

FLAKFENCE

A hardware-based Network Security Solution



By

NC Ribiea Ramzan

NC Hasnain Karim

NC Humna Bashir

PC Muhammad Gulbaran

Supervisor

Asst. Professor Waleed Bin Shahid

Department of IS

Submitted to the Faculty of Electrical Engineering, Military College of Signals, National

University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of

a B.E. Degree in Electrical Engineering

JULY 2021

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

“Flakfence: Network Security Solution”

is carried out by

NC Ribiea Ramzan, NC Hasnain Karim, NC Humna Bashir,

PC Muhammad Gulbaran

under my supervision and that in my judgement, it is fully ample, in scope and excellence,
for the degree of Bachelor of Electrical Engineering from National University of Sciences
and Technology (NUST), Islamabad.

Approved by

Asst. Prof. Waleed Bin Shahid

Department of IS

Date:

DECLARATION OF ORIGINALITY

We hereby declare that no content of work presented in this report has been submitted in support of another award of qualification or degree either in this institution or anywhere else.

PLAGIARISM CERTIFICATE (TURNITIN REPORT)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Ribiea Ramzan

00000220613

Hasnain Karim

00000237979

Humna Bashir

00000237317

Muhammad Gulbaran

00000240948

Signature of Supervisor

ACKNOWLEDGEMENTS

We are thankful to my Creator Allah Subhana-Watala to have guided us throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed, we could have done nothing without Your priceless help and guidance. Whosoever helped us throughout the course of my thesis, whether our parents or any other individual was Your will, so indeed none be worthy of praise but You.

We are profusely thankful to my beloved parents who raised us when we were not capable of walking and continued to support us throughout in every department of my life.

We would also like to express special thanks to my supervisor AP Waleed Bin Shahid (Dept of IS) for his help throughout the FYP and Thesis and also for Network Security course which he has taught me.

Finally, we would like to express my gratitude to all the individuals who have rendered valuable assistance to our study.

Dedicated to our exceptional parents and adored siblings whose tremendous support and cooperation led us to this wonderful accomplishment.

ABSTRACT

Networks are becoming more and more complex with each passing day and their usage in our daily lives is also increasing. People use networks more than they think about them. nowadays people even transfer their hard-earned money using these networks. However, with the increased use of networks, they have become more vulnerable and are at the attacker's mercy. So, a need for a product that can secure our network is great. Our product, 'Flakfence' will provide a solution. This device is controlled by the admin who sets up the policies. It would render network security by filtering ingress and egress network traffic based on a bunch of administrators characterized rules. It would protect against a wide variety of network attacks, espionage, and infiltrations and would also offer an inbound and outbound firewall with intrusion detection and prevention capabilities. The user can have an elaborate view of all connected devices along with network activities. It would also give the user complete control of the network he/she is connected to, from blocking web pages and cutting off access to unwanted devices. Moreover, Database server is set up online, thereby actively monitoring and detecting any suspicious connection from anywhere in the world. It would monitor each IP address pinged by each device on the user's network.

Table of Contents

ABSTRACT	vii
Table of Contents	viii
List of Figures	x
Chapter 1: Introduction	1
1.1. Project Overview	2
1.2. Problem Statement	3
1.3. Objectives	3
1.3.1. Academic Objectives	4
1.3.2. Industrial Objectives	4
1.4. Approach/ Research Methodology	4
1.5. Block Diagram	5
1.6. Limitations	6
1.7. Organization of Document	6
Chapter 2: Literature Review	7
2.1 Literature Review	8
2.2 Previous Solutions and their flaws	10
2.3 Novelty of Flakfence	12
2.4 Conclusion	12
Chapter 3: Technological Requirements	14
3.1. Hardware Requirements	15
3.1.1 MINI PC	15
3.1.2 Router	16
3.1.3 Raspberry pi	16
3.2 Software Requirements	18
3.2.1 Software	18
3.2.2 Operating Systems	18
3.2.3 Programming languages	19
Chapter 4: Proposed Solution ‘Flakfence’	20

4.1.1 Gateway	21
4.1.2 Setting Network Interfaces.....	22
4.1.3 Setting up Forwarding:	22
4.1.4 Setting up DHCP Server:	24
4.1.5 Setting up Hostapd:.....	25
4.1.6 Raspberry pi gateway is ready:	26
4.2 IP TABLES	26
4.3 Snort	28
4.4 Android Application.....	29
Chapter 5: Testing, Analysis and Results	31
5.1 Firewall Rules	32
5.2 Network Monitoring Logs	33
5.3 Connected devices on App.....	34
5.4 Intrusion Alerts	35
Chapter 6: Enhancements & Future Work.....	36
6.1 IPS.....	37
6.2 VPN Server	37
6.3 Blocking TOR.....	38
6.4 Deep Packet Inspection.....	38
Appendix A.....	39
Appendix B	41
Appendix C	42
Appendix D.....	44
Appendix E	46
Appendix F.....	47
References.....	49

List of Figures

Figure 1: Schematics.....	5
Figure 2: Comparison of existing firewall solutions.....	11
Figure 3: Mini PC	15
Figure 4:Router	16
Figure 5:Raspberry Pi	16
Figure 6:Flakfence	21
Figure 7:Iptables Nat Table	23
Figure 8:Linux network interface	24
Figure 9: DHCP server.....	24
Figure 10: Hostapd.....	25
Figure 11: Mobile Connected with Raspberry pi.....	26
Figure 12: Snort	28
Figure 13: Snort Configuration File.....	29
Figure 14: App Login and Menu Page.....	30
Figure 15: Firewall Rules.....	32
Figure 16: Network History	33
Figure 17: Connected Devices	34
Figure 18: Intrusion Alerts.....	35

Chapter 1: Introduction

1.1 Project Overview

1.2 Problem Statement

1.3 Objectives

1.4 Approach

1.5 Block Diagram

1.6 Limitations

1.7 Organization of Document

CHAPTER 1: Introduction

This chapter provides comprehensive introduction of the project “Flakfence”. Flakfence is a multi-purpose cyber security solution for everyone. It can be controlled and configured by a simple, user friendly Android Application and solves the problem of security conscious individuals or organizations. It would be a small hardware device that would give the detailed monitoring of connected devices. It would also generate alerts in our mobile application for any abnormal activities [1].

1.1. Project Overview

A firewall controls the progression of information and traffic to or from your network. Network data is in form of packets which might actually contain harmful codes which thusly could harm the network. It would check both the ingress and the egress traffic of the network. Firewall applies a specific arrangement of rules ingress and the egress network traffic to examine whether they align with those rules or not.

- If it does not match the blocking rules set by admin – traffic will pass.
- If it matches – it rejects or blocks the traffic.

In this way, your network remains entirely protected from such threats, be it internal or external. The firewall doesn't only manage the internet side of the network but also manages the internal side of it [2].

Flakfence gets rid of intricate firewall configurations, the lack of real time networking monitoring and using uni-purpose expensive solutions. Flakfence has the capabilities of firewalls IDS parental control, real time monitoring and very easily managed through our android app with a pocketable

device. When packet tries to enter our network, it will be checked by the IDS. An IDS evaluates a suspected intrusion and whenever it takes place it flags an alert. IDS also checks for attacks that are generated from within a system [3]. A feature of network monitoring is also a part of Flakfence. We are able to see what websites are being accessed by users. This history can later be checked by the admin. This helps the admin determine as to what sort of websites are being used by the users. If admin thinks that a website which should not be accessed is being accessed, then he may completely block the website.

1.2. Problem Statement

Efficiency and feasibility are the need of the modern era. As it stands today more and more networks are getting compromised and loss of data is reported. PCs, workstations and different gadgets that are associated with the Internet are helpless and a potential target for a range of different threats. The attacks on network have led to loss of confidential information, critical data and spread of unwanted malware in networked computers. To overcome these attacks on networks and to avoid risk of data breaches, there is a need of a system that is aware of the peculiarities and vulnerabilities that can be exploited and used against the state and individuals and be able to counteract them. A lot of such solutions exist in the market which are either too costly, too resource intensive or hard to configure for ordinary users. These hurdles have accentuated the need to come up with a handy solution, which is easy to configure, manage, update and replace.

