COMP 8006 Assignment 1

Network Security Administration 2

Linux Firewall and Packet Filter Report

By: Derek Wong (A01042588)

Instructor: Aman Abdulla

Due: 12 PM on February 4, 2021

Table of Contents

Objectives

Design, implement, and test a firewall for Linux that will implement the following rules:

- 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 hots from the outside.
- Do not accept any packets with a source address from the outside matching your internal network.
- You must ensure the firewall rejects those connections that are coming the “wrong” way (i.e., inbound SYN packets to high ports)
- 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.
- 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 implementation section will contain default and user-defined firewall rules. Create a set of default rules that will regulate network traffic to and from an internal machine.

- Permit inbound/outbound ssh packets.
- Permit inbound/outbound www packets.
- Drop inbound traffic to port 80 (http) from source ports less than 1024.
- Drop all incoming packets from reserved port 0 as well as outbound traffic to port 0.

Approach

The firewall/packet filter will be designed and implemented using Netfilter. The filter rules will be put together into a series of shell scripts to be executed. One file will contain a “User Configurable Section” which contain a set of macros defined. The rest of the script files will implement and configure the firewall and packet filter based on the macros specified in the config file.

To test, hping3 will be used to probe the firewall host with both permitted and unpermitted packets and log the response to text files. During these tests, wireshark will be used to analyze the inbound and outbound traffic between the external, firewall, and internal hosts on varying interfaces and ports. Wireshark captures and hping3 results will be compared to confirm the functionality of the firewall and whether it meets the requirements.

Firewall Design

The testbed will have one machine operating as a firewall, which will have one NIC configured to have public internet access through an access point or ethernet cable. A second machine will act as the internal host will be attached to the second NIC on the firewall which will forward datagrams to its internal hosts.

The firewall machine will be 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.

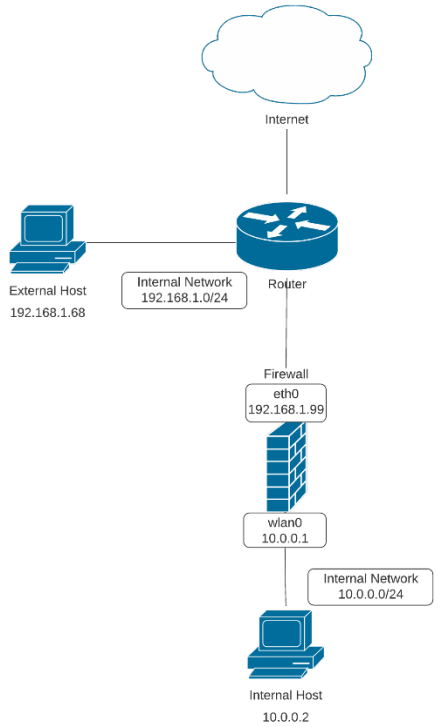


Figure 1: Network Architecture of test environment

```
[root@localhost Desktop]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.0.1        0.0.0.0         UG    0      0        0 enp0s3
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
```

Figure 2: IP routing table of internal host

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
10.0.0.0        10.0.0.1        255.255.255.0   UG    0      0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Figure 3: IP routing table of firewall host

Packet Filter Design

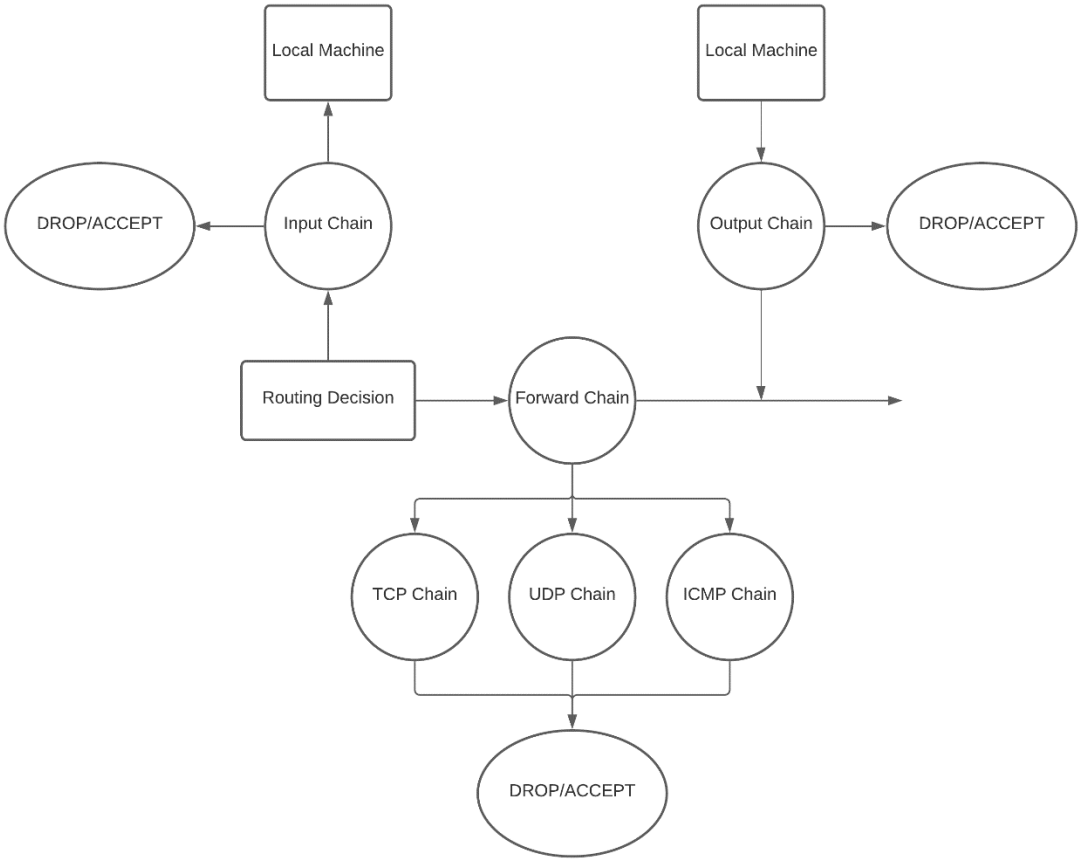


Figure 4: Packet Filtering design using iptables

Pseudocode

***MACROS are defined in the config file to be used in the script

*The order in which the rules and switches appear matter

* Three default chains are **INPUT, OUTPUT,** and **FORWARD** chains.

