COMP 8006

Assignment 2

Standalone Linux Firewall

Brij Shah – A00717689
Ramzi Chennafi - A00825005

Table of Contents

Overview

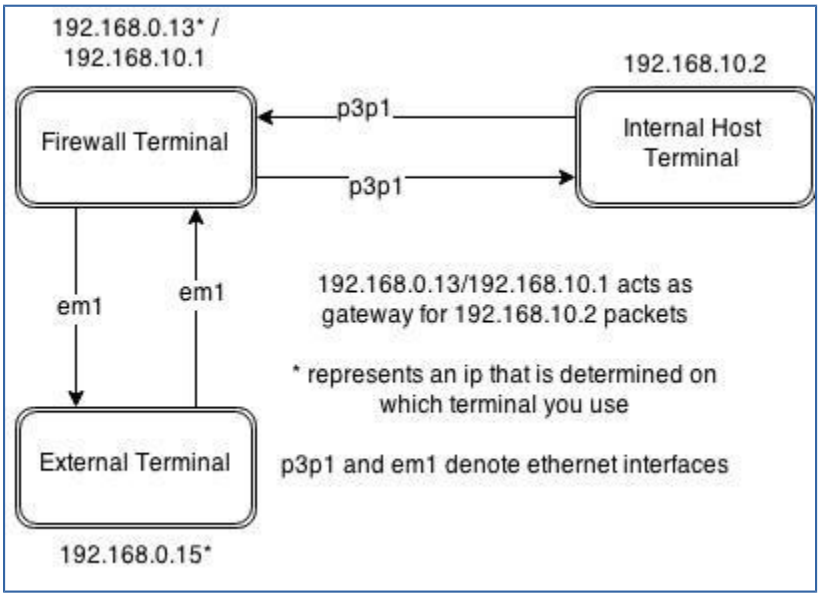
The objective of this assignment was to create and test a standalone Linux firewall and packet filter. The firewall was created using NETFILTER and follows a set of pre established rules. The following document will prove the validity of the firewall. The firewall is implemented to allow users to hide and protect servers from external maliciousness as well as allowing TCP, UDP, ICMP ports to allow or block traffic on.

Firewall Design

Assignment Constraints

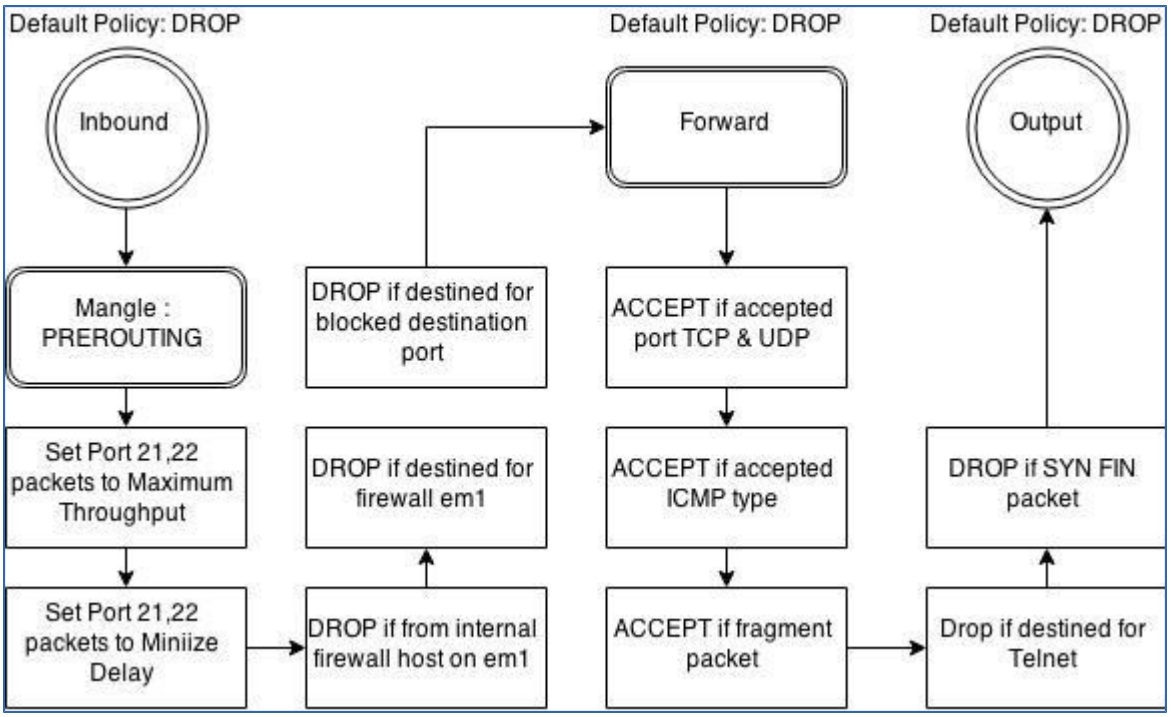
- The firewall/packet filter must be designed and implemented using NETFILTER
- The script has two sections:
    - User Configurable
    - Implementation Section(DO NOT TOUCH)
- The user Configurable section allows modifications to the following parameters:
    - Name & Location of the utility used to implement firewall
    - Internal Network Address Space & Network Device
    - Outside Address Space & Network Device
    - TCP Services which will be allowed
    - UDP Services which will be allowed
    - ICMP Services which will be allowed
- Only NEW and ESTABLISHED traffic gets through the firewall(STATEFUL Filtering)
- Ensures that traffic coming the ‘wrong way’ is rejected(inbound connection requests), unless otherwise permitted
- Firewall includes test script to validate rules implemented

