8006 Assignment 1

Design & Documentation
Eric Wu, A00961904
BTech 2020 Winter

Table of contents

Objectives

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

- Set the default policies to **DROP**
- Create a set of rules that will:
    - 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 traffic from reserved port 0 as well as outbound traffic to port 0
- Drop inbound SYN packets, unless there is a rule that permits inbound traffic
- Create a set of user **user-defined** chains that will implement **accounting rules** to keep track of www, ssh traffic, versus the rest of the traffic on system.

Approach

The actual implementation is done using **Netfilter**. The testing are done using **hping2**, **Nmap** and **wireshark** (captures).

Firewall Design

Dour user defined chain are created - **ALL**, **WWW**, **SSH** and **OTHERS**
Below figure shows an overview of the design. The shaded circles are the user defined chains.

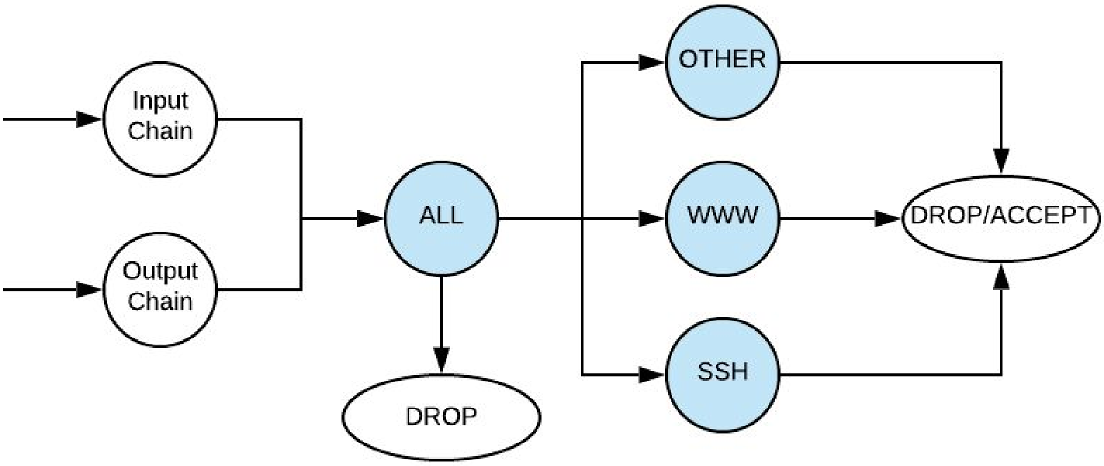


Figure1: IP Tables

(Rules implementation in detail, order matters)

INPUT

- Direct to **ALL,** drop on default

OUTPUT

- Direct to **ALL,** drop on default

FORWARD

- Drop on default

ALL

- Drop all incoming traffic from reserved port 0 as well as outbound traffic to port 0
- If incoming traffic is tcp 80, direct to **WWW**
- If incoming traffic is tcp 22, direct to **SSH**
- The rest of the traffic goes to **OTHERS**

WWW

- Drop inbound traffic to port 80 (http) from source ports less than 1024
- Accept on default

SSH

- Accept on default

OTHERS

- If traffic is DNS (port 53) or DHCP (67:68) with valid states, accept it
- Drop on default

How to use

Run the script.sh file to update the firewall tables

```
$ ./script.sh
```

Run the list.sh file to list the the default chains and user defined chains with packet count

```
$ ./list.sh
```

Test cases & Results

(*support data for each of the test can be found at **Support evidence** section)
Local machine 1: 192.168.0.41
Local machine 2: 192.168.0.44

Rule #	Test Description	Tool Used	Expected Result	Pass/ Failed
1	Verify that inbound ssh packets can get through	hping3, wireshark	● Local machine 1 should capture 5 incoming packets send from Local machine 2. ● iptable table listing on Local machine 1 should show that 10 packets hits the **SSH** chain and accepted	**Pass**
2	Verify that outbound ssh packets can get through	hping3, wireshark	● Local machine 2 should capture 5 outgoing packets send from Local machine 1. ● iptable table listing on Local machine 2 should show that 10 packets hits the **SSH** chain and accepted	**Pass**
3	Verify that inbound www packets can get through	hping3, wireshark	● Local machine 1 should capture 5 incoming packets send from Local machine 2. ● iptable table listing on Local machine 1 should show that 10 packets hits the **WWW** chain and accepted	**Pass**
4	Verify that outbound www packets can get through	hping3, wireshark	● Local machine 2 should capture 5 outgoing packets send from Local machine 1. ● iptable table listing on Local machine 1 should show that 10 packets hits the **WWW** chain	**Pass**