* Three user defined chains are created, **TCP**, **UDP**, and **ICMP**.

Flush all default/nat/mangle chains, delete all user defined chains

Set default policies, drop on default on INPUT, OUTPUT, FORWARD chains

Create new chains TCP, UDP, ICMP

Setup NAT Table

If ALLOW_FIREWALL_ SSH = true:

    Allow EXTERNAL_ADMIN_SSH_IP to ssh into firewall host

    Set up DNAT from INTERNET_IP to CLIENT_IP on INTERNET_NIC except when EXTERNAL_ADMIN_SSH_IP

Else

    Set up DNAT from INTERNET_IP to CLIENT_IP on INTERNET_NIC

Set up SNAT for outbound packets on INTERNET_NIC to INTERNET_IP

Setup Mangle Table

SSH TOS to minimize delay

FTP TOS to minimize delay and maximize throughput

SSH

If ALLOW_FIREWALL_SSH = true

    Accept INPUT SSH packets coming from EXTERNAL_ADMIN_SSH_IP to INTERNET_IP

    Accept OUTPUT SSH packets coming from INTERNET_IP to EXTERNAL_ADMIN_SSH_IP

Accept all SSH packets forwarded on firewall host

Drop all traffic to port 80 from source ports less than 1024 forwarded on firewall host

Drop all TCP/UDP packets from reserved port 0 and to reserved port 0 forwarded on firewall host

Block all connections coming from spoofed addresses matching internal network outside of firewall host

For ports in BLOCK_ALL_PORTS:

    Drop all inbound TCP/UDP traffic to internal host on port

Split traffic into TCP, UDP, ICMP user defined chains

TCP Chain

If INBOUND_TCP_BLOCK defined:

    Drop all TCP packets destined for internal host on ports in INBOUND_TCP_BLOCK

Accept only packets inbound to TCP_INBOUND_ALLOWED and packets coming from TCP_OUTBOUND_ALLOWED

Drop TCP packets failing to match with previous rules

UDP Chain

If INBOUND_UDP_BLOCK defined:

Drop all UDP packets destined for internal host on ports in INBOUND_UDP_BLOCK

Accept only packets inbound to UDP_INBOUND_ALLOWED and packets coming from UDP_OUTBOUND_ALLOWED

Drop UDP packets failing to match with previous rules

ICMP Chain

For type in ICMP_INBOUND_ALLOWED and ICMP_OUTBOUND_ALLOWED:

Accept ICMP type

Drop ICMP packets failing to match with previous rules

```
[root@node-1w7jr9qss7aicgdn0xlcgztng 8006]# iptables -L -n -v
Chain INPUT (policy DROP 2 packets, 582 bytes)
 pkts bytes target     prot opt in     out     source               destination
   10   656 ACCEPT     tcp  --  *      *       192.168.1.65         192.168.1.99        tcp dpt:22 state NEW,ESTABLISHED

Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:0
    0     0 DROP       all  --  *      *       10.0.0.0/24          10.0.0.2
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 23
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 23
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 137:139
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 137:139
    0     0 TCP        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 UDP        udp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ICMP       icmp --  *      *       0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    7   584 ACCEPT     tcp  --  *      *       192.168.1.99         192.168.1.65        tcp spt:22 state NEW,ESTABLISHED

Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            10.0.0.2            multiport dports 1024:65535
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20,21,80,443 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20,21,80,443 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 17,53 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 17,53 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain ICMP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 8 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Figure 5: Sample of iptables log on firewall host

How to Use Script

The programs are separated to four .sh file

- **setup.sh** (serve as program entry point for setting up environment, put up firewall and update firewall)
- **firewall.sh** (firewall rules implementation)
- **config.sh** (user defined parames for environment setup and firewall rules tweaking)
- **firewall-test.sh** (automatic test scripts for both internal and external testing)

Environment setup

Before setting up, make sure to change params in the "firewall host and internal host setup" section in config.sh to match your system params. And then run $ ./setup.sh

An option menu will show up, enter

"0" for firewall host setup (includes running ./firewall.sh)

"1" for internal host setup

"2" firewall update

"3" iptables listing

"4" firewall internal test

"5" firewall external test

"q" quit

Firewall setup

Choosing option **"0"** when running setup.sh will automatically put up a firewall. To tweak the firewall setting, modify the params in the "Firewall params" section in config.sh. And then run setup.sh and enter **"2"** for firewall update. A list of firewall params will show up in the console to indicate the update completion.

Firewall test

Choosing either option "4" or "5" will automatically run through predefined test cases in firewall-test.sh. The result will be logged to a text file, the filename will be whatever assigned for the macro $OUTPUT_FILE in config.sh

Firewall Macros examples

TCP_INBOUND_ALLOWED=”80,443:447”

TCP_OUTBOUND_ALLOWED=”80,443:447”

This example shows that the firewall allows inbound and outbound http and https tcp traffic

UDP_INBOUND_ALLOWED=”53,17”

UDP_OUTBOUND_ALLOWED=”53,17”

Firewall allows inbound and outbound DNS udp traffic

ICMP_INBOUND_ALLOWED=(”0” , ”8”)

ICMP_OUTBOUND_ALLOWED=(”0” , ”8”)

Firewall allows inbound icmp echo reply and outbound icmp echo request.

INBOUND_TCP_BLOCK =”1024:65535”

INBOUND_UDP_BLOCK =””

Firewall blocks inbound TCP traffic to high port numbers (incoming SYN)

BLOCK_ALL_PORTS =”23 137:139”

Firewall to block tcp/udp traffic direct to port 23 (telnet), and all traffic directed to port range 137 to 139

OUTPUT_FILE=”internal-test-results.txt”

Result of internal firewall-test will be direct to the file name internal-test-results.txt

Testing Design

Network Settings	Configuration
Internet Gateway IP	192.168.1.254
External Network	192.168.1.0/24
Internal Network	10.0.0.0/24
Firewall Host Internet IP	192.168.1.99
Firewall Host Private Network IP	10.0.0.1
Internal Host Private Network IP	10.0.0.2
External Host IPs	192.168.1.72 and 192.168.1.68

Internal Test (From internal host)

Case #	Test Description	Tool Used	Expected Result	Pass/Failed
1	Verify outbound TCP traffic from internal host can reach outside network and receive inbound TCP traffic from host in outside network on allowed ports	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
2	Verify outbound UDP traffic from internal host can reach outside network on allowed ports	hping3, wireshark	Firewall should accept packets and forward them from internal host to external host Packet capture on internal host should show sent packets to external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
3	Verify outbound ICMP traffic from internal host can reach outside network and receive inbound ICMP traffic from host in outside network on allowed type numbers	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
4	Verify all TCP, UDP, and ICMP packets generated from internal host that fall through to the default rule will be dropped	hping3, wireshark	Firewall should drop all non-matching TCP, UDP, ICMP packets	Pass. Details are attached below in supporting data.

5	Verify firewall accepts all TCP packets belonging to an existing connection on allowed ports (Only allow NEW and ESTABLISHED) traffic to go through the firewall	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
6	Verify firewall drops all TCP packets with SYN and FIN bit set	hping3, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
7	Verify firewall rejects all Telnet packets	telnet, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
8	Verify firewall permits inbound/outbound ssh packets	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
9	Verify firewall permits inbound/outbound www packets	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
10	Verify SSH traffic is being mangled to minimize delay	ssh client/server, wireshark	The *iptables* -L -v audit table should show that the traffic was mangled	Pass. Details are attached below in supporting data.
11	Verify FTP traffic is being mangled to minimize delay and maximize throughput	ftp, wireshark	The *iptables* -L -v audit table should show that the traffic was mangled	Pass. Details are attached below in supporting data.

External Test (Running from client machine outside internal network)

Case #	Test Description	Tool Used	Expected Result	Pass/Failed
1	Verify inbound UDP traffic from external host can reach internal network on allowed ports	hping3, wireshark	Firewall should accept packets and forward them to internal host from external host Packet capture on internal host should shows received packets from external host Packet capture on external should show sent packets to internal host	Pass. Details are attached below in supporting data.
2	Verify all TCP, UDP, and ICMP packets generated from external host that fall through to the default rule will be dropped	hping3, wireshark	Firewall should drop all non-matching TCP, UDP, ICMP packets	Pass. Details are attached below in supporting data.
3	Verify that all packets destined for the firewall host from outside are dropped	hping3, wireshark	Iptables listing on firewall host should show that the default policy for both INPUT and OUTPUT chain are set to DROP	Pass. Details are attached below in supporting data.
4	Verify firewall accepts all TCP packets belonging to an existing connection on allowed ports (Only allow NEW and ESTABLISHED) traffic to go through the firewall	hping3, wireshark	Firewall should accept packets and forward them to/from internal host to external host Packet capture on internal host should shows received packets from external host Packet capture on external should shows received packets from internal host	Pass. Details are attached below in supporting data.
5	Verify firewall rejects connections coming the “wrong” way (i.e., inbound SYN packets to high ports)	hping3, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
6	Verify that firewall rejects any packets with a source address from the outside matching internal network	hping3, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
7	Verify firewall drops all TCP packets with SYN and FIN bit set	hping3, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
8	Verify firewall rejects all Telnet packets	telnet, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.

9	Verify firewall drops inbound traffic to port 80 (http) from source ports less than 1024	hping3, wireshark	Firewall should drop the packets	Pass. Details are attached below in supporting data.
10	Verify firewall drops incoming packets from reserved port 0 as well as outbound traffic to port 0	hping3, wireshark	Firewall should drop the packet	Pass. Details are attached below in supporting data.

Confirmatory Data

Internal Test

Wireshark captures along with the automatic test results are in the “internal-test-data” directory.

Test Case 1

- The following test was run to send 5 TCP packets from the internal host on port 1024 to the external host on port 443

```
[root@localhost Desktop]# hping3 -c 5 -S --baseport 1024 --keep --destport 443 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.72 ttl=127 DF id=39938 sport=443 flags=SA seq=0 win=8192 rtt=237.7 ms
DUP! len=46 ip=192.168.1.72 ttl=127 DF id=39939 sport=443 flags=SA seq=0 win=8192 rtt=1016.2 ms
DUP! len=46 ip=192.168.1.72 ttl=127 DF id=39940 sport=443 flags=SA seq=0 win=8192 rtt=2012.4 ms
DUP! len=46 ip=192.168.1.72 ttl=127 DF id=39941 sport=443 flags=SA seq=0 win=8192 rtt=3014.6 ms
DUP! len=46 ip=192.168.1.72 ttl=127 DF id=39942 sport=443 flags=SA seq=0 win=8192 rtt=4015.9 ms

--- 192.168.1.72 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 237.7/2059.4/4015.9 ms
```

- Before the test, the iptables log shows:

```
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 80,443 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
  5   220 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 80,443 state NEW,ESTABLISHED
 10   400 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 state NEW,ESTABLISHED
```

- Notice that the packet count increased by 15, 5 packets are outgoing SYN packets, 5 incoming packets are SYN-ACK packets, and last 5 are RST packets are sent from the external host in response to receiving a packet on a closed socket.
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [SYN] Seq=0 Win=512 Len=0
2	0.006728	192.168.1.72	10.0.0.2	TCP	58	443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
3	0.009672	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [RST] Seq=1 Win=0 Len=0
4	0.773915	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 1024 → 443 [SYN] Seq=0 Win=512 Len=0
5	0.783761	192.168.1.72	10.0.0.2	TCP	58	443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
6	0.786564	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [RST] Seq=1 Win=0 Len=0
7	1.772635	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 1024 → 443 [SYN] Seq=0 Win=512 Len=0
8	1.781131	192.168.1.72	10.0.0.2	TCP	58	443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
9	1.783039	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [RST] Seq=1 Win=0 Len=0
10	2.773217	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 1024 → 443 [SYN] Seq=0 Win=512 Len=0
11	2.782976	192.168.1.72	10.0.0.2	TCP	58	443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
12	2.785125	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [RST] Seq=1 Win=0 Len=0
13	3.775727	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 1024 → 443 [SYN] Seq=0 Win=512 Len=0
14	3.784524	192.168.1.72	10.0.0.2	TCP	58	443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
15	3.786411	10.0.0.2	192.168.1.72	TCP	60	1024 → 443 [RST] Seq=1 Win=0 Len=0

Test Case 2

- The following test was run to send 5 UDP packets from the internal host on port 17 to the external host on port 17

