# Labs for iptables

## Command Help

### Using ping and tracepath

It can be handy to ensure basic network connectivity is in place before assuming a firewall problem. When working on a firewall rule, always be sure to use ping and tracepath to ensure the endpoints and all hops between can be reached.

### Using netcat

Instead of running real network services in the environment, we will use netcat to simulate listening services. This will give greater flexibility to testing a variety of Layer 4 rules to be implemented in this lab environment.

To emulate a TCP service:   
nc –l <port>

To emulate a TCP client:   
nc <ip address> <port>

To emulate a UDP service:   
nc –u –l <port>

To emulate a UDP client:   
nc –u <ip address> <port>

Note that the <ip address> given to the client should be the IP address of the endpoint where the service in question is being emulated.

### Using tcpdump

When troubleshooting firewall rules, it is often EXTREMELY helpful to run tcpdump commands on both firewall interfaces, concurrently. Tcpdump will provide a view of the traffic reaching the firewall, and the traffic successfully passing through the firewall.

Basic sniffing:   
tcpdump -c 100 -i <interface> -n port <port>

Sniffing with payload capture:   
tcpdump –c 100 –i <interface> -n –s 0 –A port <port>

### Using the rc.iptables script

To load the firewall base rules:   
/etc/init.d/rc.iptables start

To unload the firewall base rules:   
/etc/init.d/rc.iptables stop

To recycle the firewall base rules after making changing to iptables:   
/etc/init.d/rc.iptables stop

### Using iptables

The core firewall in the Linux kernel is iptables. The rc.iptables script loads or unloads the base firewall rules in the kernel. The base rules are listed within the function within the script.

To view all existing rules:   
iptables --list –v –n –list-numbers

iptables -nvL

To flush all existing rules:   
iptables --flush

# Exercise 1 – Validating Basic Connectivity

Disable the firewall rules. Ensure that the firewall is working in its most basic capacity as a routing device.

1. Attempt to ping each endpoint from the opposite workstation.
2. Use tracepath from each to show the route in each workstation.

# Exercise 2 - Enabling the Firewall

Enable the firewall rules.

1. How did you check the status of the firewall? How did you enable the firewall? How would you stop the firewall?
2. What type of firewall is this? Does this firewall utilize whitelisting or blacklisting?
3. How would you change the firewall to the other type? Use the appropriate commands and test with actual traffic that would otherwise have been blocked.

# Exercise 3 - Verifying the Existing Firewall Rules

One rule is included in the rc.iptables.sh script on the firewall to allow web access from one workstation to the other.

1. Which host can connect to the other’s web server?
2. Prove your answer by connecting from the client to the server.

# Exercise 4 - Allowing SSH Access Between Workstations

Create a rule to allow SSH access from the southclient to the northclient.

1. What rule(s) did you create?
2. Simulate SSH traffic and use the iptables command to show that the traffic is being processed by that rule.

# Exercise 5 - Allowing Selected SSH Access to the Firewall

Create a rule to allow SSH from the southclient to the firewall, but not from the northclient to the firewall.

1. What rule(s) did you create?
2. Simulate the traffic and use the iptables command to show that the traffic is being processed by that rule.

# Exercise 6 - Allowing DNS Access Between the Workstations

Create a rule to allow access to DNS over UDP from the northclient to the southclient.

1. What rule(s) did you create?
2. Simulate DNS traffic and use the iptables command to show that the traffic is being processed by that rule.

# Exercise 7 - Make the Firewall Rules Stateful

Change the original rules so to use iptables mechanisms for stateful traffic control.

1. Show that the rules still work by simulating the traffic that was allowed by the original rules.
2. Repeat for the rules done in #4 and #6.