Student Name: Md Karim  
Email: mkarim10@aus.edu  
Submission Date: 05/26/2024  
Class Name and Term: CSE548 Summer 2024

Packet Filter Firewall (iptables) Project

# Project Overview

This project focuses on setting up and configuring a packet filter firewall using iptables on a Linux-based system. The project involves implementing security policies to manage network traffic, ensuring proper network connectivity, and configuring necessary software services like a web server. Through this project, I gained practical experience in firewall management, troubleshooting network issues, and applying security policies in a simulated environment.

# Network Setup

A diagram of a server

Description automatically generated

# Software

I have use following tool/command to achieve goal for this project

* iptables – set routing rules
* ping – check network reachability
* ifconfig – check network configuration
* chmod – change file read write or execute permissions
* ls -l – view file permissions
* cat – view file content
* echo – update file content
* sudo – execute command as root user

# Project Description

I have set Server and Client Virtual machine using Hands-on Lab Description (file lab-cs-sys-00101-update-5\_15\_23.pdf) document provided as part of project 1.

Server side:

# Check Server IP ConfigurationA screenshot of a computer screen Description automatically generated

Server can ping 8.8.8.8 before Reset firewall to whitelist

A computer screen shot of a program

Description automatically generated

Server can ping 10.0.2.5 before Reset firewall to whitelist

A screenshot of a computer program

Description automatically generated

Reset firewall to whitelist:

A screenshot of a computer screen

Description automatically generated

Server cannot ping 8.8.8.8 after Reset firewall to whitelist

A screenshot of a computer error

Description automatically generated

Server cannot ping 8.8.8.8 after Reset firewall to whitelist

A screen shot of a computer

Description automatically generated

Server cannot ping localhost after Reset firewall to whitelist

A screen shot of a computer

Description automatically generated

Client side:

Check Client IP configuration:A screenshot of a computer program

Description automatically generated

Client can ping 8.8.8.8 before Reset firewall to whitelist

A screenshot of a computer program

Description automatically generated

Client can ping 10.0.2.6 before Reset firewall to whitelist

A computer screen with white text

Description automatically generated

Client can ping 10.0.2.6 after Reset firewall to whitelist

A screen shot of a computer error

Description automatically generated

Client cannot ping 8.8.4.4 after Reset firewall to whitelist

A computer screen with white text

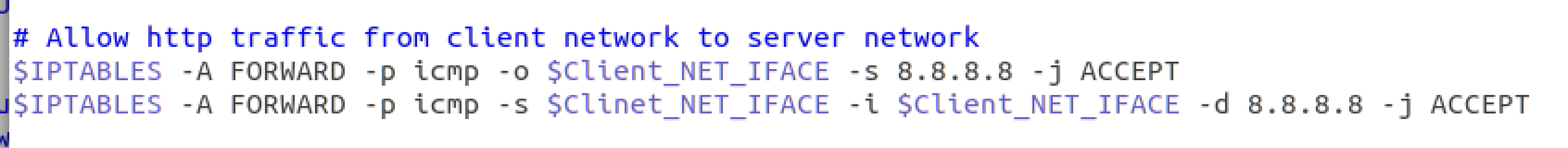
Description automatically generated

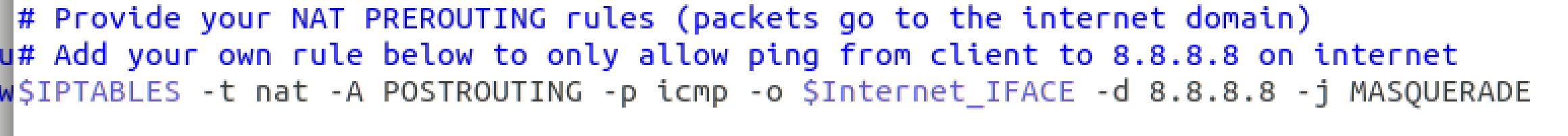
Client cannot ping 8.8.8.8 after Reset firewall to whitelistA computer screen shot of a program

Description automatically generated

**Enable 8.8.8.8 only from client:**

Update following changes in rc.firewall file in server and execute it





Client can ping 8.8.8.8 after updating rc.firewall file and execute it on server side

