COMP 8006 – Assignment 2

Kyle Gilles & Justin Tom

A00847898 - A00852990

Table of Contents

Assignment Guidelines

Objective

- To design, implement and test a standalone Linux firewall and packet filter.

Mission

- Design, implement and test a firewall for Linux that will implement the following rules:
- Set the initial default policies.
- Get user specified parameters (see constraints) and create a set of rules that will implement the firewall requirements. Specifically the firewall will control:
    - Inbound/Outbound TCP packets on allowed ports.
    - Inbound/Outbound UDP packets on allowed ports.
    - Inbound/Outbound ICMP packets based on type numbers.
    - All packets that fall through to the default rule will be dropped.
    - Drop all packets destined for the firewall host from the outside.
    - Do not accept any packets with a source address from the outside matching your internal network.
    - You must ensure the you reject those connections that are coming the “wrong” way (i.e., inbound SYN packets to high ports).
    - Accept fragments.
    - Accept all TCP packets that belong to an existing connection (on allowed ports).
    - Drop all TCP packets with the SYN and FIN bit set.
    - Do not allow Telnet packets at all.
    - Block all external traffic directed to ports 32768 – 32775, 137 – 139, TCP ports 111 and 515.
    - For FTP and SSH services, set control connections to "Minimum Delay" and FTP data to "Maximum Throughput".
- Design a test procedure that will test all your firewall rules and print the results of the test to a file. Make sure that someone reading the file contents will know exactly which rule worked and which rule failed.
- The machines in the lab are equipped with two Ethernet cards. One of them is already configured and operational. You will have to enable and configure the other one for use as the gateway to your “internal” network.
- Your testbed will then have one machine operating as a firewall. It will have an “outside” connection (eth0) and it will forward datagrams to hosts on its internal hosts on the second NIC (eth1).

Constraints

- The firewall/packet filter must be designed and implemented using Netfilter.
- Your firewall script must have two sections: a "User Configurable Section" and the "Implementation Section".
- The user configuration section will allow a user to set at least the following parameters:

    - Name and location of the utility you are using to implement the firewall.
    - Internal network address space and the network device.
    - Outside address space and the network device.
    - TCP services that will be allowed.
    - UDP services that will be allowed.
    - ICMP services that will be allowed.
- Only allow NEW and ESTABLISHED traffic to go through the firewall. In other words you are doing stateful filtering.
- You must ensure that you reject those connections that are coming the "wrong" way, meaning inbound connection requests (unless of course it is to a permitted service).
- Design test scripts to validate your firewall rules.
- You will be required to demonstrate your functional firewall in the lab on the day the assignment is due.

Program Summary

- The bash script, when executed, will configure the host's firewall settings through the use of the IPtables commands. The user will be able to specify the internal network address and NIC, the external network address and NIC as well as the TCP, UDP and ICMP services that will be allowed. It is a stateful filtering firewall, allowing NEW and ESTABLISHED traffic to go through the firewall. The commands will set all the default policies to DROP, allow inbound/outbound TCP, UDP and ICMP packets through the ports specified by the user, drops all packets destined for the firewall host from the outside, drops any packets with a source address from the outside matching your internal network, drops all connections coming from high ports (higher than 1023), accepts packet fragments, accepts TCP packets that belong to an existing connection (on allowed ports), drops all TCP packets with both SYN and FIN bit set, drops all Telnet packets, drops all external traffic directed to ports 32768-32775, 137-139, TCP ports 111 and 515 and sets FTP and SSH services to “Minimum Delay” as well as FTP data to “Maximum throughput”

Program File Structure

Firewall.sh

- This is the main file that will be executed. It contains all the IPtable commands and configurations that will design the firewall to the above specifications and restrictions. In the script, it will prompt the user for a few configuration variables such as network card names and what ports they would like open for TCP, UDP and ICMP services. There are also a few ports that are pre-defined and “default” allow. Ports 20, 21, 22, for TCP for FTP data, FTP control, SSH ports respectively. And ports 0, 3 and 8 for ICMP for regular echo-reply, port unreachable reply and echo-request ports respectively. The script sets all the default policies to DROP and runs all the DROP conditions at the top of the script since the file is read sequentially, it would be best to drop the packets that meet the specified requirements as soon as possible. Afterwards, the rest of the firewall constraints and rules are applied. For the user configuration TCP, UDP and ICMP rules, the script uses an array and a for loop to iterate through all the specified ports configured by the user and applies them to the firewall as rules. At the end of the script, the iptables is saved, restarted and then finally listed for the user to see the end result of all the rules.

FirewallConfig.sh

- This file is required to be run before the 'Firewall.sh' is ran. It is to be ran on the host you wish to act as the firewall. It prompts the user for a few configuration variables such as internal firewall and external host IP addresses and the two network card interface names. It uses the user inputted information to essentially set up and run any pre-required configuration to the routing and IPtables for forward traffic to its newly created subnet network and vice versa.

InternalHostConfig.sh

- This file is also required to be run before the 'Firewall.sh' is ran. It is to be ran on the host you wish to act as the internal host in the subnet. It prompts the user for a few configurations such as internal host and firewall IP addresses as well as the two network interface cards to help set-up and configure the internal host. It essentially disables the network interface card connected to the Internet and changes the second network interface card to connect to the firewall and adds the default gateway to point to the firewall.

TestScript.sh

- This file is used to test the rules of the firewall based on the previously outlined constraints and firewall requirements. It also requests for a few configuration parameters of external firewall IP address, the internal (subnet) address of the internal host and the subnet of the network the firewall had created in order to add the proper routing rule so the host knows what gateway to connect to when looking for the specified subnet address of your internal host. It will log all the output (while keeping the standard output to the terminal) and save to a text file under the name of 'Firewall_Test_log_' then the current time (Hour.Minutes-Month-Day-Year) in the same directory.

Reset.sh

- This file will flush any existing rules and clear all user-defined chains from the IPtables, NAT tables and mangle tables as well as setting the default policies for Input, Output and Forward as accept.

Test Case Screen Shots

- Directory that contains all the screenshots that were used to as figures in the test cases for each machine (firewall, internal, external)

Test Script Packet Captures

- Directory that contains three wireshark capture files that were recorded when running the test script file - each one for the different machine (Firewall, Internal and external)

Program Instructions

- Navigate through the directories to the folder with all the files in it.
- May have to change permissions on the files
  - `chmod 777 *`
- If the user wants the firewall to allow the internal host to be able to access websites, the `/etc/resolv.conf` files on both firewall and internal hosts must have matching nameservers.
  - During the firewall setup, you must also allow DNS, DHCP, HTTP and HTTPS
    - UDP 53, 67, 68
    - TCP 53, 80, 443
  - `nameserver 8.8.8.8` required to be added in both resolv.conf files.

- On the Internal Host execute:

```
./InternalHostConfig.sh
```

- On the standalone firewall execute:

```
./Reset.sh
./FirewallConfig.sh
./Firewall.sh
```

- On all three machines involved, enter the command: Route -n
- Ensure the results resemble the following: (If not, User Route add /del and match to below).

Internal host:

Destination	**Gateway**	**Genmask**	**Interface**
default	192.168.10.1	0.0.0.0	p3p1
192.168.10.0	0.0.0.0	255.255.255.0	p3p1

Firewall:

Destination	**Gateway**	**Genmask**	**Interface**
0.0.0.0	192.168.0.100	0.0.0.0	em1
192.168.0.0	192.168.0.5	255.255.255.0	em1
192.168.10.0	192.168.10.0	255.255.255.0	p3p1

External host:

Destination	**Gateway**	**Genmask**	**Interface**
0.0.0.0	192.168.0.100	0.0.0.0	em1
192.168.0.0	0.0.0.0	255.255.255.0	em1
192.168.10.0	192.168.0.5	255.255.255.0	em1

- Finally, onn the external host execute:

```
./TestScript.sh
```

Assumptions

- No errors are made when inputting values in prompt
    - No error checking
- User knows the pre-existing/default port rules and doesn't add duplicates for TCP (20, 21, 22) and ICMP (0, 3, 8) port rules.
    - No handling for duplicates.

Test Cases

#	Test Description	Tool Used	Expected Result	Pass/Fail
1	Drop all packets destined for the firewall from the outside.	hping3 Wireshark	The firewall should DROP the packet.	Pass
2	Allow Inbound / Outbound TCP Packets on allowed Ports.	hping3 Wireshark	The firewall should ACCEPT the packets.	Pass.
3	Allow Inbound / Outbound UDP Packets on allowed	hping3 Wireshark	The firewall should ACCEPT	Pass

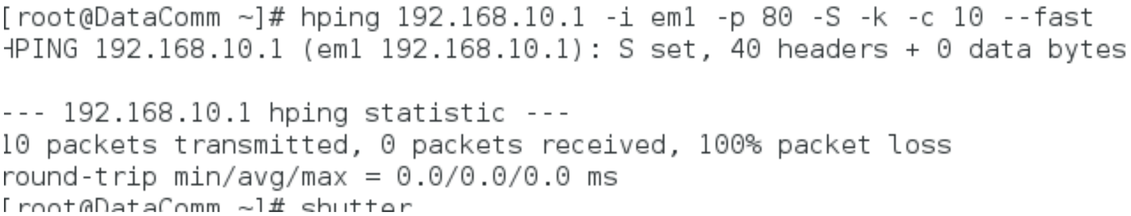
	ports		the packets.	
4	Allow Inbound / Outbound ICMP Packets on allowed ports.	hping3 Wireshark	The firewall should ACCEPT the packets.	Pass
5	Do not accept any packets with a source address from the outside matching your internal network	hping3 Wireshark	The firewall should DROP the packets.	Pass
6	Reject those connections that are coming the 'wrong' way (incoming SYN to high ports)	hping3 Wireshark	The firewall should DROP the packets.	Pass
7	Accept fragmented packets	hping3 Wireshark	The firewall should ACCEPT the packets.	Pass
8	Drop all TCP packets with the SYN and FIN bit set	hping3 Wireshark	The firewall should DROP the packets.	Pass
9	Do not allow Telnet packets at all	telnet Wireshark	The firewall should DROP the packets.	Pass
10	Block all external traffic directed to ports 32768-32775	hping3 Wireshark	The firewall should DROP the packets.	Pass
11	Block all external traffic directed to ports 137-139	hping3 Wireshark	The firewall should DROP the packets	Pass
12	Block all external traffic directed to TCP on port 111	hping3 Wireshark	The firewall should DROP the packets	Pass
13	Block all external traffic directed to TCP on port 515	hping3 Wireshark	The firewall should DROP the packets	Pass
14	Ensure FTP and SSH services have “Minimum Delay” (Requires a new DSCP column in wireshark. Value = ‘ip.dsfield.dscp’)	hping3 Wireshark	The firewall should ACCEPT the packets with the DSCP value of ‘4’ in Wireshark. Source: https://www.goo	Pass

			gle.ca/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0CAQQjBw&url=https%3A%2F%2Fblogs.manageengine.com%2Fimage%2F501000001070197%2FQoS4.png&ei=SGnZVL-CMYbwoAS744DoBw&bvm=bv.85464276,d.cGU&psig=AFQjCNH9bzW-pFb10jyUrldCl3yGy5nopg&ust=1423620747126691	
15	Ensure FTP Data service has “Maximum Throughput” (Requires a new DSCP column in wireshark. Value = ‘ip.dsfield.dscp’)	hping3 Wireshark	The firewall should ACCEPT the packets and show DSCP values of ‘2’ in Wireshark. Source: https://www.google.ca/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0CAQQjBw&url=https%3A%2F%2Fblogs.manageengine.com%2Fimage%2F501000001070197%2FQoS4.png&ei=SGnZVL-CMYbwoAS744DoBw&bvm=bv.85464276,d.cGU&psig=AFQjCNH9bzW-	Pass

			pFb10jyUrldCl3yGy5nopg&ust=1423620747126691	

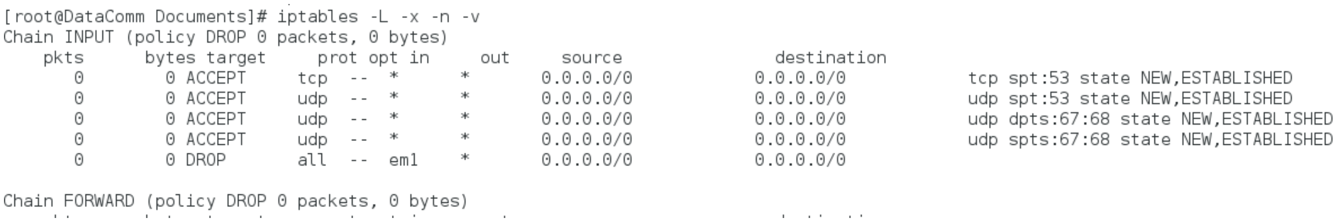
Test Case 1 Drop all packets destined for the firewall from the outside.

We entered the following command.



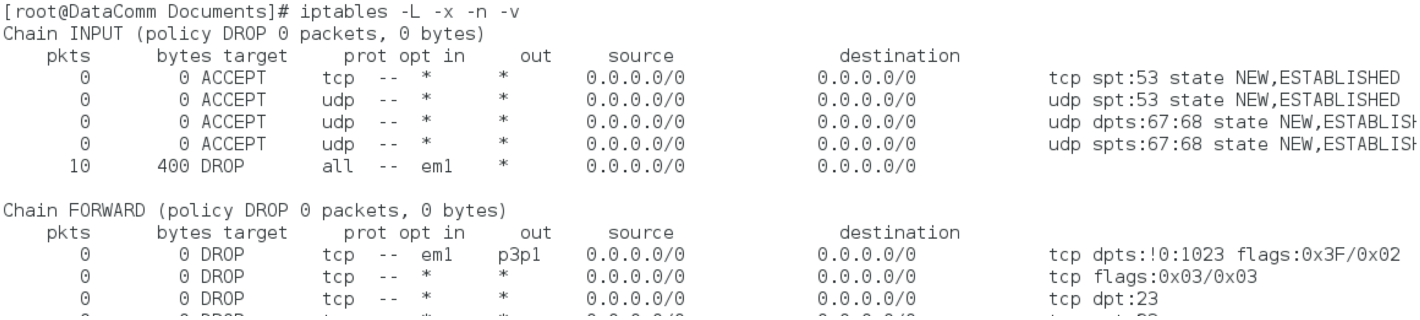
```
[root@DataComm ~]# hping 192.168.10.1 -i em1 -p 80 -S -k -c 10 --fast
HPING 192.168.10.1 (em1 192.168.10.1): S set, 40 headers + 0 data bytes

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

The IP Tables before we entered the command



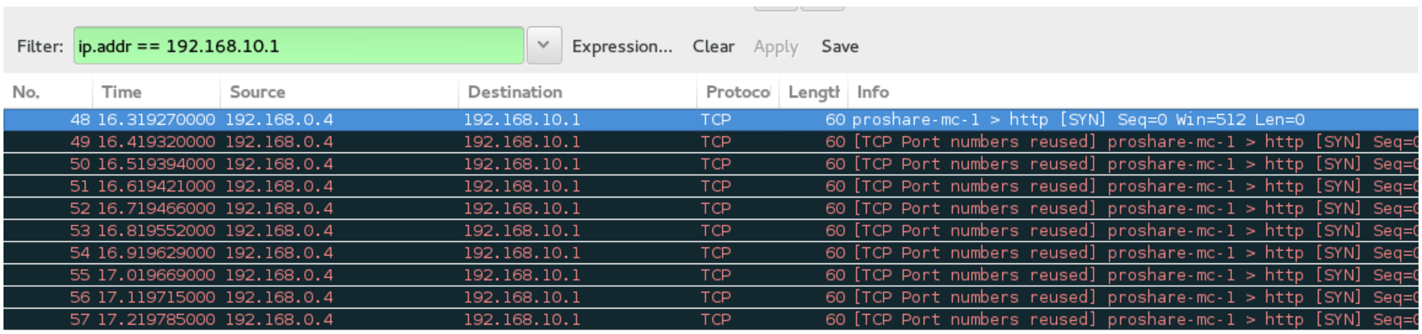
```
[root@DataComm Documents]# iptables -L -x -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:53 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:53 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpts:67:68 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spts:67:68 state NEW,ESTABLISHED
       0        0 DROP       all  --  em1    *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
```

IP tables after we entered the command. We can see the Firewall dropping the 10 packets at the rule.



