not checked and the required action is performed . This checking happens sequentially. If a packet is being checked, others need to wait for it to finish.

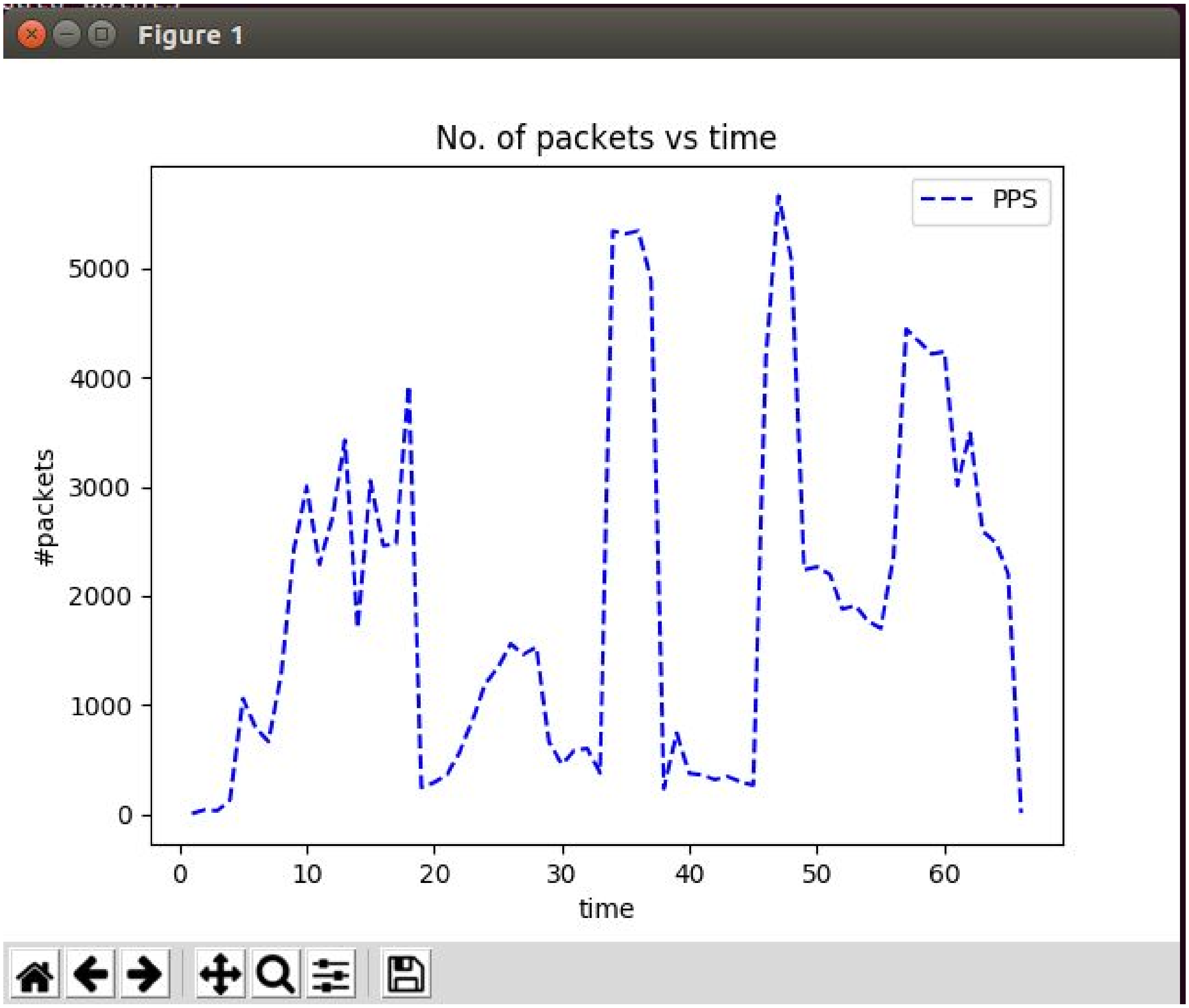
Interactive CLI
Program cli.py is run in parallel which makes a file database.json. When a rule is added via CLI, we add a new entry in the file.
In firewall callback this file is read, thereby reading the latest rules.

Task 3:

Calculation of PPS : When firewall finishes, this event is logged in log.txt along with the timestamp. When we want the PPS, this file is read & a graph is plotted depicting the PPS over time interval of 1 second.

Testing: iperf tool was used to make the required test-bed and get traffic.

MAXIMUM PPS over accepting all packets



Configuration

```
iperf3 --server --port 5201  -f K -V
iperf3 --client 172.24.1.102 --port 5201 --version4   -f K -V  --bandwidth 50M --time 60 --parallel 30
```

Results:

After testing with various bandwidth and parallel connections : the MAX PPS obtained was 5777. This result and above graph was obtained on accepting all the inputs obtained by iperf.