```
[root@localhost Desktop]# hping --fast --udp -c 5 --baseport 17 --keep --destport 17 192.168.1.68
HPING 192.168.1.68 (enp0s3 192.168.1.68): udp mode set, 28 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
0     0 ACCEPT     udp  --  *      *       10.0.0.2            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
5   140 ACCEPT     udp  --  *      *       10.0.0.2            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
```

- Notice that the packet count increased by 5, confirming that all 5 UDP packets were successfully forwarded from the internal host to the external host
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	UDP	60	17 → 17 Len=0
2	0.850159	10.0.0.2	192.168.1.72	UDP	60	17 → 17 Len=0
3	1.853543	10.0.0.2	192.168.1.72	UDP	60	17 → 17 Len=0
4	2.849770	10.0.0.2	192.168.1.72	UDP	60	17 → 17 Len=0
5	3.849638	10.0.0.2	192.168.1.72	UDP	60	17 → 17 Len=0

Test Case 3

- The following test was run to send 5 ICMP packets from the internal host to the external host

```
[root@localhost Desktop]# hping3 --icmp -c 5 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.1.72 ttl=127 id=49727 icmp_seq=0 rtt=297.1 ms
len=46 ip=192.168.1.72 ttl=127 id=49728 icmp_seq=1 rtt=9.0 ms
len=46 ip=192.168.1.72 ttl=127 id=49729 icmp_seq=2 rtt=10.8 ms
len=46 ip=192.168.1.72 ttl=127 id=49730 icmp_seq=3 rtt=15.6 ms
len=46 ip=192.168.1.72 ttl=127 id=49731 icmp_seq=4 rtt=10.8 ms

--- 192.168.1.72 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 9.0/68.6/297.1 ms
```

- Before the test, the iptables log shows:

```
0     0 ACCEPT     icmp --  *      *       0.0.0.0/0           10.0.0.2            icmptype 0 state NEW,ESTABLISHED
0     0 ACCEPT     icmp --  *      *       0.0.0.0/0           10.0.0.2            icmptype 8 state NEW,ESTABLISHED
0     0 ACCEPT     icmp --  *      *       10.0.0.2            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
0     0 ACCEPT     icmp --  *      *       10.0.0.2            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
5   140 ACCEPT     icmp --  *      *       0.0.0.0/0           10.0.0.2            icmptype 0 state NEW,ESTABLISHED
0     0 ACCEPT     icmp --  *      *       0.0.0.0/0           10.0.0.2            icmptype 8 state NEW,ESTABLISHED
0     0 ACCEPT     icmp --  *      *       10.0.0.2            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
5   140 ACCEPT     icmp --  *      *       10.0.0.2            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
```

- Notice that the packet count increased by a total of 10 coming from the internal host and the external host, confirming that all 5 ICMP packets were successfully forwarded both ways
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	ICMP	60	Echo (ping) request id=0xd924, seq=0/0, ttl=64 (reply in 2)
2	0.006395	192.168.1.72	10.0.0.2	ICMP	42	Echo (ping) reply id=0xd924, seq=0/0, ttl=127 (request in 1)
3	0.713396	10.0.0.2	192.168.1.72	ICMP	60	Echo (ping) request id=0xd924, seq=256/1, ttl=64 (reply in 4)
4	0.720067	192.168.1.72	10.0.0.2	ICMP	42	Echo (ping) reply id=0xd924, seq=256/1, ttl=127 (request in 3)
5	1.713208	10.0.0.2	192.168.1.72	ICMP	60	Echo (ping) request id=0xd924, seq=512/2, ttl=64 (reply in 6)
6	1.720813	192.168.1.72	10.0.0.2	ICMP	42	Echo (ping) reply id=0xd924, seq=512/2, ttl=127 (request in 5)
7	2.714266	10.0.0.2	192.168.1.72	ICMP	60	Echo (ping) request id=0xd924, seq=768/3, ttl=64 (reply in 8)
8	2.722292	192.168.1.72	10.0.0.2	ICMP	42	Echo (ping) reply id=0xd924, seq=768/3, ttl=127 (request in 7)
9	3.713951	10.0.0.2	192.168.1.72	ICMP	60	Echo (ping) request id=0xd924, seq=1024/4, ttl=64 (reply in 10)
10	3.720839	192.168.1.72	10.0.0.2	ICMP	42	Echo (ping) reply id=0xd924, seq=1024/4, ttl=127 (request in 9)

Test Case 4

- The following test includes 3 sub-tests to send 3 different types of traffic: TCP, UDP, ICMP packets from the internal host to the external host

```
[root@localhost Desktop]# hping -c 5 -S --baseport 100 --destport 18 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): S set, 40 headers + 0 data bytes

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

```
[root@localhost Desktop]# hping3 --fast -c 5 --udp --baseport 100 --destport 18 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): udp mode set, 28 headers + 0 data bytes

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

