Assignment #1

Simple Personal Linux Firewall

Steffen L. Norgren
A00683006

COMP 8006 - Network Administration & Security II • BCIT • January 28, 2010

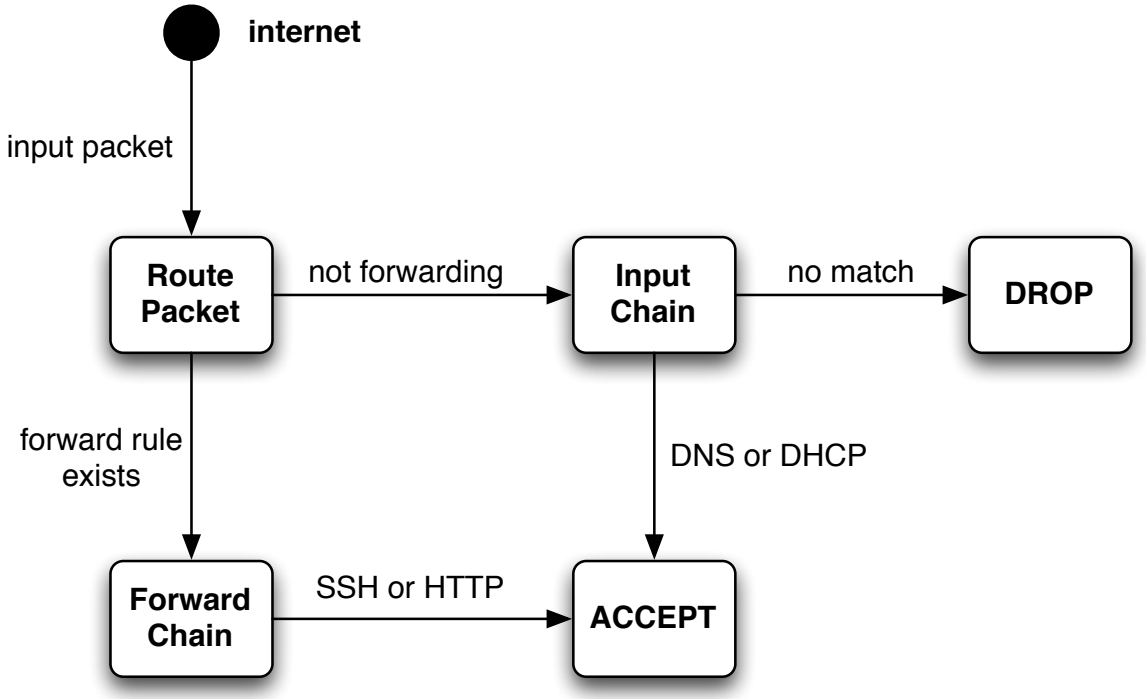
Table of Contents

Overview

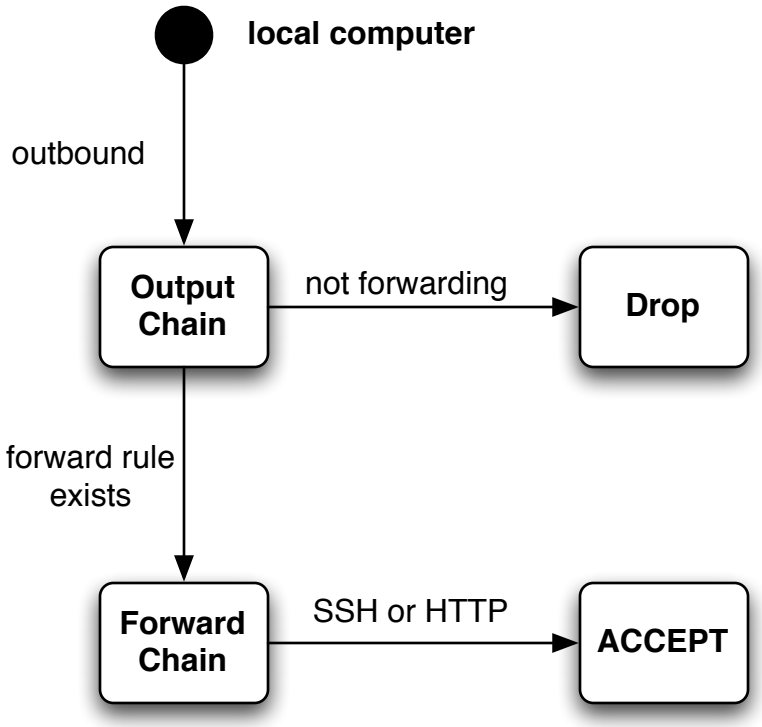
This assignment was for the purpose of creating a simple Linux firewall where everything would be blocked with the exception of SSH and HTTP traffic. To do this, I had to first create a default DROP policy and then selectively open the needed ports, which also included DHCP and DNS. Additionally, I needed make sure that the firewall allowed communication via established connections.

