USER-SIDE WI-FI HOTSPOT SPOOFING DETECTION ON ANDROID-BASED DEVICES

Lunodzo Justine Mwinuka

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of Master's in Wireless and Mobile Computing of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

June, 2022

ABSTRACT

Network spoofing is becoming a common attack in wireless networks. Similarly, there is a rapid growth of numbers in mobile devices in the working environments. The trends pose a huge threat to users since they become the prime target of attackers. More unfortunately, mobile devices have weak security measures due to their limited computational powers, making them an easy target for attackers. Current approaches to detect spoofing attacks focus on personal computers and rely on the network hosts' capacity, leaving users with mobile devices at risk. Furthermore, some approaches on Android-based devices demand root privilege, which is highly discouraged. This research aims to study users' susceptibility to network spoofing attacks and propose a detection solution in Android-based devices. The presented approach considers the difference in security information and signal levels of an access point to determine its legitimacy. On the other hand, it tests the legitimacy of the captive portal with fake login credentials since, usually, fake captive portals do not authenticate users. The detection approaches are presented in three networks: (a) open networks, (b) closed networks and (c) networks with captive portals. As a departure from existing works, this solution does not require root access for detection, and it is developed for portability and better performance. Experimental results show that this approach can detect fake access points with an accuracy of 98% and 99% at an average of 24.64 and 7.78 milliseconds in open and closed networks, respectively. On the other hand, it can detect the existence of a fake captive portal at an accuracy of 88%. Despite achieving this performance, the presented detection approach does not cover APs that do not mimic legitimate APs. As an improvement, future work may focus on pcap files which is rich of information to be used in detection.

DECLARATION

I, Lunodzo Justine Mwinuka, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this dissertation is my original work and that it has neither been submitted nor concurrently submitted for a degree or similar award in any other institution.

Lunodzo Justine Mwinuka

Name and Signature of Candidate Date

The above declaration is confirmed by:

Dr. Jema David Ndibwile

Name and Signature of Supervisor 1 Date

Prof. Shubi Felix Kaijage

Name and Signature of Supervisor 2 Date

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by The Nelson Mandela African Institution of Science and Technology, a dissertation entitled, "*User-Side Wi-Fi Hotspot Spoofing Detection on Android-Based Devices*" In partial Fulfilment of the Requirements for the Award of the Degree of Master's in Wireless and Mobile Computing of the Nelson Mandela African Institution of Science and Technology.

Dr. Jema David Ndibwile

Name and Signature of Supervisor 1 Date

Prof. Shubi Felix Kaijage

Name and Signature of Supervisor 2 Date

ACKNOWLEDGEMENTS

Firstly, I would like to extend my gratitude to my supervisors: Dr. Jema David Ndibwile and Prof. Shubi Felix Kaijage, whose expertise and moral support are invaluable from the conception of the research topic to results reporting. Their insightful feedback and constant follow-up shaped the quality and pushed to the timely accomplishment of this work. I feel lucky having been guided by supervisors of high integrity and passion for mentorship in research activities.

Secondly, I thank the Nelson Mandela African Institution of Science and Technology (NM-AIST) and the Ministry of Education, Science and Technology (MoEST) of the Government of Tanzania for a chance to study the Master's degree in Wireless and Mobile Computing (WiMC) and finance my studies. Moreover, I equally thank Mzumbe University (MU) for granting me a two-year study leave. It is evident that without these resources, my studies would have remained to be a dream.

Thirdly, I thank all the people who supported and participated in this study in one way or the other. In a very humble way, I thank my partners in research: Mr. Abel Z. Agghey and Loyani K. Loyani for their tireless efforts to see this work at its maturity. I equally register my acknowledgements to Ms. Rosemary T. Panga, Mr. Evance Nganyage, Mr. Baltazary Msoma, Ms. Aurila Daniel, Mr. Lugano Mwakapuku, and my WiMC 1st cohort classmates for their kind support.

The love, support, and motivation from my family made the process painless. I, therefore, thank my parents and brothers: Mr. Edrick and Manase Mwinuka, for their tireless support, cheers and motivation. These are the people who encouraged and believed in me even when I was falling to pieces.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

ACK	Acknowledgement
AP	Access Point
ARP	Address Resolution Protocol
AUTH	Authentication
BPSK	Binary Phase-shift keying
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
BYOD	Bring Your Own Device
CLI	Command-line Interface
CSV	Comma Separated Value
CRC	Cyclic Redundancy Code
DA	Destination Address
DB	Database
dB	Decibel
dBm	Decibels per Milliwatt
DFD	Data Flow Diagram
DNS	Domain Name System
EAP	Extensible Authentication Protocol
ENC	Encryption
ESSID	Extended Service Set Identifier
ETA	Evil-Twin Attack
FCF	Frame Control Field
FCS	Frame Check Sequence
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
ID	Identification
IDE	Integrated Development Environment
IDS	Intrusion detection system
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
LAP	Legitimate Access Point
MAC	Media Access Control

Mbps	Megabits per second
MITM	Man In The Middle Attack
ML	Machine Learning
MU	Mzumbe University
NFC	Near-Field Communication
NIC	Network Interface Card
OBSS	Overlapping Basic Service Set
OS	Operating System
OSI	Open Systems Interconnection
PC	Personal Computer
PCAP	Packet Capture
PDA	Personal Digital Assistant
PNL	Preferred Network List
RA	Receiver Address
RAD	Rapid Application development
RAP	Rogue Access Point
RDBMS	Relational Database Management System
RF	Radio Frequency
RIP	Routing Information Protocol
RSSI	Received Signal Strength Indicator
RSSID	Received Signal Strength Indicator Daemon
SA	Source Address
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SDN	Software Defined Network
SQL	Structured Query Language
SSID	Service Set Identifier
STA	Station
TCRA	Tanzania Communications Regulatory Authority
TA	Transmitter Address
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus
VM	Virtual Machine

VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WSN	Wireless Sensor Network
XML	eXtensible Markup Language
XP	Extreme Programming
XML	eXtensible Markup Language

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

The global mobile population on the Internet is rapidly expanding, counting four (4) billion unique users in the world population as of January 2021 (Johnson, 2021; O'Dea, 2021). Mobile devices account for 48% of online page views worldwide; Africa and Asia lead the list by 60.57% and 60.19% respectively, due to their wide coverage. For example, Nigeria has higher rates of 84% of Internet traffic from mobile devices than United States of America (USA) which has 47.28% (O'Dea, 2021). Generally, mobile Internet traffic shares 50.44% of global online traffic (Cisco, 2020).

Mobile devices are also preferred for usage in education (Joyce-Gibbons *et al.*, 2018; Suryasa *et al.*, 2020) and for communication and e-commerce (Einav *et al.*, 2014) due to the information they store (Bitton *et al.*, 2018) and their ease of use, portability, and reliable functions. In addition, most mobile phone users opt for wireless networks to access the Internet (Mahadevan & Kaleta, 2018). As a result, the use of wireless networks is exponentially growing (Cisco, 2020). Internet services, voice-over wireless, health care, education, and agriculture services all use it.

From 2020 to 2023, the use of wireless communications is expected to skyrocket exponentially (Cisco, 2020). AbiResearch (2021) estimated that twenty billion wireless devices were shipped, and 9.5 billion Wi-Fi networks were installed in 2018. Similarly, according to Cisco (2020), the average number of Internet-connected gadgets per person worldwide was 2.4 in 2018. The number is predicted to rise to 3.6 in 2023. The devices cover home networking, retail application, and critical business operations. Technology companies attempted to launch Wi-Fi-based projects for remote areas to attain fair distribution of services. For example, Microsoft's Rural Airband Initiative and Google's Project Loon attempted to address the digital divide using wireless communications. According to Cisco (2020), Wi-Fi hotspots are expected to grow four-fold from 2018 to 2023, counting to 628 million public Wi-Fi hotspots by 2023, up from 169 million hotspots in 2018.

Security risks occur as wireless communication becomes incorporated into important commercial operations involving financial, personal data and as it becomes part of digital routines, including Internet access. Wireless routers actively broadcast the unencrypted beacon

packets to associate a client access point (AP), making the situation even more worrisome (Lazos & Krunz, 2011). Thus, wireless security has become more vital to Internet users as it becomes the backbone of the Internet connection. As a result, billions of money are spent implementing wireless security, and billions of money are lost when security measures fail (Guo & Computing, 2019). Furthermore, connecting to an unsecured Wi-Fi hotspot poses high risks as attackers could easily access personal and financial records, which could be used to perform other attacks (Kidston & Li, 2010).

One possible attack on wireless communications is the spoofing attack, sometimes referred to as KARMA (Jindal *et al.*, 2014), rogue access point (RAP), evil twin attack (ETA) (Rech, 2012), or network spoofing attack (Mahadevan & Kaleta, 2018). Spoofing attacks on the internet work in an environment where information is transmitted between network users who are identified by Internet addresses. A successful spoofing attack's sender or receiver address is disguised to appear legitimate. As a result, the receiver does not notice the sender's valid address, and the sender sends packets to a bogus or spoofed address (Wolfe *et al.*, 2018). An attack occurs when an attacker successfully creates illegitimate Wi-Fi APs in wireless networks, and a user connects to it. Rogue Wi-Fi hotspots are one of the simplest ways for attacking users in organisations, Internet cafés, universities, airports and other public places. Hence this type of assault is considered risky. However, despite its simplicity, it has far-reaching consequences for users as bad as any other spoofing assault (Jindal *et al.*, 2014). KasperskyLab (2020) report mentions network spoofing as one of the major mobile security threats since they give room for many other forms of attacks in a network (Seigneur, 2017; Shrivastava *et al.*, 2020; Srinivas *et al.*, 2013).

In the Wi-Fi Hotspot spoofing attack, an attacker creates an open hotspot with a name similar to host organisation or common public Wi-Fi or sometimes assigns deceiving names such as FastWiFi, OpenAccess5G, and other similar names. Alternatively, attackers may de-authenticate a user from AP and suppress the original AP signals while boosting theirs with a duplicate AP name. This approach allows users to attempt re-authentication with the RAP (Kropeit, 2015). In other forms, ETA could be created to mimic the Service Set Identifier (SSID) and Media Access Control (MAC) of the legitimate AP (Kropeit, 2015). The MAC is sometimes referred to as Basic Service Set Identifier (BSSID). Another approach is to broadcast the SSIDs of a user's wireless location (Park *et al.*, 2014; Wang *et al.*, 2015). Similarly, attackers broadcast fake AP, which will make it appear as if a user is connected to a Wi-Fi hotspot even if they are not (Tchakounté *et al.*, 2020). Attackers further use AP

information from a device's preferred network list (PNL) to generate phoney AP tricking devices to connect (Chatzisofroniou, 2018).

Most users prefer free Wi-Fi over their typical data plans because they do not want to waste their Internet resources (KasperskyLab, 2020). A report by O'Dea (2021) shows that 37% of smartphone users would connect to free public Wi-Fi, while only 8 % would connect to paid public Wi-Fi. The other 55% would connect to either of the two. Since majority users usually connect to any free Wi-Fi, their association with illegitimate hotspots poses a high risk. An attacker could intercept data and inject malware into a connected device (NortonLifeLock, 2019).

"Bring Your Own Device" (BYOD) is on the rise as organisations allow their workers and clients to bring personal mobile devices such as smartphones and tablets into workplaces. The BYOD is stimulated by the growing ability of users to use mobile technologies, existing infrastructures, erratic Internet connection, and even unreliable power supply. The growing use of mobile devices increases the chances and risks of Wi-Fi hotspot spoofing (Jindal *et al.*, 2014). Furthermore, since mobile devices have small computing resources, they do not offer the same level of built-in security mechanisms as desktop computers (KasperskyLab, 2020; Ndibwile *et al.*, 2017). As a result, mobile devices have become the target to most attackers as they are widely used globally, and according to the sensitivity of the information, they store (Bitton *et al.*, 2018; Oh *et al.*, 2014).

Despite the efforts made to ensure users' safety in wireless communication, companies and security professionals still have no confidence in BYOD. Lower confidence is reported since 61% are not confident with safety, and 67% of organisations are not certain about their ability to prevent wireless attacks (KasperskyLab, 2020; Outpost24, 2020). These facts alarm users about wireless communications security threats, especially the risks against network spoofing attacks, which are among the top-rated mobile threats.

1.2 Statement of the Problem

Mobile devices have become the prime target of wireless based assaults. This is stimulated by various reasons, including the rapid growth in wireless communications, the growing number of users, and the sensitivity of data they hold (Bitton *et al.*, 2018; Park *et al.*, 2014). In addition, since wireless services are available at workplaces and other public places, Internet users prefer wireless connections to get free and convenient Internet service (KasperskyLab, 2020; NortonLifeLock, 2019; Symantec, 2017).

Threats for hotspot spoofing attacks are increasing. The threats come on the rise as users do not either pay much attention or have no control in identifying legitimate Wi-Fi hotspots when associating with them at their workplaces or open networks (Prasad & Rohokale, 2020; Symantec, 2017). Furthermore, mobile devices put users at high risk due to their wide coverage of user space. The coverage and usage convenience make users the prime target for these attacks (Garg & Baliyan, 2020). Moreover, the devices have limited computing capacities, limiting their potential to implement certain security functions. For instance, most of the security measures implemented in Desktop computers may not be implemented in mobile devices (Ndibwile *et al.*, 2017).

Usually, for a mobile device to establish a connection with a wireless AP, it employs an active scan which goes through three basic steps: (a) the discovery stage, (b) authentication, and (c) association (Kropeit, 2015). Initially, a client device sends a probe request to join the network, and the AP replies with a probe response. Finally, the client acknowledges and establishes a connection (Jaisinghani *et al.*, 2018). However, this technique does not distinguish between valid and illegitimate APs in open networks. Instead, the association procedure employs authentication mechanisms by exchanging keys against clients associating with them (Iftheker, 2008). As a result, clients could connect to any open hotspot due to the desperate need for Internet access. Attackers could take advantage of this flaw by deceiving clients to connect to their hotspots and then use that connection to launch more attacks (Bernaschi *et al.*, 2008; KasperskyLab, 2020).

In authenticated Wi-Fi APs, network hosts list devices and user accounts where users are authenticated based on registered information. Attackers may create a fake hotspot with a fake captive portal to trick network users into achieving authentication procedures. As a result, the users are compelled to submit their login credentials to attackers, who may later launch other attacks other services such as banking, which involves personal information (KasperskyLab, 2020). Because of the small size of the screen, distinguishing fake captive portals from legitimate ones on mobile phones is much more difficult than on desktop/personal computers (PC) (Ndibwile *et al.*, 2019). With a plethora of online tools, such as the Kali Software Engineering Toolkit (Pavković & Perkov, 2011), any web page can be cloned with the same quality as the original with little effort. As a result of this shift, and with a combination of web page cloning and DNS poisoning, even desktop users could become victims. Despite efforts to prevent mobile devices from connecting to the RAP, spoofing attacks are still possible due to

the circumstances or mobile users' carelessness with wireless communications (Prasad & Rohokale, 2020).

Little has been done to detect hotspot spoofing on Android devices without relying on the network host's capacity. The details on the host side that are used to detect intrusions may be difficult to be accessed at the end-user device. Furthermore, approaches that rely on network host capacities are challenged when securing end-users is required since they demand all users to be registered in the network, which is impractical to places with huge traffic of guest users. Hence, there is a shortage of applications to detect fake APs on the users' side. This study proposes an Android application prototype that detects hotspot spoofing attacks created by the fake wireless hotspots or ETA in wireless networks. The study recognises the importance of preventing spoofing attacks before the mobile device-to-AP association since the association procedure does not include mechanisms to detect fake APs. The detection could be done by determining the legitimacy of AP using details collected from their broadcasts (probe responses).

1.3 Rationale of the Study

As security concerns come on the rise and digital devices become vulnerable to security attacks, users and corporate data are prioritised in the digital components list to be protected. Android-based devices are not an exception as they cover the majority of smartphone users. Their portability makes users store sensitive information in them, including personal and banking information. Unfortunately, they cannot accommodate security features implemented in desktop PCs due to their limited computational powers. In recent years there has been an increase in damaging attacks on Android devices, posing a serious challenge to the platforms. Hence raising the need for developing countermeasures to secure the mobile systems against these assaults.

Many research works have been done to justify the impacts of spoofing attacks on wireless networks and wireless users. Some studies demonstrated how spoofing could facilitate other attacks, including MITM and phishing, as explained by Kaspersky (2021) and Kidston and Li (2010). This study covers hotspot spoofing detection on Android devices as an understudied area. Our prototype would facilitate Android devices' built-in capabilities to identify the legitimacy of Wi-Fi hotspots before associating with them.

1.4 Research Objectives

The study was guided by three specific objectives which contribute to the main objective. All the objectives were also subjected to respond to the questions in Section 1.5.

1.4.1 General Objective

The main objective of this research was to detect spoofed Wi-Fi hotspots on Android-based devices.

1.4.2 Specific Objectives

The specific objectives of this research were:

(i) To investigate Android users' knowledge and practices in Wireless networks and efforts to identify attacks on Wi-Fi hotspots.

(ii) To develop an Android application that detects spoofed Wi-Fi hotspots.

(iii) To validate functional requirements of the developed application.

1.5 Research Questions

The study was seeking to answer the following questions:

(i) What levels of effort do users put into identifying possible threats on wireless networks?

(ii) What are the requirements for developing a fake hotspot detection application?

(iii) How can a fake hotspot detection application be developed?

(iv) How effective is the developed application in detection the presence of fake hotspots in the perimeter?

1.6 Significance of the Study

Data safety remains on top of the list of security requirements amongst organisations and cybersecurity experts. However, various attacks exist that target data stored in organisations. Network related attacks are in the frontline among the attacks for data privacy. Network spoofing is among the top three mobile threats as of 2020 (KasperskyLab, 2020). According to Graham (2018), Internet research found 30 000 spoofing attacks each day on unique devices

between 2015 and 2017 alone. Furthermore, the attacks target personal data as they facilitate several other attacks.

The detection of fake APs on wireless networks would help to improve Android phones' safety and impact cybersecurity knowledge to users, particularly security on Wireless communication. Organisations can also take advantage of this study to enforce policies that the users of the organisation's network have to follow.

The solution could also be integrated into Android's wireless system services (part of Android OS kernel) to improve mechanisms in Android-to-hotspots association for safety and resilience. The research also gives a foundation for development of mitigation approaches against fake APs.

1.7 Delineation of the Study

This research is limited to Android-based devices. This category covers a large population, and it is at high risk due to the complexities of implementing security measures. The study also was conducted on the university campus networks. They facilitated easy access to the required population and support for infrastructure to simulate and test our solution. University campuses have a pool of users ranging from students to staff from diversified demographic profiles. The proposed prototype relies solely on the power of Android built-in capabilities and functionality, which limits the number of features extracted for the detection process. The detection of attacks is not enough to ensure the safety of wireless users. This scenario calls for the need for preventive measures of similar attacks.

CHAPTER TWO

LITERATURE REVIEW

2.1 Wireless Communication

Wireless communication is among the fastest-growing technologies in the communication field. It has been in place for more than 100 years (Seymour & Shaheen, 2011). The invention started with wireless telegraphy in 1897 by sending EM waves for a short distance of 100 meters (Nassa, 2011). Its evolution went over years where the world experienced changes in radio communications. Since then, wireless has advanced rapidly (Tse & Viswanath, 2005). The Institute of Electrical and Electronics Engineers (IEEE) developed the 802.11 protocol for wireless transmission to which wireless operates (Crow *et al.*, 1997). This protocol defines Wireless Local Area Networks (WLANs) for the communication between stations. It was first released in 1997 with transmission rates of up to 2 Mbit/s (Rech, 2012). The preceded standards were identified from its series of trailing alphabetic characters, i.e., 802.11b, 802.11g, 802.11n, 802.11ac, 802.11ax. The earlier versions use 2.4 GHz divided into 14 channels, separated 5 MHz in each (Kropeit, 2015). There exist limitations of channel use in different countries depending on their national regulations. These regulations are specified by the Tanzania Communications Regulatory Authority (TCRA) in Tanzania.

The 802.11 n can attain transfer speeds of up to 600 Mbit/s with many antennas, whereas 802.11 ac can exceed 1 Gbit/s. 802.11 n also employs a frequency of 5 GHz and 2.4 GHz, but 802.11 ac only uses the latter (Kropeit, 2015; Verma *et al.*, 2013). The current 802.11 ax, also known as High-Efficiency Wi-Fi and marketed as Wi-Fi 6, is designed to operate in license-except bands between 1 and 7.125 GHz. The Wi-Fi 6 operates in 2.4 GHz and 5 GHz bands. Its closer standard, the Wi-Fi 6E, operates at 6 GHz (Sidhu *et al.*, 2007). More comparisons between Wi-Fi standards are presented in Table 1. Several features are shown for each standard. Some features play an important role in developing intrusion detection systems (IDS) in WLANs.

The WLANs are compatible with Ethernet-based wired connections, and many networks employ a combination of the two protocols. Like wired networks via Ethernet, wireless networks separate the transferred data into packets classified into three categories, Data, Control and Management. Data frames contain datagrams of upper layers in the OSI model (Ciurana *et al.*, 2007). Control messages contain traffic control information that can be utilised to avert collisions (Lopez-Aguilera *et al.*, 2004; Tang & Gerla, 2000). Management frames, the

third category, are being used during network discovery and association (Malekzadeh *et al.*, 2007). It is the major frame type on which the foundation of this study is built.

"Wi-Fi ready" is a term used to describe wireless equipment. Wi-Fi is a trademark used to identify products compatible with IEEE 802.11 networks (Rech, 2012). Wireless Fidelity is not an acronym for Wi-Fi. The "Wi-Fi – The Standard for Wireless Fidelity" was an early marketing slogan contributing to this misperception (Kropeit, 2015). In wireless communications, one of the details that the stations broadcast for them to associate with others are the SSID. This information is sent during the transmission of the probe. The SSID represents the network's name, such as "LunoAP" or "NM-AIST Wi-Fi". These names are used to identify wireless APs in the perimeter. The SSID is a string of up to 32 characters encoded in 7-bit ASCII. Furthermore, every station, client and AP, is identifiable by its MAC address, also known as BSSID. In most cases, it is not shown to end-users unless special tools such as wireless sniffers are used (Iftheker Mohammad, 2008).

Table 1: Comparison of primary IEEE 802.11 specifications

	802.11a	**802.11b**	**802.11g**	**802.11n**	**802.11ac**	**802.11ax**
Standard approved	1999	1999	2003	2009	2014	2019
Maximum data rate	54 Mbps	11 Mbps	54 Mbps	600 Mbps	1.3 Gbps	10-12 Gbps
Modulation	OFDM	DSSS or CCK	DSSS or CCK or OFDM	DSSS or CCK or OFDM	256 QAM, OFDM, MIMO, QPSK, BPSK, MU-MIMO	1024 QAM, TWT, BPSK, OFDMA, BSS Colouring, MU-MIMO
RF Band	5 GHz	2.4 GHz	2.4 GHz	2.4 GHz or 5 GHz	2.4 GHz or 5 GHz	2.4 GHz or 5 GHz
Number of spatial streams	1	1	1	1,2,3, or 4	1,2,3, or 4	1, 2, 3, 4, 5, 6, 7, or 8
Channel width	20 MHz	20 MHz	20 MHz	20 MHz or 40 MHz	80 MHz or 160 MHz	20, 40, 80, 80+80, 160 MHz

2.2 The 802.11 Management Frames

Stations (STA) use management frames in 802.11 to join and leave a Basic Service Sets (BSS). They have a MAC header with three addresses fields in it. For example, if it is 802.11a/b/g, it

has a 24-byte MAC header. In contrast, the 802.11n management frame has 28 bytes (additional 4 byte HT control field) MAC header (Jiang *et al.*, 2013). There are many management frame subtypes defined by 802.11, as presented in Table 2

Table 2: Management frame subtypes

Subtype bits	Subtype description
0000	Association request
0001	Association response
0010	Re-association request
0011	Re-association response
0100	Probe request
0101	Probe response
1000	Beacon
1001	Announcement Traffic Indication Message (ATIM)
1010	Disassociation
1011	Authentication
1100	Deauthentication
1101	Action

Aung and Thant (2017)

The format of the 802.11 management frame is structured of the duration of the frame, sequence control, management frame body, the frame check sequence (FCS) and others. The structure can generally be grouped into MAC header, body and FCS (Malekzadeh *et al.*, 2007). The structure of the management frame is presented in Fig. 1. These details can be captured, read and analysed using network analysis tools.

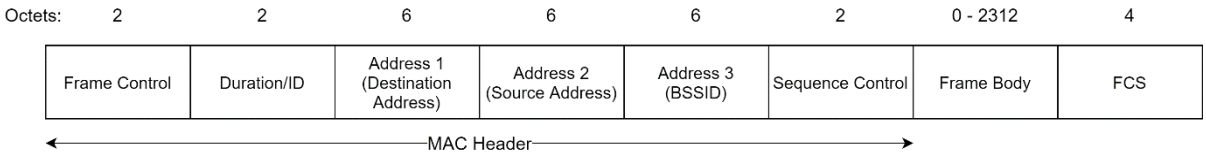


Figure 1: 802.11 Management frame structure

The MAC header contains the frame control, a duration, address (1 - 3), and sequence control fields. On the other hand, the frame body contains information specific to the frame type being carried. For example, a 32-bit cyclic redundancy code is the FCS. Additionally, the MAC frame format comprises a set of fields that occur in a fixed order in all frames. Clients and stations on wireless networks generate all these details. Since STA uses management frames to join and leave networks, attackers use this benefit by disguising the packet information. This may be done by modifying the information of the transmitted packets. On the other hand, defensive and detection approaches may be developed by detecting modification of management frames of the packets. Details of management frame components are presented in the following subchapters.

2.2.1 Frame Control Field

The frame control field (FCF) shows the frame type, whether a control, management, or data frame, and provides control information. As presented in Fig. 2, the control information includes whether the frame is to or from a destination, fragmentation information, and privacy information. The FCF consists of a version of the protocol, power management, subtype and type, more fragments, Wired Equivalent Privacy (WEP), and other fields (SA, 2021). The destination addresses could be manipulated for the rogue AP to appear legit during the association steps. For instance, an attacker could send a de-authentication packet to disassociate a client from the current connection. Usually, these packets are sent from a fake MAC address.

