Matt Favazza

Firewalls Ubuntu

Here is the IP Table for open ports on the Ubuntu Server

Text

Description automatically generated

It is empty because there currently are no rules until we enable the firewall. Type **sudo ufw enable**

Graphical user interface, text

Description automatically generated

Here is a sample of the iptables now that the firewall was enabled. See iptablesUbuntuBefore.txt

Chain INPUT (policy DROP)

target prot opt source destination

ufw-before-logging-input all -- anywhere anywhere

ufw-before-input all -- anywhere anywhere

ufw-after-input all -- anywhere anywhere

ufw-after-logging-input all -- anywhere anywhere

ufw-reject-input all -- anywhere anywhere

ufw-track-input all -- anywhere anywhere

Chain FORWARD (policy DROP)

target prot opt source destination

ufw-before-logging-forward all -- anywhere anywhere

ufw-before-forward all -- anywhere anywhere

ufw-after-forward all -- anywhere anywhere

1. To allow web services we need to open ports 80 and 443. We also need to port forward 80 to port 8080

First we need to open the ports. Do this by typing **sudo ufw allow 80** and **sudo ufw allow 443.** To test to see if the ports are open you can type **sudo ufw status** to see the ports open and the status of the firewall.

Text

Description automatically generated

Now we need to port forward port 80 to port 8080. To do this we need to enable port 8080 with **sudo ufw allow 8080**. Then we need to change a configuration file located at **/etc/ufw/before.rules**.

Type **sudo nano /etc/ufw/before.rules** to modify the file and add these following lines before the **filter** section.

\*nat

:PREROUTING ACCEPT [0:0]

-A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080

COMMIT

Text

Description automatically generated

After this has been saved, the ufw service needs to be restarted. **Sudo systemctl restart ufw**

1. Open ports for MySQL

MySQL has two different protocols for its services. MySQL protocol and New X Protocol. We will open the ports needed for client to server connections for both of the protocols. MySQL Protocol uses port 3306 and New X Protocol uses 33060

To open the ports type **sudo ufw allow 3306** and **sudo ufw allow 33060**

The ufw status page should look like this now

Graphical user interface, text

Description automatically generated

1. Dealing with SSH

For this I want to write two basic scripts that can be run to quickly allow or deny SSH. The port used for ssh is 22.

Within my SCRIPTS folder I am going to **touch** two files, one named SSHALLOW and SSHDENY

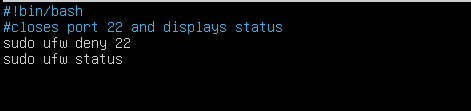
Text

Description automatically generated

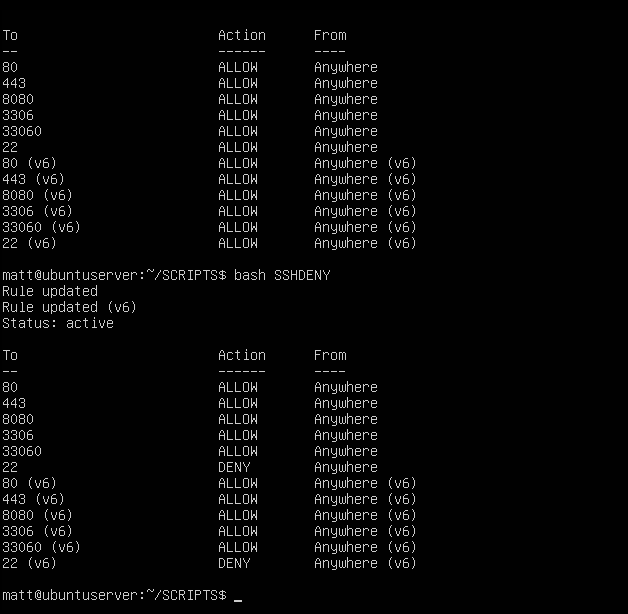
Using the nano editor I will modify each file to run the command necessary to deny or allow ssh. The commands are **sudo ufw allow 22** and **sudo ufw deny 22**

Text

Description automatically generated



Here is the output of both scripts



1. Dealing with Email services.

There are three different protocols for Email services. SMTP, POP3, and IMAP and each of those use different ports for encrypted or unencrypted traffic. We obviously want to use encrypted traffic for the security benefits. The ports we need to open are ~~465~~ 587, 995, and 993. Again, I’m going to create a simple script to allow and deny traffic on these ports.

