Configuring a Linux Firewall with iptables

Scenario

You will configure a basic firewall rule on a Linux server to block port 80 (HTTP) traffic. Next, you will reconfigure the server to accept HTTP connections, and then you will configure iptables logging for all traffic. Finally, you will display log file traffic.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objectives:

- 3.3 Given a scenario, implement secure network designs.

Lab

- Kali VM
- LX1 VM
- pfSense VM

Task 1

Configure a Linux iptables firewall for HTTP connections

You are tasked with securing a new web server. You need to verify that connectivity exists, and then insert a rule into the iptables firewall that blocks inbound port 80 traffic.

1. Sign in to the **Kali** VM as **kali** using **Pa$$w0rd** as the password.

2. Open a **Terminal** using the menu at the top of the screen.

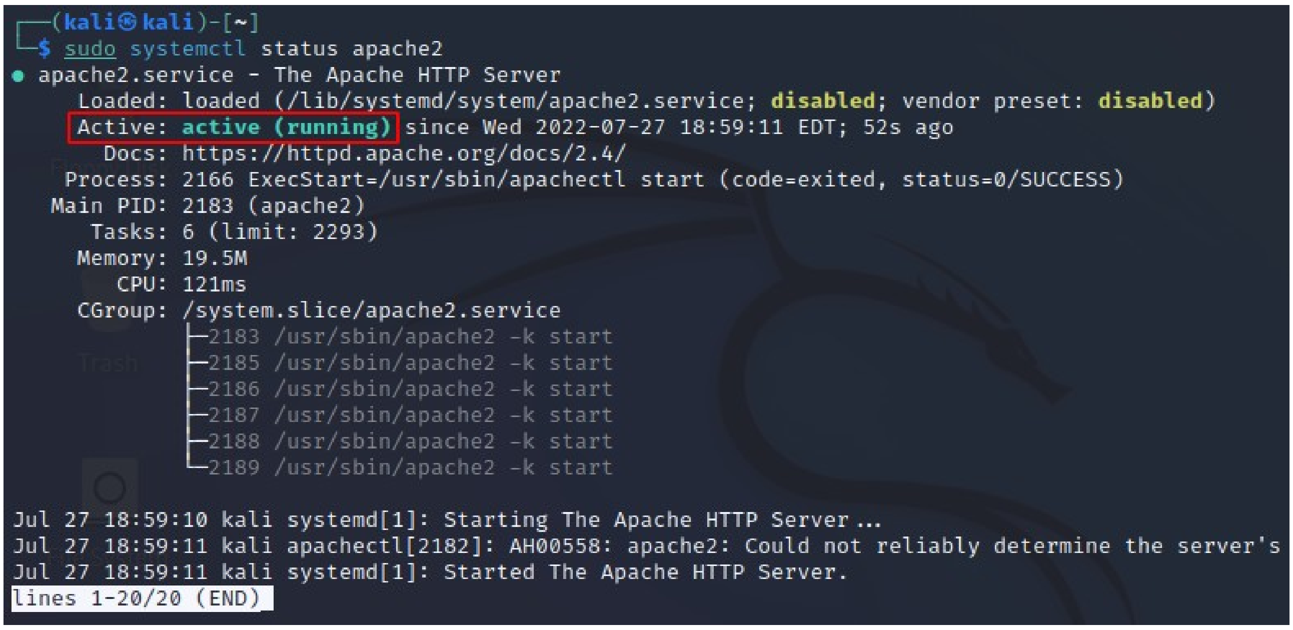
3. Run the following command to start the Apache web service:

```
sudo systemctl start apache2
```

TIP: The service is now started, but it is not persistent. If the server reboots, the service will not automatically start. You would enter s**udo systemctl enable apache2** to make the service persistent.

4. Run the following command to verify that Apache is running:

```
sudo systemctl status apache2
```



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
     Active: active (running) since Wed 2022-07-27 18:59:11 EDT; 52s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2166 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2183 (apache2)
      Tasks: 6 (limit: 2293)
     Memory: 19.5M
        CPU: 121ms
     CGroup: /system.slice/apache2.service
             ├─2183 /usr/sbin/apache2 -k start
             ├─2185 /usr/sbin/apache2 -k start
             ├─2186 /usr/sbin/apache2 -k start
             ├─2187 /usr/sbin/apache2 -k start
             ├─2188 /usr/sbin/apache2 -k start
             └─2189 /usr/sbin/apache2 -k start

Jul 27 18:59:10 kali systemd[1]: Starting The Apache HTTP Server...
Jul 27 18:59:11 kali apachectl[2182]: AH00558: apache2: Could not reliably determine the server's
Jul 27 18:59:11 kali systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

Figure 1.1 - Confirm the status of the apache2 service on the virtual machine.