```
[root@localhost Desktop]# hping3 --fast --icmp-ts -c 5 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): icmp mode set, 28 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.0.2            multiport dports 20:21,21000:21010 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       10.0.0.2             0.0.0.0/0           multiport sports 20:21,21000:21010 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            10.0.0.2            multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       10.0.0.2             0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain ICMP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    5   140 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 8 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    5   140 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.0.2            multiport dports 20:21,21000:21010 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       10.0.0.2             0.0.0.0/0           multiport sports 20:21,21000:21010 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            10.0.0.2            multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       10.0.0.2             0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    5   140 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain ICMP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    5   140 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            10.0.0.2            icmptype 8 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    5   140 ACCEPT     icmp --  *      *       10.0.0.2             0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice that for each user-defined chain TCP, UDP, ICMP, all 5 packets are dropped
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	TCP	60	100 → 18 [SYN] Seq=0 Win=512 Len=0
2	0.961773	10.0.0.2	192.168.1.72	TCP	60	101 → 18 [SYN] Seq=0 Win=512 Len=0
3	1.961904	10.0.0.2	192.168.1.72	TCP	60	102 → 18 [SYN] Seq=0 Win=512 Len=0
4	2.962864	10.0.0.2	192.168.1.72	TCP	60	103 → 18 [SYN] Seq=0 Win=512 Len=0
5	3.964727	10.0.0.2	192.168.1.72	TCP	60	104 → 18 [SYN] Seq=0 Win=512 Len=0

Test Case 5

- Test Case 1 already showed a TCP exchange of a new connection (SYN packet) and established connection (where RST packets are sent alongside SYN-ACK packets). Here we will send a SYN-ACK packet from the internal host to begin.

```
[root@localhost Desktop]# hping3 --fast -c 5 -S -A --baseport 1001 --destport 1001 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): SA set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice all 5 TCP packets set with SYN-ACK are dropped; In test case 1, SYN-ACKs aren't dropped when they are sent in response to a SYN packet and neither are RST packets
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	TCP	60	1001 → 1001 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
2	0.000260	10.0.0.2	192.168.1.72	TCP	60	1002 → 1001 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
3	0.000412	10.0.0.2	192.168.1.72	TCP	60	1003 → 1001 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
4	0.032165	10.0.0.2	192.168.1.72	TCP	60	1004 → 1001 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
5	0.129976	10.0.0.2	192.168.1.72	TCP	60	1005 → 1001 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0

Test Case 6

- The following test was run to send 5 TCP packets with SYN and FIN bits set together from the internal host to the external host on port 1001

```
[root@localhost Desktop]# hping3 --fast -c 5 -S -F --baseport 1001 --destport 1001 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): SF set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice all 5 TCP packets set with SYN-FIN are dropped
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	TCP	60	1001 → 1001 [FIN, SYN] Seq=0 Win=512 Len=0
2	0.100926	10.0.0.2	192.168.1.72	TCP	60	1002 → 1001 [FIN, SYN] Seq=0 Win=512 Len=0
3	0.204916	10.0.0.2	192.168.1.72	TCP	60	1003 → 1001 [FIN, SYN] Seq=0 Win=512 Len=0
4	0.300921	10.0.0.2	192.168.1.72	TCP	60	1004 → 1001 [FIN, SYN] Seq=0 Win=512 Len=0
5	0.401028	10.0.0.2	192.168.1.72	TCP	60	1005 → 1001 [FIN, SYN] Seq=0 Win=512 Len=0

Test Case 7

- The following test was run to send 5 TCP packets from the internal host on port 23 to the external host on port 23

```
[root@localhost Desktop]# hping3 --fast -c 5 -S --baseport 23 --keep --destport 23 192.168.1.72
HPING 192.168.1.72 (enp0s3 192.168.1.72): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    5   200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice all 5 TCP Telnet packets are dropped early confirming our working rule
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.72	TCP	60	23 → 23 [SYN] Seq=0 Win=512 Len=0
2	0.084486	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
3	0.184492	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
4	0.284562	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
5	0.384723	10.0.0.2	192.168.1.72	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0

Test Case 8

- The following test was run to send 5 TCP packets from the internal host on port 22 to the external host on port 22

```
[root@localhost Desktop]# hping3 192.168.1.68 -S -s 22 --keep -p 22 -c 5
HPING 192.168.1.68 (enp0s3 192.168.1.68): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=155.6 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=1014.9 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=2024.7 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=3015.0 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=4010.2 ms

--- 192.168.1.68 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 155.6/2044.1/4010.2 ms
```

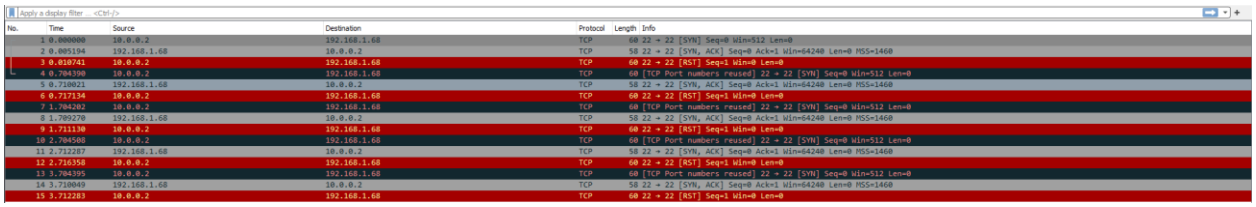
- Before the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   15   620 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
```

- Notice all 15 TCP SSH packets are accepted early confirming our working rule
- Wireshark capture on firewall host



Test Case 9

- The following test was run to send 5 TCP packets from the internal host on port 1024 to the external host on port 80

```
[root@localhost Desktop]# hping3 --fast -c 5 -S --baseport 1025 --keep --destport 80 192.168.1.68
HPING 192.168.1.68 (enp0s3 192.168.1.68): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=96.7 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=110.4 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=300.8 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=310.6 ms
DUP! len=46 ip=192.168.1.68 ttl=63 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=496.1 ms

--- 192.168.1.68 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 96.7/262.9/496.1 ms
```

