Personal Firewalls - An Introduction to Firewall Administration

David Tran - A00801942

COMP 6D

COMP 8006

British Columbia Institute of Technology

Aman Abdulla

Thursday, February 6 2014

# Table of Contents

# Background

The purpose of this assignment is to familiarize the budding administrator with the syntax and implementation of firewall rules on a Linux machine. The objective is to design a set of rules that follow the criteria of Assignment 1, as well as documenting a set of test cases properly so that the routine becomes natural. Testing in this assignment will require a tertiary computer terminal located in the same local area network. In this test case, we will assume that the tertiary computer will not have anything but the default firewall in place.

# Tools & Equipment

## Hardware

|  |  |  |
| --- | --- | --- |
| * 8GB RAM | * Intel i5 Quad Core | * 500GB HDD |
| * two computer terminals |  |  |

## Software

|  |  |  |
| --- | --- | --- |
| * Fedora Linux 19 64-bit | * hping3 | * Shell Script |
| * Wireshark | * iptables | * ssh |
| * dhclient |  |  |

# Testing Procedure

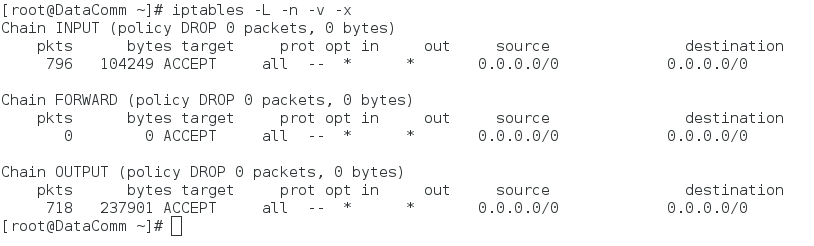
## Test Cases Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Test Case** | **Tools Used** | **Expected Outcome** | **Results** |
| 1 | Verify that iptables settings are changed after script is ran | iptables -L -n -v -x | Rules are changed according to script | PASSED. See details. |
| 2 | DNS packets are allowed to and from the host | Wireshark,  Google Chrome | Packets are allowed to travel to and from the machine | PASSED. See details. |
| 3 | DHCP packets are allowed to and from the host | ifconfig, dhclient | Packets are allowed to travel to and from the machine | PASSED. See details. |
| 4 | SSH traffic is allowed and accounted for | ssh,  iptables -L -n -v -x | SSH traffic is permitted inbound and outbound | PASSED. See details. |
| 5 | WWW(80) traffic is allowed and accounted for | hping3,  iptables -L -n -v -x | WWW traffic is permitted inbound and outbound | PASSED. See details. |
| 6 | WWW(443) traffic is allowed and accounted for | hping3,  iptables -L -n -v -x | WWW traffic is permitted inbound and outbound | PASSED. See details. |
| 7 | WWW(80) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is not permitted inbound and outbound using port 0 | PASSED. See details. |
| 8 | WWW(80) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is not permitted inbound and outbound using port 1023 | PASSED. See details. |
| 9 | WWW(80) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is permitted inbound and outbound using port 1024 | PASSED. See details. |
| 10 | WWW(443) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is not permitted inbound and outbound using port 0 | PASSED. See details. |
| 11 | WWW(443) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is not permitted inbound and outbound using port 1023 | PASSED. See details. |
| 12 | WWW(443) traffic is disallowed and accounted for when source ports are less than 1024 | hping3,  iptables -L -n -v -x | WWW traffic is permitted inbound and outbound using port 1024 | PASSED. See details. |
| 13 | Incoming packets to port 0 are dropped (TCP) | hping3,  iptables -L -n -v -x, | Incoming packets are dropped | PASSED. See details. |
| 14 | Incoming packets to port 0 are dropped (UDP) | hping3,  iptables -L -n -v -x, | Incoming packets are dropped | PASSED. See details. |
| 15 | WWW(80) incoming traffic is accounted for | hping3,  iptables -L -n -v -x | Instances of www-traffic is incremented | PASSED. See details. |
| 16 | WWW(443) incoming traffic is accounted for | hping3,  iptables -L -n -v -x | Instances of www-traffic is incremented | PASSED. See details. |
| 17 | WWW(80) outgoing traffic is accounted for | hping3,  iptables -L -n -v -x | Instances of www-traffic is incremented | PASSED. See details. |
| 18 | WWW(443) outgoing traffic is accounted for | hping3,  iptables -L -n -v -x | Instances of www-traffic is incremented | PASSED. See details. |
| 19 | SSH(22) incoming traffic is accounted for | ssh,  hping3,  iptables -L -n -v -x | Instances of ssh-traffic is incremented | PASSED. See details. |
| 20 | SSH(22) outgoing traffic is accounted for | ssh,  hping3,  iptables -L -n -v -x | Instances of ssh-traffic is incremented | PASSED. See details. |
| 21 | All inbound SYN packets are dropped | hping3 | SYN packets are dropped; exceptions are permitted | PASSED. See details. |
| 22 | All other undefined ports with inbound traffic are accounted for (TCP) | hping3,  iptables -L -n -v -x | Ports 22, 80, 443 will not increment, all others will increment | PASSED. See details. |
| 23 | All other undefined ports with outbound traffic are accounted for (TCP) | hping3,  iptables -L -n -v -x | Ports 22, 80, 443 will not increment, all others will increment | PASSED. See details. |
| 24 | All other undefined ports with inbound traffic are accounted for (UDP) | hping3,  iptables -L -n -v -x | Ports 22, 80, 443 will not increment, all others will increment | PASSED. See details. |
| 25 | All other undefined ports with inbound traffic are accounted for (UDP) | hping3,  iptables -L -n -v -x | Ports 22, 80, 443 will not increment, all others will increment | PASSED. See details. |