Design



Above is the design of the of the network. The internal host exists as a subnet behind the firewall terminal. The external terminal will simply perform testing.

The Rules



Project Files

Setup_Firewall.sh
Setup_Network.sh
Test_Network.sh
Assignment2Report.pdf

Setup

The following section explains how to setup the network and get the firewall running using the scripts provided. There are also instructions on use the the script.

How to setup Firewall or Host

In Terminal,

1. In terminal type: chmod +x Setup_Network && ./Setup_Network.sh
2. Change the user configurable settings in Setup_Network and Setup_Firewall to fit your network setup.
3. You will be prompted with a list to chose from

a. Choose ‘Firewall’ to setup machine as firewall
b. Choose ‘Host’ to setup machine as internal host
c. Choose ‘Test’ to setup the machine as a testing terminal.
d. Choose ‘Reset’ to restore the machine to its default settings.

How to run Test Script

Change the user configurable settings in Test_Firewall.sh to fit your network setup.
In terminal type: chmod +x Test_Network && ./Test_Firewall.sh

Test Cases

Test #	Test Name	Description	Tools Used	Pass/Fail
1	Allowing TCP	Checking if traffic on specified tcp ports is allowed.	wireshark, hping, iptables	Pass. See below for detailed results
2	Allowing UDP	Checking if traffic on specified udp ports is allowed.	wireshark, hping, iptables	Pass. See below for detailed results
3	Blocking ICMP	Checking if specified ICMP packets are allowed.	wireshark, hping, iptables	Pass. See below for detailed results
4	Internal from External	Testing if an external computer can send traffic with an internal IP.	wireshark, hping, iptables	Pass. See below for detailed results
5	Fragments	Checking if the network will accept fragments.	wireshark, hping, iptables	Pass. See below for detailed results
6	SYN on a High Port	Checking if SYNs to high ports will be accepted.	wireshark, hping, iptables	Pass. See below for detailed results
7	Blocked TCP	Checking if specified TCP ports are blocked.	wireshark, hping, iptables	Pass. See below for detailed results
8	SYN FIN Attack	Checking if a SYN FIN attack can be done.	wireshark, hping, iptables	Pass. See below for detailed results
9	Telnet	Checking if Telnet is blocked.	wireshark, hping, iptables	Pass. See below for detailed results
10	Specific Blocked Ports	Checking if specific ports are blocked	wireshark, hping, iptables	Pass. See below for detailed results
11	Mangle Mods	Checking if the fields are modified by mangle properly.	wireshark, hping3, iptables	Pass. See below for detailed results

Detailed Results

Test #1

```
[root@DataComm FirewallRouting]# hping3 192.168.10.2 -S -p 80 -c 1
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=63102 sport=80 flags=RA seq=0 win=0 rtt=1.1
ms

--- 192.168.10.2 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.1 ms
[root@DataComm FirewallRouting]#
```

In this test we sent a SYN packet to port 80 to the internal host. As you can see the packet was transmitted and received. The rule we are testing against is customizable TCP acceptance, of which, we set 80 to accept all traffic.