TIP: Press **q** to return to the terminal prompt.

5. Switch to the **LX1** VM. This virtual machine will act as the HTTP client for this task.

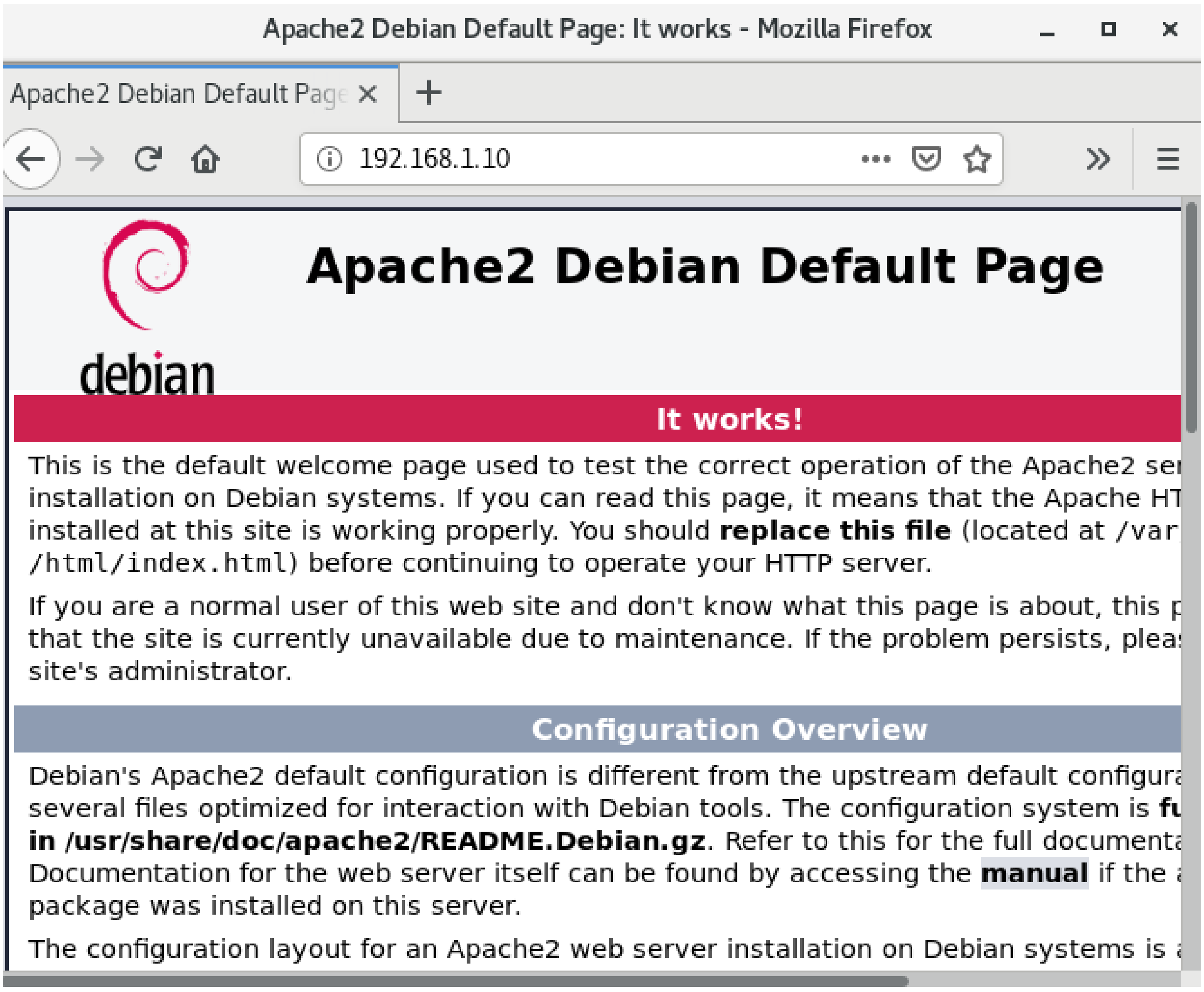
6. Sign on with the default user account of **User**, using **Pa$$w0rd** as the password.