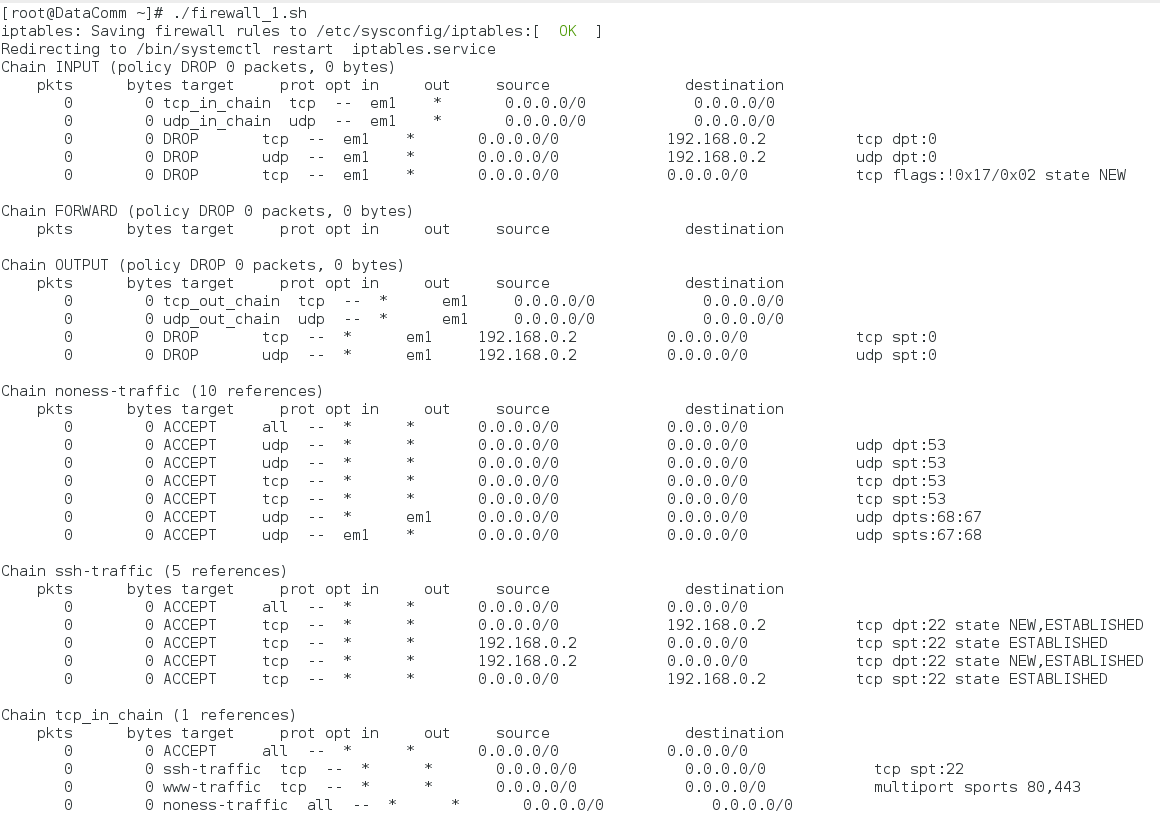
## Test Case Evidence & Details

(1) Verify that iptables settings are changed after script is ran

Before: iptables -L -n -v -x



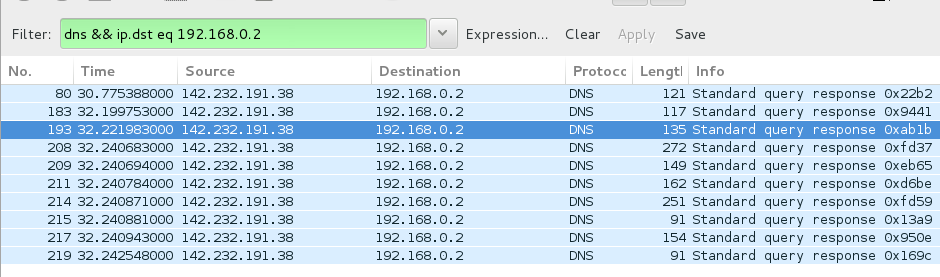
After: ./firewall\_1.sh



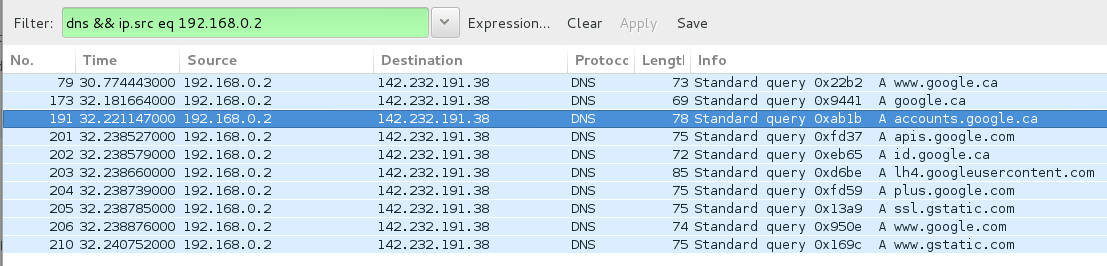
(2) DNS Packets are allowed to and from the host

To begin, we opened a browser and navigated to Google.ca (142.232.191.38)

Inbound DNS Packets:



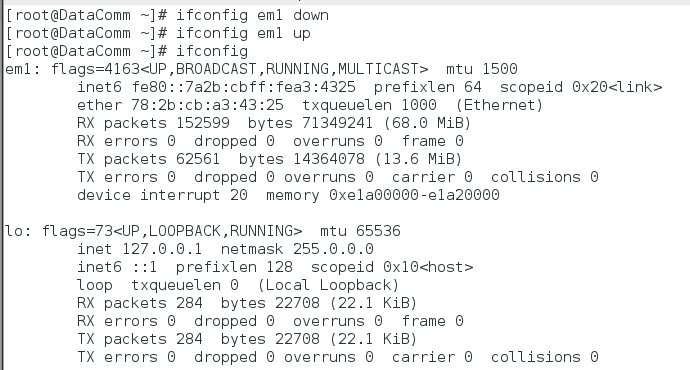
Outbound DNS Packets:



(3) DHCP Packets are allowed to and from the host

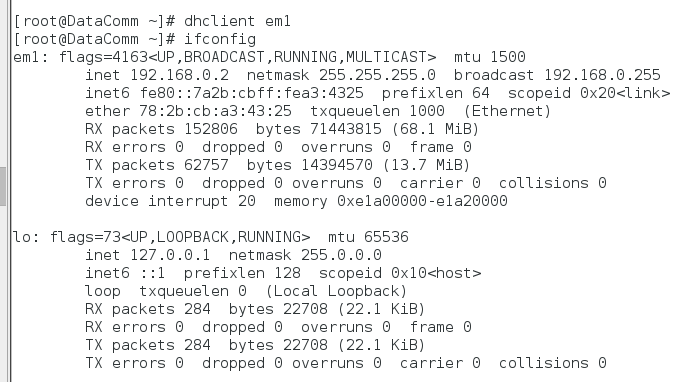
Firstly, in Terminal, we deactivated our network card and then brought it back up again using:

ipconfig em1 down; ipconfig em1 up;



Afterwards, we ran this command:

dhclient em1



We were then able to see that we were dedicated the same IPv4 Address again:

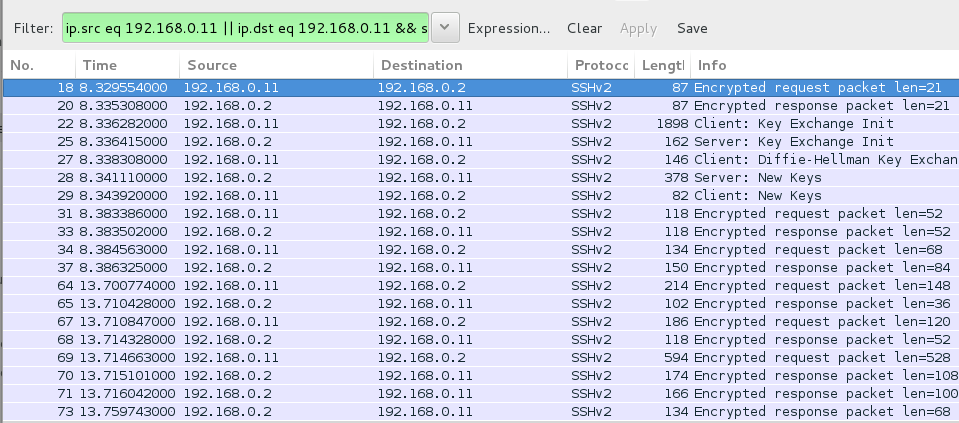
192.168.0.2

(4) SSH traffic is allowed to and from the host

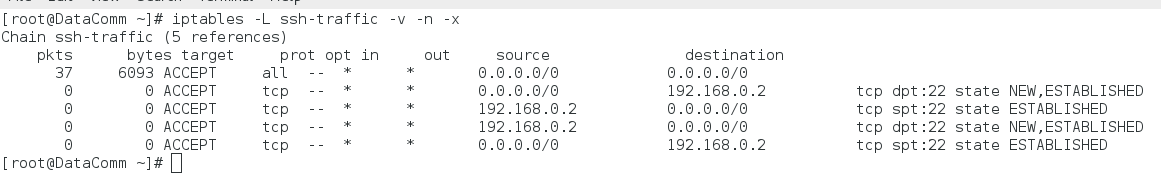
In a second computer terminal, assume that this terminal is the client. We entered the following command into Terminal:

ssh 192.168.0.2

Here’s the Wireshark capture of inbound and outbound traffic for SSHv2

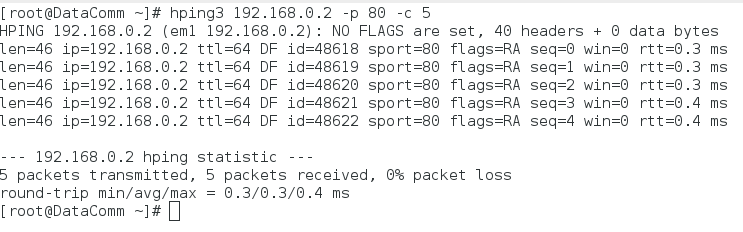


Here is the accounting of SSH traffic:

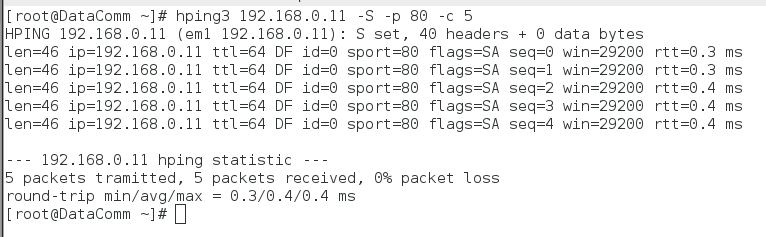


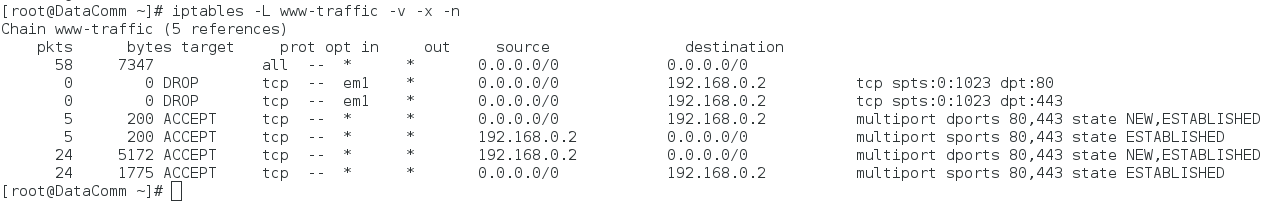
(5) WWW(80) traffic is allowed and accounted for

From the testing terminal, here are the results of the hping command to our personal firewall:



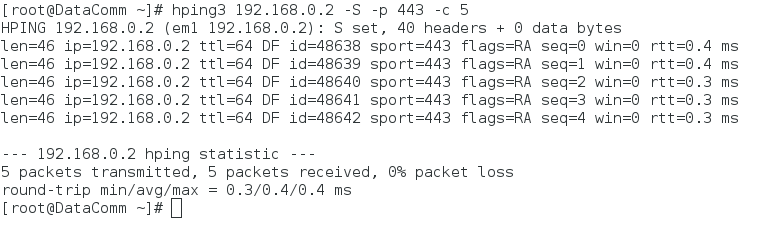
Here are the results of the machine containing our firewall sending outbound packets:

And here are the accounting information regarding WWW(80) traffic using iptables:

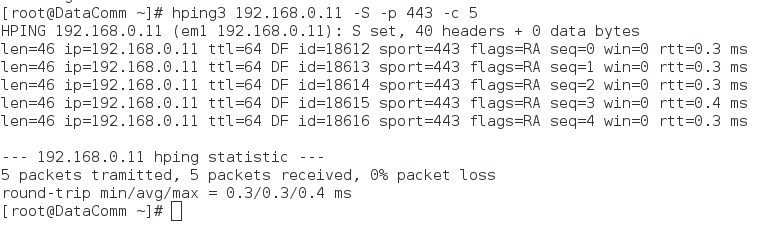


(6) WWW(443) traffic is allowed and accounted for

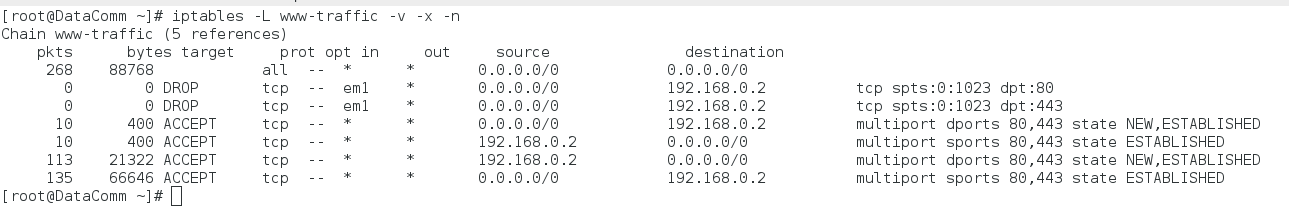
From the testing terminal, here are the results of the hping command to our personal firewall:



Here are the results of the machine containing our firewall sending outbound packets:



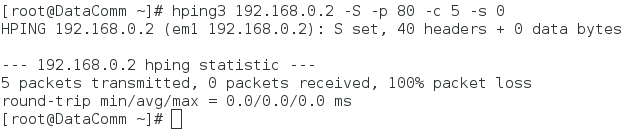
And here are the accounting information regarding WWW(80) traffic using iptables:



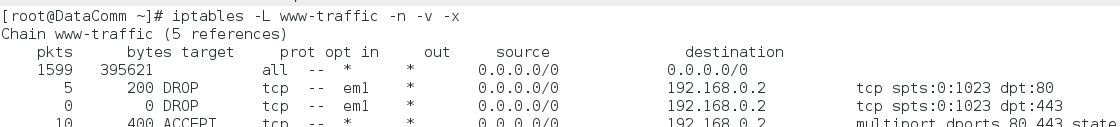
Note the increase of our packet count from 5 to 10.

(7) WWW(80) traffic is disallowed and accounted for when source ports are less than 1024

Here, we set our source port to the left-most extreme of our constraint (port 0). We will execute this on our non-firewalled terminal:



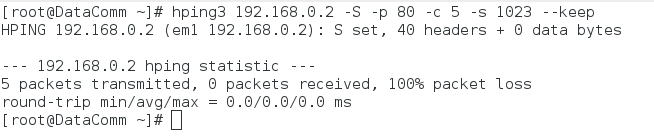
Our packets are dropped here. Let’s take a look at our packet auditing on our firewalled terminal:



(8) WWW(80) traffic is disallowed and accounted for when source ports are less than 1024

We will redo the test, except we will attempt our right-most extreme (port 1023).