```
[root@DataComm Documents]# iptables -L -x -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:53 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:53 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpts:67:68 state NEW,ESTABLISH
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spts:67:68 state NEW,ESTABLISH
      10      400 DROP       all  --  em1    *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:0x3F/0x02
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23
```

Wireshark Capture of the Firewall dropping the packets.

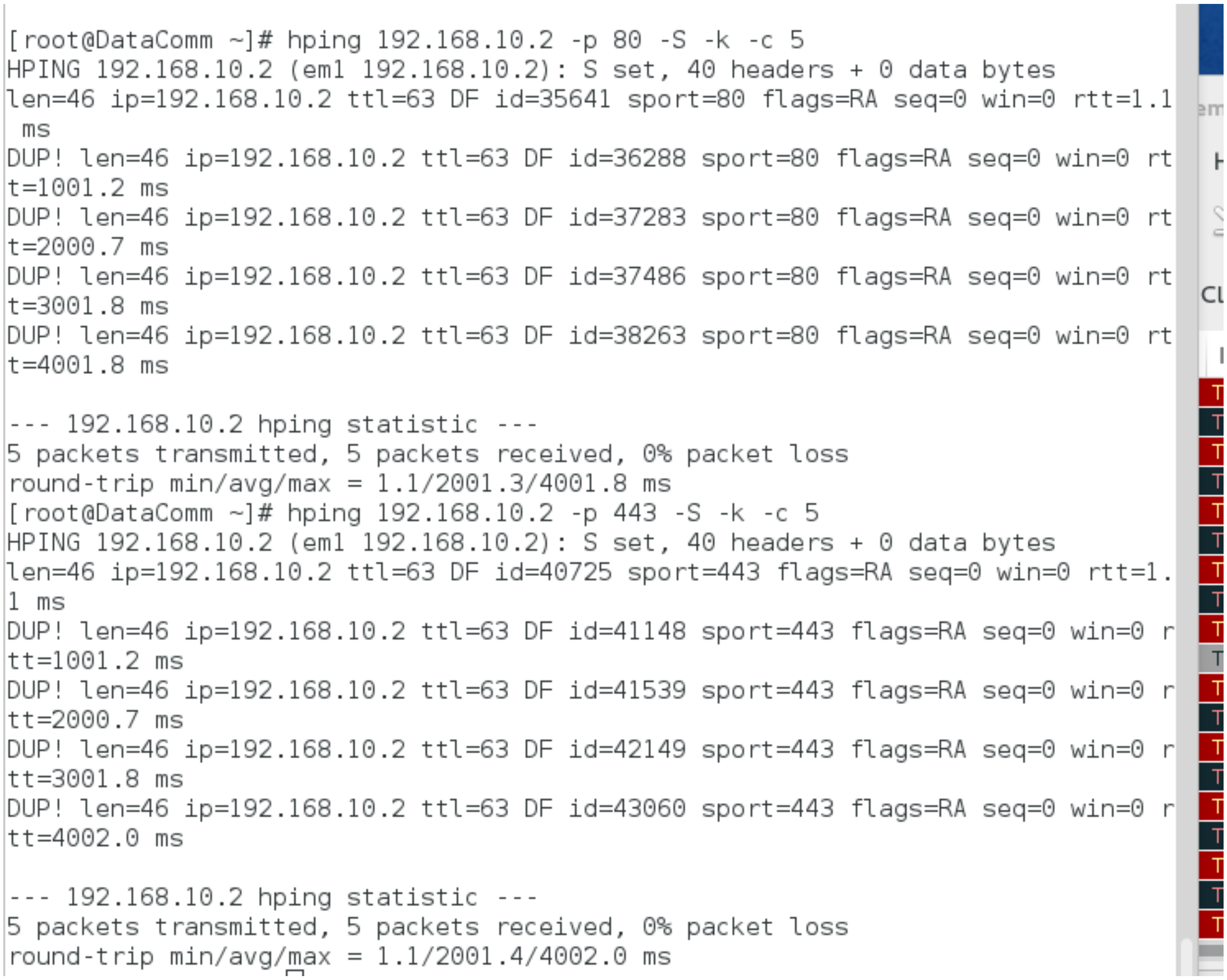


Filter: ip.addr == 192.168.10.1

No.	Time	Source	Destination	Protoco	Length	Info
48	16.319270000	192.168.0.4	192.168.10.1	TCP	60	proshare-mc-1 > http [SYN] Seq=0 Win=512 Len=0
49	16.419320000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
50	16.519394000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
51	16.619421000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
52	16.719466000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
53	16.819552000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
54	16.919629000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
55	17.019669000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
56	17.119715000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=
57	17.219785000	192.168.0.4	192.168.10.1	TCP	60	[TCP Port numbers reused] proshare-mc-1 > http [SYN] Seq=

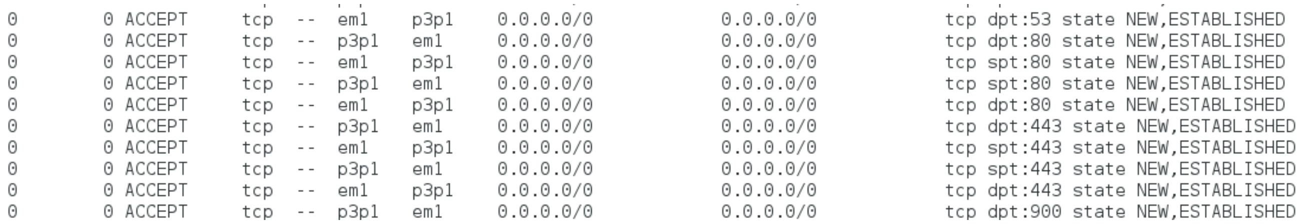
Test Case 2: Allow inbound / outbound TCP packets on allowed ports.

Upon initiating the firewall the user is prompted to allow traffic on specified ports. We opened port 80 and 443 for this test and entered the following commands.



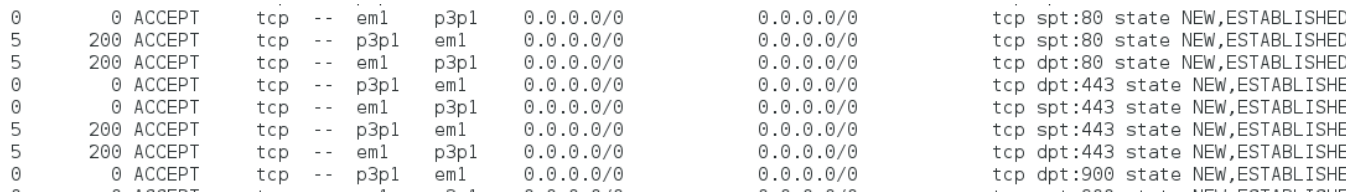
```
[root@DataComm ~]# hping 192.168.10.2 -p 80 -S -k -c 5
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=35641 sport=80 flags=RA seq=0 win=0 rtt=1.1
 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=36288 sport=80 flags=RA seq=0 win=0 rt
t=1001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=37283 sport=80 flags=RA seq=0 win=0 rt
t=2000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=37486 sport=80 flags=RA seq=0 win=0 rt
t=3001.8 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=38263 sport=80 flags=RA seq=0 win=0 rt
t=4001.8 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.3/4001.8 ms
[root@DataComm ~]# hping 192.168.10.2 -p 443 -S -k -c 5
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=40725 sport=443 flags=RA seq=0 win=0 rtt=1.
1 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=41148 sport=443 flags=RA seq=0 win=0 r
tt=1001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=41539 sport=443 flags=RA seq=0 win=0 r
tt=2000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=42149 sport=443 flags=RA seq=0 win=0 r
tt=3001.8 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=43060 sport=443 flags=RA seq=0 win=0 r
tt=4002.0 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.4/4002.0 ms
```

IPtables before the hping.



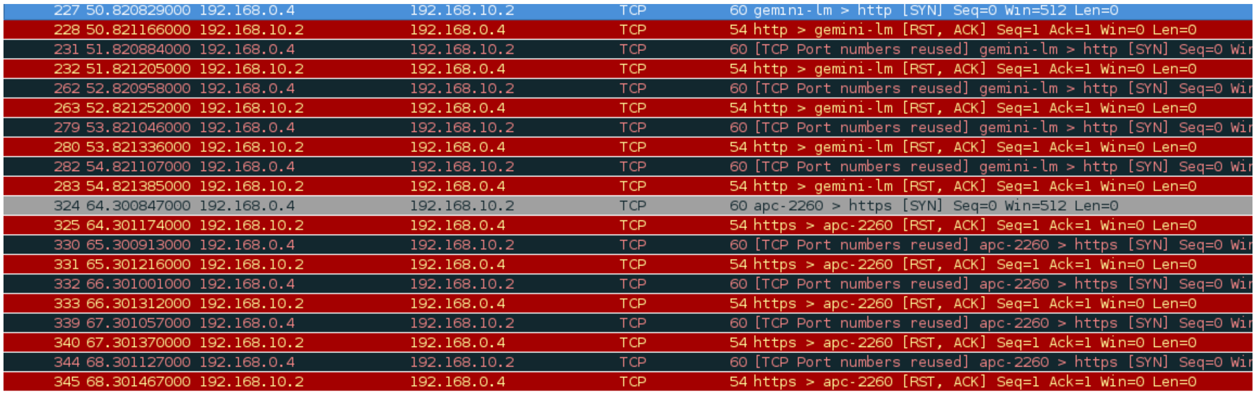
```
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:53 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:80 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:80 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:443 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:443 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:443 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:443 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:900 state NEW,ESTABLISHED
```

IPtables after the hping. We see the 5 packets travelling both to and from the host, on both port 80 and 443.



```
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,ESTABLISHED
    5      200 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,ESTABLISHED
    5      200 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:80 state NEW,ESTABLISHED
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:443 state NEW,ESTABLISHE
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:443 state NEW,ESTABLISHE
    5      200 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:443 state NEW,ESTABLISHE
    5      200 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:443 state NEW,ESTABLISHE
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:900 state NEW,ESTABLISHE
```

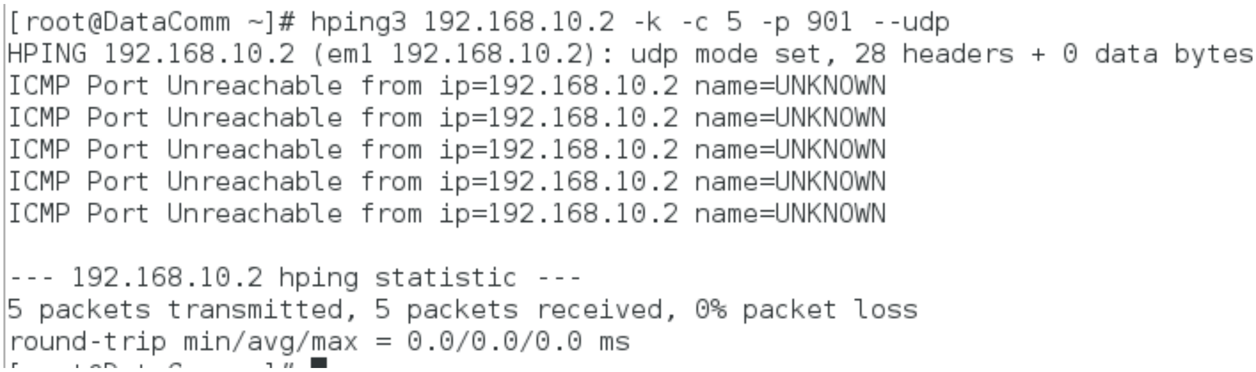
Wireshark capturing the accepted packets.



```
227 50.820829000 192.168.0.4     192.168.10.2   TCP   60 gemini-lm > http [SYN] Seq=0 Win=512 Len=0
228 50.821166000 192.168.10.2    192.168.0.4    TCP   54 http > gemini-lm [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
231 51.820884000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] gemini-lm > http [SYN] Seq=0 Wi
232 51.821205000 192.168.10.2    192.168.0.4    TCP   54 http > gemini-lm [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
262 52.820958000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] gemini-lm > http [SYN] Seq=0 Wi
263 52.821252000 192.168.10.2    192.168.0.4    TCP   54 http > gemini-lm [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
279 53.821046000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] gemini-lm > http [SYN] Seq=0 Wi
280 53.821336000 192.168.10.2    192.168.0.4    TCP   54 http > gemini-lm [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
282 54.821107000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] gemini-lm > http [SYN] Seq=0 Wi
283 54.821385000 192.168.10.2    192.168.0.4    TCP   54 http > gemini-lm [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
324 64.300847000 192.168.0.4     192.168.10.2   TCP   60 apc-2260 > https [SYN] Seq=0 Win=512 Len=0
325 64.301174000 192.168.10.2    192.168.0.4    TCP   54 https > apc-2260 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
330 65.300913000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] apc-2260 > https [SYN] Seq=0 Wi
331 65.301216000 192.168.10.2    192.168.0.4    TCP   54 https > apc-2260 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
332 66.301001000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] apc-2260 > https [SYN] Seq=0 Wi
333 66.301312000 192.168.10.2    192.168.0.4    TCP   54 https > apc-2260 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
339 67.301057000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] apc-2260 > https [SYN] Seq=0 Wi
340 67.301370000 192.168.10.2    192.168.0.4    TCP   54 https > apc-2260 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
344 68.301127000 192.168.0.4     192.168.10.2   TCP   60 [TCP Port numbers reused] apc-2260 > https [SYN] Seq=0 Wi
345 68.301467000 192.168.10.2    192.168.0.4    TCP   54 https > apc-2260 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

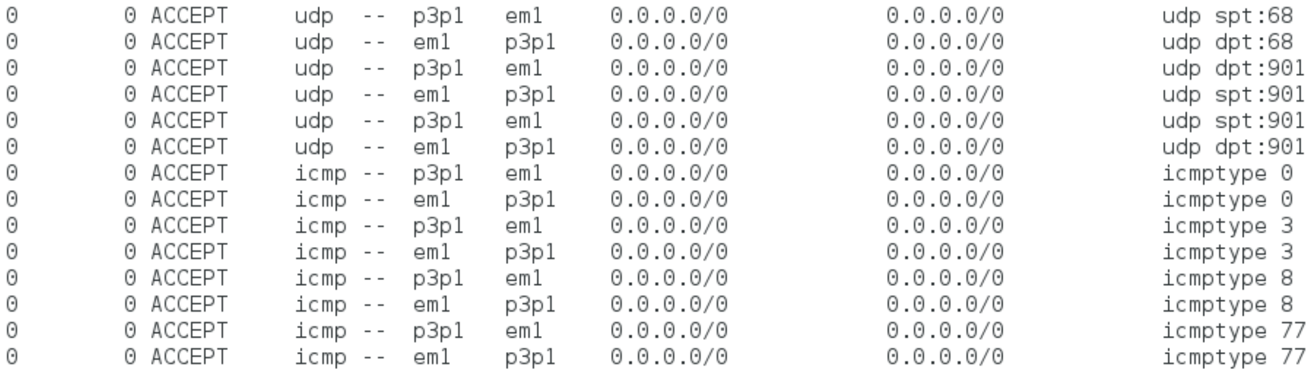
Test Case 3: Allow Inbound / Outbound UDP Packets on allowed ports

To test, we opened port 901 for UDP outgoing packets. We saw in Wireshark that “Port Unreachable” packets were being created on the internal host and sent back as ACKS, so we made a rule to accept ICMP type 3 packets to accept these.



