Firewalls

1. Deal with web server (open needed ports, and forward port 80 traffic to 8080)

These commands opens the needed ports for web server.

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

This command forwards port 80 traffic to 8080

iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080

To test that these ports can successfully be used on each distribution, I will use the command line tool netcat. Similar to telnet, netcat is a utility to monitor and test network traffic. Using netcat with the -v parameter I can display whether I can successfully connect to a port.

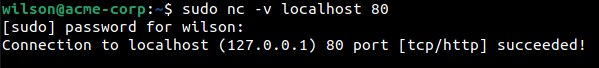
These commands using netcat display that each system can successfully use port 80 and 443.

CentOS:





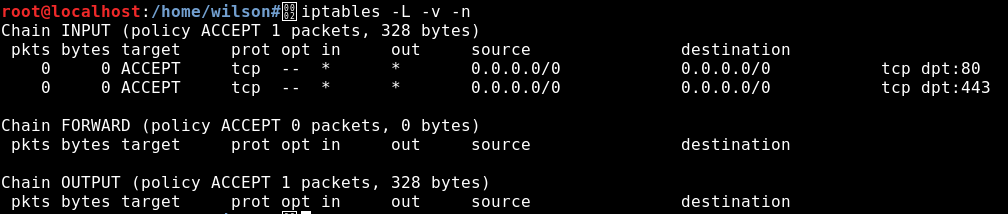
Ubuntu server:



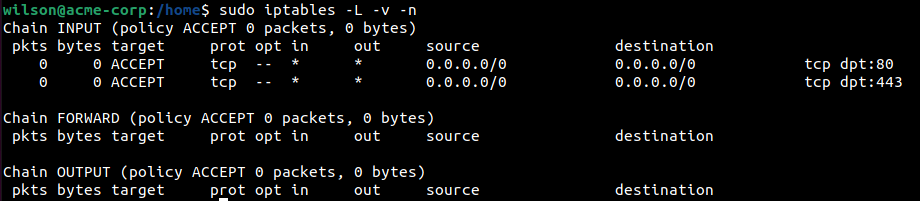


This how each IP Table is configured after inputting question 1 commands.

CentOS



Ubuntu Server



2. Deal with MySQL service (open needed ports)

This command opens port 3306 (default MySQL port). This allows the system to successfully use MySQL services through this port.

sudo iptables -A INPUT -p tcp --dport 3306 -j ACCEPT

This command using netcat display that each system can successfully use port 3306,

CentOS:

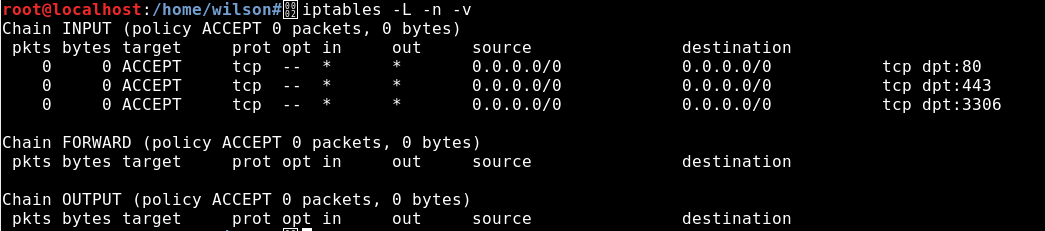


Ubuntu Server

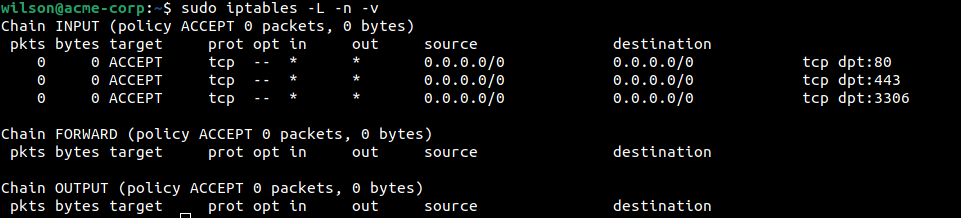


This how each IP Table is configured after inputting the question 2 commands.