Design Work

Inbound Traffic



Outbound Traffic



TESTING

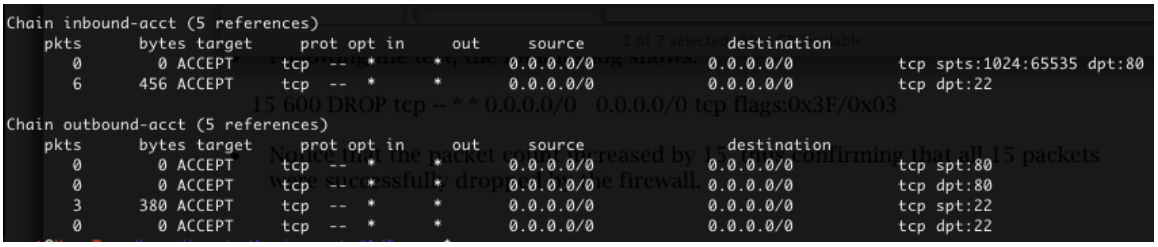
Testing of the firewall was mostly done via hping3 and verifying connectivity of the various open ports. Additionally, I verified DHCP connectivity by disabling the network card, turning on the firewall and then re-enabling the network card to see if it could grab an IP address, which it did.

Rule #	Test Description	Tool Used	Expected Results	Pass/Fail
1	Permit inbound/ outbound SSH packets.	Remotely access the SSH server	The iptables -L -n -v -x audit should show the traffic.	Pass. Detailed results are attached.
2	Permit inbound/ outbound HTTP packets.	hping3 & opening a browser session.	hping2 should show a response on port 80 and and the iptables -L -n -v -x audit should show the traffic.	Pass. Detailed results are attached.
3	Drop traffic to port 80 from source port < 1024	hping3	hping2 should not show any response and the iptables -L -n -v -x audit should NOT show the traffic.	Pass. Detailed results are attached.
4	Drop all incoming packets from/to port 0.	hping3	hping2 should not show any response and the iptables -L -n -v -x audit should show the dropped traffic.	Pass. Detailed results are attached.
5	Drop all incoming packets that have both the SYN and FIN bits set together.	hping3	hping2 should not show any response and the iptables -L -n -v -x audit should show the dropped traffic.	Pass. Detailed results are attached.
6	Allow outbound DNS & DHCP packets	nslookup	Should return results for any particular domain and the iptables -L -n -v -x audit should show the traffic.	Pass. Detailed results are attached.
7	Drop all inbound traffic except for SSH and HTTP traffic.	zenmap	Through scanning all 65535 ports, we should only see the SSH and HTTP services open. Additionally, the iptables -L -n -v -x audit should show dropped traffic.	Pass. Detailed results are attached.

Test Case #1

This was a simple connectivity test where I SSH'd into the computer remotely. Additionally I tested whether the server could also SSH out to other computers.