1.3. Objectives

Objectives are classified into the following categories:

1.3.1. Academic Objectives

- To employ our knowledge and practical skills pertaining to computer networks and network security to the best of society
- This would open further avenues of research in the security and privacy domain as it can be further enhanced to make the product commercially viable.

1.3.2. Industrial Objectives

- Designing of an all-in-one, easy-to-use, and upgradable indigenous network security solution that makes use of IDS to monitor the network traffic.
- This handy solution will institute parental control to surveil the children browsing activities and filter out unwanted content.
- The proposed solution will address security concerns and trust issues of privacy-conscious people, private and public confidentially centered organizations, corporate offices, and security researchers.
- Flakfence will also be very economical and cheap enough so that almost every household can afford one.

1.4. Approach/ Research Methodology

First, we adopted the approach of Arp spoofing. ARP spoofing is a kind of attack where an attacker sends misrepresented ARP (Address Resolution Protocol) messages over a local area network. This outcome in the connecting of an aggressor's MAC address with the IP address of a genuine PC or worker on the network [4]. However, we opted against it because we were not able to check the data of HTTPS websites. We were not able to note the traffic of the HTTPS websites. Then we

adopted another methodology of making a gateway. By making the gateway we were able to check the traffic of both HTTP and HTTPS websites. This gateway ensured that all the traffic would go through it and all the users will be connected to our device (Flakfence). Flakfence will act as an interceptor for all approaching and active network traffic. It would later channel this traffic dependent on the network rules carried out by the admin who is using Flakfence Android Application.

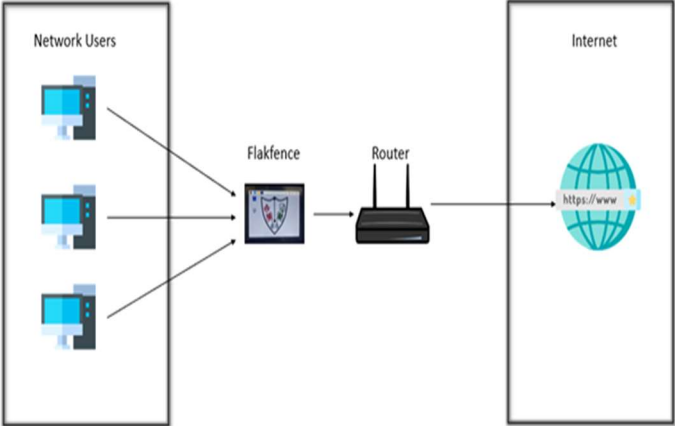


Figure 1: Schematics

1.5. Block Diagram

As our device is plug and play, it is the easy way to secure the data traffic. Flakfence sits between with the users and router acting as a bridge controlling the flow of packets/traffic. These packets are being monitored and iptables rules are applied on the features that Flakfence is providing for the security of the users. Threats to the network will be detected by IDS that are embedded into Flakfence. We can also block specific Ports and Mac-Addresses.

1.6. Limitations

- Limited amount of storage is available.
- It affects the speed of web surfing, and it also adds very little latency in the network.
- Flakfence firewalls only block unwanted data transmissions; Features like anti-virus, anti-malware is not provided [5].

1.7. Organization of Document

Chapter 1 Provides essential details and overview of Flakfence.

Chapter 2 Deals with the literature review carried out for Flakfence and its core features.

Chapter 3 Discusses the hardware and software requirements of Flakfence

Chapter 4 Discusses the proposed solution Flakfence and setup for working environment.

Chapter 5 Provide all results and analysis regarding testing and working of Flakfence

Chapter 2: Literature Review

2.1 Literature Review

2.2 Previous Solutions and their flaws

2.3 Novelty of Flakfence

2.4 Conclusion

Chapter 2: Literature Review

Literature review of Flakfence is discussed comprehensively that consists of network monitoring solutions their limitations and Flakfence novelty. Traditional network monitoring solutions need to be installed on the client's device and thus can be circumvented or deleted by the user anytime. It is also more user-intuitive and provides notifications to the admin. Flakfence addresses this issue and comes up with a solution that is ready to use; it is a plug-and-play device. At the business or corporate level, Flakfence can be utilized to watch the digital movement of the representatives. It is a finished answer for endeavors/government/military too, which can help improve representative's usefulness and security of the network. With this, the dangers of information spillage and representative extortion can be considerably decreased.

2.1 Literature Review

Network infrastructure is growing rapidly, Network traffic is extending past where customary methodologies can, in any case, be utilized to productively identify oddities. To help in defeating this issue, present-day ways to deal with traffic checking should be researched. Our writing survey comprises of investigation of various parental control solutions. We likewise educated Python programming as well. The Internet can be easily accessed nowadays in fact most children carry a smartphone and have access to the computer. Parents want to have track of what their child is surfing on the internet. This is not to control their child's freedom but to make sure that their child does not access websites that are for adults only. There is objectionable content is everywhere on the internet and most parents feel defenseless to safeguard their kids from it. In parental control parents can administer the digital devices of their children. With these controls, you can limit the

internet access of your kid's device [6]. A new report uncovers the following realities with respect to unaided access of Web to the youngsters.

- 80% of the youngsters are presented with the improper substance before age 11. • 60% of youngsters were victims of cyberbullying in the past year, 2020.
- 7 out of 10 youngsters had companions who enticed them to share improper substance or pictures via online media.
- 82% of online maltreatment wrongdoings began from long-range informal communication locales that hunters use to acquire understanding into their casualty's propensities and preferences.

Flakfence is more user-friendly, and client orientated and gives misuse warnings to the administrator. Flakfence can be utilized in homes subsequently giving total logs to the guardians. It prevents children from accessing 18+ content on the web. Parental control offered would allow parents to monitor the internet usage of their children. Parents would also be able to limit the onscreen time of their children by blocking the websites that their children visit the most. Every organization wants the most productive outcome from their employees. In the present day and age, it is impossible for most of the employees to work without the internet. However, employees abuse the facility of the internet and watch random videos on YouTube and even watch seasons in the office during their working hours. This practice is highly unethical. Companies want productive outcomes from their employees and also keep their networks safe from breaches. There is a need for a system that is aware of the peculiarities and vulnerabilities that can be exploited and used against organizations and individuals and be able to counteract them. A lot of such solutions exist in the market, which are either too costly, too resource-intensive, or hard to configure for ordinary

users. The hiring of more than required staff is not beneficial for small organizations as this reduces their profit.

2.2 Previous Solutions and their flaws

Some of the firewall solutions are following:

- **Firewalla:**

Firewalla is a networking device that acts as a bridge between connected devices and the main router. Firewalla can see and block the unwanted traffic that is passing through our network. Firewalla does not monitor the traffic of your network but only the traffic that goes to the internet.

It is very expensive to ranges from \$169-\$418 [7].

- **Pfsense:**

Pfsense can be found in both FreeBSD and closed source models. Pfsense needs to be installed on a physical computer which is very technical work. To configure it deep knowledge of networks is required. Pfsense ranges from \$199-\$699 [8].

- **Cisco ASA 5500-X:**

Portability and cloud drive efficiency yet present danger. To ensure your resources, you should see the clients, applications, gadgets, and dangers on your organization and what they are doing. Cisco ASA 5500-X firewalls convey the organization perceivability you

need, predominant danger and progressed malware assurance, and more noteworthy robotization to lessen cost and intricacy. It is expensive as its price is \$400.