CentOS



Ubuntu Server



3. Deal with SSH service (allow incoming and outgoing SSH, second script to deny SSH)

This command allows incoming SSH connection requests through port 22 by appending a new rule to the INPUT chain on the iptable.

iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

This command allows outgoing SSH connection requests through port 22 by appending a new rule to the OUTPUT chain.

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

CentOS



Ubuntu Server



This command drops all incoming traffic to port 22, therefore denying SSH.

CentOS:

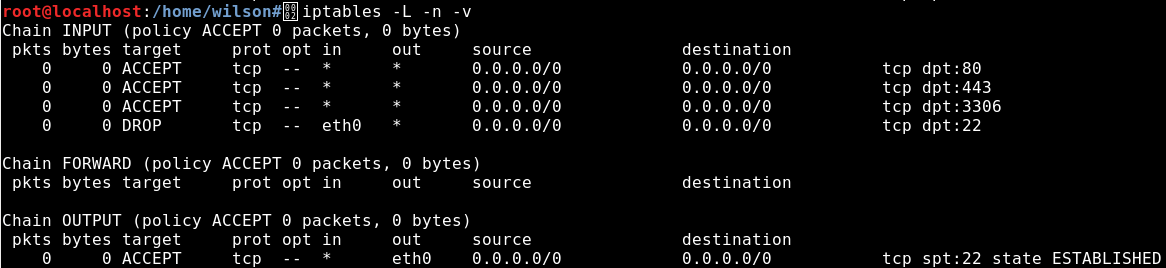


Ubuntu Server

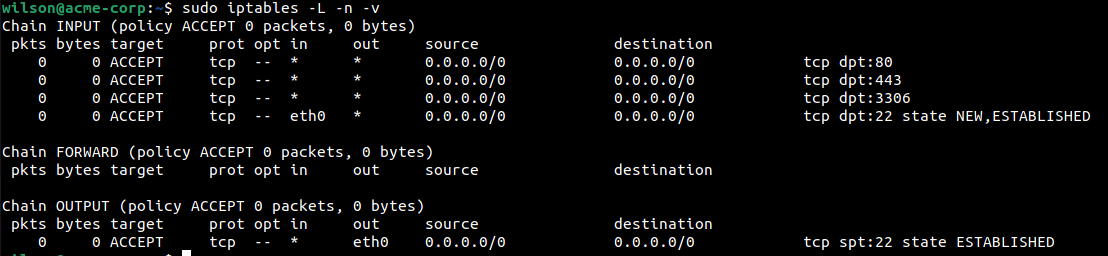


This how each IP Table is configured after inputting the question 3 commands.

CentOS



Ubuntu Server



4. Deal with Email service (Such as allow or block incoming SMTP,POP3...)

Script to allow incoming email protocols or deny them.

This set of commands enables the ports to allow ports for email protocols.

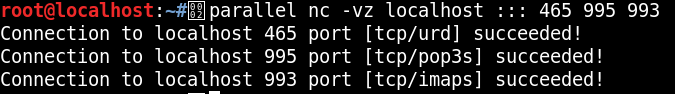
sudo iptables -A INPUT -p tcp --dport 465 -j ACCEPT

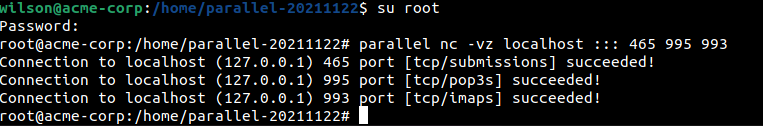
sudo iptables -A INPUT -p tcp --dport 995 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 993 -j ACCEPT

To check multiple ports at once I can use the GNU parallel package to run multiple instances of netcat.