Before the test:

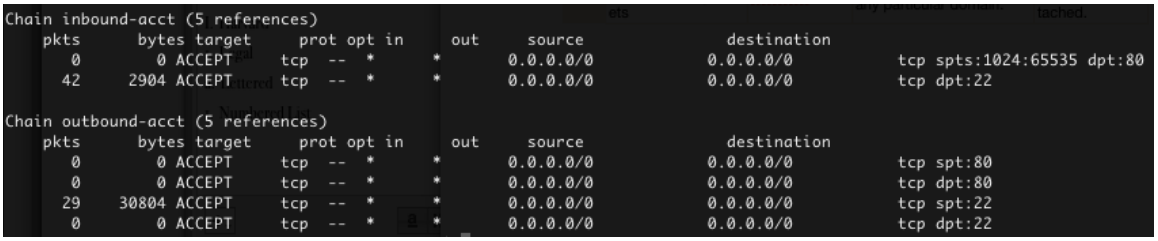


```
Chain inbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:1024:65535 dpt:80
       6      456 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22

Chain outbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
       3      380 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
```

After the test:

One can see that the inbound and outbound accounting chains registered several packets.



```
Chain inbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:1024:65535 dpt:80
      42     2904 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22

Chain outbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
      29    30804 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
```

Test Case #2

Verified whether the server responded to HTTP traffic on port 80 from source ports higher than 1024. Note that in this case there wasn't an actual web server running at the time.

```
strumpet ~ # hping2 192.168.1.185 -p 80 -s 1241 -S
HPING 192.168.1.185 (eth0 192.168.1.185): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.185 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=2.1 ms
len=46 ip=192.168.1.185 ttl=64 DF id=0 sport=80 flags=RA seq=1 win=0 rtt=242.0 ms
len=46 ip=192.168.1.185 ttl=64 DF id=0 sport=80 flags=RA seq=2 win=0 rtt=163.8 ms
len=46 ip=192.168.1.185 ttl=64 DF id=0 sport=80 flags=RA seq=3 win=0 rtt=85.4 ms
^C
--- 192.168.1.185 hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2.1/123.3/242.0 ms
strumpet ~ #
```

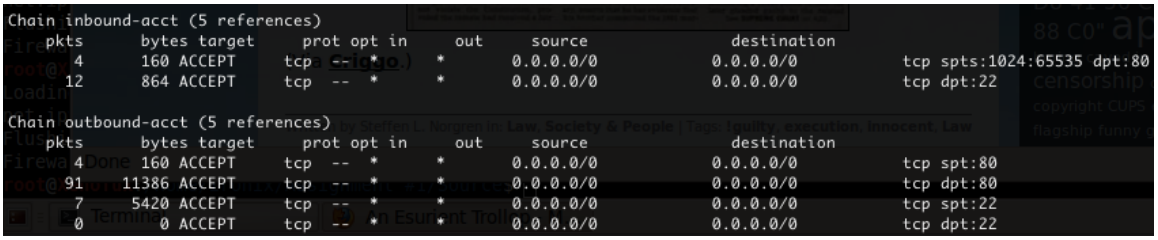
Before the test:

```
Chain inbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spts:1024:65535 dpt:80
       6      456 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22

Chain outbound-acct (5 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
       3      380 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
```

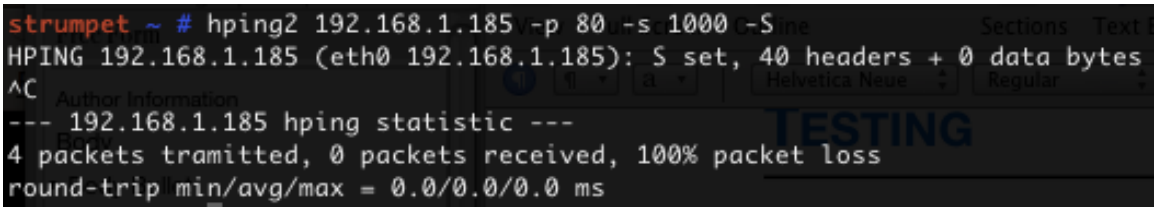
After the test:

You can see that packets were registered both arriving on port 80 in inbound chain as well as leaving on the outbound chain.