- **Fortinet FortiGate:**

FortiGate firewalls are built in such a way that they offer almost the best-in-class threat protection and performance. Their processors are built specifically for this task. Fortinet FortiGate provides users with very advanced firewalls, but they are expensive and difficult to maintain. It ranges from \$250-\$2000 [9].






	Cost	Configuration	Firewall	Parent Control	IDS/IPS	Android App
	Rs 65000 + SP	Through Android App	✓	✓	✓	✓
	Rs 52000 +SP	Hi-Tech Skills required	✓	✓	✓	✗
	Rs 61,785	Hi-Tech Skills required	✓	✓	✓	✓
	Rs 4,200 plus monthly subscription dues	Weak Security	✓	✗	✗	✗
	Rs. 8,200 per year	Modest Security	✓	✗	✗	✗

Figure 2: Comparison of existing firewall solutions

These generally utilized arrangements accompany a few issues. They cannot be arranged by an average person. Firewalla is a straightforward fitting and plays gadget however that is excessively exorbitant for an average person to manage. Whereas all the other solutions available in the

market are either too expensive or too hard to configure for a normal person. And most of the firewalls present do not offer parental control and IDS.

Our Solution to this Flakfence offers a firewall that is easy to use and can be used by anyone. It can be used by a person who works in a bank and has no knowledge as to how the networks work or even by a trained person who is so busy in his job that he cannot configure a firewall at his home. Through our mobile application, everything can be managed very easily. Admin can block websites, other users and also monitor it through their app.

2.3 Novelty of Flakfence

Flakfence has an edge over most popular parental control apps, since Flakfence is a plug and play indigenous solution that is ready to use that provides us with

- Monitor user's network traffic
- Blacklist inappropriate websites anytime for any user.
- Send real-time notification of abuse to the admin.
- Intrusion alerts with help of IDS
- An android application.
- It is cost-effective when compared to other firewalls available.

Flakfence helps confidentiality-centric individuals, organizations, and parents who want to protect their networks/organizations/families by real-time monitoring of their network through an easy-to-use android application.

2.4 Conclusion

The proposed project is a product. Past projects done in the field of organization security at MCS have not been focused on making a product that is for everyone. There have been projects at MCS that have been on network monitoring, but no project has been focused on making a product that

can be used by anyone. We offer an advanced firewall that has the capabilities of IDS. We also offer networking monitoring through which the entire network history can be seen by the administrator. Solutions like these are available in the market but none of them offer an android application that is user-friendly. Due to this application, our device can be controlled by any person who does not have deep knowledge of networks.

Chapter 3: Technological Requirements

3.1 Hardware Requirements

3.2 Software Requirements

Chapter 3: Technological Requirements

This part gives extensive insights concerning specialized prerequisites of Flakfence for example programming, equipment, and OS necessities. This section will likewise give pursuers insights concerning this firewall, its establishment, and installed packages.

3.1. Hardware Requirements

The Hardware required for the implementation of the indigenous solution includes:

3.1.1 MINI PC

The significant equipment prerequisite for this venture is a Mini PC with two Network Interface Controller (NIC); one NIC for LAN and the other one is for WAN for example Web. The LAN card will be associated further to change to which various PCs can be joined effectively and WAN card will interface the clients to the Internet.



Figure 3: Mini PC

3.1.2 Router

A router is needed to be connected with the MINI PC which has Flakfence solution ported onto it. All traffic going through the router will in this way be observed by Flakfence.



Figure 4:Router

3.1.3 Raspberry pi



Figure 5:Raspberry Pi

Raspberry pi that consists of:

- Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM (depends on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header
- 2 × micro-HDMI ports
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot
- 5V DC via GPIO header
- Power over Ethernet (PoE) [10]

We have tried and tested our project using both the raspberry pi and a mini pc. Our project works on the devices. For homes, we recommend using the raspberry since that brings down the cost significantly and for large-scale use such as offices or schools, we recommend using a mini pc. We recommend mini pc for offices because it features an SSD which is much faster than a memory thus providing better processing power.

3.2 Software Requirements

Flakfence will use certain software which ensures that our product Flakfence will be able to carry out its tasks assigned by the admin. Flakfence will behave in such a way that the user will not know as to which software is being used as the user will manage everything through an application.

3.2.1 Software

- **PyCharm**

PyCharm is a Python Integrated Development Environment (IDE) that provides a very vast range of tools. In the environment provided by the PyCharm developers can work in a very productive way. PyCharm is available in both Windows and Linux. We have done the majority of the work related to python in its Linux version [11].

- **Android Studio**

Android studio is a platform where its developers have created such an environment where we can easily build applications for mobile phones and other gadgets. It is free of cost [12].

- **Snort**

Snort is an Open-Source software. It provides us with features like Intrusion Detection System. Admin writes rules of snort which sort of program snort in a way that it would detect the malicious activity and generate an alert. Snort can also act as a packet sniffer and can also be configured according to the needs of the user [13].

3.2.2 Operating Systems

- Kali Linux
- Android OS
- Raspbian

3.2.3 Programming languages

Following programming languages for making Flakfence:

- **Python**

Python is a high-level general-purpose programming language. Python being user-friendly and simple to use is one of the most widely used programming languages in the world. Python is mainly famous because of its ease of use and the number of libraries available in it. Python is very popular amongst cybersecurity experts and data science experts. Network programmers also make use of this language. In python writing a code is much simpler and shorter when compared with C++ and other programming languages [14].

- **Java for Android**

Java is a technology used for building applications. It uses managed codes in such a way that they can be used on mobile devices. Android is a Linux-based open-source programming stage for mobile phones. Android application is created by utilizing the Java programming language and the Android SDK for Flakfence Android app [15].

- **Bash Scripting**

A Bash script is a text file that contains few commands. Any order that can be executed in the terminal can be placed into a Bash script. Any arrangement of orders to be executed in the terminal can be written in a content record, in a specific order, as a Bash script. It is utilized to computerize monotonous undertakings on the Linux filesystem. It may incorporate a bunch of orders or a solitary order, or it may contain the signs of basic programming like circles, capacities, restrictive develops, and so on Successfully, a Bash script is a PC program written in the Bash programming language [16].

Chapter 4: Proposed Solution ‘Flakfence’

4.1 Setting up Working Environment

4.2 IPTables

4.3 Snort

4.4 Android Application

Chapter 4: Proposed Solution ‘Flakfence’

4.1 Setting up working environment:

A guide for setting up raspberry pi gateway is provided.

4.1.1 Gateway

Gateway between user and router to intercept the network traffic on virtual machine.