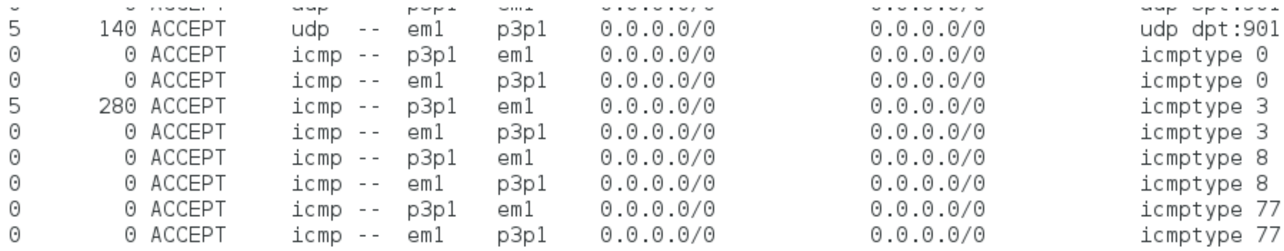
```
[root@DataComm ~]# hping3 192.168.10.2 -k -c 5 -p 901 --udp
HPING 192.168.10.2 (em1 192.168.10.2): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
ICMP Port Unreachable from ip=192.168.10.2 name=UNKNOWN
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
```

IPtables before the command.



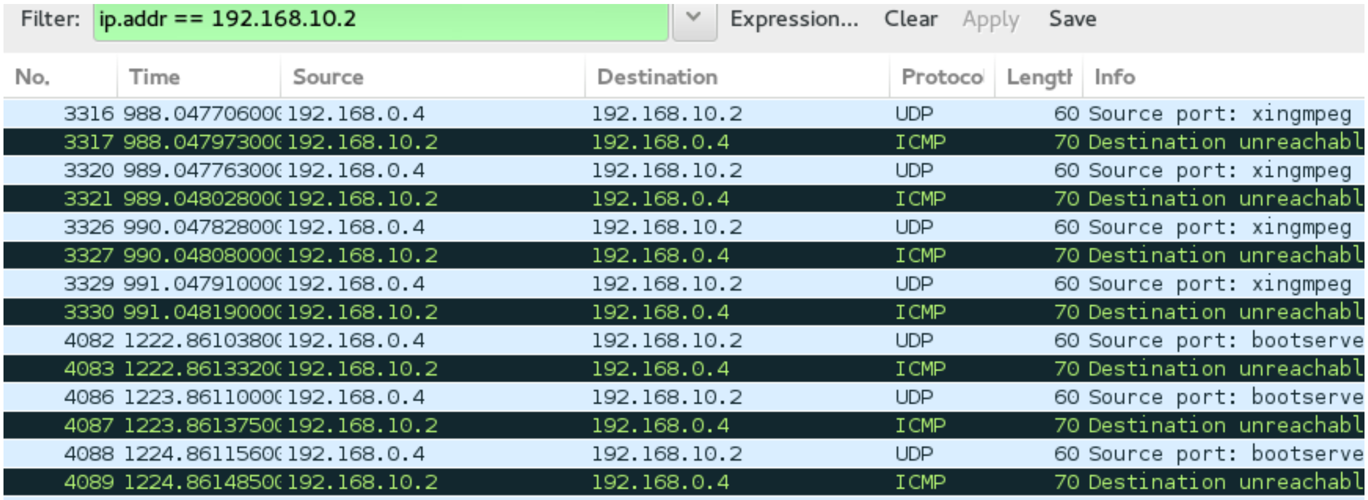
```
    0     0 ACCEPT     udp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            udp spt:68
    0     0 ACCEPT     udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpt:68
    0     0 ACCEPT     udp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            udp dpt:901
    0     0 ACCEPT     udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp spt:901
    0     0 ACCEPT     udp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            udp spt:901
    0     0 ACCEPT     udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpt:901
    0     0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 0
    0     0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 0
    0     0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 3
    0     0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 3
    0     0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 8
    0     0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 8
    0     0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 77
    0     0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 77
```

IPtables after the command. We see the accepted UDP packets and ICMP packets (our UDP ACKS).



```
5      140 ACCEPT     udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpt:901
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 0
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 0
5      280 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 3
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 3
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 8
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 8
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            icmptype 77
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            icmptype 77
```

Wireshark capture at the firewall of above. .

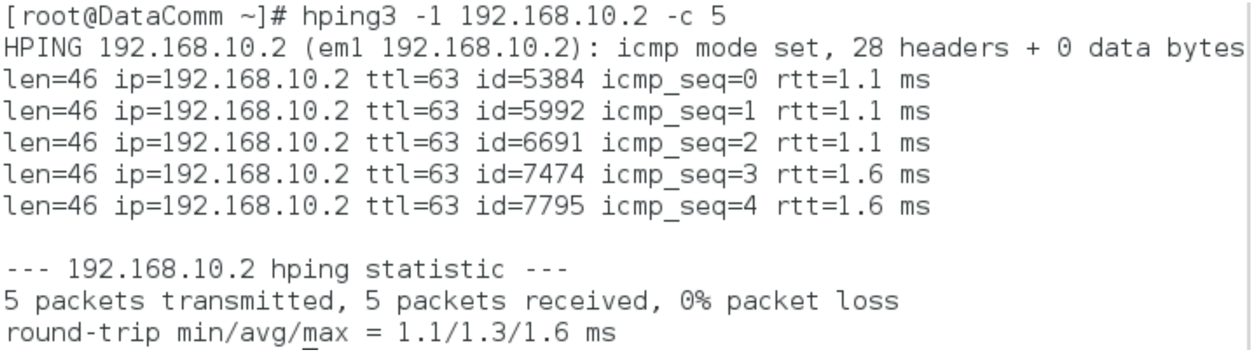


Filter: ip.addr == 192.168.10.2

No.	Time	Source	Destination	Protoco	Lengtl	Info
3316	988.04770600(	192.168.0.4	192.168.10.2	UDP	60	Source port: xingmpeg
3317	988.04797300(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
3320	989.04776300(	192.168.0.4	192.168.10.2	UDP	60	Source port: xingmpeg
3321	989.04802800(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
3326	990.04782800(	192.168.0.4	192.168.10.2	UDP	60	Source port: xingmpeg
3327	990.04808000(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
3329	991.04791000(	192.168.0.4	192.168.10.2	UDP	60	Source port: xingmpeg
3330	991.04819000(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
4082	1222.8610380(	192.168.0.4	192.168.10.2	UDP	60	Source port: bootserve
4083	1222.8613320(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
4086	1223.8611000(	192.168.0.4	192.168.10.2	UDP	60	Source port: bootserve
4087	1223.8613750(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl
4088	1224.8611560(	192.168.0.4	192.168.10.2	UDP	60	Source port: bootserve
4089	1224.8614850(	192.168.10.2	192.168.0.4	ICMP	70	Destination unreachabl

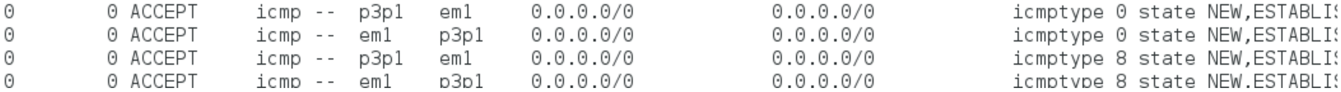
Test Case 4: Allow Inbound / Outbound ICMP Packets on allowed ports.

We allowed ICMP Type 0 and 8 for the test and entered the following command from the external host.



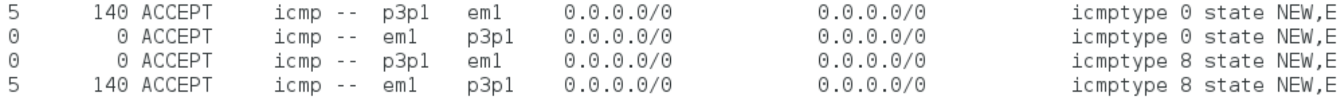
```
[root@DataComm ~]# hping3 -1 192.168.10.2 -c 5
HPING 192.168.10.2 (em1 192.168.10.2): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 id=5384 icmp_seq=0 rtt=1.1 ms
len=46 ip=192.168.10.2 ttl=63 id=5992 icmp_seq=1 rtt=1.1 ms
len=46 ip=192.168.10.2 ttl=63 id=6691 icmp_seq=2 rtt=1.1 ms
len=46 ip=192.168.10.2 ttl=63 id=7474 icmp_seq=3 rtt=1.6 ms
len=46 ip=192.168.10.2 ttl=63 id=7795 icmp_seq=4 rtt=1.6 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.3/1.6 ms
```

IPtables before entering the command.



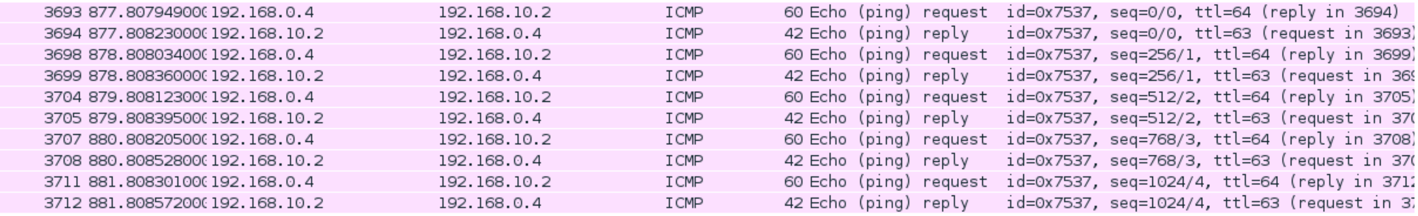
```
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0              0.0.0.0/0              icmptype 0 state NEW,ESTABLI
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0              0.0.0.0/0              icmptype 0 state NEW,ESTABLI
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0              0.0.0.0/0              icmptype 8 state NEW,ESTABLI
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0              0.0.0.0/0              icmptype 8 state NEW,ESTABLI
```

IPtables after entering the command. We can see the accepted ICMP type 0 and type 8 packets.



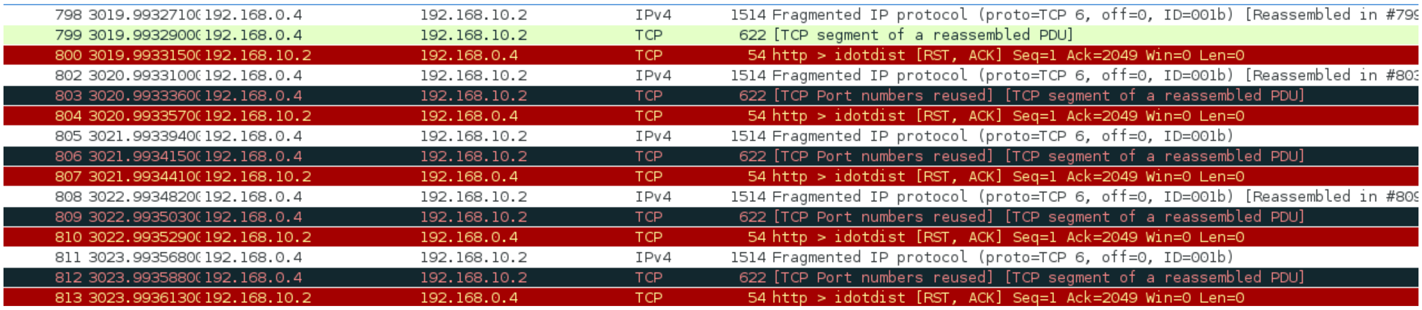
```
5      140 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0              0.0.0.0/0              icmptype 0 state NEW,E
0        0 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0              0.0.0.0/0              icmptype 0 state NEW,E
0        0 ACCEPT     icmp --  p3p1   em1     0.0.0.0/0              0.0.0.0/0              icmptype 8 state NEW,E
5      140 ACCEPT     icmp --  em1    p3p1    0.0.0.0/0              0.0.0.0/0              icmptype 8 state NEW,E
```

Firewall Wireshark capture of above.



```
3693 877.80794900 192.168.0.4     192.168.10.2    ICMP    60 Echo (ping) request  id=0x7537, seq=0/0, ttl=64 (reply in 3694)
3694 877.80823000 192.168.10.2    192.168.0.4     ICMP    42 Echo (ping) reply    id=0x7537, seq=0/0, ttl=63 (request in 3693
3698 878.80803400 192.168.0.4     192.168.10.2    ICMP    60 Echo (ping) request  id=0x7537, seq=256/1, ttl=64 (reply in 3699
3699 878.80836000 192.168.10.2    192.168.0.4     ICMP    42 Echo (ping) reply    id=0x7537, seq=256/1, ttl=63 (request in 36
3704 879.80812300 192.168.0.4     192.168.10.2    ICMP    60 Echo (ping) request  id=0x7537, seq=512/2, ttl=64 (reply in 3705
3705 879.80839500 192.168.10.2    192.168.0.4     ICMP    42 Echo (ping) reply    id=0x7537, seq=512/2, ttl=63 (request in 37
3707 880.80820500 192.168.0.4     192.168.10.2    ICMP    60 Echo (ping) request  id=0x7537, seq=768/3, ttl=64 (reply in 3708
3708 880.80852800 192.168.10.2    192.168.0.4     ICMP    42 Echo (ping) reply    id=0x7537, seq=768/3, ttl=63 (request in 37
3711 881.80830100 192.168.0.4     192.168.10.2    ICMP    60 Echo (ping) request  id=0x7537, seq=1024/4, ttl=64 (reply in 371
3712 881.80857200 192.168.10.2    192.168.0.4     ICMP    42 Echo (ping) reply    id=0x7537, seq=1024/4, ttl=63 (request in 3
```

