Strathmore University
SU+ @ Strathmore
University Library

Electronic Theses and Dissertations

2019

Rogue access point detection framework on a multivendor access point WLAN

Fredrick K. Barasa
Faculty of Information Technology (FIT)
Strathmore University

Follow this and additional works at https://su-plus.strathmore.edu/handle/11071/6700

Recommended Citation

Barasa, K. F. (2019). *Rogue access point detection framework on a multivendor access point WLAN* (Thesis, Strathmore University). Retrieved from http://su-plus.strathmore.edu/handle/11071/6700

This Thesis - Open Access is brought to you for free and open access by DSpace @Strathmore University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of DSpace @Strathmore University. For more information, please contact librarian@strathmore.edu

Rogue Access Point Detection Framework on a Multivendor Access Point WLAN

Barasa, Kunjira Fredrick

Submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology at Strathmore University

Faculty of Information Technology
Strathmore University
Nairobi, Kenya

June, 2019

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

DECLARATION

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University

Fredrick Kunjira Barasa

………………………..

11th June 2019

Approval

The thesis of Fredrick Kunjira Barasa was reviewed and approved by the following:

Dr. Vitalis Gavole Ozianyi
Senior Lecturer, Faculty of Information Technology,
Strathmore University

VT OMNES VNVM SINT

Dr. Joseph Orero
Dean, Faculty of Information Technology
Strathmore University

Prof. Ruth Kiraka
Dean, School of Graduate Studies
Strathmore University

ABSTRACT

Wireless internet access has become common throughout the world. IEEE 802.11 Wireless fidelity (Wi-Fi) is now a common internet access standard almost becoming a requirement in homes, offices, universities and public places due to developments in Bring-Your-Own-Device (BYOD), mobile telephony and telecommuting. With the proliferation of Wi-Fi comes a number of information security challenges that have to be addressed. One of the major security threats that comes with Wi-Fi is the presence of rogue access points (APs) on the network. Unsuspecting employees in a company or attackers can introduce rogue APs to a secure wired network. The problem is amplified if the wireless local area network (WLAN) consist of multi-vendor APs. Malicious people can leverage on rogue APs to perform passive or active attacks on a computer network. Therefore, there is need for network administrators to accurately, with less effort, detect and control presence of rogue APs on multivendor WLANs.

In this thesis, a solution that can accurately support detection of rogues APs on a multi-vendor AP WLAN without extra hardware or modification of AP firmware is presented. In the solution, information from beacon frames is compared to a set of approved parameters. Intervention of a network administrator is included to prevent MAC address spoofing. A structured methodology was adopted in developing the model on a Windows operating system. Python programming language was used in coding the system with Scapy and Tkinter as the main modules. SQLite database was used to store required data. The system was tested on a setup WLAN that composed of three different access points in a University lab. It was able to capture beacon frames sent by the access points and extracted MAC address, SSID and capability information as the key parameters used in identifying and classifying the access points. The system uses the captured information to automatically compare it against an existing database of authorized parameters. It is then able to classify an access point as either rogue or authorized. The system issued alerts that described the detected APs to a network administrator. The rest of this document gives details of scholarly works that are pertinent to the study, the research methodology used, implementation and testing of the model followed by discussions of findings and the conclusions and recommendations made by the researcher.

TABLE OF CONTENTS

VT OMNES VNVM SINT

VT OMNES VNVM SINT

LIST OF FIGURES

VT OMNES VNVM SINT

VT OMNES VNVM SINT

LIST OF TABLES

VT OMNES VNVM SINT

LIST OF ABBREVIATIONS

ACK	Acknowledgement
AP	Access Point
BSSID	Basic Service Set Identifier
BYOD	Bring Your Own Device
CPU	Central Processing Unit
DAIR	Dense Array of Inexpensive Radios
DNS	Domain Naming System
ER	Entity Relationship
GB	Gigabytes
GHz	Gigahertz
IAT	Inter-Arrival Time
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IOT	Internet of Things
LAN	Local Area Network
MAC	Media Access Control
NIC	Network Interface Card
NTP	Network Time Protocol
PC	Personal Computer
PHY	Physical Layer
RAM	Random Access Memory
SSID	Service Set Identifier
TCP	Transmission Control Protocol
UML	Unified Modelling Language
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WEP	Wired Equivalent Privacy
WIPS	Wireless Intrusion Prevention System
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

ACKNOWLEDGMENTS

The author is highly grateful to the supervisor for the guidance and support received during the research. The researcher also acknowledges the Dean of the Faculty and the panel for their insights and contributions during the thesis seminars. Lastly, the author appreciates his family for allowing him ample time and availing some resources needed during the research.

VT OMNES VNVM SINT

CHAPTER ONE

INTRODUCTION

1.1 Introduction

This chapter details the background of the study, the problem under study, the main aim and specific objectives of the study. Research questions are also stated; justification, scope and limitations of the study are given.

1.2 Background to the Study

Wireless internet access has become common globally. Many organizations use Wireless local area networks (WLANs) to provide access to internet and intranet for employees and clients. This facilitates flexibility and mobility. With proliferation of BYOD concept, wireless access has also grown. Wireless fidelity (Wi-Fi) is used to describe a group of IEEE 801.11 technological specifications that enable devices to establish and connect to a wireless local area network transmitted using 2.4 or 5 GHz spectrum (Tektronix, 2015). Wi-Fi came to prominence at the turn of the millennium when 801.11 standards were refined and personal computer manufacturers began to market Wi-Fi equipped computers. With the distribution of affordable wireless routers, Wi-Fi has quickly become ubiquitous in private homes. WLANs have also taken hold in offices and universities where local area Ethernet networks already exist, while some businesses such as restaurants and cafés install Wi-Fi hotspots to attract more customers (Economist, 2004). Wi-Fi hotspots can now be expected where there are intersecting flows of commerce and people (Lambert, McQuire, & Papastergiadis, 2013).

Sustainable Development Goal (SDG) 8 has six targets that seek to develop global partnership for development. These includes availing benefits of new technologies, especially ICT in collaboration with the private sector. Internet penetration is one of the enablers of this target yet it is still a big concern in developing countries (Philbeck, 2017). Internet penetration as at December 2017 is 35.2% in Africa, rest of the world is 58.4% and the world average is 54.4% (Internet World Stats, 2017). The cheapest way as at now to supply internet to rural remote areas of Kenya and Africa as such is through wireless networks. WLANs and Wireless Wide Area Networks (WWANs) are therefore expected to grow tremendously in the near future.

The increase of WLANs usage comes with growth in their challenges. One of the major concerns with wireless internet access is the inherent security risk. WLANs are known easy victims of both passive and active attacks facilitated through different man-in-the-middle attack techniques. This is mainly because information transmitted through wireless networks is broadcasted to everyone within the vicinity of the network. This research considers the most used method of attacking WLANs and in extension secure Ethernet LANs; rogue APs. Many network administrators agree that presence of rogue APs on their WLANs pose the biggest security threat to the network including the core Ethernet networks (Vanjale & Mane, 2014). A rogue access point (rogue AP) is any active wireless access point that is has not been authorized by the WLAN personnel. Such an access point can be installed by either plugging an access point directly into an Ethernet port or by using two wireless interfaces (Han H. , Sheng, Tan, Li, & Lu, 2009). In the latter, the first wireless interface is connected to an authorized access point, and the other acts as an access point to allow unsuspecting client devices to connect to it. Rogue APs connected to Ethernet are rare and easier to detect and control compared to those connected to legitimate APs.

1.2.1 WLAN Rogue Access Point Attack Characteristics

Rogue APs can be utilized by malicious individuals to perform two primary kinds of attacks on a network; passive and active attacks.

Passive attack is an information gathering attack where the attacker only listens to the traffic traversing the network but does not perform any other action like sniffing or modifying packets. It is the first step before propelling an actual attack itself. Amid passive attacks, the victim has no real way to recognize the attacker's action because the attacker is not acting (Amiel, Villegas, Feix, & Marcel, 2007). The attacker can therefore remain undetected for a long time on the network. One of the passive attacks is cracking Wired Equivalent Privacy (WEP) encryption. In breaking a WEP encryption the attacker needs to sniff an enormous number of packets. The attacker then captures several wireless frames and then attempts to crack WEP offline. The attacker does not need to communicate with the victim at all. Second technique of passive attacks is simply sitting between two communicating parties and sniffing their conversations using a special too such as Wireshark. This requires the attacker to know the encryption key of the wireless connection. The attacker has a clear view of the messages if the communicating parties are using clear text protocols such as HTTP which do not use encryption natively. Cracking WPA or WPA2 encryption is another form of passive attack. Here, the attacker has to capture the Extensible Authentication Protocol (EAP) 4-way handshake that is happening between a wireless client and access point. The attacker then uses an offline dictionary or brute-force attack on the sniffed packets. The attacker may not need to communicate with the victim at all but in some cases where the victim is already authenticated to the access point, the attacker may have to inject wireless deauthentication frames to force the victim to reauthenticate hence performing a new 4-way handshake that the attacker captures. The constant flapping of the wireless connection from authentication to de-authentication jeopardizes the Wi-Fi experience to users.

Active attacks are common because the attacker communicates with the victims hence able to be detected on the network. The attacker actively participates in a communication by capturing and modifying packets between any communicating parties. This affects WLAN operation. An attacker leveraging a rogue AP can cause a denial of service (DoS) attack by sending deauthentication frame to clients thus forcing them to disconnect from genuine access points. This leads to periodic authentication and to de-authenticate and making the overall wireless experience very bad. The attacker can also use a jamming device to interfere with wireless signals thus degrading the Wi-Fi. Man-in-Middle (MiM) attack happens when an attacker uses two network interface cards whereby one card connects to a legitimate access point as a client while the second card advertises a fake SSID to lure unsuspecting clients to connect. The attacker creates a rogue access point. Evil twin attack demonstrates this type of active attack clearly. The access point may be having the same SSID and BSSID as the authorized ones. The evil twin is set to pass traffic through to the authorized AP while sniffing the client's communication or it can cause a DoS upon capturing confidential information such as usernames, passwords and banking details.

Multivendor AP WLANs are becoming common as many network administrators want the flexibility of implementing APs of their choice to cut costs, provide wider coverage while improving performance and circumventing the restrictions imposed by specific vendors. This makes rogue AP detection even more complex as vendor specific WLAN controllers will easily report other vendor APs as rogue. For instance, if a WLAN runs on RUCKUS, CISCO, Juniper and Aruba APs will be reported as rogues on a Ruckus controller unless explicitly specified by the network administrator as knowns or neighbors.

This research proposed a model that can easily and accurately distinguish rogue APs from legitimate APs on both single and multi-vendor WLAN environment by analyzing beacon frames to get properties of APs and comparing the properties against defined parameters. The system borrows a little from some properties of network packet sniffers.

1.3 Problem Statement

Rogue APs create dangerous loopholes for attacks on both WLANs and secure local area networks. They facilitate various man-in-the-middle attacks that can greatly damage network usage through denial of service attacks and theft of data (Gopinath & Hemant, 2009). The area of rogue APs in Wi-Fi security has been actively researched in the recent past signifying the importance of the matter. Multiple solutions have been proposed and developed to detect and eliminate rogue APs on wireless networks. Most of these frameworks and solutions support a single vendor access point WLANs. Some require costly extra hardware resources or modification of existing firmware which is costly to implement in many organizations. Network administrators find some solutions too technically complex to configure and deploy especially those based on Linux systems. They require use of advanced configuration commands and techniques that are sometimes hard for an ordinary network operator to comprehend. This study sought to develop a solution that can easily and accurately perform detection of rogues APs on a multi-vendor AP WLAN without forcing network operators to use extra hardware such as USB cards or modify their access point firmware to function. The result is a system that reads and interprets beacon frames from connected APs to categorize them as either genuine, rogue or neighbor based on preset conditions.

1.4 Aim

VT OMNES VNVM SINT

The main goal of this research was to develop a suitable rogue AP detection system for multi-vendor WLAN environments that can be adopted by network operators to notice rogue APs on the network.

1.5 Specific Objectives

In order to achieve the above stated goal, the researcher worked to achieve the following objectives:

i. To evaluate current rogue AP detection techniques.
ii. To establish important parameters necessary for classifying access points on a WLAN.
iii. To create a classification criterion for access points on a multivendor WLAN based on the distinguishing parameters identified in objective (ii).

iv. To develop a model for detecting rogue APs on a multivendor WLAN.

1.6 Research Questions

In order to achieve the above stated objectives, this research answered the following questions:

i. What are the existing and proposed models for detecting rogue APs on a WLAN?
ii. What are the important parameters necessary to identify access points on a WLAN?
iii. How do we classify access points on a multivendor WLAN based on the distinguishing parameters?
iv. How will a cross-platform system for detecting rogue APs on a multivendor WLAN function?

1.7 Justification

Presence of rogue APs on a WLAN is a big security threat that keeps network administrators actively looking for proper detection solutions (Vanjale & Mane, 2014). Rogue APs are commonly used to facilitate active and passive attacks on networks. The problem is amplified if the network consists of APs from different vendors. Adoption of BYOD practice will also increase the risk of attacks occurring on an organization's network due to fake or rogue access points. Current rogue AP detection systems such as Wireless Intrusion Prevention Systems (WIPS) and vendor-specific WLAN controllers cost more money to purchase and maintain. They also work best with the specific vendor access points. These solutions do not function properly in multivendor WLAN environments. They can wrongly label APs from other vendors as rogues. There is need for an effective and less costly rogue AP detection system that can accurately identify rogue APs on a multivendor WLAN. Network administrators desire to deploy access points from more than one vendor on their network to reduce costs and utilize an access point capacity fully. This is through buying affordable access points and deploying an access point that is best suited to serve a given group of clients in a specific area of the network. Primary benefactors of this system are network administrators or operators because they need to accurately detect presence of rogue APs on their network before devising a good elimination technique on a multivendor access point WLAN. The system developed from this research is easy to deploy and configure to function on both single and multivendor WLAN AP environments. The system uses clear installation instructions that even non-technical individuals can understand. It uses graphical user interfaces that directs the user on actions to

UT OMNES VNVM SINT

take unlike command-line interfaces that require mastery of commands. Network administrators or operators will find the system a very resourceful addition to their WLAN management systems.

1.8 Scope and Limitation

This research is limited to detection of rogue wireless access points on a multivendor WLAN. Elimination of rogue APs is beyond the scope of this research. The result of the research is a cross-platform system that reads and interprets beacon frames from connected APs and compares the obtained parameters against set rules to categorize them as either authorized, rogue or external.

VT OMNES VNVM SINT

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Many researchers have already proposed various models for detecting rogue wireless access points on WLANs. Some software such as vendor specific WLAN controllers and wireless intrusion prevention systems (WIPS) like solarWinds and Airtights are already doing the job albeit with many demerits. This chapter explores some of the proposed and existing solutions for detecting and eliminating rogues APs. A critical view of these solutions is discussed to make the existing gap and the problem under study clearer. Research constraints for the topic under this research are latter discussed in the chapter followed by an elaborate conceptual framework for the proposed rogue detection framework.

2.2 Existing and Proposed Wireless Rogue AP Detection Solutions

Pesce (2006) implemented a low cost rogue access point detection system using Kismet and disposable hardware. He used Kismet_drone on Linksys WRT54GL wireless router. This required flashing of the router's firmware and using extra hardware that the user had to purchase. Configuring Kismet on Linksys WRT54GL is a complex process that is also time consuming especially if the access points are many. There were some difficulties obtaining signal information from Kismet_drone on a WRT54GL due to the nature of the wireless drivers. This type of a solution has many technical difficulties and ineffective for large scale deployment (Pesce, 2006).

Han H. et al (2009) proposed a measurement based rogue AP detection technique that uses timing information based on round trip time (RTT) to allow the client device to independently determine whether an AP is a legitimate or not without assistance from the WLAN administrator. The algorithm relies only on existing networking protocols to work, and can be applied to any regular WLAN. They considered a malicious attacker that actively controls the rogue AP to avoid detection as opposed to an accidental rogue AP deployed non malicious people (Han H. , Sheng, Tan, Li, & Lu, 2009). The algorithm can introduce significant delays on a WLAN as clients probe the access points and wait for query such as DNS responses before sending data. It may also wrongly categorize mesh APs as rogues.

Bo Yan et al.(2009) proposed a method that involves an administrator on Ethernet LAN checks using wireless traffic sniffing tools to detect rogue APs. The algorithm works well and easily allows other rogue AP models on a Wi-Fi channel that is even congested. However, attackers who knows the algorithm can block the traffic that verifies them easily on their AP to launch attacks to other APs on a WLAN.

Wei et al (2007) modelled a rogue AP detection algorithm that makes use of sequential hypothesis test to detect rogue APs on a WLAN. In their approach, inbound traffic is monitored on a router and a decision is made on the passively collected TCP-ACK pairs. However, using ACK pairs restricts this approach to TCP traffic only.

Roth et al (2008) came up with a technique that aids a client device to sense the presence of an evil-twin AP in a WLAN. In their approach, cryptographic key exchange is achieved by short authentication string protocols. The short strings are verified by encoding the short strings as a sequence of color, carried out sequentially by user device and from particular access point. Wireless APs are required to have a light with ability to show double colors. The devices should have a minimum of one button in addition to displaying the double colors. Every string of authentication is changed into a sequence of color made up of two unique colors. The wireless AP and the client device both render the sequences one color at a time. The AP's light must be mounted where users can see it and trust that it is a genuine AP. The person using the device presses a defined button on their device causing both lights to display subsequent color in the series. The colors continue displaying for the length of time the button is pressed down and therefore users can see the colors as long as they need to see comparison between them effectively. The user can choose how much of the sequence to compare depending on the desired security level. Once the user has completed comparing, they indicate whether they accept or deny the connection by the designated button or other modes of input. This approach is limiting and requires user devices to have specific button. Most user-engaging solutions are not effective because once a user has a successful connection, they do not bother checking on the whether the connecting AP is genuine or rogue.