- Before the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 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 dpt:53 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 80,443 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 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 dpt:53 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
    5   200 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 80,443 state NEW,ESTABLISHED
    5   200 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 80,443 state NEW,ESTABLISHED
```

- Notice all 10 TCP HTTP packets are going outbound and inbound confirming our working rule. Since nothing is running on port 80 on the external host, we are returned a RST and ACK
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	192.168.1.68	TCP	60	1025 → 80 [SYN] Seq=0 Win=512 Len=0
2	0.009243	192.168.1.68	10.0.0.2	TCP	54	80 → 1025 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	0.021383	10.0.0.2	192.168.1.68	TCP	60	[TCP Port numbers reused] 1025 → 80 [SYN] Seq=0 Win=512 Len=0
4	0.026252	192.168.1.68	10.0.0.2	TCP	54	80 → 1025 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5	0.211699	10.0.0.2	192.168.1.68	TCP	60	[TCP Port numbers reused] 1025 → 80 [SYN] Seq=0 Win=512 Len=0
6	0.216254	192.168.1.68	10.0.0.2	TCP	54	80 → 1025 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	0.220918	10.0.0.2	192.168.1.68	TCP	60	[TCP Port numbers reused] 1025 → 80 [SYN] Seq=0 Win=512 Len=0
8	0.225543	192.168.1.68	10.0.0.2	TCP	54	80 → 1025 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	0.379772	10.0.0.2	192.168.1.68	TCP	60	[TCP Port numbers reused] 1025 → 80 [SYN] Seq=0 Win=512 Len=0
10	0.412585	192.168.1.68	10.0.0.2	TCP	54	80 → 1025 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Test Case 10

- For this test an external client was used to log on to the internal server running the ssh service through the firewall
- Before running the test with an SSH client, iptables on the firewall reported:

```
[root@node-1w7jr9qss7aicgdn0xlcgztng 8006]# iptables -L -t mangle -v -Z
Chain PREROUTING (policy ACCEPT 40 packets, 9306 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ssh  TOS setMinimize-Delay
```

- After running the test, iptables reported:

```
[root@node-1w7jr9qss7aicgdn0xlcgztng 8006]# iptables -L -t mangle -v
Chain PREROUTING (policy ACCEPT 574 packets, 132K bytes)
 pkts bytes target     prot opt in     out     source               destination
   14  2445 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ssh  TOS setMinimize-Delay
```

- This above confirms that the iptables successfully mangled the SSH traffic

Test Case 11

- For this test an external client was used to log on to the internal server running the ftp service through the firewall
- Before running the test with an ftp client, iptables on the firewall reported:

```
Chain PREROUTING (policy ACCEPT 10 packets, 1281 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ssh  TOS setMinimize-Delay
    0     0 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ftp  TOS setMinimize-Delay
    0     0 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ftp-data  TOS setMaximize-Throughput
```

- After running the test, iptables reported:

```
[root@node-1w7jr9qss7aicgdn0xlcgztng 8006]# iptables -L -t mangle -v
Chain PREROUTING (policy ACCEPT 264 packets, 50909 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ssh  TOS setMinimize-Delay
   28  1959 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ftp  TOS setMinimize-Delay
   25  6524 TOS        tcp  --  any    any     anywhere             anywhere            tcp spt:ftp-data  TOS setMaximize-Throughput
```

- This above confirms that the iptables successfully mangled the FTP traffic

External Test

Test Case 1

- The following test was run to send 5 UDP packets from the internal host on port 17 to the external host on port 17

```
[root@localhost Desktop]# hping3 --fast --udp -c 5 --baseport 17 --keep --destport 17 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): udp mode set, 28 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    5   140 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice that the packet count increased by 5, confirming that all 5 UDP packets were successfully forwarded from the external host to the internal host
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	10.0.0.2	UDP	42	17 → 17 Len=0
2	0.102006	192.168.1.68	10.0.0.2	UDP	42	17 → 17 Len=0
3	0.199806	192.168.1.68	10.0.0.2	UDP	42	17 → 17 Len=0
4	0.300542	192.168.1.68	10.0.0.2	UDP	42	17 → 17 Len=0
5	0.401279	192.168.1.68	10.0.0.2	UDP	42	17 → 17 Len=0

Test Case 2

- The following test includes 3 sub-tests to send 3 different types of traffic: TCP, UDP, ICMP packets from the external host to the internal host

```
[root@localhost Desktop]# hping3 --fast -c 5 -S --baseport 100 --destport 18 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

```
[root@localhost Desktop]# hping3 --udp --fast -c 5 -S --baseport 100 --destport 18 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): udp mode set, 28 headers + 0 data bytes

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

```
[root@localhost Desktop]# hping3 --fast --icmp-ts -c 5 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): icmp mode set, 28 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain ICMP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

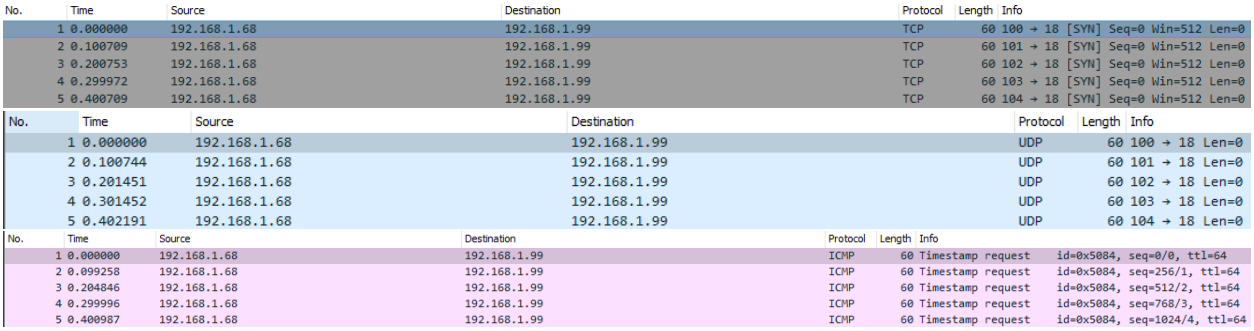
```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 17 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 17 state NEW,ESTABLISHED
    5   140 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain ICMP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 0 state NEW,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmptype 8 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice that for each user-defined chain TCP, UDP, ICMP, all 5 packets are dropped