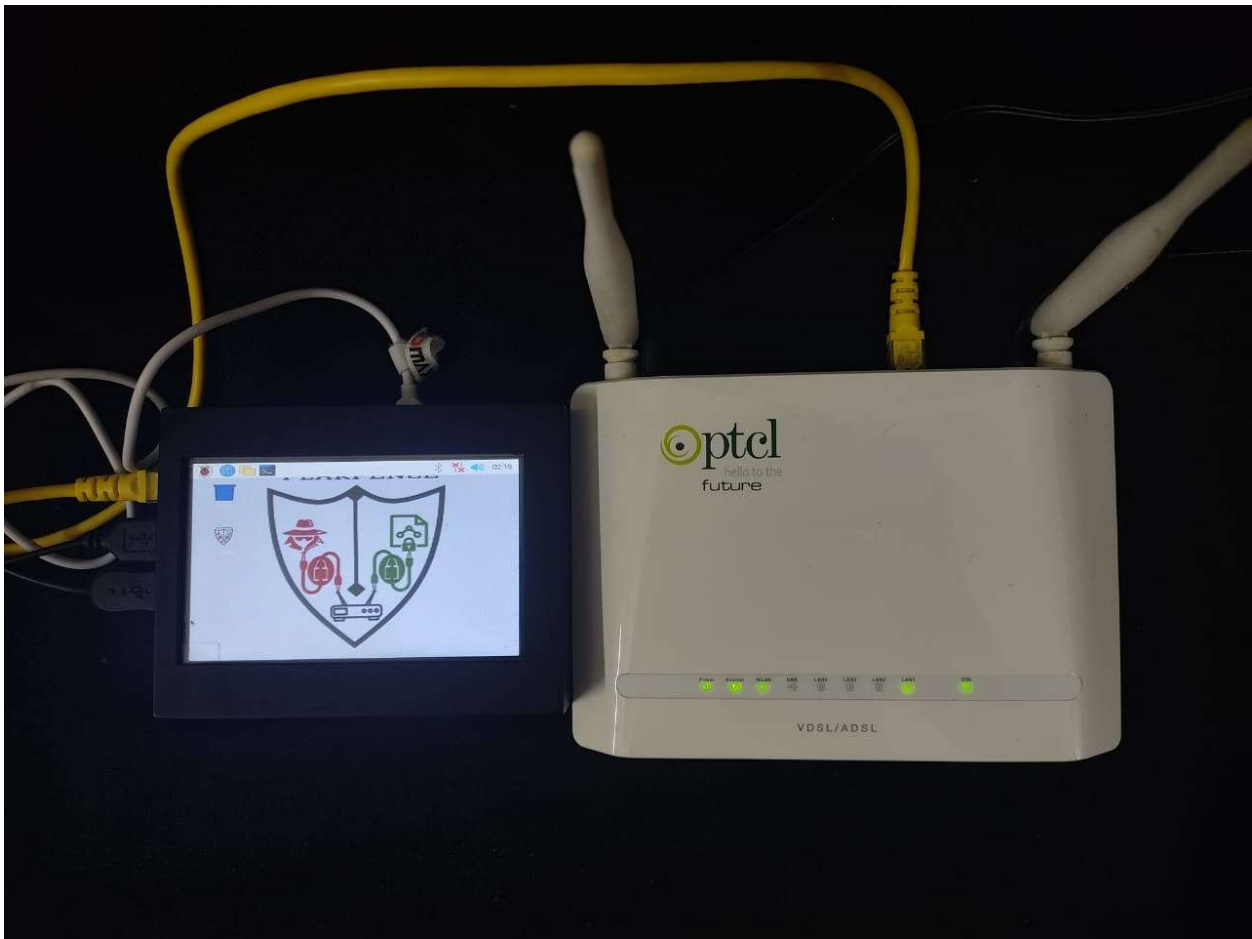


Figure 6:Flakfence

Providing services that router provides us:

- NAT, multiplex a single connection (iptables)
- Forwarding, to forward to external network (iptables)
- DHCP, manage ip address leases (isc-dhcp server)
- Wi-Fi service, for connection of clients (Hostapd)

This gateway has been established so that the user will be to connected to our device rather than the router.

4.1.2 Setting Network Interfaces

Open terminal

Then open network interfaces file: `/etc/network/interfaces`

Save the file and exit. Then up wlan interface

Restart networking service by command: `~$ sudo service networking restart`

Then check network connections by `ifconfig`:

Interface settings are done.

4.1.3 Setting up Forwarding

We are using interfaces of Linux wlan and eth0. Eth0 is the interface that is connected to the router.

Our devices are connected to the wireless interface. But our wireless adaptor is not connected to the internet. It is acting as a hotspot. Now to provide all the devices connected with internet forwarding is done. This way all the traffic that comes on wlan goes to eth0. If this is not done then internet will not work on the devices that are connected to our wlan interface.

For forwarding we will be using iptables:

Allow NAT masquerading in POSTROUTING chain of NAT table of iptables.

```
(root@kali)~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

(root@kali)~# iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 109 packets, 7399 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain INPUT (policy ACCEPT 109 packets, 7399 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
 pkts bytes target    prot opt in     out     source         destination
    0    0 MASQUERADE all  --  *        eth0    0.0.0.0/0      0.0.0.0/0

(root@kali)~#
```

Figure 7:Iptables Nat Table

Then check default policy of Forwarding chain of FILTER table: (if policy is accept leave it as it is)

If policy is not ACCEPT, then change it by following command: ~\$ sudo iptables -P FORWARD ACCEPT

Then enable forwarding in sysctl.conf and set /proc/sys/net/ipv4/ip_forward =1_

Save Iptables rules so that we can reuse them. And attach them in network interfaces file so that whenever raspberry is rebooted, nat masquerading rules will automatically be added in iptables.

_pi@raspberrypi: ~\$ sudo nano /etc/network/interfaces

Forwarding Setting is done.

```
auto eth0
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0

up iptables-restore < /etc/iptables_rules.nat
```

Figure 8:Linux network interface

4.1.4 Setting up DHCP Server

DHCP server is set up so that it can allocate IP to the all the users that are connected with Flakfence automatically.

Install isc-dhcp-server.

Edit /etc/dhcp/dhcpd.conf and /etc/default/isc-dhcp-server.

Save the file and exit. Start dhcp server and see status:

```
(root@kali)~# service isc-dhcp-server start

(root@kali)~# service isc-dhcp-server status
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Mon 2021-05-31 04:12:43 EDT; 4s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1327 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 4 (limit: 5840)
   Memory: 6.3M
      CPU: 37ms
  CGroup: /system.slice/isc-dhcp-server.service
          └─1343 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf wlan0

May 31 04:12:41 kali systemd[1]: Starting LSB: DHCP server ...
May 31 04:12:41 kali isc-dhcp-server[1327]: Launching IPv4 server only.
May 31 04:12:41 kali dhcpd[1343]: Wrote 0 leases to leases file.
May 31 04:12:41 kali dhcpd[1343]: Server starting service.
May 31 04:12:43 kali isc-dhcp-server[1327]: Starting ISC DHCPv4 server: dhcpd.
May 31 04:12:43 kali systemd[1]: Started LSB: DHCP server.

(root@kali)~# ss
```

Figure 9: DHCP server

4.1.5 Setting up Hostapd

Hostapd is used to make a WIFI hotspot so that all the users can connect directly with Flakfence.

Our device basically acts as an access point. With our device acting as access point users can easily connect to Flakfence

Install Hostapd

Open and edit hostapd.conf file.

Save and exit the file.

Then open /etc/default/hostapd file

Unmask Hostapd service:

Now start hostapd service and see status:

```
(root@kali)~# service hostapd start
(root@kali)~# service hostapd status
● hostapd.service - Access point and authentication server for Wi-Fi and Ethernet
   Loaded: loaded (/lib/systemd/system/hostapd.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-05-31 04:13:55 EDT; 4s ago
     Docs: man:hostapd(8)
   Process: 1377 ExecStart=/usr/sbin/hostapd -B -P /run/hostapd.pid -B $DAEMON_OPTS ${DAEMON_CONF} (code=exited, status=0/SUCCESS)
  Main PID: 1393 (hostapd)
    Tasks: 1 (limit: 5840)
   Memory: 2.9M
      CPU: 42ms
   CGroup: /system.slice/hostapd.service
           └─1393 /usr/sbin/hostapd -B -P /run/hostapd.pid -B /etc/hostapd/hostapd.conf

May 31 04:13:54 kali systemd[1]: Starting Access point and authentication server for Wi-Fi and Ethernet ...
May 31 04:13:54 kali hostapd[1377]: Configuration file: /etc/hostapd/hostapd.conf
May 31 04:13:55 kali hostapd[1377]: Using interface wlan0 with hwaddr 00:c0:ca:59:be:b3 and ssid "flakfence"
May 31 04:13:55 kali hostapd[1377]: wlan0: interface state UNINITIALIZED→ENABLED
May 31 04:13:55 kali hostapd[1377]: wlan0: AP-ENABLED
May 31 04:13:55 kali systemd[1]: hostapd.service: Can't open PID file /run/hostapd.pid (yet?) after start: Operation not permitted
May 31 04:13:55 kali systemd[1]: Started Access point and authentication server for Wi-Fi and Ethernet.

(root@kali)~#
```