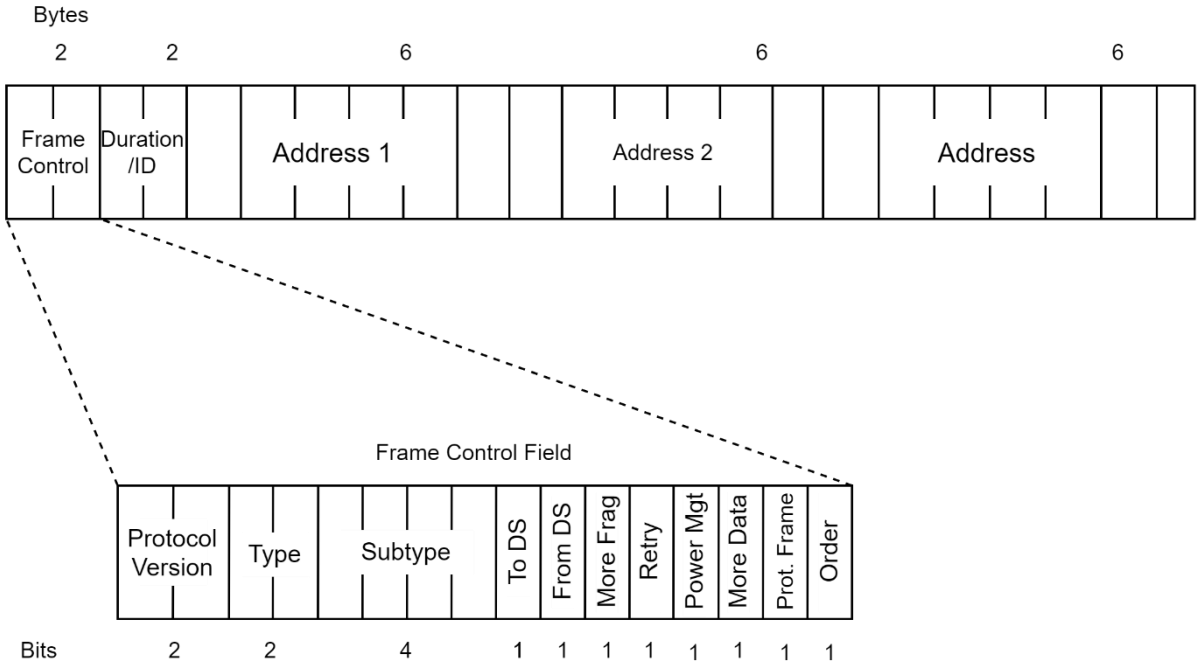


Figure 2: Frame Control Field/subfield

2.2.2 Duration/ Identification

This is a 16 bits field. If used as a duration field, it indicates the time (in microseconds) the channel will be allocated to transmit a MAC frame successfully. In some control frames, this field contains an association or connection identifier. The Duration/ID field, for example, in control type frames of subtype Power Save (PS)-Poll, contains the association identity of the node. The identity sent the frame in the 14 least significant bits, with the two most significant bits both set to 1. From 1 through 2007, the association identity value ranges from 1 to 7. This field holds duration values as stated for each of the frames in all other frames. This field is set to 32 768 for all frames transmitted during the contention-free period. This field updates the

Network Allocation Vector (NAV) if the value is less than 32768 (Jiang *et al.*, 2013). Data frames transmitted between APs are of no difference; they contain a duration ID that holds the time to which the STA will stay in a communication channel.

2.2.3 Address Fields

In an 802.11 WLAN MAC frame, there are four address fields. The BSSID, source address, a destination address, transmitting station address, and receiving station address are the fields that the WLAN MAC frame explain. These fields are not necessarily present in all of the frames. The number and context of the 48-bit address fields depend on context (Yu *et al.*, 2020). The BSSIDs of stations joined the BSS that transmit and receive frames over the WLAN are used to send and receive packets from other communicating STA. These IDs are also used in broadcast and multicast.

The SSID identifies the wireless LAN in which a frame is transmitted. In the case of an IBSS, the SSID represents a random number generated when the network is formed. For a WLAN that is part of the main configuration, the SSID identifies the BSS over which the frame is transmitted; specifically, the SSID is the MAC-level address of the AP for the BSS. Finally, the source and destination addresses are the MAC addresses of stations, wireless or otherwise. The source address may be the same as the transmitter address. In contrast, the destination address may be similar to the receiver address (Tang & Gerla, 2000). Addresses are unique identifiers of stations in the network. Unfortunately, attackers could still disguise this information to their benefit.

2.2.4 Beacon Frames

The beacon frames are one of the management frames in 802.11. Access Points and stations use beacons to communicate throughout the perimeter. It is used to determine the characteristics of the connection offered to clients. Clients also use the beacon frames when connecting to the network they had once connected (Kwak *et al.*, 2012). Beacons are sent within a period called Target Beacon Transmission Time (TBTT). It has a timestamp, interval, capability information, SSID and supported rates as its mandatory fields (Gupta & Rohil, 2013). Therefore, the beacon frames present very useful information that could be used to determine the legitimacy of the APs.

2.2.5 Probe Requests and Responses Frames

Wireless stations and clients in the network are associated with the probes. The STA does active scanning during the discovery process by sending a probe request management frame asking for available networks in the channel (Gupta & Rohil, 2013). It is usually sent to the destination address ff:ff:ff:ff:ff:ff. When probe requests are sent, available networks reply with the probe response indicating all information for a client to associate. Once the STA receives the probe response, it should send an acknowledgement frame (ACK) to the associating AP.

2.3 Practical Details of 802.11

The majority of the details of the wireless standard may be found in the various protocol versions and revisions. On the other hand, few implementation specifics are not defined and are left to the manufacturers. Furthermore, other characteristics, such as the wireless interface's mode of operation, are not specific to WLANs and are not described in IEEE wireless standards (Kropeit, 2015).

As explained before, a wireless device can be used in different modes, managed, promiscuous, and monitor. The default setting in the majority of devices is the managed mode. The mode enables clients to connect to a wireless network. The client must additionally connect to a network in promiscuous mode (Malekzadeh *et al.*, 2007). All transmitted packets, including those meant for other clients, can also be handled. Monitor, the third mode, is similar to promiscuous mode but is not limited to one network. Every packet received can now be processed regardless of the originating network (Gupta & Rohil, 2013). Those packets, however, come from an encrypted network, and just part of it can be read. Monitor mode is generally chosen over promiscuous mode in practice because it processes more data and is in line with a wider range of devices. However, because the interface is set to a particular frequency, data can only be captured on one channel.

Channel hopping is one technique to get around this. The gadget usually changes its channel in a predetermined interval, mostly between 200 and 250 milliseconds, utilising the channel hopping technique (Rech, 2012). This is necessary for determining which channels are in use. Continuous data transfers, on the other hand, are not possible to record using this method since the capture process has chances to be interrupted when the channel is switched.

Because WLANs must cover a large area, such as university-wide or company-wide networks, devices must be able to transit between various APs effortlessly. WLAN roaming can help

achieve this. However, the 802.11 specifications do not define the technical implementation of roaming, i.e., the client's condition to change the AP (Rech, 2012). Except for the frequency, all APs must be configured identically in all settings. The frequency is not a necessary configuration, but it is good to minimise interference. Because client devices only recognise WLANs based on their SSID, the AP's MAC address is disregarded. The authentication and association process is not abbreviated on consumer networks, i.e., WLANs using a Pre-Shared Key (PSK). The only advantage is a reduced configuration effort. A re-association is required on enterprise networks that deploy the authentication server since the device has been authenticated successfully.

Based on this review, another relevant consideration is that management frames mostly do not provide authentication services. Hence, beacon, probe request, probe response, de-authentication and many other frames can be manipulated. As a result, an attacker can create and send these frames with any content they want. A Denial of Service (DoS) attack can be launched by sending a large number of these packets. The problem is particularly in the context of de-authentication frames. An attacker may create forged frames and disconnect clients at any time. In efforts to address this, the main goal of task group 802.11w was to work on the issue of spoofing in management frames by injecting a Message Integrity Code (MIC) (Walker, 2009). An additional 4-way handshake derives the required shared secret. Unfortunately, the standard is not popular, and the task committee that oversees it has not updated the draft since May 2009 (Walker, 2009). This leaves the need for more spoofing detection approaches, especially solutions targeting specific user groups. Furthermore, the wireless devices have different behaviours and are classified between different 802.11 standards based on host organisations.

2.4 Mobile Phone Users' Knowledge and Practices on Wi-Fi

Smartphone security, especially on Android-based devices, has been well studied. The majority of previous literature focuses on threats and malware described by Breitinger *et al.* (2020). Recent studies bring the users knowledge and practice into scrutiny. A survey by Breitinger and Nickel (2010) assessed user security practices on mobile devices. These studies had discovered poor security practices in mobile devices because of low-security awareness and partly due to the low acceptance rate of authentication measures.

On the other hand, Breitinger *et al.* (2020) conducted a study to explore users awareness, choices and education on cybersecurity. This study found that most users have recommended

lock screen settings. However, they ignore other security practices like virtual private networks (VPN) when connecting to public Wi-Fi. In the study by Breitinger and Nickel (2010), it was found that 86% of respondents did not implement any authentication measures like a PIN to access their phone. A similar survey was done by Imgraben *et al.* (2014), where it was found that users are not aware of the risks posed when they leave their wireless and Bluetooth open. This study recommended that education and awareness programs would essentially address misconceptions and usage behaviours.

Vecchiato (2016) raised concerns on awareness of users on security dangers behind user-defined configurations. Their study had found that only 18 settings are correctly set. As a remedy, Das and Khan (2016) point out that one way to ensure best security practices is to enable important security practices by default. However, Furnell (2005) had pointed out that one level of security cannot be expected to suffice for all users. Despite all the pointed issues concerning awareness and practices, Murray (2014) indicated that most users have adequate knowledge regarding security risks in their devices. However, this study has been challenged since it had involved only the tech-savvy and the challenge posed on deficiencies in device configuration stays intact. These works of literature have one thing in common: public Wi-Fi poses a huge threat to end-users and, worse, to smartphone users. The situation is worse in Android devices due to defects in configurations and the inadequacy of default settings.

Most internet users use wireless services (Mahadevan & Kaleta, 2018). While many works of literature have studied the safety and risk posed to Wi-Fi users, Breitinger *et al.* (2020) explored users behaviours on Wi-Fi. Their study found that most users follow weak security practices. On the other hand, most Internet users preferred public Wi-Fi for Internet access (Mahadevan & Kaleta, 2018). Similarly, they use default settings that are inadequate to secure them against Wi-Fi attacks. The pull factor towards using public Wi-Fi is mostly saving data plans (Sombatruang *et al.*, 2016). Swanson *et al.* (2010) pointed out that despite users' awareness of cybersecurity risks, they often do not believe that the risks will be realised. On the other hand, Klasnja *et al.* (2009) found that most users believe the default settings were adequate to secure them.

Another study by Jeske *et al.* (2014) investigated the factors that drive participants to choose a Wi-Fi network over the other and found that the padlock icon's absence or presence influences their decision. The study assumed that users were committed to connecting to Wi-Fi and had no other options. Generally, previous studies point out that users have a good awareness of cybersecurity. However, they would trade off their safety over utility.

2.5 Wireless Attacks Facilitating Rogue Access Point

Wireless networks are considered of high risk to users due to their nature of communication. They have four main components: Data transmission via radio frequencies, access points (APs) that link to the corporate network, client devices (laptops, PDAs, and others), and users. These components can be used to launch an attack that compromises one or more of the three basic security objectives of confidentiality, integrity, and availability. The attacks could be simulated at a user level, during data transmission or at the physical level. It is even easier to attack wireless networks since wireless STA transfers unencrypted data during communication. This section presents specific attacks in wireless networks that facilitate AP forgery.

2.5.1 Accidental Association

Several methods and intentions can be used to gain illegal access to wireless and wired networks. The term "accidental association" refers to one of these ways. When users turn on their computer and connect to a wireless AP from a neighbouring company's overlapping network, they may not realise it (Shedden *et al.*, 2016). However, this signifies the presence of a security breach. The signs come because a confidential firm's information has been disclosed, and there may now be a relationship between the two companies. This is especially true if the laptop is simultaneously connected to the internet via a wired connection (Bryksa & MacMillan, 2015).

2.5.2 Malicious Association

Malicious associations occur when crackers can actively connect wireless devices to a company network through their cracking laptop instead of a company AP (Harmon, 2018). *Soft APs* are laptops generated when a cracker uses software to make their wireless network card appear to be a legitimate AP (Bryksa & MacMillan, 2015). Once a cracker has gained access can steal passwords, launch attacks on the wired network, or plant trojans. Since wireless networks operate at the Layer 2 level, Layer 3 protections such as network authentication and VPNs offer no barrier. Wireless 802.1x authentications do help with protection but are still vulnerable to cracking.

2.5.3 Identity Safety (Spoofing)

Identity theft (or MAC spoofing) occurs when an attacker can listen to network traffic and identify the MAC address with network privileges (Chen *et al.*, 2007). Most wireless networks provide MAC filtering, restricting network access to only authorised computers with certain

MAC addresses (Madani & Vlajic, 2021). There are, however, several programs that can sniff networks (Kumar & Gambhir, 2014). When these tools are combined with additional software that allows a computer to pretend to have any MAC address the cracker wants, the cracker can easily overcome this barrier (Kropeit, 2015).

2.5.4 Man-in-the-Middle Attack

A man-in-the-middle attacker persuades machines to connect to a computer that has been configured as a *Soft AP*. After that, the hacker connects to a real AP using another wireless card, allowing traffic to flow freely from the transparent hacking machine to the real network (Kumar & Gambhir, 2014; Premnath *et al.*, 2021). The hacker can then sniff the traffic. One type of man-in-the-middle attack relies on security faults in the challenge and handshake protocols to execute a "de-authentication attack" (Segura & El-Moussa, 2014). This attack forces AP connected computers to drop their connections and reconnect with the cracker's *Soft AP*. Man-in-the-middle attacks are enhanced by *LANjack* and *AirJack*, which automate multiple process steps (Jiang *et al.*, 2013). Script kids can now perform what used to need some intensive skills. Because there is little to no protection on these networks, hotspots are extremely open to any assault.

2.5.5 Packet Injection

In a network injection attack, a cracker can use access points exposed to unfiltered network traffic to broadcast network traffic such as "Spanning Tree" (802.1D), OSPF, RIP and HSRP (Kropeit, 2015). The cracker injects fake networking re-configuration commands to routers, switches, and smart hubs. This can bring down a whole network, necessitating resetting or even reprogramming all intelligent networking equipment (Jindal *et al.*, 2014). To do this, an attacker is forced to mimic at least one of the addresses of the legitimate AP, preferably the BSSID. Then, proceed with the transmission of packets to destinations or clients associated with the targeted BSSID.

2.6 Wi-Fi Spoofing Detection

There exist many scholarly materials in wireless spoofing attack detection. The majority are focused on ARP spoofing and MAC address spoofing. This section reviews the works in MAC address spoofing, which is part of ETA and a generally wireless spoofing attack.

2.7 MAC Address Spoofing

The MAC address spoofing attack detection in wireless approaches has been well covered in the previous literature. It started far way back in work by Faria and Cheriton (2006). They proposed using signal strength, mostly referred to as RSSI, as a variable to detect physical address spoofing attacks in wireless networks. Their detection model assumed the presence of more than one AP that is capable of reading signals from all nodes in the network. So, the received RSSI measures from an AP were aggregated into a single profile.

Chen *et al.* (2007) and Wu *et al.* (2018) used a K-means clustering algorithm to detect signal spoofing by fake APs. These works are built under the assumptions that "the sequence of last *n* RSSI values received from an AP would have minimum fluctuations around the mean in the absence of another rogue AP (an Evil Twin AP)". Clusters are developed based on mean values collected from the received RSSI values. A huge distance between the nodes of the two clusters would indicate the existence of an Evil Twin AP with its unique RSSI distribution.

Unfortunately, implementing the proposed solutions in Android devices may be resource-demanding. Therefore, this study adopts the clustering of AP's RSSI values based on their means and defined threshold value of the first broadcasting AP. In our study, it is assumed as the legitimate AP. Furthermore, this research study opts for visible wireless AP beacon frame parameters to compare the legitimacy of wireless hotspots. The parameters that Android devices could easily read without affecting efficiency and performance were chosen.

Sheng *et al.* (2008) explored antenna diversity effects in wireless APs and their impact on signal strength device fingerprinting and detection of spoofing attacks. They showed that RSSI values from a static receiver collected at a stationary transmitter form a mixture of two Gaussian distributions due to antenna diversity permitted under the 802.11 protocol. As a result, they utilised a log-likelihood ratio test on the sequence of the latest received RSSI at each AP from a given MAC address to train a Gaussian mixture model for each wireless node and AP pair in the network. If the ratio test fails by more than *n* Gaussian mixture models—where *n* is less than the number of accessible APs in the network and must be set empirically—a transmitting node is deemed fake. An opponent can easily alter its transmission power to circumvent detection by this model using readily available off-the-shelf hacking tools.

Madani and Vlajic (2021) present a practical machine learning (ML) approach to detect MAC address spoofing using RSSI. They explore RSSI-based device profiling in dynamic real-world environments/networks with moving objects. Their ML approach uses a multi-model Long

Short-Term Memory (LSTM) auto-encoder—a deep recurrent neural network. However, the study uses RSSI profiling only to detect MAC spoofing, leaving the use of a combination of MAC, RSSI and security protocols in AP as a room for exploration. Furthermore, their study did not indicate the range of devices their solution could be deployed.

2.7.1 Evil-twin Attacks

Various works exist to detect network spoofing attacks based on ETA. The research work by Gonzales *et al.* (2010) devised a context-leashing technique. They claim that publicly accessible APs, such as those found in franchise coffee shops (e.g., Starbucks), share SSID and are frequently unauthenticated. This allows adversaries to spoof such SSIDs and deceive clients into connecting to a RAP (e.g., after performing a dissociation attack). The defence against the Evil-twin APs proposed by Demirbas and Song (2006) also assumes a context-leashing engine. The context-leashing engine would collect a list of contexts upon association with a publicly available AP, which contains a list of all visible SSIDs and their accompanying average RSSI values that are reachable at the moment of association with a given SSID in the environment. A new context list is constructed and compared to the previously stored one for any future re-association with a given SSID. Assume that the context list of available neighbouring SSIDs and their average RSSI values do not overlap considerably (empirically defined) with the historical context list. In this situation, the related SSID has been designated as an Evil-twin, and the connection should be terminated. In this work, the assumption that the list of SSIDs at particular geolocation keeps substantially unchanged over time is their method's fundamental flaw. This assumption is unreasonable given the existing tethering capabilities of mobile devices and cell phones.

Another work by Shrivastava *et al.* (2020) focuses on detecting the ETA. They developed the *EvilScout*, an evil twin detection and mitigation framework that utilises the information of the IP prefix distribution by the legitimate access point (LAP). The tool exploits software-defined network (SDN) potential for detecting an evil-twin without the need for any additional hardware or modifications at the AP or client. However, this approach could be limited when attackers create APs with SSID, which are not similar to most of the hotspots in an organisation. Furthermore, identifying IP prefix distribution by LAP is challenging because LAPs cannot be identified at the user level. Nevertheless, implementing the *EvilScout* features on Android devices would greatly improve spoofed hotspot detection for Android devices.

2.7.2 Rogue Access Points and Client-Side Solutions

A study by Chirumamilla and Ramamurthy (2003) developed an agent-based IDS to discover illegal APs as part of their research. Developed agents keep a list of registered APs. The administrator notifies the MAC addresses of all AP agents when an AP is added or withdrawn from any agent. As a result, each agent will have a current AP list. Agents are also tasked with looking for phoney APs. After each scan, their work matches the AP's MAC addresses found with the AP MAC addresses in the host list. The administrator will be notified by SMS if the AP is not on the current list. The study's agent includes a wireless interface and two network interface cards. This work relies on the power of the host network to detect the presence of fake AP.

Ballai (2010) presents a system and methods to detect unauthorised APs accessing wireless communication. The study collects the beacon details from the transmitting AP and determines their validity. The measuring process involves comparing the details with a pre-existing database of the APs in a communication network. This work is limited because the implementation relies on the host's database of all APs in the network. Similarly, Lim and Kim (2013) present an AP verification approach that includes a controller to control the connection with the AP. There is also an AP determination unit to connect with the AP and determine if it is vulnerable or not. Moreover, a method to determine the security status of their AP goes through identifying a connection, connecting the terminal with the AP, determining if the AP is vulnerable and controlling the connection with the AP.

Segura and El-Moussa (2014) present methods for authenticating APs as an improvement from Ballai (2010). This approach demands that each AP be authenticated first before authorising it to use the network service. The approach has an authentication server, and two identifiers are set on the host network (wired) and wireless device. In addition, an information server is configured with a comparator. If the details from the two matches, then the AP is genuine.

Kao *et al.* (2014) developed an algorithm by looking at the serial numbers, timestamps, and range of beacon messages to prevent fake AP attacks. Their study saw that the attackers could change the serial numbers, timestamps, and signal intervals of fake APs. However, they suggested that the method they proposed was successful in detecting these attacks.

Bryksa and MacMillan (2015) proposed mechanisms to secure wireless networks using authentication mechanisms of the wireless client device, including an access controller to establish an encrypted connection between nodes and hotspots. This approach demands an

environment where APs and Wi-Fi client associations are authenticated. Its implementation relies on the network host. However, most of the Wi-Fi APs in public are open. On the benefit of this, attackers create RAPs perpetrating the legitimacy of the host network.

Matte *et al.* (2015) created rogue Wi-Fi APs to leverage geolocation details available on geotagged services like Facebook and Twitter and simulate other attacks. They also used BSSID and RSSID parameters to predict the location of the AP. The BSSID and RSSI make a good combination for identifying fake APs. However, this approach cannot be feasible when the focus is to detect spoofing attacks on the client devices or mobile users connecting to hotspots since legitimate BSSID cannot be identified from end-users despite the possibilities to capture them. Our proposed solution creates an Android application to leverage the advantage of captured hotspot parameters to determine their legitimacy.

Kropeit (2015) explores the detection of hotspot spoofing on smartphones by the "KARMA attack" through challenging the features to create every requested network. The author uses *iw, iwconfig* and *iwlist*, all available in a Linux based environment. The author also uses the *aircrack-ng* suite, one of the most popular network auditing software, as part of security tools to crack the WEP and WAP. The detection program sends multiple directed Probe requests using randomly generated SSIDs. However, the approach could be further extended by capturing and evaluating more packets and getting more from the packets. For example, AP capability and encryption information was not used in this research. The SSID-Mixer packets could be tagged to recognise them during capture and disregard them during evaluation. Furthermore, methods for detecting rogue APs and the honeypot AP using WEP were left undone.

Deshpande and Davenport (2018) present a mechanism to detect RAPs. It uses messages sent from the user's device. Initially, a user sends the first AP discovery message, including a pre-stored identifier of previously associated APs. It then sends an additional message, including some identifiers of non-existent AP. Finally, using a combination of sent messages, the device detects the existence of RAPs.

Harmon (2018) also presents a system and method to detect illegitimate APs. This works on a computer system by: (a) detecting when a device attempt to connect to the wireless AP that resembles the legit one, (b) identifying the location of the computing devices and APs, and (c) detecting illegitimate AP by comparing the geographical location of the APs. However,

implementing the described solutions depends on external devices, systems, or the network host side.

Tchakounté *et al.* (2020) demonstrate an approach that focuses on detecting wireless spoofing attacks focusing on SSID, BSSID, communication mode and the security protocol. This work assumes that the administrator verifies LAPs. It further assumes that the network administrator knows the entire network and maintains the legitimate, illegitimate and suspected hosts database. Usually, users connecting to Wi-Fi do not see the network in terms of configurations and clients associated with it.

Kim (2020) suggests a system and methods for detecting RAP and a fake user device. The invention uses common beacon characteristics and a database to store beacon parameters. The invention uses a one-time URL to detect fake hotspots and address resolution protocol (ARP). With a combination of these parameters, the invention demands a detection server to incorporate the prescribed elements. The invention leaves room for exploration on RAP detection on mobile devices without an external device.

Generally, little has been done to help Android devices to detect fake Wi-Fi hotspots. Moreover, it is rare to find solutions for Android devices and especially without the possibility of rooting a device. According to Alsop (2020) report on mobile OS market share as of July 2020, it covers 74.6% of mobile phone users and devices connecting to Wi-Fi.

2.8 Gaps in Literature

The majority of research works focus on host networks assuming that the list of all APs is known. However, this assumption is not valid when focusing on end-users. They are aware of neither device connecting to the network nor the network structure. Also, most solutions focus on the localisation of spoofing Wi-Fi hotspots in wireless networks. At the same time, some research works require specialised hardware and human intervention. These approaches leave mobile phone users behind since the proposed solutions are expensive to implement on mobile devices. Furthermore, some Android devices' approaches demand a root privilege that is considered highly dangerous to be granted to user applications. Therefore, we propose comparing beacon frames broadcasted by a Wi-Fi hotspot to determine their legitimacy using an Android phone without the need to root the device. The comparison of features between the proposed solution and existing approaches is presented in Table 3.

The review of previous research based on detection of hotspot spoofing detection has resulted in the realisation of the following research gaps:

(i) The identification of fake hotspots/hotspot spoofing in Android devices using security capabilities, SSID and BSSID has not been explored.

(ii) Detection approaches in Android devices are scarce, and those available require root access, which is considered dangerous to end-users.

Table 3: Comparison of detection approaches between the proposed system and existing approaches

Features	Ballai (2010)	Segura and El-Moussa (2014)	Bryksa and MacMillan (2015)	Al-Zubaidie *et al.* (2019)	Matte *et al.* (2015)	Deshpande and Davenport (2018)	Harmon (2018)	Madani and Vlajic (2021)	Chen and Yang (2012)	Tchakounté *et al.* (2020)	Kropeit (2015)	Proposed solution
Host independence	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✓	✓
Use of Wi-Fi beacons	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
End-user focus	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✓
Support for Android device	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Root free	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

CHAPTER THREE

MATERIALS AND METHODS

3.1 Research Design and Study Area

This study has taken several approaches to achieve its objectives. The initial research activities led the study to develop and refine the application requirements and learn the visible characteristics of fake hotspots through an experimental research design. Furthermore, this research is designed under the guidance of the development of a hotspot spoofing detection prototype for Android-based devices. The design involved studying users' practices, knowledge and susceptibility on Wi-Fi networks, design and development of the detection prototype and validation of the prototype.

The first stage involved studying users' susceptibility to fake hotspots and studying characteristics of fake hotspots. Fake hotspots can easily be created by an attacker using a normal Wi-Fi router by broadcasting APs with SSIDs similar to the legitimate APs of the host organisation. For an attack to be successful, attackers usually leave their network open for anyone to connect. They would also broadcast with a stronger signal than the rest of broadcasting APs (Park *et al.*, 2014; Wang *et al.*, 2015). Therefore, the logs of users connecting to our networks were collected and studied for users' practices based on the time they would stay connected in unknown networks and the number of times they would connect to the same. On the other hand, packets of fake APs were collected and compared with legitimate APs to identify the differences between the two.

The second stage involved a survey in studying user knowledge, practices, and efforts to identify possible wireless network attacks. Again, the focus was set when users were connecting to APs. Moreover, compliance with the organisation's best-practice recommendations was also studied. This approach was used to define the characteristics of the studied population, and it made it easy to recruit participants for the study.

The study focused mainly on two academic institutions, the Nelson Mandela African Institution of Science and Technology (NM-AIST) in Arusha Region - Tanzania and the Mzumbe University (MU) in Morogoro Region – Tanzania between March and June 2021. These institutions were chosen due to their nature and Wi-Fi coverage throughout their campuses. These institutions further gave us the target population as they accommodate students and workers at all levels and professions. For instance, MU enrols students from certificate level to PhD. On the other hand, NM-AIST enrols master's and PhD level students, giving an equally

distributed population in terms of academic level. Moreover, their employees are of different qualifications and professions distributed at various academic levels. Therefore, students currently enrolled for MU and NM-AIST and staff at MU and NM-AIST were involved in this study.

The third stage of the study involved the design and development of the prototype based on requirements gathered during the previous two stages. First, the requirements were considered based on the user requirements developed from the first stage. Second, system requirements were set based on activities done in the second stage of this study involving the experimental setup.

3.2 Data Collection Methods and Tools

In this study, various methods for data collection were employed at different stages of the research work. Both quantitative and simulation research methods were used to collect primary data and to study users' practices on wireless networks, respectively. The experiments were conducted by simulating an attack in our study areas, and quantitative approaches were employed using a survey. Surveys are effective when the study involves a large population, and its characteristics must be described. Surveys are also the best choice in probability sampling studies (Owens, 2002).

The first stage of this study involved learning the visible patterns of spoofing attacks and looking into ways users fall susceptible to Wi-Fi attacks. A simulation was employed to create fake APs and count the number of users connecting to the fake network. The experiments were simulated in three consecutive days at both NM-AIST and MU to have data for an informed decision and backing up evidence of users' susceptibility. A NETGEAR Nighthawk® X6 AC4000 Tri-Band Wi-Fi router in Figure 3 was used for this exercise. It is a tri-band Wi-Fi providing three (3) dedicated bands optimised for speed. The First Wi-Fi band had 2.4 GHz with up to 750 Mbps. The second Wi-Fi band had 5 GHz with 1.625 Gbps dual-band, and the third had 5 GHz with 1.625 Gbps. These features allowed us to simultaneously broadcast six (6) APs with distinct SSIDs. Logs (Appendix 1) were collected in the *.txt* file format and analysed to filter unique devices connected to our network based on the device's MAC address.