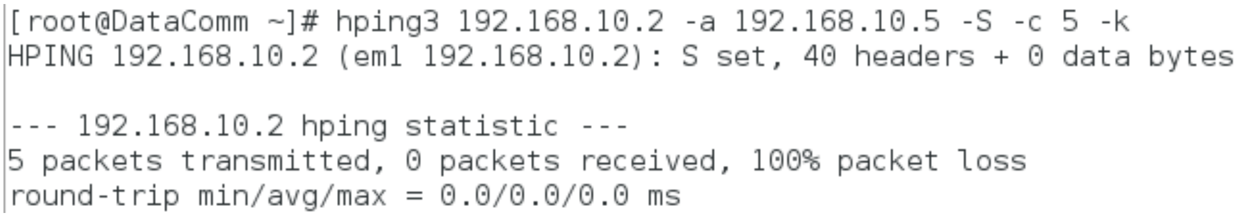
Internal Host Wireshark capture of above.



```
798 3019.9932710 192.168.0.4     192.168.10.2    IPv4    1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=001b) [Reassembled in #79
799 3019.9932900 192.168.0.4     192.168.10.2    TCP      622 [TCP segment of a reassembled PDU]
800 3019.9933150 192.168.10.2    192.168.0.4     TCP       54 http > idotdist [RST, ACK] Seq=1 Ack=2049 Win=0 Len=0
802 3020.9933100 192.168.0.4     192.168.10.2    IPv4    1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=001b) [Reassembled in #80
803 3020.9933360 192.168.0.4     192.168.10.2    TCP      622 [TCP Port numbers reused] [TCP segment of a reassembled PDU]
804 3020.9933570 192.168.10.2    192.168.0.4     TCP       54 http > idotdist [RST, ACK] Seq=1 Ack=2049 Win=0 Len=0
805 3021.9933940 192.168.0.4     192.168.10.2    IPv4    1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=001b)
806 3021.9934150 192.168.0.4     192.168.10.2    TCP      622 [TCP Port numbers reused] [TCP segment of a reassembled PDU]
807 3021.9934410 192.168.10.2    192.168.0.4     TCP       54 http > idotdist [RST, ACK] Seq=1 Ack=2049 Win=0 Len=0
808 3022.9934820 192.168.0.4     192.168.10.2    IPv4    1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=001b) [Reassembled in #80
809 3022.9935030 192.168.0.4     192.168.10.2    TCP      622 [TCP Port numbers reused] [TCP segment of a reassembled PDU]
810 3022.9935290 192.168.10.2    192.168.0.4     TCP       54 http > idotdist [RST, ACK] Seq=1 Ack=2049 Win=0 Len=0
811 3023.9935680 192.168.0.4     192.168.10.2    IPv4    1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=001b)
812 3023.9935880 192.168.0.4     192.168.10.2    TCP      622 [TCP Port numbers reused] [TCP segment of a reassembled PDU]
813 3023.9936130 192.168.10.2    192.168.0.4     TCP       54 http > idotdist [RST, ACK] Seq=1 Ack=2049 Win=0 Len=0
```

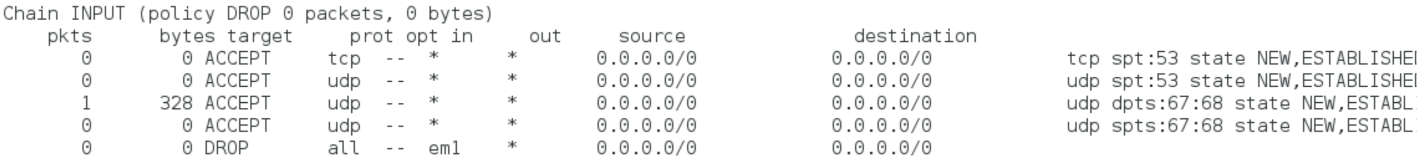
Test Case 5: Do not accept any packets with a source address from the outside matching your internal network.

To test, we sent the command below.



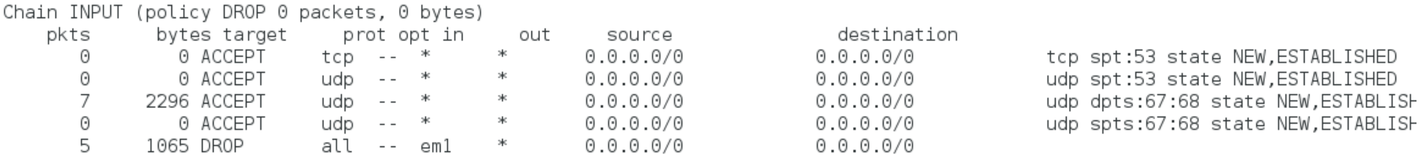
```
[root@DataComm ~]# hping3 192.168.10.2 -a 192.168.10.5 -S -c 5 -k
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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
```

IPtables before entering the command.



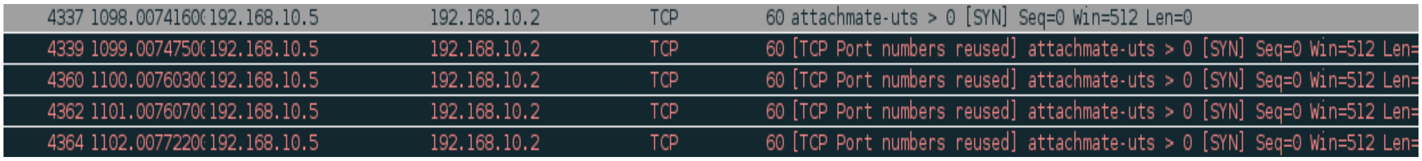
```
Chain INPUT (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:53 state NEW,ESTABLISHE
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:53 state NEW,ESTABLISHE
       1      328 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpts:67:68 state NEW,ESTABL
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spts:67:68 state NEW,ESTABL
       0        0 DROP       all  --  em1    *       0.0.0.0/0            0.0.0.0/0
```

IPtables after entering the command.



```
Chain INPUT (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:53 state NEW,ESTABLISHED
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:53 state NEW,ESTABLISHED
       7     2296 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpts:67:68 state NEW,ESTABLIS
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spts:67:68 state NEW,ESTABLIS
       5     1065 DROP       all  --  em1    *       0.0.0.0/0            0.0.0.0/0
```

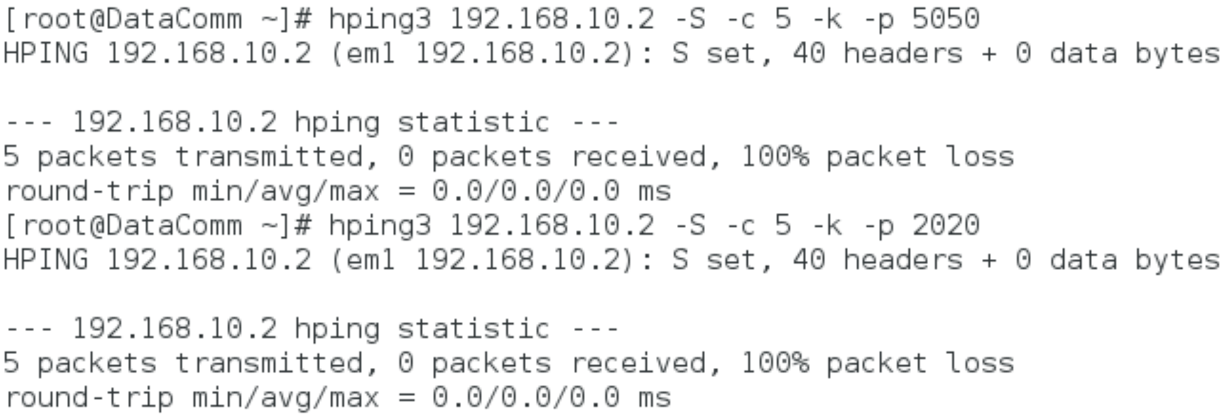
Wireshark capturing the dropped packets because the source IP matched the internal network.



```
4337 1098.0074160( 192.168.10.5    192.168.10.2    TCP    60 attachmate-uts > 0 [SYN] Seq=0 Win=512 Len=0
4339 1099.0074750( 192.168.10.5    192.168.10.2    TCP    60 [TCP Port numbers reused] attachmate-uts > 0 [SYN] Seq=0 Win=512 Len=
4360 1100.0076030( 192.168.10.5    192.168.10.2    TCP    60 [TCP Port numbers reused] attachmate-uts > 0 [SYN] Seq=0 Win=512 Len=
4362 1101.0076070( 192.168.10.5    192.168.10.2    TCP    60 [TCP Port numbers reused] attachmate-uts > 0 [SYN] Seq=0 Win=512 Len=
4364 1102.0077220( 192.168.10.5    192.168.10.2    TCP    60 [TCP Port numbers reused] attachmate-uts > 0 [SYN] Seq=0 Win=512 Len=
```

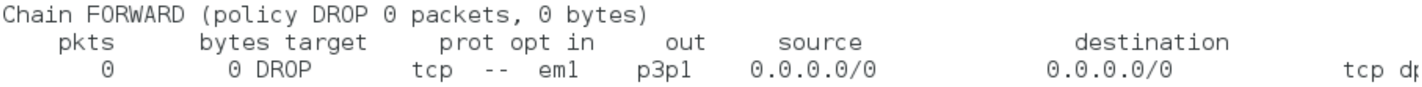
Test Case 6: Reject those connections that are coming the 'wrong' way (incoming SYN to high ports).

To test. we entered the below commands:



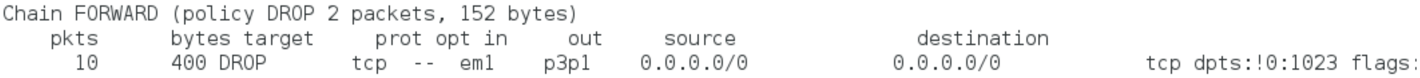
```
[root@DataComm ~]# hping3 192.168.10.2 -S -c 5 -k -p 5050
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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 ~]# hping3 192.168.10.2 -S -c 5 -k -p 2020
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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
```

IPtables before the command



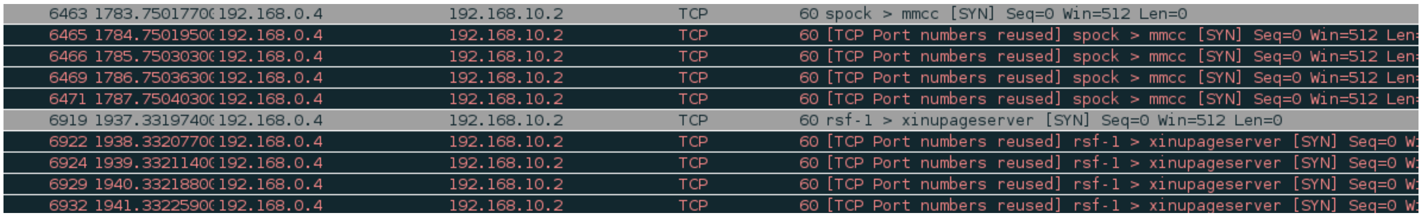
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0         0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dp
```

IPtables after the command. We see our rule dropping the 10 packets.



```
Chain FORWARD (policy DROP 2 packets, 152 bytes)
    pkts      bytes target     prot opt in     out     source               destination
      10      400 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:
```

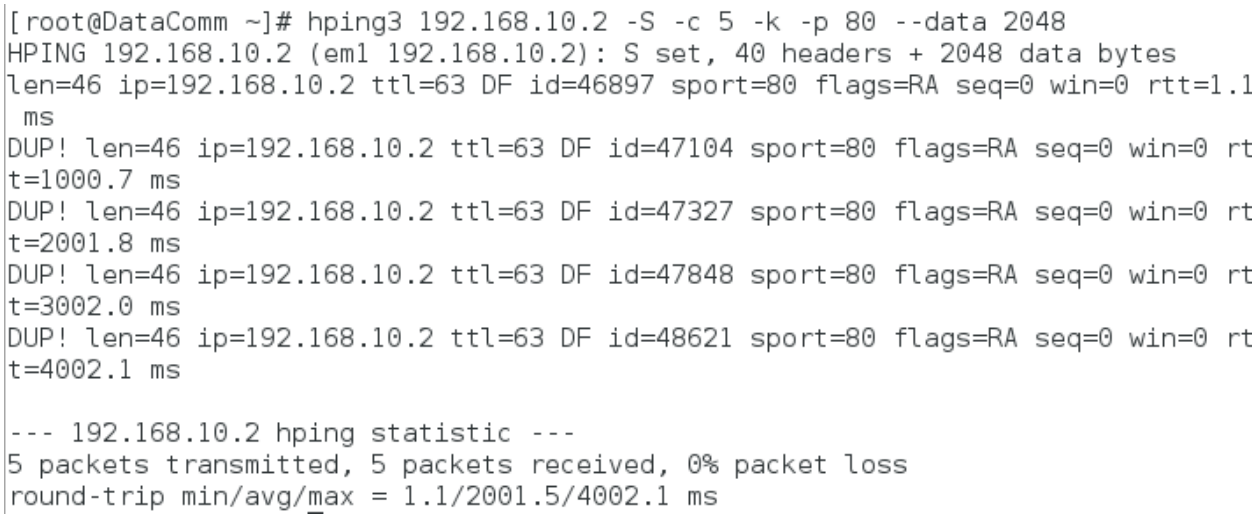
Firewall Wireshark capture of above.