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:23
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:23
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:89
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp spt:89
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:86
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp spt:86
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:88
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp spt:88
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:443
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp spt:443
       1       40 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:80
       1       40 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp spt:80
```

As you can see, port 80 on the forward chain accepted one packet on both the source and destination ports, showing a successful transfer.

Test #2

In this test we are checking UDP ports. Here we had the UDP port 89 opened on the firewall. As you can see, we sent packets and received a response from the internal machine.

```
[root@DataComm FirewallRouting]# hping3 192.168.10.2 -2 -c 5 -k -p 89
HPING 192.168.10.2 (em1 192.168.10.2): udp mode set, 28 headers + 0 data
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

No.	Time	Source	Destination	Protoco
1	0.000000000	fe80::20e:cff:fe51:28b4	ff02::2	ICMPv6
2	3.112197000	192.168.0.12	192.168.10.2	UDP
3	3.112256000	192.168.10.2	192.168.0.12	ICMP
4	4.112233000	192.168.0.12	192.168.10.2	UDP
5	4.112286000	192.168.10.2	192.168.0.12	ICMP
6	5.112331000	192.168.0.12	192.168.10.2	UDP
7	5.112380000	192.168.10.2	192.168.0.12	ICMP
8	6.112440000	192.168.0.12	192.168.10.2	UDP
9	6.112487000	192.168.10.2	192.168.0.12	ICMP
10	7.112540000	192.168.0.12	192.168.10.2	UDP
11	7.112588000	192.168.10.2	192.168.0.12	ICMP

As you can see, this wireshark capture from the internal host shows that it received the UDP packet from the testing terminal and responded to it. If we look at our IP tables, we should also see accepted UDP packets.

```
    0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABL
    0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABL
    0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ESTABL
    5      140 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:89
    0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:89
```

As you can see, the 5 packets were accepted by the firewall in the forwarding chain.

Test #3

In this test, we sent 3 ICMP packet of type 3

```
[root@DataComm ~]# hping3 192.168.10.2 --icmp --icmptype 3
HPING 192.168.10.2 (em1 192.168.10.2): icmp mode set, 28 headers + 0 data bytes
^C
--- 192.168.10.2 hping statistic ---
7 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@DataComm ~]#
```

If we check the IPTABLES entry, we will see that 7 ICMP packets were accepted. This test was a success.

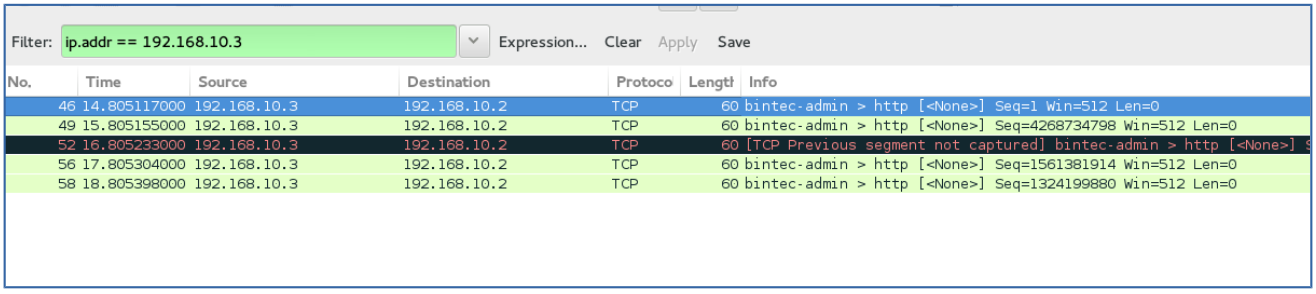
```
    0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:68
    0        0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 1
    0        0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 2
    7      392 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 3
```

If we look to the IP tables entry we can see that ICMP packet type 3 was in fact accepted.