Figure 3: A NETGEAR Nighthawk® X6 AC4000

The second stage aimed to study the visible characteristics of fake APs to develop functional and non-functional requirements. The AP may be created from a simple router or simulated using the common Wi-Fi penetration tools. For the experiment to be successful, an ETA, packet injection and de-authentication attacks were simulated in a lab environment. The setup involved two PCs, one installed with the Kali Linux 2021.2 and another with Ubuntu 21.04. The study used the Kali Linux 2021.2 to simulate the attacks with the help of the Alfa One AWUS036H 1000 mW Chipset RTL8187L[1]. The PC running Ubuntu 21.04 OS had a built-in wireless chipset with the ability to run in monitor mode. A list of commands used to simulate an attack is presented in Table 4. Initially, the Wi-Fi chipset was set into monitor mode using the *airmon-ng* command. The second step was to check the broadcasting APs in the perimeter using *airodump-ng*. The command enables the collection of AP details that could be used to simulate an attack. Finally, the *airbase-ng* command created an evil twin AP based on details collected in the second step.

The second PC was used to capture packets in pcap format and then analyse them using the network protocol analyser. Protocol analysers are the tools that can be deployed in a PC or in the network to capture network traffic and perform analysis. The tools help network administrators examine the live network data or saved pcap files to identify problems with the network traffic or potential malicious activities.

The study chose *Wireshark* since it is open-source and has a graphical user interface (GUI). Furthermore, the *Wireshark* is noted for its filter language and the support for more than 1100 protocols with detailed information on more than 90 000 protocol fields. *Wireshark's* ability to

[1] https://shop.secpoint.com/shop/alfa-awus036h-1000-253p.html as in July, 2021.

filter packets during and after capturing makes it the best for all users, newbies and professionals. Additionally, its availability on all platforms and its open-source nature made it the most preferred network protocol analyser and has huge community support.

Additionally, the common wireless filters that can be run in the *Wireshark* are presented in Appendix 5. These commands were used to filter the targeted probes based on simulated attacks. The probes were filtered and observed for noticeable differences that could be used to develop the detection prototype.

Table 4: List of commands used to simulate an attack

Command	Function
airmon-ng	Set Wi-Fi adapter into monitor mode.
airodump-ng	Scan broadcasting APs
airbase-ng	Attacking clients as opposed to the AP (Create fake AP)
aireplay-ng	Used to inject packets, de-authentication attack

The Alfa AWUS036H was chosen since its chipset was compatible with other tools used in this study, including the *Android PCAP* library and the SAMSUNG Galaxy SIII.

Figure 4: Alfa One AWUS036H

The third stage of the study involved a survey method to study user practices, knowledge, and efforts to identify attacks on Wi-Fi using a questionnaire. The self-administered questionnaires were chosen due to respondents' amount and nature of the information. The questionnaire was designed bearing a flow of questions separated into nine sections. The survey design went through pre-testing to ensure that the questions asked were accurate and enhanced the requirement development process and establishment of the problem. We followed the Yaddanapudi and Yaddanapudi (2019) guide to avoid leading questions and ensure maximum clarity of the questionnaire. The structure of the questionnaire is presented in Appendix 3.

The questionnaire was distributed as the responses were collected. The recruitment process was done in person as the questionnaire was not supplied for everyone to respond. However, the process was monitored to avoid sample biases. The monitoring process was done by distributing the questionnaire across diverse respondents with varying ages, gender, education level, professions, work status, and institutions they belong. The questionnaire was sent to targeted respondents via email, social media, and text messages. The target audience was Android device users who are currently studying or working at either NM-AIST or MU. Participant data were kept confidential, and personal identifying information was not collected.

The questionnaire had 32 questions, distributed between multiple-choice questions and check box questions prepared using the Google forms. Questions were made generic enough to be responded to by diverse respondents except for Sections III and IV intended for students and workers. The majority of questions in our questionnaire used the 5-point Likert scale questions that allowed respondents to choose appropriate answers. The rating was arranged from *strongly disagree, disagree, neutral, agree, and strongly agree.*

3.3 Sample Size Determination and Sampling Technique

The Nelson Mandela African Institution of Science and Technology had 225 active students based on the institution's capacity to host students in-campus. The Institution had a capacity of 226 workers as of June 2021. On the other hand, MU had 282 certificate students, 151 diploma students, 4956 bachelor degree students, 144 Master’s students, and 4 PhD candidates, making 5537 students in total as of June 2021. The institution has also employed more than 500 employees. This makes a total population of 6488 in our study area. Figure 5 presents the structure of the population in the study area.

The total population for the study covers the total number of students and workers in the two institutions. The target population is the intended group of people researched for the

information required to be ascertained (Banerjee & Chaudhury, 2010; Saunders *et al.*, 2009). This study required candidates who are Android device users and regular Wi-Fi users. In this case, the assumption made was that all Android device users mostly use Wi-Fi to access the Internet. Since the study could not obtain the number of Wi-Fi users in the host institutions, 70% of the population covering the Android users was considered making the target population be 4541. The study made this assumption based on the global mobile market, where Android covers more than 70% of the global market (O'Dea, 2021).

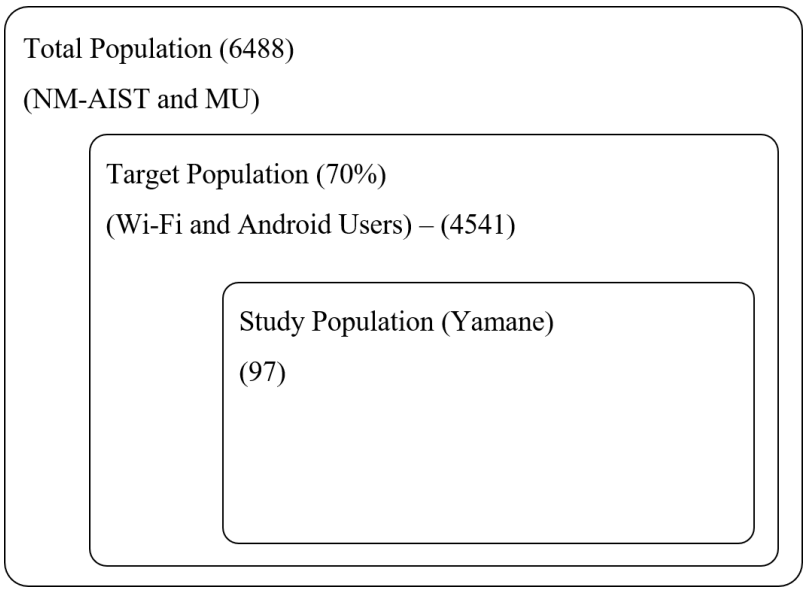


Figure 5: Population structure and sample calculation

As shown in Appendix 2, the sample size was determined using the Yamane simplified formula for proportions (Yamane, 1967). The calculations were done based on the target population of 4541 people obtained from 6488 people at the precision of 10%. Therefore, the minimum sample size obtained was 97 people.

Participants were sampled using the stratified random sampling technique. Stratification is the process by which the population is divided into subgroups/strata (Singh & Masuku, 2014). The method was chosen since the studied population was heterogeneous. Therefore, the sample was grouped into homogeneous groups, commonly known as strata. Participants were randomly selected based on their stratum. The sampling process was based on respondent status, working or studying, respondents' professions, education levels, and gender.

3.4 Data Analysis

The collected data were downloaded from the google form and saved into comma-separated values (CSV) format. Data were coded and analysed in RStudio. RStudio is an Integrated

Development Environment (IDE) for R. The R is a programming language made for statistical analyses and graphics (Racine, 2012). The RCharts, tables, and descriptive statistics were generated in RStudio using a variety of packages.

The study chose R as it is considered one of the most powerful and flexible statistical language tools. In addition, it is open-source, and its IDE and packages are freeware. Other benefits include the growing user community, platform independence, and integrating with Microsoft Excel and other commercial software such as SPSS, SAS, Matlab and Statistica (Racine, 2012).

As described earlier, the first stage of the analysis phase involved data cleaning and processing. The collected data were structured in CSV format, which the RStudio can read. On the other hand, incomplete responses, data with errors, duplicate responses were handled in the basic settings of the Google forms. Other cleaning processes were done using several R packages.

The study employed descriptive analysis during this stage. The descriptive analysis describes and summarises data points into more readable and understandable (Banerjee & Chaudhury, 2010; Lawless & Heymann, 2010). This method allowed the study to interpret the data distribution and identify the similarities among variables. The descriptive analysis method is popular for its objectivity and neutrality. It is considered more vast than other quantitative methods and provides a broader picture of an event (Lawless & Heymann, 2010). Furthermore, the descriptive analysis has less margin of error as the trends are extracted straight from the data properties. In this case, it has helped the study develop clear requirements for designing and developing the prototype based on user practices on Wi-Fi and justify the problem's existence.

For the collected data to be meaningful, reliability and validity of data were ensured. Data validity is how a given data set is accepted for the research to provide the required outcome (Golafshani, 2003; Stake, 2010). While reliability is the degree to which the findings would be consistent if the study were done under the same environment with a different sample of the same group (Golafshani, 2003).

3.5 Software Development Approaches

Software Development Life Cycle (SDLC) approaches were considered in developing the solution. The SDLC is the framework that defines activities in every stage of the software development process, covering the details for building, deploying and maintaining the software (Ragunath *et al.*, 2010). The main goal of the SDLC is to ensure the highest quality of the software is produced at minimal costs.

As presented in Fig. 6, SDLC defines the complete software development cycle, which incorporates all the tasks involved in planning, creating, testing, and deploying a software product. The software development process will not be systematic and disciplined if it does not choose an appropriate life cycle model. The choices may be determined by the team's level of expertise, business requirements, time, and budget.

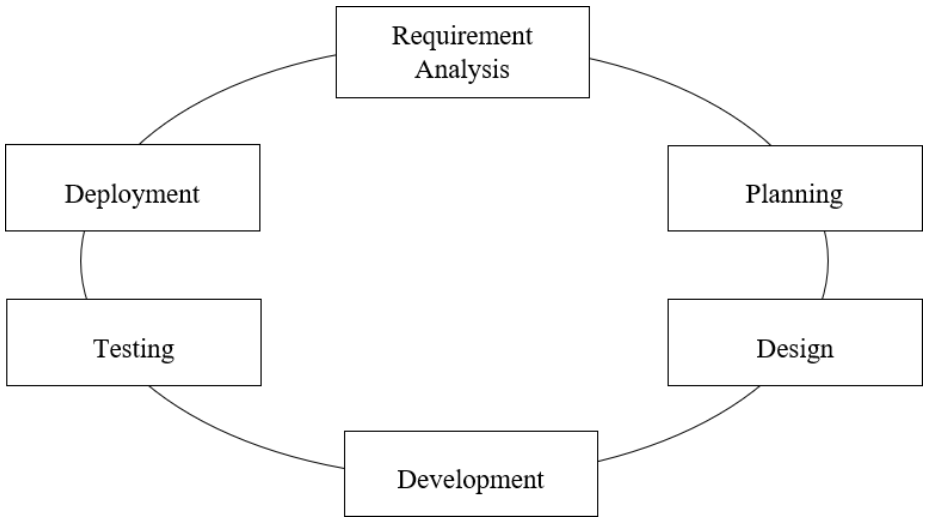


Figure 6: The software development life cycle (Leau *et al.*, 2012)

The SDLC presents different models. The comparisons between traditional and agile methods are shown in Table 5. In addition, the table compares the chosen models that are commonly used.

As observed in Fig. 7, the Agile methods represent a group of methodologies coordinating teams and organisations to put the agile mindset into practice. The main target is to increase the agility of the business. The methods define a cycle of activities starting from requirements gathering to testing activities. The development process goes through all the basic SDLC procedures in every functionality cycle to deliver. When the functionalities are being developed, the product is sent to customers or users for approval. At this stage, the product goes into deployment and the maintenance stage.

Table 5: Comparison between Agile methods and Traditional methods

Factor	Agile	Traditional
User requirements	Iterative acquisition	Detailed user requirements are well-defined before coding/implementation
Rework cost	Low	High
Development direction	Readily changeable	Fixed
Testing	On every iteration	After the coding phase completed
Customer involvement	High	Low
Extra quality required for developers	Interpersonal skills & basic business knowledge	Nothing in particular
Suitable Project scale	Low to medium-scaled	Large-scaled

Leau *et al.* (2012)

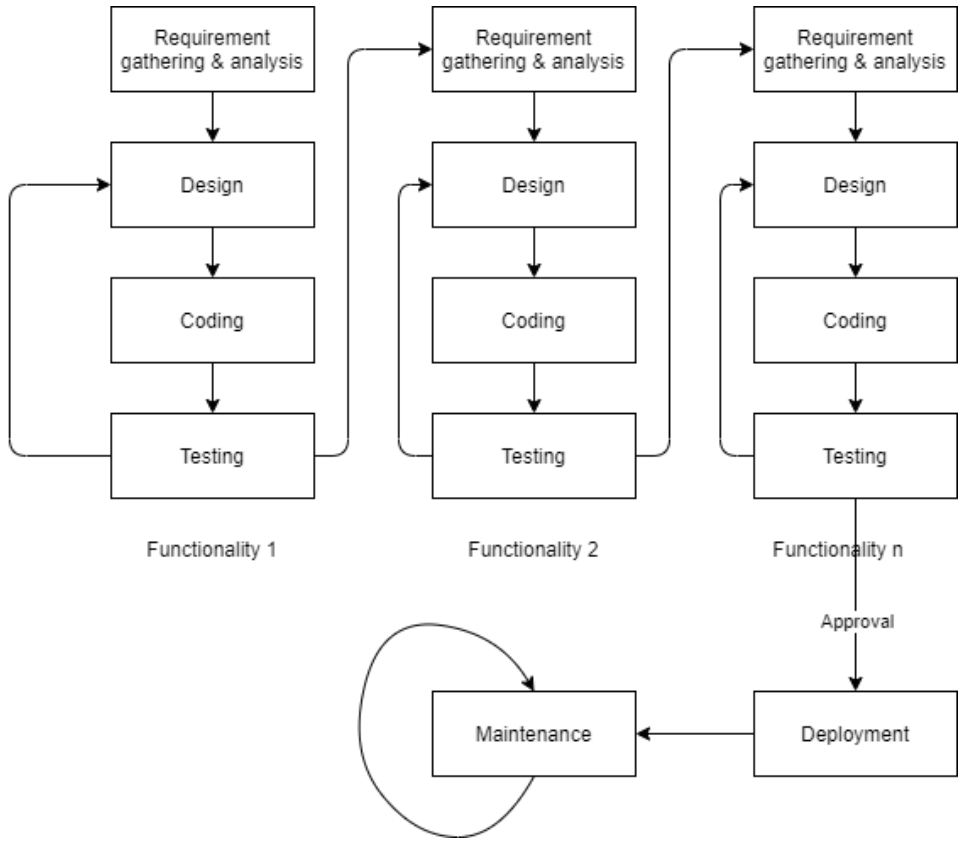


Figure 7: Agile software development methodology

This study employed the extreme programming (XP) development methodology, one of the agile methods. Most Agile methods emphasise teams, customer collaborations, and responding to changes. In contrast, other traditional methods focus on contracts, plans, documents and tools (Leau *et al.*, 2012). The XP is a lightweight methodology for small teams in developing

software under rapidly changing requirements. It is based on communication, simplicity, feedback, courage, and respect.

Furthermore, it nominated coding as the key activity throughout a software project with a single goal, to deliver a software project with the right functionality and deliverables within a timeline. The XP was chosen due to its ability to deliver results quickly and accommodate the requirements changes at any work point during the development process. In addition, the method allowed the study to focus on specific functionalities timely and complete various stages of development processes in time and at reasonable costs.

3.5.1 Requirement Elicitation

In this study, the development process of the prototype was preceded by the requirement gathering process. Analytic and brainstorming methods were employed to collect all the requirements for the application. Various kinds of literature were reviewed to gather requirements for the solution. The technique was used to supplement other ways of gathering requirements during the development process. The review helped the study gain insights into the study domain, existing systems, and the current situation.

On the other hand, the study employed a brainstorming method to gather the requirements. Brainstorming leads to better problem understanding and feelings of shared ownership (Paetsch *et al.*, 2003). Furthermore, the attacker's behaviour and visible characteristics of the fake APs were studied in an experimental study.

3.5.2 System Design

The proposed solution's designs were presented using the Unified Modelling Language (UML). UML is a graphical language for describing, visualizing, building, documenting, and sharing software system artefacts. Use case diagrams, system sequence diagrams, class diagrams, and other artefacts are among the diagrams included (Booch, 2005). In addition, UML provides various language tools that can help developers communicate with each other and explain the relationships between software components, actors, and subsystems.

The implementation of the hotspots spoofing detection prototype was preceded by system modelling. The system design involved mapping core functionalities to different system actors and depicting system processes. First, the processes mapping was done into Data Flow Diagram (DFD). Then, various models were conceptualised to visualise the operation of the solution. These models guided the development process to ensure the study developed the right solution.

This study used *draw.io* and *LibreOffice Draw* to generate the modelling diagrams; these packages were chosen due to their flexibility and freeware nature.

3.5.3 System Development

The previous steps' requirement elicitation and system modelling processes guided the choice of tools to implement the solution. Then, various tools were selected based on their ease of use, availability, cost, and best tools that could result in a desirable product. The tools chosen for this study are described below:

(i) Android Studio

Android Studio is the Integrated Development Environment (IDE) for Android OS. It is bulti on JetBrains' IntelliJ IDEA software (Esmaeel, 2015; Wolfson & Felker, 2013). It is specially developed for Android development. Android Studio is built with many libraries that ease implementing an Android application (Craig & Gerber, 2015). The detection prototype is developed based on the Android OS in this research. The Android Studio was chosen because it offers dependency management through its built-in Gradle system. This feature makes it easy to interact with low-level features of Android devices. It was also chosen because of its built-in functionality and libraries. Some useful libraries for this study were the *WebView* and Wi-Fi capabilities.

(ii) SQLite Database

SQLite is a relational database management system (RDBMS) contained in a C library. The SQLite is popular for its ability to be embedded in the end programs (Vogel, 2010). The SQLite is designed to operate on structured data presented in a relational model. It is a light version of databases designed to work on mobile devices locally. The SQLite comes as a library stored locally in Android devices, which is attached to the Application to operate under the logical design of the host application (Bhosale *et al.*, 2015). It supports transactional features and serverless operations with almost zero configurations. It is an embedded SQL Database engine without any separate server process, unlike any other SQL database (Aditya & Karn, 2014).

The SQLite database was chosen since it supports all the relational database features and is the open-source compact library that is by default present in Android (Vogel, 2010). Its storage may base on either disk or in-memory, and each database is stored in a single disk file to be used on cross platforms. It is very fast and needs very little memory to operate (Lee, 2012). Furthermore, the SQLite database presents its structures in a relational database design. Unlike

other database engines, SQLite supports Toolchain and comes embedded in the Android development environment (Aditya & Karn, 2014). It is highly customisable, giving the ability of programmers to design their preferred structure. Taking advantage of the existing SQL, the SQLite presents data that can be debugged (Vogel, 2010).

Figure 8 presents the main parts of the SQLite architecture composed of core parts, compiler, backend, and accessories. The parts contain the user interface (UI), SQL command processor, and the virtual machine (VM). On the other hand, the SQL compiler includes a *tokeniser*, a parser, and a code generator. The backend block contains B-Tree, page cache, and OS interface. The last block, the accessories block, has utilities-related functionality such as memory allocation and test codes containing caseless string comparison routines in the "*util.c"* file.

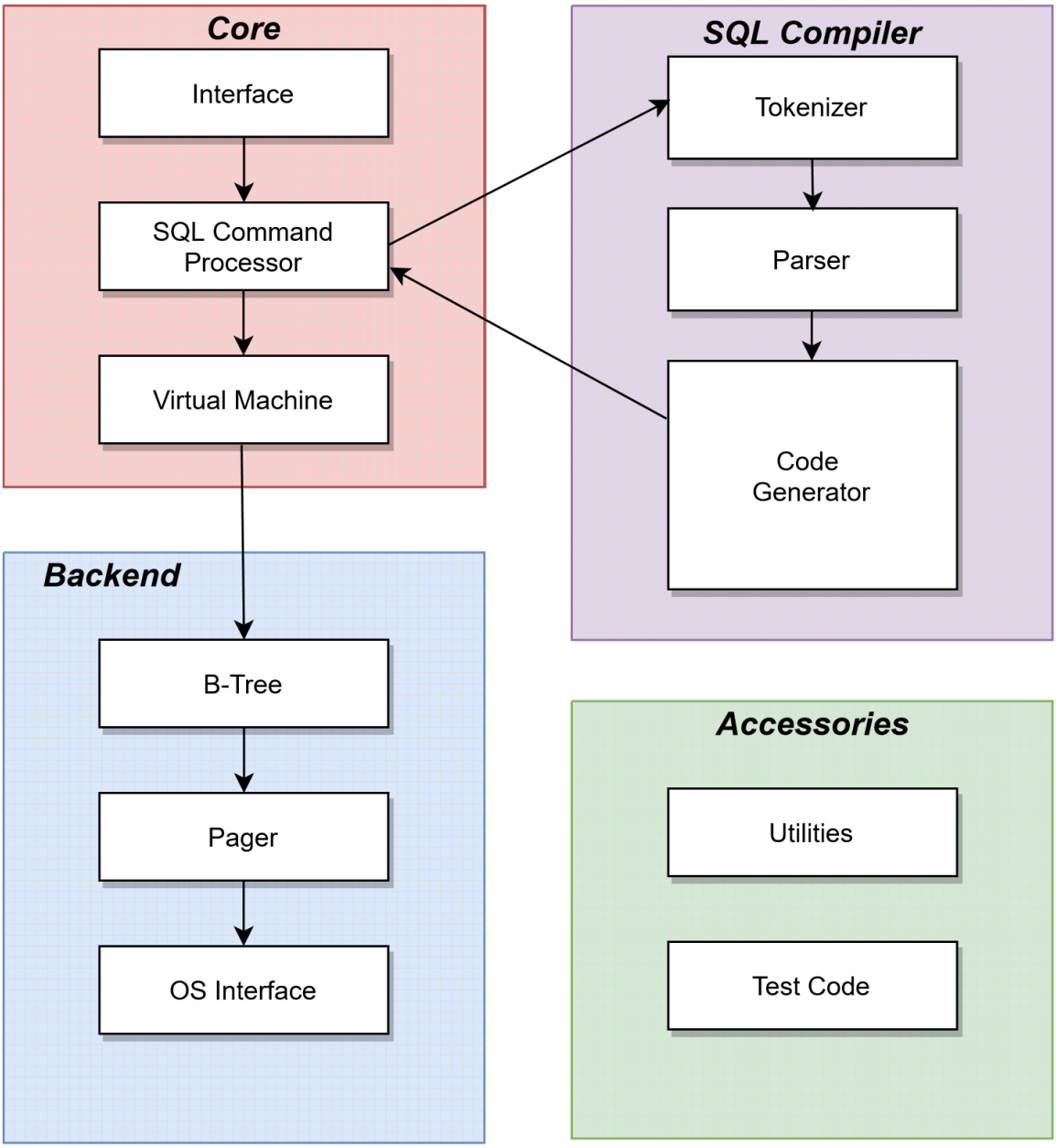


Figure 8: SQLite architecture

(iii) eXtensible Mark-up Language

Extensible Mark-up Language (XML) is a computer language for displaying, storing, and transferring data independent of other software and hardware (Marty & Larry, 2001). Self-describing data formats and structures can be electronically transferred with XML, providing a uniform vocabulary for information interchange across applications in messaging systems. It is also open-source software that is well-supported and has a wealth of technical knowledge (Bray *et al.*, 2008).

In this study, XML was mainly employed for developing the user interfaces in Android mobile applications. This choice was because it is less expensive (free) and easier to offload and reload data to and from the database while keeping the appropriate data and user interface appearance. The user experience on the mobile application is enhanced thanks to the chosen user interface:

(i) Java

Java is a general-purpose, object-oriented programming language designed to run across platforms. It assures software development since it is always fast, safe, portable, stable, and capable of doing numerous tasks at once (Android & Hagos). Java was chosen for this study because it is integrated into the Android Studio IDE and provides syntactical and library support.

(ii) Android PCAP Library

Android PCAP Capture is a utility for capturing raw 802.11 frames in monitor mode or Promiscuous mode. It can work with many wireless cards and mobile devices and manipulate the PCAP files during capture. The resulting pcap files can be viewed on a computer using Eye P.A., Wireshark, Tcpdump, and similar tools, or online using *CloudShark*. The library was chosen to allow packet capturing in Android devices for analysis. During this study, the library did not need root privilege to work.

(iii) SAMSUNG Galaxy SIII

Samsung Galaxy SIII (Galaxy S3) is one of the smartphones designed, developed, and marketed by SAMSUNG Electronics[2] in their Galaxy S series. It supports up to Android 4.4,

[2] Samsung Africa | Mobile | TV | Home Appliances

KitKat GT-I9301I Neo only. Other later Android versions can be installed via LineageOS. As introduced in earlier sections, this study chose the Android PCAP library and Alfa One chipset, which the Samsung Galaxy SIII supports. The phone specifications are presented in Table 6.

Table 6: SAMSUNG Galaxy SIII specifications

Segments	Specifications
Processor	1.5 GHz dual-core processor
Memory	32 gigabytes of storage and 2 gigabytes of RAM
Connectivity	Bluetooth, Wi-Fi, NFC.
Built-in sensors	Accelerometer, gyroscope, proximity, compass, and barometer

3.5.4 System Testing and Validation

The system testing and validation were done with an experimental strategy, using a lab setup similar to the requirements elicitation process. The testing was carried out in iterations, with features produced using the Unit testing method: The technique of evaluating a system's small, independent functionalities in isolation from other functionalities. Eliminating faults at the low level is trivial with a single unit test. Therefore, each self-contained feature was evaluated separately during the development of this prototype to find flaws inside its boundaries.

A combination of all features, packets capturing, storage, and retrieval of data, the evil twin detection, and fake APs enabled a successful validation process. In addition, the application was exposed to different network structures used during the development process to validate the functionalities developed.

3.6 Ethical Clearance and Consent

Participants in this study were required to consent to participate in the survey before responding to the questionnaire. More details were attached to the statement introducing the Institution where the study was conducted and presented. The research aims and the study's commitment to ensure respondents' information confidentiality were stipulated.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Demographic Characteristics of Respondents

The survey approach involved the recruitment of participants from our study area. The study equally recruited participants for our survey based on their demographic characteristics to reduce sample bias. The demographic characteristics of survey participants were based on gender, age, education levels, and their field of work or study. In addition, respondents were recruited based on the two institutions involved. Other demographic characteristics are presented in Table 7. This process required the recruitment of participants in the study by considering the demographic characteristics. These characteristics are important in analysing users' practices in various studies (Singh & Masuku, 2014). The study engaged 100 participants, 50% female and 50% male. The selection of respondents was based on several criteria including, regular Wi-Fi users who use the Internet on Android phones to whom were contacted in person. In addition, participants were recruited from two institutions during the study, the NM-AIST and the MU, sampling 31% and 69% from each institution, respectively, based on their population distribution.

Age and levels of education were taken into account to achieve the diversity of the sample population. The 43% of respondents came from the 18 – 24 age group, 39% from the 25 – 34 age group, 16% from the 35 – 44 age group, and 2% from the above 45 age group. Participants were recruited from certificates to PhD education levels. At the same time, most respondents originated from bachelor degrees, equating to 47 % of the participants. Other levels include, certificate (5%), diploma (13%), advanced diploma (2%), masters' degree (23 %) and PhD (10%).