Another approach looked at a composite system that detects and counter-control the rogue APs on a WLAN. The approach proposed a centralized system that gathers wireless data simply with the help of its own access points. The wireless data is analyzed to detect rogue AP. After the data analysis, if rogue AP is found, then the central system uses a switch to disable the port to which rogue access point is connected. The solution is effective and low cost and also works on existing WLAN. If the central system takes the wireless data from rogue access point for analysis then whole system will not work correctly (Srisalak, K, & A, 2009).

Nikbakhsh et al (2012) proposed a client side approach for the detection of Man-in-the-Middle attack and Evil-Twin attack performed by rogue access point. This method compares the gateways and routes through which packets travels and warns users to avoid connecting to the rogue APs. The method can easily be implemented without modifying a network and without involving network admin. It is also easy to implement on mobile devices but an attacker can easily break its security using a sniffing programs.

Vanjale and Mane (2014) developed a rogue access point elimination model that uses two approaches, learning mode and detection mode, as shown in their system architecture in figure 2.1 below.

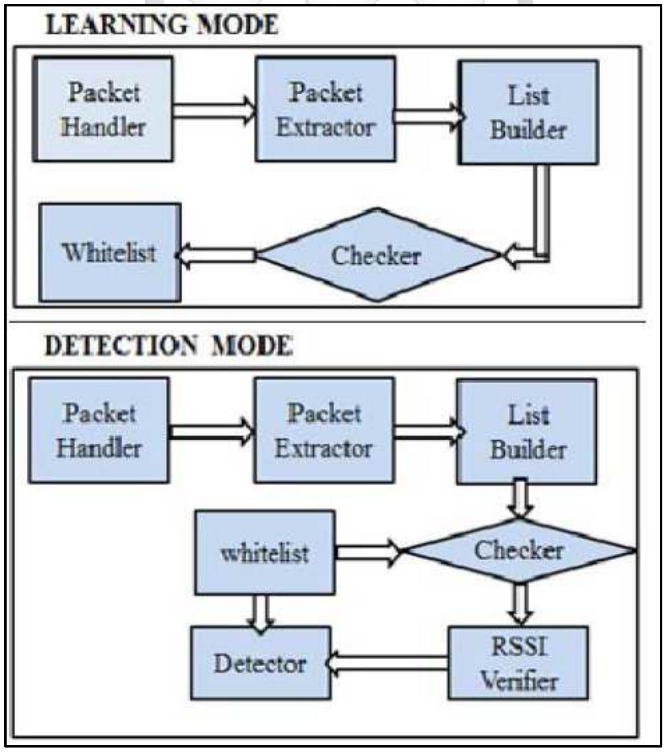


Figure 2.1: Learning and Detection Mode System Architecture

Source: Vanjale and Mane (2014, p3)

The learning mode creates a list of authorized APs with their parameters such as MAC address, SSID and Received Signal Strength (RSSI) of the access point. Their mode of detecting rogue APs inputs the list of authorized access points parameters and checks against the detected parameters. Their sytem checks one parameter after the other beginning with SSID, then MAC address and finally RSSI. If the detected AP has matching parameters or RSSI in the range of +10 to -10 it is categorized as authorized automatically. The system can work easily on small networks but it can be very slow on large networks due to processing time need. This approach is also prone to MAC address spoofing and it is possible to give false positives to system administrators.

Jana & Kasera (2008) provides server side approach using clock skews of access point in a wireless network. This approach has a number of challenges. First, it cannot detect MAC address spoofing. It also lacks accuracy. It is slow when calculating clock skews in TCP/ICMP. The approach does not favor lightweight solution. However, the approach has some advantages, which includes ability to measure effect of temperature variations, virtualizations and Network Time Protocol (NTP) synchronization on clock skews. Clock skews can act like fingerprints and thus they can be unique to each AP.

To help in reducing the cost of deploying sensors, Bahl, et al. (2006) developed an approach that uses dense array of dedicated sensor nodes which uses cheap radio devices like USB wireless adapters. Their proposed framework called Dense Array of Inexpensive Radios (DAIR) was found useful in detecting rogue APs attached to corporate networks and also detecting Denial of Service attacks on Wi-Fi networks. Their proposed solution was based on two assumptions that there are many desktop machines with good wired connection, and extra CPU and disk resources in every enterprise environment. Secondly, cheap USB-based wireless adapters can be easily obtained in the market. They attached these cheap adapters to desktop computers, and dedicated the adapters to fully monitor the Wi-Fi network. This way they achieved a low cost wireless management infrastructure. This solution will however fail in an environment where different vendor APs operate. The desktop computers must be in the range of the rogue APs signal for detection and this can be highly limiting and causing inflexibility on the WLAN topology. Their two assumptions are also not necessarily true. Figure 2.2 below displays the arrangement of components on their proposed sensor infrastructure. This set up will fail on a multivendor environment.

Chirumamilla & Ramamurthy (2003) and Sriram, Sahoo, & Agrawal (2010) proposed a system to detect and response to intrusion that works using agent model for rogue APs. Based on their system, every agent device has network cards that have capability to sniff communication traffic on a given network and return detailed information on packets of any newly added APs to a central server.

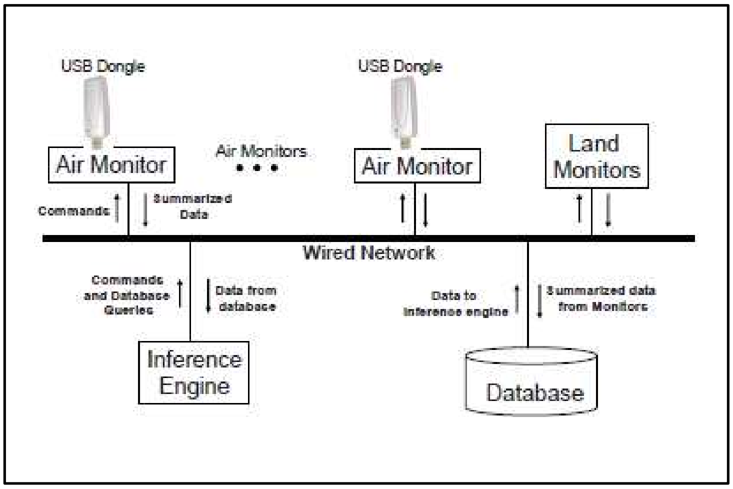


Figure 2.2: System Architecture of DAIR

Source : Bahl, et al. (2006, p3)

The server compares the information collected from the APs to a database of known APs manually created to detect rogue APs. This approach requires installation of Mobile Agent System and Client Application on APs, a task that can be complex and difficult to achieve in a multi-vendor AP environment. Their system structure is as depicted in figure 2.3 below.

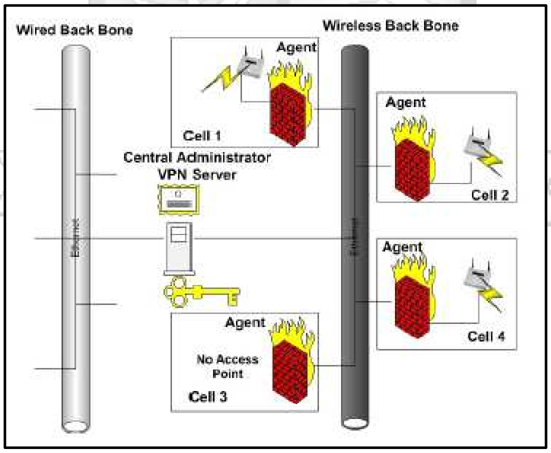


Figure 2.3: System Architecture of an Agent Model

Source: Chirumamilla & Ramamurthy (2003, p 493)

The approach is also vulnerable to MAC address spoofing making it less suitable for actual use.

Kohno, Broido, & Claffy (2005) have demonstrated that the skew of the clock of a given device stays unchanged in the course of time but vary importantly from one device to another. Jana &

Kasera (2008) went further to explore the use of clock skew of a WLAN AP as a fingerprint to uniquely identify rogue APs. They calculated every AP's clock skews by collecting their beacons and probe messages. If an AP's clock skew differs from the clock skews saved in a database, the AP is classified as rogue. This approach is effective for identifying rogue APs connected to a WLAN by malicious outsiders, but cannot effectively identify rogue APs connected to a network by malicious insiders due to periodic clock synchronization among nodes on the network.

Corbett, Beyah, & Copeland (2006) proposed the use of a spectral analysis to sense wireless traffic and identify the type of Network Interface Cards (NIC) on a network. NICs that are different from that of authorized node are noticed because 802.11 physical layer consists of many data transfer rates whereby each rate relates to a different physical layer (PHY) frequency modulation scheme. The rate switching algorithm specified by the vendor in each (node) AP selects the proper rate (frequency modulation scheme) for transmitting packets. The rate may change during frame transmission and hence significant and unique jumps in the Inter-packet Arrival Time (IAT) occurs. Any person can perform artificial production of the variations and the unique signatures to the wireless traffic on the network will be created by malicious attackers. Any unidentified signatures on the network correspond to rogue nodes (APs). This approach is not effective on a multivendor AP WLAN.

Kaoa, Liaob, & Lib (2009) proposed a client-side bottleneck bandwidth as a distinguishing feature between wired and wireless hosts. Their approach establishes whether packets from a given IP address are originating from APs based on client-side bottleneck bandwidth. This reduces network monitoring to a stationary office job by examining the packets traversing the core switch. Accuracy of the method is claimed to be over 99% when the parameter, sliding window size, of the proposed algorithm is larger than 20. Though the method effectively reduces monitoring workload while improving network security, it is subject MAC spoofing and ineffective on a multivendor WLAN.

Song, Yang, & Gu (2010) leveraged on the packet transmission structure and extra hops present in an evil twin attack set up. Where an evil twin AP exists, the client reaches a remote server through an evil twin AP and an authorized AP. This introduces one more wireless hop in the communication. Using the Inter-packet Arrival Time (IAT), it is possible to detect presence of rogue AP. This can achieve detection of rogue APs on the network but will fail on a meshed AP set up.

Monica & Ribeiro (2011) developed WiFiHop, a client-sided tool that uses the intrinsic multi-hop traits of the evil twin attack, to detect the attack. Their tool claims technology independence and detects the attacks before any user traffic is transmitted. In their approach, the user sends a watermarked packet to the echo server, and then listens on different channels. If an evil twin attack exists, the watermark will of course be seen on the link between the evil twin AP and the legitimate AP. This method will also fail where a meshed network exists by wrongly classifying an AP on mesh as rogue.

Chumchu, Saelim, & Sriklauy (2011) proposed the use of Physical Layer Convergence Protocol (PLCP) header of IEEE 802.11 frames to detect a rogue AP from an authorized AP. The frequency variation types and data rates in PLCP header are dependent on rate algorithm used in the drivers of the wireless adaptors or a vendor's AP and the environment. Hence, it is very difficult to imitate the modulation. But it is possible that the data rate of a rogue AP can be similar to the data rate of an authorized AP because of the limited data rates and modulation types found in 802.11 specifications thus making the approach vulnerable to rogue AP attacks.

Kagan (2003) suggested a TCP Fingerprinting method where the differences in the way a target AP responds to some well-defined packets are recorded with the intention of establishing the operating system target AP system. Unique packet fields from various operating systems are captured. This approach is advantageous because it uses simple tools like NMap to perform a scan of the entire network and collect the analyzed information form rogue APs. It however takes a long time to scan a large network. Also, TCP fingerprinting cannot achieve 100% accuracy. Nmap makes guesses of the operating systems using the data it obtains but it may give false positives or false negatives sometimes.

Kangsuk, et al. (2012) proposed a user-side framework based on the security condition of AP's which includes the encryption and authentication type already specified by the APs vendors. They agued that it is hard to imitate the authentication type that is specified as IEEE 802.1X by the AP vendors. This framework sniffs beacon frames to collect necessary information. Genuine APs' SSID and security level stored in a database by the network operator are compared to collected AP information. An AP whose information varies from the stored information is categorized as rogue. Inaddition, if suspected rogue AP's security level is lower, the user can optionally decide to use a more secure channel suggested by the framework. This framework is not effective on a multivendor AP WLAN because any varying AP is detected as rogue.

2.2.5 Research Constraints

This research focused on developing a framework for detecting rogue access points on a multivendor WLAN. It was however limited to detection only. Elimination of the detected rogue APs under the proposed framework will be explored in future due to limited time available for this research. The research was done within a tight schedule, being part of the author's postgraduate course requirement, and as such some very fine concepts of WLAN may not be covered.

2.3 Conceptual Framework

Detection of rogue access points is dependent on the presence of rogue APs on a WLAN. The system captures beacon frames and extract parameters such as SSID, ESSID and capability information that include encryption standards from the beacon frames broadcasted by the connected different vendor APs. The sniffed parameters represent an active AP on the network. The system compares the sniffed parameters to pre-stored authorized parameters, if they differ, the AP is marked as rogue. If the sniffed parameters are the same as those of an existing authorized AP, the detected AP is put on a waiting list until the network administrator approves the AP as authorized. This ensures that even if a rogue AP spoofs a MAC address of an authorized AP (MAC address spoofing is common in Wi-Fi attacks), the network administrator will still notice the rogue AP. This also prevents the system from wrongly categorizing a rogue AP as authorized (issuing false alerts). Output from the system is communicated to the network administrator inform of alerts in a summary table by the system. The concept of the system can be depicted diagrammatically as shown in figure 2.4 below:

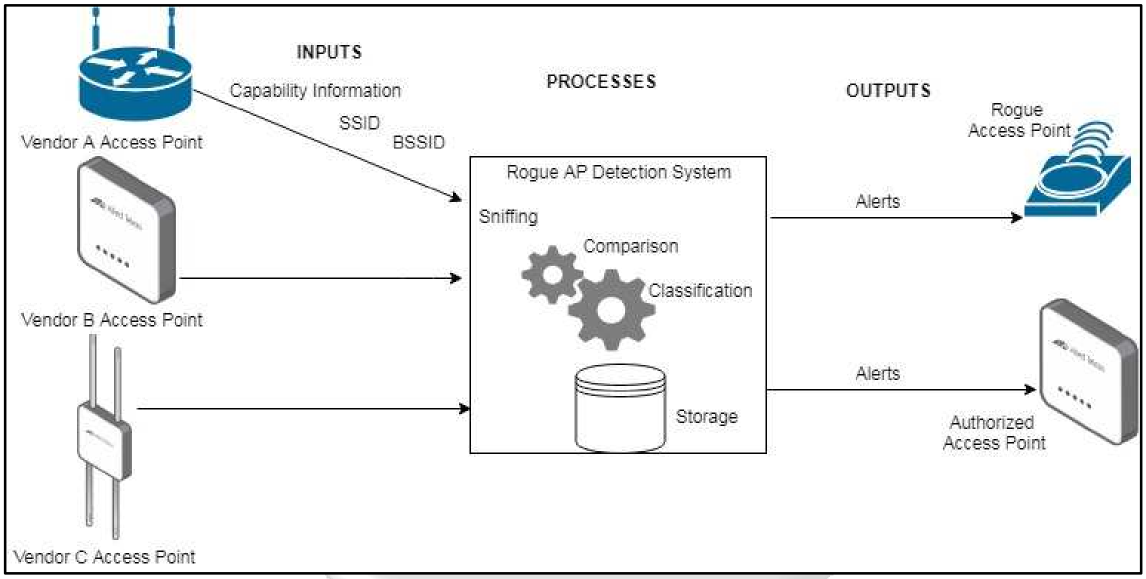


Figure 2.4: Conceptual Framework

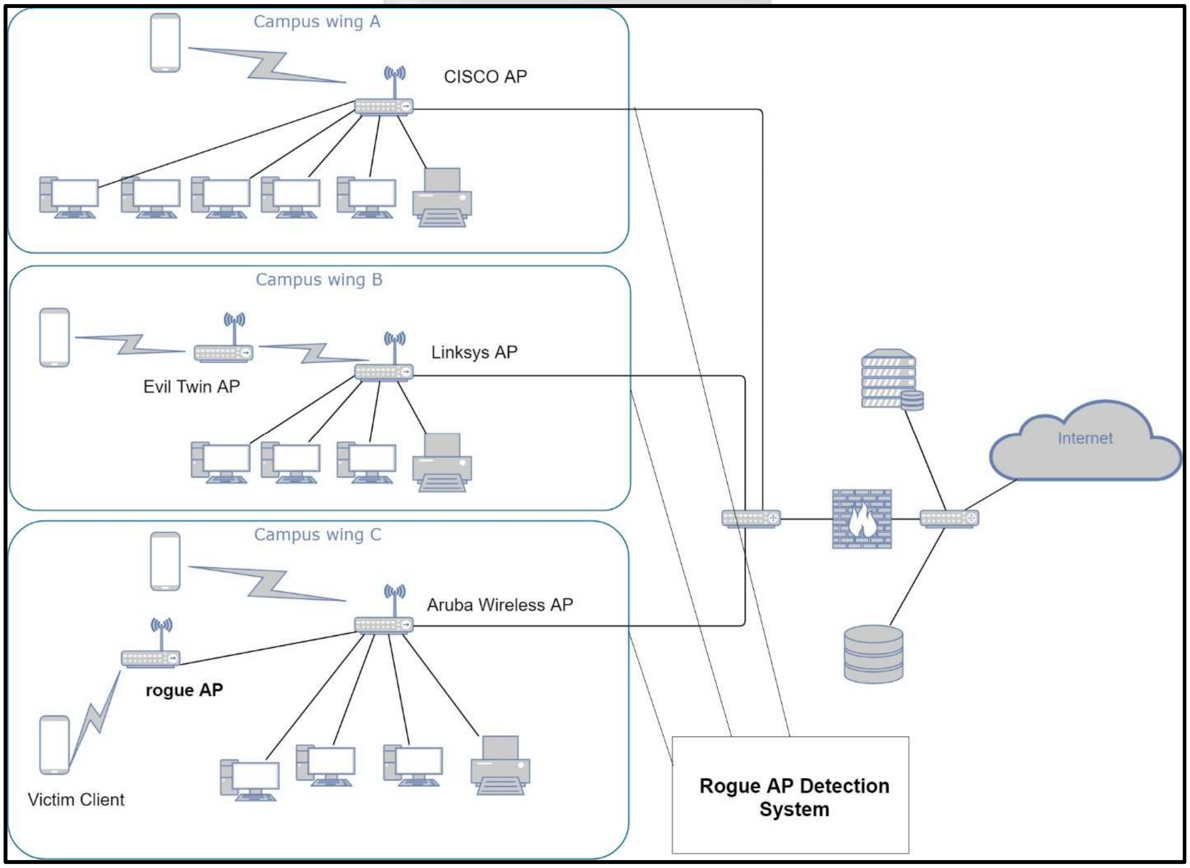


Figure 2.5: Multivendor WLAN with Rogue AP Detection System

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses in details the way the researcher will carry out the study and construct the proposed solution. It outlines the tools and techniques that will be used by the researcher to collect and analyze necessary data. Contents of the chapter are therefore organized into following subheadings; research design, location of the study, target population, data collection and analysis.