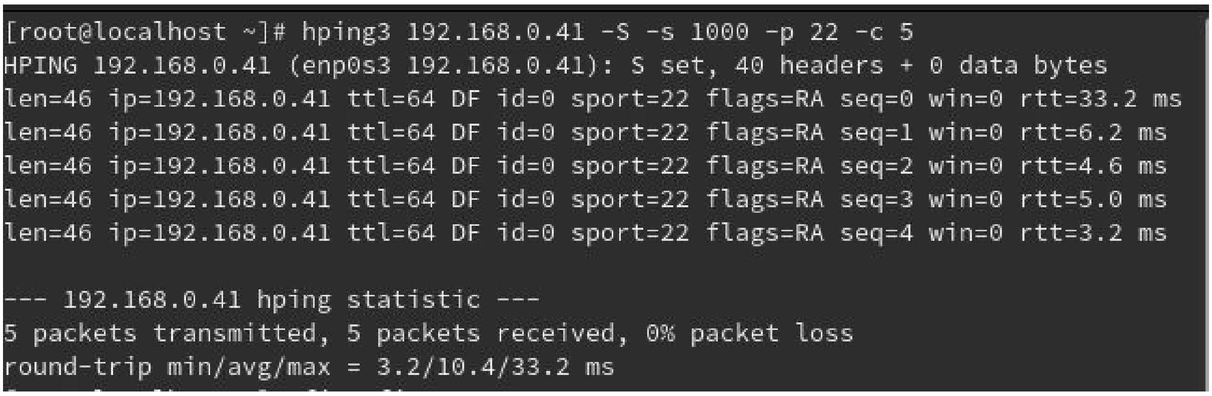
			and accepted	
5	Dorp inbound traffic to port 80 (http) from source port less than 1024	hping3, wireshark	- Local machine 1 should capture 5 incoming packets send from Local machine 2. - iptable table listing on Local machine 1 should show that 5 packets hits the **WWW** chain and dropped	**Pass**
6	Drop inbound traffic from reserved port 0	hping3, wireshark	- Local machine 1 should capture 5 incoming packets send from Local machine 2. - iptable table listing on Local machine 1 should show that 5 packets hits the **ALL** chain and dropped	**Pass**
7	Drop outbound traffic from reserved port 0	hping3, wireshark	- iptable table listing on Local machine 1 should show that 5 packets hits the **ALL** chain and dropped - No packets that goes to Local machine 2 should show up on wireshark capture	**Pass**

Support evidence

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

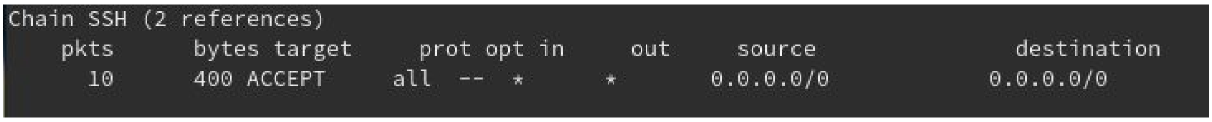
Detail: Machine 2 probes Machine 1 with ssh packets five times and got response. So a total of 10 packets should should up in the **SSH** chain that goes to **ACCEPT** (from both input and output chain). The wireshark captures shows that 5 incoming traffic (port 1000 to port 22) was received from the network.