Figure 10: Hostapd

For keep Dhcp server and hostapd running after reboot:

```
sudo update-rc.d hostapd enable
```

```
sudo update-rc.d isc-dhcp-server enable
```

4.1.6 Setup Completed

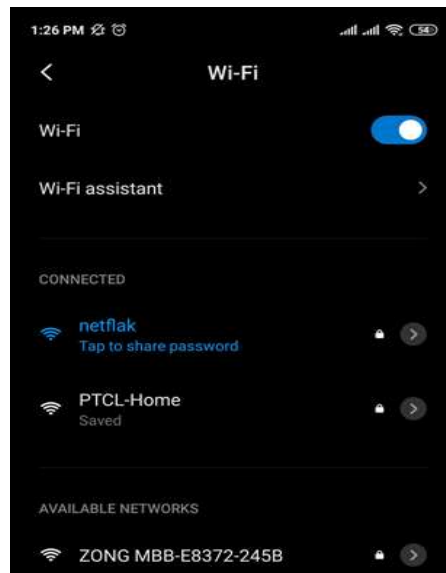


Figure 11: Mobile Connected with Raspberry pi

4.2 IP TABLES

Iptables is a firewall utility that is built specifically for Linux operating systems. Packet filtering and NAT rules can be managed by using iptables. It is one of the most versatile firewalls available.

iptables is a command-line firewall utility. It makes use of policy chains. on the basis of these chains and rules set by the admin, it decides whether to allow or block the traffic.

A few built-in tables for IPTABLES:

1. Filter Table

It is the default table for iptables. If one does not define their table while using iptables then iptables would choose this table for default.

Built-in chains in the Filter Table:

INPUT chain – coming towards the firewall.

OUTPUT chain – going away from the firewall.

FORWARD chain – Packet for another NIC on the local server.

2. NAT table

Iptables's NAT table comes with the following chains already built-in:

PREROUTING chain – Alters packets before routing takes place. This helps in translation of destination Ip address that matches the routing on the local server.

POSTROUTING chain – Alters packets after routing takes place. i.e., This helps to translate the source IP address that matches the routing on the destination server. This is used for source NAT.

INPUT chain and OUTPUT chain – INPUT and OUTPUT chains are traversed for packets delivered to and sent from applications running on the local machine. NAT for locally generated packets on the firewall.

Rules have a criterion and a target. In the event that the models are coordinated then, it is sent to the rules that executes the special values mentioned in the target [17].

In the event that the models are not coordinated then it decides to move on to the next rule.

Target Values:

Special values that can be specified:

ACCEPT – Firewall accepts the packet.

DROP – Firewall drops the packet and do not answer that packet

REJECT– Rejects the packet but it does send a packet back.

QUEUE – Firewall passes the packet to the user space.

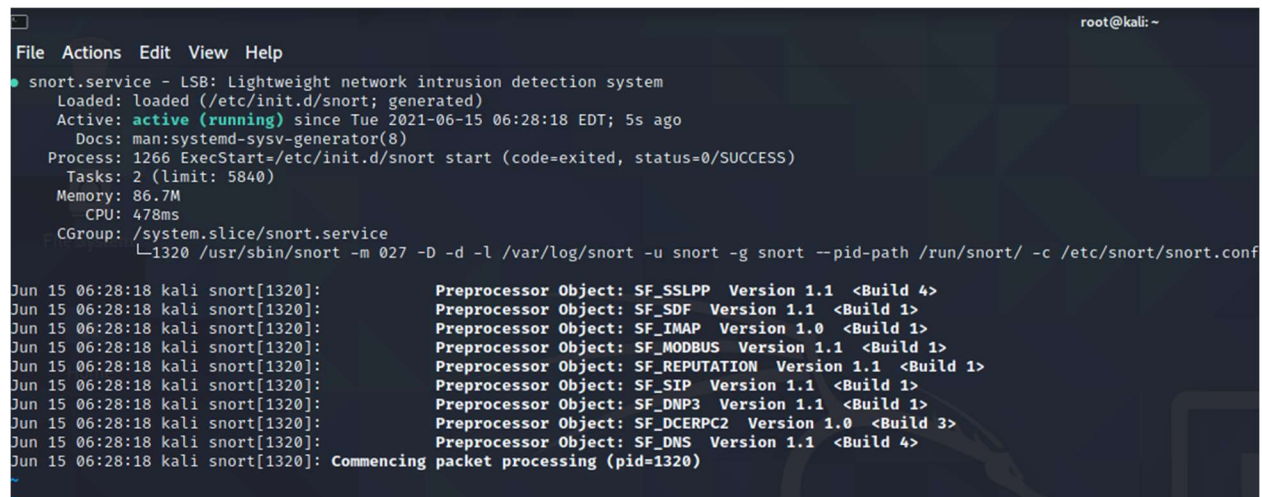
RETURN – Firewall will send the control back to the controlling chain.

In our project, we have used the NAT table post routing in order to set up NAT masquerading. We have used the forward chain so that all the rules applied would work on the entire network. With the help of IPTABLES we were able to block specific mac addresses, ports, protocols, and websites. Python scripts and bash scripts are used to implement the iptables rules. When the device gets a command from android app via socket connection like BlockMAC, BlockPort, BlockWebsite, FamilyControl etc, scripts for respective command is implemented adding the rules in iptables.

4.3 Snort

In Flakfence we have used snort to perform IDS (intrusion detection system). IDS checks whether any malicious activity is taking place or not and if it does it sends an alert to the admin through android application. This IDS works on the entire network and checks all the traffic. In general firewall only checks the packet header but with the help of snort we are able to check packet header and payload. In default configuration snort will work on the default rules that come as standard.

Install snort by using command terminal: apt-get install snort

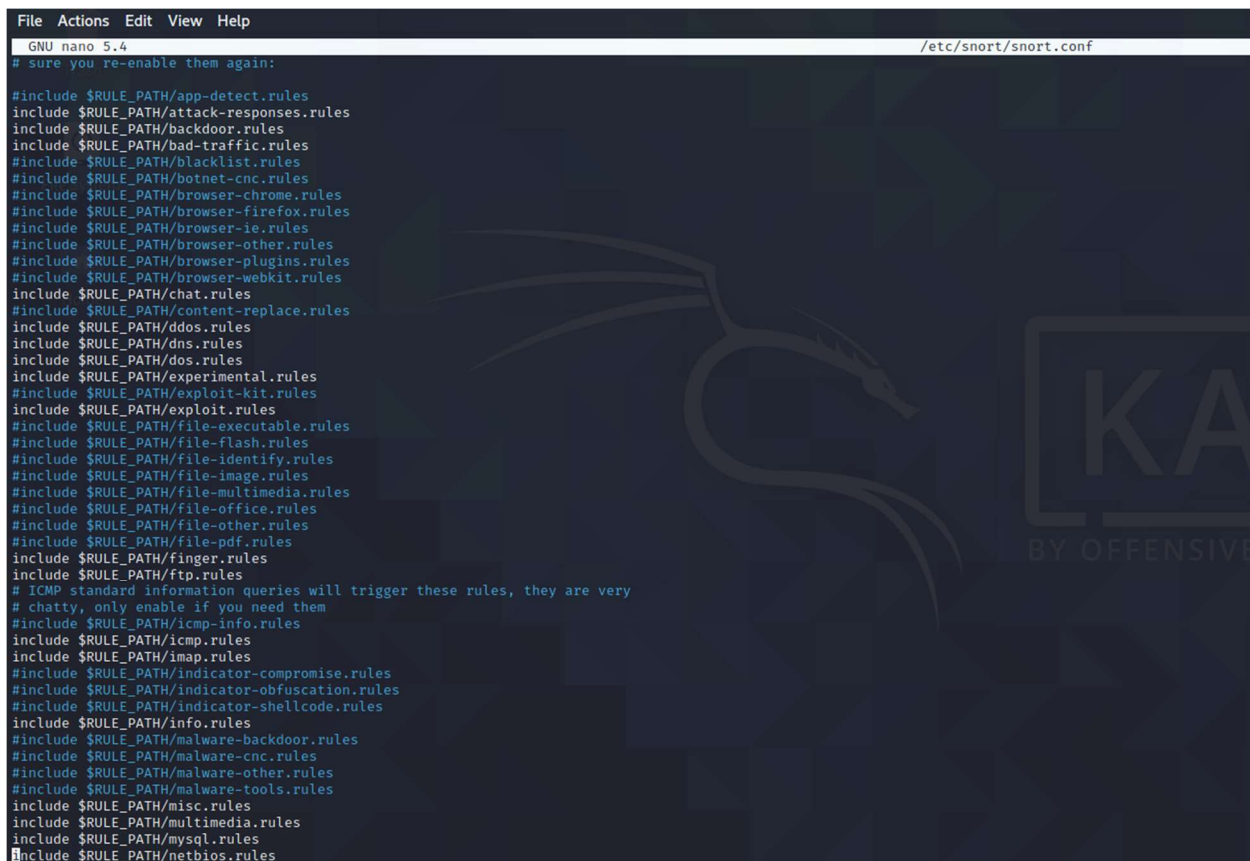


```
root@kali: ~
File Actions Edit View Help
● snort.service - LSB: Lightweight network intrusion detection system
   Loaded: loaded (/etc/init.d/snort; generated)
   Active: active (running) since Tue 2021-06-15 06:28:18 EDT; 5s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1266 ExecStart=/etc/init.d/snort start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 5840)
   Memory: 86.7M
      CPU: 478ms
   CGroup: /system.slice/snort.service
           └─1320 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g snort --pid-path /run/snort/ -c /etc/snort/snort.conf

Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Jun 15 06:28:18 kali snort[1320]:      Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Jun 15 06:28:18 kali snort[1320]: Commencing packet processing (pid=1320)
```

