Information and Networking Security
Term: Spring 2025

sLaboratory Exercise 5-1 – Hands-on with Firewall Configuration

1. Overview

This individual laboratory exercise will provide some hands-on experience with firewall configuration and intrusion detection.

2. Resources required

This exercise requires the Ubuntu with Snort and Other Tools exercise in the Cyber Range.

3. Initial Setup

Log in to your Cyber Range account and select the Exercise Environment “Ubuntu with Snort and Other Tools”.

Click “start” to start your environment. When the systems are ready, click the “join” button and select the Primary Machine (desktop.example.com) from the dropdown menu to get to your Linux desktop login. If you are asked to log in, credentials for this Ubuntu Linux VM are below.

Username: **student**
Password: **student**

Note that you will need to use the Linux **sudo** utility to execute many of the commands in the following sections. The student account has **sudo** access using the password above. For more information on the Linux **sudo** command, type “**man sudo**” in a terminal window on your Linux VM.

Task: Firewall Configuration

[Note: be careful with firewall rule configuration changes on your Cyber Range virtual machine. If you set the rules improperly you could break your network connection to the range VM. Fortunately, this can almost always be fixed by restarting your virtual environment. To do this, close the browser tab displaying your Ubuntu VM desktop, go to the Cyber Range tab showing the *Ubuntu with Snort* environment description and select the “Stop” button, then restart the environment and re-join.]

Use the following command to set the host-based firewall on your Linux system to a default policy that we have specified:

```
$ sudo /etc/default_firewall.sh
```

[You may have to enter your student password: ***student***]

Linux host-based firewalls are configured using the **iptables** command. There is a pretty good (and short) tutorial at http://fideloper.com/iptables-tutorial. To review the firewall rules set by the default policy, use the following command:

```
$ sudo iptables -L -n -v
```

Simple packet filtering firewalls usually have a default policy to DROP packets and only to accept traffic that meets specific criteria. When a packet arrives on a host, the firewall tries to match firewall rules starting with the first rule in the chain. The firewall will apply the first rule that matches and the default rule is applied last, so if there is a rule that “ACCEPTs” a packet before the default DROP, the packet will be accepted. In general, if a specific input or output IP address, port, or protocol is not specified, the rule applies to 'any' IP address, port, or protocol.

Review the default firewall configuration (`$ sudo iptables –L –n -v`) and answer these questions.

1. **What is the *default* policy on the INPUT, OUTPUT, and FORWARD chains after you apply the `default_firewall.sh` configuration?**

   ANS:

   INPUT: DROP (the default policy on INPUT chain is DROP)
   FORWARD: ACCEPT (the default policy in FORWARD chain is to ACCEPT)
   OUTPUT: ACCEPT (the default policy on OUTPUT chain is to ACCEPT.)

2. **What firewall rules are in place on the INPUT chain? Specify protocols and ports for which packets are allowed by the rules provided, and under what conditions those packets are allowed.**

   ANS:
   ICMP protocol is allowed from any source to any destination with no specific conditions.

   UDP protocol is allowed from any source to any destination with no specific conditions.

   TCP protocol on port 22 (SSH) via Ethernet (eth0) is allowed from any source to any destination, condition: state NEW or ESTABLISHED.

   TCP protocol on port 3389 (RDP) via Ethernet (eth0) is allowed from any source to any destination, condition: state NEW or ESTABLISHED.

   All protocols on the loopback interface (lo) are allowed from any source to any destination with no specific conditions.

3. **What firewall rules are in place on the OUTPUT chain? Specify protocols and ports for which packets are allowed by the rules provided, and under what conditions those packets are allowed.**

Protocol	**Interface**	**Ports**	**Condition**	**Action**
TCP	**Eth0**	**22(SSH)**	**State Established**	**ACCEPT**

TCP	Eth0	3389 (RDP)	State Established	ACCEPT
ANY	Lo (LOOPBACK)	N/A	FROM/TO LOCAL	ACCEPT

TCP protocol on port 22 is allowed via eth0 in the ESTABLISHED state.

TCP protocol on port 3389 is allowed via eth0 in the ESTABLISHED state.

All traffic on the loopback interface (lo) is allowed unconditionally.

Part 2. Configure firewall

4. **What is the difference between a stateful and a stateless firewall? Is the Linux iptables utility stateful or stateless?**

ANS: A Stateful firewall keeps track of the state active connections, automatically allows return traffic, which enhances security and make it more efficient in managing networks traffic. On the other hand, a stateless firewall handles each packets individually without any awareness of previous interactions, requiring manual rules for return traffic.
Stateless firewalls are faster, they offer less security ,iptables utility operates as a stateful firewall.`

We will use two shell scripts to modify the firewall configuration. A script called '/etc/extingui.sh' will clear all firewall rules and set the default policy on the INPUT, OUTPUT, and FORWARD chains to ALLOW all traffic in and out of your server. Execute this script as follows.

```
$ sudo /etc/extingui.sh
```

[Enter your student password if required.]

Once the firewall rules are cleared, you will modify the script '/home/student/lab2/firewall.sh' to add firewall configuration commands. Use the text editor of your choice to edit this script (one option is “mousepad”, available at *Applications-->Accessories-->mousepad*).