The study further considered respondents' roles in the institutions where the study was conducted, as presented in Fig. 9. The majority of participants (64%) were sampled from the students' population, 29% from the employed population and 7% from students and employed respondents. Among the students' respondents, 18.8% came from the first year of studies, 46.9% came from the second year of studies, and 34.4% came from the third year of studies. While among workers respondents, 55.6% came from the less experienced group (less than three years), 27.8% came from the mid-experienced group (4 – 6 years), and 16.7% came from the experienced group.

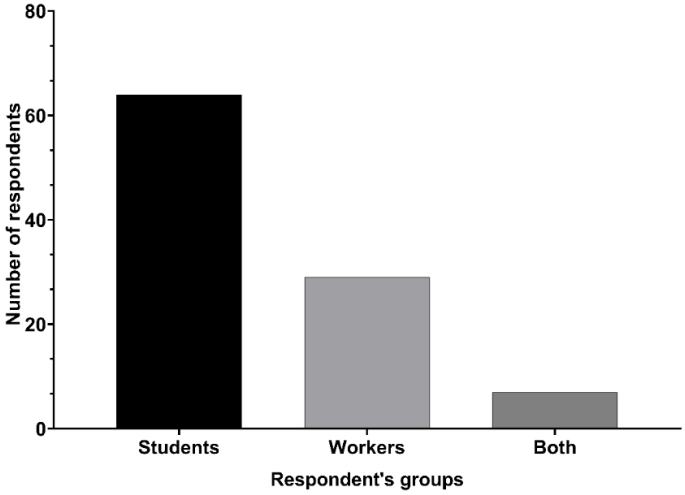


Figure 9: Respondent's working status

The study also considered the professions and field of study of workers and students, respectively, as presented in Fig. 10. Again, the occupations were grouped into IT fields and others. Among the student respondents, 43.8% came from IT-related disciplines, while 56.3 % came from other disciplines. On the other hand, 52.8% of employed participants work in the IT field, while 47.2% work in areas other than IT.

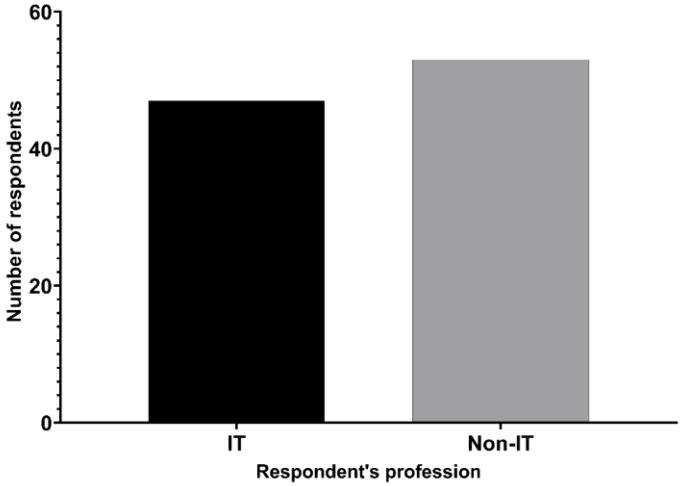


Figure 10: Respondent's professions distribution

Table 7: Demographic characteristics of respondents

Variable	Frequency (N)	Percentage
Gender		
Male	50	50
Female	50	50
Age		
18 – 24	43	43
25 – 34	39	39
35 – 44	16	16
45 and above	2	2
Education Level		
Certificate	5	5
Diploma	13	13
Advanced Diploma	2	2
Bachelor Degree	47	47
Master's Degree	23	23
PhD	10	10
Institution		
NM-AIST	31	31
MU	69	69
Work status		
Working	29	29
Studying	64	64
Working & studying	7	7
Profession/Studies		
IT	47	47
Others	53	53

4.2 Users' Susceptibility to Fake Access Points

The study collected data based on users ' susceptibility in the three-day experiment at MU and NM-AIST. Six (6) Wi-Fi APs were broadcasted with the settings attached in Fig. 11 and its scan result in Fig. 12. The naming of APs was based on: (a) the locations where the APs are broadcasting, (b) the deceiving names, and (c) similar broadcasting APs in the perimeter. All created APs were left open for everyone to be able to associate. In the backend of the router, the existing logs were cleared during each broadcast session to capture new logs.

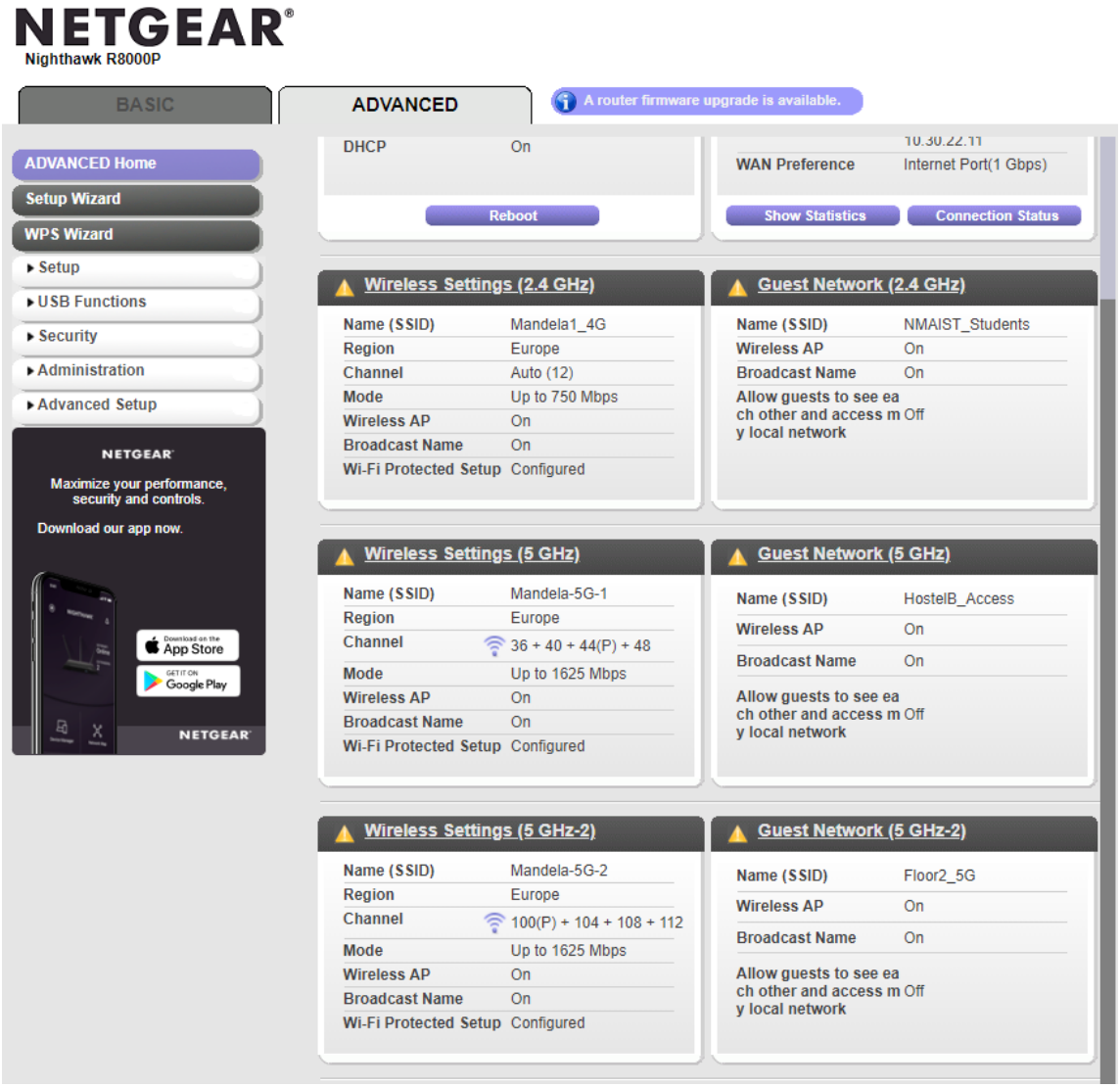


Figure 11: The NETGEAR settings for the six (6) broadcasting APs

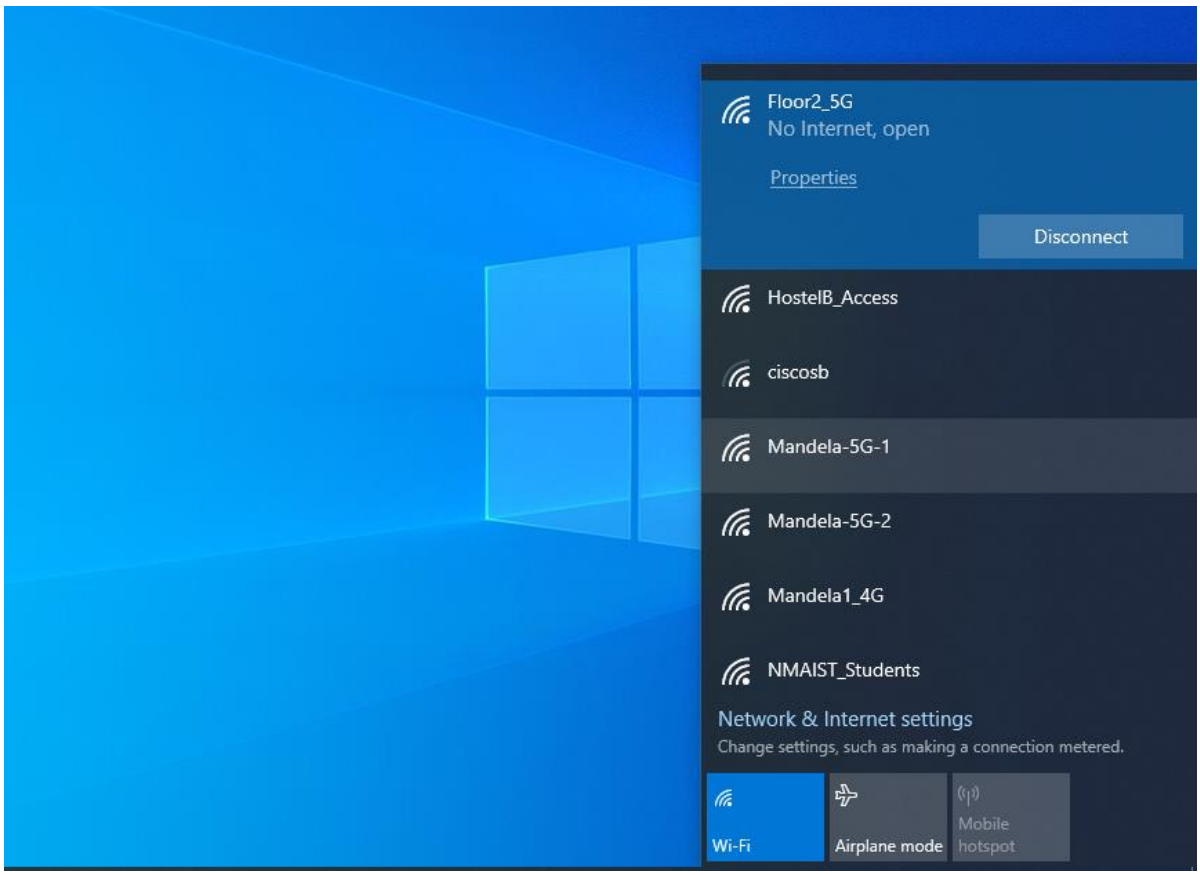


Figure 12: The six (6) broadcasting APs as scanned in PC

The logs collected at MU show that an average of 56 devices connected daily in our six Wi-Fi APs with 34 unique MAC addresses (60.7% of all devices) connecting in every broadcast. On the other hand, the experiments conducted at NM-AIST show that an average of 20 devices connected daily in the created APs, with 12 (60% of all devices) devices connecting in every broadcast. Generally, an average of 76 devices were connected in fake APs daily, at which their personal information may further be harvested. Among 76 devices, 34 devices, more than 60%, had connected in all of the broadcasts, which signifies that they had the fake APs saved on their device's PNLs. This further proves the statement presented in the Symantec (2017) report that users would connect to any available APs provided they are free.

Based on information collected during this experiment and user's practices demonstrated, the necessity of detecting fake APs before association would be more relevant than detection after association. Early detection is important since users do not put effort to identify fake AP. Instead, they would connect to any broadcasting AP in the perimeter.

4.3 Features of Fake Access Points

The hotspot spoofing attacks may be conducted in various ways using several tools existing in the market. Popular attacks are ETA and RAP, which can be simulated by simple software tools available in Kali Linux. The study simulated the two attacks to read the features of AP spoofing attacks. As presented in Fig. 11, an ETA would usually mimic the details of legitimate broadcasting AP in the perimeter. In addition, the commands for creating an ETA can disguise the SSID, BSSID, and channel number, among other details. Fortunately, both the RSSI and security information were not mimicked using the commands to create an ETA.

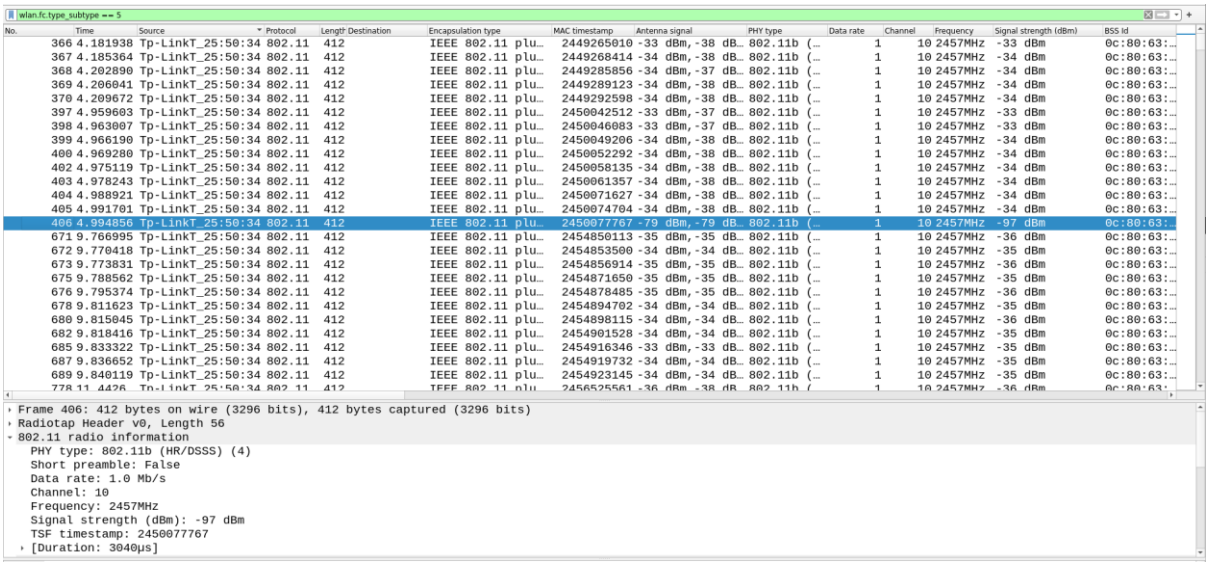


Figure 13: Notable packet details presenting the difference in signal level

The signal strength remains unchanged as it relies on the power of broadcasting AP. However, an attacker could boost their signal to be higher than the legitimate network to attract users to connect. Alternatively, they would position an AP to the location where the target would see similar signal levels between legitimate and illegitimate APs. This approach is effective when the location of the target is known. On the other hand, attackers would leave their APs open even though the legitimate APs were secured since they aim to harvest as many user details as possible. Other details that would remain unchanged are the serial number of packets, timestamp, and range of beacon messages used by Kao *et al.* (2014) to develop the detection of fake APs algorithms.

4.4 Android Users' Practices, Knowledge and Compliance on Wireless Networks

The study aimed to look at users' practices, knowledge, and efforts in identifying attacks while exposed on wireless networks. This was done using a survey focusing on users' practice,

knowledge, device settings and compliance to best practices as recommended by host organisations.

4.4.1 Demographic Characteristics of the Respondents

(i) User Practices and Choices on Wi-Fi

Generally, most respondents (52%) admit to using Wi-Fi more than they do with cellular data. In comparison, 24% chose otherwise, and the other 24% have a balance between Wi-Fi and their data plan. Additionally, less (41%) would choose Internet access on Wi-Fi over Internet cables than the 52% of those who would choose Wi-Fi over their cellular data plans. Accessing the Internet through cables is safer than Internet access on Wi-Fi. However, 41% prefer Internet access on Wi-Fi compared to 38% who prefer Internet access through cables. This shows that most users are on Wi-Fi than cables, at which Wi-Fi is more risk than the counter option.

Respondents indicated several factors that affect their choices of Wi-Fi hotspots to connect, as presented in Fig. 14. Signal strength, speed, and free service were the leading factors ranging from 64%, 52%, and 48% of respondents. The majority of students had indicated free Wi-Fi, signal strength and speed as their factors in choosing an AP to associate with at 36%, 30% and 36%, respectively. In comparison, workers, on the other hand, had indicated signal strength (25%) as their lead factor, followed by free Wi-Fi (11%), speed (13%) and SSID (10%). These choices put users at high risk since the factors determining their Wi-Fi choices are not considered best practices. For instance, attackers could easily create fake AP with a stronger signal than legitimate APs. On the other hand, attackers could put their APs open and sometimes have faster Internet speed than the legitimate AP.

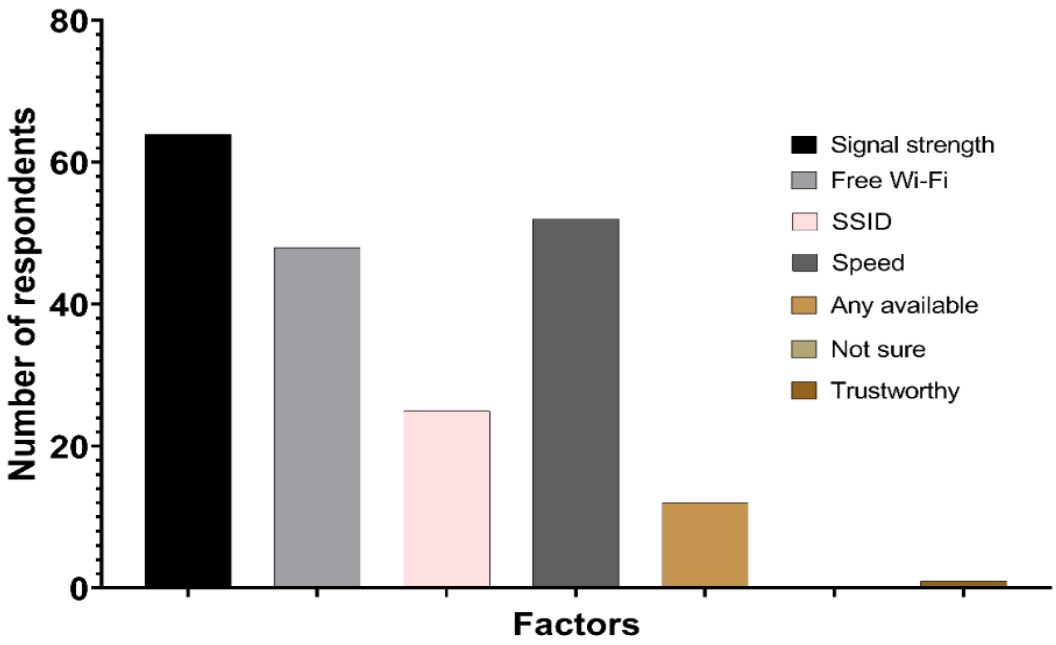


Figure 14: Factors users consider to associate with Wi-Fi APs

Moreover, 25% of respondents said their choices are affected by the Wi-Fi SSID. With a simple ETA, an attacker could mimic the SSID of the legitimate network or create deceiving names for users to connect. Surprisingly, only 1 % of respondents would consider network trustworthy as the factor to connect to Wi-Fi services. Although, most users (62%) would choose secured APs over open APs.

These findings mean that most users would connect to any available Wi-Fi as long as it has a stronger signal than the rest in the perimeter. The result would further suggest that many users (52%) connect to any available AP to test for speed, then later choose for highly performing AP in the perimeter. Hence, Wi-Fi users are at very high risk as they connect to Wi-Fi with a strong signal and free, which attackers could easily create. Generally, factors considered in choosing APs to associate with put users in danger. These results support the experimental study in Section 4.1, where users connected to a randomly created wireless network, and some added them to PNLs.

(ii) Users' Knowledge on Wi-Fi safety and Efforts to Identify Attacks

An analysis of users' knowledge focused mostly on what they would or would not do while connected to Wi-Fi APs. The emphasis is on assessing practices that might subject users to security risks. The risks include information they would share on Wi-Fi, device settings, and concerns about what information is being accessed while connected.

While most studies (Breitinger *et al.*, 2020; Chin *et al.*, 2012; Zaidi *et al.*, 2016) indicate that users knowingly or unknowingly share their details and sensitive information while connected on Wi-Fi, this research found that only 28% of the respondents would share their details. In comparison, 58% would not share their details. Amongst 58%, 33% had indicated a high level of confidence in their response. The minor group of 14% was not sure if they would share their details or not, as presented in Fig. 15. This shows that most users know the risks subjected to Wi-Fi networks. However, there is still 14% of the user group which is not sure of their practice which indicates that there are chances that they would fall victims due to poor practices.

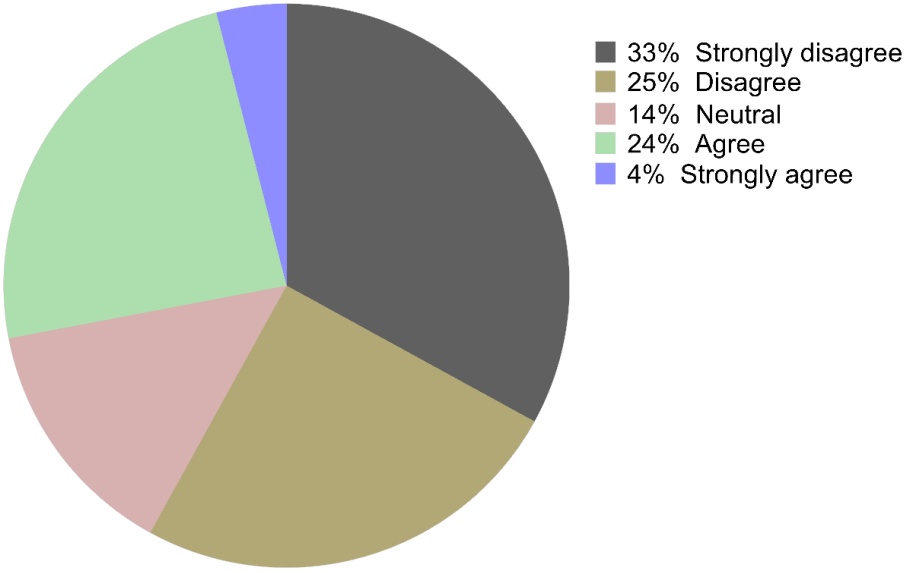


Figure 15: Users' response about sharing personal information while connected to Wi-Fi

On the contrary, more users (31% compared to 28% of those who would not share personal details) indicated that they would do banking operations while connected to Wi-Fi. The number of those denying going low to 47%. In the same case, respondents with high confidence lowered to 19%, from 33% of those who confidently denied sharing personal details on Wi-Fi as presented in Fig. 16. Thus, users are most likely to do banking operations on Wi-Fi than share their details. The two choices are considered dangerous as they act as the bridge to many other forms of attacks.

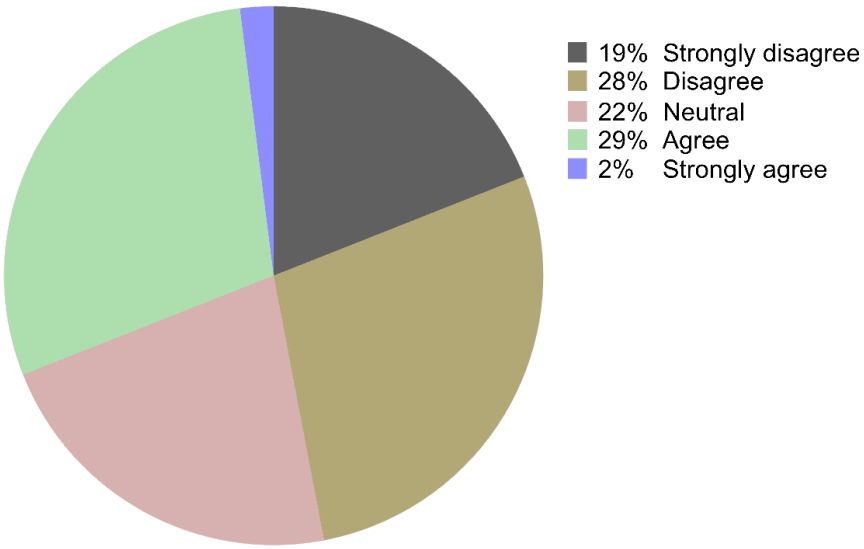


Figure 16: Users' response about doing banking operation on Wi-Fi

In a similar case, most users (55%) indicated that they would connect to Wi-Fi despite being warned of possible dangers that may cause to their devices. In addition, the same number of users (55%) allow sharing options while connected to Wi-Fi. The findings portray that, users would connect to any available Wi-Fi as long as they need internet service. Since 34% said they are usually not concerned with which Wi-Fi hotspot they are connected to. However, 54% of respondents said they were concerned. Nevertheless, many users would connect to Wi-Fi despite the described risks.

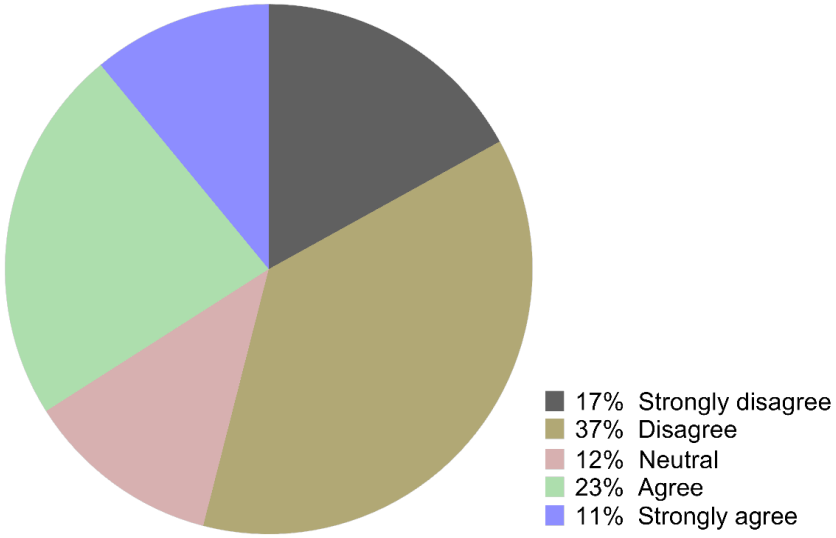


Figure 17: Respondents' expression about not being concerned with Wi-Fi AP they connect

Few questions focused on general comfortability while accessing the Internet on Wi-Fi. Many users (41%) indicated that they do not feel safe on Wi-Fi, while 29% said they feel safe, only 3% show high safety confidence, and 30% were not sure if they are safe. This indicates that users are aware of the risks associated with Wi-Fi, despite their practices indicating otherwise.