Case# 1 192.168.0.44 -> 192.168.0.41
(hping3 output on 192.168.0.44)



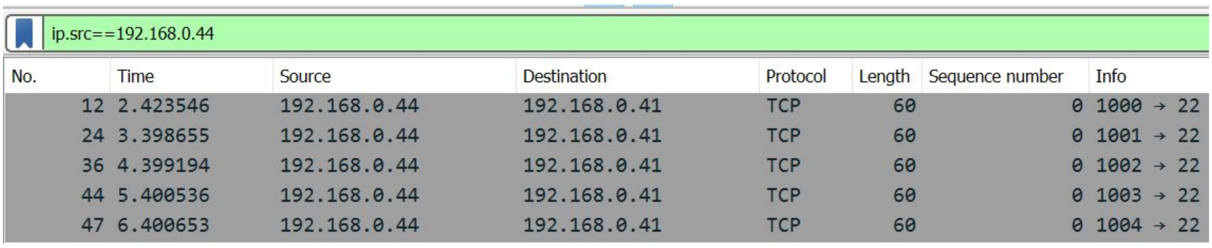
```
[root@localhost ~]# hping3 192.168.0.41 -S -s 1000 -p 22 -c 5
HPING 192.168.0.41 (enp0s3 192.168.0.41): S set, 40 headers + 0 data bytes
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=33.2 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=RA seq=1 win=0 rtt=6.2 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=RA seq=2 win=0 rtt=4.6 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=RA seq=3 win=0 rtt=5.0 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=22 flags=RA seq=4 win=0 rtt=3.2 ms

--- 192.168.0.41 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.2/10.4/33.2 ms
```

(iptable listing on 192.168.0.41)



```
Chain SSH (2 references)
    pkts      bytes target     prot opt in     out     source               destination
      10       400 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

(wireshark capture on 192.168.0.41)



ip.src==192.168.0.44

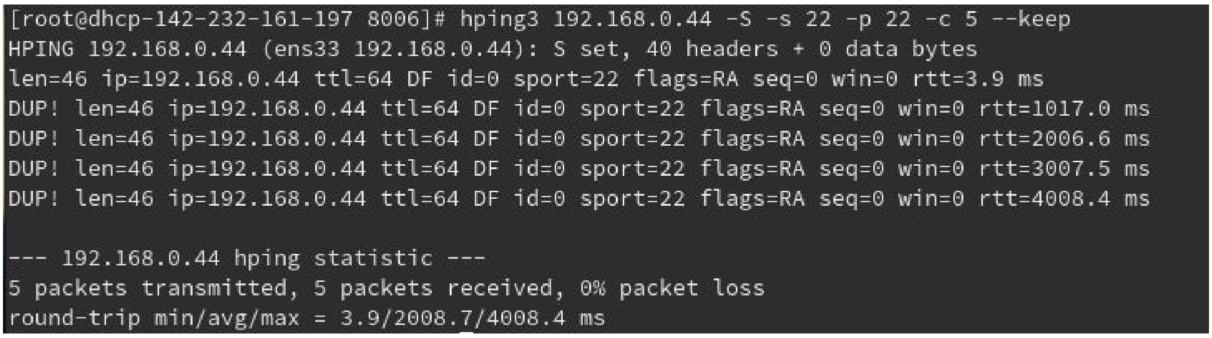
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
12	2.423546	192.168.0.44	192.168.0.41	TCP	60	0	1000 → 22
24	3.398655	192.168.0.44	192.168.0.41	TCP	60	0	1001 → 22
36	4.399194	192.168.0.44	192.168.0.41	TCP	60	0	1002 → 22
44	5.400536	192.168.0.44	192.168.0.41	TCP	60	0	1003 → 22
47	6.400653	192.168.0.44	192.168.0.41	TCP	60	0	1004 → 22

Case #2 192.168.0.41 -> 192.168.0.44

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

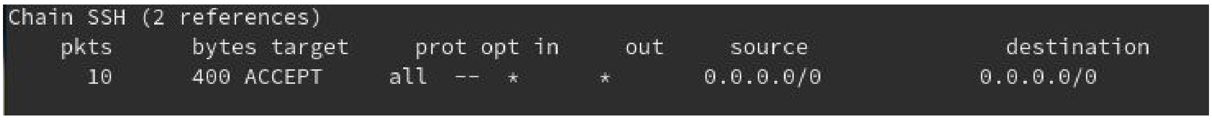
Detail: Machine 1 probes Machine 2 with ssh packets five times and got response. So a total of 10 packets should should up in the **SSH** chain that goes to **ACCEPT**(from both input and output chain). The wireshark captures shows that 5 outgoing traffic (port 22 to port 22) was put onto the network.

(hping3 output on 192.168.0.41)



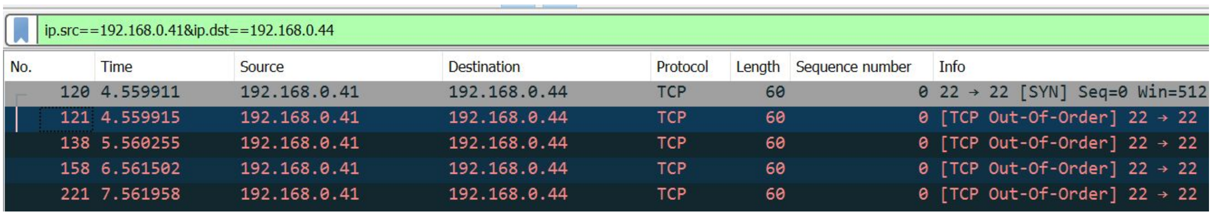
```
[root@dhcp-142-232-161-197 8006]# hping3 192.168.0.44 -S -s 22 -p 22 -c 5 --keep
HPING 192.168.0.44 (ens33 192.168.0.44): S set, 40 headers + 0 data bytes
len=46 ip=192.168.0.44 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=3.9 ms
DUP! len=46 ip=192.168.0.44 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=1017.0 ms
DUP! len=46 ip=192.168.0.44 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=2006.6 ms
DUP! len=46 ip=192.168.0.44 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=3007.5 ms
DUP! len=46 ip=192.168.0.44 ttl=64 DF id=0 sport=22 flags=RA seq=0 win=0 rtt=4008.4 ms