7. From the **Applications** menu, select **Firefox**.

8. In the Firefox address bar, enter **http://192.168.1.10**

NOTE: The Kali Linux virtual machine IP address is assigned via DHCP. Check the Kali ip address with **ifconfig** to connect using the web browser.

The connection attempt succeeds. Iptables currently accepts all inbound connection attempts.


Apache2 Debian Default Page: It works - Mozilla Firefox

192.168.1.10

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 se
installation on Debian systems. If you can read this page, it means that the Apache HT
installed at this site is working properly. You should **replace this file** (located at /var
/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this p
that the site is currently unavailable due to maintenance. If the problem persists, plea
site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configura
several files optimized for interaction with Debian tools. The configuration system is **fu**
in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documenta
Documentation for the web server itself can be found by accessing the **manual** if the a
package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is a

Figure 1.2 – The default Apache web page on Kali VM.

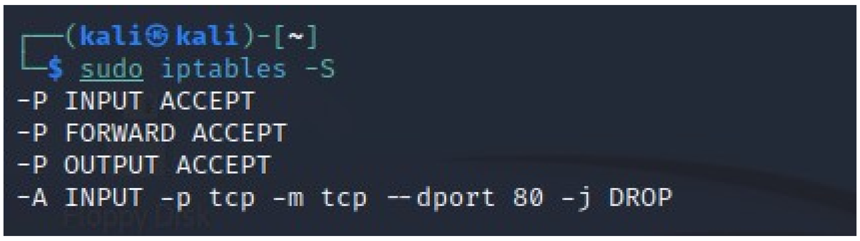
9. Close the **Firefox** web browser.

10. Switch to the **Kali** VM, and the configure the **iptables** service to DROP inbound HTTP connections by port number 80 by using the following command:

```
sudo iptables -I INPUT 1 -p tcp --destination-port 80 -j DROP
```

11. Display the iptables rules and observe that the HTTP service is specified by port number 80.

```
sudo iptables -S
```



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j DROP
```

Figure 1.3 – Displaying the iptables rules.

12. Switch to the **LX1** VM, launch **Firefox**, and then attempt to connect to the Kali **http://192.168.1.12** test site again.

NOTE: You might need to refresh the Firefox windows to test.

Firefox spends several seconds attempting to connect. Eventually, the connection attempt fails. The iptables firewall is blocking the connection.

Task 2

Display iptables log files

You will now reverse the firewall configuration, once again permitting inbound traffic over port 80. You will then configure iptables to log all inbound connections.

1. Select the **Kali** VM.

2. Insert a new iptables rule at line 1 so that connections for port 80 are accepted:

```
sudo iptables -I INPUT 1 -p tcp –destination-port 80 -j ACCEPT
```

TIP: The easiest way is to use the up arrow key to recall recent commands, and then backspace over DROP and type in ACCEPT.

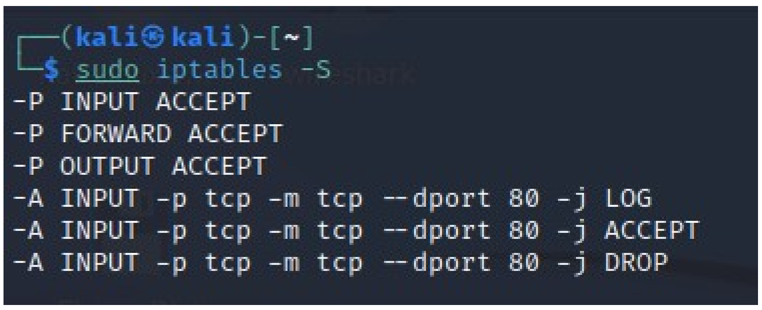
3. Enable iptables logging:

```
sudo iptables -I INPUT 1 -p tcp –destination-port 80 -j LOG
```

TIP: Observe that you are now placing the LOG rule at the top of the list. All traffic will be logged before being processed by the rules that follow after the LOG rule.

4. Display the iptables rules and observe that the destination port 80 ACCEPT and LOG rules are listed above the DROP rule. Firewall rules are processed in order.