3.2 Research Design

This research design was mainly experimental whereby the researcher performed experiments with different access points in a computer laboratory in the research location. The researcher performed experiments on different vendor access points to establish common and distinguishing characteristics of the APs. Experimental method was adopted because the result from the experiments were expected to be specific and should be able to applied to other similar projects once analyzed. It can also be used together with other research methods (University of Southern California, 2010). Complete observations were made and access point parameters contained in beacon frames were recorded by the researcher. The researcher further reviewed vendor specific access point manuals and documentation on APs to obtain more information necessary in describing the access points. Quantitative approach was important in analyzing and understanding properties of different vendor APs and the common parameters between them. Results from the system testing included some numerical data which were well analyzed before drawing conclusions.

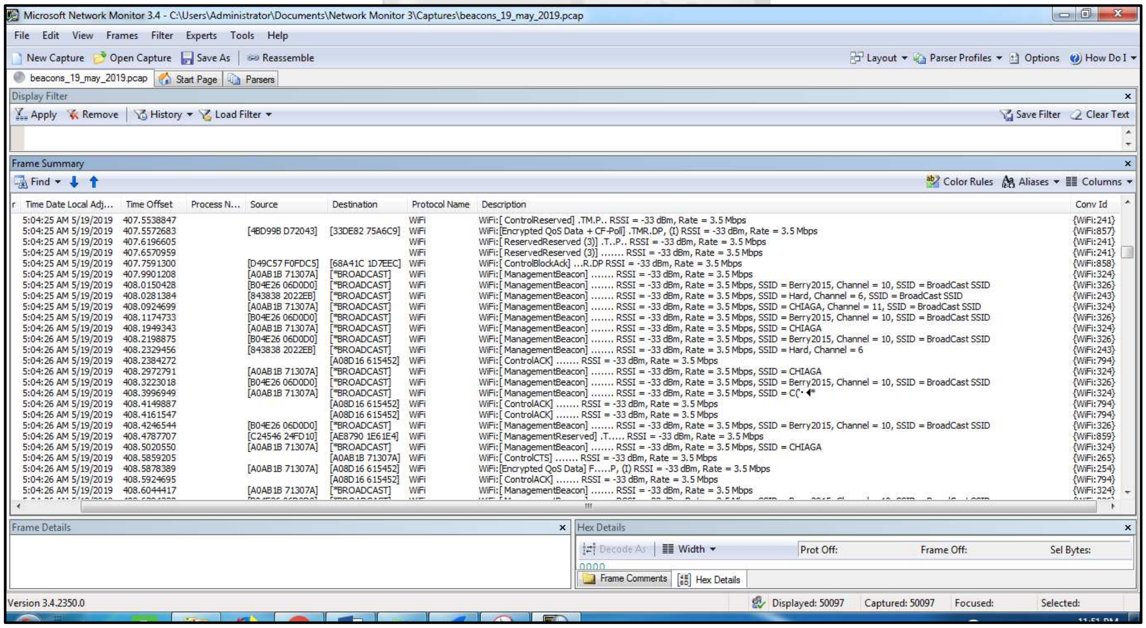
3.3 Location of the Study

The study was conducted in Strathmore University in Nairobi city in Kenya because it was easier for the researcher to obtain permission to setup an experimental WLAN in one of the university research labs. The researcher, being a student of the university, had access to the lab freely. There was no need of any special access to the network because the researcher only needed a working space with access points to experiment with and ordinary user access rights to the network. The researcher had total control of the experimental WLAN setup.

3.4 Data Collection

3.4.1 The experiment

The experiment consisted of active D-Link, Tp-Link and a Samsung phone with hotspot on as the access points, software tools and a laptop. The D-Link access point was configured to broadcast an SSID named “CHIAGA”, Tp_Link broadcasted “Berry2015” while the Samsung hotspot broadcasted “Hard”. The software component included Microsoft Network Monitor 3.4 for changing the laptop’s network interface card into monitor mode under Windows 7 operating system and also sniffing beacon frames, Wireshark was used to analyze the captured frames to help the researcher make observations of important parameters common in beacon frames. The Microsoft Network Monitor 3.4 should first be configured to enable it to sniff on appropriate interface as needed by the user. The network interface card must be changed into monitor mode to enable it listen to management frames such as beacons on the network. The access points were powered up. Figure 3.1 below shows actual sniffing of Wi-Fi management frames of type beacon frames using Microsoft Network Monitor 3.4 tool.



Microsoft Network Monitor 3.4 - C:\Users\Administrator\Documents\Network Monitor 3\Captures\beacons_19_may_2019.pcap

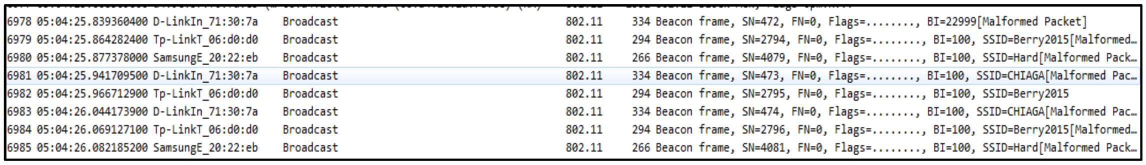
Time Date Local Adj...	Time Offset	Process N...	Source	Destination	Protocol Name	Description	Conv Id
5:04:25 AM 5/19/2019	407.5538847				WiFi	WiFi:[ControlReserved] .TM.P.. RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:241}
5:04:25 AM 5/19/2019	407.5572683		[4BD99B D72043]	[33DE82 75A6C9]	WiFi	WiFi:[Encrypted QoS Data + CF-Poll] .TMR.DP, (I) RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:857}
5:04:25 AM 5/19/2019	407.6196605				WiFi	WiFi:[ReservedReserved (3)] .T..P.. RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:241}
5:04:25 AM 5/19/2019	407.6570959				WiFi	WiFi:[ReservedReserved (3)] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:241}
5:04:25 AM 5/19/2019	407.7591300		[D49C57 F0FDC5]	[68A41C 1D7EEC]	WiFi	WiFi:[ControlBlockAck] ...R.DP RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:858}
5:04:25 AM 5/19/2019	407.9901208		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:324}
5:04:25 AM 5/19/2019	408.0150428		[B04E26 06D0D0]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Berry2015, Channel = 10, SSID = BroadCast SSID	{WiFi:326}
5:04:25 AM 5/19/2019	408.0281384		[843838 2022EB]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Hard, Channel = 6, SSID = BroadCast SSID	{WiFi:243}
5:04:25 AM 5/19/2019	408.0924699		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = CHIAGA, Channel = 11, SSID = BroadCast SSID	{WiFi:324}
5:04:25 AM 5/19/2019	408.1174733		[B04E26 06D0D0]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Berry2015, Channel = 10, SSID = BroadCast SSID	{WiFi:326}
5:04:26 AM 5/19/2019	408.1949343		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = CHIAGA	{WiFi:324}
5:04:26 AM 5/19/2019	408.2198875		[B04E26 06D0D0]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Berry2015, Channel = 10, SSID = BroadCast SSID	{WiFi:326}
5:04:26 AM 5/19/2019	408.2329456		[843838 2022EB]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Hard, Channel = 6	{WiFi:243}
5:04:26 AM 5/19/2019	408.2384272			[A08D16 615452]	WiFi	WiFi:[ControlACK] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:794}
5:04:26 AM 5/19/2019	408.2972791		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = CHIAGA	{WiFi:324}
5:04:26 AM 5/19/2019	408.3223018		[B04E26 06D0D0]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Berry2015, Channel = 10, SSID = BroadCast SSID	{WiFi:326}
5:04:26 AM 5/19/2019	408.3996949		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = C(·◀*	{WiFi:324}
5:04:26 AM 5/19/2019	408.4149887			[A08D16 615452]	WiFi	WiFi:[ControlACK] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:794}
5:04:26 AM 5/19/2019	408.4161547			[A08D16 615452]	WiFi	WiFi:[ControlACK] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:794}
5:04:26 AM 5/19/2019	408.4246544		[B04E26 06D0D0]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = Berry2015, Channel = 10, SSID = BroadCast SSID	{WiFi:326}
5:04:26 AM 5/19/2019	408.4787707		[C24546 24FD10]	[AE8790 1E61E4]	WiFi	WiFi:[ManagementReserved] .T..... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:859}
5:04:26 AM 5/19/2019	408.5020550		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps, SSID = CHIAGA	{WiFi:324}
5:04:26 AM 5/19/2019	408.5859205			[A0AB1B 71307A]	WiFi	WiFi:[ControlCTS] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:265}
5:04:26 AM 5/19/2019	408.5878389		[A0AB1B 71307A]	[A08D16 615452]	WiFi	WiFi:[Encrypted QoS Data] F.....P, (I) RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:254}
5:04:26 AM 5/19/2019	408.5924695			[A08D16 615452]	WiFi	WiFi:[ControlACK] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:794}
5:04:26 AM 5/19/2019	408.6044417		[A0AB1B 71307A]	[*BROADCAST]	WiFi	WiFi:[ManagementBeacon] .... RSSI = -33 dBm, Rate = 3.5 Mbps	{WiFi:324}

Version 3.4.2350.0 Displayed: 50097 Captured: 50097 Focused: Selected:

Figure 3.1: Capturing 802.11 Beacon Frames using Network Monitor 3.4

3.4.2 Observation

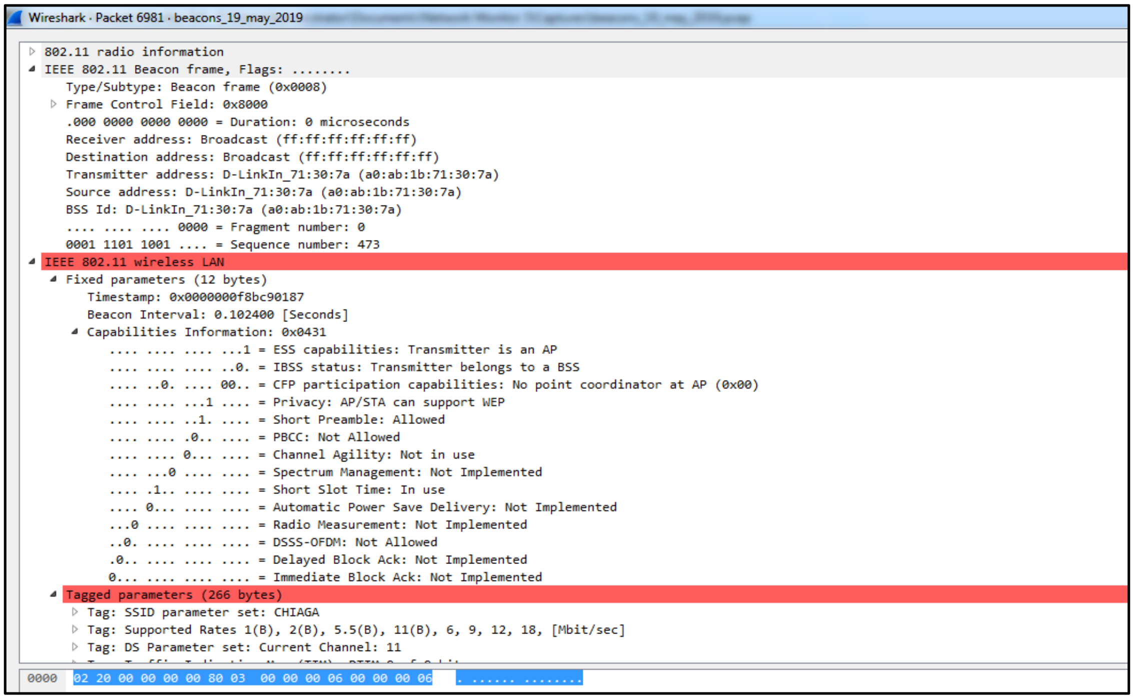
The researcher tested and made observations from the three live access points from different vendors on the most common and distinguishing properties of the APs on a WLAN. Captured information was stored in a pcap file for analysis. Wireshark was used to analyze the information. Figure 3.2 below shows a snippet of Wireshark after opening the pcap file.



```
6978 05:04:25.839360400 D-LinkIn_71:30:7a    Broadcast          802.11    334 Beacon frame, SN=472, FN=0, Flags=........, BI=22999[Malformed Packet]
6979 05:04:25.864282400 Tp-LinkT_06:d0:d0    Broadcast          802.11    294 Beacon frame, SN=2794, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed…
6980 05:04:25.877378000 SamsungE_20:22:eb    Broadcast          802.11    266 Beacon frame, SN=4079, FN=0, Flags=........, BI=100, SSID=Hard[Malformed Pack…
6981 05:04:25.941709500 D-LinkIn_71:30:7a    Broadcast          802.11    334 Beacon frame, SN=473, FN=0, Flags=........, BI=100, SSID=CHIAGA[Malformed Pac…
6982 05:04:25.966712900 Tp-LinkT_06:d0:d0    Broadcast          802.11    294 Beacon frame, SN=2795, FN=0, Flags=........, BI=100, SSID=Berry2015
6983 05:04:26.044173900 D-LinkIn_71:30:7a    Broadcast          802.11    334 Beacon frame, SN=474, FN=0, Flags=........, BI=100, SSID=CHIAGA[Malformed Pac…
6984 05:04:26.069127100 Tp-LinkT_06:d0:d0    Broadcast          802.11    294 Beacon frame, SN=2796, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed…
6985 05:04:26.082185200 SamsungE_20:22:eb    Broadcast          802.11    266 Beacon frame, SN=4081, FN=0, Flags=........, BI=100, SSID=Hard[Malformed Pack…
```

Figure 3.2: Wireshark Analysis of Beacon Frames

The researcher performed a deeper analysis of beacon frames from each of the access points. Figure 3.3 below shows a snapshot of what was revealed when the D-Link packet highlighted in figure 3.2 above was opened for deeper analysis. It was noted that a beacon frame is of layer IEEE Dot11 standard. It is of type 0 and subtype 8. It carries detailed information about the sending access point.