- Wireshark capture on firewall host



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	100 → 18 [SYN] Seq=0 Win=512 Len=0
2	0.100709	192.168.1.68	192.168.1.99	TCP	60	101 → 18 [SYN] Seq=0 Win=512 Len=0
3	0.200753	192.168.1.68	192.168.1.99	TCP	60	102 → 18 [SYN] Seq=0 Win=512 Len=0
4	0.299972	192.168.1.68	192.168.1.99	TCP	60	103 → 18 [SYN] Seq=0 Win=512 Len=0
5	0.400709	192.168.1.68	192.168.1.99	TCP	60	104 → 18 [SYN] Seq=0 Win=512 Len=0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	UDP	60	100 → 18 Len=0
2	0.100744	192.168.1.68	192.168.1.99	UDP	60	101 → 18 Len=0
3	0.201451	192.168.1.68	192.168.1.99	UDP	60	102 → 18 Len=0
4	0.301452	192.168.1.68	192.168.1.99	UDP	60	103 → 18 Len=0
5	0.402191	192.168.1.68	192.168.1.99	UDP	60	104 → 18 Len=0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	ICMP	60	Timestamp request id=0x5084, seq=0/0, ttl=64
2	0.099258	192.168.1.68	192.168.1.99	ICMP	60	Timestamp request id=0x5084, seq=256/1, ttl=64
3	0.204846	192.168.1.68	192.168.1.99	ICMP	60	Timestamp request id=0x5084, seq=512/2, ttl=64
4	0.299996	192.168.1.68	192.168.1.99	ICMP	60	Timestamp request id=0x5084, seq=768/3, ttl=64
5	0.400987	192.168.1.68	192.168.1.99	ICMP	60	Timestamp request id=0x5084, seq=1024/4, ttl=64

Test Case 3

- All packets destined for the firewall host will be pre-routed to the internal host via DNAT. By default the input and output chain are set to DROP.
- Iptable log of INPUT chain

```
Chain INPUT (policy DROP 2 packets, 582 bytes)
 pkts bytes target     prot opt in     out     source               destination
    8   448 ACCEPT     tcp  --  *      *       192.168.1.65         192.168.1.99         tcp dpt:22 state NEW,ESTABLISHED
```

- Iptable log of OUTPUT chain

```
Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    3   376 ACCEPT     tcp  --  *      *       192.168.1.99         192.168.1.65         tcp spt:22 state NEW,ESTABLISHED
```

- Iptable log of NAT table

```
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  416 29212 DNAT       all  --  eth0   *      !192.168.1.65         192.168.1.99         to:10.0.0.2
    0     0 DNAT       all  --  eth0   *       0.0.0.0/0            10.0.0.2             to:192.168.1.99

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  194 10550 SNAT       all  --  *      eth0    0.0.0.0/0            0.0.0.0/0            to:192.168.1.99
```

Test Case 4

- The following test was run to send 5 TCP packets from the external host on port 22 to the internal host on port 22, each time incrementing the baseport by one.

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --baseport 22 --destport 22 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.99 ttl=63 DF id=0 sport=22 flags=SA seq=1 win=64240 rtt=9.0 ms
len=46 ip=192.168.1.99 ttl=63 DF id=0 sport=22 flags=SA seq=2 win=64240 rtt=9.7 ms
len=46 ip=192.168.1.99 ttl=63 DF id=0 sport=22 flags=SA seq=3 win=64240 rtt=9.7 ms
len=46 ip=192.168.1.99 ttl=63 DF id=0 sport=22 flags=SA seq=4 win=64240 rtt=7.8 ms
len=46 ip=192.168.1.99 ttl=63 DF id=0 sport=22 flags=SA seq=0 win=64240 rtt=1060.6 ms

--- 192.168.1.99 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 7.8/219.4/1060.6 ms
```

- Before the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
```

- Following the test, the iptables log shows:

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   11   444 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22 state NEW,ESTABLISHED
    4   176 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
```

- Notice that the packet count increased by 15, 5 packets are incoming SYN packets, 5 incoming packets are SYN-ACK packets, and last 5 are RST packets are sent from the external host in response to receiving a packet on a closed socket. Since the baseport increases by one each time from a permitted port to a non-explicitly permitted port, it should be rejected under normal circumstances. However, we see that they are accepted by the firewall since it is stateful.
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	22 → 22 [SYN] Seq=0 Win=512 Len=0
2	0.100746	192.168.1.68	192.168.1.99	TCP	60	23 → 22 [SYN] Seq=0 Win=512 Len=0
3	0.103217	192.168.1.99	192.168.1.68	TCP	58	22 → 23 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
4	0.108525	192.168.1.68	192.168.1.99	TCP	60	23 → 22 [RST] Seq=1 Win=0 Len=0
5	0.202235	192.168.1.68	192.168.1.99	TCP	60	24 → 22 [SYN] Seq=0 Win=512 Len=0
6	0.204829	192.168.1.99	192.168.1.68	TCP	58	22 → 24 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
7	0.209998	192.168.1.68	192.168.1.99	TCP	60	24 → 22 [RST] Seq=1 Win=0 Len=0
8	0.302277	192.168.1.68	192.168.1.99	TCP	60	25 → 22 [SYN] Seq=0 Win=512 Len=0
9	0.305081	192.168.1.99	192.168.1.68	TCP	58	22 → 25 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
10	0.310742	192.168.1.68	192.168.1.99	TCP	60	25 → 22 [RST] Seq=1 Win=0 Len=0
11	0.402234	192.168.1.68	192.168.1.99	TCP	60	26 → 22 [SYN] Seq=0 Win=512 Len=0
12	0.404544	192.168.1.99	192.168.1.68	TCP	58	22 → 26 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
13	0.408526	192.168.1.68	192.168.1.99	TCP	60	26 → 22 [RST] Seq=1 Win=0 Len=0
14	1.054842	192.168.1.99	192.168.1.68	TCP	58	22 → 22 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
15	1.060006	192.168.1.68	192.168.1.99	TCP	60	22 → 22 [RST] Seq=1 Win=0 Len=0

Test Case 5

- The following test was run to send 5 TCP and 5 UDP packets from the external host with high destination ports.

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --baseport 20 --keep --destport 60000 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
    0     0 DROP       udp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
```

- Following the test, the iptables log shows:

```
    5   200 DROP       tcp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
    0     0 DROP       udp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
```

```
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
    5   140 DROP       udp  --  *      *       0.0.0.0/0            10.0.0.2             multiport dports 1024:65535
```