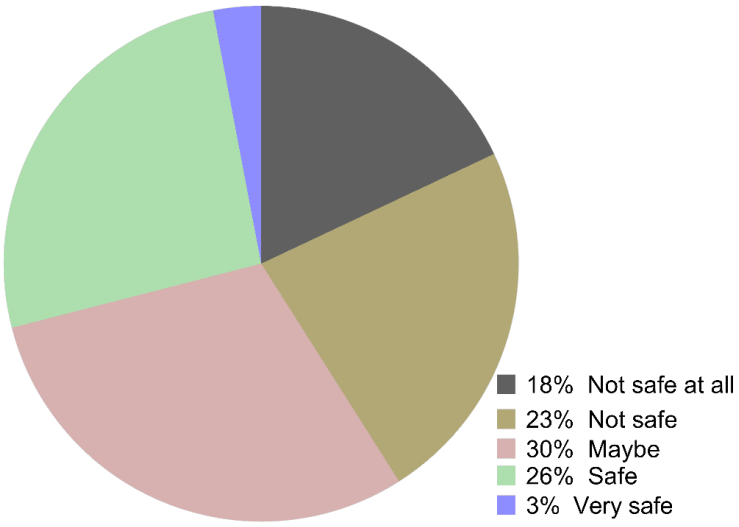


Figure 18: Respondents' expression about feeling safe on Wi-Fi

While most android devices' default settings are not adequate to ensure users' safety (Breitinger *et al.*, 2020), most would still not change the default settings (Ndibwile *et al.*, 2018). This shows poor cybersecurity practices and a lack of effort to ensure digital safety.

Generally, respondents are aware of risks associated with Wi-Fi, although their practices could not reflect. This is realised from user practices and how they typically set up their devices for Wi-Fi uses. For example, 53% of respondents' devices connect automatically to Wi-Fi hotspots, and 85% of respondents have set their devices to remember Wi-Fi hotspots they connect to, making it easy for spoofing attacks. Additionally, only 38% of respondents changed Wi-Fi-related settings, while 40% did not change, and 22% were unsure of their position. Among 38% who usually change Wi-Fi-related settings, most respondents (66.7%) change sharing options. The 54% change auto connection settings, 33.3% change secured/unsecured settings, 28.6% change download settings, 25.4% change VPN settings, 22.2% change Wi-Fi direct settings, and 3.2% do not change any of the settings. Even among those who would change Wi-Fi settings, very few would change VPN-related settings, which is considered the first line of defence in wireless networks.

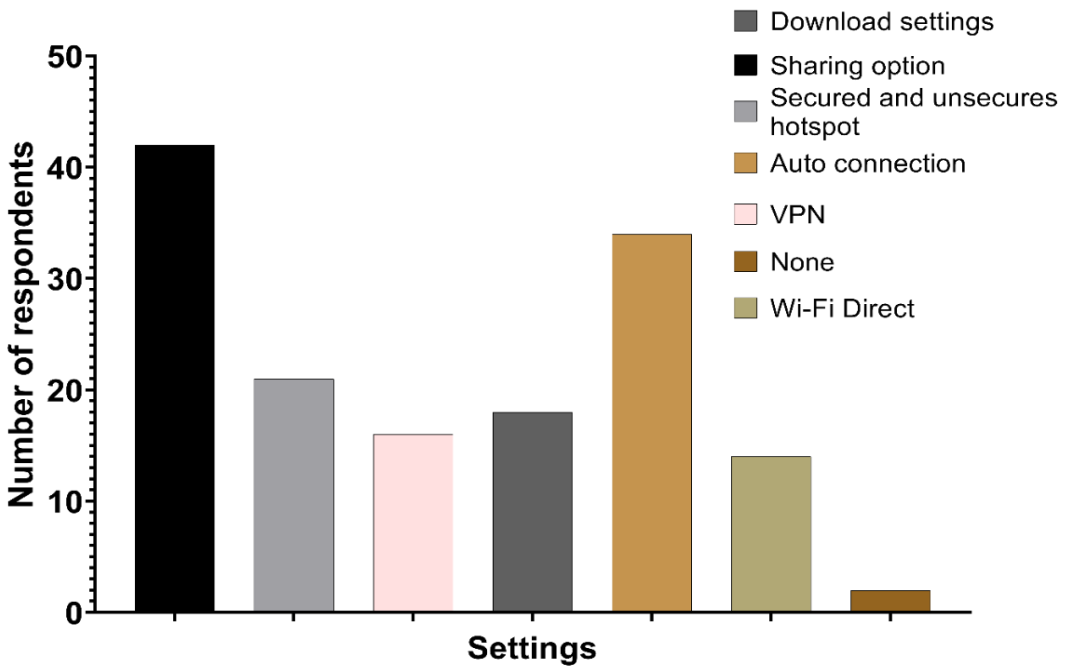


Figure 19: Wi-Fi-related settings that respondents change on their devices

4.4.2 Organisational Efforts

Following the observed user practices from the literature (Breitinger & Nickel, 2010; Breitinger *et al.*, 2020), The study further aimed to determine if organisations share best practices with users and whether users follow the recommendations on users' points of view. Most respondents (41%) said that their organisations do not share clear policies guiding their association with Wi-Fi. Contrary, 37% said organisations share clear policies, and 22% are neutral. Generally, most users are not informed of policies guiding their association with Wi-Fi. Furthermore, 42% of respondents said their organisations recommend best practices while using campus Wi-Fi, 43% voted against it and 15% were neutral. As presented in Fig. 20, among respondents who said their organisation recommend best practices, the majority (61%) follows the recommendation. In comparison, the other 15% do not. A reasonable number of respondents were not sure of their position.

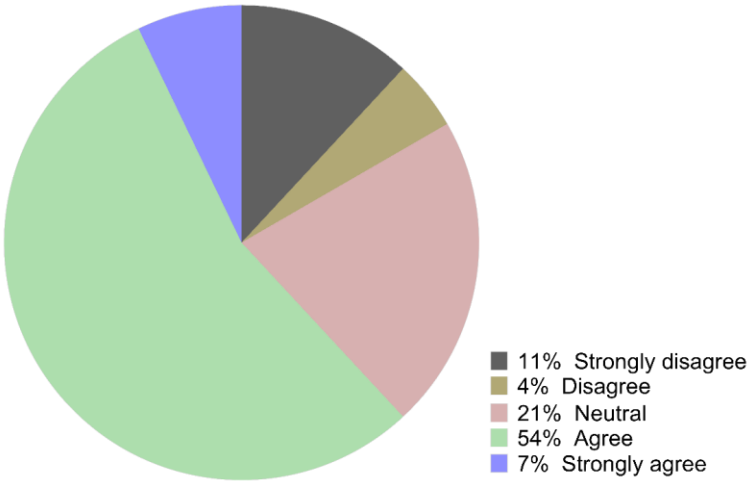


Figure 20: Users' response on whether they follow organisational recommendations or not

The study also examined the source of knowledge for cybersecurity, as presented in Fig. 21. More than a half (55) which is equivalent to 55% of respondents learn from formal education and class lecturers. Other sources include personal efforts (43%), online tutorial (35%), articles (26%), their organisations (15%), and regulatory (14%). Only 1% of respondents indicated to have learned from friends. There is little chance that users would learn about cybersecurity at workplaces since most indicate that they had learned in formal education.

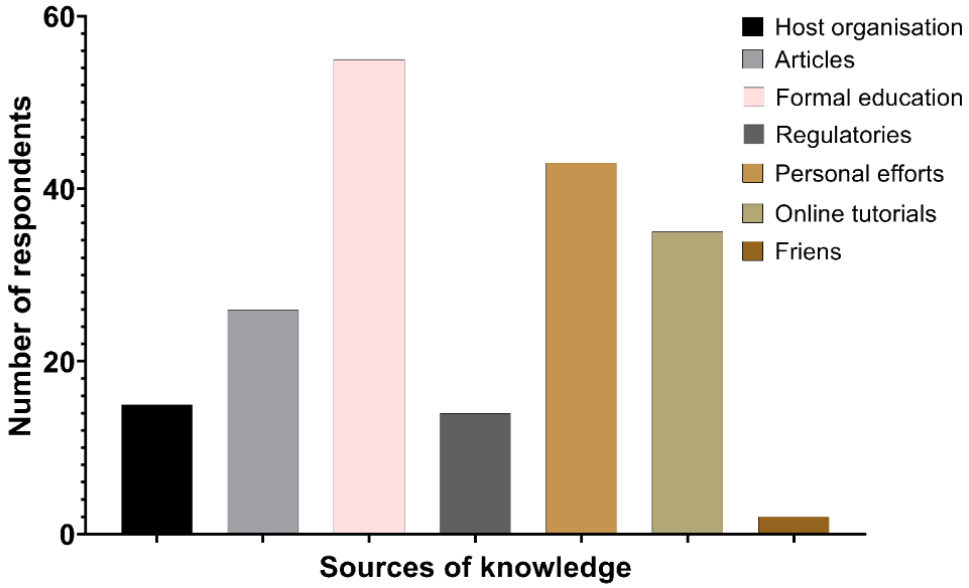


Figure 21: Respondents' sources of cybersecurity knowledge

4.4.3 Summary of Findings

Generally, this study of users' practices on Wi-Fi has found that most Wi-Fi users are aware of the risks associated with the wireless network. However, they would not care as the utility forces surpass the safety concerns. On the other hand, the factors that users account for their

association with Wi-Fi APs are not best recommended for safety on the internet. Furthermore, users would do essential operations while connected to Wi-Fi without considering the Wi-Fi APs. On the other hand, users' devices are mostly left with default settings which are usually not enough to secure devices against cyber-attacks. A combination of poor user practices and weaknesses in Android-based devices brings the need for spoofing attack detection systems that regular users cannot easily detect.

4.5 Application's Requirements Definition

4.5.1 Nature of Probes

The communication between two wireless devices goes through the association process. Several procedures are incorporated for a successful connection. The flow is presented in Fig. 22. The APs are bridges between a mobile station and other devices on the network. Before sending traffic through an AP, it must be in the appropriate connection state. The states are: (a) not authenticated or associated, (b) authenticated but not associated, and (c) authenticated and associated. For successful communication, the devices must be in the third state. To be in this state, the communicating devices exchange a series of wireless management frames.

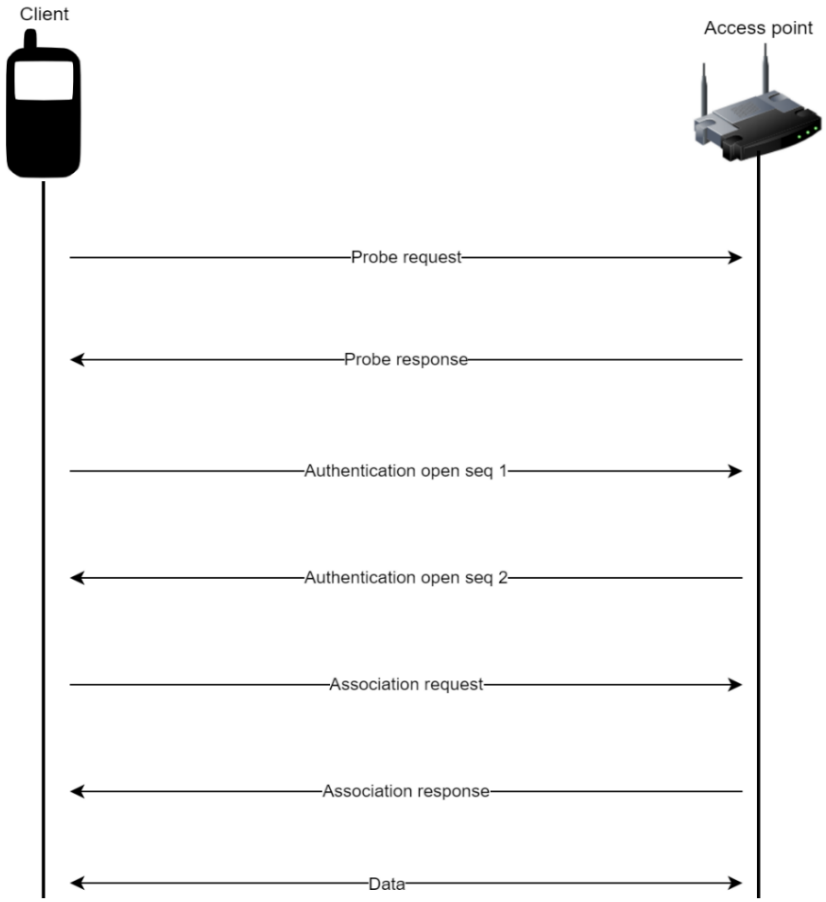


Figure 22: Client-AP association

A client device initiates communication by sending a probe request to discover wireless networks in the perimeter. The probe request announces the client device with its data rates and 802.11 capabilities. The APs in the perimeter receives the request and check if the client has at least one supported data rate. If they happen to be compatible, a probe response is sent advertising the SSID of an AP, supported rates, encryption details and many other capabilities information. After that, authentication and association procedures are carried out.

This study relies on details of probe responses to determine the legitimacy of the broadcasting APs in the perimeter. As presented in Fig. 23, probe responses usually contain the source BSSID with its advertising SSID, 802.11 management details, including probe timestamp and beacon interval, rates, Overlapping Basic Service Set (OBSS) colour, vendor-specific information and many more. This study relies on the capacity of these details to detect fake

Figure 23: The structure of probe response

APs in the perimeter. Therefore, the focus shall be on using details that can be captured in the Android phone.

4.5.2 User Practices and Android Security Configurations

Based on survey results in Section 4.4, most users had demonstrated poor practices while associating with wireless APs. First, their AP choices are driven by utility and usually do not identify the existence of fake APs in the perimeter. Previous studies explained that, despite users' awareness of associated risks in wireless networks, they still believe that the risks would not affect them (Swanson *et al.,* 2010). Taking the challenges of poor user practices and inadequate built-in security in Android-based devices into consideration, and the practice by Ndibwile *et al.* (2019), this study found the necessity of developing detection measures deployed before client-station association. This would be done while collecting probe responses from the list of broadcasting APs in the perimeter.

4.5.3 Functional and Non-functional Requirements

The gathered requirements in previous Sections 4.4.1 and 4.4.2 were grouped into functional and non-functional requirements that guide the development of most information systems (Ebert, 1997). Table 8 and Table 9 present the functional and non-functional requirements, respectively. The requirements are shaped based on users' practices and knowledge on wireless and network and the nature of wireless networks by looking at details of the broadcasts during association, i.e., probes or management frames.

Table 8: Functional requirements for the FakeAP Detector

Requirement	Description	Actor
Information gathering	An application should gather all broadcasting APs in the perimeter.	Android system, NIC, User
Store broadcasting APs	An application should store the details of broadcasting APs in each scan round.	Android application
Determining duplicate APs	An application should fetch the list of APs in the perimeter from the database and check for duplicates.	Android application
Determine the existence of fake AP	An application should determine the existence of fake AP in the perimeter based on details fetched from the database	User, Android application
Determine the type of Spoofing attack	An application should be able to determine if an attack is ETA or fake AP.	Android application

Table 9: Non-functional requirements for the FakeAP Detector

Requirement	Description
Security	• The system should not collect personal details from network traffic. • The system should wipe the details of broadcasting APs after determining their legitimacy.
Responsiveness	The system should respond timely
Scalability	The system should allow additional features in the future
Robustness	The system should be complete and able to handle normal malfunctions.
Operating System	The system should run on all supported Android versions.

4.6 System Modelling and Design

The modelling and design process of the detection prototype was guided by the study objectives and the designed requirements in the previous section. This chapter presents the conceptual use case diagram and the sequential diagram that had driven the prototype's development process in the forthcoming sections.

4.6.1 Conceptual Use Case

The study has indicated how different actors play role in the detection system based on the designed functionalities. The interaction between actors and functionalities can precisely be presented using the use case diagram. Use case diagrams depicts the interaction between actors and the system (Booch, 2005). External systems can also be mapped using use case diagrams. Figure 24 and 25 depict an abstract level of functionalities of the detection prototype in detecting ETA and fake captive portal, respectively. The prototype presents two main actors, the end-user and the Android system. All work interchangeably to accomplish the detection functionality. In the ETA detection, a user initiates the scanning process where the Android system takes over until scan results are stored. Then, the user starts the detection process, and the system takes over again until detection results are given. In the case of a fake captive portal, a user has to connect to a network, and the system handles the rest by generating fake credentials and submit into a captive portal. Later results are given out showing whether the captive portal is legit or fake.

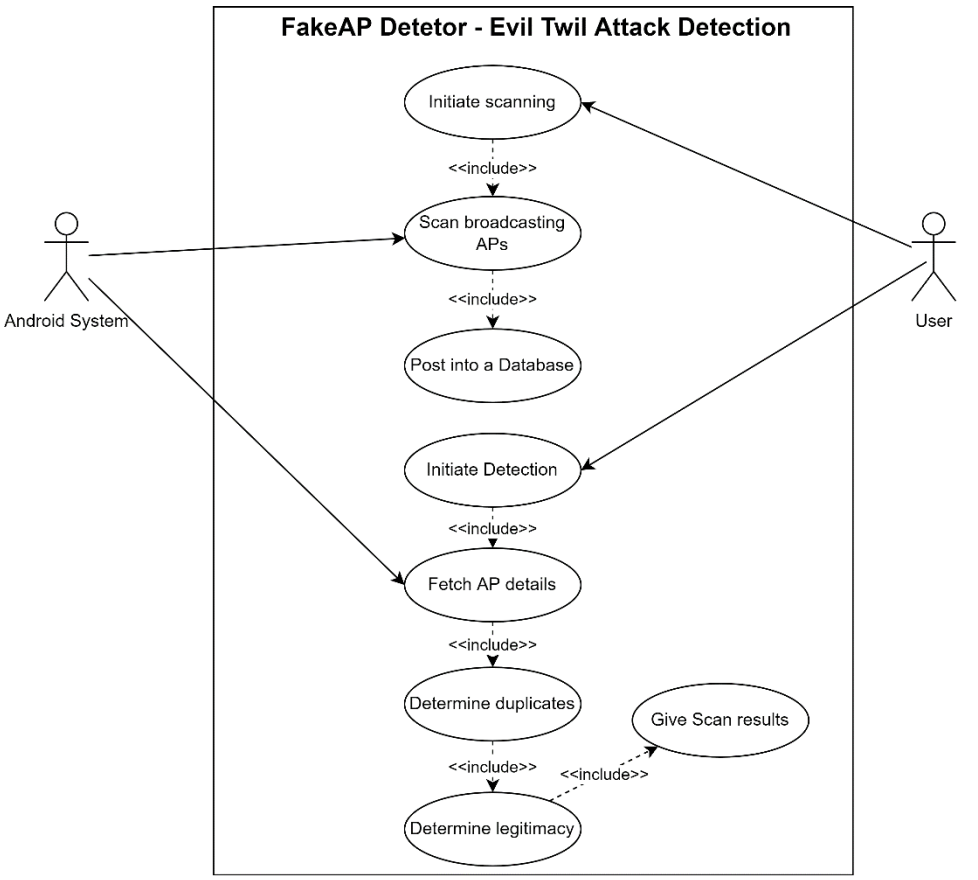


Figure 24: A use case diagram to show the interaction of actors in evil-twin detection

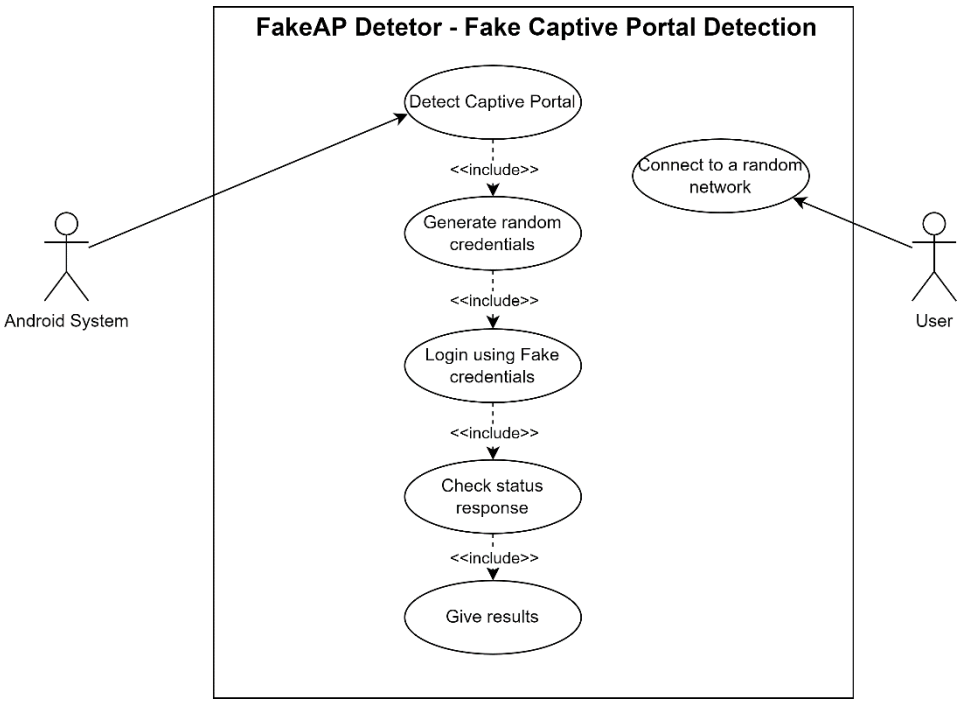


Figure 25: Use case diagram depicting the interaction of actors in the detection of fake captive portals

4.6.2 Sequence Design

The study presents the sequence diagram to see a dynamic view of system activities. Sequence diagrams are known for presenting how objects interact within a system. Software developers

and business professionals use these diagrams to understand a new system's requirements or document an existing process (Booch *et al.*, 1997). Initially, a user initiates the scan process, after which the scan results are posted into a database. Similarly, the application displays the scan results posted into a database. In the second act, a user initiates the detection process. The detection process fetches the details from the database, after which the database responds by returning the requested details. Finally, the algorithm to compare the legitimacy of each returned result is launched, and the detection results are then displayed to a user.

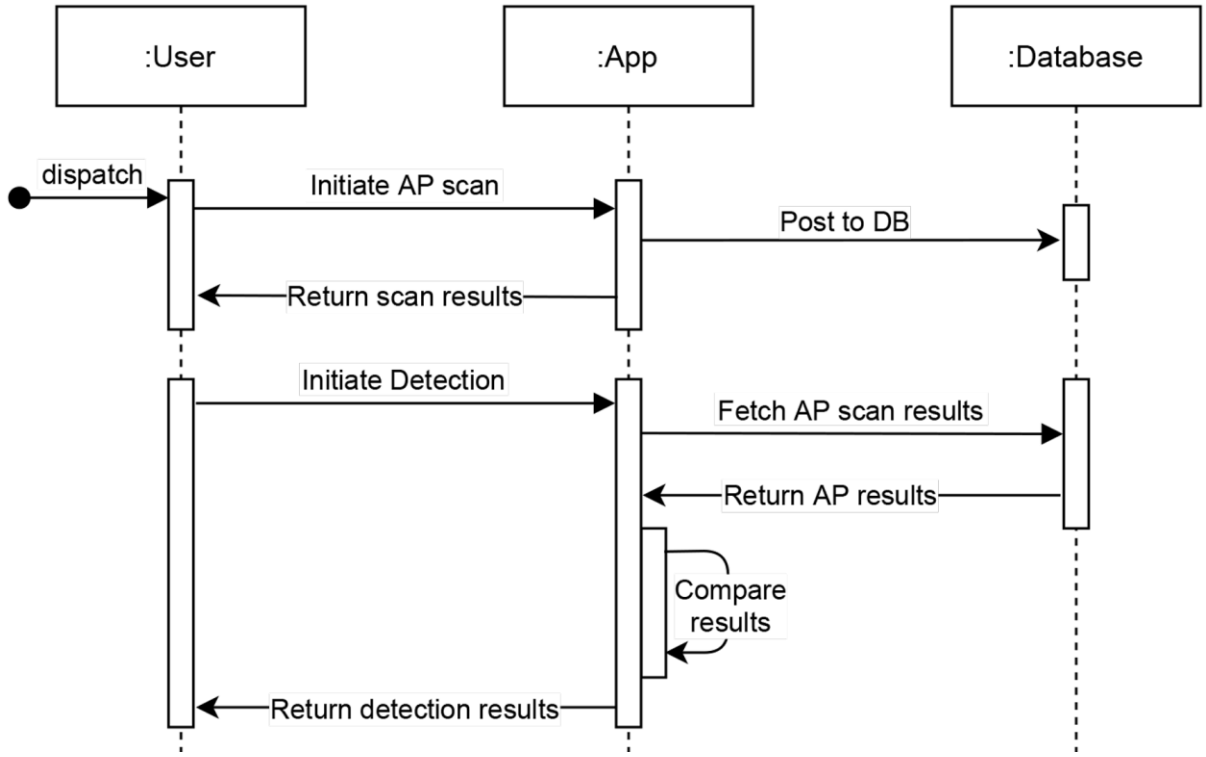


Figure 26: Sequence diagram for the detection prototype

4.7 System Implementation

The designed system was developed and implemented for Android-based devices as a proof of concept. Tools and methods were selected based on the needs of Android systems. This section presents the implementation of the developed detection prototype named the *FakeAP Detector*: An Android-based application that detects fake APs in the perimeter.

4.7.1 System Assumptions

The study presents the prototype focusing on Android-based devices. However, it might also be feasible for other mobile devices as the challenges are similar. Additionally, users of most mobile device OSs in the market have demonstrated similar behaviours (Breitinger *et al.*, 2020). Furthermore, since the solution uses the built-in Android phone features and development tools, few features were extracted from broadcasting APs in the perimeter due to

the Android OS limitations. More features could be extracted with the help of an external device or by rooting the device. However, the few features collected were sufficient to develop the *FakeAP Detector*. The presented prototype detects fake APs that broadcast during the scanning process period.

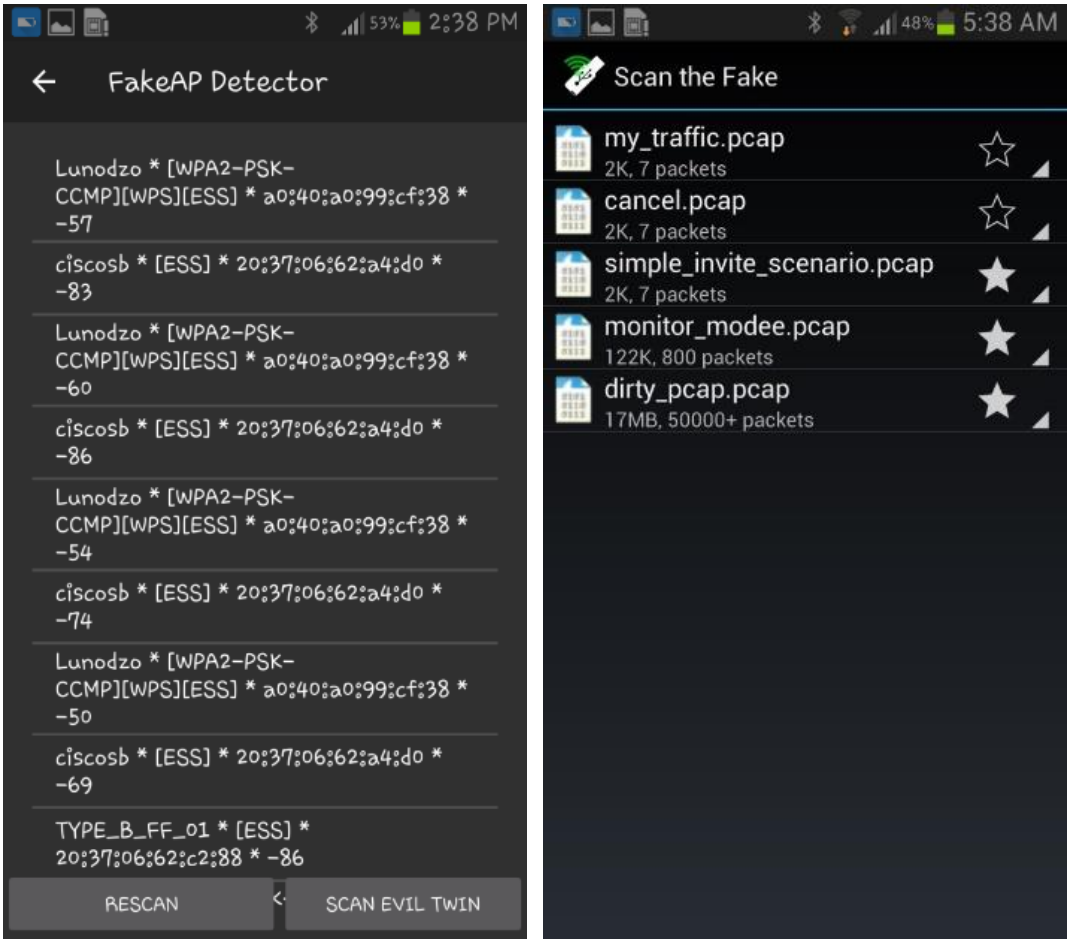
On the other hand, in the captive portal networks with captive portals, we have assumed that a captive portal pops up automatically after a user is connected to a network. The designed captive portal is meant to read usernames and passwords from users. The HTTP responses could be used to determine the legitimacy of broadcasting AP and its captive portal. The same was implemented in our experimental study. Throughout the scanning process for broadcasting APs and determining their legitimacy, the devices involved in the process are assumed to be in a static position. Network spoofing attacks exist in various forms. Some behave differently based on tools used to simulate attacks. This study did not cover spoofing attacks created using deceiving SSIDs that do not imitate the features of the legitimate network AP. The study assumes further that an attacker creates fake APs mimicking AP details from the legitimate network. Since it is unlikely for an attacker to simulate ETA before the legitimate APs are broadcasted.