\*After completing this part of the assignment, I realized 465 was not the correct port, for secured SMTP. The scripts have been changed accordingly

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Here is the output from the allow script and the deny script

Text

Description automatically generatedText

Description automatically generated

1. Block specific hosts and MAC addresses

To block hosts, we have two options. **Ufw deny** and **ufw reject**. These both have their uses, but in this case one is better. **Ufw reject** will deny connection attempts and send back a packet that the connection was refused. **Ufw deny** will drop the connection attempts and not send back any information. In this case where blocking denial of service attacks, we do not want any attackers knowing that we are rejecting their connection.

For the mac addresses, we need to use **IPTABLES** or edit the **/etc/ufw/before.rules** file. Since we are making a scripts, well use the **IPTABLES** option.

To block specific addresses, I’m making one script to block them, as these addresses will likely be blocked forever.

Text

Description automatically generated

The IPV4 address will be used for testing purposes while the MAC address is there for proof of concept unless actually needed.

Here we can see that the IP address 192.168.139.250 is blocked

Text

Description automatically generated

Using **sudo iptables -L** to show all the rules (you may have to **redirect >>** to a txt) we can see the rule for the blocked MAC address.



(The entry is in this list twice because I ran the block ips script twice)

1. Blocking TelNet and ICMP (pings)

To block TelNet we can simply use **ufw** as it runs on port 23 by default.

**Sudo ufw deny 23**

Text

Description automatically generated

To block pings we need to modify the **/etc/ufw/before.rules** file

First lets make a copy of the file incase we need to revert back to the original if something wrong were to happen.

**Sudo cp /etc/ufw/before.rules /etc/ufw/before.rulesBACKUP**

now edit the original file and look for the following lines

Text

Description automatically generated

At the end of the line where it says ACCEPT needs to be changed to DENY to block incoming ICMP requests. We can manually change it ourselves or use a SED command.

**sudo sed -i '/ufw-before-input.\*icmp/s/ACCEPT/DROP/g' /etc/ufw/before.rules**

Again, Im going to use two scripts to easily switch between having pings blocked and allowed.

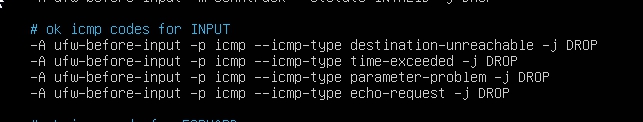
Text

Description automatically generated

Text

Description automatically generated

After running the block script the **/etc/ufw/before.rules** ICMP section will look like this



For testing purposes all the scripts to block certain ports will be ran, including blocking pings.

A picture containing text

Description automatically generated

These are all the ports we will be testing. See ‘Firewalls Testing’ document.

There is also the output of **iptables -L** after the modifications to the firewall. See iptablesUbuntuAfter.txt

Resources that helped me

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

<https://www.journaldev.com/34113/opening-a-port-on-linux>

<https://serverfault.com/questions/238563/can-i-use-ufw-to-setup-a-port-forward>

<https://www.cyberciti.biz/faq/how-to-block-an-ip-address-with-ufw-on-ubuntu-linux-server/>

<https://askubuntu.com/questions/410023/blocking-specific-mac-address-from-contacting-me>

<https://linuxconfig.org/how-to-deny-icmp-ping-requests-on-ubuntu-18-04-bionic-beaver-linux>