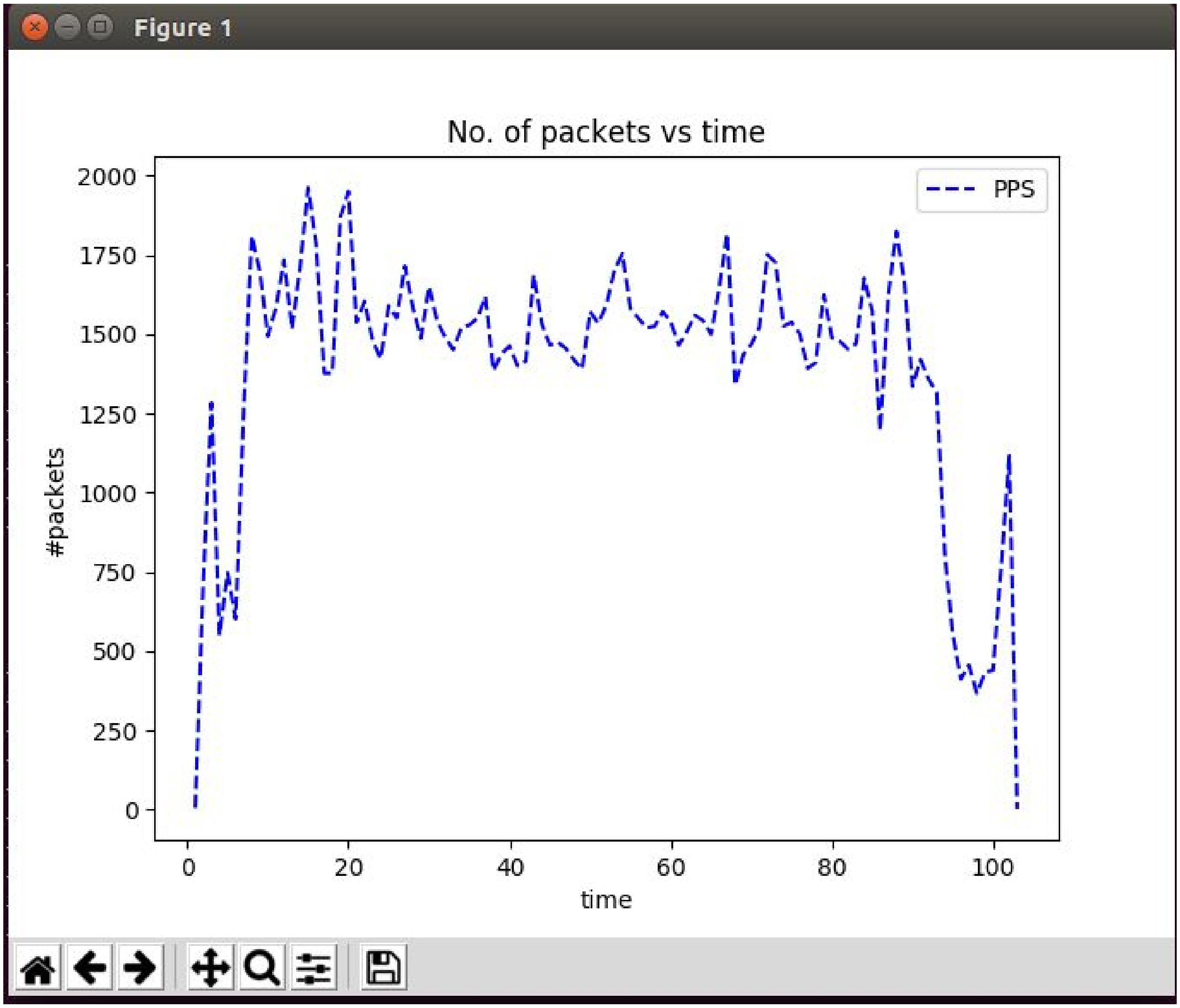
Various no of rules matching with traffic equally distributed

Configuration
The bash files used are submitted along with this assignment to make the configuration.