Recall that iptables commands are of the following form:

```
iptables [command-type] [pattern-match options] -j [action]
```

Where `[command-type]` specifies whether the rule will be added or deleted on a specified chain, `[pattern-match-options]` specifies the port, interface, address, etc. to match, and `[action]` specifies what action to take if the packet matches the pattern (DROP, REJECT, ACCEPT, LOG).

In our simple packet filtering firewall, all of our rules will be added to the INPUT or OUTPUT chains and our actions will either be ACCEPT or DROP; in this exercise, all of your rules will be of the form:

```
iptables -A INPUT  [pattern-match options]  -j [ACCEPT or DROP]
```

Pattern match options that you will use include:

-s	source IP address or address range (can use CIDR addressing)
-d	destination IP address or address range
-p	transport layer protocol (tcp, udp, or icmp)
-m	match a specific property (such as ‘state’)
--dport	destination port number (must be used with a protocol specified by the -p option)
--sport	source port number (must be used with a protocol specified by the -p option)
--state	connection state (NEW, ESTABLISHED, etc.)

An example rule using the above options is here:

```
# Allow inbound packets to TCP port 20 from subnet 192.168.1.0/24
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 20 -j ACCEPT
```

There are more examples in the /home/student/lab2/firewall.sh script described below.

Add rules to your /home/student/lab2/firewall.sh script that will allow outbound connection attempts on port 80 and the return traffic. Once you have edited the firewall.sh file, save and apply at the command line as follows:

```
$ sudo /home/student/lab2/firewall.sh
```

You can test this configuration by trying to browse to one of the below http-only websites. You should delete your browsing history between tests so you don’t get cached results.

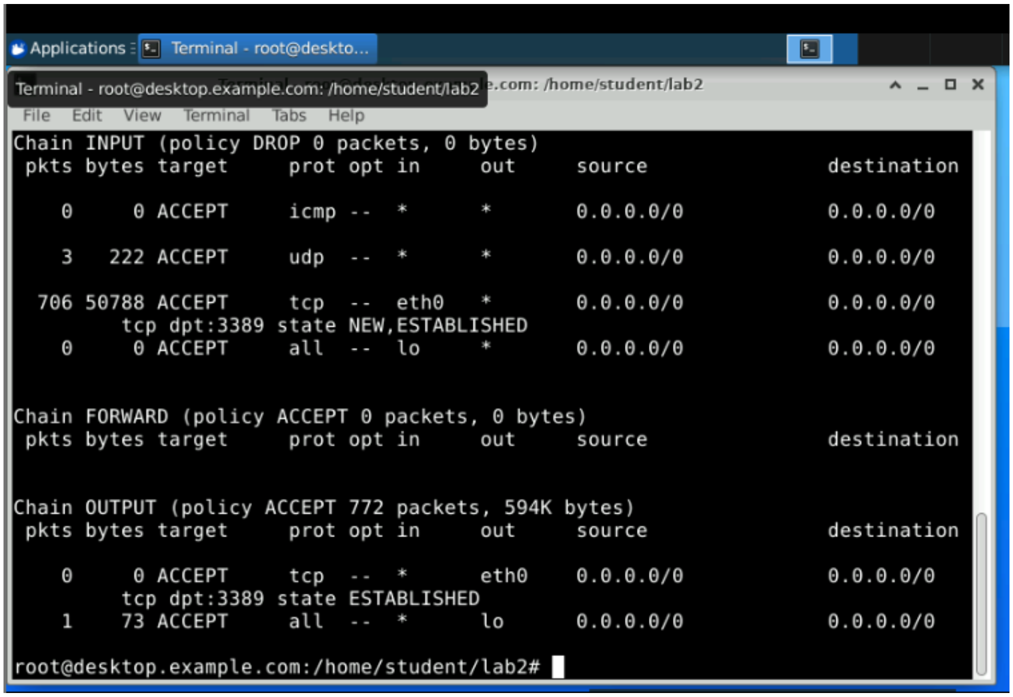
- http://terminal.example.com/ - this site is inside your cyber range environment.
- http://httpforever.com/ - this is an external website; access will go through the cyber range web proxy.

Be specific with your rules. Your score will reflect rules that are too general (for example, allows all inbound and outbound traffic).

5. **List the rule(s) that you added to the firewall to allow outbound HTTP requests (port 80) and responses.**

Ans: Allow outbound HTTP requests on port 80
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW, ESTABLISHED -j ACCEPT

Allow incoming response traffic for HTTP requests
iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT


Applications
Terminal - root@deskto...
Terminal - root@desktop.example.com: /home/student/lab2
.com: /home/student/lab2
File Edit View Terminal Tabs Help

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

    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0

    3   222 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0

  706 50788 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         tcp dpt:3389 state NEW,ESTABLISHED
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0


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


Chain OUTPUT (policy ACCEPT 772 packets, 594K bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  *      eth0    0.0.0.0/0            0.0.0.0/0
         tcp dpt:3389 state ESTABLISHED
    1    73 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0

root@desktop.example.com:/home/student/lab2#
```

5. References

There is a good iptables tutorial at http://fideloper.com/iptables-tutorial.