Figure 12: Snort

We can add rules in snort by including rule file in snort configure file. A user can add rules according to their need but for that they have to manually access the device and change the configuration file.



```
File Actions Edit View Help
GNU nano 5.4 /etc/snort/snort.conf
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/experimental.rules
#include $RULE_PATH/exploit-kit.rules
include $RULE_PATH/exploit.rules
#include $RULE_PATH/file-executable.rules
#include $RULE_PATH/file-flash.rules
#include $RULE_PATH/file-identify.rules
#include $RULE_PATH/file-image.rules
#include $RULE_PATH/file-multimedia.rules
#include $RULE_PATH/file-office.rules
#include $RULE_PATH/file-other.rules
#include $RULE_PATH/file-pdf.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
# ICMP standard information queries will trigger these rules, they are very
# chatty, only enable if you need them
#include $RULE_PATH/icmp-info.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/imap.rules
#include $RULE_PATH/indicator-compromise.rules
#include $RULE_PATH/indicator-obfuscation.rules
#include $RULE_PATH/indicator-shellcode.rules
include $RULE_PATH/info.rules
#include $RULE_PATH/malware-backdoor.rules
#include $RULE_PATH/malware-cnc.rules
#include $RULE_PATH/malware-other.rules
#include $RULE_PATH/malware-tools.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/mysql.rules
#include $RULE_PATH/netbios.rules
```

Figure 13: Snort Configuration File

4.4 Android Application

A very user-friendly and easy-to-use app is created for admin to manage and control the network. Android Studio platform is used in making of app. Application is built in java language. Socket programming is used to connect the app with the device (mini pc/raspberry pi) for enforcing the rules. Async task is used for socket programming to run in background and don't put extra load on main frame. Mostly relative layout is used for activities. Intent is used for communicating between the activity. For storage purpose, we used firebase database. Mobile app will send commands to device like BlockMAC 00:11:22:33:44:55 or BlockPort 22 and scripts at backend

will run accordingly blocking that specific MAC or port. For now, we have made app only for android. We will make iOS version in future.

Android App permissions:

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Internet and Network state permission is required for socket programming. External storage access permission is required for making pdf logs of network history and intrusion alerts.

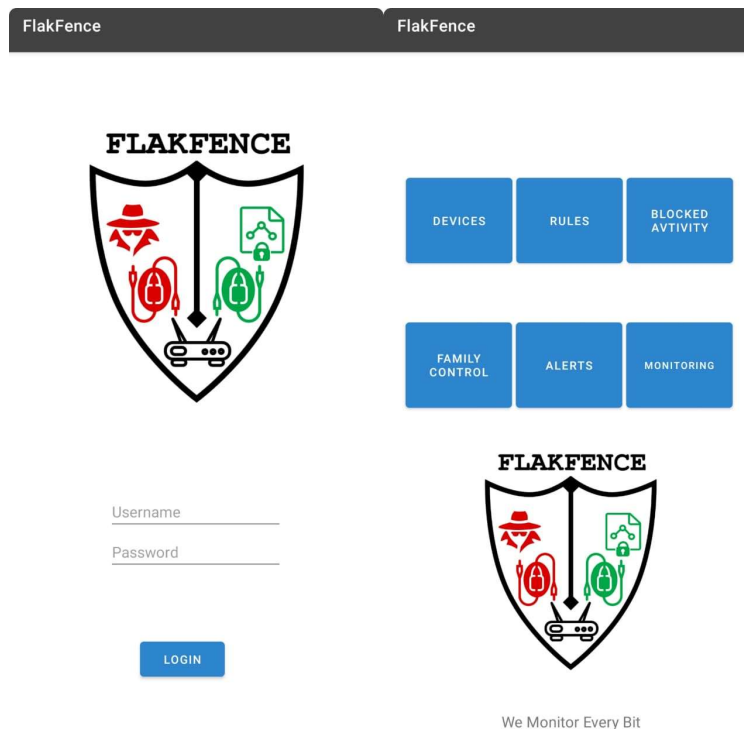


Figure 14: App Login and Menu Page

Chapter 5: Testing, Analysis and Results

5.1 Firewall Rules

5.2 Network Monitoring Logs

5.3 Connected devices on App

5.4 Intrusion Alerts

Chapter 5: Testing, Analysis and Results

This chapter describes the detailed results and outcomes of Flakfence.

5.1 Firewall Rules

When the site is gotten to, its URL is signed in the information base and checked whether that specific site is permitted or denied by the admin. If it is permitted, the requester will get the relevant page. Something else, evaluating hindered site will give an entrance denied page to the client. We are not only blocking websites, but we can also block ports and MAC-Addresses. This way we can completely block a user from accessing the internet. We can also block specific Ports with the help of IPTABLES.

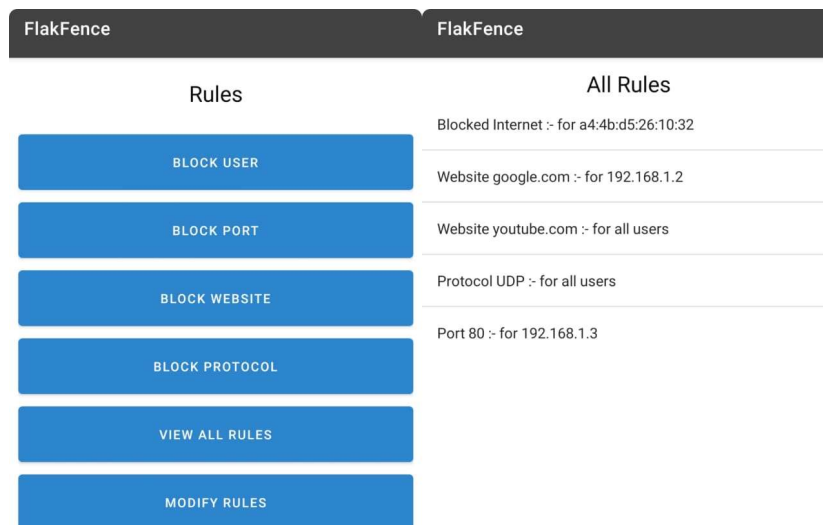


Figure 15: Firewall Rules

5.2 Network Monitoring Logs

After implementing a gateway, we made a python script in which we used Scapy and Pyshark. Scapy is a python library that can also be used from the command line. With the help of Scapy, we are able to analyze, and capture packets. Pyshark is basically a Wireshark in python it can be used to analyze the traffic coming in and going out of our device. We then sent this traffic to our Firebase database. We can now check the logs of both HTTP and HTTPS sites. All of these logs are then displayed in the application in a list view. Admin can view these logs on his phone.