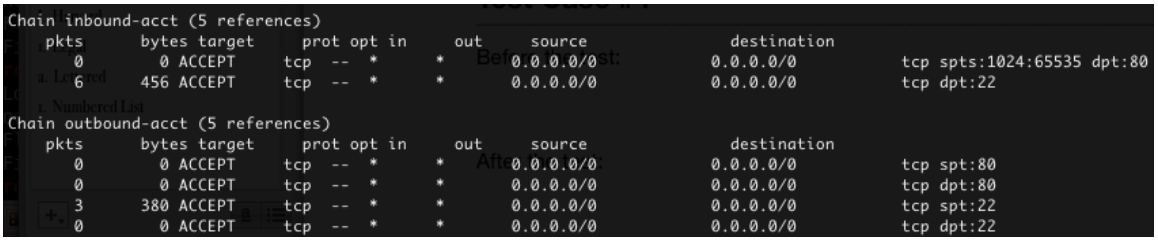


Test Case #3

Again, tested HTTP connectivity, except this time I use a source port lower than 1024. In this case the packets were in fact dropped.

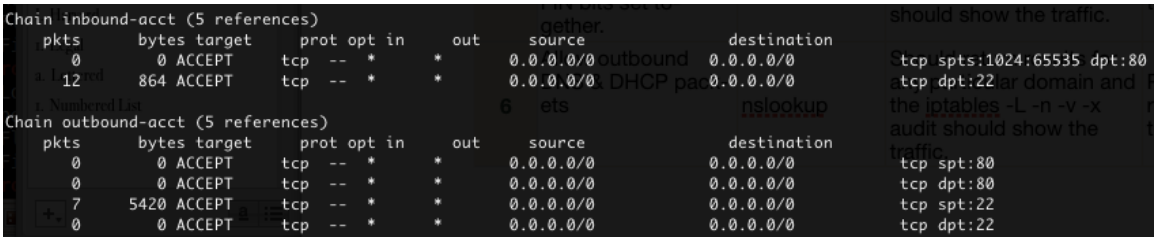


Before the test:



After the test:

As you can see, no packets arrived or departed via the accounting chains.



Test Case #4

Testing how the server handles packets arriving from a source or destination port of 0, which is a reserved system port.

```
root@XenoTux:~$ hping3 192.168.227.130 -s 0 -d 22 -S
HPING 192.168.227.130 (eth1 192.168.227.130): S set, 40 headers + 22 data bytes
^C
--- 192.168.227.130 hping statistic ---
9 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@XenoTux:~$ hping3 192.168.227.130 -s 1234 -d 0 -S
HPING 192.168.227.130 (eth1 192.168.227.130): S set, 40 headers + 0 data bytes
^C
--- 192.168.227.130 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@XenoTux:~$
```

Before the test:

```
Chain INPUT (policy DROP 1 packets, 52 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
       0        0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp 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 dpt:0
       0        0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
       0        0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
      40     3040 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
       0        0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

After the test:

The INPUT chain shows that these packets were properly dropped.

```
Chain INPUT (policy DROP 1 packets, 52 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
       1       62 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
       0        0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:0
      13      696 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 dpt:0
       0        0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
       0        0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
       2      274 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
      44     3344 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
       0        0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