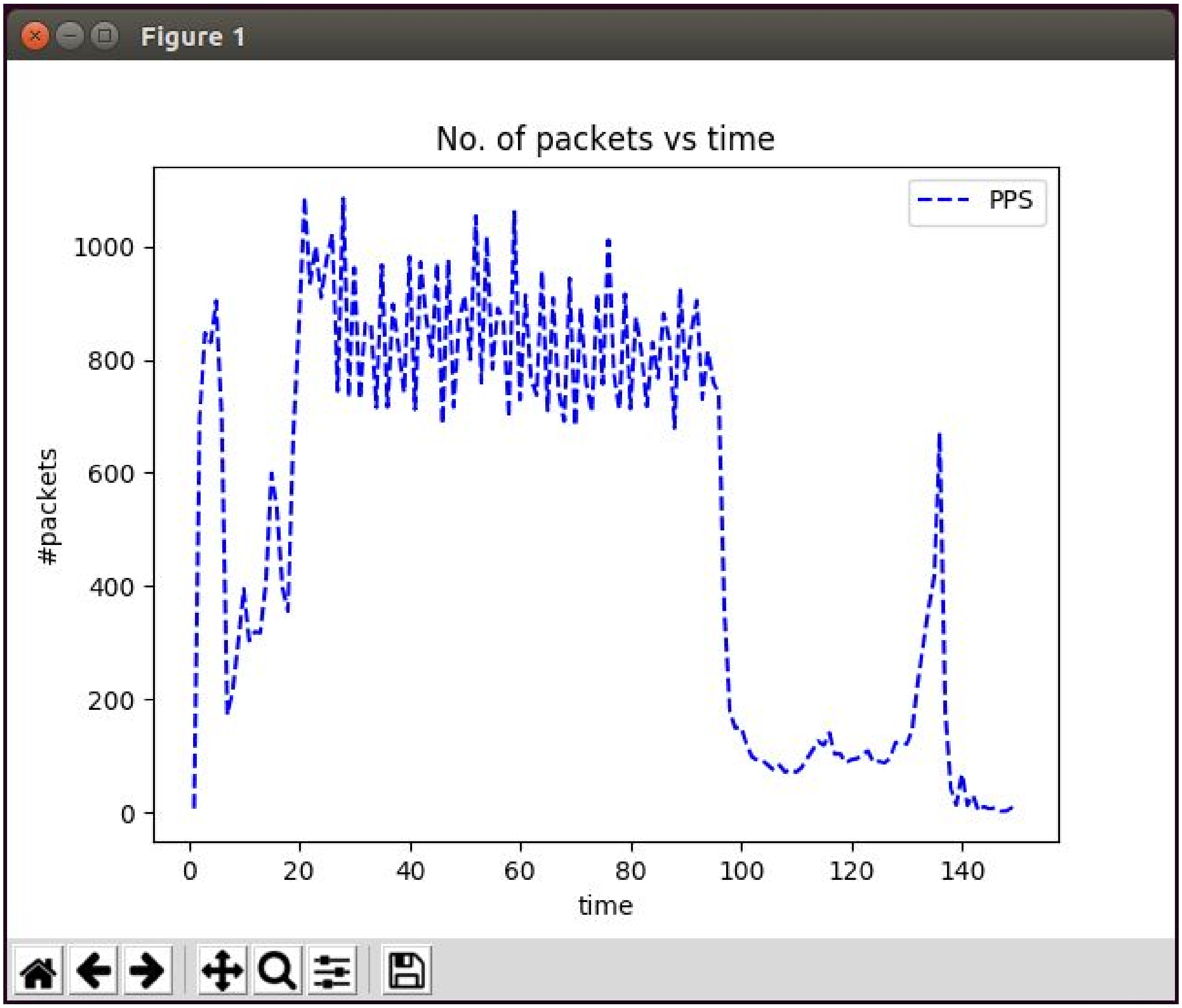
Testbed:
10, 25, 50 TCP rules were made with various destination port. Then those many iperf servers were spawned at the mentioned port. IPerf clients were made to generate traffic on the mentioned ports.