Email Address	Email Address
Thu Jun 3 18:33:03 2021 : 192.168.1.3====> fjr04s07-in-f14.1e100.net	Thu Jun 3 18:33:37 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:26 2021 : 192.168.1.3====> fjr04s07-in-f14.1e100.net	Thu Jun 3 18:33:38 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:03 2021 : 192.168.1.3====> fjr04s07-in-f14.1e100.net	Thu Jun 3 18:33:38 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:03 2021 : 192.168.1.3====> fjr04s07-in-f14.1e100.net	Thu Jun 3 18:33:38 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:03 2021 : 192.168.1.3====> fjr04s07-in-f14.1e100.net	Thu Jun 3 18:33:39 2021 : 192.168.1.3====> fjr04s07-in-f10.1e100.net
Thu Jun 3 18:33:26 2021 : 192.168.1.3====> lfbn-lil-1-1334-181.w90-110.abo.wanadoo.fr	Thu Jun 3 18:33:39 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:27 2021 : 192.168.1.3====> MyRouter.Home	Thu Jun 3 18:33:39 2021 : 192.168.1.3====> fjr01s01-in-f14.1e100.net
Thu Jun 3 18:33:27 2021 : 192.168.1.3====> MyRouter.Home	Thu Jun 3 18:33:40 2021 : 192.168.1.3====> fjr01s01-in-f14.1e100.net
Thu Jun 3 18:33:27 2021 : 192.168.1.3====> MyRouter.Home	Thu Jun 3 18:33:40 2021 : 192.168.1.3====> fjr01s01-in-f14.1e100.net
Thu Jun 3 18:33:28 2021 : 192.168.1.3====> MyRouter.Home	Thu Jun 3 18:33:40 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:28 2021 : 192.168.1.3====> fjr04s07-in-f3.1e100.net	Thu Jun 3 18:33:41 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net
Thu Jun 3 18:33:28 2021 : 192.168.1.3====> par10s22-in-f238.1e100.net	Thu Jun 3 18:34:03 2021 : 192.168.1.3====> wm-in-f188.1e100.net

Figure 16: Network History

5.3 Connected devices on App

We have used the library of the subprocess in python. We have also used ARP scanning and Firebase database. ARP scanner shows every active device on our subnet. It is a low-level tool that can be used to identify network assets. It used from a command line in Linux. ARP scanner shows both the IP Address and the MAC-Address of all the devices connected to the network. After successfully implementing the android.



FlakFence			FlakFence		
Name	IP address	MAC	Name	IP address	MAC
CHONGQING	192.168.1.3	40:23:43:d2:00:9b	CHONGQING	192.168.1.3	40:23:43:d2:00:9b
Xiaomi	192.168.1.2	a4:4b:d5:26:10:32	Xiaomi	192.168.1.2	a4:4b:d5:26:10:32
			Intel	192.168.1.5	40:23:43:D2:00:9B
			Huawei	192.168.1.7	40:23:44:D2:1B:9C

Figure 17: Connected Devices

5.4 Intrusion Alerts

We have used snort to make an intrusion detection system that would generate alerts and send them to the admin. Database is used to store the history of intrusions so that it can be displayed at mobile application. These alerts will be available on the android application of the admin.

FlakFence	FlakFence
Intrusion Alerts	Intrusion Alerts
Email Address	Email Address
CREATE AND SEND PDF	CREATE AND SEND PDF
Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3	Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3
06/03-18:26:13.650854 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3	06/03-18:16:22.407259 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3
06/03-18:26:13.650893 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3	06/03-18:17:22.332329 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:26:13.736656 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3	06/03-18:17:28.049243 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:26:19.745553 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.3	06/03-18:17:29.058536 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:28:44.526979 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 142.250.181.110 -> 192.168.1.3	06/03-18:18:30.507107 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:30:15.701857 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 103.200.36.146 -> 192.168.1.3	06/03-18:18:31.530360 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:30:38.223206 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.60 -> 192.168.1.3	06/03-18:18:32.563081 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3
06/03-18:30:45.124777 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.60 -> 192.168.1.3	06/03-18:18:33.545083 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 69.171.250.35 -> 192.168.1.3

Figure 18: Intrusion Alerts

Chapter 6: Enhancements & Future Work

6.1 IPS

6.2 VPN server

6.3 TOR BLOCKING

6.4 DPI (Deep Packet Inspection)

Chapter 6: Enhancement and Future Work

6.1 IPS

An intrusion prevention system (IPS) is a type of network security that attempts to distinguish and halt the recognized threats. Intrusion prevention systems continuously checks our network for any sort of malicious activity and also gather information about it. IPS then logs all of the information and also informs the admin. It is then up to the admin to check the report and take action against it. IPS solutions can also be used to check if rules are being violated in the offices about how their network should be used. It can identify issues with security policies of the organizations [18].

Nowadays hackers are becoming more and more sophisticated, and they are using more sophisticated tools to compromise the network. They are able to get past most of the basic security solutions present in the market. Hence IPS is very important to have in our product and this would make our product significantly better than the rest of the competition.

6.2 VPN Server

VPN Server would allow our administrator to remotely access the network and be a part of it. When our user is in a public place and uses the public Wi-Fi then they can connect to our VPN so that they are able to hide their private information. This would ensure maximum privacy for our user.

6.3 Blocking TOR

TOR utilizes a very complex Onion routing mechanism that is very difficult to block. We can block ports and websites but completely blocking only Tor is a complex thing to do. Blocking Tor is a very rough task. Its use of volunteer-ran nodes and relays used to reroute traffic and disguises and sort of changes the original IP addresses. Perhaps the easiest way to block TOR traffic would be find the list of Tor exit nodes and configure your firewall to block these nodes [19].

6.4 Deep Packet Inspection

Deep packet inspection (DPI) is an advanced method where the network traffic is examined and managed. It is a type of packet filtering in which data is located identified and classified. conventional packet filtering techniques only examine the headers of the packet. It is usually a feature that is embedded into next generation firewalls. Deep packet inspection works at the application level of the firewall. Deep packet inspection checks the contents of packets passing and then makes real time decision based on the rules set by the admin of the network. Previously available firewalls only checked the header of the packet which would mean that they only checked the outer layer of the packet and not what is inside it. This was because of the limitations present in the technology available. Firewalls did not used to have enough processing power to the check the packets deeply in real time. With the advancements in the technology, we can now perform deep packet inspection. Deep packet inspection can examine what is inside the messages and also specify which application or service the packet comes from [20]. Having DPI in our product would help us take it to a level where it can be categorized as a next generation firewall.