CentOS



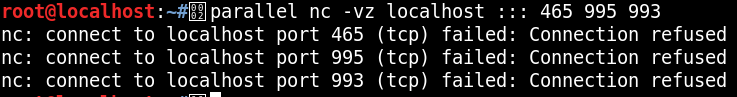
Ubuntu Server

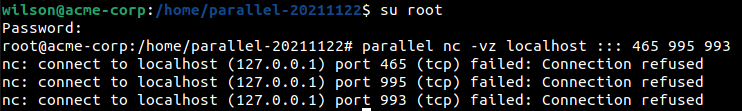
This set of commands rejects the ports for email protocols.

sudo iptables -A INPUT -p tcp --dport 465 -j REJECT

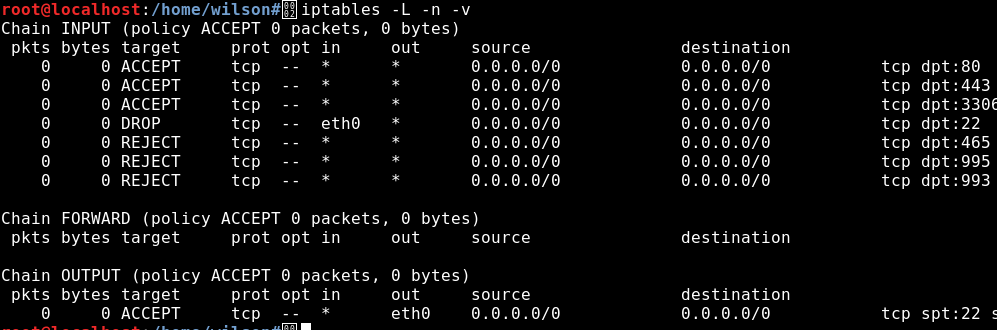
sudo iptables -A INPUT -p tcp --dport 995 -j REJECT

CentOS

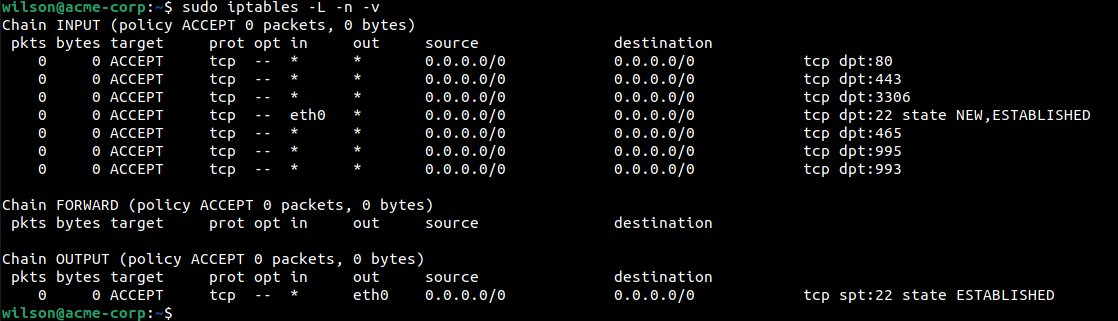


Ubuntu Server

This how each IP Table is configured after inputting the question 4 commands.

CentOS

Ubuntu Server



5. Script to allow/block specific hosts, MAC addresses

This command drops all connection coming from a certain MAC address. Replace your-mac-address with the MAC address to drop.

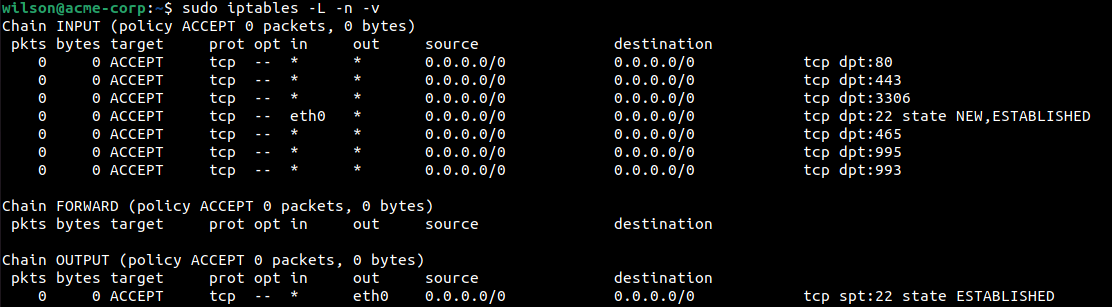
sudo iptables -A INPUT -m mac --mac-source YOUR-MAC-ADDRESS-HERE -j DROP