Wireshark · Packet 6981 · beacons_19_may_2019

```
▷ 802.11 radio information
◢ IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  ▷ Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    Source address: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    BSS Id: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    .... .... .... 0000 = Fragment number: 0
    0001 1101 1001 .... = Sequence number: 473
◢ IEEE 802.11 wireless LAN
  ◢ Fixed parameters (12 bytes)
      Timestamp: 0x0000000f8bc90187
      Beacon Interval: 0.102400 [Seconds]
    ◢ Capabilities Information: 0x0431
        .... .... .... ...1 = ESS capabilities: Transmitter is an AP
        .... .... .... ..0. = IBSS status: Transmitter belongs to a BSS
        .... ..0. .... 00.. = CFP participation capabilities: No point coordinator at AP (0x00)
        .... .... ...1 .... = Privacy: AP/STA can support WEP
        .... .... ..1. .... = Short Preamble: Allowed
        .... .... .0.. .... = PBCC: Not Allowed
        .... .... 0... .... = Channel Agility: Not in use
        .... ...0 .... .... = Spectrum Management: Not Implemented
        .... .1.. .... .... = Short Slot Time: In use
        .... 0... .... .... = Automatic Power Save Delivery: Not Implemented
        ...0 .... .... .... = Radio Measurement: Not Implemented
        ..0. .... .... .... = DSSS-OFDM: Not Allowed
        .0.. .... .... .... = Delayed Block Ack: Not Implemented
        0... .... .... .... = Immediate Block Ack: Not Implemented
  ◢ Tagged parameters (266 bytes)
    ▷ Tag: SSID parameter set: CHIAGA
    ▷ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    ▷ Tag: DS Parameter set: Current Channel: 11

0000  02 20 00 00 00 00 80 03  00 00 00 06 00 00 00 06   . ...... ........
```

Figure 3.3: Wireshark Analysis of a single Beacon Frame

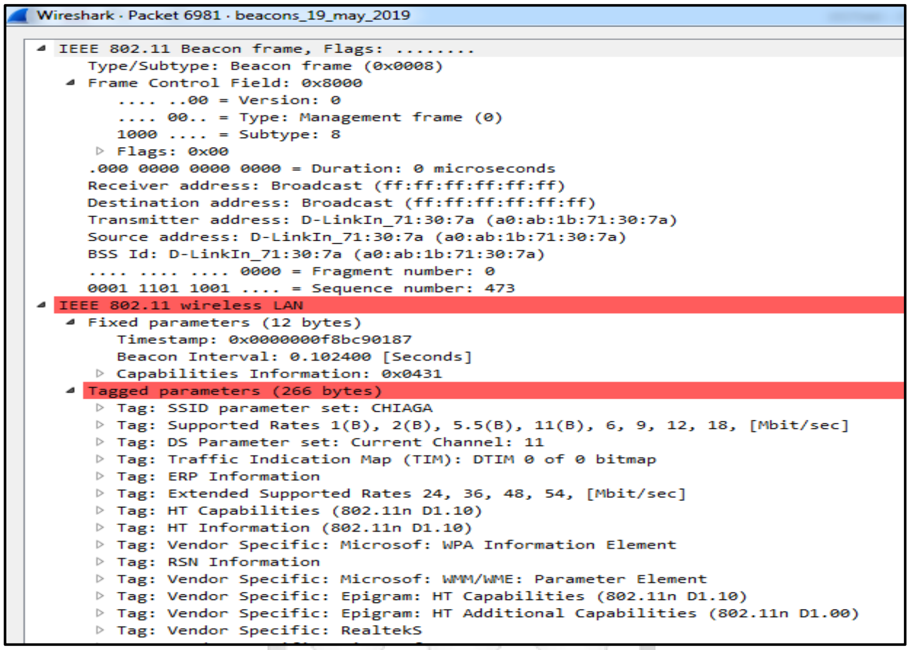
The common and unique parameters of interest to this research as observed from the Wireshark analysis in figure 3.3 above are summarized as shown in table 3.1 on the next page.

Parameter	Value Set	Description
Type/Subtype	0x0008 (common)	Serves to distinguish 802.11 beacon frames from other data frames on the network
Receiver/Destination Address	ff:ff:ff:ff:ff:ff (common)	This is a broadcast MAC address indicating that the access points send beacon frames as a broadcast so that any nearby client can receive the frames
Transmitter/Source address/BSSID	D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a) (unique)	This is the MAC address of the access point that is originating the beacon frame
Capability information	0x0431 WEP (unique)	It shows the type of encryption supported by the sending access point
SSID	CHIAGA (unique)	Service Set Identifier is a configured name of the Wi-Fi signal broadcasted by an access point

Table 3.1: Observed AP Parameters

3.5 Data Analysis

Descriptive analysis was used to summarize the captured data and to describe the parameters required in the research. This was necessary for a better understanding of what went on or what happened during the experiments and the observations. Beacon frames captured by the Microsoft Network Monitor 3.4 tool were analyzed with the help of Wireshark tool as discussed in the data collection and observation subsections of this chapter. 802.11 beacon frame is a layered frame. It consists of the following segments; 24-bit MAC header, 16-bit mandatory part with varying frame body and SSID and a 48-bit optional with varying part robust security network, extended rates, country information and Time Indication Map for each access point. The beacon frame structure is common across all access points. Destination MAC address was similar for all access points. Source MAC address, SSID and encryption were varying with each access point. SSID and encryption parameters were set by the researcher. The Wireshark capture in figure 3.4 below shows the different sections of a beacon frame.



Wireshark · Packet 6981 · beacons_19_may_2019

```
◢ IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  ◢ Frame Control Field: 0x8000
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      1000 .... = Subtype: 8
    ▷ Flags: 0x00
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    Source address: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    BSS Id: D-LinkIn_71:30:7a (a0:ab:1b:71:30:7a)
    .... .... .... 0000 = Fragment number: 0
    0001 1101 1001 .... = Sequence number: 473
◢ IEEE 802.11 wireless LAN
  ◢ Fixed parameters (12 bytes)
      Timestamp: 0x0000000f8bc90187
      Beacon Interval: 0.102400 [Seconds]
    ▷ Capabilities Information: 0x0431
  ◢ Tagged parameters (266 bytes)
    ▷ Tag: SSID parameter set: CHIAGA
    ▷ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    ▷ Tag: DS Parameter set: Current Channel: 11
    ▷ Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
    ▷ Tag: ERP Information
    ▷ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    ▷ Tag: HT Capabilities (802.11n D1.10)
    ▷ Tag: HT Information (802.11n D1.10)
    ▷ Tag: Vendor Specific: Microsof: WPA Information Element
    ▷ Tag: RSN Information
    ▷ Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
    ▷ Tag: Vendor Specific: Epigram: HT Capabilities (802.11n D1.10)
    ▷ Tag: Vendor Specific: Epigram: HT Additional Capabilities (802.11n D1.00)
    ▷ Tag: Vendor Specific: RealtekS
```

Figure 3.4: Beacon Frame Structure in Wireshark

The structure can be well understood using figure 3.5 below that was created by Gast (2005).

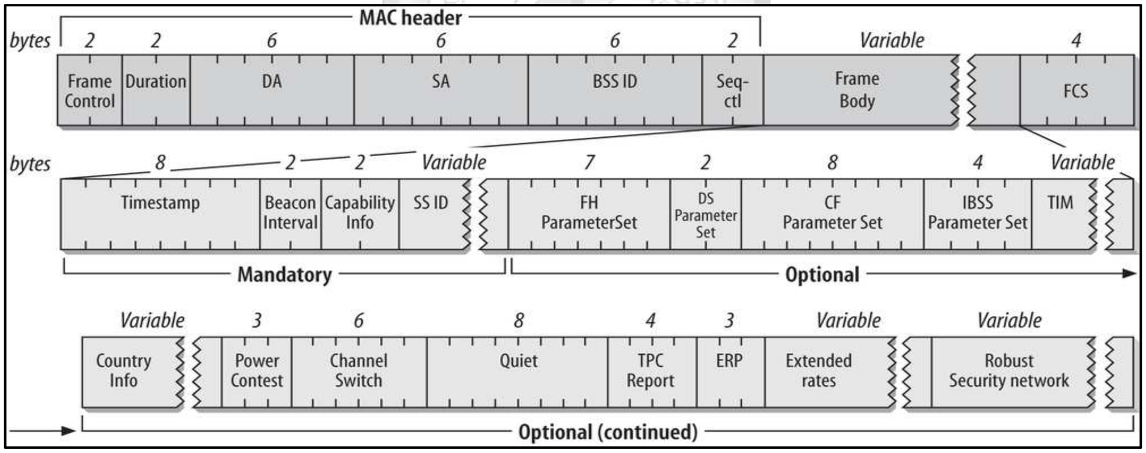


Figure 3.5: Beacon Frame in Detail

Source : Gast (2005)

The researcher established that frame type and subtype, source address, SSID and encryption found in the Capability Information were important parameters in developing the rogue AP detection model on a multivendor WLAN environment. In the model, authorized access points at least similar SSID and encryption. The model is able to detect these parameters automatically by sniffing beacon frames and using an algorithm to find the specific parameters. This is possible using Python’s Scapy module which is very good for network packet analysis. The sniffed parameters are compared to a database of authorized access points. If a match is found, the detected access point is put on a waiting list for further approval by the network administrator. This ensures that no rogue access point can spoof the MAC address of an authorized access point unnoticed. In the case of a mismatch, the algorithm classifies the detected access point as rogue and it is added to a list of rogue APs that are displayed to the network administrator. The researcher was able to develop the framework for detecting rogue APs on a multivendor WLAN developed using the parameters established in this chapter. The model was designed and developed and eventually tested as an operational rogue AP detection system prototype.

3.6 System Design and Development

The system development methodology used was Waterfall. Waterfall methodology is advantageous because it allowed for departmentalization and control of the rogue AP detection system development process. It was easier to set a schedule with deadlines for each stage of development and the system proceeded through the development stages one after the other. Development begun with the concept, which moved to system design, implementation and testing phase. The maintenance phase was left out of the methodology at this point of the study because the model is still a prototype that requires further development. Each phase of development followed in the strict order as depicted in figure 3.1 below adopted from Saracco (2018).

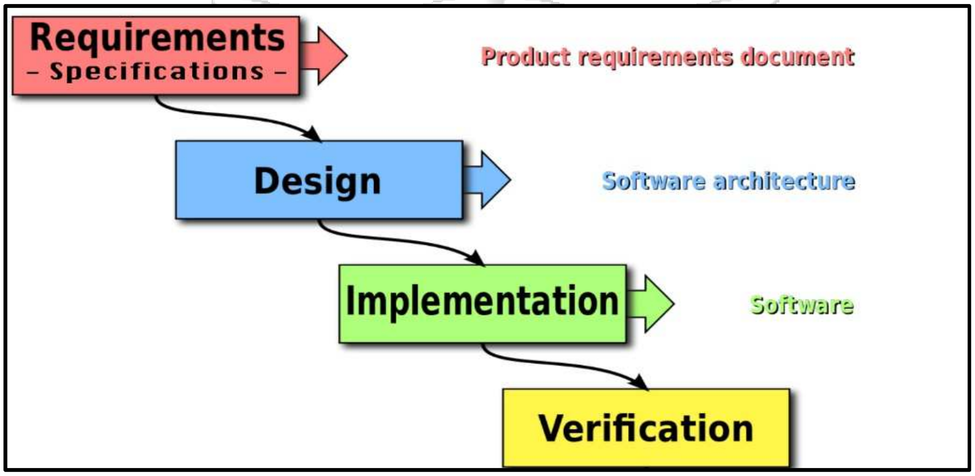


Figure 3.6: Modified Waterfall Methodology

Source : Saracco (2018)

3.6.1 System Analysis

Object oriented analysis approach was used in the study because as Kung and Lei (2016) noted; it has the ability to represent complex relationships, data and processing of the data in a stable notation that makes it easier to convert analysis outcome into designs. Using this approach enabled the researcher to create a model of the rogue AP system's functional requirements that were independent of actual implementation constraints.

3.6.2 System Design

Conceptualization and design of the system used draw.io which is a free online tool for making flowcharts, UML diagrams, charts, ER and network diagrams. Use cases and interaction diagrams were drawn using this tool. Database design was accomplished using MySQL Workbench tool which the researcher found readily available and easier to use within the limited time frame of the research.

3.6.3 System Implementation and Development

The system was developed using Python as the main language. This was mainly because Python is both a powerful high level and low-level programming language that has APIs and frameworks that can easily capture frames and packets on a WLAN for analysis. Python was also found easier to use.

3.6.4 System Testing

The rogue detection system first underwent a hardware/software testing to ensure that interactions between the hardware and software components were working as intended. A usability testing was also carried out during system testing to evaluate the system's adherence to its requirements. Test results were analyzed to draw conclusions as detailed in chapter five of this document.

3.7 Research Quality

The researcher considers quality to be key in conducting this research. The research will carefully follow the research methodology described in this chapter. Correct tools and techniques combined with the researcher's expertise were used to ensure the research maintained high quality.

3.7.1 Validity

According to Heale and Twycross (2015) validity is the extent to which a research concept is accurately measured in a quantitative study. Data collection and analysis instruments and methods used in the research were of good quality and measured correct variables as intended by the researcher. Data collected was valid as per the needs of the research.

3.7.2 Objectivity

This research was carried out without any bias. Instruments and methods used to collect and analyze data were as per the true requirements of the research.

3.7.3 Reliability

Heale and Twycross (2015) describes reliability as the consistency of a research instrument having the same results if it is used repeatedly in the same situation. Tools and Instruments used in the research were reliable.

3.8 Ethical Considerations

Privacy of data sniffed from WLANs during the research could present an ethical issue. The researcher was open to signing non-disclosure agreements (NDA) with parties that could have felt uncomfortable disclosing data. However, the researcher was keen and avoided analyzing data from Strathmore University network. Data analyzed was captured from experimental WLAN created in the lab.

VT OMNES VNVM SINT

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

The researcher gathered data through questionnaires and performing experiments with APs. The data was analyzed and requirements were specified from the collected information. System models were designed with the help of use case diagram, sequence diagram, domain model, entity relationship diagram and database schema.

4.2 Requirements Analysis

User requirements and system requirements were considered based on the feedback from the interviews with network administrators in Strathmore University and GT Bank and the observations made on different AP from different vendors. Tests done on these APs also informed the formulated requirements.

4.21 User Requirements

The rogue AP detection system must meet the following user requirements:

i. The system uses graphical user interfaces while the user is interacting with it. Some output may be displayed in a console but the user does not need to perform input into the system using a command line interface. This simplifies the system usage because the user is not required to master technical command line language.

ii. There is a screen that allows capturing of access point parameters that is, MAC address, SSID, and Wi-Fi encryption used. Alternative to this screen is the system to automatically pick the parameters from detected APs.

iii. The system displays a list of all authorized access points in a tabular format on a graphical user interface

iv. The system displays detected rogue APs in a summary table on an interface screen

v. The system performs detection and classification of access points without the user noticing delays.

vi. The system allows network administrator to confirm an access point on a waiting or rogue list as authorized or simply change that status of an access point from rogue to authorized through a graphical user interface.

vii. System can operate on different platforms with minimum or no technical modifications that require advanced skills.

4.22 Functional Requirements

These are properties of the system that makes the system to work or be functional. The proposed system meets the following functional requirements:

i. Classification of APs: The system is able to compare registered properties of authorized APs with those of rogue APs and accurately classify the rogue APs for the user to see.

ii. Reporting: The system generates reports on detected APs. The report can further be customized to show rogue APs, authorized APs or neighbor APs according to the user needs.

4.23 Non Functional Requirements

The following non-functional requirements makes the system to meet the user requirements specified above:

i. Reliability: The proposed system will not fail to detect rogue APs whenever they exist on the WLAN. Network operators will find it dependable and easier to operate.

ii. The system will be able to issue true alerts to avoid ambiguity and confusion currently seen in most wireless rogue AP detection systems available on the market today.

iii. Interoperability: The proposed system will be able to run smoothly on different platforms because of the Python technology used in developing it. If need be, it should require very minimal reconfiguration to run on different platforms.

iv. Authentication: It only allows authorized users to access it. This is a security feature to prevent unauthorized modification of the system properties and registering unauthorized APs.

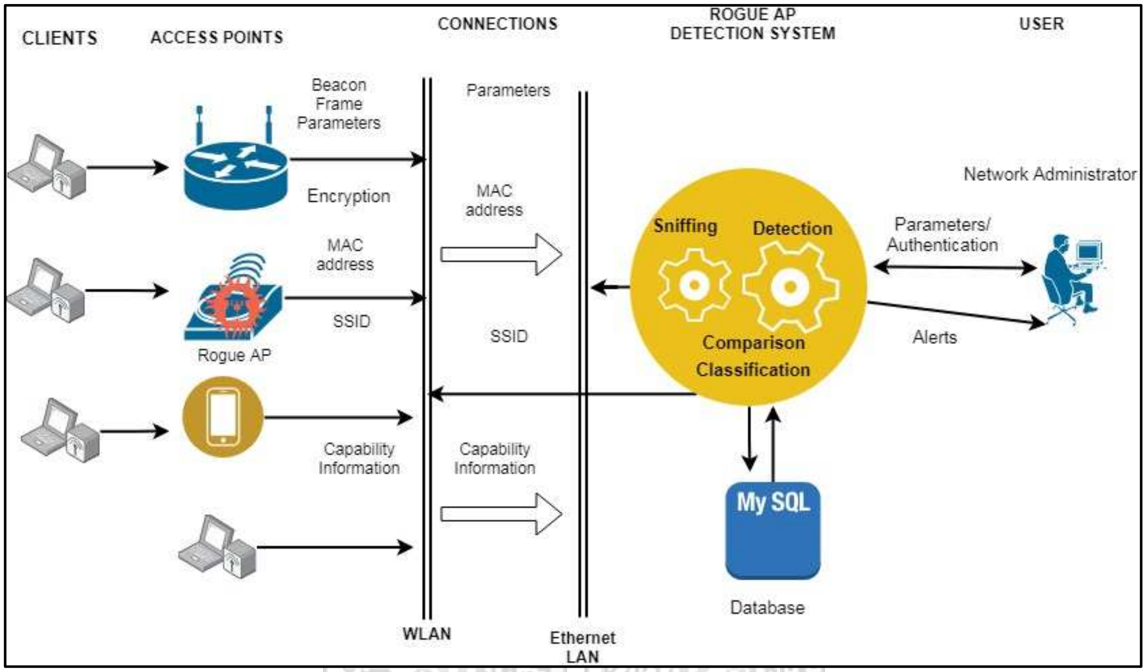
v. Secure storage of data: The system uses a relational database to securely store data from both authorized and detected rogue APs.

4.3 System Architecture

The network administrator can manually add authorized APs into the system or authorize access points that have been automatically detected by the system once authenticated. The administrator reads alerts arising from classification of access points in the system. The system securely keeps a record of all authorized APs and their parameters. When an access points get

VT OMNES VNVM SINT

connected on the network, the system collects its MAC address, SSID, encryption by sniffing and examining the beacon frames. The system then compares the captured parameters with the authorized APs parameters. If a match is found, the system puts the AP on a list of genuine APs waiting approval or disapproval from the network operator. Disapproved APs are automatically listed as rogues. If no match is found, the system will automatically classify the AP as either rogue or neighbor based on the configured AP properties. The system the presents the alerts in tabular form on interfaces where the network administrator can see. Figure 4.1 below gives a general view of the proposed system and its operations.