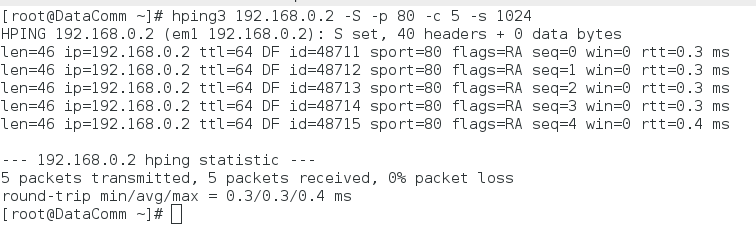
NOTE: we have to specify the --keep switch, otherwise, our ports will increment



(9) WWW(80) traffic is disallowed and accounted for when source ports are less than 1024

We will redo our previous test, except this time, we will try a valid port (port 1024+)

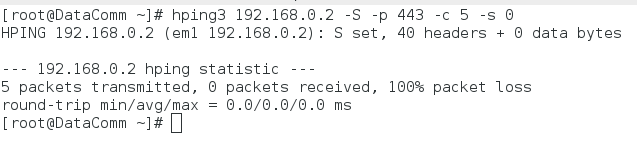
NOTE: we won’t have to specify --keep switch, because anything greater than 1024 is permitted



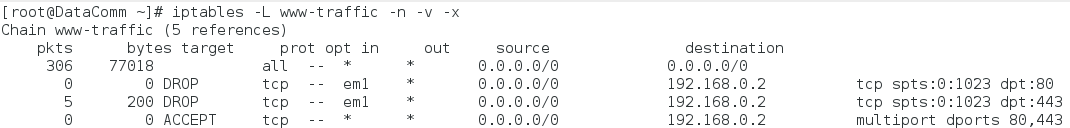
(10) WWW(443) traffic is disallowed and accounted for when source ports are less than 1024

We will flush our accounting traffic and restart the last 3 tests again for port 443 (https)

Starting with our left-most extreme (port 0):



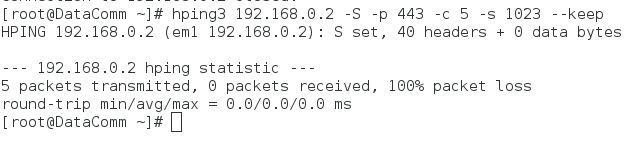
And then checking our accounting tables:



(11) WWW(443) traffic is disallowed and accounted for when source ports are less than 1024

NOTE: we have to specify --keep switch, otherwise our source port will increment

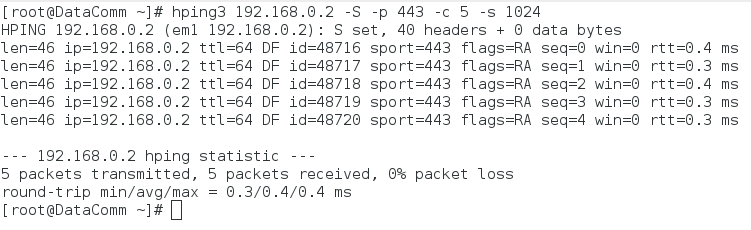
Starting with our right-most extreme (port 1023):



(12) WWW(443) traffic is disallowed and accounted for when source ports are less than 1024

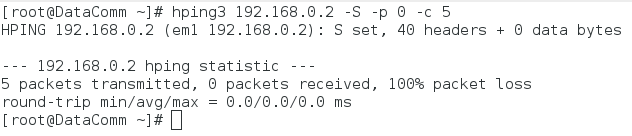
NOTE: we won’t have to specify --keep switch, because anything greater than 1024 is permitted

Using port 1024:

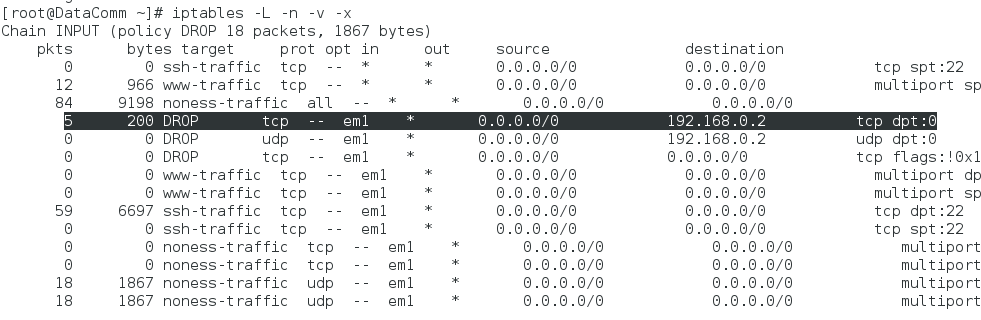


(13) Incoming packets to port 0 are dropped (TCP)

Here is our command:

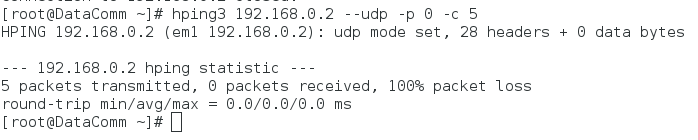


As you can see, no responses. Here is our accounting table to prove it:

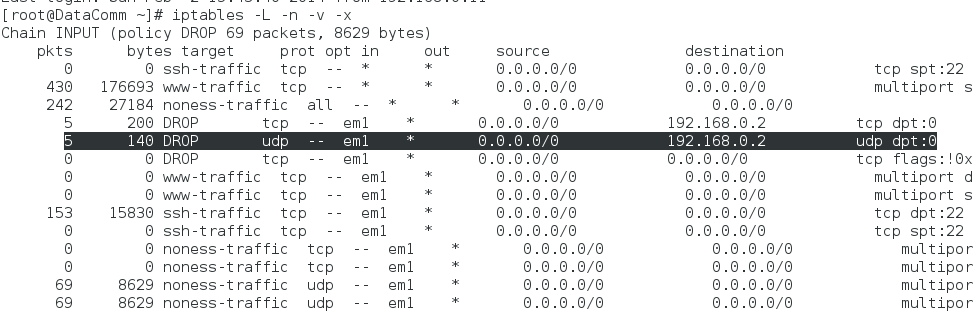


(14) Incoming packets to port 0 are dropped (UDP)