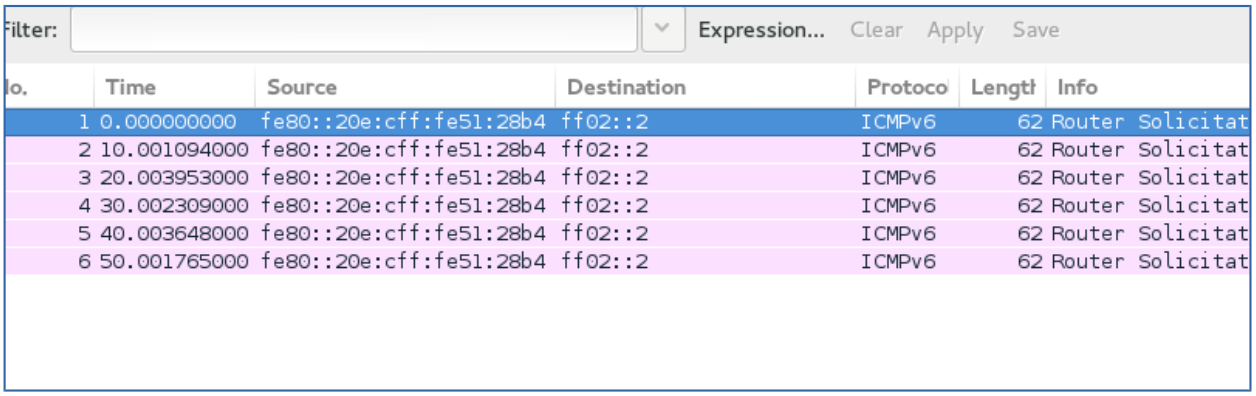
Unfortunately, this rule does not appear to be tracked by iptables, even though the packets were dropped no counters go up on the iptables entries.

Test #4

In this test, we sent a packet from an external terminal with a spoofed IP of 192.168.10.3 destined for the internal host.



This capture was taken from the firewall machine, if we look to the internal machine over wireshark we see that there is a complete lack of packets coming to it.



Test#6

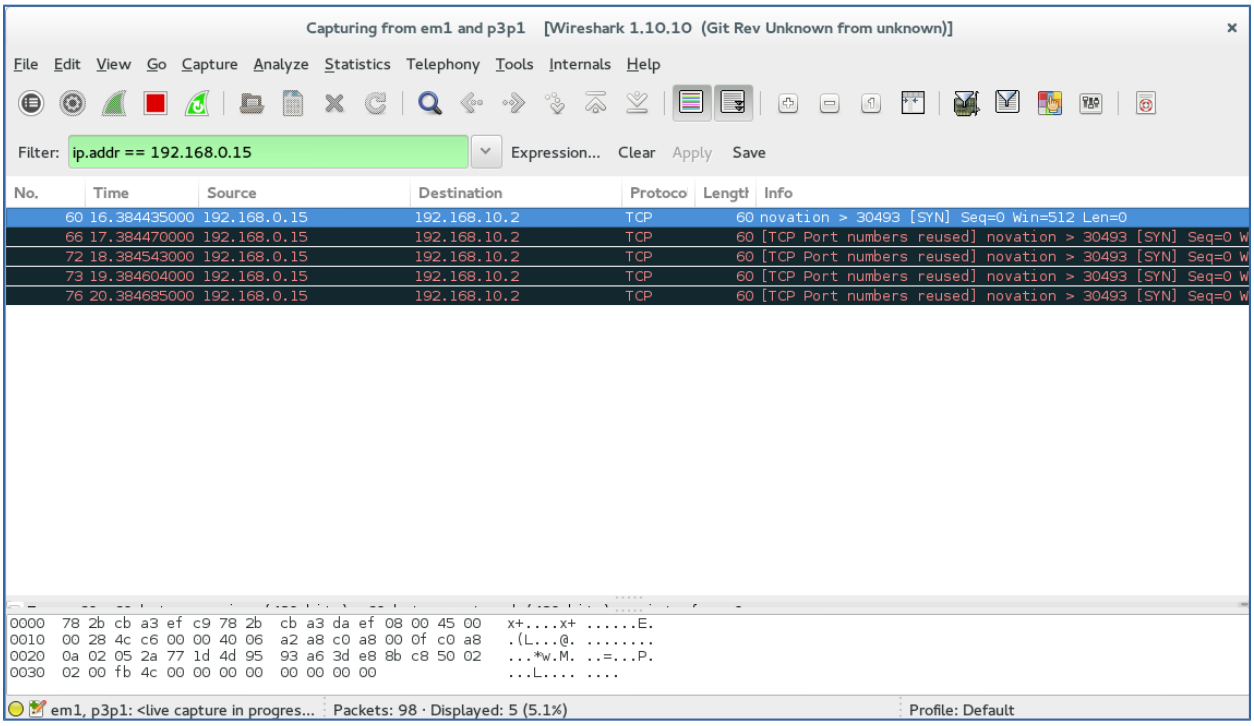
```
[root@DataComm ~]# hping3 192.168.10.2 -p 30493 -S -c 5 -k
HPING 192.168.10.2 (eml 192.168.10.2): S set, 40 headers + 0 data

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

In this test, we sent a SYN packet to a high port. As you can see, according to the drop rule the packet was thrown away. If we look to the forward chain in IPTABLES we can see that the packets were all dropped as well.