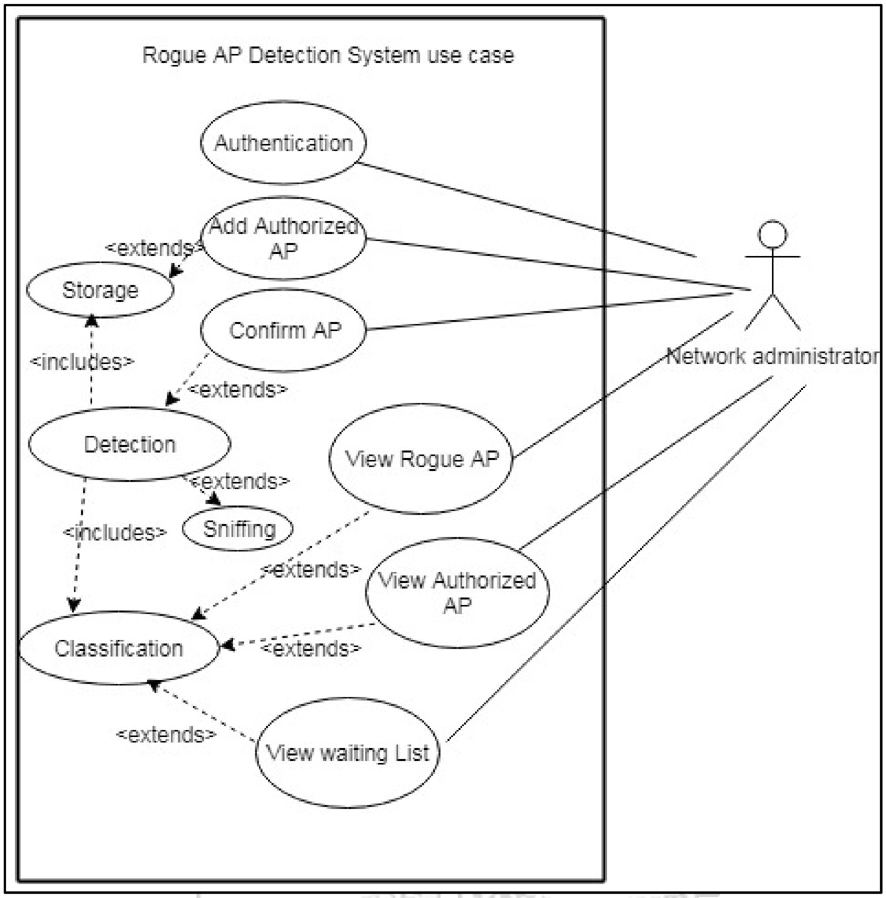


VT OMNES UNUM SINT

Figure 4.1: General System Structure

4.4 Use Case Diagram

Figure 4.2 on the next page shows the use case diagram with one actor, network administrator or operator who constantly interacts with the system. The system authenticates the user before accepting input from the user and giving any output in terms of tabular reports and alerts to the network administrator. The user is responsible of registering all authorized APs on the network into the system. The system can also automatically scan and discover APs that are plugged on to the network. Detected APs are placed on a waiting list that the network administrator must approve or disapprove for the APs to be categorized as authorized or rogue. The network administrator can also approve a rogue AP and make it authorized. Data about APs will be securely save in a relational database that can be easily accesses by the system.



VT OMNES VNVM SINT

Figure 4.2: Use Case Diagram

4.5 Sequence Diagram

Figure 4.3 shown on the next page is a sequence diagram illustrating how a user interacts with the rogue AP detection system. A series of activities and messages are passed to the system by the user or through automatic reading of beacon frames. Messages are also passed within the system itself and from the system to the user. The user is first authenticated to access the system functionalities. The system validates the login credentials by checking against a database of authorized users and login. Authentic users can register or enter parameters of authorized APs of their choice into the system. The system keeps the parameters in a database of authorized APs. The system can also perform automatic reading and extracting of parameters of any connected access point by tapping and interpreting beacon frames from the local area network. A classification of the detected APs is done within the system by comparing detected APs parameters against already defined parameters. APs whose parameters do not match the defined ones are classified as rogues while those that match are put on a waiting list until the network administrator approves them as genuine. The system displays brief reports on the different APs existing on the network. The user can also select specific reports from the system anytime they access the system.

VT OMNES VNVM SINT

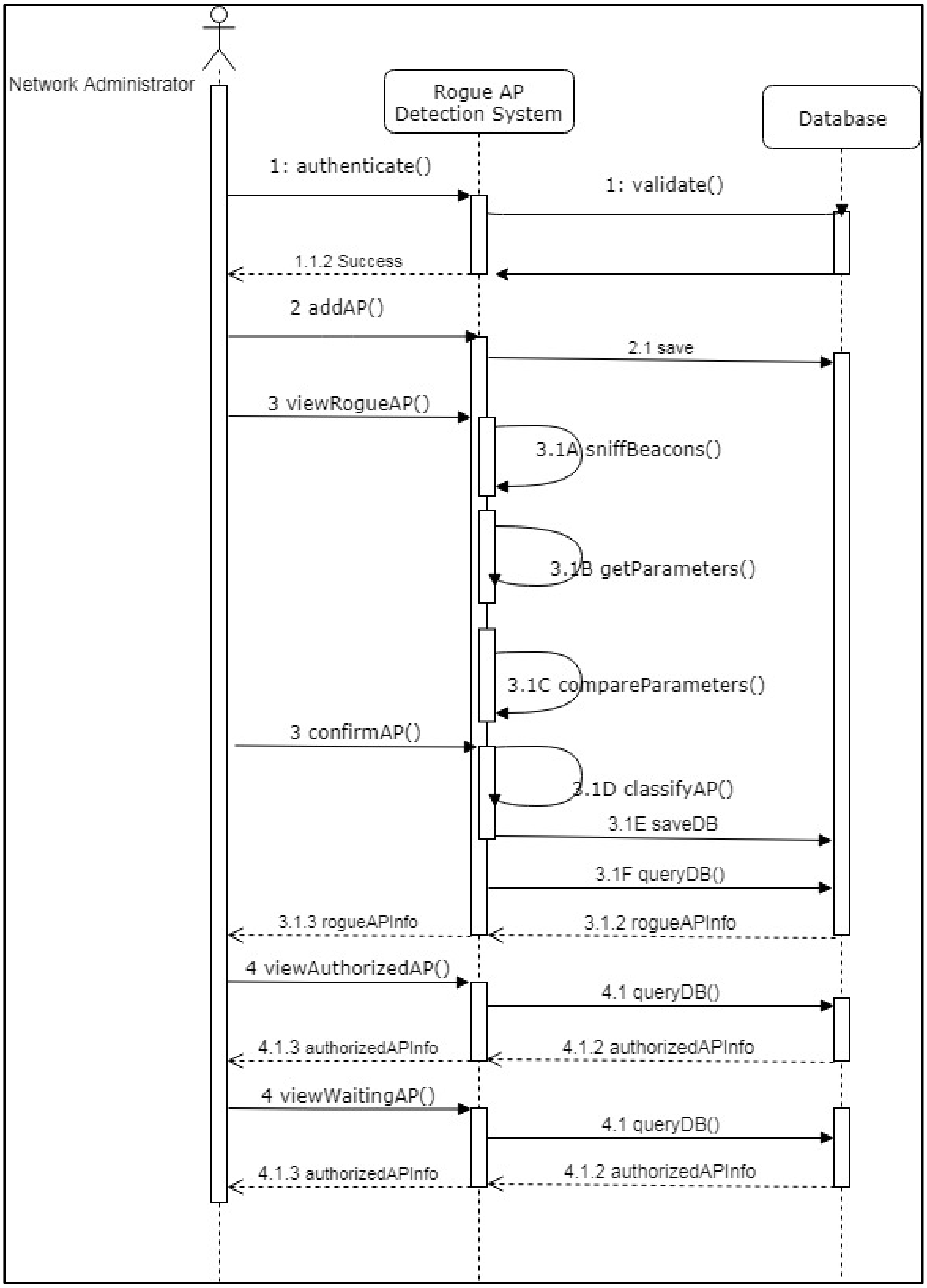


Figure 4.3: Sequence Diagram

4.6 Domain Model

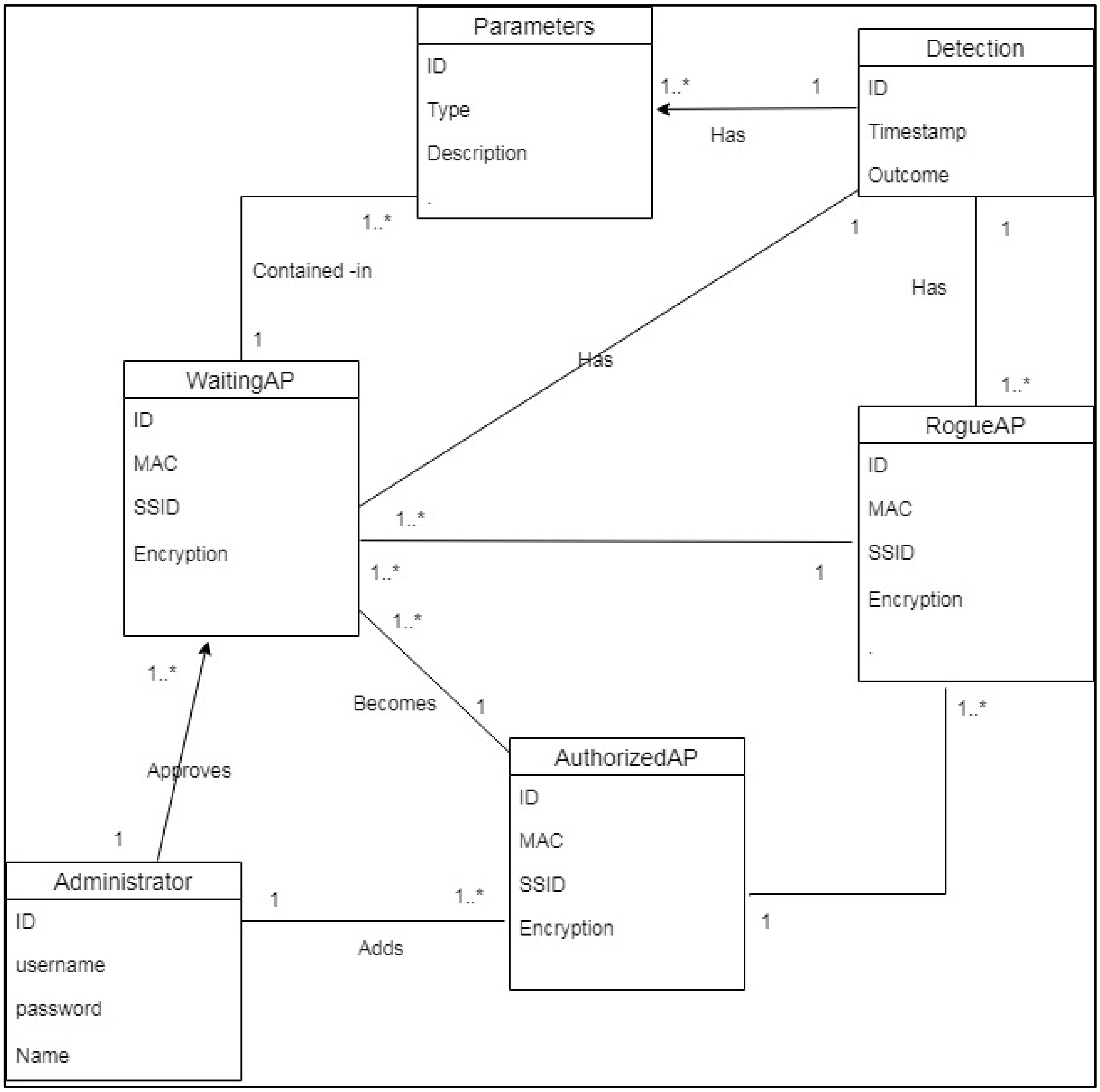


Figure 4.4: Partial Domain Model of Rogue AP Detection System

Figure 4.4 above visualizes and relates words or conceptual classes in the domain of rogue AP detection. It also represents an abstraction of the conceptual classes in the system rather than using word to describe the classes. This model shows a partial view which is also an abstraction of the entire system while ignoring unnecessary details at this point of modelling. When a beacon frame is sniffed, it is broken down to parameters relating to the sending AP. The parameters are passed through a comparison and classification process (detection) which results into the sending AP being categorized as either rogue or put on a waiting list for further approval by the network administrator. Once approved, the AP is added to the list of authorized

APs. Unapproved APs will be marked as rogues. The network administrator can also confirm a rogue AP as authorized. The system will then add the access point to a list of authorized APs.

4.7 Entity Relationship Diagram

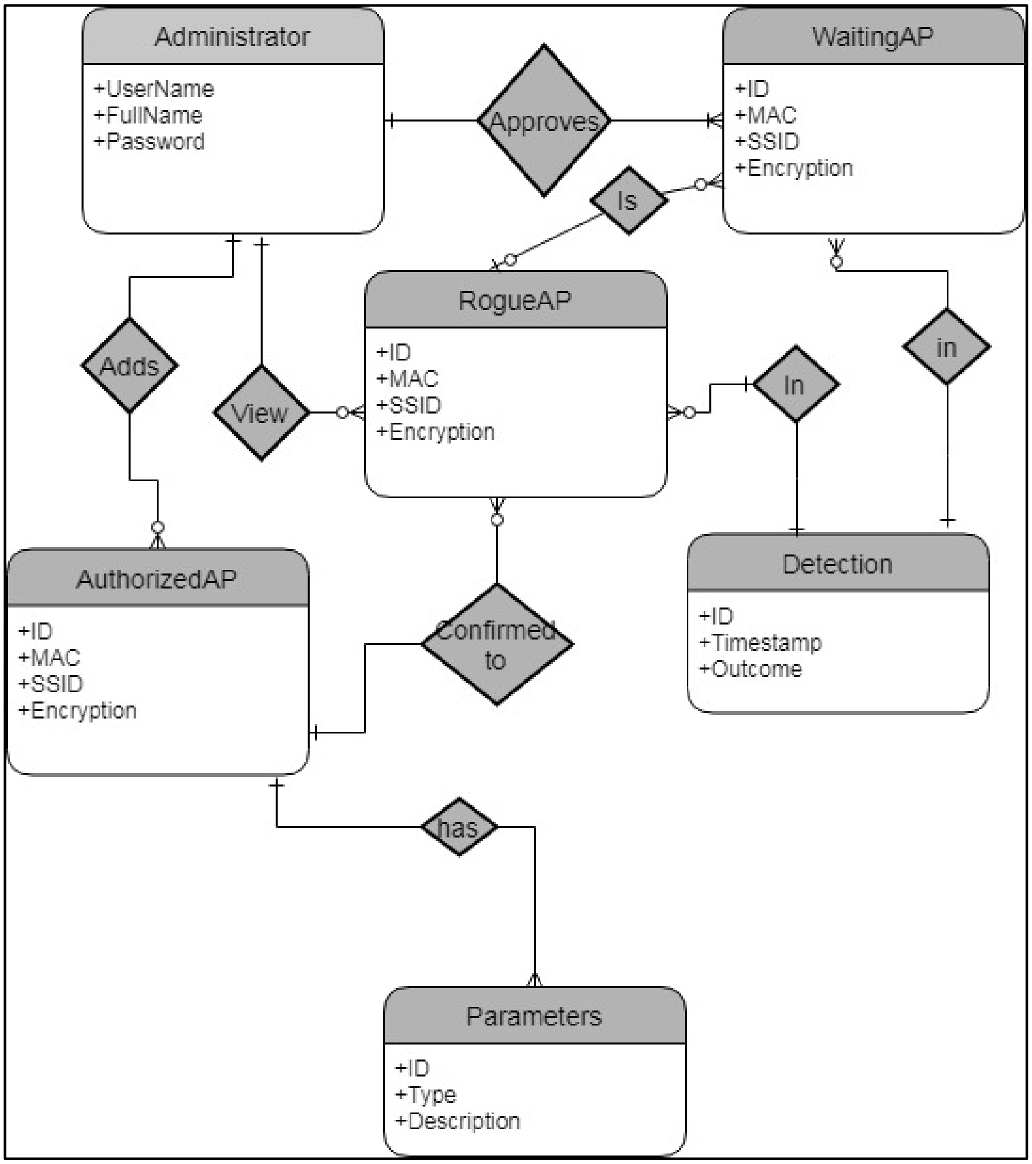


Figure 4.5: Entity Relationship Diagram

The system has a number of entities represented with their attributes as shown in figure 4.5 above. Access points have attributes such as MAC address, SSID, encryption type. Administrator entity has ID, username, full name and password attributes.

A single user can only have a single set of login details. Once authenticated the administrator can add many or zero authorized APs to the system. He can also view at least one authorized

AP. An authorized AP has many approved parameters that describe it. Many waiting APs or none can be confirmed as either rogue or authorized by the network administrator.