Here is our command; note the --udp switch:



As you can see, no responses. Here is our accounting table to prove it; note the difference between the TCP and UDP count:

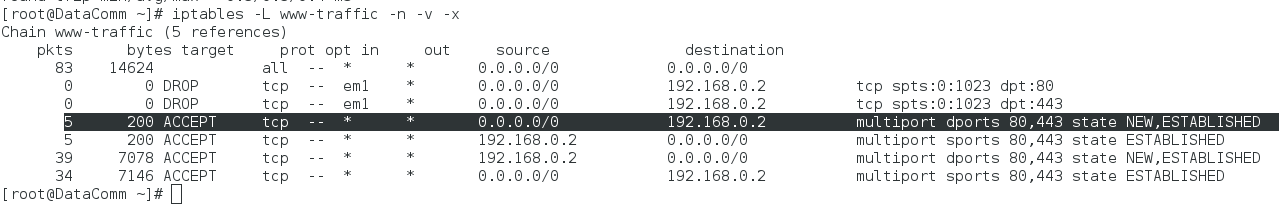


(15) WWW(80) incoming traffic is accounted for

In our non-firewall terminal, we ran this command:

hping3 192.168.0.2 -S -p 80 -c 5

Here is our accounting table:

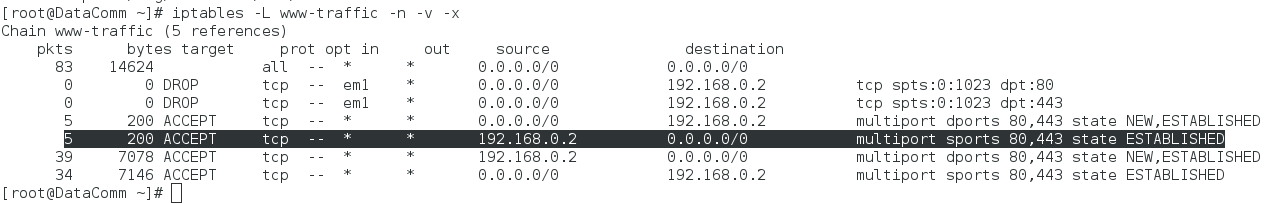


(16) WWW(80) outgoing traffic is accounted for

In our firewalled terminal, we ran this command:

hping3 192.168.0.11 -S -p 80 -c 5

Here is our accounting table:



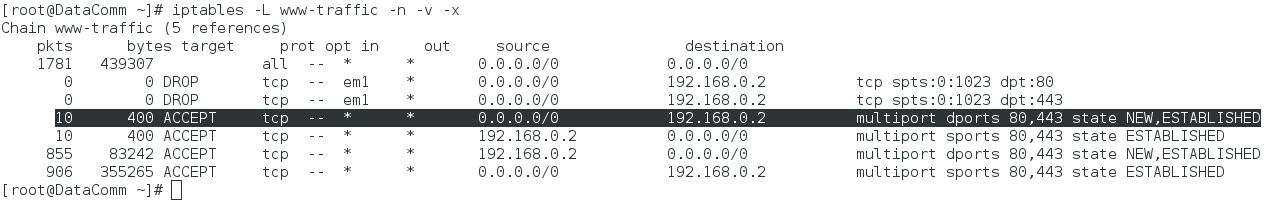
NOTE the two different highlighted lines: the former depicts incoming while the latter depicts outgoing.

(17) WWW(443) incoming traffic is accounted for

In our non-firewall terminal, we ran this command:

hping3 192.168.0.2 -S -p 443 -c 5

Here is our accounting table:

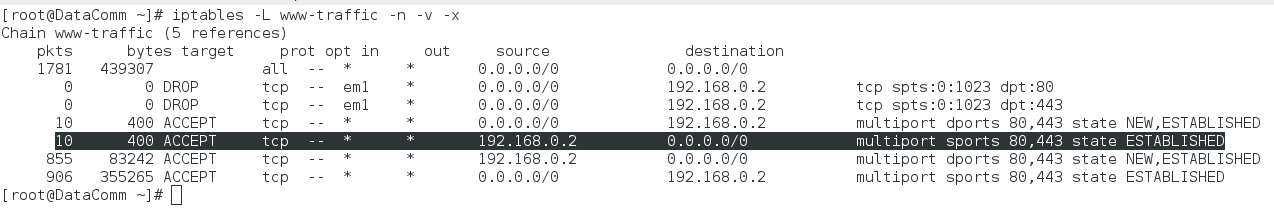


(18) WWW(443) outgoing traffic is accounted for

In our firewalled terminal, we ran this command:

hping3 192.168.0.11 -S -p 443 -c 5

Here is our accounting table:



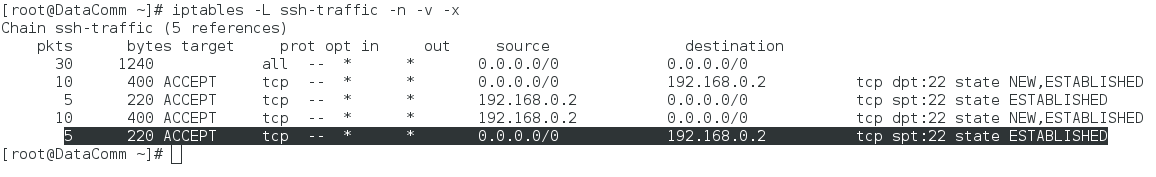
NOTE the two different highlighted lines: the former depicts incoming while the latter depicts outgoing. Also note the increase of packets from 5 to 10.