```
Chain FORWARD (policy DROP 5 packets, 200 bytes)
    pkts      bytes target     prot opt in     out     source
```

Finally, when we open wireshark on the firewall, we do see these packets making their way to 10.2 before they are dropped.



Test #7

In this test, I will show some packet transfers to blocked ports. First we sent 5 packets to port 120 on TCP.

```
[root@DataComm ~]# hping3 192.168.10.2 -p 120 -S -c 5 -k
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data by

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@DataComm ~]#
```

As you can see, they were dropped.

```
Chain FORWARD (policy DROP 5 packets, 200 bytes)
    pkts      bytes target     prot opt in     out      sou
       0        0 DROP       tcp  --  *      *        0.0.0
       0        0 DROP       tcp  --  *      *        0.0.0
```

If we look to the wireshark transfer we see 5 TCP packets being sent.
This shows that the TCP port blocking is effective.

No.	Time	Source	Destination	Protoco	Lengtl	Info
6	1.077312000	192.168.0.15	192.168.10.2	TCP	60	cadencecontrol > sf
9	2.077343000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers r
14	3.077444000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers r
17	4.077525000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers r
34	5.077542000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers r

Test #8

In this test I will send several SYN FIN packets to the internal host over port 80, an open port.

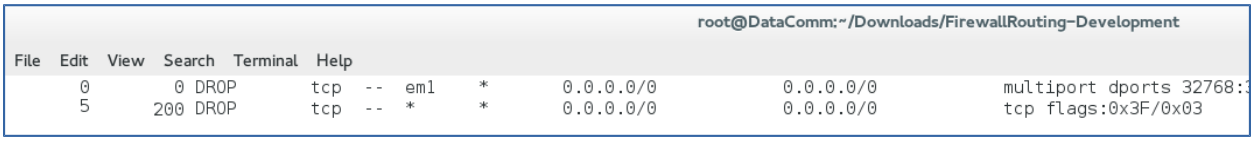
```
[root@DataComm ~]# hping3 192.168.10.2 -p 80 -S -F -c 5 -k
HPING 192.168.10.2 (em1 192.168.10.2): SF set, 40 headers + 0 data byt

