Lab Sheet 6

Security

Our sixth lab is about firewalls. The Linux kernel embeds the *netfilter* firewall. There is no turn-key solution for configuring any firewall since network and user requirements differ. However, you can control *netfilter* from user space with the **iptables** and **ip6tables** commands. The difference between these two commands is that the former works for IPv4 networks, whereas the latter works on IPv6. Since both network protocol stacks will probably be around for many years, both tools will need to be used in parallel. You can also use the excellent GUI-based **fwbuilder** tool, which provides a graphical representation of the filtering rules.

1 Task 1: Identify all open ports

Check the open ports:

```
root@kali:~# netstat -tulpen
root@kali:~# iptables -n -L INPUT
```

If you have ports you blocked, or previous iptables rules, you can drop them all:

```
root@kali:~# iptables -F INPUT
root@kali:~# iptables -P INPUT ACCEPT
root@kali:~# iptables -P FORWARD ACCEPT
root@kali:~# iptables -P OUTPUT ACCEPT
```

Now check to see if you can connect to port 4444 on your machine by running netcat in the following way. Note that in this exercise, your IP addresses will, of course differ:

```
root@kali:~# nc -lnvp 4444
listening on [any] 4444 ...
```

From your host machine, or another machine, try to connect to the listening netcat instance. Once connected, type some characters, and they should appear on the Kali VM nc listener:

```
root@HOST_MACHINE:~#  nc -v 172.16.161.136 4444
aaaaaaaa
```

Note: If you do not see the characters you typed in your Kali nc listener, there's a problem. Get that resolved before you continue. If in a VM, switch to bridged networking instead of NAT, etc until this nc example works.

2 Task 2: Allow inbound TCP connections on ports 22, 80, and 443 only

Configure the firewall with commands similar to the following:

```
iptables –P INPUT DROP
iptables –A INPUT –i lo –j ACCEPT
iptables –A INPUT –m state ––state ESTABLISHED,RELATED –j ACCEPT
iptables –A INPUT –m state ––state NEW –p tcp ––dport 22 –j ACCEPT
iptables –A INPUT –m state ––state NEW –p tcp ––dport 80 –j ACCEPT
iptables –A INPUT –m state ––state NEW –p tcp ––dport 443 –j ACCEPT
```

Now check to see if you can connect to port 4444 on the firewalled machine by running netcat in the following way:

```
root@kali:~# nc –lvp 4444
listening on [any] 4444 ...
```

3 Task 3: Verify other ports are blocked with netcat

From your host machine, try to connect to the listening netcat instance. It should fail:

```
root@HOST_MACHINE:~# nc –v 172.16.161.136 4444
nc: connectx to 172.16.161.136 port 4444 (tcp) failed: Operation timed out
```

4 Task 4: Make sure rules persist after reboot

Now, create an iptables script from these rules:

```
root@kali:~# iptables–save > /usr/local/etc/myconfig.fw
```

And register the configuration script in a pre-up directive of the /etc/network/interfaces file. Reboot to see if the rules persist!

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
pre–up iptables–restore < /usr/local/etc/myconfig.fw
```