Setting Up a Firewall

1. View current firewall configuration.

Text

Description automatically generated

* 1. **iptables** is used to manage the kernel’s net filter firewall. It can be used to configure packet filtering, NAT, and other packet mangling. It can also be used to create rules to accept, reject, or drop traffic based on source and destination IP addresses, ports, protocols, and more. The **-L** option displays the IP packet filtering rules for a given chain. If no chain is given, it displays the rules for all chains. Currently, there are no rules for INPUT, FORWARD, or OUTPUT. The output above shows our current firewall configuration. This configuration allows all incoming and outgoing traffic.

1. Define firewall policy. Determine what traffic is allowed to pass through the firewall and what traffic should be blocked.
   1. For incoming traffic, we typically only use port port 22/tcp (SSH). But to be safe, we will also leave port 80/tcp (HTTP) and port 443/tcp (HTTPS) open. Everything else will be blocked. As far as forwarding traffic, everything is blocked. And for outgoing traffic, everything is accepted. So, we are able to send anything out.
2. Create your firewall rules: Use iptables to create the rules for your firewall. You can add or remove rules as needed. To do this, I created a bash script. To create the script, I did **vi configureFirewall.sh**. To be able to enter text into the file you must enter INSERT mode. This can be done by pressing the INSERT button. After that, you can start typing. On the first line of the script I added **#!/bin/bash**, this is to make the Bash shell be used to interpret the script. That is followed by a comment that explains what the script does.

Text

Description automatically generated

Text

Description automatically generated

* 1. These rules are a set of firewall rules that are being added to the machine to provide some basic level of network security. The first rule sets the default incoming traffic policy to DROP. So, all incoming traffic will be automatically dropped unless it is specified by a rule in the chain. The next three rules allow incoming SSH, HTTP, and HTTPS traffic through ports 22(tcp), 80(tcp), and 443(tcp) respectively. The next rule sets the default forward traffic policy to DROP. So, any traffic that is forwarded through the machine will be dropped. The next rule sets the default outgoing traffic policy to ACCEPT. So, all outgoing traffic will be automatically accepted. The last three rules allow incoming SSH, HTTP, and HTTPS traffic through ports 22(tcp), 80(tcp), and 443(tcp) respectively. While this may be redundant, this is simply a safety measure since those 3 ports are most important.

1. Test your firewall: Once you have created your firewall rules, you should test them to make sure they are working as expected. You can use a tool like **nmap** to scan your virtual machine and see which ports are open.

Text

Description automatically generated

* 1. The scan shows that the host is up, and **nmap** has identified three open/closed ports. Specifically, port 22/tcp (SSH) is open while ports 80/tcp (HTTP) and 443/tcp (HTTPS) are closed. The output also shows that 997 ports have been filtered, which means that **nmap** could not determine whether they are open or closed. **nmap** couldn’t determine whether they are open or closed because our firewall configuration rules blocked them. So, **nmap** couldn’t identify them.

1. Save your firewall rules: Finally, you should save your firewall rules so that they are consistent across reboots. The exact process for doing this may vary depending on your Linux distribution, but in general you can use the following commands:

A screenshot of a computer

Description automatically generated with medium confidence

* 1. This will save the iptables and ip6tables rules specified above to their respective files.

A screenshot of a computer

Description automatically generated with medium confidence

This will apply those rules to boot.

1. Lastly, you need to save these changes to the script file. This can be done by pressing **ESC** to exit insert mode. After exiting, you can enter **SHIFT + Z + Z**. to save the changes and exit the file. Next, you need to give yourself permission to run the file. This can be done by entering **chmod + x [name of file]**. After that, you run the command by entering **./[name of file]**.