--- 192.168.0.44 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.9/2008.7/4008.4 ms
```

(iptable listing on 192.168.0.41)



```
Chain SSH (2 references)
    pkts      bytes target     prot opt in     out     source               destination
      10       400 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

(wireshark capture on 192.168.0.41)



ip.src==192.168.0.41&ip.dst==192.168.0.44

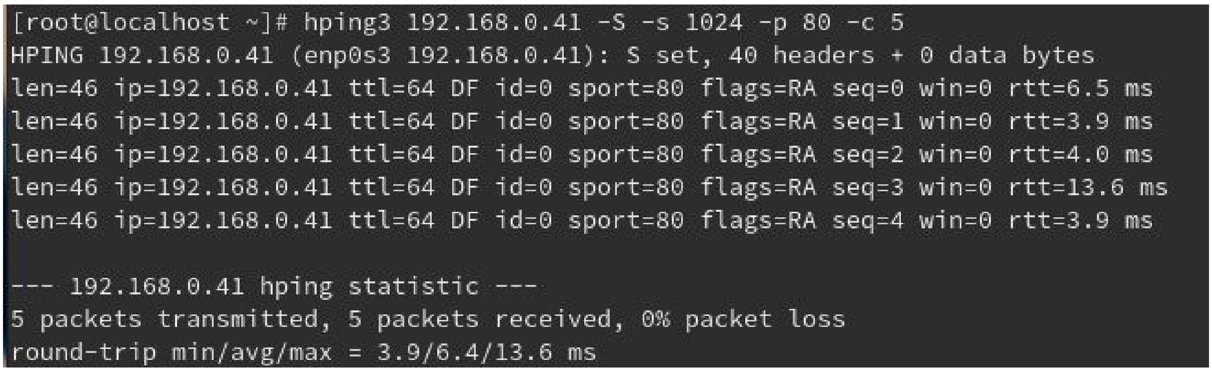
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
120	4.559911	192.168.0.41	192.168.0.44	TCP	60	0	22 → 22 [SYN] Seq=0 Win=512
121	4.559915	192.168.0.41	192.168.0.44	TCP	60	0	[TCP Out-Of-Order] 22 → 22
138	5.560255	192.168.0.41	192.168.0.44	TCP	60	0	[TCP Out-Of-Order] 22 → 22
158	6.561502	192.168.0.41	192.168.0.44	TCP	60	0	[TCP Out-Of-Order] 22 → 22
221	7.561958	192.168.0.41	192.168.0.44	TCP	60	0	[TCP Out-Of-Order] 22 → 22