Test Case #5

Here we test whether the server responds to SYN,FIN packets. The hping2 (typo) results show that all the packets were lost.

```
strumpet ~ # hping2 192.168.1.185 -p 22 -s 1274 -S -F
HPING 192.168.1.185 (eth0 192.168.1.185): SF set, 40 headers + 0 data bytes
^C
--- 192.168.1.185 hping statistic ---
4 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Before the test:

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp 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 dpt:0
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
    6   456 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

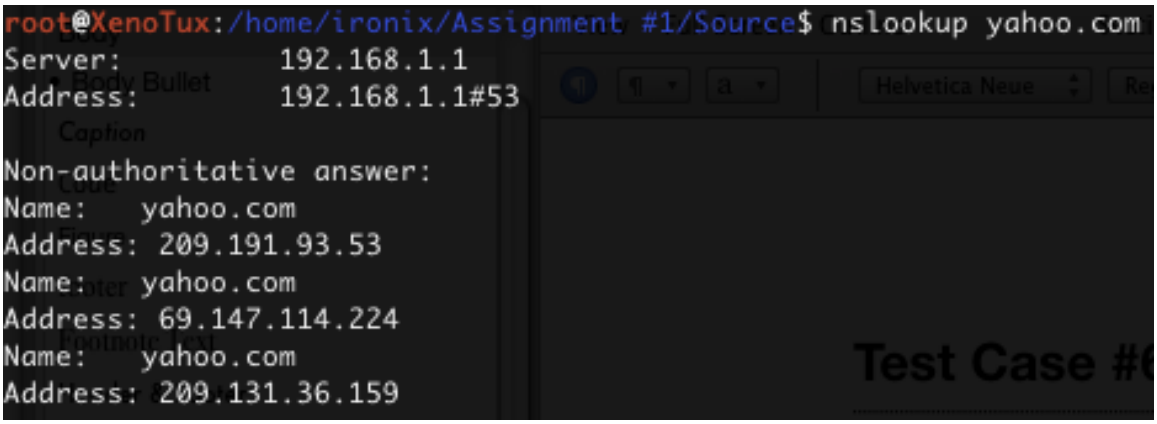
After the test:

Here we can see that all the packets were dropped as invalid.

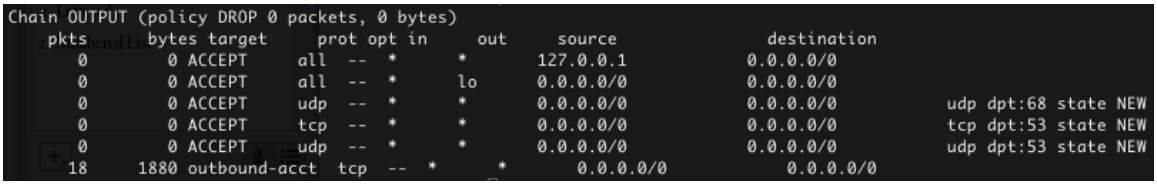
```
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp 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 dpt:0
    7   280 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    5   823 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
   11   812 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

Test Case #6

Using nslookup to verify that DNS resolution takes place successfully. This is done from the firewalled computer itself.

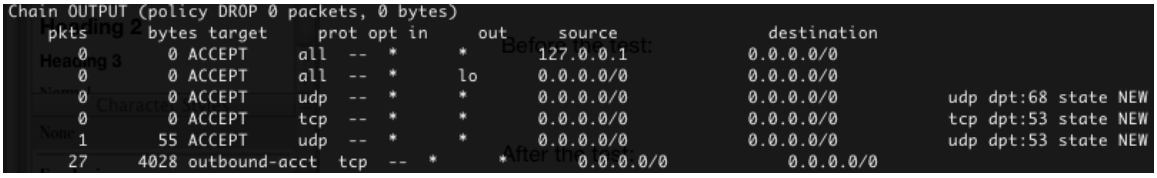


Before the test:



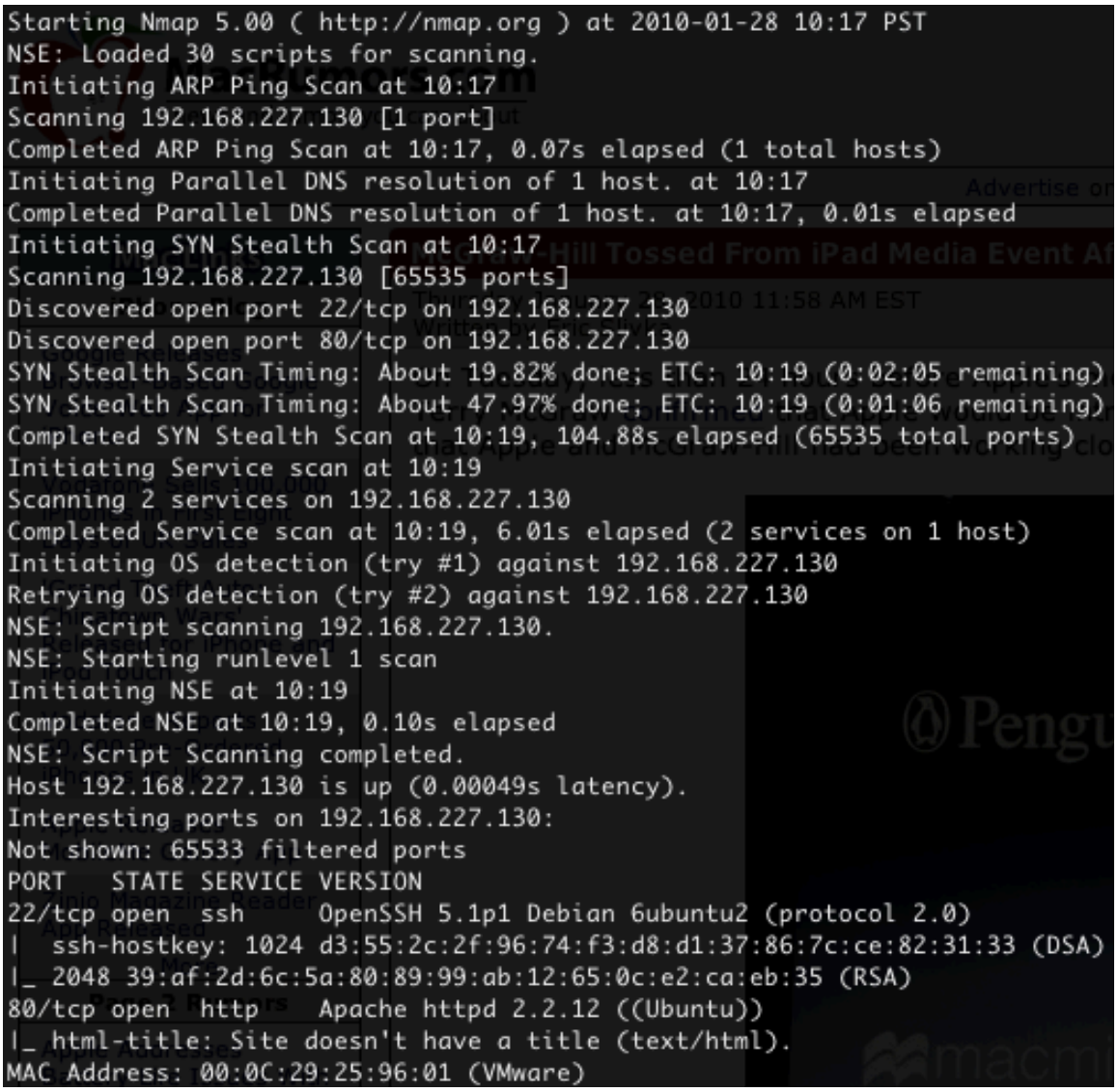
After the test:

There is a registered UDP packet that registered on the OUTPUT chain with the associated rule.