Results

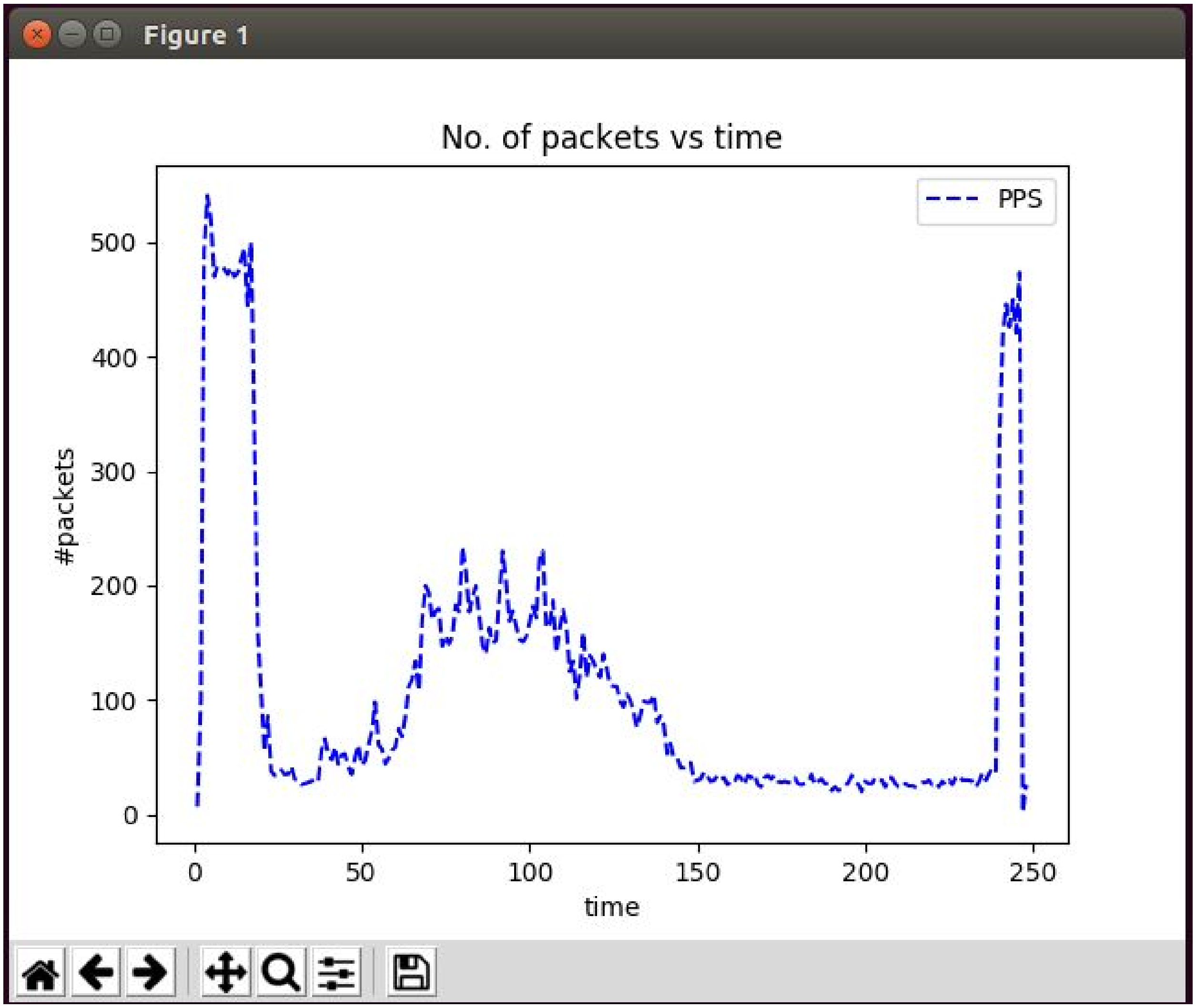
No. of Rules = 10



No. of Rules = 25



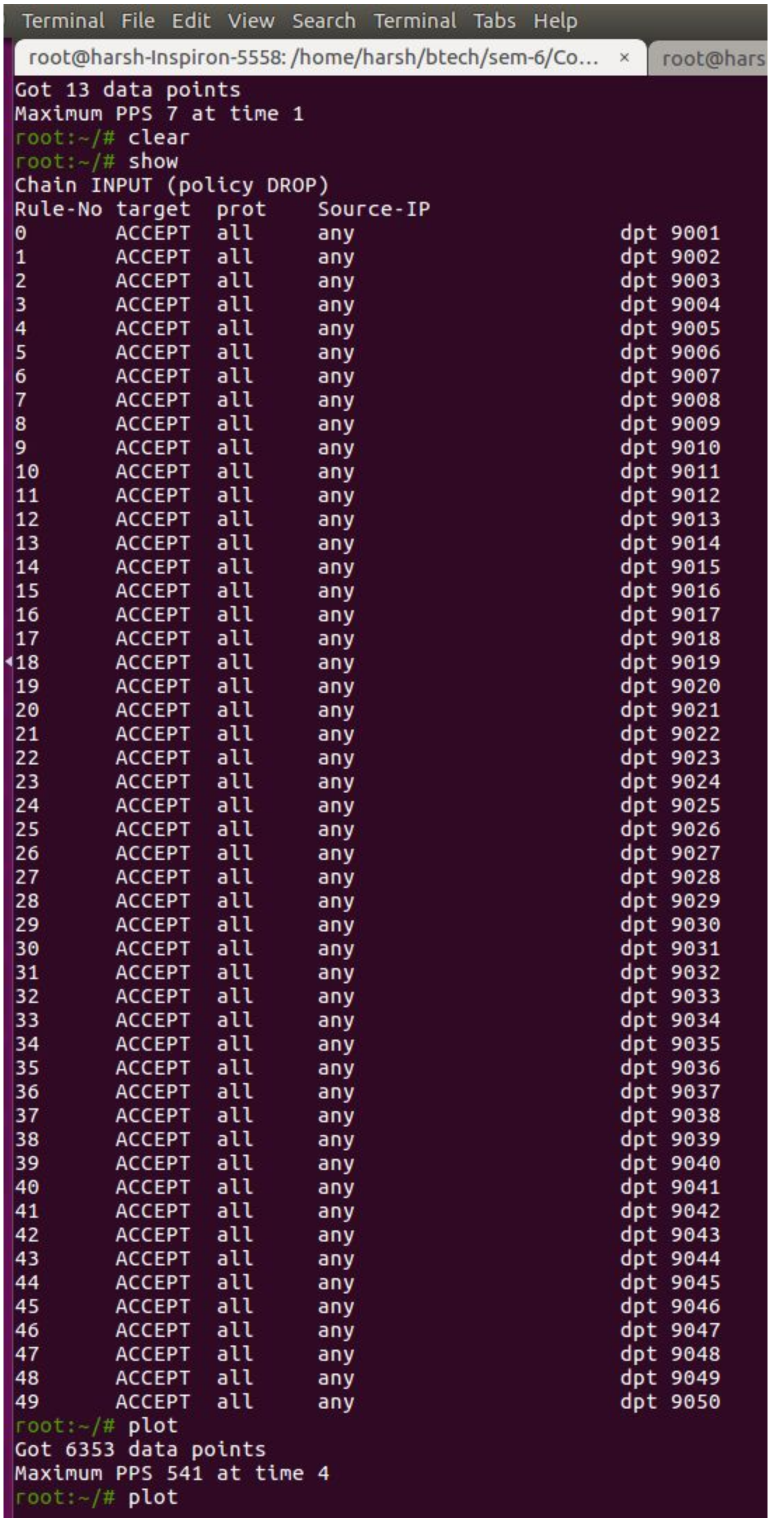
Maximum PPS :: 1088



Max PPS :: 541

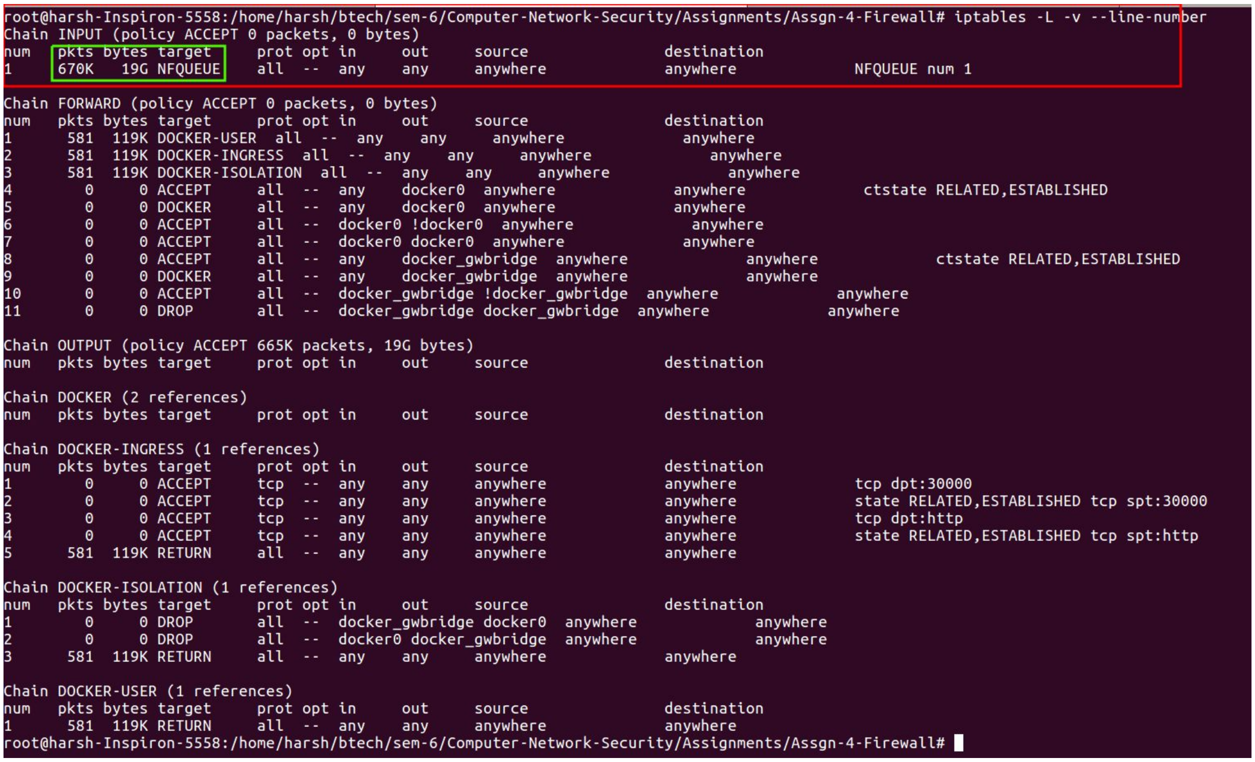
Observations

With more rules, PPS decreases.
This is because more computation needs to be done per packet. And next packet only goes for processing once previous callback returns
Since results for each experiment were conducted for


Terminal File Edit View Search Terminal Tabs Help
root@harsh-Inspiron-5558: /home/harsh/btech/sem-6/Co… ×
root@hars