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@DataComm ~]#
```

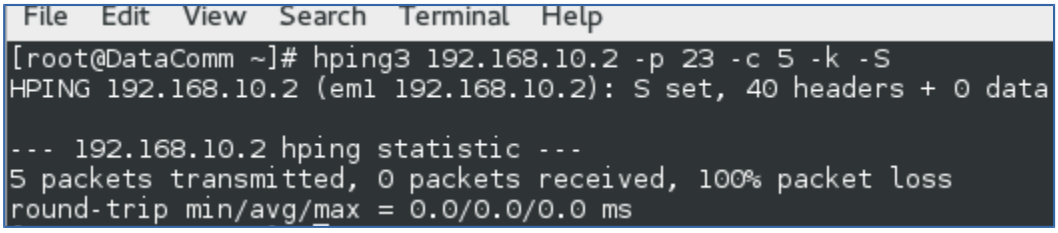
As you can see, none of these packets were able to return anything. If we look to the wireshark transfer we see them in progress.

No.	Time	Source	Destination	Protoco	Lengtl	Info
132	40.099037000	192.168.0.15	192.168.10.2	TCP	60	iec-104 > http [FIN, SYN] Seq=0 Win=512 Len=0
136	41.099106000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers reused] iec-104 > http [FIN, SY
142	42.099170000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers reused] iec-104 > http [FIN, SY
143	43.099228000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers reused] iec-104 > http [FIN, SY
145	44.099307000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers reused] iec-104 > http [FIN, SY

And finally, when we look to IPTABLES we can see that these 5 packets were dropped by the rule.

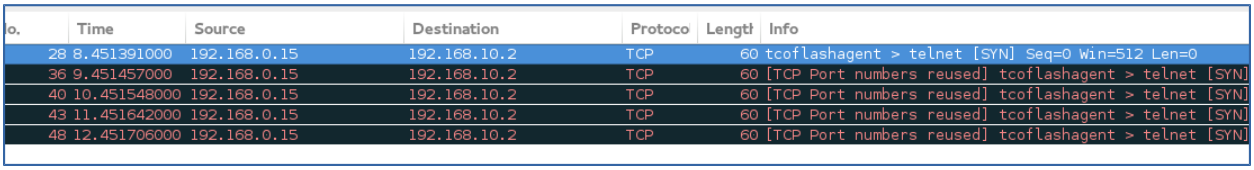


Test #9



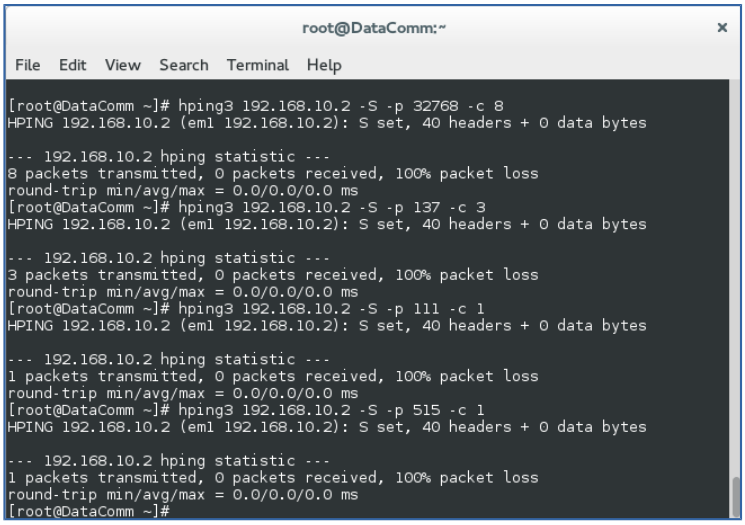
In this test, we sent 5 packets to the telnet port of the external host, to demonstrate that no Telnet traffic can get through. As you can see below, this traffic was dropped.

If we look to the transfer we see that 5 telnet packets were indeed sent. The test was a success.



Test #10

In this test, we checked the blocked sockets to see that they were actually blocking. As you can see, we sent packets to all 13 of these ports.



No.	Time	Source	Destination	Protoco	Lengtl	Info
398	124.12200100(	192.168.0.15	192.168.10.2	TCP	60	confluent > filenet-tms [SYN] Seq=0 Win=512 Len=0
401	125.12209000(	192.168.0.15	192.168.10.2	TCP	60	lansource > filenet-tms [SYN] Seq=0 Win=512 Len=0
402	126.12212000(	192.168.0.15	192.168.10.2	TCP	60	nms-topo-serv > filenet-tms [SYN] Seq=0 Win=512 Le
420	127.12221000(	192.168.0.15	192.168.10.2	TCP	60	localinfosrvr > filenet-tms [SYN] Seq=0 Win=512 Le
421	128.12227800(	192.168.0.15	192.168.10.2	TCP	60	docstor > filenet-tms [SYN] Seq=0 Win=512 Len=0
424	129.12233400(	192.168.0.15	192.168.10.2	TCP	60	dmdocbroker > filenet-tms [SYN] Seq=0 Win=512 Len=
455	130.12232900(	192.168.0.15	192.168.10.2	TCP	60	insitu-conf > filenet-tms [SYN] Seq=0 Win=512 Len=
473	131.12242800(	192.168.0.15	192.168.10.2	TCP	60	1491 > filenet-tms [SYN] Seq=0 Win=512 Len=0
571	146.28497800(	192.168.0.15	192.168.10.2	TCP	60	device2 > netbios-ns [SYN] Seq=0 Win=512 Len=0
576	147.28507900(	192.168.0.15	192.168.10.2	TCP	60	mobrien-chat > netbios-ns [SYN] Seq=0 Win=512 Len=
577	148.28514700(	192.168.0.15	192.168.10.2	TCP	60	blackboard > netbios-ns [SYN] Seq=0 Win=512 Len=0
616	159.09000500(	192.168.0.15	192.168.10.2	TCP	60	iapp > sunrpc [SYN] Seq=0 Win=512 Len=0
660	172.04202800(	192.168.0.15	192.168.10.2	TCP	60	radio-sm > printer [SYN] Seq=0 Win=512 Len=0

If we look to wireshark, we see these 13 packets being sent out.

And if we look to the IPTABLES entry, we can see that these packets were caught. This test was a success.