Appendix A

Android Manifest File

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.flakfence">

    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/extruded"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.FlakFence">

        <activity android:name=".FamilyControl"></activity>

        <activity android:name=".Alerts" />

        <activity android:name=".ModifyRules" />

        <activity android:name=".BlockWebsiteUser" />

        <activity android:name=".Monitoring" />

        <activity android:name=".ViewRules" />
```

```
<activity android:name=".BlockWebsite" />
<activity android:name=".BlockProtocolUser" />
<activity android:name=".blockportuseroption" />
<activity android:name=".Blockportoption" />
<activity android:name=".BlockProtocol" />
<activity android:name=".Blockportuser" />
<activity android:name=".BlockPort" />
<activity android:name=".BlockUser" />
<activity android:name=".Devices" />
<activity android:name=".Rules" />
<activity android:name=".Menu" />
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
</application>

</manifest>
```

Appendix B

Socket Async Task:

```
public static class Socket_AsyncTask extends AsyncTask<String, Void, Void> {  
  
    Socket socket1;  
  
    @Override  
  
    protected Void doInBackground(String... voids) {  
  
        try {  
  
            String ports = voids[0];  
  
            InetAddress inetAddress = InetAddress.getByName(SERVER_IP);  
  
            socket1 = new java.net.Socket(inetAddress, SERVERPORT);  
  
            PrintWriter writer1 = new PrintWriter(socket1.getOutputStream());  
  
            writer1.write(CMD);  
  
            writer1.write(ports);  
  
            writer1.flush();  
  
            writer1.close();  
  
            socket1.close();  
  
        } catch (IOException e) {  
  
            e.printStackTrace();  
  
        }  
  
        return null;  
  
    }  
  
}
```

Appendix C

PopUp Menu for selecting Rules:

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()){  
        case R.id.alluser:  
            map= prt[1];  
            Socket_AsyncTask block = new Socket_AsyncTask();  
            block.execute(map);  
            String x="Port"+map+" :- for all users";  
            ViewRules.incre_index();  
            new FirebaseRules(x);  
            return true;  
        case R.id.specuser:  
            Intent specuser= new Intent(this, Blockportuser.class);  
            String map1= prt[1];  
            specuser.putExtra("message", map1);  
            startActivity(specuser);  
            return true;  
        case R.id.AllUser:  
            Intent textrule= new Intent(BlockPort.this,Blockportooption.class);  
            String port1=port.getText().toString();  
            textrule.putExtra("message1p",port1);
```

```
startActivity(textrule);  
  
return true;  
  
case R.id.SpecUser:  
  
    Intent textrule1=new Intent(this,blockportuseroption.class);  
  
    String port2= port.getText().toString();  
  
    textrule1.putExtra("message",port2);  
  
    startActivity(textrule1);  
  
    return true;  
  
}  
  
return false;  
  
}
```

Appendix D

Alerts Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/newh"
>
<TextView
    android:id="@+id/alert"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/intrusion_alerts"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:textColor="#090909"
    android:textSize="18pt"/>
<Button
    android:id="@+id/pdf_alerts"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/alert"
```

```
        android:text="Create PDF" />
<ListView
    android:id="@+id/listviewalert"
    android:layout_below="@+id/pdf_alerts"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</RelativeLayout>
```

Appendix E

Fetching Data from database:

```
DatabaseReference reference_history= database.getReference().child("NW History");

reference_history.addValueEventListener(new ValueEventListener() {

    @Override

    public void onDataChange(@NonNull DataSnapshot snapshot) {

        arraylist_source.clear();

        for(DataSnapshot dataSnapshot: snapshot.getChildren()){

            Fetch_History fetch_history=dataSnapshot.getValue(Fetch_History.class);

            assert fetch_history!=null;

            String source=fetch_history.getSource();

            String dest=fetch_history.getDestination();

            arraylist_source.add(source+": "+dest);

        }

        arrayAdapter_source.notifyDataSetChanged();

    }

    @Override

    public void onCancelled(@NonNull DatabaseError error) {

    }

});
```


Appendix F

Connected Devices

```
import pyrebase

import subprocess

process = subprocess.Popen("arp-scan -I wlan0 -l", shell=True, stdout=subprocess.PIPE)

out = process.stdout.read()

output = out.decode("utf-8")

asset = output.split("\n")

length = len(asset)

config = {

    "apiKey": "AIzaSyB6rRK7Wyg6ICv5SnVhXYuXRPEkfxaLK74",

    "authDomain": "fireapp-5.firebaseio.com",

    "databaseURL": "https://fireapp-5.firebaseio.com/",

    "storageBucket": "fireapp-5.appspot.com"

}

firebase = pyrebase.initialize_app(config)

auth = firebase.auth()

db = firebase.database()

db.child("Connected Users").remove()
```

```
for i in range(2, length - 4):  
    connections = asset[i].split()  
    data = {  
        "name": connections[2],  
        "IP": connections[0],  
        "MAC": connections[1]  
    }  
    results = db.child("Connected Users").child(i-1).set(data)  
    print(data)
```

References

- [1] <https://firewalla.com/>
- [2] geekflare.com
- [3] <https://searchsecurity.techtarget.com/definition/intrusion-detection-system>
- [4] <https://www.veracode.com › security › arp-spoofing>
- [5] <https://www.hitechwhizz.com/2020/03/5-advantages-and-disadvantages-drawbacks-benefits-of-firewall.html>
- [6] <https://us.norton.com/internetsecurity-kids-safety-top-reasons-to-use-parental-controls.html>
- [7] <https://firewalla.com/>
- [8] <https://pfsense.com/>
- [9] <https://www.fortinet.com/tw/products/next-generation-firewall>
- [10] <https://components101.com/microcontrollers/raspberry-pi-3-pinout-features-datasheet>
- [11] <https://www.jetbrains.com/pycharm/>
- [12] <https://developer.android.com/studio/features>
- [13] <https://www.snort.org/>
- [14] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [15] https://en.wikipedia.org/wiki/Android_software_development
- [16] <https://medium.com/sysf/bash-scripting-everything-you-need-to-know-about-bash-shell-programming-cd08595f2fba#:~:text=Bash%20is%20the%20improved%20version,nano%20to%20edit%20a%20file.>
- [17] <https://www.thegeekstuff.com/2011/01/iptables-fundamentals/>

[18] <https://www.forcepoint.com/cyber-edu/intrusion-prevention-system-ips>

[19] <https://support.opendns.com/hc/en-us/articles/115005917723-Can-OpenDNS-Block-Tor->

[20] <https://digitalguardian.com/blog/what-deep-packet-inspection-how-it-works-use-cases-dpi-and-more>

Flakfence

ORIGINALITY REPORT

9%

SIMILARITY INDEX

7%

INTERNET SOURCES

4%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universidad Nacional de Colombia Student Paper	1%
2	www.greenfinch.com Internet Source	1%
3	Submitted to Muskego High School Student Paper	1%
4	xserver.app Internet Source	1%
5	Submitted to 8442 Student Paper	<1%
6	Submitted to Campbellsville University Student Paper	<1%
7	iugspace.iugaza.edu.ps Internet Source	<1%
8	Submitted to University of Derby Student Paper	<1%
9	medium.com Internet Source	<1%

10

Submitted to University of Wales central
institutions

Student Paper

<1 %

11

github.com

Internet Source

<1 %

12

Submitted to IIT Delhi

Student Paper

<1 %

13

www.bromosapien.net:8080

Internet Source

<1 %

14

repository.neelain.edu.sd:8080

Internet Source

<1 %

15

www.raspberrypi.org

Internet Source

<1 %

16

Submitted to University of Glasgow

Student Paper

<1 %

17

qiita.com

Internet Source

<1 %

18

Submitted to Bolton Institute of Higher
Education

Student Paper

<1 %

19

Submitted to University of Sunderland

Student Paper

<1 %

20

erepository.uonbi.ac.ke:8080

Internet Source

<1 %

21	lup.lub.lu.se Internet Source	<1 %
22	Submitted to Modern College of Business and Science Student Paper	<1 %
23	support.opendns.com Internet Source	<1 %
24	whatis.techtarget.com Internet Source	<1 %
25	www.powells.com Internet Source	<1 %
26	"Firewalling Your Hosts", Hardening Linux, 2005 Publication	<1 %
27	www.grin.com Internet Source	<1 %
28	uir.unisa.ac.za Internet Source	<1 %
29	rajangauchan10.wordpress.com Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off