4.7.2 Information Gathering and Database Structure

(i) AP Details Capture

The initial step towards detecting fake APs includes scanning active APs broadcasting in the perimeter. The process was done in two ways, first by using an external NIC and second by using the Android's built-in NIC. The scan results with an external NIC were presented in Fig. 27(b). Each session of scan stores the packets in a different file. This process was adapted from the *Android PCAP* Library available in the Google Play store. Figure 27(a) shows the built-in wireless card's scan results. The SSID, BSSID, RSSI, and capabilities were captured and stored in the SQLite database in rounds.

The study could not work further with the raw pcap files due to the limitation of the *io.pkts*. After the capture, the prototype was supposed to filter packets relevant for the detection application, i.e., the probe response from APs. Unfortunately, the *io.pkts* did not support parsing of the 802.11 protocol as of August 2021.



(**a**) (**b**)

Figure 27: Scan results of the FakeAP Detector

(iv) Data Storage

The AP information captures were stored in the single table of the SQLite database. The designed table covered basic information from the AP and added a few, which helps identify scans uniquely. Sample scan results are shown in Fig. 27 as stored in the database. The probes are scanned in ten rounds (the number of rounds could be adjusted to any) to capture different RSSI values. Each round is then posted into the database. When the scan rounds are finished, the detection process starts. The detection method retrieves data from the database and uses the details to compare each AP's legitimacy. To avoid overloading an application with useless data, the database is wiped during the start of every scanning process.

id	ssid	bssid	level	capabilities	round	comment	time
F...	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	kisarouter	78:54:2e:92:2f:ce	-59	[WPA2-PSK-CCMP][WPS][ESS]	1	0	22:40:27.088
2	NMAIST-CDAC 17	14:02:ec:4c:2c:90	-62	[WEP][ESS]	1	0	22:40:27.119
3	VLIR_L CLASES	dc:9f:db:62:a5:a0	-73	[ESS]	1	0	22:40:27.123
4	kisarouter	78:54:2e:92:2f:ce	-62	[WPA2-PSK-CCMP][WPS][ESS]	2	0	22:40:28.753
5	NMAIST-CDAC 17	14:02:ec:4c:2c:90	-64	[WEP][ESS]	2	0	22:40:28.780
6	VLIR_L CLASES	dc:9f:db:62:a5:a0	-73	[ESS]	2	0	22:40:28.785
7	kisarouter	78:54:2e:92:2f:ce	-63	[WPA2-PSK-CCMP][WPS][ESS]	3	0	22:40:30.404
8	NMAIST-CDAC 17	14:02:ec:4c:2c:90	-62	[WEP][ESS]	3	0	22:40:30.422
9	VLIR_L CLASES	dc:9f:db:62:a5:a0	-72	[ESS]	3	0	22:40:30.427

Figure 28: Sample scan results (three rounds) as it can be seen in the SQLite database

4.7.3 Application Interface

The *FakeAP Detector* has simplified interfaces designed for better usability and performance. On landing, a user has an interface to initiate a scan of broadcasting APs in the perimeter or capture the packets. The home page of the application is presented in Fig. 29. The application shows the status of NIC if it is plugged on the external USB or not. Below it, channel numbers are presented, which dictates the channel lists from which the application would collect details. Finally, the item below it shows the logging status. When pcap files are logged in, the status changes to active, these functionalities were adapted from the *Android PCAP* library. At its bottom, two buttons are presented. One in the left for scanning the logged pcap files. Another button in the right for scanning details of broadcasting APs. The results for the two buttons are presented in Fig. 27. Figure 27(a) shows the results of broadcasting APs in the perimeter. The *FakeAP Detector* captured the SSID, BSSID, security protocols, and signal strength.

On the other hand, Fig. 27 (b) presents pcap capture using an external NIC. Captures are stored in the form of pcap files. Unfortunately, we could not develop the detection algorithms for the pcap files due to the current limitations of the *pkts.io* package.

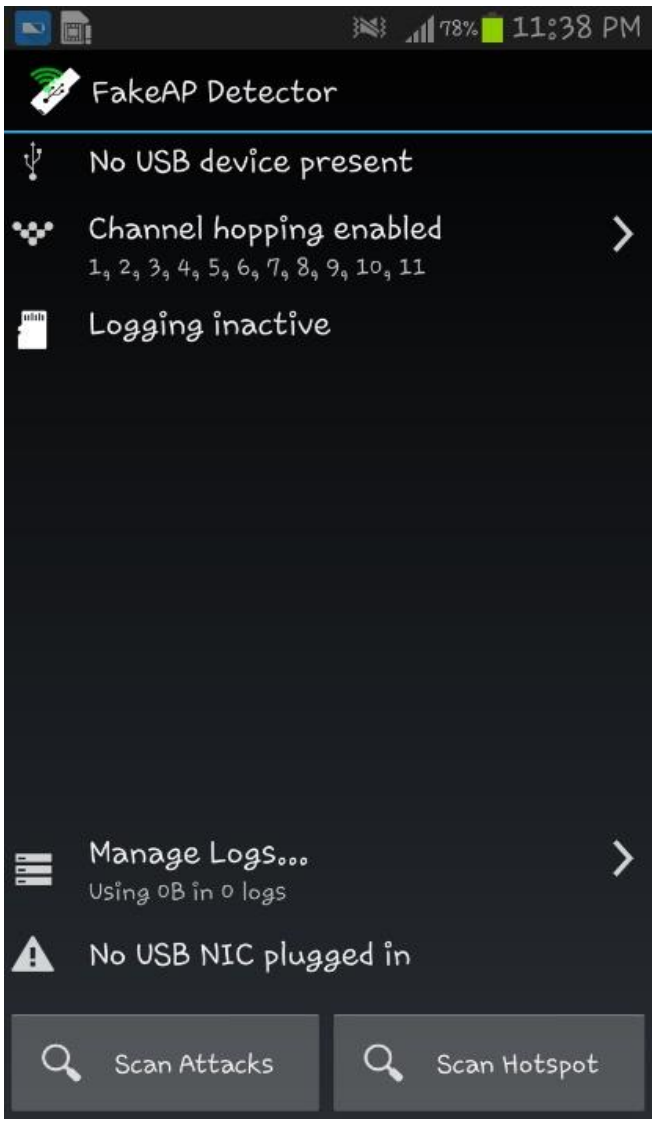


Figure 29: The homepage of the FakeAP Detector

4.7.4 Detection Approaches

The built-in Android Wi-Fi card can capture details of the broadcasting APs. However, we captured and used a few for our detection prototype: SSID, MAC (BSSID), capabilities and RSSI. The capabilities include encryption referred to as ENC, CIPHER, and authentication type referred to as AUTH. These details help us detect attacks in which an attacker mimics features of legitimate APs. For example, most attacks targeting wireless networks using the *airckack-ng* suite mimic SSID, MAC, Channel and others. Fortunately, they cannot mimic RSSI since they are generated based on the power of AP. So, they cannot be manipulated by adversaries in wireless networks (Chen & Yang, 2012; Madani & Vlajic, 2021; Tang *et al.*, 2017). Additionally, in authenticated APs, most attackers ignore mimicking AP capabilities information since they want to leave their network open for clients to associate. Despite the possibility of creating an ETA with WEP, WAP, or WAP2.

The *FakeAP Detector* starts by scanning for available APs in the perimeter. The details are then stored in the database as they are captured. The AP details are retrieved from the database and compared for similarities when the scan is complete. If two or more APs broadcast with the same SSID and BSSID, then one or more among them is questionable. To further be sure, the remaining two features, RSSI and capabilities of the twin-APs, are compared. If there are differences in RSSI or capabilities, it is confirmed that fake AP(s) exist in the perimeter. Unfortunately, the RSSI value is not constant and is affected by several environmental factors, such as obstacles (Madani & Vlajic, 2021).

Furthermore, capabilities information may not return the desired result in open networks since both legitimate and illegitimate APs broadcast with the same capabilities information. In addressing these challenges, the study uses the difference in the means of RSSI values to detect fake AP. Two alternatives were employed. The suggested solutions are classified into two categories based on network characteristics: open and closed networks detection.

(i) Open Networks

According to data captured in an Android phone during the open network experiments, these networks do not employ any security mechanisms. As a result, both legitimate and bogus networks may have similar capabilities because an attacker typically creates an open network to which anyone can connect. As noted before, it is unlikely for an attacker to create an ETA before the legitimate network broadcasts its APs. In this case, the solution is to rely on the RSSI value, which is not static. The problem could be solved by scanning for broadcasting APs in multiple rounds, resulting in a cluster of RSSI values for comparison. The FakeAP Detector scans in ten rounds, with the results stored in the database after each round. The solution works by comparing the average RSSI values of duplicate broadcasting APs to the average RSSI value of the first broadcasting AP. Because an ETA imitates the original networks, the second broadcasting AP among the twin-APs with a different RSSI value could be fake.

This approach benchmarked the RSSI value of the first AP to broadcast among the duplicate APs. The benchmarked value will then be used to retrieve the RSSI values from the same SSID, which fall in the range of (+5 and -10) dBm. Finally, the difference of the means for the collected RSSI values from duplicate APs will be calculated. If the difference exceeds three dBm units, then one duplicate AP is fake. This range is defined based on RSSI fluctuations captured in the experimental observation, as presented in Fig. 30. The scanning was simulated in twenty rounds, each round capturing broadcasts in one second.

For instance, with AP1, the benchmarked value is -84 dBm, at which the strongest signal was -81 dBm. The weakest was -93 dBm making a difference of (+3 and -9) dBm in the high and low signal fluctuations, respectively. The AP2 had a difference of (+16 and -4) dBm from its benchmarked value of -83. The AP3 had a difference of (+1 and -8) dBm. Lastly, the AP4 had a difference of (+1 and -19) dBm, as presented in Table 10. The highest and lowest signal strength range was calculated to be 12, 20, 9, and 19 for the AP1, AP2, AP3, and AP4, respectively. These make an average range of 15 units of dBm to which signal strength could fluctuate. Our study used the +5 and -10 difference to select RSSI values to calculate the mean for a specific AP.

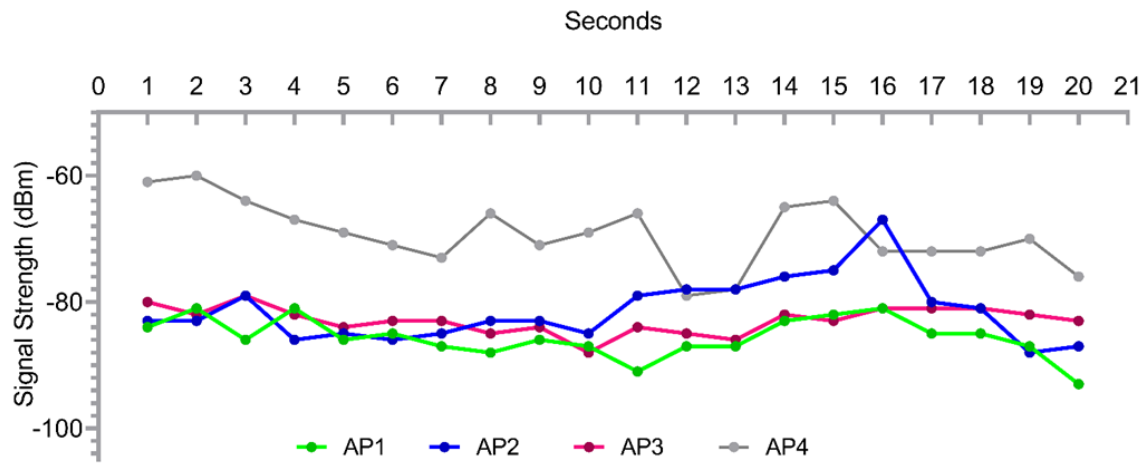


Figure 30: Signal strength fluctuations along with time in seconds

Table 10: The presentation of benchmark RSSI value, highest signal, lowest signal, and the range between highest and lowest signal

	AP1	AP2	AP3	AP4
Benchmarked value	-84	-83	-80	-61
Highest signal	-81	-67	-79	-60
Lowest signal	-93	-87	-88	-79
Range (High – Low)	12	20	9	19

(ii) Closed Networks

In closed networks, the detection is straightforward under the assumption that attackers create open fake APs for everyone to connect. All closed network broadcasts with security information indicating the protocol and encryption used when this is the case. Contrary, an open network would broadcast without showing the security protocol and encryption used. Hence, to detect fake APs, the study first checks for twin-AP. If they exist, then capabilities information is compared. When the difference is noticed, the AP without security protocol and encryption information, which came when the other started broadcasting, is marked as fake. In cases where an attacker creates an ETA with security information similar to legitimate AP, the

comparison of RSSI values is done on top of capabilities (security) information to increase efficiency in detecting fake APs. The detection process is presented in Fig. 31.

(iii) Networks with Captive Portal

In this context, the solution is built under the foundation of web protocols and Android's *WebView* class. In other words, the detection relies on response messages from Hypertext Transfer Protocol (HTTP). Two HTTP responses are considered: the login attempt failure message and the success message. During authentication on the web, error message 401 is given for unauthorised users and 403 for forbidden requests (Ndibwile *et al.*, 2017). Then, response code 200 is given when the submitted request has succeeded. Therefore, this research focuses on error response 401 captured by the *WebResourceResponse* class of the *WebView* package. These implementations are presented in Fig. 33.

To determine the legitimacy of the network, the *FakeAP Detector* generates random login credentials using the JavaScript code presented in Fig. 34 which are then submitted into a captive portal. Next, the response codes of the portal are determined. The first and second lines of the script generate random usernames and passwords. Then, the third and fourth lines auto-fill the form inputs by identifying form input names and submitting values generated in the first and second lines. Lastly, the values are automatically submitted to the captive portal web server in the fourth line. After submission, when the success code is returned or no error messages are received, the captive portal originates from a fake network (Fig. 34).

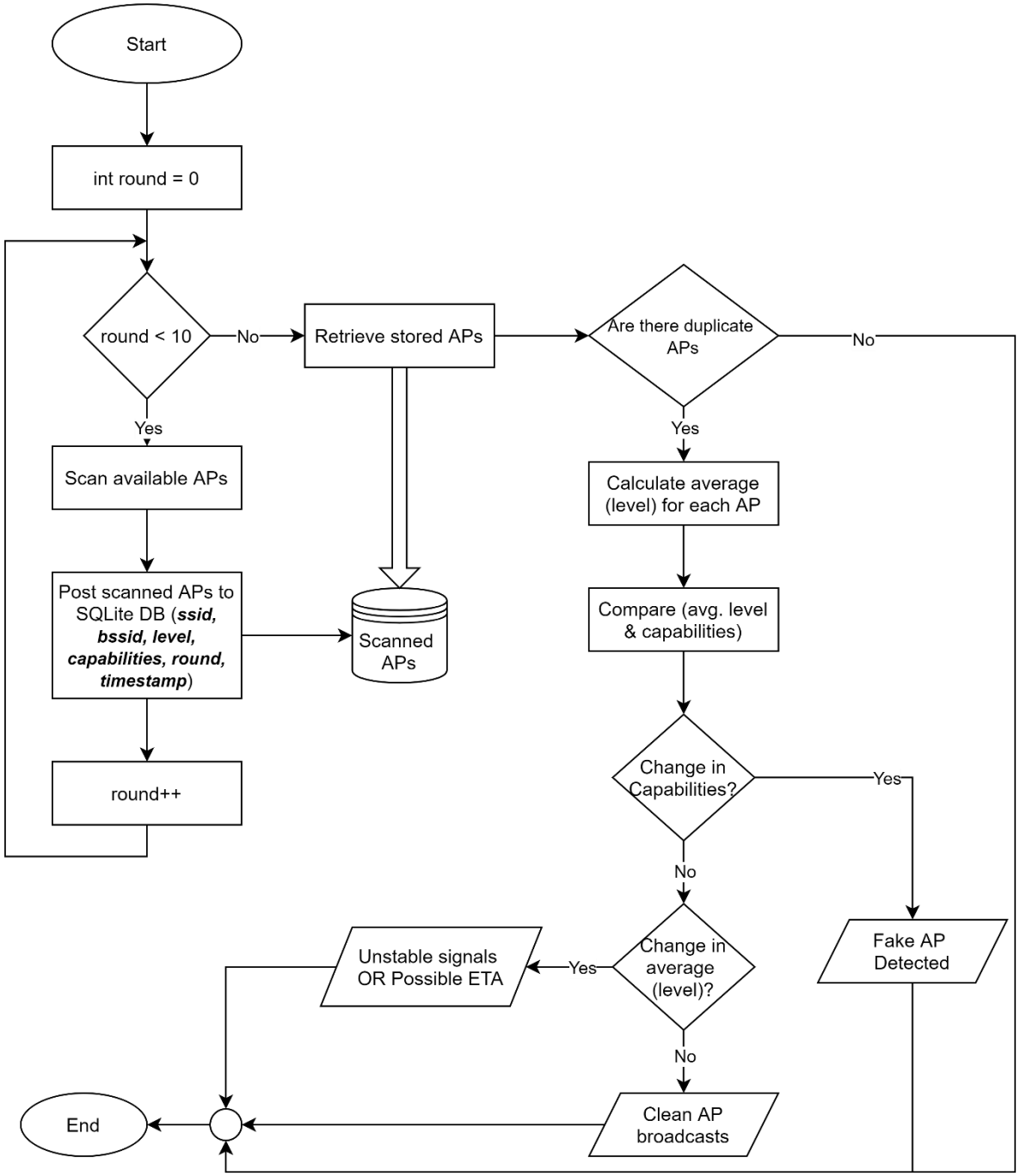
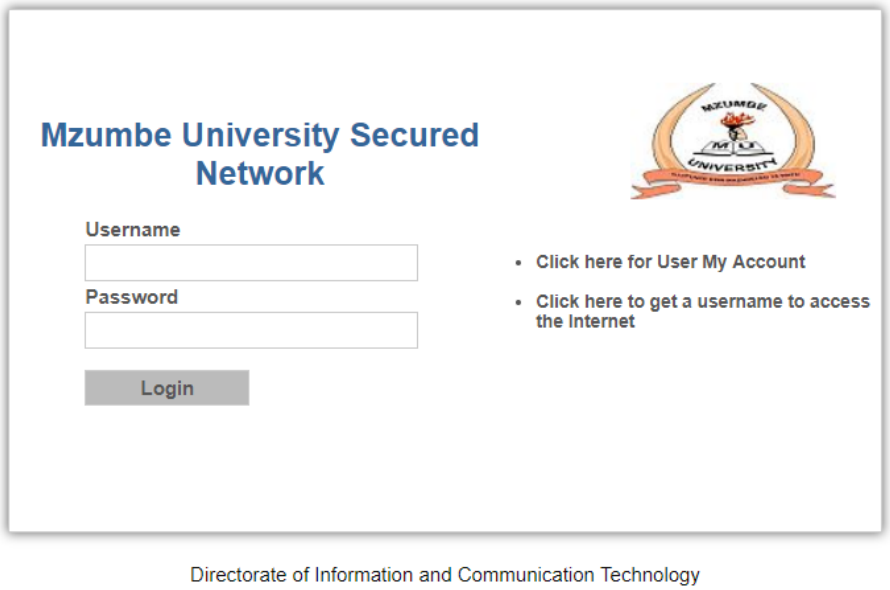
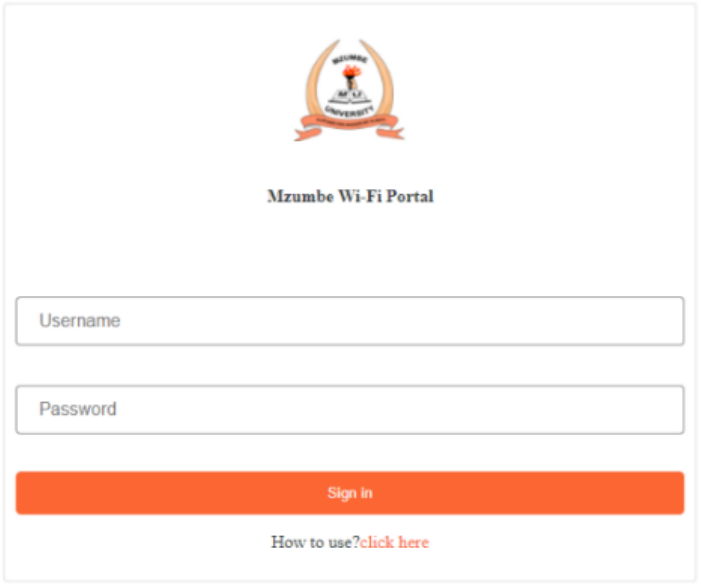


Figure 31: Fake AP detection flowchart



(**a**)



(**b**)

Figure 32: The captive portal screenshot: (a) Legitimate captive portal and (b) Fake captive portal

```
WebResourceResponse errorResponse = null;
int statusCode = errorResponse.getStatusCode();
if(statusCode != 401){
   Toast.makeText(getApplicationContext(), "Fake Captive Portal"+statusCode,
Toast.LENGTH_SHORT).show();
}else{
    Toast.makeText(getApplicationContext(), "Legit Captive Portal"+statusCode,
Toast.LENGTH_SHORT).show();
}
```

Figure 33: Script to detect fake captive portal

```
String username = CaptivePortal.randomString(10);
String password = CaptivePortal.randomString(10);
webView.loadUrl("javascript:document.getElementsByName('username').value = "+username);
webView.loadUrl("javascript:document.getElementsByName('password').value = "+password);
webView.loadUrl("javascript:document.forms['submit'].submit()");
```

Figure 34: JavaScript code automating captive portal the login process on the

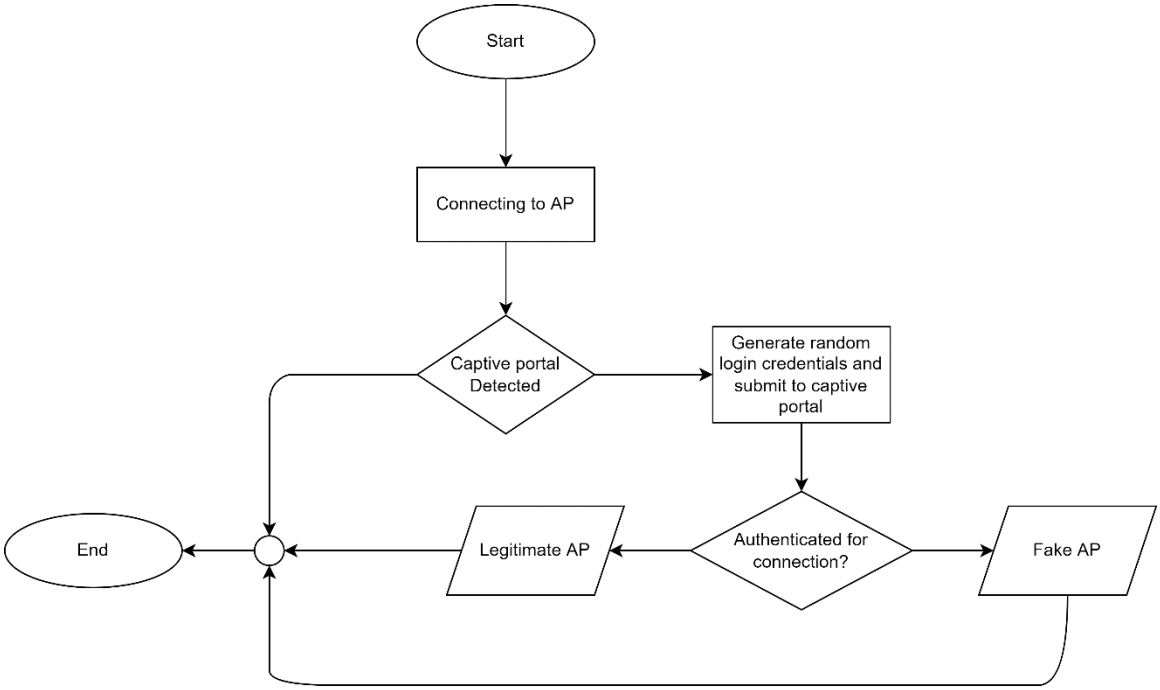


Figure 35: Fake captive portal detection flowchart

Since the application collects details of the broadcasting APs at a certain period and then stores them into a database, the detection prototype fetches the scanned details from the database. First, we use SQL statements to determine duplicate APs in each round based on SSID and BSSID information. These statements are presented in Fig. 36, 37 and 38. The duplicate APs are then compared with their details to determine their legitimacy based on the flowchart shown in Fig. 31. This process was categorised into open networks and closed networks. In closed networks, usually, the legitimate network has WEP, WPA or WPA2 enabled. The first line into the detection approach starts with comparing the capabilities information. The SQL statement

to fetch duplicate APs with different capabilities is presented in Fig. 36. The implementation is further explained with the pseudo-code in Fig. 39.

```
SELECT ssid, bssid from accesspoints
GROUP BY ssid, bssid
HAVING min(capabilities) <> max(capabilities)
```

Figure 36: The SQL statement that returns duplicate APs with different capabilities (SQL 1)

An attacker usually has most of the details similar to legitimate AP in open networks, including the capabilities. In this case, we focus on RSSI values. We created a temporary table (view) in the database where AP details are stored. The view stored all duplicate APs whose capabilities, SSID and BSSID, were the same. The implementation is presented in Fig. 37.

```
CREATE TEMP view duplicateCapabilities AS
    SELECT ssid, bssid, capabilities, level,time FROM accesspoints a1
WHERE EXISTS(
    SELECT 1 FROM accesspoints a2 WHERE
        a1.ssid = a2.ssid AND a1.bssid = a2.bssid AND
        a1.capabilities = a2.capabilities AND capabilities = \[ESS]\)
EXCEPT
    SELECT ssid, bssid, capabilities, level, time FROM accesspoints a1
WHERE EXISTS(
    SELECT 1 FROM accesspoints a2 WHERE
        a1.ssid = a2.ssid AND a1.bssid = a2.bssid AND
        a1.capabilities != a2.capabilities)
```

Figure 37: The SQL statement that creates a view that stores duplicate open APs (SQL 2)

From the list of duplicate APs, we compare their RSSI values determining the legitimacy of each AP. Figure 38 presents the SQL statement, which determines the difference in average RSSI values based on the benchmarked RSSI value.

```
SELECT * FROM duplicateCapabilities dp1
WHERE EXISTS (
SELECT 1 FROM duplicateCapabilities dp2
WHERE
dp1.ssid = dp2.ssid AND
dp1.bssid = dp2.bssid AND
dp1.capabilities = dp2.capabilities AND
average_level NOT BETWEEN
(storeFirstSignal-10)  AND (storeFirstSignal+5))
```

Figure 38: The SQL statement that returns results of APs with average levels not falling in the defined range (SQL 3)

Finally, the duplicate APs with average RSSI values not falling in a defined range are marked as fake. The presence of fake AP is determined by the SQL statement in Fig. 38, which is further detailed with the pseudo-code in Fig. 40.