4.8 Database Schema

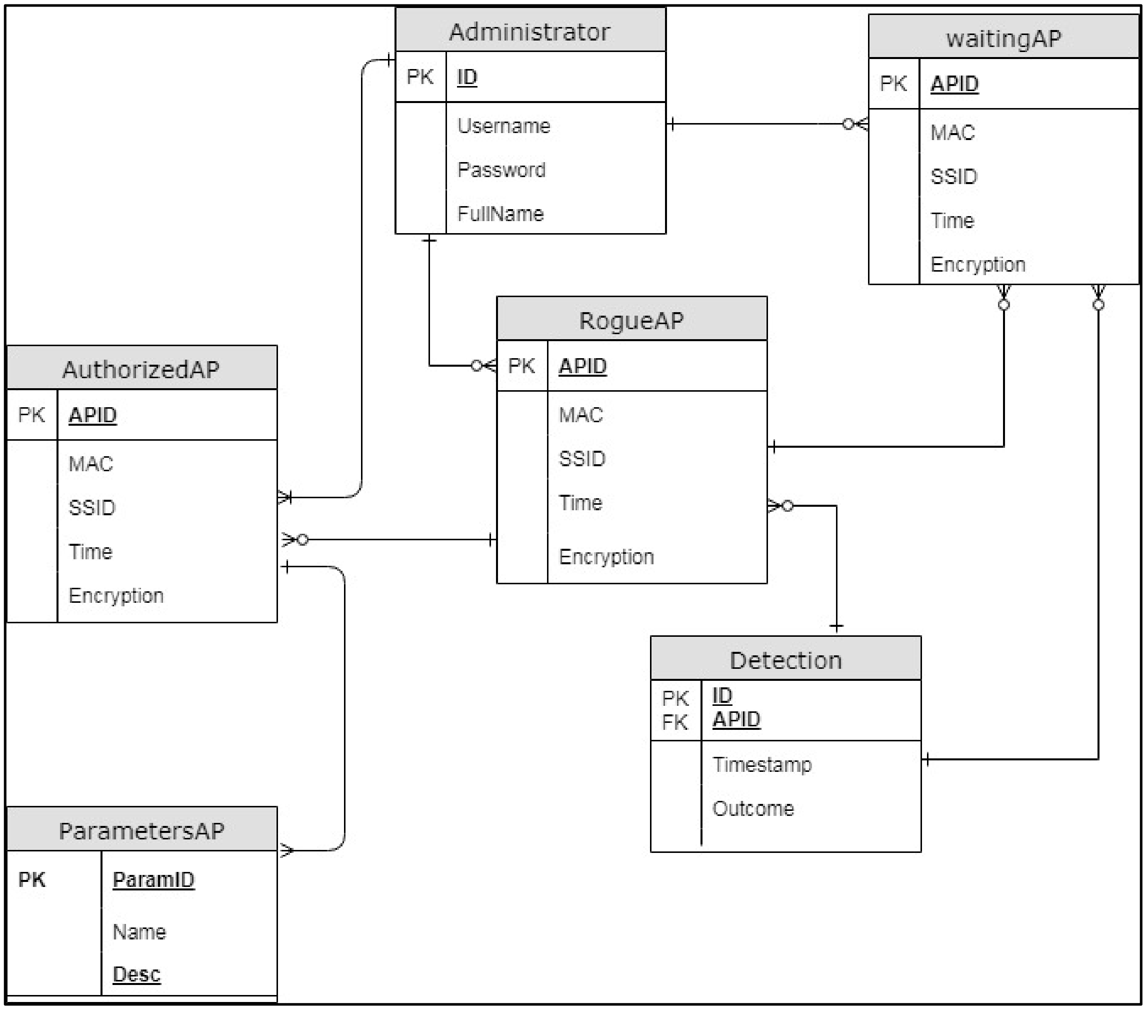


Figure 4.6: Database Schema

The database schema shown in figure 4.6 above shows six tables used for holding data in the rogueAP detection system. The Administrator table contains the details of the network operator and any individuals authorized to access the system to perform functions like approving waiting APs as either rogue or authorized and adding authorized APs to the system. It has attributes such as the users first name, last name and email. There is a separate login table that hold the users' login details such as username and password. The parameters table stores parameters that are common to all APs on the network regardless of the vendor. Authorized APs table will have unique parameters that the network administrator has defined for the network. Any AP whose details do not match those in the authorizedAP table will automatically be classified as rogue and added to rogueAP table. The system will automatically read beacon frames sent onto the network by connect APs and add any detected AP onto waiting list if its parameters matches those of authorized APs; otherwise it will be classified as rogue and added to rogueAP table automatically.

VT OMNES VNVM SINT

CHAPTER FIVE

IMPLEMENTATION AND TESTING

5. 1 Introduction

This chapter details the conversion of the system designs shown in the previous chapter into actual software components and working prototypes. The chapter discussed various model items and the test foundation that the researcher put together at the implementation and testing stages. The chapter further details the actual implementation of the system where the system performs sensing of frames, extraction of parameters, and comparison of the parameters with stored parameters and finally classification of the sensed APs. Capturing of authorized parameters is also shown. Pseudocodes of the system implementation are shown. The chapter also reports on the testing and results that the researcher observed while using the system and other tools such as Wireshark. Screenshots are shown. At the end, the researcher draws conclusions by evaluating some challenges experienced at the implementation and testing stages.

5.2 Application Components

Software and hardware parts used in the research followed the type of functionality that the system was meant to achieve as per the researchers objectives. They are detailed as shown below:

1. **Hardware**

i. A Personal Computer with at least the following properties

  a. 4 GB RAM
  b. 500 GB hard disk storage capacity
  c. Intel duo core processor with 3 GHz speed.
  d. Wi-Fi enabled
  e. NIC that can be in monitor mode

ii. A local area network with WLAN running

iii. Wireless access points

2. **Software**

i. Windows 7 Operating System (development and testing operating system)

ii. Linux (any distribution many need small configuration of the system to match the platform)

iii. Full Python 3.7.3 installed

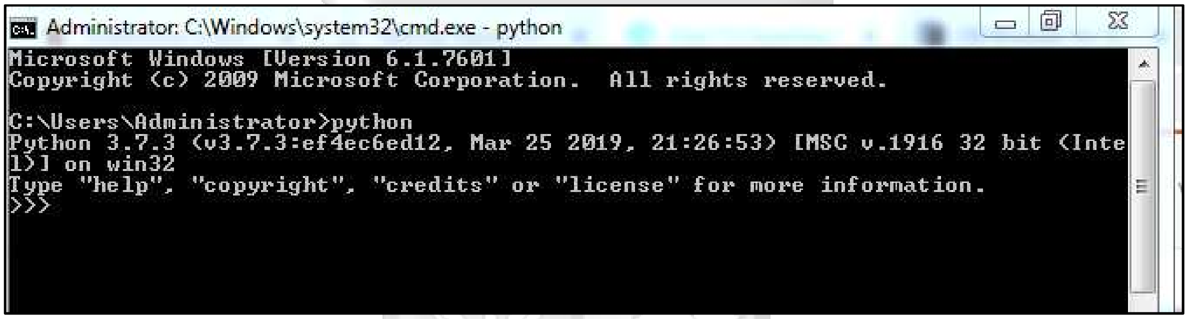
iv. SQLite3 database for storage of data

v. NPcap to facilitate packet capturing

vi. Wireshark version 3.0.1 for packet analysis

5.3 Test Environment

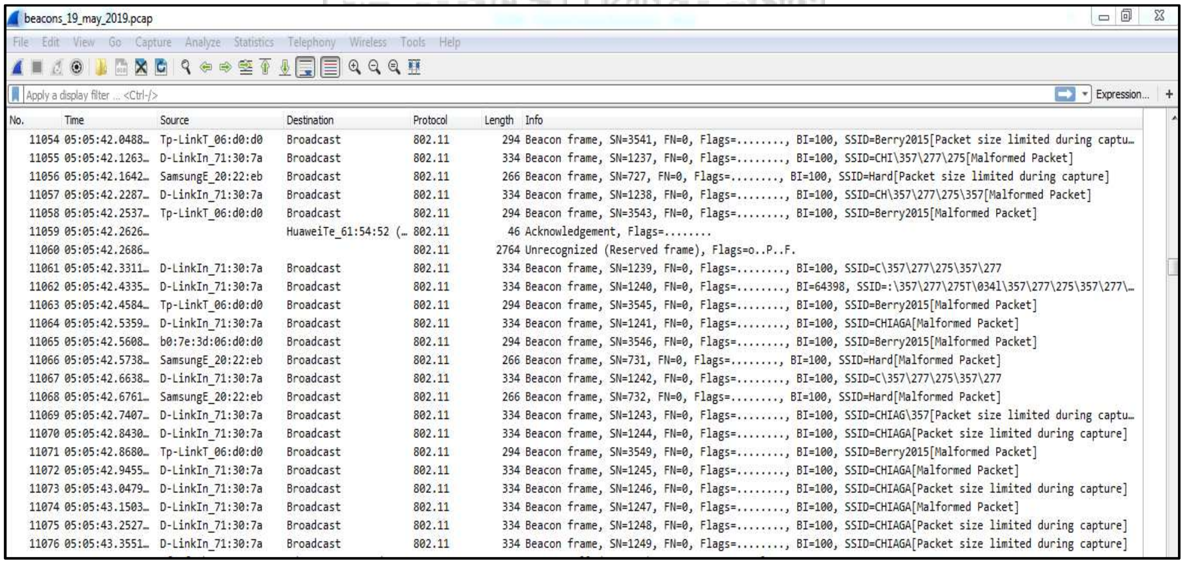
The test environment consisted of a small setup of a WLAN with a PC with two access points connected to the network and a Samsung mobile phone hotspot running. The PC running Window 7 operating system had Python 3.7 installed. Most Linux distributions come with the latest python version preinstalled but it is good to verify that the right version is indeed installed. Figure 5.1 below shows a simple confirmation of python 3.7.3 installation on a Windows 7 computer of 32 bit system.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 5.1: Python version on Windows 7

Using Wireshark, it was possible to see that connected access points are broadcasting beacon frames in search of clients to connect with as shown in figure 5.2 below.



No.	Time	Source	Destination	Protocol	Length	Info
11054	05:05:42.0488…	Tp-LinkT_06:d0:d0	Broadcast	802.11	294	Beacon frame, SN=3541, FN=0, Flags=........, BI=100, SSID=Berry2015[Packet size limited during captu…
11055	05:05:42.1263…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1237, FN=0, Flags=........, BI=100, SSID=CHI\357\277\275[Malformed Packet]
11056	05:05:42.1642…	SamsungE_20:22:eb	Broadcast	802.11	266	Beacon frame, SN=727, FN=0, Flags=........, BI=100, SSID=Hard[Packet size limited during capture]
11057	05:05:42.2287…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1238, FN=0, Flags=........, BI=100, SSID=CH\357\277\275\357[Malformed Packet]
11058	05:05:42.2537…	Tp-LinkT_06:d0:d0	Broadcast	802.11	294	Beacon frame, SN=3543, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed Packet]
11059	05:05:42.2626…		HuaweiTe_61:54:52 (…	802.11	46	Acknowledgement, Flags=........
11060	05:05:42.2686…			802.11	2764	Unrecognized (Reserved frame), Flags=o..P..F.
11061	05:05:42.3311…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1239, FN=0, Flags=........, BI=100, SSID=C\357\277\275\357\277
11062	05:05:42.4335…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1240, FN=0, Flags=........, BI=64398, SSID=:\357\277\275T\034l\357\277\275\357\277\…
11063	05:05:42.4584…	Tp-LinkT_06:d0:d0	Broadcast	802.11	294	Beacon frame, SN=3545, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed Packet]
11064	05:05:42.5359…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1241, FN=0, Flags=........, BI=100, SSID=CHIAGA[Malformed Packet]
11065	05:05:42.5608…	b0:7e:3d:06:d0:d0	Broadcast	802.11	294	Beacon frame, SN=3546, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed Packet]
11066	05:05:42.5738…	SamsungE_20:22:eb	Broadcast	802.11	266	Beacon frame, SN=731, FN=0, Flags=........, BI=100, SSID=Hard[Malformed Packet]
11067	05:05:42.6638…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1242, FN=0, Flags=........, BI=100, SSID=C\357\277\275\357\277
11068	05:05:42.6761…	SamsungE_20:22:eb	Broadcast	802.11	266	Beacon frame, SN=732, FN=0, Flags=........, BI=100, SSID=Hard[Malformed Packet]
11069	05:05:42.7407…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1243, FN=0, Flags=........, BI=100, SSID=CHIAG\357[Packet size limited during captu…
11070	05:05:42.8430…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1244, FN=0, Flags=........, BI=100, SSID=CHIAGA[Packet size limited during capture]
11071	05:05:42.8680…	Tp-LinkT_06:d0:d0	Broadcast	802.11	294	Beacon frame, SN=3549, FN=0, Flags=........, BI=100, SSID=Berry2015[Malformed Packet]
11072	05:05:42.9455…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1245, FN=0, Flags=........, BI=100, SSID=CHIAGA[Malformed Packet]
11073	05:05:43.0479…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1246, FN=0, Flags=........, BI=100, SSID=CHIAGA[Packet size limited during capture]
11074	05:05:43.1503…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1247, FN=0, Flags=........, BI=100, SSID=CHIAGA[Malformed Packet]
11075	05:05:43.2527…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1248, FN=0, Flags=........, BI=100, SSID=CHIAGA[Packet size limited during capture]
11076	05:05:43.3551…	D-LinkIn_71:30:7a	Broadcast	802.11	334	Beacon frame, SN=1249, FN=0, Flags=........, BI=100, SSID=CHIAGA[Packet size limited during capture]

Figure 5.2: Wireshark capture of Beacon frames

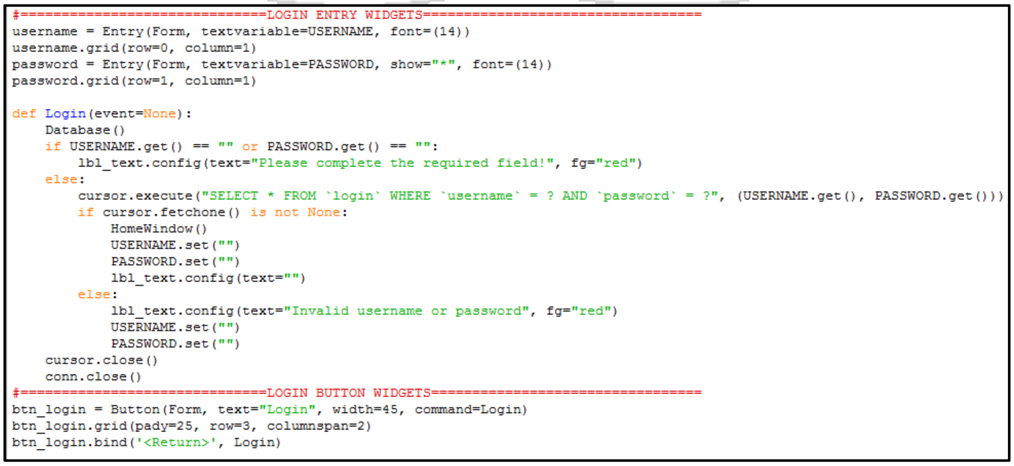
5.4 Application Implementation

The system implementation involved actual writing of python scripts to achieve the detection, design of interfaces to interact with the administrator and SQLite3 relational database tables for storage of data.

The researcher settled on Python as a language because of a number of reasons; Python language has vast large standard support libraries with most programing tasks already bundled into it that shortens the programming duration. It mingles well with other web service tools and directly interacts with C, C++ and Java especially via Jython. It handles XML and other markup languages and is capable of running on current operating systems easily. Pythonn as a programming language highly improves programmers' productivity because of the extensive support libraries and very clean object-oriented designs compared to other languages. Still on productivity, python's strong process integration features, unit testing approach and good control capabilities makes writing codes faster for applications. Finally the researcher found that python is a very good for building scalable multi-protocol network systems and applications.

SQLite3 is a lightweight, easy to configure relational database that can run very fast with this system on any given device including IOT devices in future developments.