```
sudo iptables -S
```



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j LOG
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j DROP
```

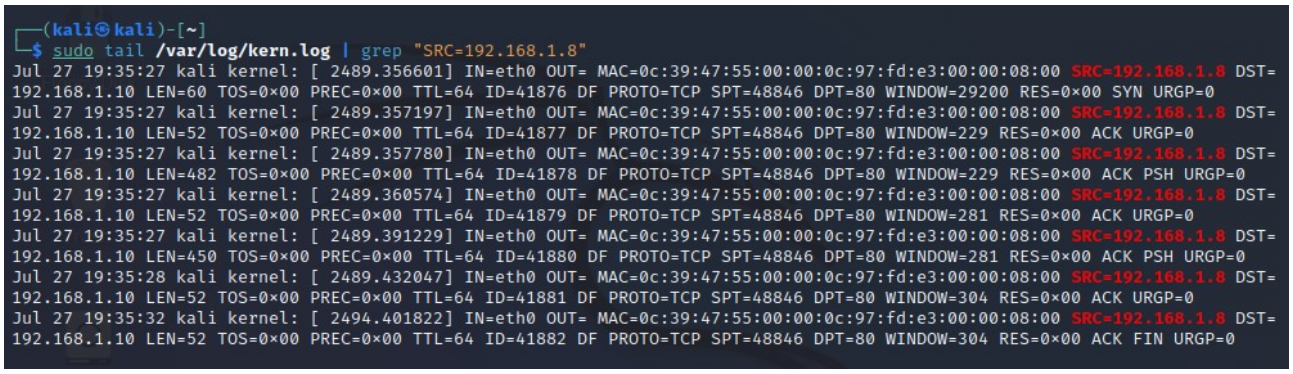
Figure 2.1 – Displaying the iptables rules.

5. Switch back to the **LX1** VM, and attempt or refresh the **http://192.168.1.12** failed web connection again with Firefox. You should be able connect again as before.

6. Switch to the **Kali** VM.

7. Display destination port 80 traffic in the **/var/log/kern.log** log file by using the **tail** command.

```
sudo tail /var/log/kern.log | grep “SRC=192.168.1.8”
```



```
┌──(kali㉿kali)-[~]
└─$ sudo tail /var/log/kern.log | grep "SRC=192.168.1.8"
Jul 27 19:35:27 kali kernel: [ 2489.356601] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=60 TOS=0×00 PREC=0×00 TTL=64 ID=41876 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=29200 RES=0×00 SYN URGP=0
Jul 27 19:35:27 kali kernel: [ 2489.357197] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=52 TOS=0×00 PREC=0×00 TTL=64 ID=41877 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=229 RES=0×00 ACK URGP=0
Jul 27 19:35:27 kali kernel: [ 2489.357780] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=482 TOS=0×00 PREC=0×00 TTL=64 ID=41878 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=229 RES=0×00 ACK PSH URGP=0
Jul 27 19:35:27 kali kernel: [ 2489.360574] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=52 TOS=0×00 PREC=0×00 TTL=64 ID=41879 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=281 RES=0×00 ACK URGP=0
Jul 27 19:35:27 kali kernel: [ 2489.391229] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=450 TOS=0×00 PREC=0×00 TTL=64 ID=41880 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=281 RES=0×00 ACK PSH URGP=0
Jul 27 19:35:28 kali kernel: [ 2489.432047] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=52 TOS=0×00 PREC=0×00 TTL=64 ID=41881 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=304 RES=0×00 ACK URGP=0
Jul 27 19:35:32 kali kernel: [ 2494.401822] IN=eth0 OUT= MAC=0c:39:47:55:00:00:0c:97:fd:e3:00:00:08:00 SRC=192.168.1.8 DST=
192.168.1.10 LEN=52 TOS=0×00 PREC=0×00 TTL=64 ID=41882 DF PROTO=TCP SPT=48846 DPT=80 WINDOW=304 RES=0×00 ACK FIN URGP=0
```

Figure 2.2 – Checking the kern.log file with tail.

8. Close the **Terminal** window typing:

```
exit
```