```
Chain FORWARD (policy DROP 13 packets, 520 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:23
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:23
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ES
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,ES
```

Test #11

In the below terminals, we sent packets out to both port 20 and 22, ports which will be edited for Maximum throughput and Minimized delay.

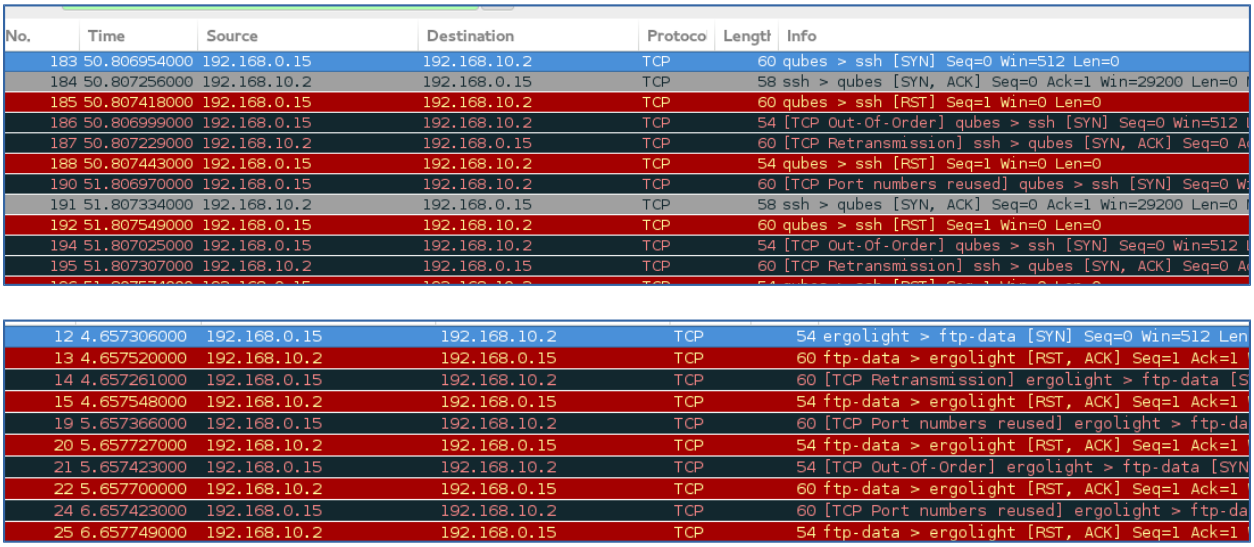
```
--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.1/4001.9 ms
[root@DataComm FirewallRouting]# hping3 192.168.10.2 -p 20 -S -c 5 -k
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=8501 sport=20 flags=RA seq=0 win=0 rtt=1.1 r
s
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=8511 sport=20 flags=RA seq=0 win=0 rtt=
1000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=9272 sport=20 flags=RA seq=0 win=0 rtt=
2000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=9526 sport=20 flags=RA seq=0 win=0 rtt=
3001.9 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=9755 sport=20 flags=RA seq=0 win=0 rtt=
4002.0 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
```