The figure 5.3 below show the prototype authentication code for authorizing the network administrator into the system at the beginning. The user requires a username and a password. For demonstration purposes, both the username and password are set to "admin".



```
#==============================LOGIN ENTRY WIDGETS===================================
username = Entry(Form, textvariable=USERNAME, font=(14))
username.grid(row=0, column=1)
password = Entry(Form, textvariable=PASSWORD, show="*", font=(14))
password.grid(row=1, column=1)

def Login(event=None):
    Database()
    if USERNAME.get() == "" or PASSWORD.get() == "":
        lbl_text.config(text="Please complete the required field!", fg="red")
    else:
        cursor.execute("SELECT * FROM `login` WHERE `username` = ? AND `password` = ?", (USERNAME.get(), PASSWORD.get()))
        if cursor.fetchone() is not None:
            HomeWindow()
            USERNAME.set("")
            PASSWORD.set("")
            lbl_text.config(text="")
        else:
            lbl_text.config(text="Invalid username or password", fg="red")
            USERNAME.set("")
            PASSWORD.set("")
    cursor.close()
    conn.close()
#==============================LOGIN BUTTON WIDGETS==================================
btn_login = Button(Form, text="Login", width=45, command=Login)
btn_login.grid(pady=25, row=3, columnspan=2)
btn_login.bind('<Return>', Login)
```

Figure 5.3: Sample of Authentication code

When run, the code yields a login graphical user interface as shown in figure 5.4 below:

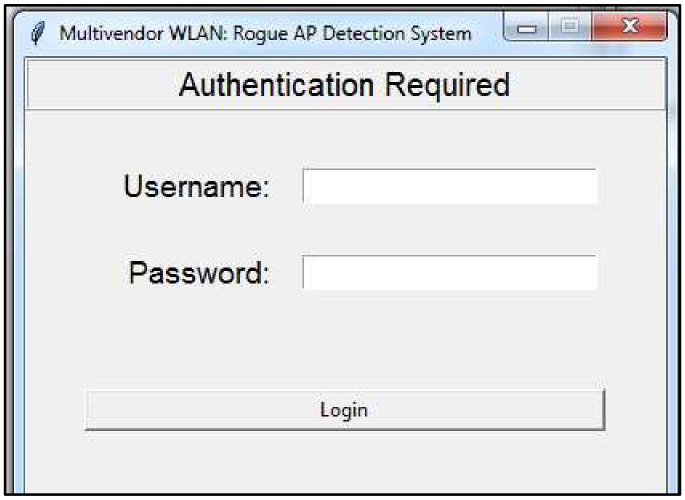
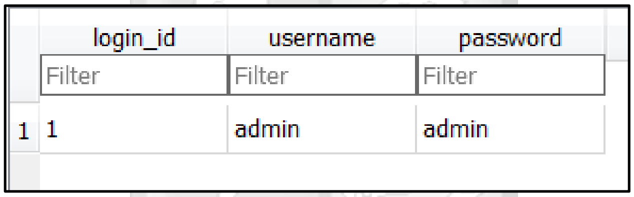


Figure 5.4: Login Interface

The login credentials are retrieved through SQL statement shown in figure 5.3 from an SQLite database table whose structure is a shown in figure 5.5 below:



	login_id	username	password
	Filter	Filter	Filter
1	1	admin	admin

Figure 5.5: Login table

Once a network administrator is authenticated, they are able to add and view access points through various interfaces as shown in the figures displayed below. The administrator can navigate from one interface to another through connecting buttons on the interfaces. The system captures authorized APs in two different way: One the network administrator can manually add access point parameters using the interface shown in figure 5.6 below. Secondly, the system can automatically discover APs connected to the network. It puts them on a waiting list if at least the running SSID and encryption match the authorized parameters. The network administrator can then confirm the auto-discovered APs as either rogue of authorized.

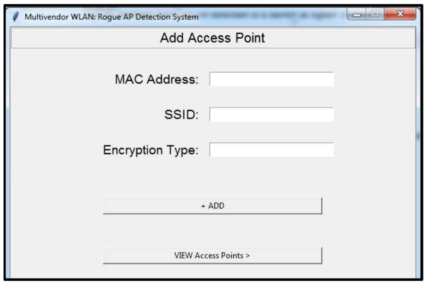
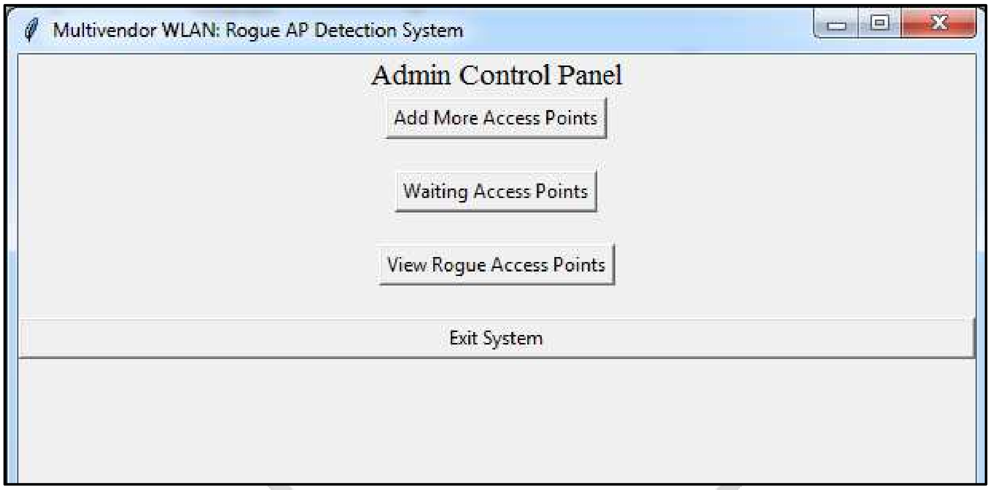


Figure 5.6: Add Access Point interface

Figure 5.7 below shows the main control panel accessed by authenticated users. The user is able to access more system functionalities from this interface.



Multivendor WLAN: Rogue AP Detection System

Admin Control Panel

Figure 5.7: Admin Control Interface

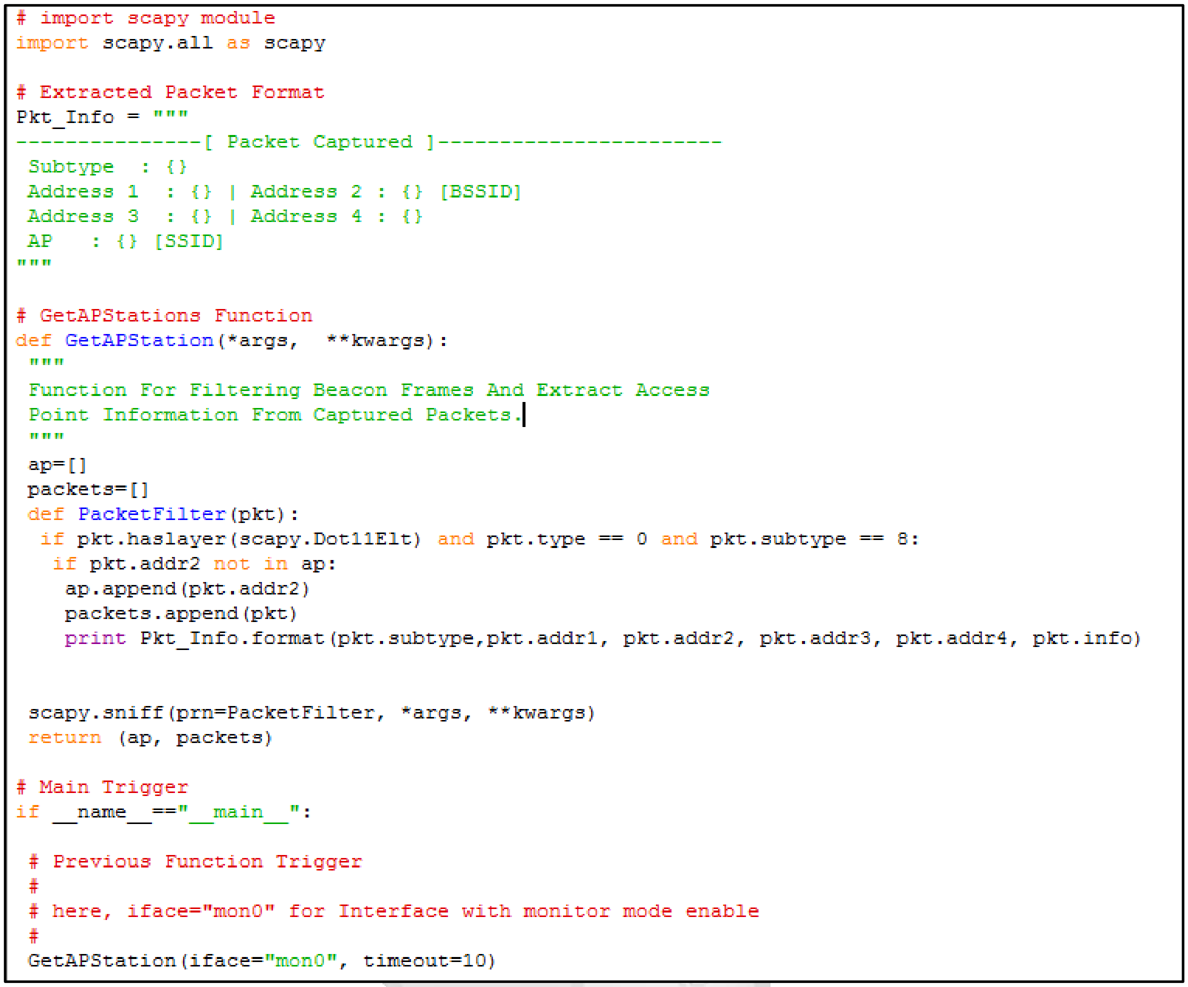
With the help of python's scapy module, the code snippet shown in figure 5.8 below attempts to capture beacon frames available and attempts to extract and decode the frames to discover accessible access points. This code is borrowed from Singh (2017) who attempted to create a Wi-Fi packet sniffer. Beacon frames carry the access points SSID and BSSID (MAC address) that we require together with encryption set to compare with authorized parameters to classify rogue access points from genuine ones. This piece of code is still work in progress and under development and as such one may experience bugs when running it. The researcher invites interested people to refine and develop it further as may be needed.

VT OMNES VNVM SINT

5.5 Application Testing

Authorization input test to sanitize login input is as shown in figure 5.9 below. The system does not accept empty string logins. A user must supply a username and a password to access it.

Figure 5.9 on the next page shows an interface displaying an error when a user attempts to login with empty values.

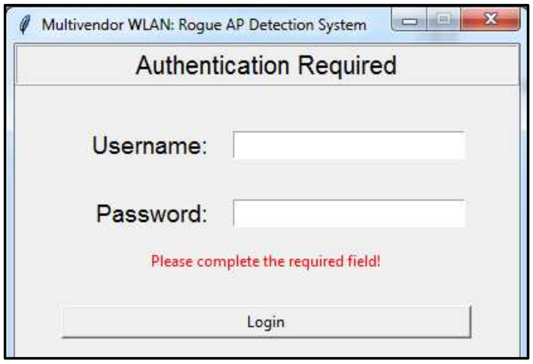


```
# import scapy module
import scapy.all as scapy

# Extracted Packet Format
Pkt_Info = """
---------------[ Packet Captured ]-----------------------
 Subtype  : {}
 Address 1  : {} | Address 2 : {} [BSSID]
 Address 3  : {} | Address 4 : {}
 AP   : {} [SSID]
"""

# GetAPStations Function
def GetAPStation(*args,  **kwargs):
 """
 Function For Filtering Beacon Frames And Extract Access
 Point Information From Captured Packets.
 """
 ap=[]
 packets=[]
 def PacketFilter(pkt):
  if pkt.haslayer(scapy.Dot11Elt) and pkt.type == 0 and pkt.subtype == 8:
   if pkt.addr2 not in ap:
    ap.append(pkt.addr2)
    packets.append(pkt)
    print Pkt_Info.format(pkt.subtype,pkt.addr1, pkt.addr2, pkt.addr3, pkt.addr4, pkt.info)


 scapy.sniff(prn=PacketFilter, *args, **kwargs)
 return (ap, packets)

# Main Trigger
if __name__=="__main__":

 # Previous Function Trigger
 #
 # here, iface="mon0" for Interface with monitor mode enable
 #
 GetAPStation(iface="mon0", timeout=10)
```

Figure 5.8: Extraction of IEEE 802.11 Beacon Frames

Source : Singh (2017)

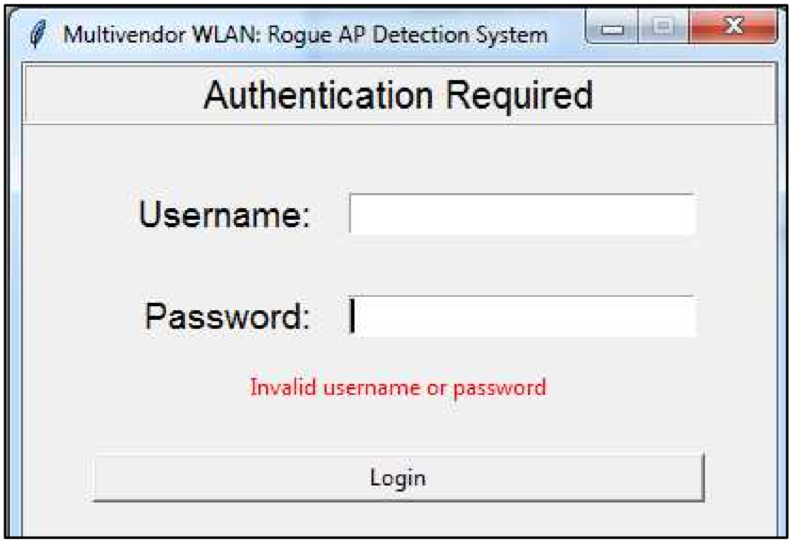
VT OMNES VNVM SINT



Please complete the required field!

Figure 5.9: Login Input Testing

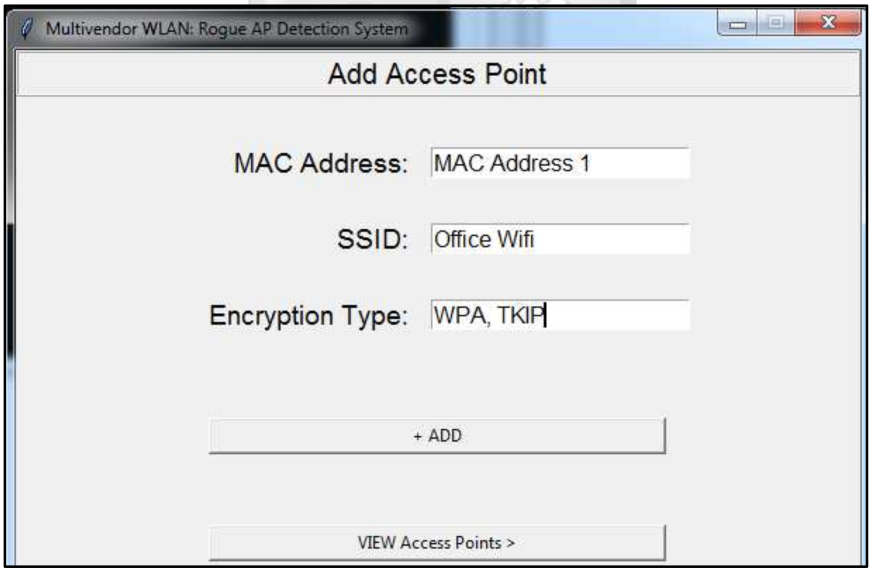
The system will deny access if the supplied username and password are wrong as shown in figure 5.10 below.



Invalid username or password

Figure 5.10: Invalid Login

Authenticated users can enter AP details and add as shown in figure 5.11 below.



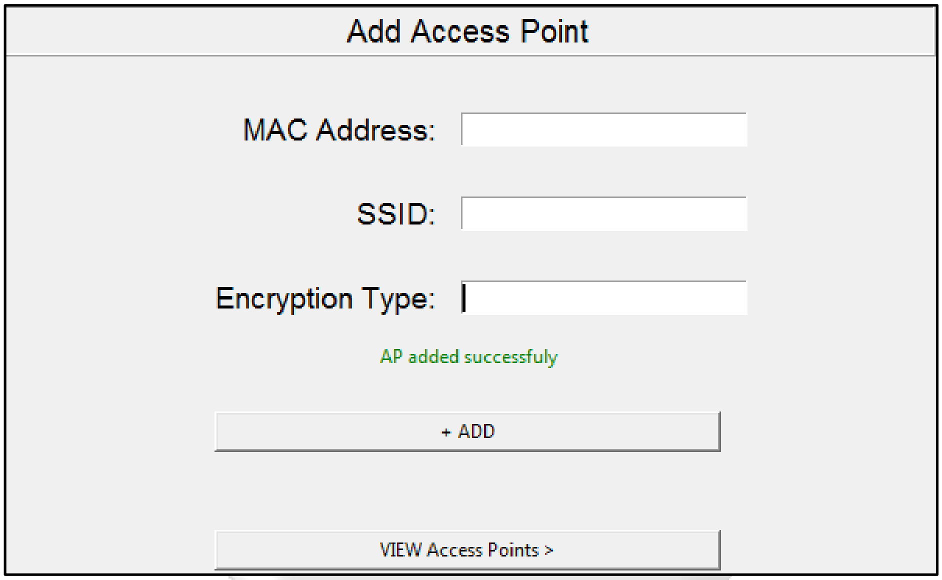
MAC Address: MAC Address 1

SSID: Office Wifi

Encryption Type: WPA, TKIP

Figure 5.11: Add Access Point

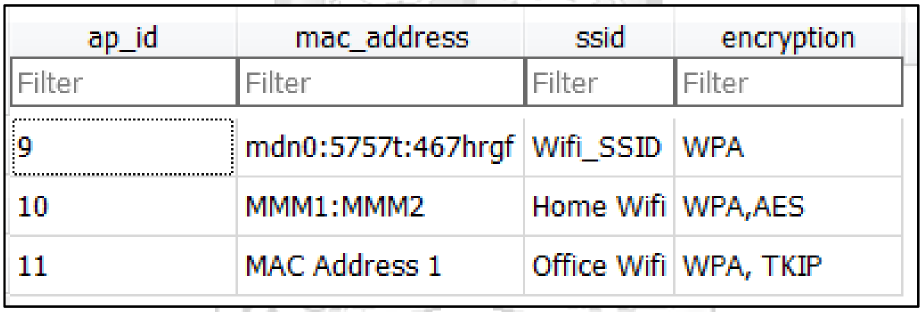
On successful addition of access point, the system notifies the user with a confirmation of successful addiction as indicated in figure 5.12 below.