```
6463 1783.7501770 192.168.0.4    192.168.10.2    TCP    60 spock > mmcc [SYN] Seq=0 Win=512 Len=0
6465 1784.7501950 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] spock > mmcc [SYN] Seq=0 Win=512 Len
6466 1785.7503030 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] spock > mmcc [SYN] Seq=0 Win=512 Len
6469 1786.7503630 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] spock > mmcc [SYN] Seq=0 Win=512 Len
6471 1787.7504030 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] spock > mmcc [SYN] Seq=0 Win=512 Len
6919 1937.3319740 192.168.0.4    192.168.10.2    TCP    60 rsf-1 > xinupageserver [SYN] Seq=0 Win=512 Len=0
6922 1938.3320770 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] rsf-1 > xinupageserver [SYN] Seq=0 W
6924 1939.3321140 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] rsf-1 > xinupageserver [SYN] Seq=0 W
6929 1940.3321880 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] rsf-1 > xinupageserver [SYN] Seq=0 W
6932 1941.3322590 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] rsf-1 > xinupageserver [SYN] Seq=0 W
```

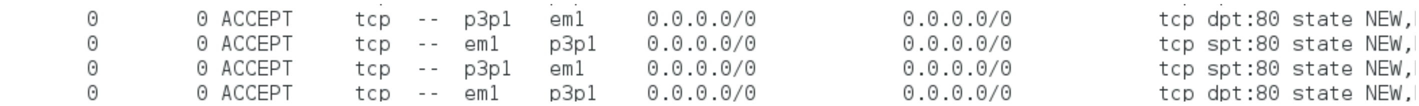
Test Case 7: Accept fragmented packets

We sent fragments using the following command:



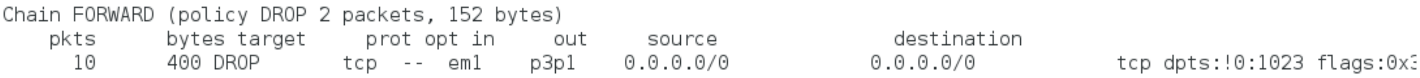
```
[root@DataComm ~]# hping3 192.168.10.2 -S -c 5 -k -p 80 --data 2048
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 2048 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=46897 sport=80 flags=RA seq=0 win=0 rtt=1.1
 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=47104 sport=80 flags=RA seq=0 win=0 rt
t=1000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=47327 sport=80 flags=RA seq=0 win=0 rt
t=2001.8 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=47848 sport=80 flags=RA seq=0 win=0 rt
t=3002.0 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=48621 sport=80 flags=RA seq=0 win=0 rt
t=4002.1 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.5/4002.1 ms
```

IPtables before the command.



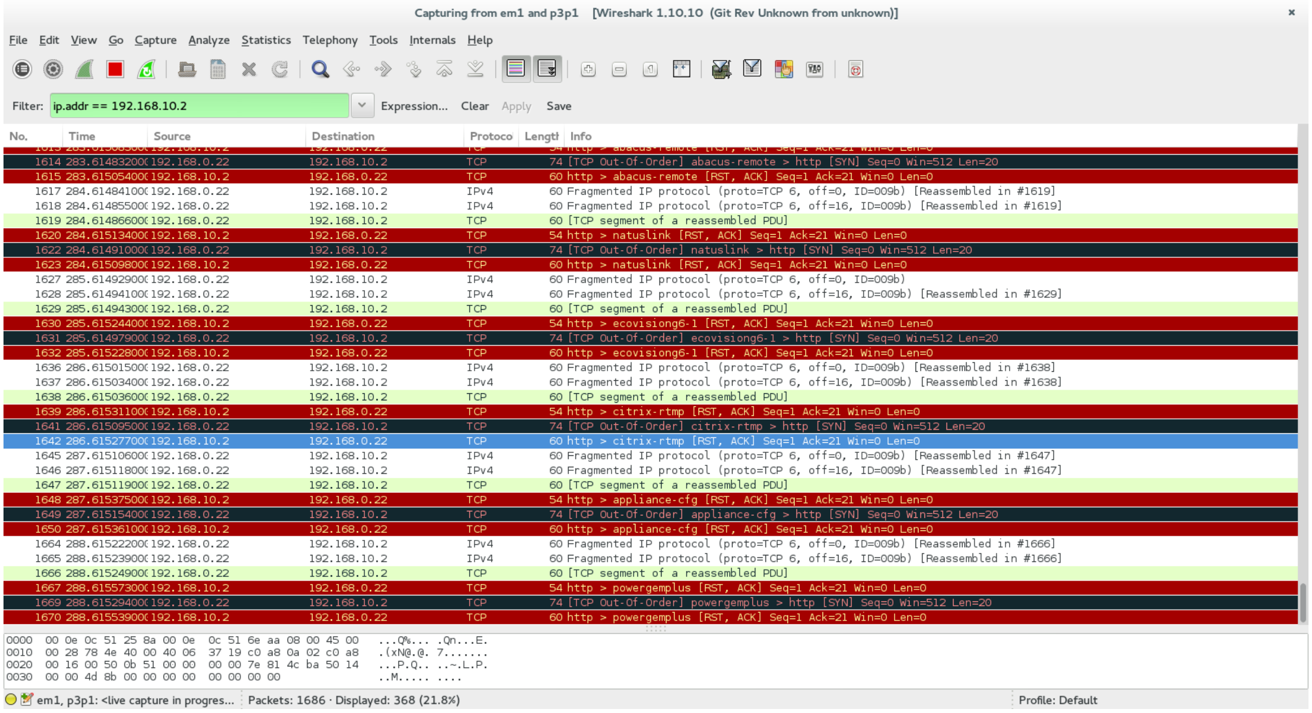
```
       0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:80 state NEW,
       0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,
       0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:80 state NEW,
       0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:80 state NEW,
```

IPtables after the command.



```
Chain FORWARD (policy DROP 2 packets, 152 bytes)
    pkts      bytes target     prot opt in     out     source               destination
      10      400 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:0x3
```

Wireshark Capture of above. We see the packets being fragmented by the Firewall, and then accepted by the internal host.



Capturing from em1 and p3p1   [Wireshark 1.10.10 (Git Rev Unknown from unknown)]

Filter: ip.addr == 192.168.10.2

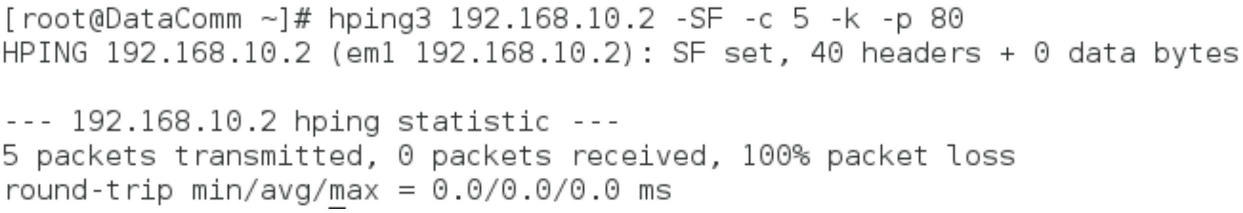
```
No.   Time          Source         Destination    Protoco Length Info
1613 283.61508300( 192.168.10.2   192.168.0.22   TCP     54 http > abacus-remote [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1614 283.61483200( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] abacus-remote > http [SYN] Seq=0 Win=512 Len=20
1615 283.61505400( 192.168.10.2   192.168.0.22   TCP     60 http > abacus-remote [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1617 284.61484100( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=0, ID=009b) [Reassembled in #1619]
1618 284.61485500( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=16, ID=009b) [Reassembled in #1619]
1619 284.61486600( 192.168.0.22   192.168.10.2   TCP     60 [TCP segment of a reassembled PDU]
1620 284.61513400( 192.168.10.2   192.168.0.22   TCP     54 http > natuslink [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1622 284.61491000( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] natuslink > http [SYN] Seq=0 Win=512 Len=20
1623 284.61509800( 192.168.10.2   192.168.0.22   TCP     60 http > natuslink [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1627 285.61492900( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=0, ID=009b)
1628 285.61494100( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=16, ID=009b) [Reassembled in #1629]
1629 285.61494300( 192.168.0.22   192.168.10.2   TCP     60 [TCP segment of a reassembled PDU]
1630 285.61524400( 192.168.10.2   192.168.0.22   TCP     54 http > ecovisiong6-1 [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1631 285.61497900( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] ecovisiong6-1 > http [SYN] Seq=0 Win=512 Len=20
1632 285.61522800( 192.168.10.2   192.168.0.22   TCP     60 http > ecovisiong6-1 [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1636 286.61501500( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=0, ID=009b) [Reassembled in #1638]
1637 286.61503400( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=16, ID=009b) [Reassembled in #1638]
1638 286.61503600( 192.168.0.22   192.168.10.2   TCP     60 [TCP segment of a reassembled PDU]
1639 286.61531100( 192.168.10.2   192.168.0.22   TCP     54 http > citrix-rtmp [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1641 286.61509500( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] citrix-rtmp > http [SYN] Seq=0 Win=512 Len=20
1642 286.61527700( 192.168.10.2   192.168.0.22   TCP     60 http > citrix-rtmp [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1645 287.61510600( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=0, ID=009b) [Reassembled in #1647]
1646 287.61511800( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=16, ID=009b) [Reassembled in #1647]
1647 287.61511900( 192.168.0.22   192.168.10.2   TCP     60 [TCP segment of a reassembled PDU]
1648 287.61537500( 192.168.10.2   192.168.0.22   TCP     54 http > appliance-cfg [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1649 287.61515400( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] appliance-cfg > http [SYN] Seq=0 Win=512 Len=20
1650 287.61536100( 192.168.10.2   192.168.0.22   TCP     60 http > appliance-cfg [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1664 288.61522200( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=0, ID=009b) [Reassembled in #1666]
1665 288.61523900( 192.168.0.22   192.168.10.2   IPv4    60 Fragmented IP protocol (proto=TCP 6, off=16, ID=009b) [Reassembled in #1666]
1666 288.61524900( 192.168.0.22   192.168.10.2   TCP     60 [TCP segment of a reassembled PDU]
1667 288.61557300( 192.168.10.2   192.168.0.22   TCP     54 http > powergemplus [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
1669 288.61529400( 192.168.0.22   192.168.10.2   TCP     74 [TCP Out-Of-Order] powergemplus > http [SYN] Seq=0 Win=512 Len=20
1670 288.61553900( 192.168.10.2   192.168.0.22   TCP     60 http > powergemplus [RST, ACK] Seq=1 Ack=21 Win=0 Len=0
```

```
0000  00 0e 0c 51 25 8a 00 0e  0c 51 6e aa 08 00 45 00   ...Q%... .Qn...E.
0010  00 28 78 4e 40 00 40 06  37 19 c0 a8 0a 02 c0 a8   .(xN@.@. 7.......
0020  00 16 00 50 0b 51 00 00  00 00 7e 81 4c ba 50 14   ...P.Q.. ..~.L.P.
0030  00 00 4d 8b 00 00 00 00  00 00 00 00               ..M..... ....
```

em1, p3p1: <live capture in progres... Packets: 1686 · Displayed: 368 (21.8%) Profile: Default

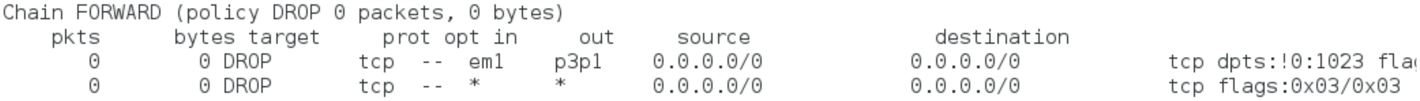
Test Case 8: Drop all TCP packets with the SYN and FIN bit set

To test, we sent a packet on an open TCP port (80) with the S and F set. The below command was used.



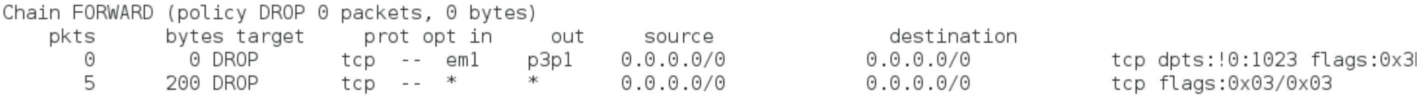
```
[root@DataComm ~]# hping3 192.168.10.2 -SF -c 5 -k -p 80
HPING 192.168.10.2 (em1 192.168.10.2): SF set, 40 headers + 0 data bytes

--- 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
```

IPtables before the command:



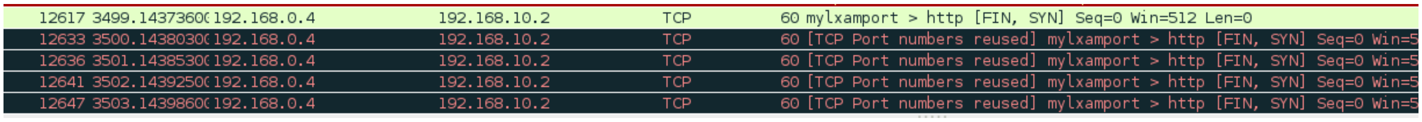
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination          
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flag
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
```

IPtables after:



```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination          
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:0x3
       5      200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
```

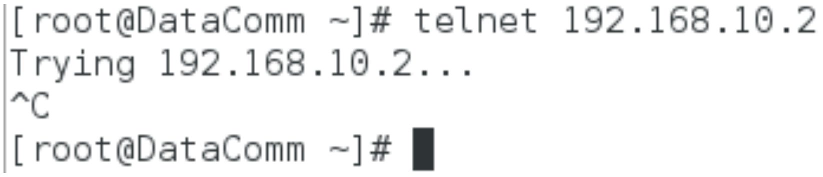
Firewall Wireshark Capture of the dropped packets.



```
12617 3499.1437360( 192.168.0.4    192.168.10.2    TCP    60 mylxamport > http [FIN, SYN] Seq=0 Win=512 Len=0
12633 3500.1438030( 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] mylxamport > http [FIN, SYN] Seq=0 Win=5
12636 3501.1438530( 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] mylxamport > http [FIN, SYN] Seq=0 Win=5
12641 3502.1439250( 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] mylxamport > http [FIN, SYN] Seq=0 Win=5
12647 3503.1439860( 192.168.0.4    192.168.10.2    TCP    60 [TCP Port numbers reused] mylxamport > http [FIN, SYN] Seq=0 Win=5
```

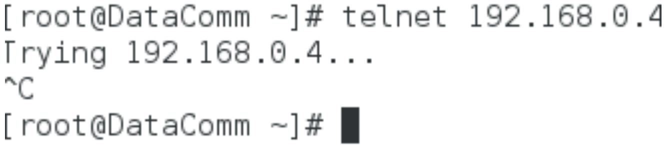
Test Case 9: Do not allow Telnet packets

To test, we tried to Telnet the internal host:



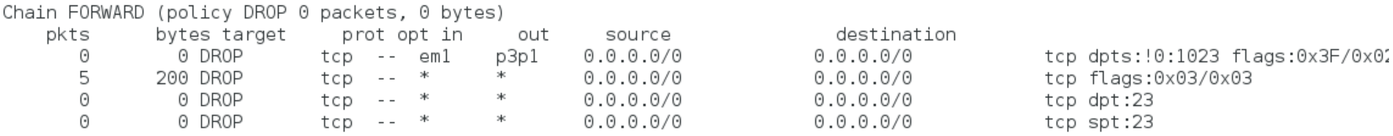
```
[root@DataComm ~]# telnet 192.168.10.2
Trying 192.168.10.2...
^C
[root@DataComm ~]# █
```

We also tried to telnet the external host from the internal host.



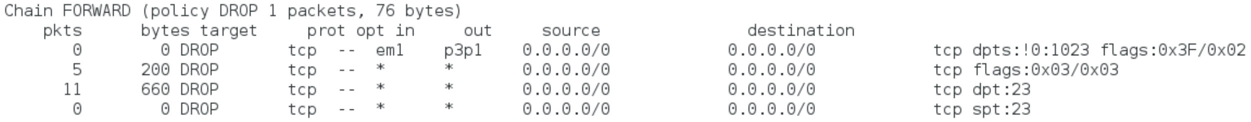
```
[root@DataComm ~]# telnet 192.168.0.4
Trying 192.168.0.4...
^C
[root@DataComm ~]# █
```

IPtables before the attempted Telnet connections:



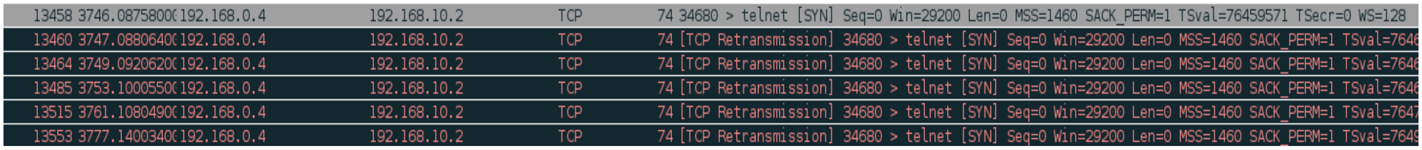
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:0x3F/0x0
       5      200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       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
```

IPtables after the attempted Telnet connection:



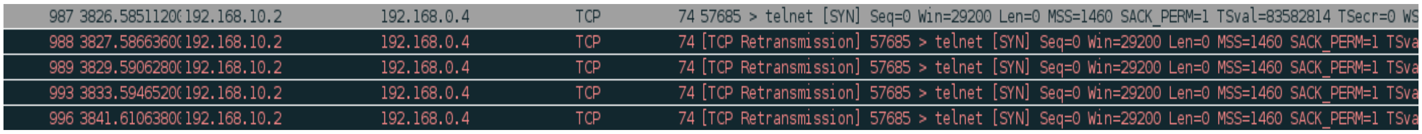
```
Chain FORWARD (policy DROP 1 packets, 76 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:!0:1023 flags:0x3F/0x02
       5      200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
      11      660 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
```

Firewall connection Wireshark capture of the Telnet attempt:



```
13458 3746.0875800 192.168.0.4    192.168.10.2    TCP    74 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=76459571 TSecr=0 WS=128
13460 3747.0880640 192.168.0.4    192.168.10.2    TCP    74 [TCP Retransmission] 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7646
13464 3749.0920620 192.168.0.4    192.168.10.2    TCP    74 [TCP Retransmission] 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7646
13485 3753.1000550 192.168.0.4    192.168.10.2    TCP    74 [TCP Retransmission] 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7646
13515 3761.1080490 192.168.0.4    192.168.10.2    TCP    74 [TCP Retransmission] 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7647
13553 3777.1400340 192.168.0.4    192.168.10.2    TCP    74 [TCP Retransmission] 34680 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=7649
```

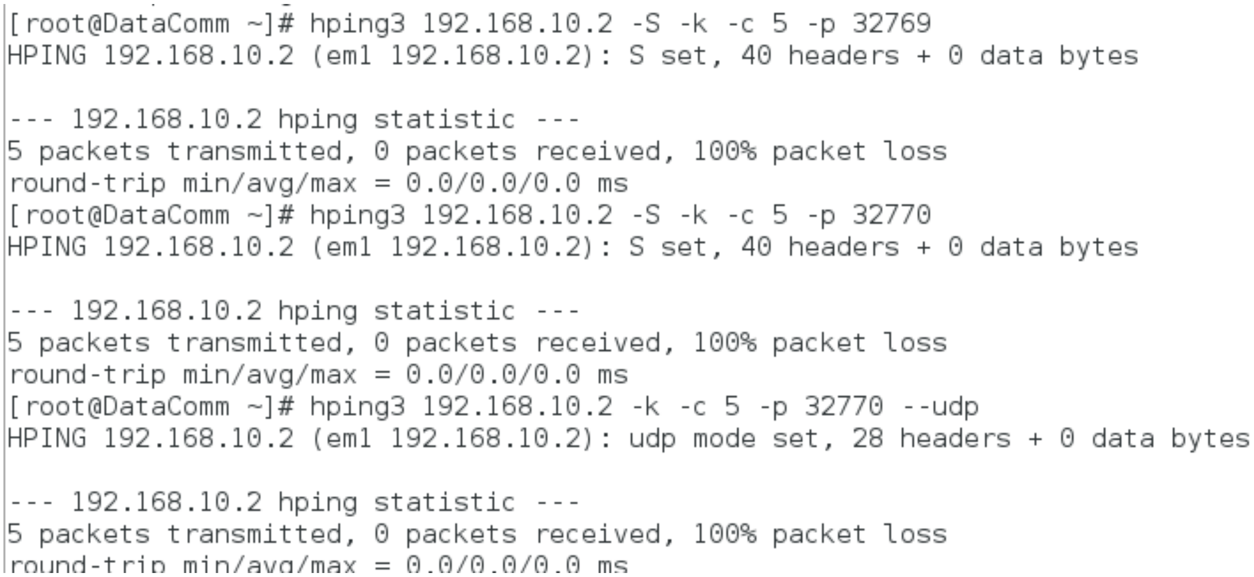
Internal Host Wireshark capture of the Telnet connection:



```
  987 3826.5851120 192.168.10.2    192.168.0.4    TCP    74 57685 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=83582814 TSecr=0 WS
  988 3827.5866360 192.168.10.2    192.168.0.4    TCP    74 [TCP Retransmission] 57685 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
  989 3829.5906280 192.168.10.2    192.168.0.4    TCP    74 [TCP Retransmission] 57685 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
  993 3833.5946520 192.168.10.2    192.168.0.4    TCP    74 [TCP Retransmission] 57685 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
  996 3841.6106380 192.168.10.2    192.168.0.4    TCP    74 [TCP Retransmission] 57685 > telnet [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
```

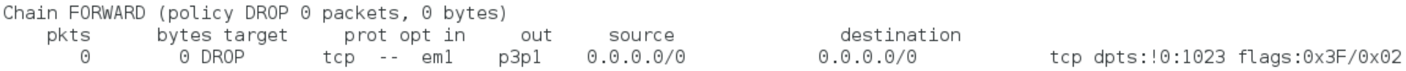
Test Case 10: Block all external traffic directed to ports 32768-32775

To test we entered the commands below:



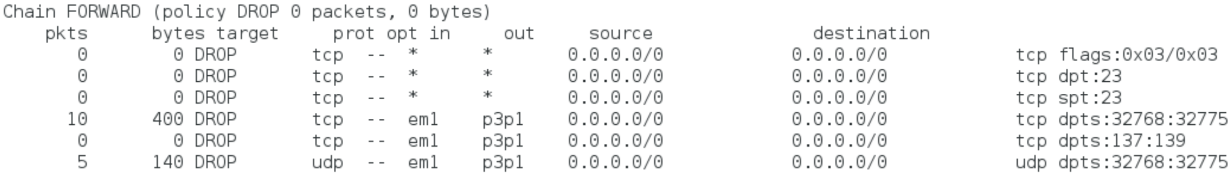
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 32769
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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 ~]# hping3 192.168.10.2 -S -k -c 5 -p 32770
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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 ~]# hping3 192.168.10.2 -k -c 5 -p 32770 --udp
HPING 192.168.10.2 (em1 192.168.10.2): udp mode set, 28 headers + 0 data bytes

--- 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
```

IPtables before the command:



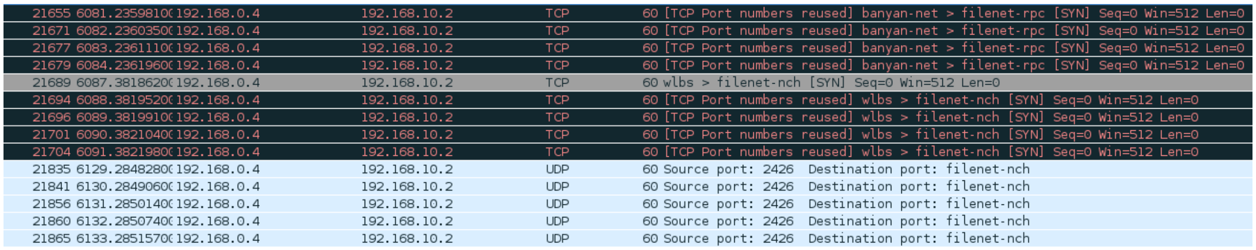
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0           tcp dpts:!0:1023 flags:0x3F/0x02
```

IPtables after:



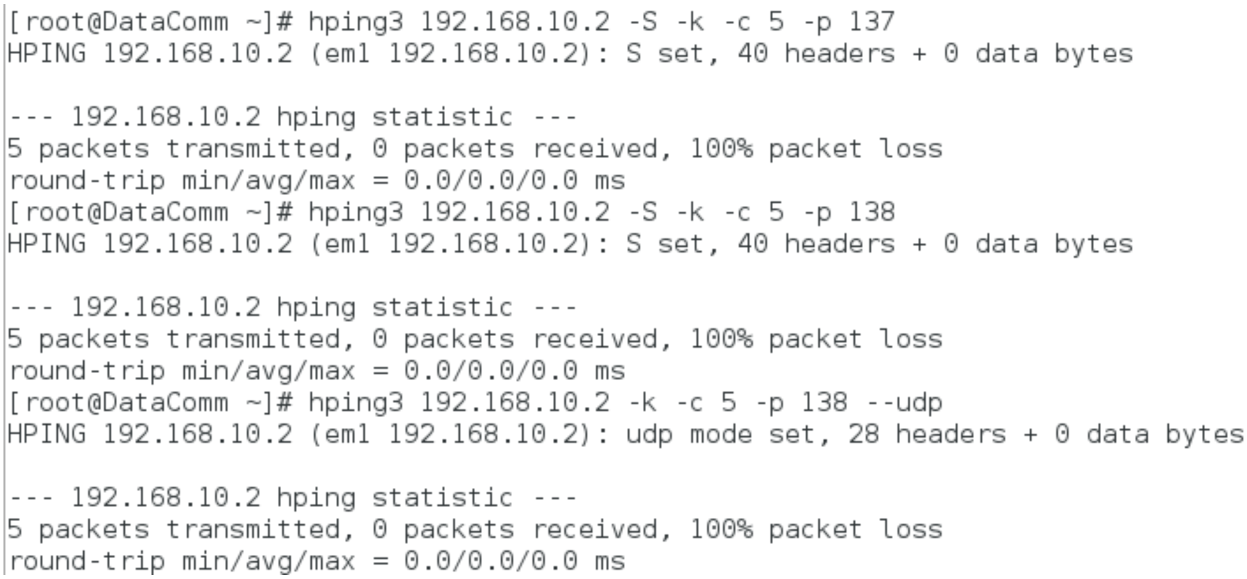
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       0        0 DROP       tcp  --  *      *       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
      10      400 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:32775
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       5      140 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:32775
```

Firewall Wireshark capture of above:



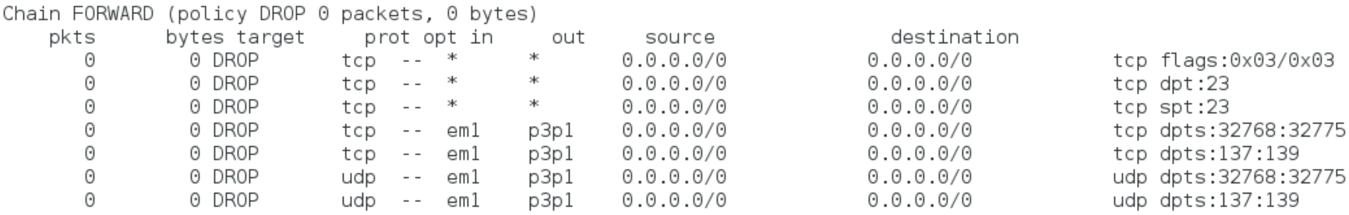
```
21655 6081.2359810(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] banyan-net > filenet-rpc [SYN] Seq=0 Win=512 Len=0
21671 6082.2360350(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] banyan-net > filenet-rpc [SYN] Seq=0 Win=512 Len=0
21677 6083.2361110(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] banyan-net > filenet-rpc [SYN] Seq=0 Win=512 Len=0
21679 6084.2361960(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] banyan-net > filenet-rpc [SYN] Seq=0 Win=512 Len=0
21689 6087.3818620(192.168.0.4         192.168.10.2         TCP        60 wlbs > filenet-nch [SYN] Seq=0 Win=512 Len=0
21694 6088.3819520(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] wlbs > filenet-nch [SYN] Seq=0 Win=512 Len=0
21696 6089.3819910(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] wlbs > filenet-nch [SYN] Seq=0 Win=512 Len=0
21701 6090.3821040(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] wlbs > filenet-nch [SYN] Seq=0 Win=512 Len=0
21704 6091.3821980(192.168.0.4         192.168.10.2         TCP        60 [TCP Port numbers reused] wlbs > filenet-nch [SYN] Seq=0 Win=512 Len=0
21835 6129.2848280(192.168.0.4         192.168.10.2         UDP        60 Source port: 2426  Destination port: filenet-nch
21841 6130.2849060(192.168.0.4         192.168.10.2         UDP        60 Source port: 2426  Destination port: filenet-nch
21856 6131.2850140(192.168.0.4         192.168.10.2         UDP        60 Source port: 2426  Destination port: filenet-nch
21860 6132.2850740(192.168.0.4         192.168.10.2         UDP        60 Source port: 2426  Destination port: filenet-nch
21865 6133.2851570(192.168.0.4         192.168.10.2         UDP        60 Source port: 2426  Destination port: filenet-nch
```

Test Case 11: Block all external traffic directed to ports 137-139
We tested all 3 ports by entereing the following commands:



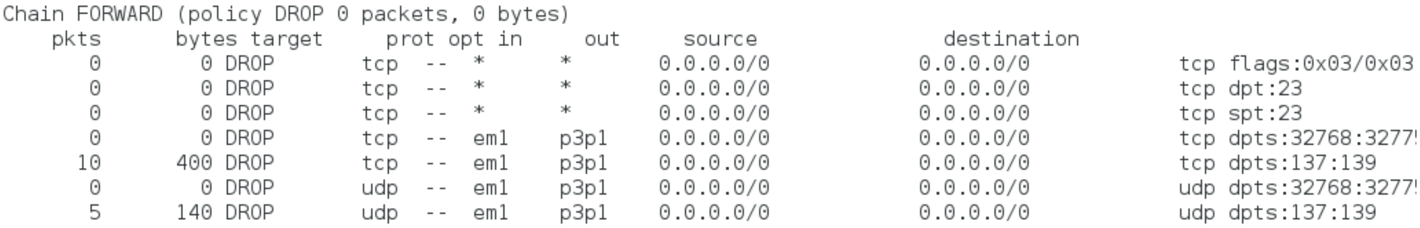
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 137
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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 ~]# hping3 192.168.10.2 -S -k -c 5 -p 138
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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 ~]# hping3 192.168.10.2 -k -c 5 -p 138 --udp
HPING 192.168.10.2 (em1 192.168.10.2): udp mode set, 28 headers + 0 data bytes

--- 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
```

IPtables before:



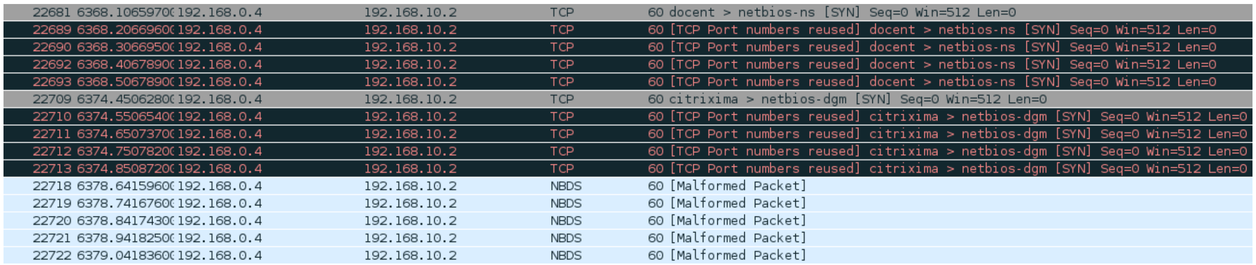
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0         0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       0         0 DROP       tcp  --  *      *       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:32775
       0         0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0         0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:32775
       0         0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
```

IPtables after:



```
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 flags:0x03/0x03
       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:3277
      10       400 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0         0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:3277
       5       140 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
```