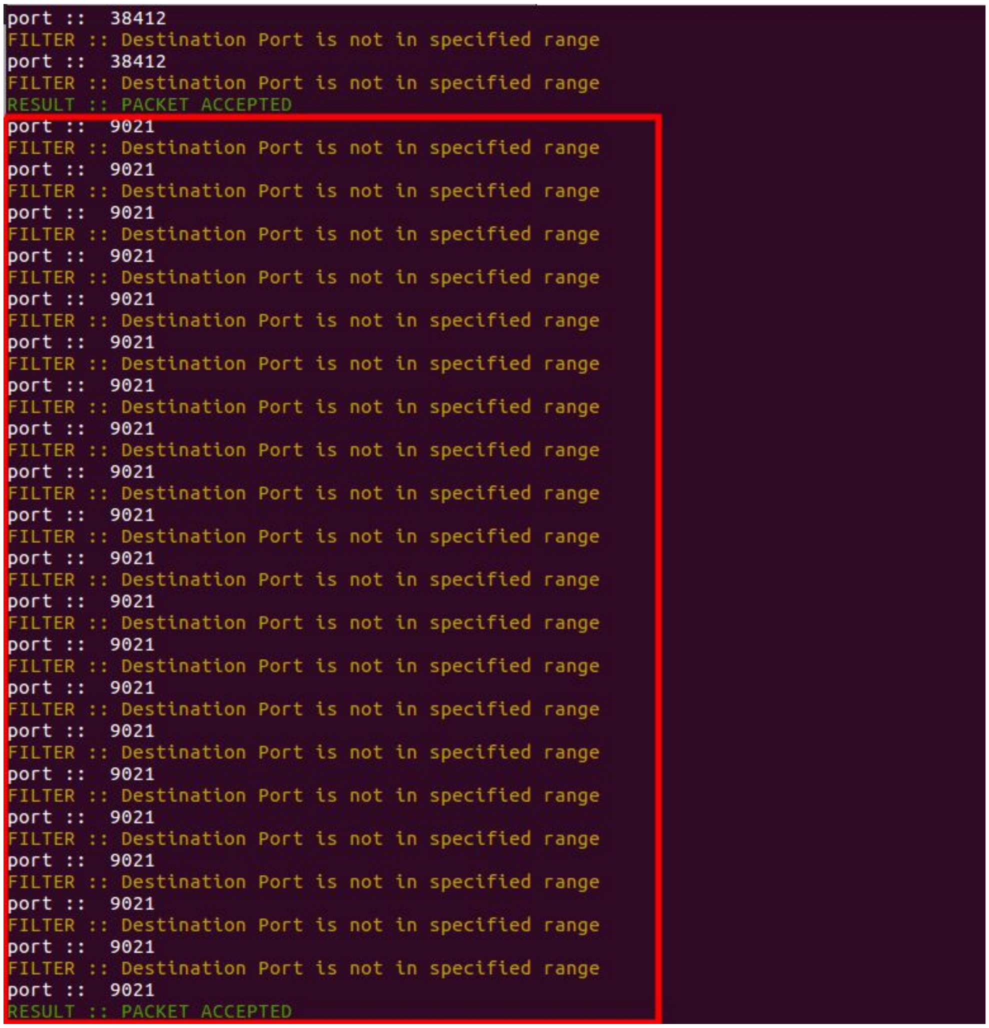
```
Got 13 data points
Maximum PPS 7 at time 1
root:~/# clear
root:~/# show
Chain INPUT (policy DROP)
Rule-No target  prot    Source-IP
0       ACCEPT  all     any                     dpt 9001
1       ACCEPT  all     any                     dpt 9002
2       ACCEPT  all     any                     dpt 9003
3       ACCEPT  all     any                     dpt 9004
4       ACCEPT  all     any                     dpt 9005
5       ACCEPT  all     any                     dpt 9006
6       ACCEPT  all     any                     dpt 9007
7       ACCEPT  all     any                     dpt 9008
8       ACCEPT  all     any                     dpt 9009
9       ACCEPT  all     any                     dpt 9010
10      ACCEPT  all     any                     dpt 9011
11      ACCEPT  all     any                     dpt 9012
12      ACCEPT  all     any                     dpt 9013
13      ACCEPT  all     any                     dpt 9014
14      ACCEPT  all     any                     dpt 9015
15      ACCEPT  all     any                     dpt 9016
16      ACCEPT  all     any                     dpt 9017
17      ACCEPT  all     any                     dpt 9018
18      ACCEPT  all     any                     dpt 9019
19      ACCEPT  all     any                     dpt 9020
20      ACCEPT  all     any                     dpt 9021
21      ACCEPT  all     any                     dpt 9022
22      ACCEPT  all     any                     dpt 9023
23      ACCEPT  all     any                     dpt 9024
24      ACCEPT  all     any                     dpt 9025
25      ACCEPT  all     any                     dpt 9026
26      ACCEPT  all     any                     dpt 9027
27      ACCEPT  all     any                     dpt 9028
28      ACCEPT  all     any                     dpt 9029
29      ACCEPT  all     any                     dpt 9030
30      ACCEPT  all     any                     dpt 9031
31      ACCEPT  all     any                     dpt 9032
32      ACCEPT  all     any                     dpt 9033
33      ACCEPT  all     any                     dpt 9034
34      ACCEPT  all     any                     dpt 9035
35      ACCEPT  all     any                     dpt 9036
36      ACCEPT  all     any                     dpt 9037
37      ACCEPT  all     any                     dpt 9038
38      ACCEPT  all     any                     dpt 9039
39      ACCEPT  all     any                     dpt 9040
40      ACCEPT  all     any                     dpt 9041
41      ACCEPT  all     any                     dpt 9042
42      ACCEPT  all     any                     dpt 9043
43      ACCEPT  all     any                     dpt 9044
44      ACCEPT  all     any                     dpt 9045
45      ACCEPT  all     any                     dpt 9046
46      ACCEPT  all     any                     dpt 9047
47      ACCEPT  all     any                     dpt 9048
48      ACCEPT  all     any                     dpt 9049
49      ACCEPT  all     any                     dpt 9050
root:~/# plot
Got 6353 data points
Maximum PPS 541 at time 4
root:~/# plot
```

Rule Table containing 50 rules



```
root@harsh-Inspiron-5558:/home/harsh/btech/sem-6/Computer-Network-Security/Assignments/Assgn-4-Firewall# iptables -L -v --line-number
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1     670K   19G NFQUEUE    all  --  any    any     anywhere             anywhere             NFQUEUE num 1

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1      581  119K DOCKER-USER  all  --  any    any     anywhere             anywhere
2      581  119K DOCKER-INGRESS  all  --  any    any     anywhere             anywhere
3      581  119K DOCKER-ISOLATION  all  --  any    any     anywhere             anywhere
4        0     0 ACCEPT     all  --  any    docker0  anywhere             anywhere             ctstate RELATED,ESTABLISHED
5        0     0 DOCKER     all  --  any    docker0  anywhere             anywhere
6        0     0 ACCEPT     all  --  docker0 !docker0  anywhere             anywhere
7        0     0 ACCEPT     all  --  docker0 docker0  anywhere             anywhere
8        0     0 ACCEPT     all  --  any    docker_gwbridge  anywhere             anywhere             ctstate RELATED,ESTABLISHED
9        0     0 DOCKER     all  --  any    docker_gwbridge  anywhere             anywhere
10       0     0 ACCEPT     all  --  docker_gwbridge !docker_gwbridge  anywhere             anywhere
11       0     0 DROP       all  --  docker_gwbridge docker_gwbridge  anywhere             anywhere

Chain OUTPUT (policy ACCEPT 665K packets, 19G bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain DOCKER (2 references)
num   pkts bytes target     prot opt in     out     source               destination

Chain DOCKER-INGRESS (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:30000
2        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             state RELATED,ESTABLISHED tcp spt:30000
3        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
4        0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             state RELATED,ESTABLISHED tcp spt:http
5      581  119K RETURN     all  --  any    any     anywhere             anywhere

Chain DOCKER-ISOLATION (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 DROP       all  --  docker_gwbridge docker0  anywhere             anywhere
2        0     0 DROP       all  --  docker0 docker_gwbridge  anywhere             anywhere
3      581  119K RETURN     all  --  any    any     anywhere             anywhere

Chain DOCKER-USER (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1      581  119K RETURN     all  --  any    any     anywhere             anywhere
root@harsh-Inspiron-5558:/home/harsh/btech/sem-6/Computer-Network-Security/Assignments/Assgn-4-Firewall#
```

