Configuring a Firewall

(CompTIA Security + SY – 601)

Objectives:

- To configure a basic firewall rule on a basic Linux server to block 80 (HTTP) traffic
- Reconfigure the server to accept HTTP connections
- Configure iptables logging for all traffic
- To display log file traffic

Resources:

- Kali Virtual Machine (PT1-Kali)
- Iptables tool
- CentOS Virtual Machine (LX1)

Instructions:

Configure a Linux iptables firewall for HTTP Connections

- Sign-in to **PT1-Kali** VM
- Open a terminal using the menu at the top of the screen
- Run the following command to start the Apache web services: **systemctl start apache2**

```
root@KALI:~# systemctl start apache2
root@KALI:~# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/s
ystemd/system/apache2.service.
```

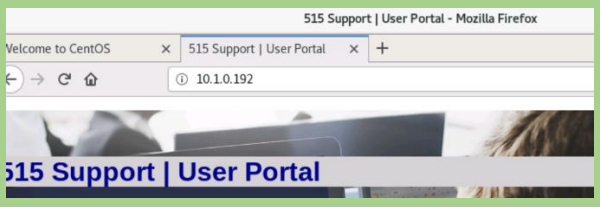
- Run the following command to verify that Apache is running: **systemctl status apache2**

```
root@KALI:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: d>
     Active: active (running) since Sun 2024-07-21 09:27:11 PDT; 1min 27s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1354 (apache2)
      Tasks: 6 (limit: 7058)
     Memory: 21.0M
     CGroup: /system.slice/apache2.service
             ├─1354 /usr/sbin/apache2 -k start
             ├─1355 /usr/sbin/apache2 -k start
             ├─1356 /usr/sbin/apache2 -k start
             ├─1357 /usr/sbin/apache2 -k start
             ├─1358 /usr/sbin/apache2 -k start
             └─1359 /usr/sbin/apache2 -k start

Jul 21 09:27:11 KALI systemd[1]: Starting The Apache HTTP Server...
Jul 21 09:27:11 KALI apachectl[1353]: AH00558: apache2: Could not reliably determin>
Jul 21 09:27:11 KALI systemd[1]: Started The Apache HTTP Server.
```

- Switch to the **LX1** VM
- Sign-on with the default centos account, using **Pa$$w0rd** as the password
- From the Applications menu, select **Firefox**

- In the Firefox address bar, enter **http://10.1.0.192** to connect to the **515 support** web site



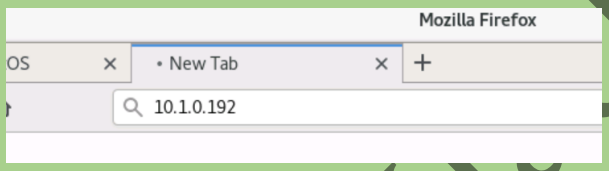
- Close Firefox
- Switch to the **PT1-Kali** VM and then configure the iptables service to **DROP** inbound **HTTP** connection by port number 80

```
root@KALI:~# iptables -I INPUT 1 -p tcp --destination-port 80 -j  DROP
```

- Display the iptables rules and observe that the HTTP service is specified by port number 80 **iptables -S**
- Run the following command to redirect the output of the iptables -S command to a text file for scoring: **iptables -S > ~/iptables.txt**

```
root@KALI:~# iptables -S > ~/iptables.txt
```

- Switch to **LX1** VM , launch Firefox then attempt to connect to the **http://10.1.0.192** kali site test again



Display iptables log files

- Select the **PT1-Kali** VM
- Insert a new iptables rule at the line 1 so that the connections for port 80 are accepted

```
root@KALI:~# iptables -I INPUT 1 -p tcp --destination-port 80 -j  DROP
```

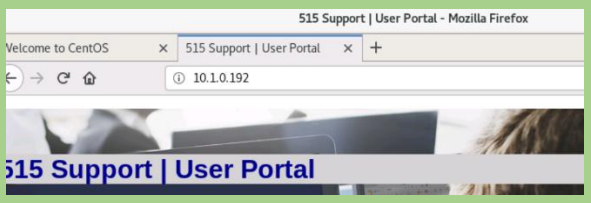
- Enable iptables logging

```
root@KALI:~# iptables -I INPUT 1 -j LOG
```

- 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

```
root@KALI:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -j LOG
-A INPUT -p tcp -m tcp --dport 80 -j DROP
-A INPUT -p tcp -m tcp --dport 80 -j DROP
```

- Switch back to the **LX1 VM** and attempt to refresh the **http://10.1.0.192** web connection again with Firefox



- Switch to the **PT1-Kali** VM
- Display destination port 80 traffic in the **/var/log/kern.log** by using the **tall** command

```
root@KALI:~# tail /var/log/kern.log | grep "80"
Jul 21 10:18:38 KALI kernel: [ 3191.148398] IN=eth0 OUT= MAC=00:15:5d:01:ca:4a:00:15
:5d:01:ca:55:08:00 SRC=10.1.0.10 DST=10.1.0.192 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=
23319 DF PROTO=TCP SPT=46602 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
Jul 21 10:18:46 KALI kernel: [ 3198.908638] IN=eth0 OUT= MAC=00:15:5d:01:ca:4a:00:15
:5d:01:ca:55:08:00 SRC=10.1.0.10 DST=10.1.0.192 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=
48420 DF PROTO=TCP SPT=46600 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
Jul 21 10:18:46 KALI kernel: [ 3199.164811] IN=eth0 OUT= MAC=00:15:5d:01:ca:4a:00:15
:5d:01:ca:55:08:00 SRC=10.1.0.10 DST=10.1.0.192 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=
23320 DF PROTO=TCP SPT=46602 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
Jul 21 10:19:02 KALI kernel: [ 3214.941847] IN=eth0 OUT= MAC=00:15:5d:01:ca:4a:00:15
:5d:01:ca:55:08:00 SRC=10.1.0.10 DST=10.1.0.192 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=
48421 DF PROTO=TCP SPT=46600 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
```

Observations:

- Apache web services were successfully started and verified.
- HTTP connections were initially blocked by iptables.
- Reconfiguration allowed HTTP connections and enabled logging.
- Log files displayed HTTP traffic as expected.

Results:

- Successfully configured firewall rules to block and then allow HTTP traffic.
- Enabled and verified iptables logging for HTTP connections.

Conclusion:

This lab demonstrated the configuration and management of firewall rules using iptables on a Linux server. By blocking and then allowing HTTP traffic, and enabling logging, the exercise highlighted key firewall management techniques essential for securing network traffic.

Future Work:

- Automate iptables rule management with scripts.
- Integrate advanced firewall rules for specific IP ranges.
- Implement comprehensive logging and monitoring solutions for network traffic analysis.