Case #3 192.168.0.44 -> 192.168.0.41

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

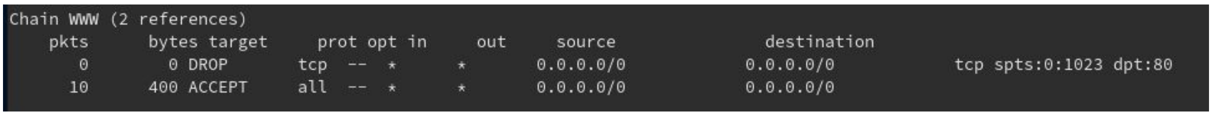
Detail: Machine 2 probes Machine with www packets 1 five times and got response. So a total of 10 packets should should up in the **WWW** chain that goes to **ACCEPT**(from both input and output chain). The wireshark captures shows that 5 incoming traffic (port 1024~ to port 80) was received from the network.

(hping3 output on 192.168.0.44)



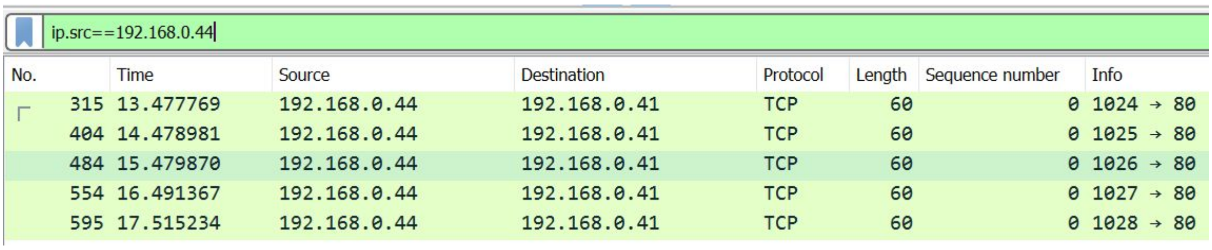
```
[root@localhost ~]# hping3 192.168.0.41 -S -s 1024 -p 80 -c 5
HPING 192.168.0.41 (enp0s3 192.168.0.41): S set, 40 headers + 0 data bytes
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=6.5 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=80 flags=RA seq=1 win=0 rtt=3.9 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=80 flags=RA seq=2 win=0 rtt=4.0 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=80 flags=RA seq=3 win=0 rtt=13.6 ms
len=46 ip=192.168.0.41 ttl=64 DF id=0 sport=80 flags=RA seq=4 win=0 rtt=3.9 ms

--- 192.168.0.41 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.9/6.4/13.6 ms
```