In the green box, the total bytes of packet is seen that was obtained while testing and getting the results

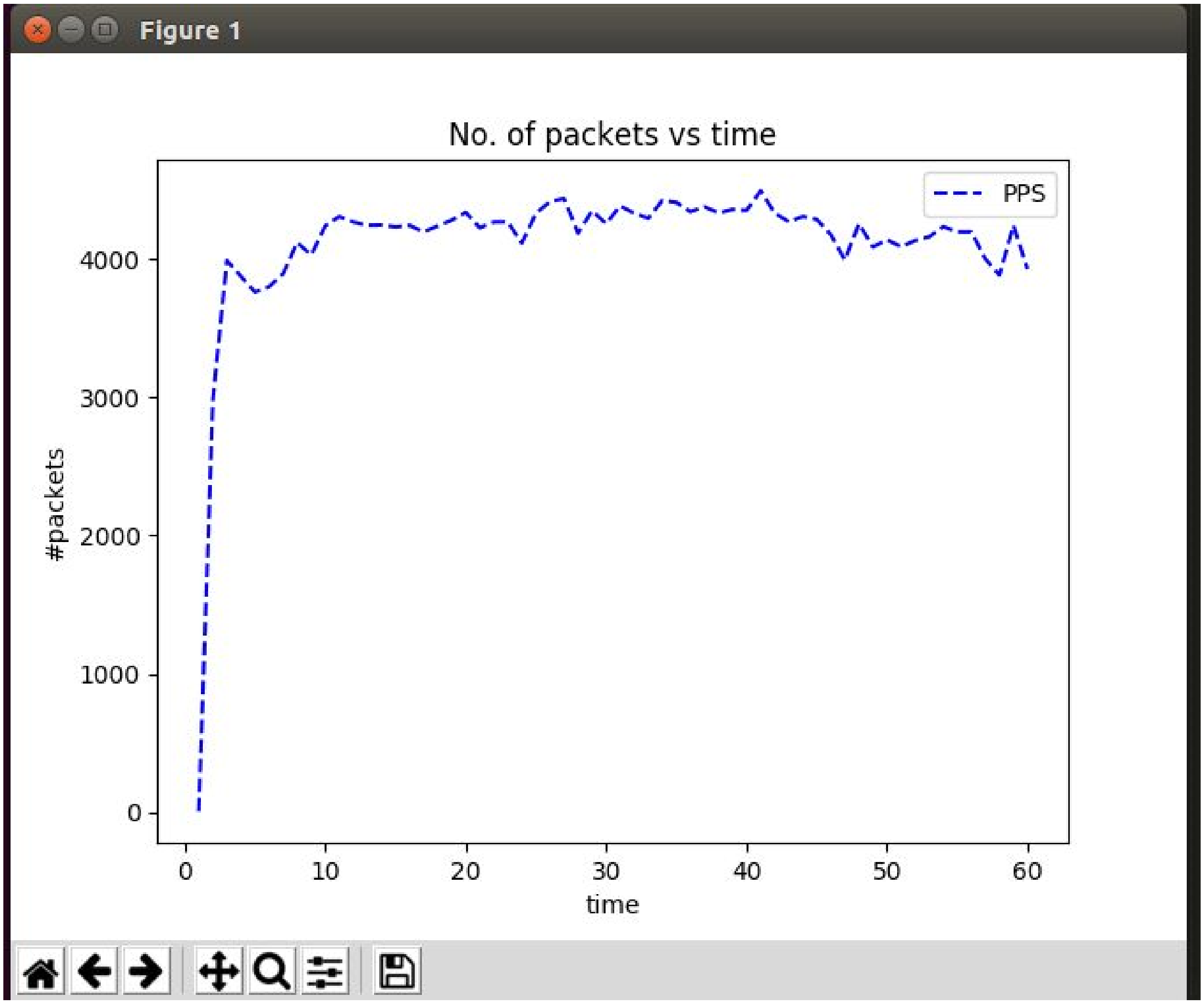


```
port ::  38412
FILTER :: Destination Port is not in specified range
port ::  38412
FILTER :: Destination Port is not in specified range
RESULT :: PACKET ACCEPTED
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
FILTER :: Destination Port is not in specified range
port ::  9021
RESULT :: PACKET ACCEPTED
```

Since callback is sequential, we can see that ports are checked from 9001 to 9020, before an acceptance rule is passed

Many matching fields

RULE :: ADD -dport 5201 -s 127.0.0.1 -p tcp -j accept
iperf3 --server --port 5201 -f K -V
iperf3 --client 127.0.0.1 --port 5201 --version4   -f K -V --bandwidth 50M --time 60 --parallel 30
Result ::



Max PPS obtained was 4495.