- A total of 10 packets show up in the forward chain that go to drop which confirms that our all packets were not forwarded to high ports
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	20 → 60000 [SYN] Seq=0 Win=512 Len=0
2	0.100730	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 20 → 60000 [SYN] Seq=0 Win=512 Len=0
3	0.201829	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 20 → 60000 [SYN] Seq=0 Win=512 Len=0
4	0.301830	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 20 → 60000 [SYN] Seq=0 Win=512 Len=0
5	0.409275	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 20 → 60000 [SYN] Seq=0 Win=512 Len=0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	UDP	60	20 → 60000 Len=0
2	0.098453	192.168.1.68	192.168.1.99	UDP	60	20 → 60000 Len=0
3	0.198455	192.168.1.68	192.168.1.99	UDP	60	20 → 60000 Len=0
4	0.298452	192.168.1.68	192.168.1.99	UDP	60	20 → 60000 Len=0
5	0.399257	192.168.1.68	192.168.1.99	UDP	60	20 → 60000 Len=0

Test Case 6

- The following test was run to send 5 TCP packets, with a spoofed address of 10.0.0.3 from external host to the internal host on port 443

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --spoof 10.0.0.3 --baseport 443 --keep --destport 443 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
    0     0 DROP       all  --  *      *       10.0.0.0/24          10.0.0.2
```

- Following the test, the iptables log shows:

```
    5   200 DROP       all  --  *      *       10.0.0.0/24          10.0.0.2
```

- A total of 5 packets show up in the forward chain that go to drop which confirms that our all spoofed packets were not forwarded to the internal network
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.3	192.168.1.99	TCP	60	443 → 443 [SYN] Seq=0 Win=512 Len=0
2	0.095156	10.0.0.3	192.168.1.99	TCP	60	[TCP Port numbers reused] 443 → 443 [SYN] Seq=0 Win=512 Len=0
3	0.195160	10.0.0.3	192.168.1.99	TCP	60	[TCP Port numbers reused] 443 → 443 [SYN] Seq=0 Win=512 Len=0
4	0.295890	10.0.0.3	192.168.1.99	TCP	60	[TCP Port numbers reused] 443 → 443 [SYN] Seq=0 Win=512 Len=0
5	0.396618	10.0.0.3	192.168.1.99	TCP	60	[TCP Port numbers reused] 443 → 443 [SYN] Seq=0 Win=512 Len=0

Test Case 7

- The following test was run to send 5 TCP packets with SYN and FIN bits set, from external host to the internal host on ports 443

```
[root@localhost osboxes]# hping3 --fast -c 5 -S -F --baseport 443 --destport 443 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): SF set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,80,443 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,80,443 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    5   200 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- A total of 5 packets show up in the TCP Forward chain that go to drop which confirms that our all packets were not forwarded to the internal network
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	443 → 443 [FIN, SYN] Seq=0 Win=512 Len=0
2	0.099698	192.168.1.68	192.168.1.99	TCP	60	444 → 443 [FIN, SYN] Seq=0 Win=512 Len=0
3	0.200438	192.168.1.68	192.168.1.99	TCP	60	445 → 443 [FIN, SYN] Seq=0 Win=512 Len=0
4	0.302184	192.168.1.68	192.168.1.99	TCP	60	446 → 443 [FIN, SYN] Seq=0 Win=512 Len=0
5	0.401170	192.168.1.68	192.168.1.99	TCP	60	447 → 443 [FIN, SYN] Seq=0 Win=512 Len=0

Test Case 8

- The following test was run to send 5 TCP packets from the external host on port 23 to the internal host on port 23

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --baseport 23 --keep --destport 23 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20:21,10011:10101 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Following the test, the iptables log shows:

```
Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    5   200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 0,23,111,515
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports 20,21,80,443 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport sports 20,21,80,443 state NEW,ESTABLISHED
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

- Notice all 5 TCP Telnet packets are dropped early confirming our working rule
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	23 → 23 [SYN] Seq=0 Win=512 Len=0
2	0.104747	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
3	0.200642	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
4	0.301440	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0
5	0.402485	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 23 → 23 [SYN] Seq=0 Win=512 Len=0

Test Case 9

- The following test was run to send 5 TCP packets from the external host on port 70 (Ports below 1024) to the internal host on port 80

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --baseport 70 --keep --destport 80 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
```

- Following the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    5   200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
```

- Notice all 5 TCP incoming packets are dropped early confirming our working rule
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	70 → 80 [SYN] Seq=0 Win=512 Len=0
2	0.099982	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 70 → 80 [SYN] Seq=0 Win=512 Len=0
3	0.201443	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 70 → 80 [SYN] Seq=0 Win=512 Len=0
4	0.302917	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 70 → 80 [SYN] Seq=0 Win=512 Len=0
5	0.405872	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 70 → 80 [SYN] Seq=0 Win=512 Len=0

Test Case 10

- The following test was run to send 5 TCP packets from the external host on port 80 (Reserved port) to the internal host on port 0

```
[root@localhost osboxes]# hping3 --fast -c 5 -S --baseport 70 --keep --destport 80 192.168.1.99
HPING 192.168.1.99 (enp0s3 192.168.1.99): S set, 40 headers + 0 data bytes

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

- Before the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:0
```

- Following the test, the iptables log shows:

```
Chain FORWARD (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 dpt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    5   200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:0
```

- Notice all 5 TCP incoming packets are dropped early confirming our working rule
- Wireshark capture on firewall host

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.68	192.168.1.99	TCP	60	80 → 0 [SYN] Seq=0 Win=512 Len=0
2	0.097782	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 80 → 0 [SYN] Seq=0 Win=512 Len=0
3	0.198516	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 80 → 0 [SYN] Seq=0 Win=512 Len=0
4	0.298518	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 80 → 0 [SYN] Seq=0 Win=512 Len=0
5	0.399508	192.168.1.68	192.168.1.99	TCP	60	[TCP Port numbers reused] 80 → 0 [SYN] Seq=0 Win=512 Len=0