A screenshot of a computer program

Description automatically generated

sudo nmap -sT -p- 10.0.2.6 after applying firewall rulesA screenshot of a computer

Description automatically generated

sudo nmap -sU -p- 10.0.2.6 after applying firewall rules

A screenshot of a computer

Description automatically generated

Install Apache web server

Execute “Sudo apt-get install apache2” to install apache2 server

A screenshot of a computer

Description automatically generated

Check apache2 server status

A computer screen shot of a program

Description automatically generated

Check web page port configuration for apache2 server

A computer screen with white text

Description automatically generated

Update index.html file context

A computer screen with white text

Description automatically generated

Before applying firewall, rules client can view demo:

A screenshot of a computer

Description automatically generated

After applying firewall without whitelisting client, the client rules can view demo:

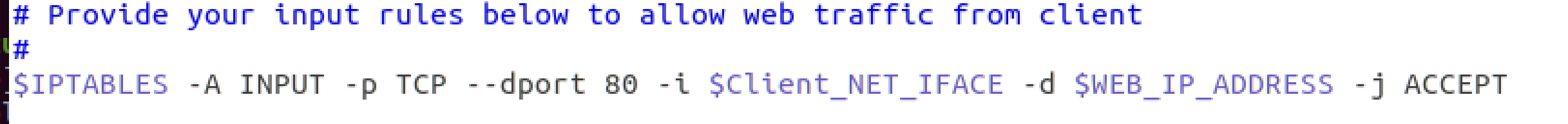
A screenshot of a computer

Description automatically generated

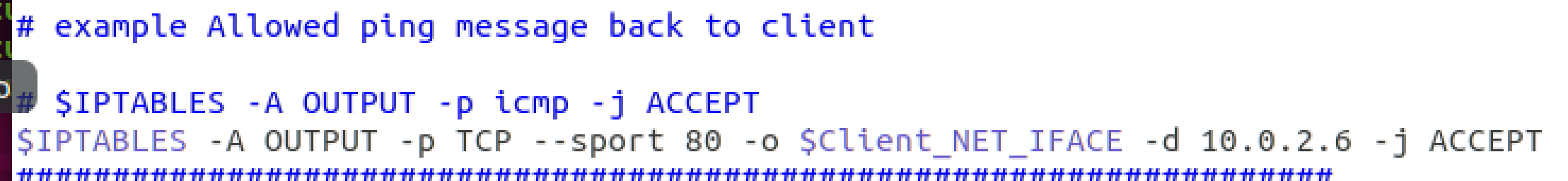
The following change was made in rc.firewall file in the server side and execute it to allow client to access web page

A computer code with text

Description automatically generated with medium confidence



Response back:



Client can view web page after whitelisting client: A screenshot of a computer

Description automatically generated

Server Ip address 10.0.2.6 map with www.test-and-demo.com

A computer screen with white text

Description automatically generated

[www.test-and-demo.com](http://www.test-and-demo.com) is accessible from client after mapping ip with url:

A screenshot of a computer

Description automatically generated

# Conclusion

In this project, I learned how iptables controls packet flow under various use cases. The best way to debug and correctly set up iptables was to use the following command in another terminal tab: *sudo watch -d iptables -L -n -v*

Using this command, I could see the counter changes for each table, which helped me determine if the filter rules were correct.

I invested a significant amount of time setting up the virtual machine. Initially, I tried running the VMs on my OSX-based machine, but I encountered many performance issues that caused the virtual machine to crash and not work properly. Despite following various fixes and different blogs, none of them helped.

Finally, I decided to run the virtual machine on a native Ubuntu machine. This attempt also had its challenges. Fortunately, the guides and explanations I found online helped me fix the issues and start working on the project.

# APPENDIX B: ATTACHED FILE

# References

# Linux Nat Tutorial: https://www.karlrupp.net/en/computer/nat\_tutorial

# Ubuntu :https://help.ubuntu.com/community/IptablesHowTo

# Firewall rules: https://phoenixnap.com/kb/iptables-tutorial-linux-firewall