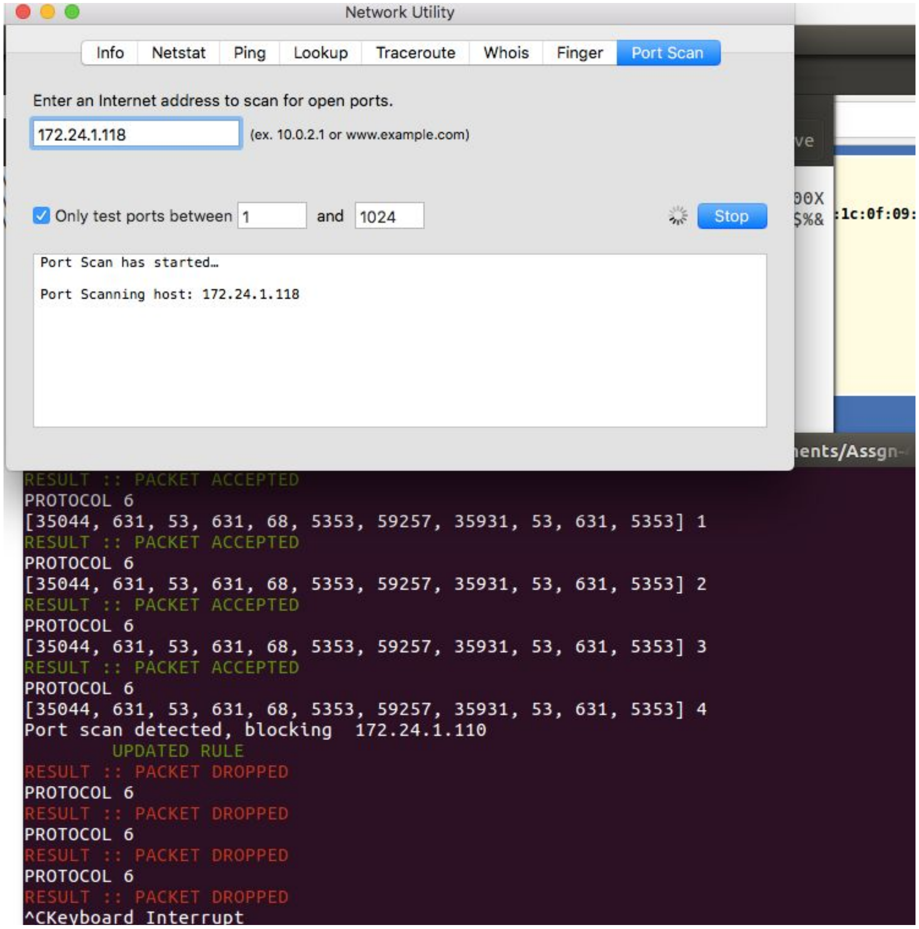
This is less than the rule of Accepting all(5777) because more time is spent per packet.

Task 4 - Detecting Port Scanning Attacks

The firewall is capable of detecting and blocking port scanning attacks. It does in the following manner. The firewall records all the invalid port scans (ie. the port scans on the ports that were closed at the the time of the request) for every IP address in the last one minute. If the number of invalid port scans exceeds a certain threshold in the last minute, the IP address is blocked. It gets the list of open ports by getting all the open UNIX sockets (similar to `netstat -lntu`), and if we receive a request for port number which is not in that list, we increment the invalid port scans count. To test this:

- Start the firewall, and add the rule “ADD -j ACCEPT” through cli.py.
- Run a port scan using any utility

You should see that it gets blocked and a new DROP rule is added to the firewall.



Enter an Internet address to scan for open ports.

172.24.1.118 (ex. 10.0.2.1 or www.example.com)

☑ Only test ports between 1 and 1024

```
Port Scan has started…

Port Scanning host: 172.24.1.118
```

```
RESULT :: PACKET ACCEPTED
PROTOCOL 6
[35044, 631, 53, 631, 68, 5353, 59257, 35931, 53, 631, 5353] 1
RESULT :: PACKET ACCEPTED
PROTOCOL 6
[35044, 631, 53, 631, 68, 5353, 59257, 35931, 53, 631, 5353] 2
RESULT :: PACKET ACCEPTED
PROTOCOL 6
[35044, 631, 53, 631, 68, 5353, 59257, 35931, 53, 631, 5353] 3
RESULT :: PACKET ACCEPTED
PROTOCOL 6
[35044, 631, 53, 631, 68, 5353, 59257, 35931, 53, 631, 5353] 4
Port scan detected, blocking  172.24.1.110
        UPDATED RULE
RESULT :: PACKET DROPPED
PROTOCOL 6
RESULT :: PACKET DROPPED
PROTOCOL 6
RESULT :: PACKET DROPPED
PROTOCOL 6
RESULT :: PACKET DROPPED
^CKeyboard Interrupt
```

As can be seen, the ip address of the attacker was blocked after it detected 5 consecutives failed pings.

Reference

https://pypi.python.org/pypi/NetfilterQueue : for using NFQUEUE
http://iperf.fr/ : for testing & benchmarking