In SQL statements presented in Fig. 36 and 38, the statement that returns the number of rows from one and above indicates that the results contain fake APs.

```
begin
  run SQL 1
  if(results >= 1)
    then
      Fake AP Detected
    else
      Clean
  end if
end
```

Figure 39: The pseudo-code that shows detection of fake AP in a closed environment

```
begin
  run SQL 2
  if(results >=1)
    then
      run SQL 3
        if(results >= 1)
          then
            Fake AP
          else
            Clean
        end if
    else
      Clean
  end if
end
```

Figure 40: The pseudo-code that shows the flow to detect fake APs in open networks

4.8 System Validation

We have designed three experiments following the designed attack detection methods developed. The first two setups involve the testing approaches for the fake APs based on open and closed network structures. The third setup was for the network that has implemented a captive portal to authenticate clients. These experiments used a wireless router, Android device, wireless USB adapter, and a PC. The tests were simulated assuming a constant position of an adversary (attacking machine), legitimate APs, and the detecting device.

In the first experiment, we broadcast APs (legitimate and illegitimate) in a closed network. The illegitimate network of APs mimicked the legitimate network. Similarly, we broadcast an open network with its details, imitating the second experiment's legitimate network. To obtain reliable results, we did a hundred test experiments, and in each, we calculated accuracy and detection speed. Finally, the average of each performance indicator was obtained.

In the third experiment, we simulated a network with the captive portal. The fake captive portal was then tested with fake credentials in the *FakeAP Detector*. Finally, the HTTP responses were captured to determine the legitimacy of the captive portal.

To obtain reliable results and build evident conclusions from the detection prototype, we run the scans and detection in a clean and attacked environments to see the test scores. The experiment shows 2% and 1% false positives in open and closed networks respectively and 98% and 99% true negatives in open and closed networks respectively, as presented in Table 11 and Table 12. During the tests in clean environment, the average detection speed of the prototype was calculated to be 24.98 milliseconds when AP scan results were already stored in the database. We noticed further that; time spent in detection is affected by the number of broadcasting APs. The presented results are based on a network with a total of seven broadcasting APs in the perimeter.

Table 11: Detection test results in open network

Condition	Accuracy in Percentage
True positives (Attack present and detected)	99%
False positives (Attack not present and detected)	2%
True negative (Attack not present and not detected)	98%
False negative (Attack present and not detected)	1%

The same experiment was conducted in a network with an attack. In this case, we considered two different attacks: An attack targeting open APs and a second attack targeting closed APs. In the open APs, the results show that the *FakeAP Detector* has achieved 99% true positives and had 1% false negatives, as presented in Table 11 with an average of 24.64 milliseconds of detection time. This experiment had seven broadcasting APs with one being fake. In a closed network, the results show that the FakeAP Detector has achieved 99.7% of true positives and had 0.3% of false negatives with an average of 5.78 milliseconds of running time. The detection time in this attack was significantly low since the algorithm first checks for differences in security information. If the difference is noticed, then the process ends (Fig. 31). The detection accuracy for this experiment is shown in Table 12.

Table 12: Detection test results in closed network

Condition	Accuracy in Percentage
True positives (Attack present and detected)	99.7%
False positives (Attack not present and detected)	1%
True negative (Attack not present and not detected)	99%
False negative (Attack present and not detected)	0.3%

In the captive portal, our detection accuracy was 88 % in one hundred tests done. The captive portal reads and returns a success message immediately after the credential is successfully posted into a database without verifying the details. Here, the performance was highly affected by the availability of the Internet and server where the fake captive page was hosted.

These results were also calculated to obtain precision, recall and F1-Score metrics scores. The FakeAP Detector achieved a 98% precision, 99% recall and 98.4% F1-score in open networks. On the other hand, it has achieved 99% precision, 99.7% recall and 99.3% F1-score in a closed network. These performances are competitive compared to similar recent studies, including the work by Madani and Vlajic (2021) as presented in Table 13.

Table 13: Performance comparison between the FakeAP Detector and Deep

	FakeAP Detector		Deep Learning	
	Open network	Closed network	Day classifier	Night classifier
Precision	0.980	0.990	0.97	0.99
Recall	0.990	0.997	1.0	1.0
F1-Score	0.984	0.993	0.98	0.99

In the detection approaches presented by Ballai (2010), Segura and El-Moussa (2014) and Bryksa and MacMillan (2015) still adversaries could manipulate beacon details, and authenticate themselves into accessing the network. Approaches by Ballai (2010) focus on protecting the host network against rogue AP. However, attackers may create fake APs by mimicking details of legitimate networks without the need to associate with the host network. In this case, the performance of the proposed approaches would be very poor in this kind of attack.

The work by Matte *et al.* (2015) presented the fraction to which the spoofing attacks could be successful. The efficiency of the presented approach is at 95%, with dependence on geotagged

service and geolocation features. Our detection approaches in open and closed networks outperform this work by 3% and 4% detection accuracy, respectively.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The study aimed to develop a detection application against hotspot spoofing attacks in wireless networks, focusing on Android devices. The study evaluated users' experience and practices while connecting to wireless networks by looking at several factors, including their knowledge, practices, and efforts to identify possible attacks. On the other hand, users' susceptibility was tested by creating a deceiving network in our study area. Furthermore, functional and non-functional requirements were developed based on the study results.

This study found that users are susceptible to network attacks since they connect to any available network in the perimeter. Various reasons are associated with this, a few of them being the availability of free Wi-Fi or the associated internet costs by their cellular networks. Users are also susceptible to deceiving networks. The majority connected to our fake networks, and some even put them into the PNL. On the other hand, users demonstrated a reasonable level of skills on wireless security and associated risks despite users demonstrating to behave disparately while associating with wireless networks.

The study found that attackers targeting wireless networks usually behave in very similar ways since most aim to harvest user data or target companies for financial benefits. The need for personal data drives most attacks to be ETA since attackers would not want to be suspicious in a network. On the other hand, they would like to lure users into connecting to their networks.

This research study proposed a prototype of an Android application to detect hotspot spoofing attacks in wireless network settings using features collected from broadcasting APs. The proposed system collects SSID, BSSID, RSSI, and capabilities information which is being compared. The comparison decisions are made depending on the nature of the network. APs were compared for the differences in capabilities, and RSSI values were used in open and closed networks. On the other hand, the method presented challenges the networks with a fake captive portal by submitting fake login credentials to test their legitimacy. The study analysed the performance of the *FakeAP Detector*, which had an accuracy detection of 98 % and 99% in open and closed networks, respectively. The prototype had shown the best detection time while detecting in a closed network where an average of 5.78 milliseconds was spent. In the

open network, an application had spent an average of 24.64 milliseconds in detection. This performance is competitive when compared to recent studies, including one conducted by Madani and Vlajic (2021).

On the other hand, the fake captive portal detection was tested and achieved an accuracy of 88%. These approaches focused on a lightweight solution on Android-based devices; hence using the built-in Android resources yields better results and performance without rooting the device. We had used the *WifiManager*, *WebView* and *SQLite* packages to make this possible.

5.2 Recommendations

Android OS is a circular improving OS. Various improvements are made in each of their releases. This makes it challenging for researchers to develop sustainable solutions focusing on Android-based devices. Nevertheless, the studies can contribute to the body of knowledge. This study was conducted to contribute to the body of knowledge. So far, this research has covered the detection of fake APs that mimic legitimate networks. However, fake APs created with random APs or those that do not mimic the structure of legit networks were not covered in this study, and they would demand a different approach

This work did not manipulate details captured in pcap files. The manipulation would result in more information relevant for detecting fake APs by comparing legitimate APs from the fake APs packets. As a remedy, future works may need to redevelop their classes following the *io.pkts* documentation guide. The prototype would result in rich features, including vendor-specific information and round-trip time, strengthening detection effectiveness.

The detection approach using RSSI values may sometimes result in false positives if legitimate and fake APs broadcast with similar RSSI values, especially in open networks. Therefore, this approach has to be used with other parameters collected from the broadcasting APs. On the other hand, an attacker could create an ETA with features similar to legitimate AP, including security information. In this case, the prototype relies on RSSI value only.

Determining the estimated location of legitimate APs using the RSSI value also leaves a promising research gap. Since RSSI is one of the features that an attacker cannot mimic, a strong fake AP detection solution could be developed based on an AP's location. This would be effective if combined with other features from a broadcasting AP. The detection of fake AP that de-authenticates clients and lets them connect to their network was not done due to limited

features collected from the broadcasting APs. A pcap file could easily help since the packets are classified based on their types, including the de-authentication packets. Features currently available in 802.11ax, such as BSS colouring, also leave a promising possibility for detecting fake APs. Furthermore, features like packet duration, address fields, vendor-specific information, and sequence control fields leave promising possibilities for efficient detection measures against fake APs. The study invites research to build a prevention system for network spoofing attacks.

Organisations and companies should constantly be educating their users about the risks associated with wireless networks. On the other hand, open Wi-Fi is highly discouraged, considering its risks. Therefore, there is a need for the organisation to develop effective mechanisms for authenticating wireless APs and users in the network.

REFERENCES

AbiResearch. (2021). *Digital Transformation Doesn't Have To Be Disruptive*. ABI Research. https://www.abiresearch.com/

Aditya, S. K., & Karn, V. K. (2014). *Android sQLite essentials*. Packt Publishing. https://www.google.com

Al-Zubaidie, M., Zhang, Z., & Zhang, J. (2019). Ramhu: A new robust lightweight scheme for mutual users authentication in healthcare applications. *Security and Communication Networks, 2019*, 1-27.

Alsop, T. (2020). *WLAN connected devices worldwide 2016-2021*. https://lb-aps-frontend.statista.com/statistics/802706/world-wlan-connected-device/

Android, E. J. B., & Hagos, T. (2020). *Learn Android Studio 4.* https://www.oreilly.com

Aung, M. A. C., & Thant, K. P. (2017). *Detection and mitigation of wireless link layer attacks. 2017 IEEE 15th international conference on software engineering research, management and applications.* https://www.google.com

Ballai, P. N. (2010). *System and method for detection of a rouge wireless access point in a wireless communication network*. https://www.google.com

Banerjee, A., & Chaudhury, S. (2010). Statistics without tears: Populations and samples. *Industrial Psychiatry Journal, 19*(1), 60.

Bernaschi, M., Ferreri, F., & Valcamonici, L. (2008). Access points vulnerabilities to DoS attacks in 802.11 networks. *Wireless Networks Volume*, 4(2), 159-169.

Bhosale, S., Patil, M., & Patil, P. (2015). Sqlite: Light database system. *International Journal of Computer Science and Mobile Computing, 44*(4), 882-885.

Bitton, R., Finkelshtein, A., Sidi, L., Puzis, R., Rokach, L., & Shabtai, A. (2018). Taxonomy of mobile users' security awareness. *Computer Security*, 73, 266-293.

Booch, G. (2005). *The unified modeling language user guide*. Pearson Education India. https://www.google.com

Booch, G., Rumbaugh, J., & Jacobson, I. (1997). *UML: Unified Modeling Language*. https://www.google.com

Bray, R. A., Gibson, D. I., & Jones, A. (2008). *Keys to the Trematoda, Volume 3*. https://www.degruyter.com

Breitinger, F., & Nickel, C. (2010). *User survey on phone security and usage. BIOSIG 2010: Biometrics and Electronic Signatures. Proceedings of the Special Interest Group on Biometrics and Electronic Signatures.* https://www.google.com

Breitinger, F., Tully-Doyle, R., & Hassenfeldt, C. (2020). A survey on smartphone user's security choices, awareness and education. *Computers & Security*, *88*, 101647.

Bryksa, E. W., & MacMillan, A. T. (2015). Authorizing secured wireless access at hotspot having open wireless network and secure wireless network. https://www.google.com

Chatzisofroniou, G. (2018). *The Known Beacons Attack (34th Chaos Communication Congress)*. https://census-labs.com/news/2018/02/01/known-beacons-attack-34c3/

Chen, Y., Trappe, W., & Martin, R. P. (2007). *Detecting and localizing wireless spoofing attacks. The 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks.* https://www.google.com

Chen, Y., & Yang, J. (2012). *Defending against identity-based attacks in wireless networks. In Handbook on Securing Cyber-Physical Critical Infrastructure (pp. 191-222). Elsevier Inc.* https://www.google.com

Chin, E., Porter Felt, A., Sekar, V., & Wagner, D. (2012). *Measuring user confidence in smartphone security and privacy. Proceedings of the eighth symposium on usable privacy and security*. https://www.google.com

Chirumamilla, M. K., & Ramamurthy, B. (2003). *Agent based intrusion detection and response system for wireless LANs. IEEE International Conference on Communications, 2003. ICC'03.* https://www.google.com

Cisco. (2020). *Cisco Annual Internet Report (2018–2023).* https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Ciurana, M., Barcelo-Arroyo, F., & Izquierdo, F. (2007). *A ranging system with IEEE 802.11 data frames. 2007 IEEE radio and wireless symposium*. https://www.google.com

Craig, C., & Gerber, A. (2015). *Learn Android Studio: Build Android Apps Quickly and Effectively*. Apress. https://www.google.com

Crow, B. P., Widjaja, i., Kim, J. G., & Sakai, P. T. (1997). IEEE 802.11 wireless local area networks. *IEEE Communications magazine, 35*(9), 116-126.

Das, A., & Khan, H. U. (2016). Security behaviors of smartphone users. *Information & Computer Security*, 24(1), 116-134.

Demirbas, M., & Song, Y. (2006). *An RSSI-based scheme for sybil attack detection in wireless sensor networks. 2006 International symposium on a world of wireless, mobile and multimedia networks.* https://www.google.com

Deshpande, S. D., & Davenport, T. J. (2018). *Detection of rogue access point. In: Google Patents.* https://www.google.com

Ebert, C. (1997). Dealing with nonfunctional requirements in large software systems. *Annals of Software Engineering*, *3*(1), 367-395.

Einav, L., Jonathan, L., Igor, P., & Neel, S. (2014). Growth, adoption, and use of mobile E-commerce. *American Economic Review, 104*(5), 489-494.

Esmaeel, H. R. (2015). Apply android studio tools. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(5), 88-93.

Faria, D. B., & Cheriton, D. R. (2006). *Detecting identity-based attacks in wireless networks using signalprints. Proceedings of the 5th ACM workshop on Wireless security*. https://www.google.com

Furnell, S. (2005). Why users cannot use security. *Computers & Security*, *24*(4), 274-279.

Garg, S., & Baliyan, N. (2020). Android Security Assessment: A Review, Taxonomy and Research Gap Study. *Computers & Security*, 100(1),102087.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report*, *8*(4), 597-607.

Gonzales, H., Bauer, K., Lindqvist, J., McCoy, D., & Sicker, D. (2010). P*ractical defenses for evil twin attacks in 802.11. 2010 IEEE Global Telecommunications Conference GLOBECOM 2010.* https://www.google.com

Graham, M. (2018). *New Internet Research Shows 30 000 Spoofing Attacks Per Day*. DELL Technologies. https://www.delltechnologies.com/en-us/perspectives/new-internet-research-shows-30000-spoofing-attacks-per-day/

Guo, R. (2019). Survey on WiFi infrastructure attacks. *International Journal of Wireless and Mobile Computing*, *16*(2), 97-101.

Gupta, V., & Rohil, M. K. (2013). Bit-stuffing in 802. 11 beacon frame: Embedding non-standard custom information. *International Journal of Computer Applications*, 63(2), 6-12.

Harmon, J. (2018). *Systems and methods for detecting potentially illegitimate wireless access points.* https://www.google.com

Iftheker, M. A. E. (2008). *Wireless LAN Security*. https://www.google.com

Imgraben, J., Engelbrecht, A., Kwang, K., & Choo, R. (2014). Always connected, but are smart mobile users getting more security savvy? A survey of smart mobile device users. *Behaviour & Information Technology*, *33*(12), 1347-1360.

Jaisinghani, D., Singh, G., Fulara, H., Maity, M., & Naik, V.S. (2018). *Elixir: Efficient Data Transfer in WiFi-based IoT Nodes. Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. https://www.google.com

Jeske,D., Coventry, L., & Briggs, P. (2014). *Decision justifications for wireless network selection. The 2014 Workshop on Socio-Technical Aspects in Security and Trust*. https://www.google.com

Jiang, Z., Zhao, J., Li, X., Han, J., & Xi, W. (2013). *Rejecting the attack: Source authentication for wi-fi management frames using csi information. 2013 Proceedings IEEE INFOCOM.* https://www.google.com

Jindal, K., Dalal, S., & Sharma, K. K. (2014). *Analyzing spoofing attacks in wireless networks. 2014 Fourth International Conference on Advanced Computing & Communication Technologies.* https://www.google.com

Johnson, J. (2021). *Worldwide digital population as of January 2021*. Retrieved 25/05/2021 from https://www.statista.com/statistics/617136/digital-population-worldwide/

Joyce, G., Andrew, G., David, M., Andrew, M., Sylvester, P., & Madeleke, D. (2018). Mobile phone use in two secondary schools in Tanzania. *Education and Information Technologies volume,* 23(1), 73-92.

Kao, K. F., Chen, W. C., Chang, J. C., & Chu, H. T. (2014). *An accurate fake access point detection method based on deviation of beacon time interval. 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion.* https://www.google.com

Kaspersky. (2021). *Mobile Security: Android vs iOS — which one is safer?* https://www.google.com

KasperskyLab. (2020). *Top 7 Mobile Security Threats in 2020*. https://www.kaspersky.com/resource-center/threats/top-seven-mobile-security-threats-smart-phones-tablets-and-mobile-internet-devices-what-the-future-has-in-store

Kidston, D., & Li, L. (2010). *Management through cross-layer design in mobile tactical networks. 2010 IEEE Network Operations and Management Symposium-NOMS 2010.* https://www.google.com

Kim, H. Y. (2020). *System and method for detecting rogue access point and user device and computer program for the same.* https://www.google.com

Klasnja, P., Consolvo, S., Jung, J., Benjamin, M., Louis, G., Pauline, L., & Wetherall, *D.* (2009). *" When I am on Wi-Fi, I am fearless" privacy concerns & practices in eeryday Wi-Fi use. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* https://www.google.com

Kropeit, T. (2015). *Don't Trust Open Hotspots: Wi-Fi Hacker Detection and Privacy Protection via Smartphone*. https://www.google.com

Kumar, U., & Gambhir, S. (2014). A literature review of security threats to wireless networks. *International Journal of Future Generation Communication and Networking*, *7*(4), 25-34.

Kwak, J., Lee, H., & Lee, K. B. (2012). *A Study on the airtime occupied by beacon frame in 802.11 hotspot environment.* https://www.google.com

Lawless, H. T., & Heymann, H. (2010). Descriptive analysis. In *Sensory evaluation of food*. https://www.google.com

Lazos, L., & Krunz, M. J. I. N. (2011). Selective jamming/dropping insider attacks in wireless mesh networks. https://dl.acm.org/doi/abs/10.1109/MNET.2011.5687950

Leau, Y., Loo, W. K., Tham, W.Y., & Tan, S. F. (2012). *Software Development Life Cycle AGILE vs Traditional Approaches.* https://www.google.com

Lee, W. M. (2012). *Beginning android 4 application Development*. John Wiley & Sons. https://www.google.com

Lim, J. G., & Kim, M. J. (2013). *Terminal and method for access point verification.* https://www.google.com

Lopez-Aguilera, E., Casademont, J., & Cotrina, J. (2004). *IEEE 802.11 g performance in presence of beacon control frames. 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754)*. https://www.google.com

Madani, P., & Vlajic, N. (2021). RSSI-Based MAC-Layer Spoofing Detection: Deep Learning Approach. *Journal of Cybersecurity and Privacy*, *1*(3), 453-469.

Mahadevan, L., & Kaleta, J. P. (2018). Free Wi-Fi: To buy or not to buy. *Journal of Computer Information Systems*, 60(4), 359-369.

Malekzadeh, M., Ghani, A. A., Zulkarnain, Z. A., & Muda, Z. (2007). Security improvement for management frames in IEEE 802.11 wireless networks. *International Journal of Computer Science and Network Security*, *7*(6), 276-284.

Marty, H., & Larry, B. (2001). *Microsoft PowerPoint-XML. ppt-xml. pdf.* https://www.google.com

Matte, C., Achara, J. P., & Cunche, M. (2015). *Device-to-identity linking attack using targeted wi-fi geolocation spoofing. Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. https://www.google.com

Murray, C. (2014). *Smartphone Security Risks: The Extent of User Security Awareness.* https://www.google.com

Nassa, V. K. (2011). Wireless Communications: Past, Present and Future. *Dronacharya Research Journal, 3(2), 50-54.*

Ndibwile, J. D., Kadobayashi, Y., & Fall, D. (2017). *UnPhishMe: phishing attack detection by deceptive login simulation through an android mobile app. 2017 12th Asia Joint Conference on Information Security (AsiaJCIS).* https://www.google.com

Ndibwile, J. D., Luhanga, E. T., Fall, D., Miyamoto, D., Blanc, G., & Kadobayashi, Y. (2019). An Empirical Approach to Phishing Countermeasures Through Smart Glasses and Validation Agents. *Access*, 7, 130758-130771.

Ndibwile, J. D., Luhanga, E.T., Fall, D., Miyamoto, D., & Kadobayashi, Y. (2018). *Smart4Gap: Factors that Influence Smartphone Security Decisions in Developing and Developed Countries. Proceedings of the 2018 10th International Conference on Information Management and Engineering.* https://www.google.com

NortonLifeLock. (2019). *Why hackers love public Wi-Fi*. https://us.norton.com/internetsecurity-wifi-why-hackers-love-public-wifi.html

O'Dea, S. (2021). *Number of smartphone users worldwide from 2016 to 2023*. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

Oh, J. S., Park, M. W., Chung, T. M. (2014). *The multi-level security for the android OS. International Conference on Computational Science and Its Applications.* https://www.google.com

Outpost24. (2020). *Wireless Security: Internet of Evil Things Report*. https://marketing.outpost24.com/mkg/whitepaper/internet-of-evil-things-2020-guide

Owens, L. K. (2002). Introduction *to survey research design. SRL fall 2002 seminar series.* https://www.google.com

Paetsch, F., Eberlein, A., & Maurer, F. (2003). *Requirements engineering and agile software development. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.* https://www.google.com

Park, M., Choi, Y., Eom, J., & Chung, T. (2013). Dangerous Wi-Fi access point: attacks to benign smartphone applications. *Personal and Ubiquitous Computing*, 18(6), 1373-1386.

Pavković, N., & Perkov, L. (2011). *Social Engineering Toolkit: A systematic approach to social engineering. 2011 Proceedings of the 34th International Convention MIPRO.* https://www.google.com

Prasad, R., & Rohokale, V. (2020). Mobile Device Cyber Security. In *Cyber Security: The Lifeline of Information and Communication Technology* (pp. 217-229). https://www.google.com

Premnath, S. N., Ahmadzadeh, S. A., & Das, S. M. (2021). *U.S. Patent No. 10,979,906. Washington, DC: U.S. Patent and Trademark Office.* https://www.google.com

Racine, J. S. (2012). *RStudio: A platform-independent IDE for R and Sweave*. https://www.google.com

Ragunath, P. K., Velmourougan, S., Davachelvan, P., Kayalvizhi, S., & Ravimohan, R. (2010). Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). *International Journal of Computer Science and Network Security*, *10*(1), 112-119.

Rech, J. (2012). *Wireless LANs: 802.11-WLAN-Technologie und praktische Umsetzung im Detail*. Heise Verlag. https://www.google.com

SA, I. (2021). *IEEE 802.11-2020 - IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium*

Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standards Association. https:// standards.ieee.org/standard/802_11-2020.html

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students.* Pearson education. https://www.google.com

Segura, X. J., & El-Moussa, F. (2014). *Method and system for authenticating a point of access.* https://www.google.com

Seigneur, J. M. J. J. O. I. S. (2017). *Wi-Trust: Computational Trust and Reputation Management for Stronger Hotspot 2.0 Security*. https://www.google.com

Seymour, T., & Shaheen, A. (2011). History of wireless communication. *Review of Business Information Systems*, *15*(2), 37-42.

Shedden, P., Ahmad, A., Smith, W., Tscherning, H., & Scheepers, R. (2016). Asset identification in information security risk assessment: A business practice approach. *Communications of the Association for Information Systems,* 39(1), 15.

Sheng, Y., Tan, K., Chen, G., Kotz, D., & Campbell, A. (2008). *Detecting 802.11 MAC layer spoofing using received signal strength. In IEEE INFOCOM 2008-The 27th Conference on Computer Communications.* https://www.google.com

Shrivastava, P., Jamal, M. S., & Kataoka, K. (2020). EvilScout: Detection and mitigation of evil twin attack in SDN enabled WiFi. *Transactions on Network and Service Management*, 17(1), 89-102.

Sidhu, B., Singh, H., & Chhabra, A. (2007). Emerging wireless standards-wifi, zigbee and wimax. *World Academy of Science, Engineering and Technology*, 25(2007), 308-313

Singh, A. S., & Masuku, M. B. (2014). Sampling techniques & determination of sample size in applied statistics research: An overview. *International Journal of Economics, Commerce and Management*, *2*(11), 1-22.

Sombatruang, N., Sasse, M. A., & Baddeley, M. (2016). *Why do people use unsecure public Wi-Fi? An investigation of behaviour and factors driving decisions. In Proceedings of the 6th Workshop on Socio-Technical Aspects in Security and Trust (pp. 61-72).* https://www.google.com

Srinivas, V. B., & Umar, S. (2013). Spoofing attacks in wireless sensor networks. International *Journal of Science, Engineering and Computer Technology*, 3(6), 201.

Stake, R. E. (2010). *Qualitative research: Studying how things work.* https://www.google.com

Suryasa, W., Zambrano, R., Mendoza, J., Moya, M., & Rodríguez, M. (2020). Mobile devices on teaching-learning process for high school level. *International Journal of Psychosocial Rehabilitation,* 20(4), 330-340.

Swanson, C., Urner, R., & Lank, E. (2010, June). *Naïve security in a Wi-Fi world. In IFIP International Conference on Trust Management (pp. 32-47).* Springer, Berlin, Heidelberg. https://www.google.com

Symantec. (2017). *Norton Wi-Fi Risk Report*. https://www.google.com

Tang, K., & Gerla, M. (2000). *MAC layer broadcast support in 802.11 wireless networks. MILCOM 2000 Proceedings.The 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No. 00CH37155)*. https://www.google.com

Tang, Z., Zhao, Y., Yang, L., Qi, S., Fang, D., Chen, X., Gong, X., & Wang, Z. (2017). Exploiting wireless received signal strength indicators to detect evil-twin attacks in smart homes. *Mobile Information Systems,* 2017, 1-15.

Tchakounté, F., Nakoe, M., Yenke, B. O., & Udagepola, K. P. (2019). 'Recognizing illegitimate access points based on static features: A case study in a campus WiFi network. Int. J. Cyber-Secur. *Digit. Forensics,* 8(4), 279-291.

Tse, D., & Viswanath, P. (2005). *Fundamentals of wireless communication*. Cambridge university press. https://www.google.com

Vecchiato, D. A. (2016). *Benchmarking user-defined security configurations of Android devices.* https://www.google.com

Verma, L., Fakharzadeh, M., & Choi, S. (2013). Wifi on steroids: 802.11 ac and 802.11 ad. *Wireless Communications,* 20(6), 30-35.