(iptable listing on 192.168.0.41)



```
Chain WWW (2 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spts:0:1023 dpt:80
      10      400 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

(wireshark capture on 192.168.0.41)



ip.src==192.168.0.44

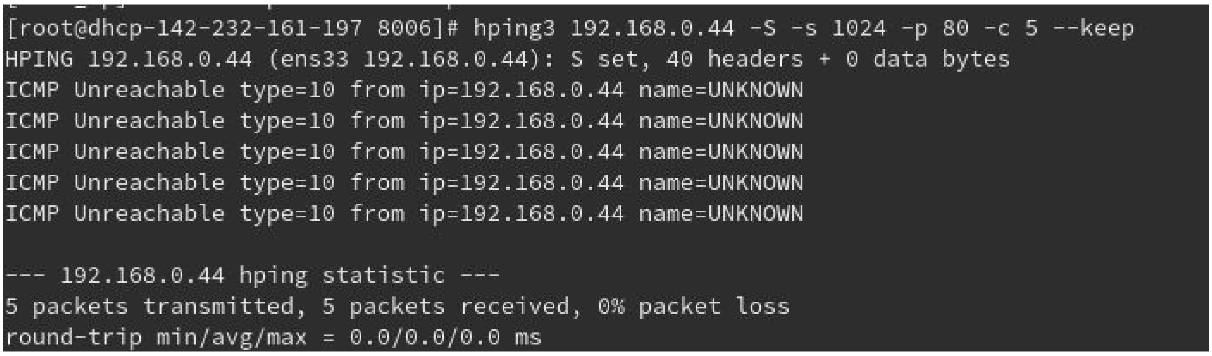
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
315	13.477769	192.168.0.44	192.168.0.41	TCP	60	0	1024 → 80
404	14.478981	192.168.0.44	192.168.0.41	TCP	60	0	1025 → 80
484	15.479870	192.168.0.44	192.168.0.41	TCP	60	0	1026 → 80
554	16.491367	192.168.0.44	192.168.0.41	TCP	60	0	1027 → 80
595	17.515234	192.168.0.44	192.168.0.41	TCP	60	0	1028 → 80

Case #4 192.168.0.41 -> 192.168.0.44

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

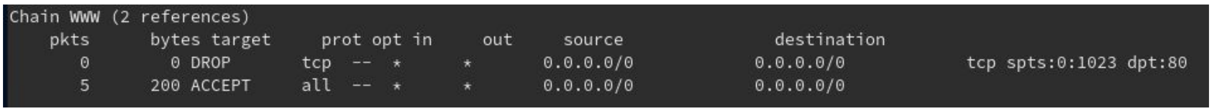
Detail: Machine 1 probes Machine 2 with www packets five times and got no response. So a total of 5 packets should should up in the **WWW** chain that goes to **ACCEPT** (from output chain). The wireshark captures shows that 5 outgoing traffic (port 1024 to port 80) was put onto the network. *the unreachable simply indicates that the receiver did not respond back

(hping3 output on 192.168.0.41)



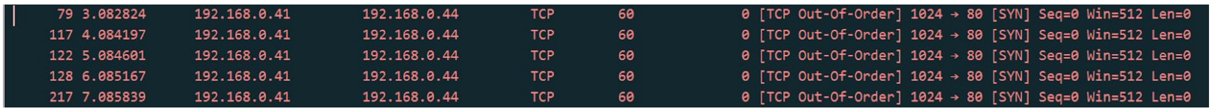
```
[root@dhcp-142-232-161-197 8006]# hping3 192.168.0.44 -S -s 1024 -p 80 -c 5 --keep
HPING 192.168.0.44 (ens33 192.168.0.44): S set, 40 headers + 0 data bytes
ICMP Unreachable type=10 from ip=192.168.0.44 name=UNKNOWN
ICMP Unreachable type=10 from ip=192.168.0.44 name=UNKNOWN
ICMP Unreachable type=10 from ip=192.168.0.44 name=UNKNOWN
ICMP Unreachable type=10 from ip=192.168.0.44 name=UNKNOWN
ICMP Unreachable type=10 from ip=192.168.0.44 name=UNKNOWN