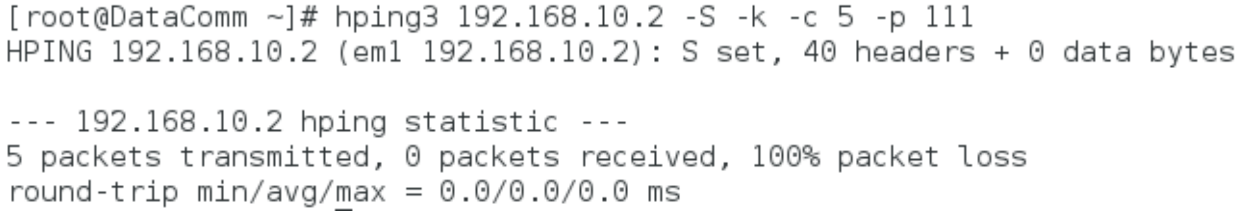
Firewall Wireshark of the dropped packets.



```
  22681 6368.1065970( 192.168.0.4          192.168.10.2       TCP         60 docent > netbios-ns [SYN] Seq=0 Win=512 Len=0
  22689 6368.2066960( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] docent > netbios-ns [SYN] Seq=0 Win=512 Len=0
  22690 6368.3066950( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] docent > netbios-ns [SYN] Seq=0 Win=512 Len=0
  22692 6368.4067890( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] docent > netbios-ns [SYN] Seq=0 Win=512 Len=0
  22693 6368.5067890( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] docent > netbios-ns [SYN] Seq=0 Win=512 Len=0
  22709 6374.4506280( 192.168.0.4          192.168.10.2       TCP         60 citrixima > netbios-dgm [SYN] Seq=0 Win=512 Len=0
  22710 6374.5506540( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] citrixima > netbios-dgm [SYN] Seq=0 Win=512 Len=0
  22711 6374.6507370( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] citrixima > netbios-dgm [SYN] Seq=0 Win=512 Len=0
  22712 6374.7507820( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] citrixima > netbios-dgm [SYN] Seq=0 Win=512 Len=0
  22713 6374.8508720( 192.168.0.4          192.168.10.2       TCP         60 [TCP Port numbers reused] citrixima > netbios-dgm [SYN] Seq=0 Win=512 Len=0
  22718 6378.6415960( 192.168.0.4          192.168.10.2       NBDS        60 [Malformed Packet]
  22719 6378.7416760( 192.168.0.4          192.168.10.2       NBDS        60 [Malformed Packet]
  22720 6378.8417430( 192.168.0.4          192.168.10.2       NBDS        60 [Malformed Packet]
  22721 6378.9418250( 192.168.0.4          192.168.10.2       NBDS        60 [Malformed Packet]
  22722 6379.0418360( 192.168.0.4          192.168.10.2       NBDS        60 [Malformed Packet]
```

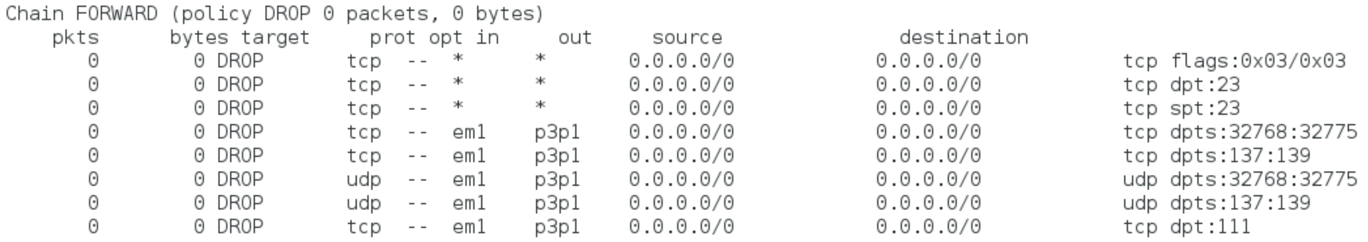
Test Case 12: Block all external traffic directed to TCP on port 111

We entered the following command:



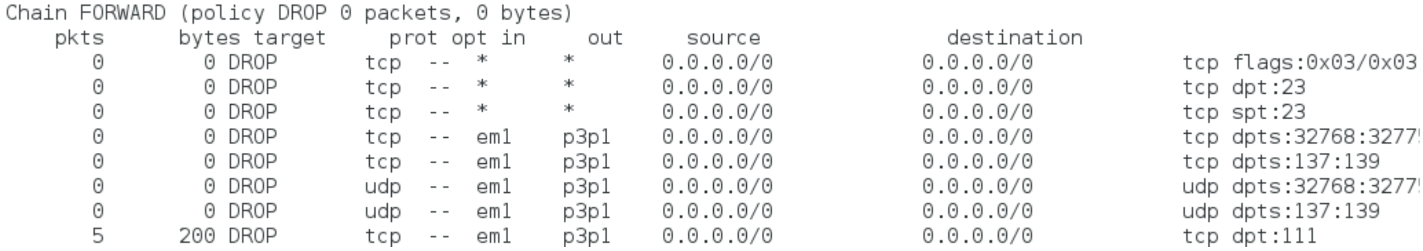
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 111
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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
```

IPtables before:



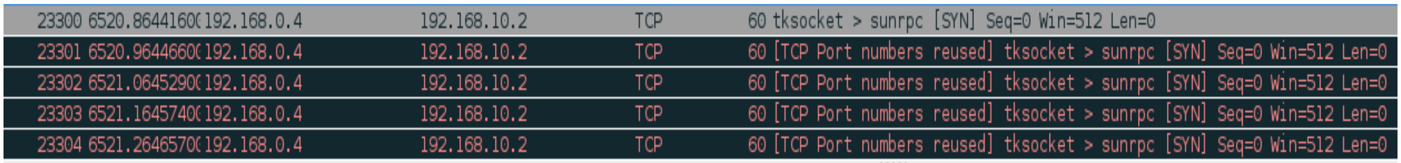
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       0        0 DROP       tcp  --  *      *       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:32775
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:32775
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:111
```

IPtables after:



```
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 flags:0x03/0x03
       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:3277
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:3277
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
       5      200 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:111
```

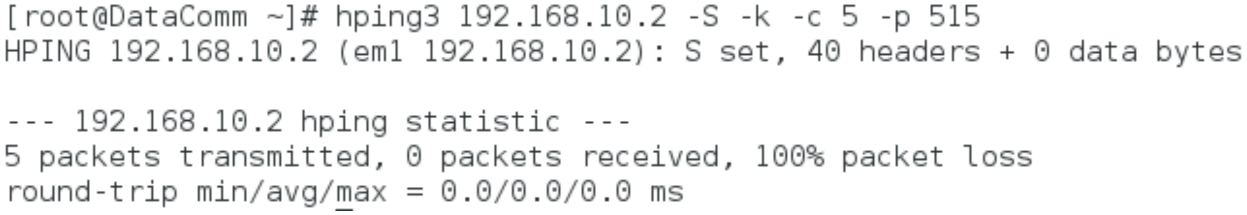
Firewall Wireshark capture of above:



```
23300 6520.8644160 192.168.0.4      192.168.10.2      TCP      60 tkscocket > sunrpc [SYN] Seq=0 Win=512 Len=0
23301 6520.9644660 192.168.0.4      192.168.10.2      TCP      60 [TCP Port numbers reused] tksocket > sunrpc [SYN] Seq=0 Win=512 Len=0
23302 6521.0645290 192.168.0.4      192.168.10.2      TCP      60 [TCP Port numbers reused] tksocket > sunrpc [SYN] Seq=0 Win=512 Len=0
23303 6521.1645740 192.168.0.4      192.168.10.2      TCP      60 [TCP Port numbers reused] tksocket > sunrpc [SYN] Seq=0 Win=512 Len=0
23304 6521.2646570 192.168.0.4      192.168.10.2      TCP      60 [TCP Port numbers reused] tksocket > sunrpc [SYN] Seq=0 Win=512 Len=0
```

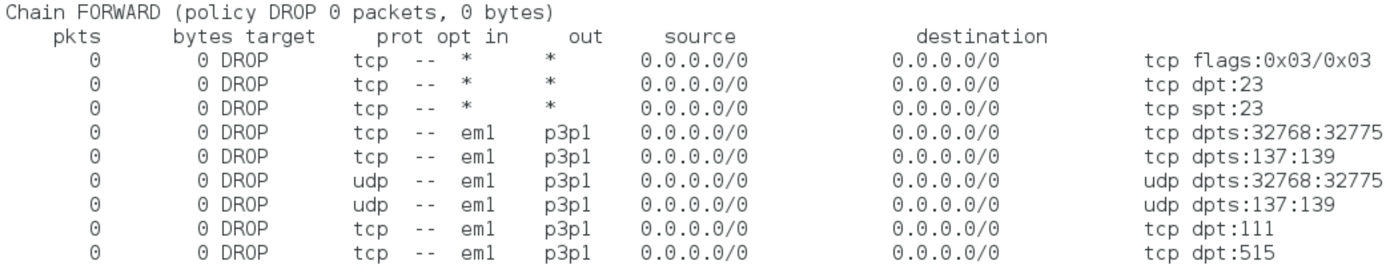
Test Case 13: Block all external traffic directed to TCP on port 515

To test we entered the below command:



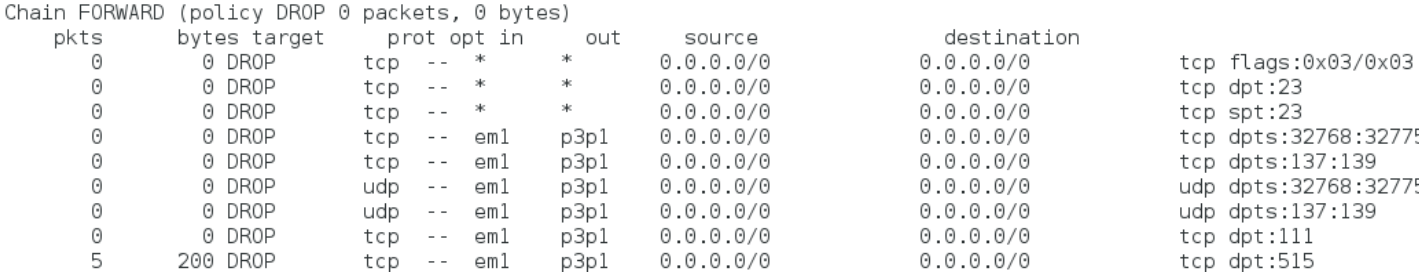
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 515
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes

--- 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
```

IPtables before:



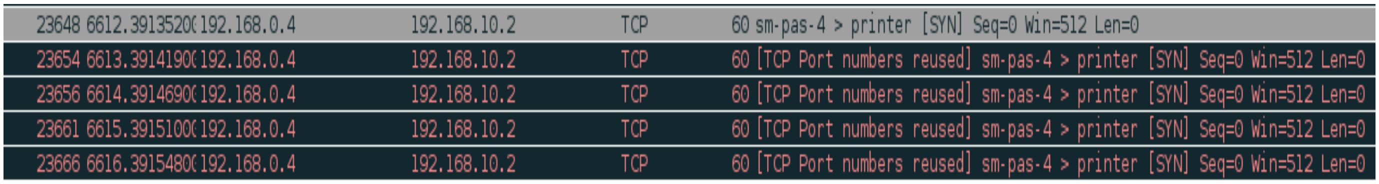
```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp flags:0x03/0x03
       0        0 DROP       tcp  --  *      *       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:32775
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:32775
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:111
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:515
```

IPtables after:



```
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 flags:0x03/0x03
       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 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:32768:32775
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpts:137:139
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:32768:32775
       0        0 DROP       udp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            udp dpts:137:139
       0        0 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:111
       5      200 DROP       tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:515
```

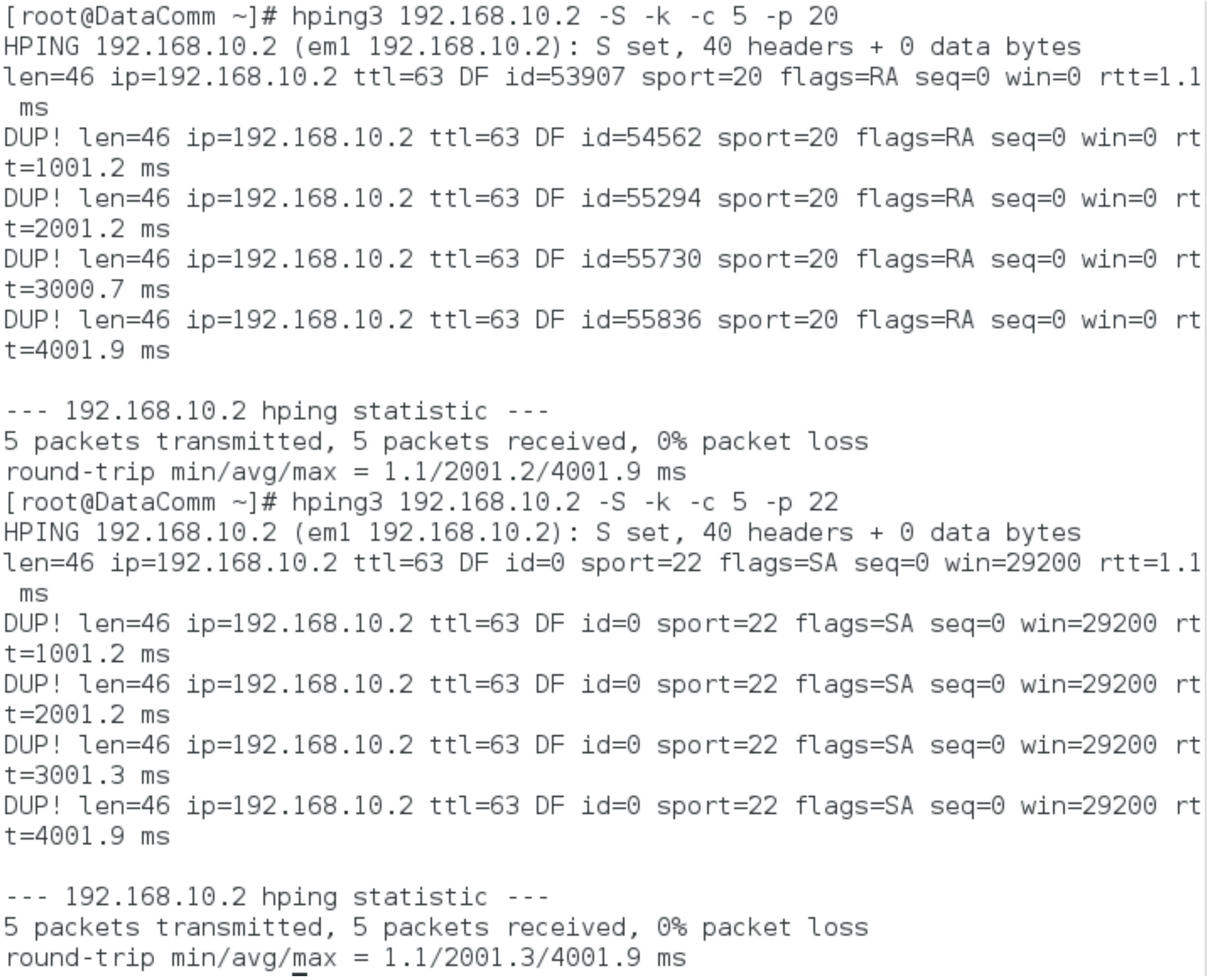
Wireshark Capture of above:



```
23648 6612.3913520 192.168.0.4     192.168.10.2     TCP     60 sm-pas-4 > printer [SYN] Seq=0 Win=512 Len=0
23654 6613.3914190 192.168.0.4     192.168.10.2     TCP     60 [TCP Port numbers reused] sm-pas-4 > printer [SYN] Seq=0 Win=512 Len=0
23656 6614.3914690 192.168.0.4     192.168.10.2     TCP     60 [TCP Port numbers reused] sm-pas-4 > printer [SYN] Seq=0 Win=512 Len=0
23661 6615.3915100 192.168.0.4     192.168.10.2     TCP     60 [TCP Port numbers reused] sm-pas-4 > printer [SYN] Seq=0 Win=512 Len=0
23666 6616.3915480 192.168.0.4     192.168.10.2     TCP     60 [TCP Port numbers reused] sm-pas-4 > printer [SYN] Seq=0 Win=512 Len=0
```

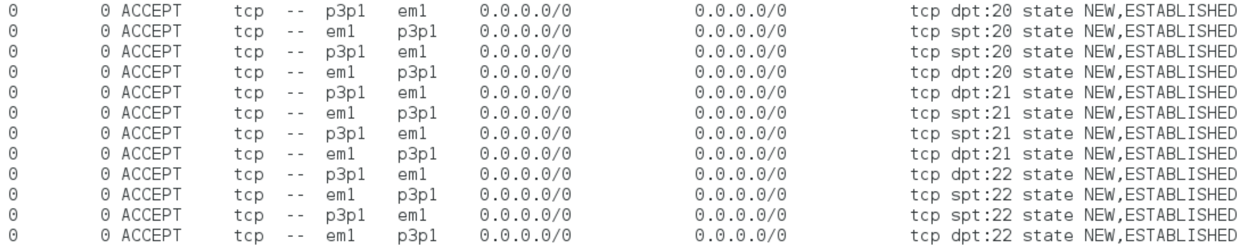
Test Case 14: Ensure FTP and SSH services have “Minimum Delay” (Requires a new DSCP column in wireshark. Value = ‘ip.dsfield.dscp’)

We entered the below command to test FTP followed by SSH:



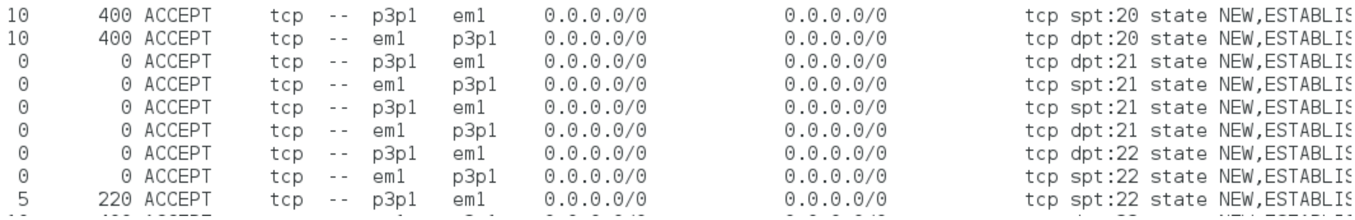
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 20
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=53907 sport=20 flags=RA seq=0 win=0 rtt=1.1
 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=54562 sport=20 flags=RA seq=0 win=0 rt
t=1001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=55294 sport=20 flags=RA seq=0 win=0 rt
t=2001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=55730 sport=20 flags=RA seq=0 win=0 rt
t=3000.7 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=55836 sport=20 flags=RA seq=0 win=0 rt
t=4001.9 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.2/4001.9 ms
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 22
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
t=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
t=2001.2 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
t=3001.3 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=29200 rt
t=4001.9 ms

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/2001.3/4001.9 ms
```

IPtables before:



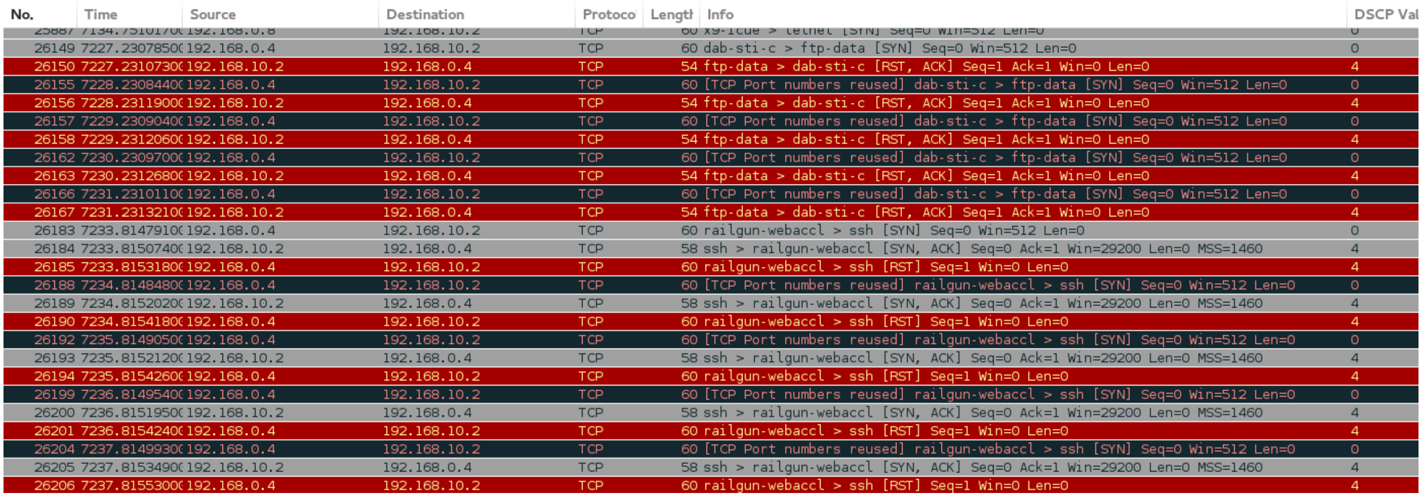
```
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:20 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:20 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:20 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:20 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:21 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:21 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:21 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:21 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:22 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:22 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:22 state NEW,ESTABLISHED
 0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:22 state NEW,ESTABLISHED
```

IPtables after:



```
   10      400 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:20 state NEW,ESTABLIS
   10      400 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:20 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:21 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:21 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:21 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:21 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:22 state NEW,ESTABLIS
    0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:22 state NEW,ESTABLIS
    5      220 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:22 state NEW,ESTABLIS
```

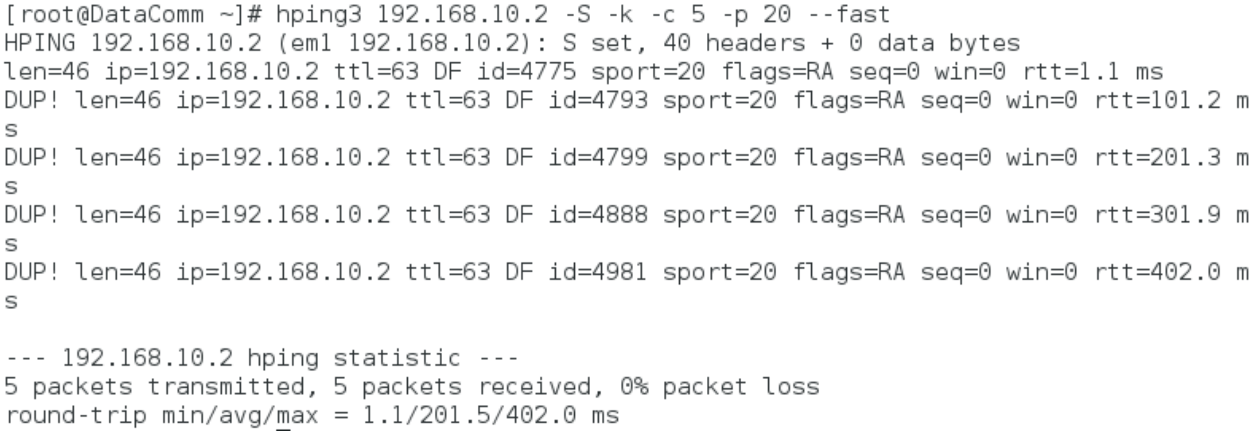
Wireshark capture of above. We see the DSCP column show a DCSP value of 4, which signifies the packet has minimum delay.



No.	Time	Source	Destination	Protoco	Lengtl	Info	DSCP Val
25887	7134.7510170	192.168.0.8	192.168.10.2	TCP	60	x9-icue > telnet [SYN] Seq=0 Win=512 Len=0	0
26149	7227.2307850	192.168.0.4	192.168.10.2	TCP	60	dab-sti-c > ftp-data [SYN] Seq=0 Win=512 Len=0	0
26150	7227.2310730	192.168.10.2	192.168.0.4	TCP	54	ftp-data > dab-sti-c [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	4
26155	7228.2308440	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] dab-sti-c > ftp-data [SYN] Seq=0 Win=512 Len=0	0
26156	7228.2311900	192.168.10.2	192.168.0.4	TCP	54	ftp-data > dab-sti-c [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	4
26157	7229.2309040	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] dab-sti-c > ftp-data [SYN] Seq=0 Win=512 Len=0	0
26158	7229.2312060	192.168.10.2	192.168.0.4	TCP	54	ftp-data > dab-sti-c [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	4
26162	7230.2309700	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] dab-sti-c > ftp-data [SYN] Seq=0 Win=512 Len=0	0
26163	7230.2312680	192.168.10.2	192.168.0.4	TCP	54	ftp-data > dab-sti-c [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	4
26166	7231.2310110	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] dab-sti-c > ftp-data [SYN] Seq=0 Win=512 Len=0	0
26167	7231.2313210	192.168.10.2	192.168.0.4	TCP	54	ftp-data > dab-sti-c [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	4
26183	7233.8147910	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [SYN] Seq=0 Win=512 Len=0	0
26184	7233.8150740	192.168.10.2	192.168.0.4	TCP	58	ssh > railgun-webaccl [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460	4
26185	7233.8153180	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [RST] Seq=1 Win=0 Len=0	4
26188	7234.8148480	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] railgun-webaccl > ssh [SYN] Seq=0 Win=512 Len=0	0
26189	7234.8152020	192.168.10.2	192.168.0.4	TCP	58	ssh > railgun-webaccl [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460	4
26190	7234.8154180	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [RST] Seq=1 Win=0 Len=0	4
26192	7235.8149050	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] railgun-webaccl > ssh [SYN] Seq=0 Win=512 Len=0	0
26193	7235.8152120	192.168.10.2	192.168.0.4	TCP	58	ssh > railgun-webaccl [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460	4
26194	7235.8154260	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [RST] Seq=1 Win=0 Len=0	4
26199	7236.8149540	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] railgun-webaccl > ssh [SYN] Seq=0 Win=512 Len=0	0
26200	7236.8151950	192.168.10.2	192.168.0.4	TCP	58	ssh > railgun-webaccl [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460	4
26201	7236.8154240	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [RST] Seq=1 Win=0 Len=0	4
26204	7237.8149930	192.168.0.4	192.168.10.2	TCP	60	[TCP Port numbers reused] railgun-webaccl > ssh [SYN] Seq=0 Win=512 Len=0	0
26205	7237.8153490	192.168.10.2	192.168.0.4	TCP	58	ssh > railgun-webaccl [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460	4
26206	7237.8155300	192.168.0.4	192.168.10.2	TCP	60	railgun-webaccl > ssh [RST] Seq=1 Win=0 Len=0	4

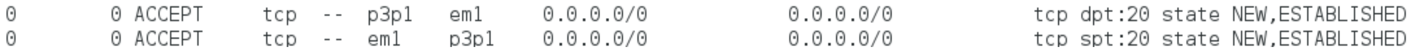
Test Case 15: Ensure FTP Data service has “Maximum Throughput”
(Requires a new DSCP column in wireshark. Value = ‘ip.dsfield.dscp’)

To test we sent FTP packets at port 20 with the following command:



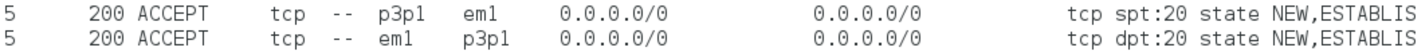
```
[root@DataComm ~]# hping3 192.168.10.2 -S -k -c 5 -p 20 --fast
HPING 192.168.10.2 (em1 192.168.10.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.10.2 ttl=63 DF id=4775 sport=20 flags=RA seq=0 win=0 rtt=1.1 ms
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=4793 sport=20 flags=RA seq=0 win=0 rtt=101.2 m
s
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=4799 sport=20 flags=RA seq=0 win=0 rtt=201.3 m
s
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=4888 sport=20 flags=RA seq=0 win=0 rtt=301.9 m
s
DUP! len=46 ip=192.168.10.2 ttl=63 DF id=4981 sport=20 flags=RA seq=0 win=0 rtt=402.0 m
s

--- 192.168.10.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.1/201.5/402.0 ms
```

IPTables before:



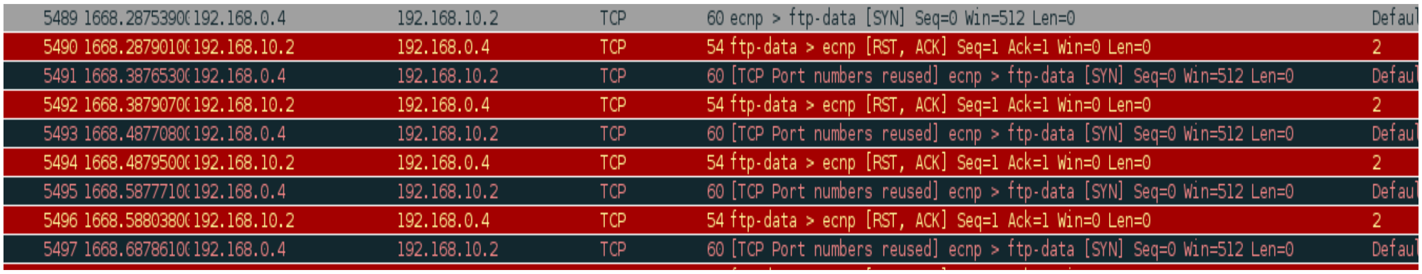
```
0        0 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp dpt:20 state NEW,ESTABLISHED
0        0 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp spt:20 state NEW,ESTABLISHED
```

IPtables after:



```
5      200 ACCEPT     tcp  --  p3p1   em1     0.0.0.0/0            0.0.0.0/0            tcp spt:20 state NEW,ESTABLIS
5      200 ACCEPT     tcp  --  em1    p3p1    0.0.0.0/0            0.0.0.0/0            tcp dpt:20 state NEW,ESTABLIS
```

Wireshark Capture of above. We see the DSCP column show a value of 2, which signifies maximum throughput.



```
5489 1668.2875390(192.168.0.4      192.168.10.2    TCP      60 ecnp > ftp-data [SYN] Seq=0 Win=512 Len=0                               Defaul
5490 1668.2879010(192.168.10.2     192.168.0.4     TCP      54 ftp-data > ecnp [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                      2
5491 1668.3876530(192.168.0.4      192.168.10.2    TCP      60 [TCP Port numbers reused] ecnp > ftp-data [SYN] Seq=0 Win=512 Len=0     Defaul
5492 1668.3879070(192.168.10.2     192.168.0.4     TCP      54 ftp-data > ecnp [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                      2
5493 1668.4877080(192.168.0.4      192.168.10.2    TCP      60 [TCP Port numbers reused] ecnp > ftp-data [SYN] Seq=0 Win=512 Len=0     Defaul
5494 1668.4879500(192.168.10.2     192.168.0.4     TCP      54 ftp-data > ecnp [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                      2
5495 1668.5877710(192.168.0.4      192.168.10.2    TCP      60 [TCP Port numbers reused] ecnp > ftp-data [SYN] Seq=0 Win=512 Len=0     Defaul
5496 1668.5880380(192.168.10.2     192.168.0.4     TCP      54 ftp-data > ecnp [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                      2
5497 1668.6878610(192.168.0.4      192.168.10.2    TCP      60 [TCP Port numbers reused] ecnp > ftp-data [SYN] Seq=0 Win=512 Len=0     Defaul
```