(19) SSH(22) incoming traffic is accounted for

In our non-firewalled terminal, we ran this command:

hping3 192.168.0.2 -S -p 22 -c 5

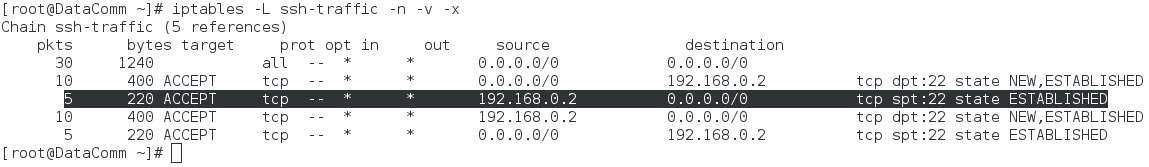
Here is our accounting table:



(20) SSH(22) outgoing traffic is accounted for

In our firewalled terminal, we ran this command:

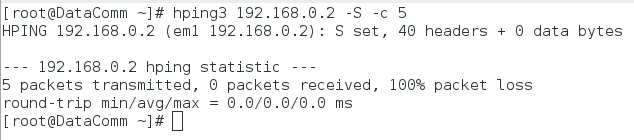
hping3 192.168.0.11 -S -p 22 -c 5

Here is our accounting table:  


NOTE the two different highlighted lines: the former depicts incoming while the latter depicts outgoing.

(21) All incoming SYN packets are dropped unless permitted

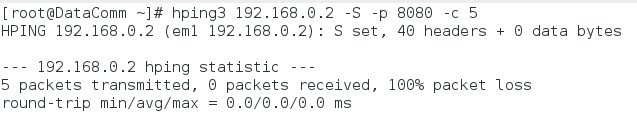
In our non-firewalled terminal, we ran this command with the following results:



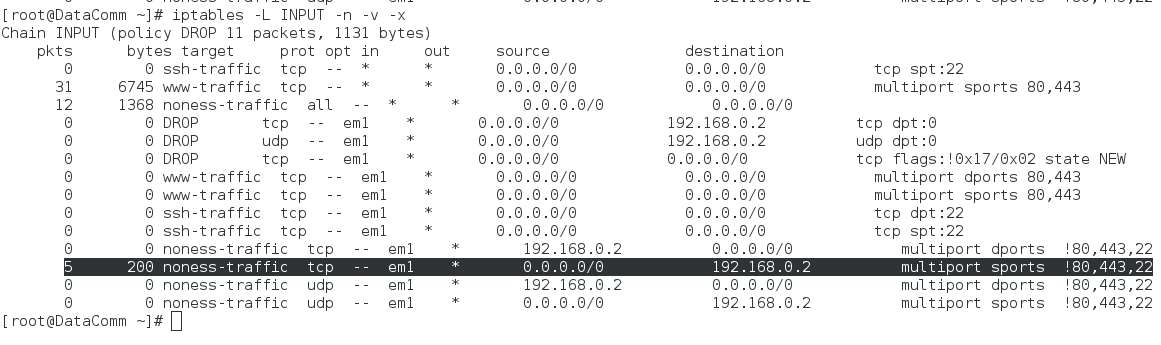
NOTE: There is no -p switch. We assume that SYN packets that are allowed are specified in previous test cases. Any other unspecified TCP packets coming in are therefore dropped because they are not explicitly permitted.

(22) All other undefined ports with inbound traffic are accounted for (TCP)

We ran hping3 with an undefined port of 8080:



We expect that it will drop these packets but we will see that it is accounted for in our tables:

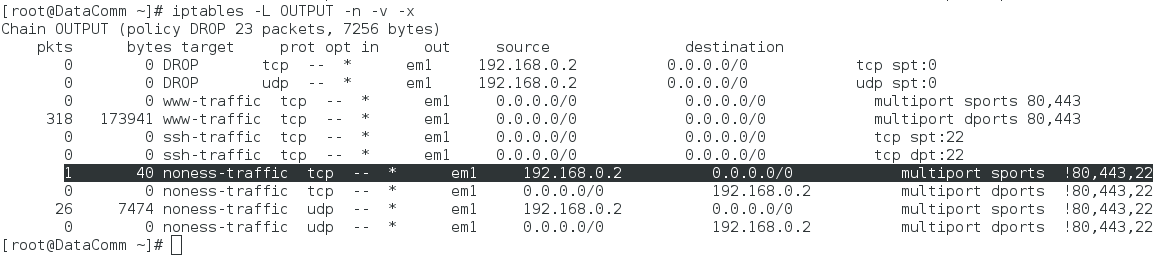


(23) All other undefined ports with outbound traffic are accounted for (TCP)

We ran hping3 with an undefined port of 8080 from our firewalled terminal:



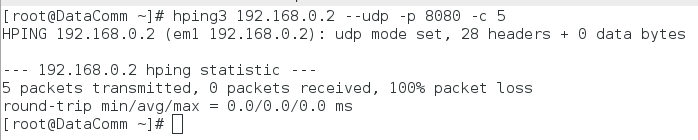
We expect that it will drop these packets because we should be a “good Internet neighbour”. Therefore, our firewall restricts us from being a bad neighbour. However, at least one instance of our “probing” must go through. It is recorded in our tables here:



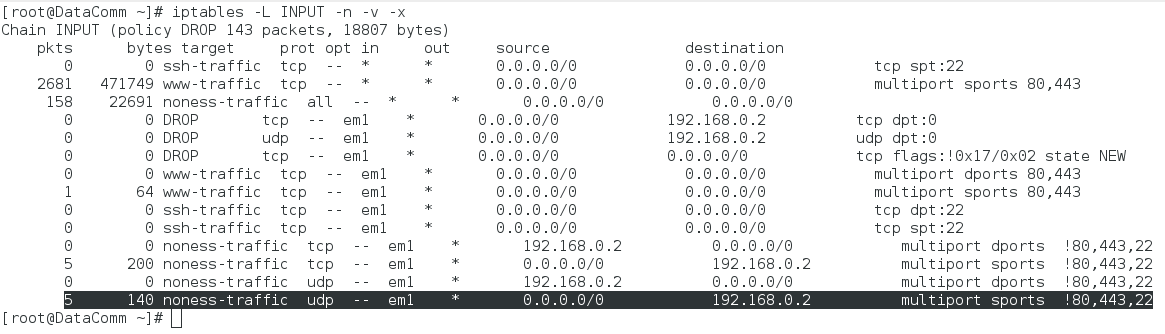
Note the one instance of outbound traffic.

(24) All other undefined ports with inbound traffic are accounted for (UDP)

We ran hping3 with an undefined port of 8080 with the --udp switch from our non-firewalled terminal:

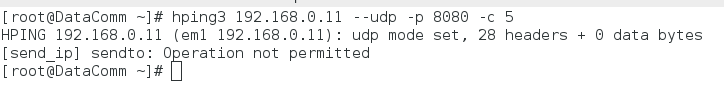


We expect that it will drop these packets but we will see that it is accounted for in our tables:

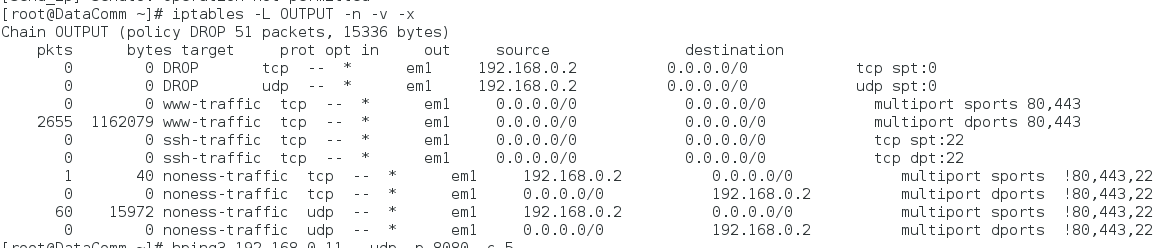


(25) All other undefined ports with outbound traffic are accounted for (UDP)

We ran hping3 with an undefined port of 8080 with the --udp switch from our firewalled terminal:

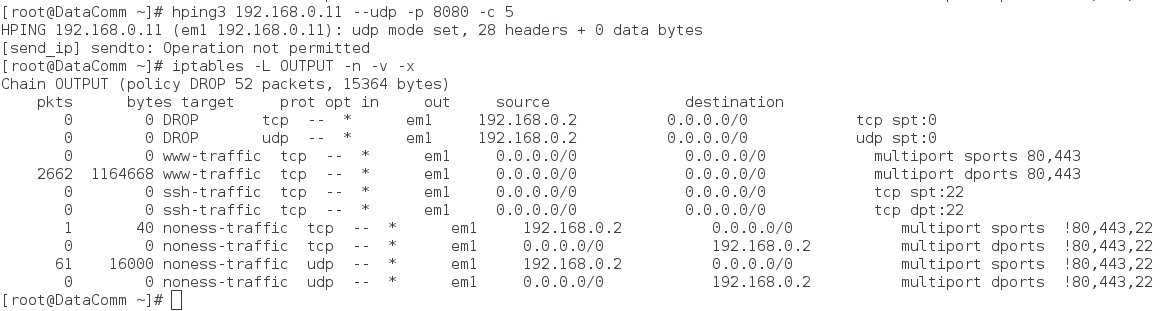


We expect this to happen because we are not being a “good Internet neighbour”. Thus, our firewall blocks our attempt to probe. However, because our UDP rules are universal, we will have to check a before and after screenshot of our tables:



Note the 60…

After running another instance of our hping3 command with the same switches, and then re-running our command to display our accounting tables, note the increment in outbound UDP packets:



Similar to our TCP instance, the firewall must capture at least one before hping3 gets terminated after one instance of probing.

# Conclusion

Through the extensive test cases of this assignment, it is sufficient to say that our firewall implementation has covered the relevant criteria of Assignment 1. They are the following:

* Set the default policies to DROP.
* Permit inbound and outbound ssh packets.
* Permit inbound and 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.
* Create a set of user-defined chains that will implement accounting rules to keep track of
* www, ssh traffic, versus the rest of the traffic on your system.
* Use Netfilter for your firewall implementation.
* Ensure the the firewall drops all inbound SYN packets, unless there is a rule that permits inbound traffic.
* Remember to allow DNS and DHCP traffic

We have also included the functionality to check WWW port 443 (https) in our rule set as well.

# 

# 

# Appendix

Located on disk are the following:

* Personal Firewalls - An Introduction to Firewall Administration (.pdf)
* Personal Firewalls - Design and Preliminary Testing (.pdf)
* firewall\_1.sh
* README.txt