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

(iptable listing on 192.168.0.41)



```
Chain WWW (2 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spts:0:1023 dpt:80
       5      200 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

(wireshark capture on 192.168.0.41)



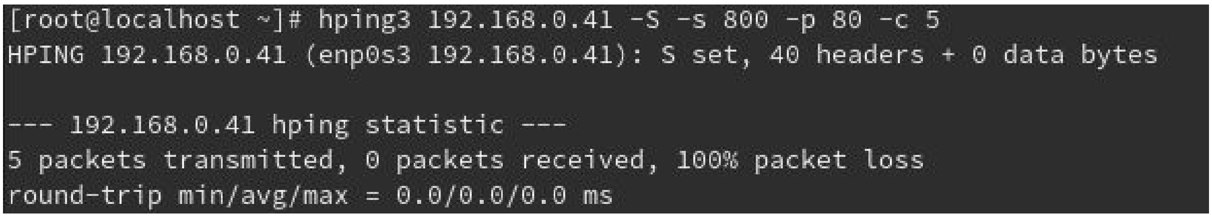
```
  79 3.082824     192.168.0.41     192.168.0.44     TCP     60     0 [TCP Out-Of-Order] 1024 → 80 [SYN] Seq=0 Win=512 Len=0
 117 4.084197     192.168.0.41     192.168.0.44     TCP     60     0 [TCP Out-Of-Order] 1024 → 80 [SYN] Seq=0 Win=512 Len=0
 122 5.084601     192.168.0.41     192.168.0.44     TCP     60     0 [TCP Out-Of-Order] 1024 → 80 [SYN] Seq=0 Win=512 Len=0
 128 6.085167     192.168.0.41     192.168.0.44     TCP     60     0 [TCP Out-Of-Order] 1024 → 80 [SYN] Seq=0 Win=512 Len=0
 217 7.085839     192.168.0.41     192.168.0.44     TCP     60     0 [TCP Out-Of-Order] 1024 → 80 [SYN] Seq=0 Win=512 Len=0
```

Case #5 192.168.0.44 -> 192.168.0.41

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

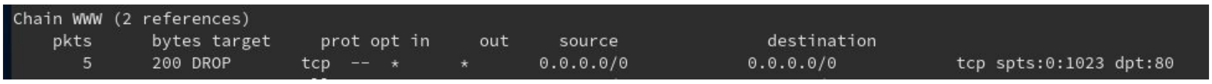
Detail: Machine 2 probes Machine 1 with www packets from port 800 five times and got no response. So a total of 5 packets should should up in the **WWW** (from input)**.** Because the rule denies any incoming traffic from port <1024 to port 80, the packets were sent to **DROP**. The wireshark captures shows that 5 incoming traffic (port 800~ to port 80) was received from the network.

(hping3 output on 192.168.0.44)



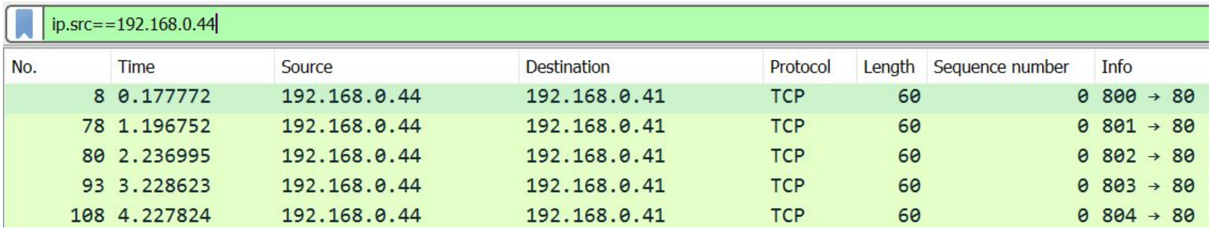
```
[root@localhost ~]# hping3 192.168.0.41 -S -s 800 -p 80 -c 5
HPING 192.168.0.41 (enp0s3 192.168.0.41): S set, 40 headers + 0 data bytes

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

(iptable listing on 192.168.0.41)



```
Chain WWW (2 references)
    pkts      bytes target     prot opt in     out     source               destination
       5        200 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spts:0:1023 dpt:80
```

(wireshark capture on 192.168.0.41)



ip.src==192.168.0.44

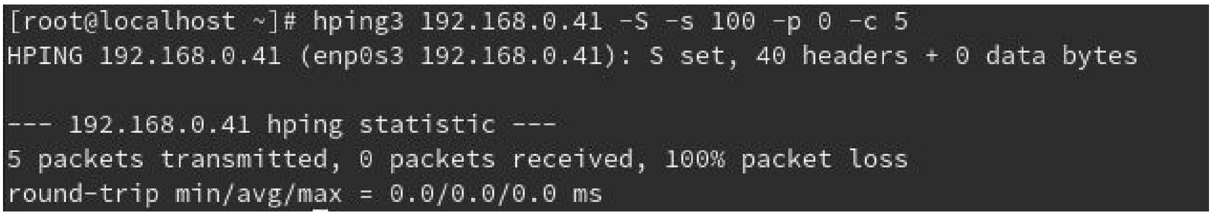
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
8	0.177772	192.168.0.44	192.168.0.41	TCP	60	0	800 → 80
78	1.196752	192.168.0.44	192.168.0.41	TCP	60	0	801 → 80
80	2.236995	192.168.0.44	192.168.0.41	TCP	60	0	802 → 80
93	3.228623	192.168.0.44	192.168.0.41	TCP	60	0	803 → 80
108	4.227824	192.168.0.44	192.168.0.41	TCP	60	0	804 → 80

Case #6 192.168.0.44 -> 192.168.0.41

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

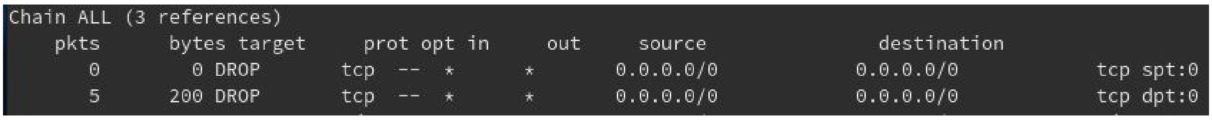
Detail: Machine 2 probes Machine 1 with 0 dport packets five times and got no response. So a total of 5 packets should should up in the **ALL** chain (from input). Because the rule denies any incoming traffic to port 0, the packets were sent to **DROP**. The wireshark captures shows that 5 incoming traffic (port 100~ to 0) was received from the network.