```
[root@DataComm FirewallRouting]# hping3 192.168.10.2 -p 22 -S -c 5 -k
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rtt=1.1
ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
=1001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
=2000.8 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
=3000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
=4001.9 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
```

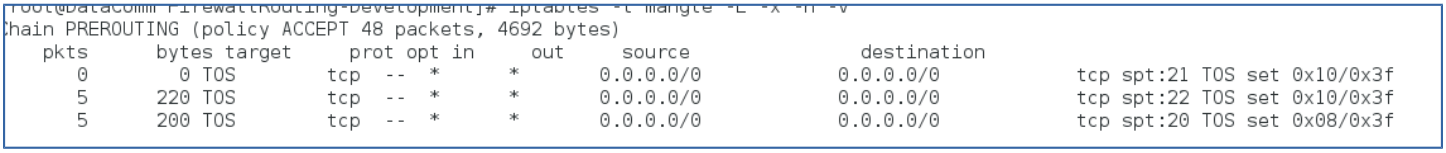
And if we look to wireshark, we can see these packets being sent.



No.	Time	Source	Destination	Protoco	Lengtl	Info
183	50.806954000	192.168.0.15	192.168.10.2	TCP	60	qubes > ssh [SYN] Seq=0 Win=512 Len=0
184	50.807256000	192.168.10.2	192.168.0.15	TCP	58	ssh > qubes [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
185	50.807418000	192.168.0.15	192.168.10.2	TCP	60	qubes > ssh [RST] Seq=1 Win=0 Len=0
186	50.806999000	192.168.0.15	192.168.10.2	TCP	54	[TCP Out-Of-Order] qubes > ssh [SYN] Seq=0 Win=512
187	50.807229000	192.168.10.2	192.168.0.15	TCP	60	[TCP Retransmission] ssh > qubes [SYN, ACK] Seq=0 A
188	50.807443000	192.168.0.15	192.168.10.2	TCP	54	qubes > ssh [RST] Seq=1 Win=0 Len=0
190	51.806970000	192.168.0.15	192.168.10.2	TCP	60	[TCP Port numbers reused] qubes > ssh [SYN] Seq=0 W
191	51.807334000	192.168.10.2	192.168.0.15	TCP	58	ssh > qubes [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
192	51.807549000	192.168.0.15	192.168.10.2	TCP	60	qubes > ssh [RST] Seq=1 Win=0 Len=0
194	51.807025000	192.168.0.15	192.168.10.2	TCP	54	[TCP Out-Of-Order] qubes > ssh [SYN] Seq=0 Win=512
195	51.807307000	192.168.10.2	192.168.0.15	TCP	60	[TCP Retransmission] ssh > qubes [SYN, ACK] Seq=0 A

12	4.657306000	192.168.0.15	192.168.10.2	TCP	54 ergolight > ftp-data [SYN] Seq=0 Win=512 Len
13	4.657520000	192.168.10.2	192.168.0.15	TCP	60 ftp-data > ergolight [RST, ACK] Seq=1 Ack=1
14	4.657261000	192.168.0.15	192.168.10.2	TCP	60 [TCP Retransmission] ergolight > ftp-data [S
15	4.657548000	192.168.10.2	192.168.0.15	TCP	54 ftp-data > ergolight [RST, ACK] Seq=1 Ack=1
19	5.657366000	192.168.0.15	192.168.10.2	TCP	60 [TCP Port numbers reused] ergolight > ftp-da
20	5.657727000	192.168.10.2	192.168.0.15	TCP	54 ftp-data > ergolight [RST, ACK] Seq=1 Ack=1
21	5.657423000	192.168.0.15	192.168.10.2	TCP	54 [TCP Out-Of-Order] ergolight > ftp-data [SYN
22	5.657700000	192.168.10.2	192.168.0.15	TCP	60 ftp-data > ergolight [RST, ACK] Seq=1 Ack=1
24	6.657423000	192.168.0.15	192.168.10.2	TCP	60 [TCP Port numbers reused] ergolight > ftp-da
25	6.657749000	192.168.10.2	192.168.0.15	TCP	54 ftp-data > ergolight [RST, ACK] Seq=1 Ack=1

Finally, if we look to IPTABLES we see that mangle modified these packets. Proving this to be a success.



```
root@datacomm FirewallRouting-Development]# iptables -t mangle -L -x -n -v
Chain PREROUTING (policy ACCEPT 48 packets, 4692 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 TOS        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:21 TOS set 0x10/0x3f
       5      220 TOS        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 TOS set 0x10/0x3f
       5      200 TOS        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:20 TOS set 0x08/0x3f
```