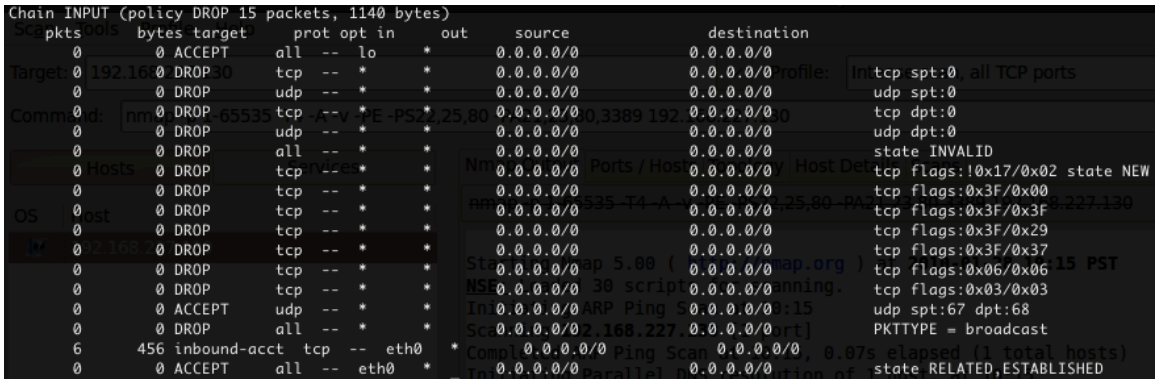
Test Case #7

Only the SSH and HTTP ports were discovered during the scan.



```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-01-28 10:17 PST
NSE: Loaded 30 scripts for scanning.
Initiating ARP Ping Scan at 10:17
Scanning 192.168.227.130 [1 port]
Completed ARP Ping Scan at 10:17, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:17
Completed Parallel DNS resolution of 1 host. at 10:17, 0.01s elapsed
Initiating SYN Stealth Scan at 10:17
Scanning 192.168.227.130 [65535 ports]
Discovered open port 22/tcp on 192.168.227.130
Discovered open port 80/tcp on 192.168.227.130
SYN Stealth Scan Timing: About 19.82% done; ETC: 10:19 (0:02:05 remaining)
SYN Stealth Scan Timing: About 47.97% done; ETC: 10:19 (0:01:06 remaining)
Completed SYN Stealth Scan at 10:19, 104.88s elapsed (65535 total ports)
Initiating Service scan at 10:19
Scanning 2 services on 192.168.227.130
Completed Service scan at 10:19, 6.01s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.227.130
Retrying OS detection (try #2) against 192.168.227.130
NSE: Script scanning 192.168.227.130.
NSE: Starting runlevel 1 scan
Initiating NSE at 10:19
Completed NSE at 10:19, 0.10s elapsed
NSE: Script Scanning completed.
Host 192.168.227.130 is up (0.00049s latency).
Interesting ports on 192.168.227.130:
Not shown: 65533 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.1p1 Debian 6ubuntu2 (protocol 2.0)
|  ssh-hostkey: 1024 d3:55:2c:2f:96:74:f3:d8:d1:37:86:7c:ce:82:31:33 (DSA)
|_ 2048 39:af:2d:6c:5a:80:89:99:ab:12:65:0c:e2:ca:eb:35 (RSA)
80/tcp open  http    Apache httpd 2.2.12 ((Ubuntu))
|_ html-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:25:96:01 (VMware)
```

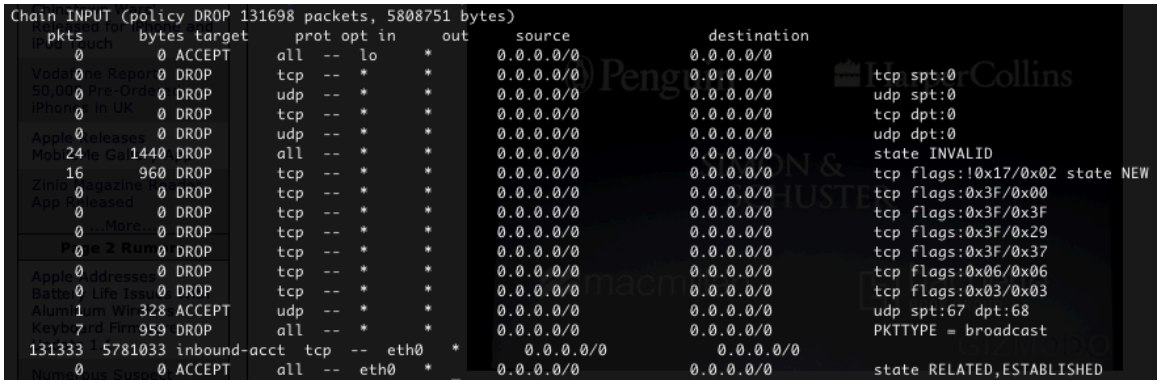
Before the test:



```
Chain INPUT (policy DROP 15 packets, 1140 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp 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 dpt:0
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
    6   456 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

After the test:

As one can see, many, many packets were dropped as part of this scan, which included stealth SYN scanning as well.



```
Chain INPUT (policy DROP 131698 packets, 5808751 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp 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 dpt:0
   24  1440 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
   16   960 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:!0x17/0x02 state NEW
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x00
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x3F
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x29
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x3F/0x37
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x06/0x06
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp flags:0x03/0x03
    1   328 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    7   959 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           PKTTYPE = broadcast
131333 5781033 inbound-acct  tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```