This command allows SSH (port 22 by default) for a specific MAC address. Replace mac-address-here with the MAC address.

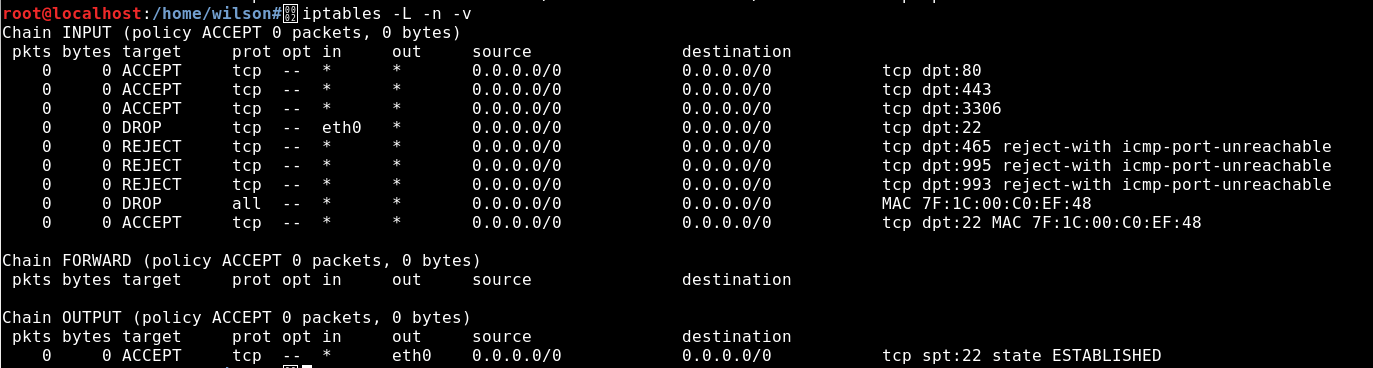
sudo iptables -A INPUT -p tcp --destination-port 22 -m mac --mac-source 7F-1C-00-C0-EF-48 -j ACCEPT

This how each IP Table is configured after inputting the question 4 commands. For the MAC address, I used 7F-1C-00-C0-EF-48 as an example.

Ubuntu Server



CentOS



6. A script/command to block telnet, block ping

This command blocks the default port for telnet.

This command blocks port 23 which is used by telnet

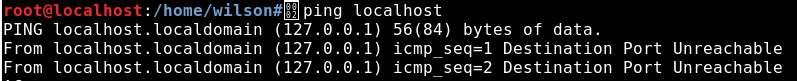
sudo iptables -A INPUT -p tcp --dport 23 -j REJECT

This command blocks any ICMP ping request sent between the local host.

sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT

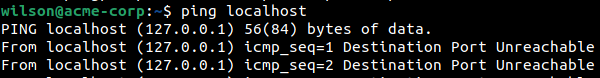
CentOS



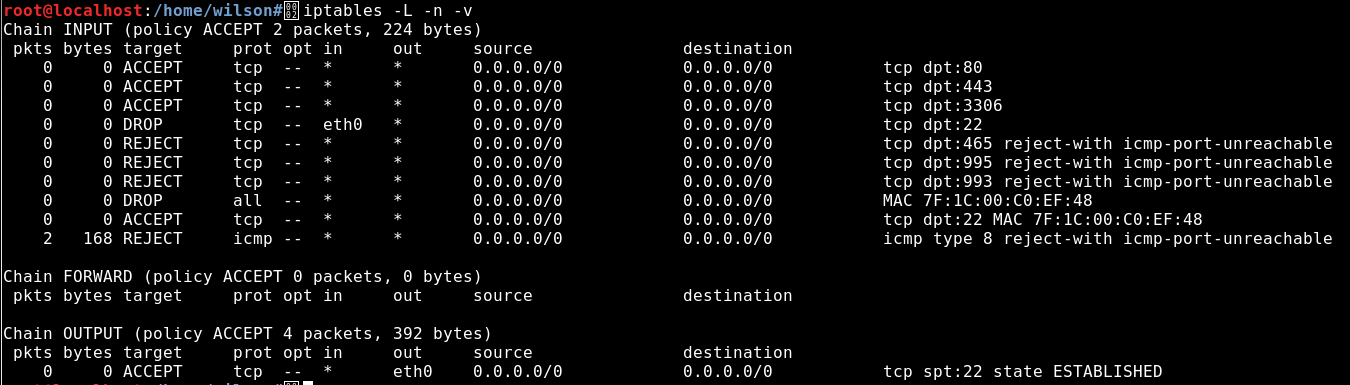


Ubuntu Server

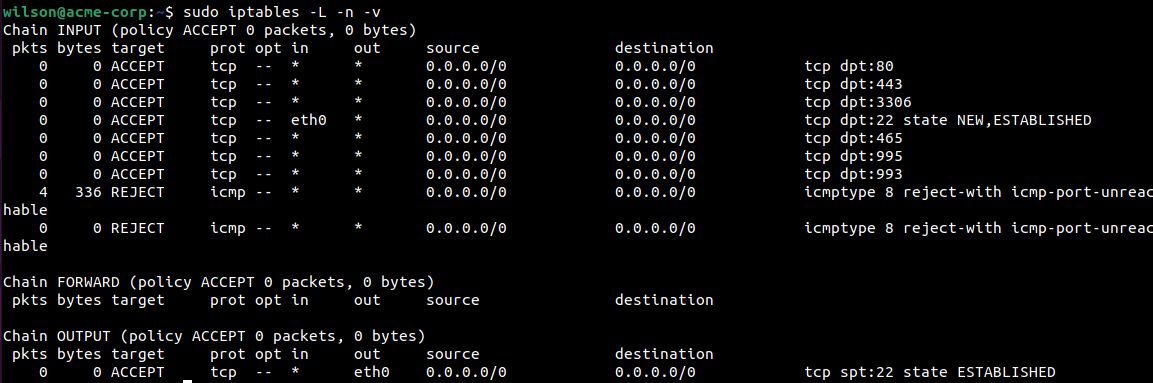




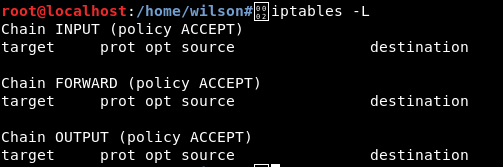
Final CentOS IP Table

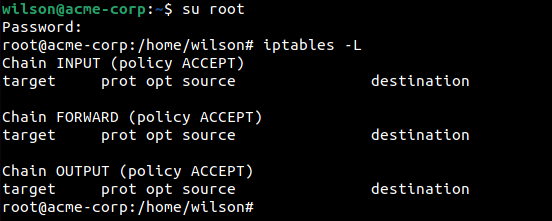


Final Ubuntu IP Table



Starting CentOS IP Table



Starting Ubuntu IP Table

A DDOS attack (distributed denial of service attack) is an a malicious attack method in which the attacker floods and overwhelms a network with requests. By setting up a proper configured IP table, the network administrator can close possible attack route ports, block certain IP addresses and MAC addresses. When creating a ruleset, a network administrator must make sure that a network is completely secure and only allow identified ports and IPs into their network.

Sources:

<https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>

<https://javapipe.com/blog/iptables-ddos-protection/>