(hping3 output on 192.168.0.44)



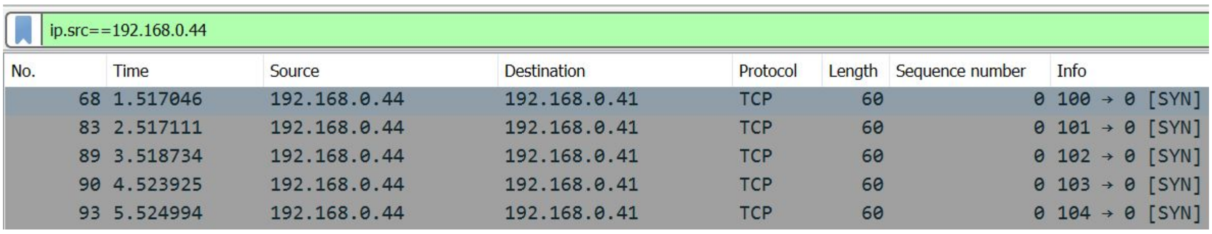
```
[root@localhost ~]# hping3 192.168.0.41 -S -s 100 -p 0 -c 5
HPING 192.168.0.41 (enp0s3 192.168.0.41): S set, 40 headers + 0 data bytes

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

(iptable listing on 192.168.0.41)



```
Chain ALL (3 references)
    pkts      bytes target     prot opt in     out     source               destination
       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
```

(wireshark capture on 192.168.0.41)



ip.src==192.168.0.44

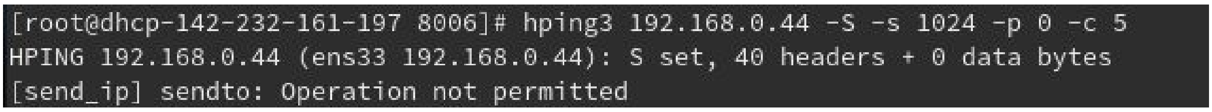
No.	Time	Source	Destination	Protocol	Length	Sequence number	Info
68	1.517046	192.168.0.44	192.168.0.41	TCP	60	0	100 → 0 [SYN]
83	2.517111	192.168.0.44	192.168.0.41	TCP	60	0	101 → 0 [SYN]
89	3.518734	192.168.0.44	192.168.0.41	TCP	60	0	102 → 0 [SYN]
90	4.523925	192.168.0.44	192.168.0.41	TCP	60	0	103 → 0 [SYN]
93	5.524994	192.168.0.44	192.168.0.41	TCP	60	0	104 → 0 [SYN]

Case #7 192.168.0.41 -> 192.168.0.44

Local machine 1 ip: 192.168.0.41
Local machine 2 ip: 192.168.0.44

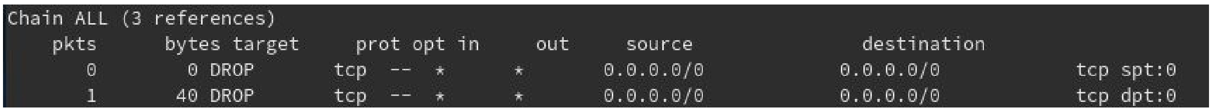
Detail: Machine 1 probes Machine 2 with 0 dport packets five times and got “Operation not permitted”. So a total of 1 packets should should up in the **ALL** chain (from output). Because the rule denies any outgoing traffic from port 0, the packets were sent to **DROP**. No wireshark capture for this case because none of the packets were put onto the network.

(hping3 output on 192.168.0.41)



```
[root@dhcp-142-232-161-197 8006]# hping3 192.168.0.44 -S -s 1024 -p 0 -c 5
HPING 192.168.0.44 (ens33 192.168.0.44): S set, 40 headers + 0 data bytes
[send_ip] sendto: Operation not permitted
```

(iptable listing on 192.168.0.41)



```
Chain ALL (3 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp spt:0
       1       40 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:0
```