Walker, J. (2009). *Protected Management Frames*. https://grouper.ieee.org/groups/802/11/Reports/tgw_update.htm

Wang, K., Chen, S., & Pan, A. (2015). Time and position spoofing with open source projects. *Black hat Europe*, 148, 1-8.

Wolfe, C. S. G., Mills, R., Nykl, S., & Simon, P. (2018). *Securing data in power-limited sensor networks using two-channel communications. In Critical Infrastructure Protection XII: 12th IFIP WG 11.10 International Conference, ICCIP 2018, Arlington, VA, USA, March 12-14, 2018.* Springer. https://www.google.com

Wolfson, M., & Felker, D. (2013). *Android developer tools essentials: Android Studio to Zipalign*. " O'Reilly Media, Inc.". https://www.google.com

Wu, W., Gu, X., Dong, K., Shi, X., & Yang, M. (2018). PRAPD: A novel received signal strength–based approach for practical rogue access point detection. *International Journal of Distributed Sensor Networks*, 14(8), 1550147718795838.

Yaddanapudi, S., & Yaddanapudi, L. (2019). How to design a questionnaire. *Indian Journal of Anaesthesia*, *63*(5), 335.

Yamane, T. (1967). *Statistics: An introductory analysis*. https://www.google.com

Yu, J., Kim, E., Kim, H., & Huh, J. H. (2020). Design of a framework to detect device spoofing attacks using network characteristics. *Consumer Electronics Magazine,* 9(2), 34-40.

Zaidi, S. F. A., Shah, M. A., Kamran, M., Javaid, Q., & Zhang, S. (2016). A survey on security for smartphone device. *International Journal of Advanced Computer Science and Applications,* 7(4), 206-219.

APPENDICES

Appendix 1: Sample Logs of Connected Users on Our Fake Network

[admin login] from source 192.168.1.31, Saturday, Mar 06,2021 12:51:13
[DHCP IP: (192.168.1.32)] to MAC address C4:E3:9F:18:AB:CB, Saturday, Mar 06,2021 12:49:48
[DHCP IP: (192.168.1.35)] to MAC address 34:23:87:FE:04:19, Saturday, Mar 06,2021 12:45:10
[DHCP IP: (192.168.1.25)] to MAC address AC:37:43:4A:B2:D6, Saturday, Mar 06,2021 12:42:31
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 12:37:06
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 12:37:04
[Internet connected] IP address: 172.30.102.228, Saturday, Mar 06,2021 12:24:38
[DHCP IP: (192.168.1.40)] to MAC address F8:DA:0C:43:E8:0D, Saturday, Mar 06,2021 12:20:12
[DHCP IP: (192.168.1.35)] to MAC address 34:23:87:FE:04:19, Saturday, Mar 06,2021 12:17:14
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 12:16:39
[DHCP IP: (192.168.1.35)] to MAC address 34:23:87:FE:04:19, Saturday, Mar 06,2021 12:16:05
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 12:13:09
[DHCP IP: (192.168.1.32)] to MAC address C4:E3:9F:18:AB:CB, Saturday, Mar 06,2021 12:13:04
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:2B:9B:59:16, Saturday, Mar 06,2021 12:34:06
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 12:13:04
[DHCP IP: (192.168.1.32)] to MAC address C4:E3:9F:18:AB:CB, Saturday, Mar 06,2021 12:12:48
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 12:10:45
[DHCP IP: (192.168.1.39)] to MAC address 18:E7:77:F5:BD:2F, Saturday, Mar 06,2021 12:04:43
[DHCP IP: (192.168.1.34)] to MAC address 74:C1:7D:8E:DB:7D, Saturday, Mar 06,2021 12:04:42
[DHCP IP: (192.168.1.38)] to MAC address 9A:4E:9C:28:1C:AE, Saturday, Mar 06,2021 12:04:37
[DHCP IP: (192.168.1.37)] to MAC address EE:BE:3E:87:A8:65, Saturday, Mar 06,2021 12:04:12
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:DF:C6:96, Saturday, Mar 06,2021 12:04:04
[DHCP IP: (192.168.1.9)] to MAC address D0:53:49:BD:80:E8, Saturday, Mar 06,2021 12:03:59
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:DF:C6:96, Saturday, Mar 06,2021 12:03:47
[DHCP IP: (192.168.1.15)] to MAC address C4:8E:8F:78:E3:F7, Saturday, Mar 06,2021 12:03:45
[DHCP IP: (192.168.1.36)] to MAC address AC:2D:A9:91:EC:C6, Saturday, Mar 06,2021 12:03:32
[DHCP IP: (192.168.1.15)] to MAC address C4:8E:8F:78:E3:F7, Saturday, Mar 06,2021 12:03:30
[DHCP IP: (192.168.1.35)] to MAC address 34:23:87:FE:04:19, Saturday, Mar 06,2021 12:03:29
[DHCP IP: (192.168.1.34)] to MAC address 74:C1:7D:8E:DB:7D, Saturday, Mar 06,2021 12:03:15
[DHCP IP: (192.168.1.33)] to MAC address 90:56:FC:9D:3B:5D, Saturday, Mar 06,2021 12:03:03
[DHCP IP: (192.168.1.13)] to MAC address 74:E5:0B:E6:2D:C4, Saturday, Mar 06,2021 12:02:33
[DHCP IP: (192.168.1.32)] to MAC address C4:E3:9F:18:AB:CB, Saturday, Mar 06,2021 12:01:29
[DHCP IP: (192.168.1.31)] to MAC address 74:E5:0B:E6:57:D8, Saturday, Mar 06,2021 12:00:45
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:59:57
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:57:31
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:56:52
[DHCP IP: (192.168.1.30)] to MAC address 00:12:36:2B:3A:86, Saturday, Mar 06,2021 11:55:53
[DHCP IP: (192.168.1.29)] to MAC address AC:FD:CE:6B:47:25, Saturday, Mar 06,2021 11:55:39
[DHCP IP: (192.168.1.28)] to MAC address 82:03:6C:AE:DB:E6, Saturday, Mar 06,2021 11:54:45
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:52:07
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:DF:C6:96, Saturday, Mar 06,2021 11:51:38
[DHCP IP: (192.168.1.27)] to MAC address 54:88:0E:D4:6A:45, Saturday, Mar 06,2021 11:50:30
[Internet connected] IP address: 172.30.102.228, Saturday, Mar 06,2021 11:45:58
[DHCP IP: (192.168.1.5)] to MAC address E0:94:67:0C:5A:2E, Saturday, Mar 06,2021 11:45:05
[DHCP IP: (192.168.1.5)] to MAC address E0:94:67:0C:5A:2E, Saturday, Mar 06,2021 11:45:03
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:44:39
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:44:37
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:44:03

[DHCP IP: (192.168.1.15)] to MAC address C4:8E:8F:78:E3:F7, Saturday, Mar 06,2021 11:43:22
[DHCP IP: (192.168.1.26)] to MAC address 74:E5:0B:DB:22:48, Saturday, Mar 06,2021 11:42:20
[DHCP IP: (192.168.1.25)] to MAC address AC:37:43:4A:B2:D6, Saturday, Mar 06,2021 11:42:20
[DHCP IP: (192.168.1.4)] to MAC address 2A:B8:7F:DC:E4:28, Saturday, Mar 06,2021 11:42:15
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:41:51
[DHCP IP: (192.168.1.24)] to MAC address F4:09:D8:F3:96:F1, Saturday, Mar 06,2021 11:40:56o
[DHCP IP: (192.168.1.10)] to MAC address 74:29:AF:E8:C1:99, Saturday, Mar 06,2021 11:40:15
[DHCP IP: (192.168.1.16)] to MAC address E8:B1:FC:61:E1:6C, Saturday, Mar 06,2021 11:40:09
[DHCP IP: (192.168.1.8)] to MAC address E0:06:E6:A2:E7:BC, Saturday, Mar 06,2021 11:39:58
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:39:47
[DHCP IP: (192.168.1.23)] to MAC address 10:A5:D0:D1:DD:1F, Saturday, Mar 06,2021 11:39:07
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:DF:C6:96, Saturday, Mar 06,2021 11:38:39
[DHCP IP: (192.168.1.22)] to MAC address 74:E5:0B:D5:C4:AA, Saturday, Mar 06,2021 11:38:39
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:38:34
[DHCP IP: (192.168.1.21)] to MAC address 42:AA:68:E9:EA:7D, Saturday, Mar 06,2021 11:38:31
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:EC:D5:66, Saturday, Mar 06,2021 11:38:29
[DHCP IP: (192.168.1.20)] to MAC address 74:E5:0B:CA:C6:57, Saturday, Mar 06,2021 11:38:24
[DHCP IP: (192.168.1.19)] to MAC address 74:E5:0B:E6:5B:78, Saturday, Mar 06,2021 11:38:08
[DHCP IP: (192.168.1.18)] to MAC address 74:E5:0B:9B:59:16, Saturday, Mar 06,2021 11:38:07
[DHCP IP: (192.168.1.17)] to MAC address 18:3D:A2:7F:0B:44, Saturday, Mar 06,2021 11:38:06
[DHCP IP: (192.168.1.16)] to MAC address E8:B1:FC:61:E1:6C, Saturday, Mar 06,2021 11:37:58
[DHCP IP: (192.168.1.15)] to MAC address C4:8E:8F:78:E3:F7, Saturday, Mar 06,2021 11:37:46
[DHCP IP: (192.168.1.14)] to MAC address 74:E5:0B:F1:7C:8C, Saturday, Mar 06,2021 11:37:40
[DHCP IP: (192.168.1.13)] to MAC address 74:E5:0B:E6:2D:C4, Saturday, Mar 06,2021 11:37:31
[DHCP IP: (192.168.1.12)] to MAC address BC:91:B5:BE:73:6F, Saturday, Mar 06,2021 11:37:20
[DHCP IP: (192.168.1.11)] to MAC address 74:C1:7D:92:07:D7, Saturday, Mar 06,2021 11:37:14
[DHCP IP: (192.168.1.9)] to MAC address D0:53:49:BD:80:E8, Saturday, Mar 06,2021 11:37:02
[DHCP IP: (192.168.1.10)] to MAC address 74:29:AF:E8:C1:99, Saturday, Mar 06,2021 11:37:02
[DHCP IP: (192.168.1.6)] to MAC address E0:1F:88:FF:56:61, Saturday, Mar 06,2021 11:37:00
[DHCP IP: (192.168.1.7)] to MAC address 80:19:67:7E:C4:A6, Saturday, Mar 06,2021 11:36:56
[DHCP IP: (192.168.1.5)] to MAC address E0:94:67:0C:5A:2E, Saturday, Mar 06,2021 11:36:47
[DHCP IP: (192.168.1.4)] to MAC address 2A:B8:7F:DC:E4:28, Saturday, Mar 06,2021 11:36:45
[DHCP IP: (192.168.1.3)] to MAC address 46:2D:85:10:BA:B8, Saturday, Mar 06,2021 11:36:31
[DHCP IP: (192.168.1.2)] to MAC address 98:E0:D9:8A:69:47, Saturday, Mar 06,2021 11:35:41
[Internet connected] IP address: 172.30.102.228, Saturday, Mar 06,2021 11:34:56
[DHCP IP: (192.168.1.45)] to MAC address 98:E0:D9:9A:52:76, Saturday, Mar 06,2021 11:34:38
[DHCP IP: (192.168.1.2)] to MAC address 50:7A:55:EB:8B:FB, Wednesday, Dec 16,2020 09:55:56
[Internet disconnected] Wednesday, Dec 16,2020 09:54:05
[DHCP IP: (192.168.1.2)] to MAC address 50:7A:55:EB:8B:FB, Wednesday, Dec 16,2020 09:54:05
[Initialized, firmware version: V1.4.1.68_1.3.28] Wednesday, Dec 16,2020 09:54:02
[DHCP IP: (192.168.1.3)] to MAC address 50:7A:55:EB:8B:FB, Wednesday, Dec 16,2020 10:08:13
[DHCP IP: (192.168.1.6)] to MAC address 88:78:73:AA:F5:37, Wednesday, Dec 16,2020 10:02:14
[DHCP IP: (192.168.1.2)] to MAC address 82:03:6C:AE:DB:E6, Wednesday, Dec 16,2020 09:58:23
[DHCP IP: (192.168.1.6)] to MAC address 88:78:73:AA:F5:37, Wednesday, Dec 16,2020 09:54:16
[Internet disconnected] Wednesday, Dec 16,2020 09:54:16
[DHCP IP: (192.168.1.2)] to MAC address 82:03:6C:AE:DB:E6, Wednesday, Dec 16,2020 09:54:15
[Initialized, firmware version: V1.4.1.68_1.3.28] Wednesday, Dec 16,2020 09:54:11
[Time synchronized with NTP server] Wednesday, Mar 03,2021 18:08:50
[DHCP IP: (192.168.1.4)] to MAC address 72:1D:38:BB:1B:1C, Wednesday, Mar 03,2021 18:05:02
[Time synchronized with NTP server] Wednesday, Mar 03,2021 18:03:48
[DHCP IP: (192.168.1.4)] to MAC address 72:1D:38:BB:1B:1C, Wednesday, Mar 03,2021 18:03:46
[Internet connected] IP address: 172.30.90.221, Wednesday, Mar 03,2021 18:03:44
[Time synchronized with NTP server] Wednesday, Mar 03,2021 17:59:30

[DHCP IP: (192.168.1.3)] to MAC address 70:66:55:05:97:3B, Wednesday, Mar 03,2021 17:58:11
[DHCP IP: (192.168.1.2)] to MAC address 28:16:7F:A5:85:6E, Wednesday, Mar 03,2021 17:57:51
[DHCP IP: (192.168.1.6)] to MAC address 88:78:73:AA:F5:37, Wednesday, Mar 03,2021 17:54:43
[Time synchronized with NTP server] Wednesday, Mar 03,2021 17:54:32
[Internet connected] IP address: 172.30.90.221, Wednesday, Dec 16,2020 09:54:56
[Internet disconnected] Wednesday, Dec 16,2020 09:54:17
[Initialized, firmware version: V1.4.1.68_1.3.28] Wednesday, Dec 16,2020 09:54:11

Appendix 2: Sampling Formula

Yamane (1967) provides a simplified formula for proportions. It is one of many sample size calculation formulas.

$$n = \frac{N}{1 + N(e)^2}$$

Where n is the sample size,

N is the population size, and

e is the level of precision.

Appendix 3: Questionnaire

User Experience on Wireless Networks

Dear respondent, thank you for taking part in this, you are of great help.

I am Lunodzo Mwinuka, a Master's student at the Nelson Mandela African Institution of Science and Technology (NM-AIST). We are researching to study users' experience and practices while accessing the Internet through Wireless access points (Hotspots).

Our target respondents are everyone who works or study at either Nelson Mandela African Institution of Science Technology (NM-AIST) or Mzumbe University (MU).

Responses collected on this survey shall strictly be used for this study.

We kindly ask you to spare some minutes of your time to answer few questions. By proceeding to respond to the coming questions, you consent to practice in our study.

Don't hesitate to contact +255765268371 when you have any challenges related to this questionnaire or study.

SECTION I: Respondents Profile

1. What is your Gender?
   (*Mark only one*)

   ☐Female

   ☐Male

2. What is your age?
   (Mark only one)
   ☐18 – 24
   ☐25 – 34
   ☐35 – 44
   ☐45 and Above

3. What is your Education Level?
   (*Mark only one*)
   ☐Certificate
   ☐Diploma
   ☐Advanced Diploma
   ☐Bachelor Degree
   ☐Maste's Degree
   ☐PhD

4. Which Institution do you belong to?
   (*Mark only one*)
   ☐Nelson Mandela African Institution of Science and Technology (NM-AIST)

☐Mzumbe University (MU)

5. What is your role at your current Institution?
   (*Mark only one*)
   ☒ I am currently studying (*Skip to question 6*)
   ☐ I am currently working (*Skip to question 8*)
   ☐ I am studying and working (*Skip to question 8*)

SECTION II: Questions for students

6. Year of study
   (*Mark only one*)
   ☐Year One
   ☐Year Two
   ☐Year Three
   ☐Year Four
   ☐Year Five

7. Field of study
   ☐Computer related programmes
   ☐Other programmes

SECTION III: Questions for workers

8. Working experience
   ☐Less than 3 years
   ☐4 – 6 Years
   ☐Above 6 years
9. Field of work
   ☐Computer related
   ☐Others: _____.
10. Level
    (*Mark only one*)
    ☐Junior
    ☐Mid-career
    ☐Senior

SECTION IV: Users' behaviour while connecting to Wireless networks.

11. What factors do you consider while connecting to Wireless Hotspots
    (*Tick all that apply*)
    ☐Signal strength
    ☐Free Wi-Fi
    ☐SSID (Wireless name)
    ☐Speed
    ☐Any available
    ☐Not sure
    ☐Other: _____.

12. While I am in-campus, I use Wi-Fi more than cellular data.
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

13. I prefer Internet access on Wi-Fi than on direct Internet cables
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

14. I prefer secured Wi-Fi (One with password) over open Wi-Fi (One without password)
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

15. What factors affect your choices between Internet access on Wi-Fi, Cable, and Cellular Data?
    (*Choose all that apply*)
    - ☐Cost
    - ☐Performance (Speed)
    - ☐Ease of access
    - ☐Other: _____.

SECTION V: Wireless safety

16. I can share personal information including authentication details, passwords and security PIN via email, social media, and input forms etc. while connected to Wi-Fi.
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

17. I am likely to do money transactions/Banking operations while connected to Wi-Fi
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

18. I usually connect to Wi-Fi despite being warned of the possible dangers they may cause to my devices
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

19. I allow sharing option while connected to Wireless networks
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

20. I am not concerned with which Wi-Fi hotspot I am connected to.
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

21. I usually use different Wi-Fi hotspot when accessing Internet on Mobile Phone and on Computer.
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

SECTION VI: Wireless networks and Organisation Policies

22. My Institution shares clear policies guiding our association to Wireless Networks
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

23. My Institution recommends best practices while using Wireless services
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral (Go to SECTION VIII)
    - ☐Disagree (Go to SECTION VIII)
    - ☐Strongly disagree (Go to SECTION VIII)

SECTION VII: If Institution recommends best practices on Wireless connections.

24. I follow best practices recommended by my Institution
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

SECTION VIII: General Wireless Knowledge

25. Do you feel safe on Wireless Networks?
    - ☐Not safe at all
    - ☐Not safe
    - ☐May be
    - ☐Safe
    - ☐Very safe

26. Where did/do you learn the most about Cyber security?
    (*Choose all that apply*)
    - ☐My Organisation
    - ☐Articles
    - ☐Class Lectures/Formal education
    - ☐Regulatory
    - ☐Personal efforts/experience
    - ☐Online tutorials
    - ☐Other: _____.

27. I am concerned if my information were accessed on Wireless networks
    - ☐Strongly agree
    - ☐Agree
    - ☐Neutral
    - ☐Disagree
    - ☐Strongly disagree

28. I have witnessed network fraud while connected on wireless network
    - ☐Yes
    - ☐No
    - ☐May be

29. My device connects automatically to Wi-Fi hotspots
    - ☐True
    - ☐False
    - ☐Not sure

30. I have set my device to remember Wi-Fi hotspots which I connect

☐Yes
☐No
☐May be

31. I usually change my device's Wireless settings
☐Strongly agree
☐Agree
☐Neutral
☐Disagree
☐Strongly disagree

SECTION IX: Wireless settings

32. Which settings do you usually change while setting your wireless adapter? (Choose all that apply)
☐Sharing options
☐Secured and unsecured hotspot
☐Virtual Private Networks (VPN)
☐Download settings
☐Auto connection/Manual connection
☐Wi-Fi Direct
☐None of the above
☐Other: _____.

Appendix 4: Sample Codes

(i) Sample Codes: Detecting Fake Captive Portal

```
package net.kismetwireless.android.pcapcapture;
    import android.app.Activity;
    import android.os.Build;
    import android.os.Bundle;
    import android.webkit.WebResourceResponse;
    import android.webkit.WebView;
    import android.webkit.WebViewClient;
    import android.widget.Toast;
    import androidx.annotation.Nullable;
    import androidx.annotation.RequiresApi;

    public class CaptivePortal extends Activity {
        String url = "http://fakeap.lunodzo.com";

        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
        @Override
        protected void onCreate(@Nullable Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            WebView webView = new WebView(this);
            setContentView(webView);

            webView.setWebViewClient(new WebViewClient(){
                @Override
                public void onPageFinished(WebView view, String url) {
                    super.onPageFinished(view, url);
                }
            public void onReceivedError(WebView view, int errorCode, String desc, String failU
            webView.loadUrl("https://sis.nm-aist.ac.tz/");
                }
            });
            webView.loadUrl(url);
            webView.getSettings().setJavaScriptEnabled(true);
           String username = "Abcdefghhrj";
           String password = "BDHHSD78678";
           webView.loadUrl("javascript:document.getElementsByName('username').value = "+userna
           webView.loadUrl("javascript:document.getElementsByName('password').value = "+passwo
           webView.loadUrl("javascript:document.forms['submit'].submit()");
           //Read HTTP response
           WebResourceResponse errorResponse = null;
           int statusCode = errorResponse.getStatusCode();
           if(statusCode != 401){
           Toast.makeText(getApplicationContext(), "Fake Captive Portal"+statusCode,
           Toast.LENGTH_SHORT).show();
           }else{
           Toast.makeText(getApplicationContext(), "Legit Captive Portal"+statusCode,
           Toast.LENGTH_SHORT).show();
            }
        }
    }
```

(ii) Sample Codes: Wi-Fi Scanning and Storing into DB (ten rounds)

```
BroadcastReceiver wifiReceiver = new BroadcastReceiver() {
   @Override
   public void onReceive(Context context, Intent intent) {
      results = wifiManager.getScanResults();
      int size = results.size();
      DatabaseHandler databaseHandler = new DatabaseHandler(context);
      //Check if there is any AP
      if (size > 0) {
        //Loop according to the number of APs OR for each available AP, post the details into
        for (int i = 0; i < size; i++) {
            ScanResult scanResult = wifiManager.getScanResults().get(i);
            //AP parameters
            String ssid = scanResult.SSID;
            String bssid = scanResult.BSSID;
            int rssi = scanResult.level;
            String capabilities = scanResult.capabilities;
            //long another = scanResult.timestamp;
            java.util.Date date = new java.util.Date();
            java.sql.Date currentTime = new java.sql.Date(date.getTime());
            SimpleDateFormat dft = new SimpleDateFormat("HH:mm:ss.SSS");
            String time = dft.format(currentTime);

            //Write a statement to post these into Database
            boolean insert = databaseHandler.addAccessPoints(ssid, bssid, rssi, capabilities,
            count, false, time);

            if(insert == false){
            Toast.makeText(getApplicationContext(), "Data insert Failed",
            Toast.LENGTH_SHORT).show();
             }
        }
        count++;
        if(count <= 10){
           //Another scan
           wifiManager.startScan();
        }else{
           unregisterReceiver(this);
        }
      }else{
          unregisterReceiver(this);
          Toast.makeText(getApplicationContext(), "No Access Points found..",
          Toast.LENGTH_SHORT).show();
       }

       for (ScanResult scanResult: results){
          arrayList.add(scanResult.SSID +" * "+ scanResult.capabilities + " * "+
          scanResult.BSSID+ " * "+ scanResult.level);

          arrayAdapter.notifyDataSetChanged();
       }
     }

   };
```

(iii) Sample Codes: SQLite Database Handler

```
public class DatabaseHandler extends SQLiteOpenHelper{
            private static final String TAG = "DatabaseHelper";
            private static final int DATABASE_VERSION = 1;
            private static final String DATABASE_NAME = "fakeapdetect";
            private static final String TABLE_AP = "accesspoints";
            private static final String ID = "id";
            private static final String SSID = "ssid";
            private static final String BSSID = "bssid";
            private static final String SIGNAL = "level";
            private static final String CAPABILITIES = "capabilities";
            private static final String SCAN_ROUND = "round";
            private static final String COMMENT = "comment";
            private static final String TIME = "time";
            Time time = new Time();

            public DatabaseHandler(Context context) {
                super(context, DATABASE_NAME, null, DATABASE_VERSION);
            }

            @Override
            public void onCreate(SQLiteDatabase db) {
                String CREATE_AP_TABLE = "CREATE TABLE " + TABLE_AP + "("
                        + ID + " INTEGER PRIMARY KEY,"
                        + SSID + " TEXT,"
                        + BSSID + " TEXT,"
                        + SIGNAL + " INTEGER,"
                        + CAPABILITIES + " TEXT,"
                        + SCAN_ROUND + " INTEGER,"
                        + COMMENT + " BOOLEAN,"
                        + TIME + " TEXT"
                        + ")";
                db.execSQL(CREATE_AP_TABLE);
            }

            @Override
            public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
                db.execSQL("DROP TABLE IF EXISTS " + TABLE_AP);
                onCreate(db);
            }

            //Add AP to Database
            public boolean addAccessPoints(String ssid, String bssid, int signal, String
               capabilities, int round, boolean comment, String time){

                SQLiteDatabase db = this.getWritableDatabase();
                ContentValues values = new ContentValues();
                values.put(SSID, ssid);
                values.put(BSSID, bssid);
                values.put(SIGNAL, signal);
                values.put(CAPABILITIES, capabilities);
                values.put(SCAN_ROUND, round);
                values.put(COMMENT, comment);
                values.put(TIME, String.valueOf(time));

                Log.d(TAG, "addAccessPoints: Adding " +ssid+ " and "+bssid+" to "+TABLE_AP);
                long result = db.insert(TABLE_AP, null, values);

                if (result == -1){
                    return false;
                }else{
                    return true;
```

Appendix 5: Wireless most common 802.11 filters v1.1

Category	Description	Filters
Addresses	Specific client by MAC address	wlan.addr == *MAC*
	Transmitter address (TA)	wlan.ta == *MAC*
	Receiver address (RA)	wlan.ra == *MAC*
	Source address (SA)	wlan.sa == *MAC*
	Destination address (DA)	wlan.da == *MAC*
Wi-Fi networks	Filter by BSSID (AP)	wlan.bssid == *AP_MAC*
	Filter by SSID	wlan_mgt.ssid == *"SSID"*
802.11 Management Frames	All management frames	wlan.fc.type == 0
	Association request (subtype 0x0)	wlan.fc.type_subtype == 0
	Association response (subtype 0x1)	wlan.fc.type_subtype == 1
	Re-association request (subtype 0x2)	wlan.fc.type_subtype == 2
	Re-association response (subtype 0x3)	wlan.fc.type_subtype == 3
	Probe request (subtype 0x4)	wlan.fc.type_subtype == 4
	Probe response (subtype 0x5)	wlan.fc.type_subtype == 5
	Beacons (subtype 0x8)	wlan.fc.type_subtype == 8
	ATIM (subtype 0x9)	wlan.fc.type_subtype == 9
	Disassociation (subtype 0xa)	wlan.fc.type_subtype == 10
	Authentication (subtype 0xc)	wlan.fc.type_subtype == 11
	De authentication (subtype 0xd)	wlan.fc.type_subtype == 12
	Action (subtype 0xd)	wlan.fc.type_subtype == 13
Radio Tap Header Information	Specific channel	radiotap.channel.freq == *F*
	Specific data rate	radiotap.datarate == *Mbps*
	Signal strength	radiotap.dbm_antsignal == *dBm*

RESEARCH OUTPUTS

(i) Publication

Mwinuka, L. J., Agghey, A. Z., Kaijage, S. F., & Ndibwile, J. D. (2022). FakeAP Detector: An Android-Based Client-Side Application for Detecting Wi-Fi Hotspot Spoofing. *Access*, 10, 13611-13623.

(ii) Poster Presentation