AP added successfuly

Figure 5.12: Successful AP Addition

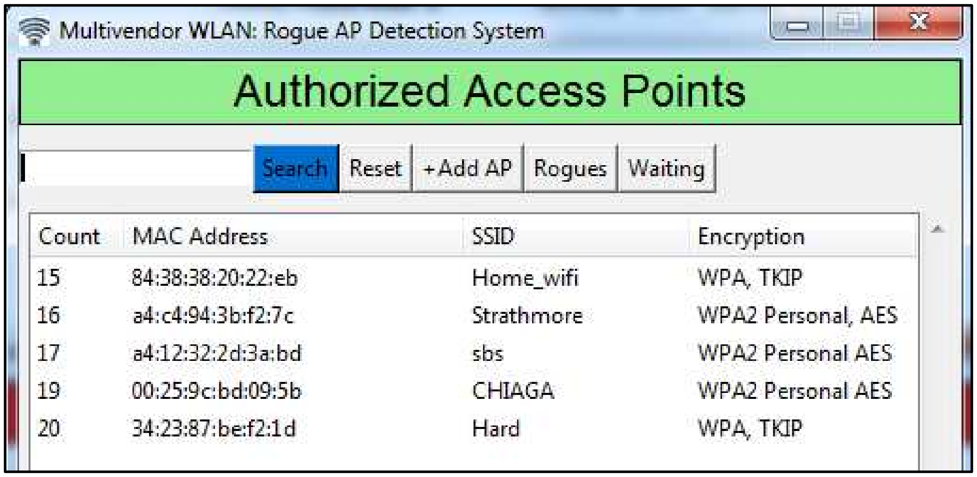
Successfully added access points are stored in a database table as a new record with a unique access point number as shown in the figure 5.13 below:



ap_id	mac_address	ssid	encryption
Filter	Filter	Filter	Filter
9	mdn0:5757t:467hrgf	Wifi_SSID	WPA
10	MMM1:MMM2	Home Wifi	WPA,AES
11	MAC Address 1	Office Wifi	WPA, TKIP

Figure 5.13: Added AP Record

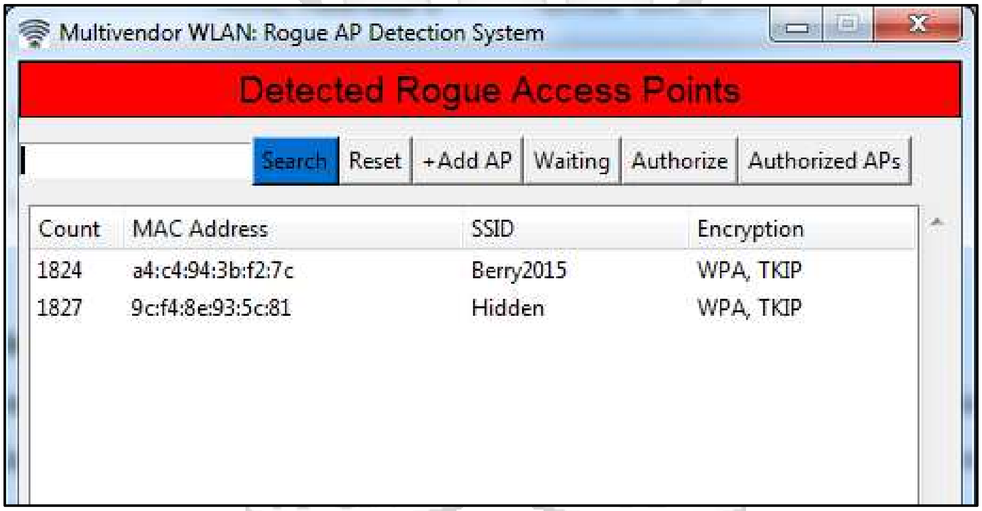
The system is able to show a list of authorized APs as shown in figure 5.14 below.



Count	MAC Address	SSID	Encryption
15	84:38:38:20:22:eb	Home_wifi	WPA, TKIP
16	a4:c4:94:3b:f2:7c	Strathmore	WPA2 Personal, AES
17	a4:12:32:2d:3a:bd	sbs	WPA2 Personal AES
19	00:25:9c:bd:09:5b	CHIAGA	WPA2 Personal AES
20	34:23:87:be:f2:1d	Hard	WPA, TKIP

Figure 5.14: Listing authorized APs

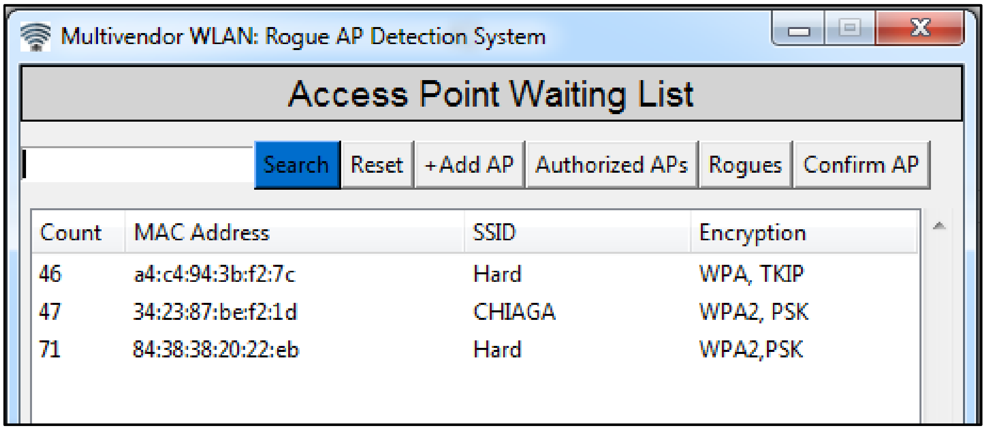
Rogue APs are listed as shown in figure 5.15 below.


Multivendor WLAN: Rogue AP Detection System
Detected Rogue Access Points
Search
Reset
+Add AP
Waiting
Authorize
Authorized APs

Count	MAC Address	SSID	Encryption
1824	a4:c4:94:3b:f2:7c	Berry2015	WPA, TKIP
1827	9c:f4:8e:93:5c:81	Hidden	WPA, TKIP

Figure 5.15: Listing rogue APs

Figure 5.16 below shows a listing of APs on a waiting list. The network administrator is able to confirm APs from this interface as authorized.


Multivendor WLAN: Rogue AP Detection System
Access Point Waiting List
Search
Reset
+Add AP
Authorized APs
Rogues
Confirm AP

Count	MAC Address	SSID	Encryption
46	a4:c4:94:3b:f2:7c	Hard	WPA, TKIP
47	34:23:87:be:f2:1d	CHIAGA	WPA2, PSK
71	84:38:38:20:22:eb	Hard	WPA2,PSK

Figure 5.16: Listing APs on Waiting List

5.6 Application Testing Results

The researcher performed various tests on the system. The tests are detailed in this section which explains the activities done as part of the testing of 'Multi-vendor WLAN rogue AP Detection system' application.

i) **In Scope Testing**

A functional testing of the following modules fall under the category of scope of testing

Authentication to ensure only authorized users can access and use the system

Adding of authorized AP parameters to the database

Capturing of Beacon frames and extraction of parameters

Storage of AP data in SQLite Database and Database Connectivity

ii) Items not tested

Running of the system on a multiple platforms was not tested by the time this document was written. This is mainly because the system is still undergoing modification and time constraints relating to submission of the document could not allow the researcher to fully implement. Once the system is complete, it will be packaged and tested on different operating systems.

iii) Metrics

Table 5.1: System Testing Clases

Test Class	**Inspection point**	**Importance**
Functional	Does the system allow authenticated users only to access it?	High
Functional	Does the system sniff beacon frame from the network and extract SSID, MAC and encryption parameters?	High
Functional	Can the user add authorized access points to the system?	High
Functional	Does the system keep records of access points and their parameters securely?	High
Functional	Is the system able to compare stored parameters with beacon frame extracted parameters to determine rogue access points?	High
Non-functional	Does the system operate well on multiple platforms	Low
Non-functional	Did the developer adhere to software standards shown by the system implementation?	Low

Table 5.1 above shows the various test classes performed, the testing criteria and their level of importance. Table 5.2 on the next page shows the test results of two main test classes namely functional and non-functional. It also shows the results obtained and the author's comments on each test class.

UT OMNES UNUM SINT

5.7 Challenges Faced During Implementation

The researcher faced some challenges while attempting to fulfil all the research objectives. These challenges affected the functionality of the system. Limited time of working on the research and learning the new technology and implementing it as explained under testing section above was the main challenge. Complexity of understanding the technology was also a challenge to the researcher.

Table 5.2: System Testing

Test Class	**Test Result**	**Comments**
Functional	Pass	Authorized users were able to access the system and add more access points. The system was able to keep a record of the access points in a SQLite database. The system was able to sniff beacon frames and could compare sniffed parameters and authorized parameters to classify an access point as rogue, authorized or waiting.
Non-Functional	Pass	The developer adhered to software standards. Python is a multiplatform programming language and as such the system will run on different platforms when it is finally packaged. The system implements necessary security controls.

5.7.1 Complexity

Python language is easier to learn but using it to capture network packets and extract parameters was complex to the researcher. It required reconfiguration of network interface cards into a sniffing mode which was complex to implement on Windows 7 operating system that was mainly used during the research. This challenge will be overcome as the research develops the system further. Working with some python modules such as scapy on Windows environment is complex as it requires reconfiguration of some components of the operating system such as transforming a Wi-Fi interface into monitor mode.

Windows operating system does not support network interface card monitor mode by default and as such it took the researcher some time to figure out and install Microsoft Network Monitor tool to make the monitor mode setting possible.

VT OMNES VNVM SINT

CHAPTER SIX

DISCUSSION

6.1 Introduction

This chapter expounds the test results realized in the previous chapter by focusing the results into goal of the research. The chapter evaluates the properties of access points on a multivendor WLAN and how the python script was able to compare the parameters to determine rogue access points on the network. The chapter also gives more light on the test results that were obtained during the research.

6.2 Important Parameters of APs on a Multivendor WLAN

6.2.1 MAC Address

This is a unique identifying number assigned to a network interface card of APs by the vendors. Single vendor WLAN controllers assume that an AP whose MAC address does not match the vendors MAC address is rogue by default. This does not give room for APs from other vendors to operate on the WLAN. Each AP on the multivendor WLAN has its own unique MAC address specified by its vendor that it broadcasts through beacon frames as BSSID. The system developed from this research allows APs from different vendors to work on a WLAN without being listed as rogues based on their MAC address only. The MAC addresses of all the authorized APs are securely stored in a database. Rogues APs can easily be noticed because their MAC addresses are not recognized on the system. It is important to know that advanced hackers can actually clown a MAC of an authorized AP and assign it to their rogue AP. In this case, the rogue AP may be able to connect on the WLAN undetected unless the system checks the SSID and encryption parameters discussed below. If all parameters of the sniffed APs match those of a genuine AP, the system puts the sniffed AP on a list of waiting APs for further approval by the network administrator. This further prevents MAC address spoofing that is common with commercial WLAN controllers.

UT OMNES UNUM SINT

6.2.2 WLAN SSID

Service set identifier (SSID) is the wireless name set by the network administrator. It is usually broadcasted by APs for clients to connect. It is the easiest for hackers to clown and therefore it must be used together with MAC, and encryption standards of AP to detect rogue APs on the network. It is one of the parameters captured when the system sniff beacon frames. Any access point whose SSID does not match the approved SSID is automatically listed as rogue. If the SSID matches the approved SSID, the system further checks the MAC address and encryptions set on the AP. This prevents SSID masquerade attack that is common with commercial single vendor WLAN controllers.

6.2.3 Encryption Standards

The network administrators determine which Wi-Fi encryption standard they set for their WLANs. All the APs on a multivendor WLAN must use authorized encryption standards. Encryption standards include WEP, WPA and WPA2. Encryption standards insure that information communicated on a WLAN is not easily readable by unintended recipients. Table below gives brief description of the common wireless encryption standards in use.

Table 6.1: Encryption standards

Encryption standard	**Description**	**How it works**
Wired Equivalent Privacy (WEP)	It was the first 802.11 security standard. It is easier to hack due to its use of 24-bit initialization vector and weak authentication.	It uses RC4 stream cipher and 64 or 128 – bit keys. Static master key must be manually entered into each device
Wi-Fi Protected Access (WPA)	Was developed to address WEP weaknesses. It is backwards compatible with WEP devices. It operates on personal or enterprise modes	It uses RC4 cipher but longer initialization vectors and 256-bit keys. Each client uses new key with TKIP. It operates on enterprise mode and uses stronger authentication via 802.1x and EAP
WPA 2	It is a current standard that works well with modern hardware without affecting hardware operating efficiency. It uses both personal and enterprise modes	It uses CCMP and AES as opposed to RC4 and TKIP algorithms or advanced authentication and encryption

VT OMNES VNVM SINT

An access point whose encryption standard does not match authorized standards in the system is automatically listed as rogue. An access point with no encryption standard specified is also listed as rogue automatically.

6.2.4 AP Unique Number on Multivendor WLAN

Authorized Access points on the multivendor WLAN are each assigned a unique random number automatically by the system to identify them on the network. This number together with other parameters will are used to distinguish rogue APs from authorized APs by the system. This number does not come from beacon frames, rather it is assigned to an AP by the system.

6.3 Classification of APs on a Multivendor WLAN

Commercial WLAN controllers use a combination of different internal heuristics, AP classification rules and manual classification by the user to differentiate rogue APs from genuine APs. Their main disadvantage is that if an AP does not bear a MAC address from the same vendor as the controller, the AP is automatically listed as rogue. This is the inflexibility that network administrators want to overcome for them to realize benefits of deploying multivendor APs as discussed in chapter one of this research. Multivendor WLAN have APs with MAC addresses from different vendors. The system developed from this research is able to work with APs from different vendors. An AP is classified s authorized or rogue following a combination of MAC address, SSID and encryption that it broadcasts.

VT OMNES VNVM SINT

6.3.1 Authorized AP

These are access points which are allowed to operate on the network. They are registered in the system and known to the network administrators. They carry authorized MAC address, SSID and encryption parameters specified and accepted by the network administrators on the multivendor AP WLAN. They can be added to the system manually by the network administrator or automatically discovered by the system during sniffing.

6.3.2 Rogue AP

These are access points which are not allowed to operate on the network. They are not registered in the system and are unknown to the network administrators. They lack one or all the authorized parameters specified and acceptable by the network administrators on the multivendor AP WLAN. They are a big threat on the network as they can be leveraged by malicious people to launch attacks on the network. These type of APs should be detected and eliminated from the network as soon as they appear.

6.3.3 Waiting AP

Waiting access points are those which the system has automatically detected on the network from the extracted beacon frames and the one or two parameters matching the authorized parameters. The system cannot immediately classify them as rogue or authorized. They are kept on a waiting list until the network administrator confirms them as authorized or rogues. If approved, they become authorized otherwise they are put in the category of rogue APs.

6.4 Rogue AP Detection on a Multivendor WLAN

The researcher established the MAC address, SSID and encryption are the important parameters required to detect rogue APs on a multivendor AP WLAN. The system performs rogues detection in three straight forward steps as long as all the parameters are supplied. The system will first sniff beacon frames from the network and then extract MAC address, SSIDs and encryption supplied by connected access points. Secondly, the system compares the extracted parameters with the authorized parameters. Third stage is classification of the APs. APs whose all parameters match the authorized parameters are added to the list of authorized APs. APs whose one or two parameters match authorized parameters are added to a waiting list for further approval by the network administrator. APs whose all parameters do not match any of the authorized parameters are automatically added to the list of rogues APs. The MAC addresses of APs operating on the WLAN do not have to be from a single vendor unlike in a single vendor AP WLAN. Multiple SSIDs can also be configured to run on the network as long as they are listed as authorized in the system.

CHAPTER SEVEN

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusion

Wi-Fi usage is gaining popularity in homes, offices and public spaces. One of the major concerns that comes with Wi-Fi is security. The presence of rogue access points (APs) on the wireless networks poses a major security threat as hackers can leverage the rogue APs to launch multiple attacks on the network. Probability of rogues APs occurring on a WLAN is amplified if the network has APs from different vendors. Network administrators are increasingly looking into rolling out multi-vendor AP WLANs to achieve flexibility and save on costs. With the growth of BYOD, unsuspecting employees in a company can also introduce rogue APs to a secure wired network. The problem is amplified if the wireless local area network (WLAN) consist of multi-vendor APs. Malicious people can leverage on rogue APs to perform passive or active attacks on a computer network. Therefore, the researcher saw the need for network administrators to accurately, with less effort, detect and control presence of rogue APs on multivendor WLANs.

There exists different solutions to detect and control rogue APs on a WLAN but most of them only support a single vendor APs WLAN or require extra hardware resources or modification of existing AP firmware. In this research, a model that supports detection of rogues APs on a multi-vendor AP WLAN without adding extra hardware or modification of AP firmware has been developed. The research took an experimental research design approach whereby the researcher performed experiments with different access points in a computer laboratory in the research location to help in establishing important parameters pertinent to the study. A working prototype of the proposed model was developed using a structured waterfall approach and using Python programming language. The developed system was tested a setup WLAN. I was able to read and interpret beacon frames from connected APs to categorize them as either authorized, rogue or waiting based on parameters discussed in the study. The system issues alerts that describe the detected APs to the network administrator for further action. The research objectives were therefore met. The developed system is easier to install and configure to work on a LAN. It uses graphical user interfaces that enable the network administrator to interact with it easily.

7.2 Recommendations

Wireless usage in homes, offices and public spaces is growing and will continue to grow in the near future owing to the increase of wireless enable mobile devices in the market. Administrators of WLANs will continuously have headaches trying to detect rogue APs on their WLANs. The following recommendations are made regarding the research and the model:

i) The system should be used at a point on the network where it is possible and easier to access traffic from all the devices on the network. This will increase chances of all rogue access points being detected.

ii) Authorized AP parameters should be well pre-specified into the system. If the network administrator does the registration manually, care should be observed to ensure that the parameters are captured correctly. This will enhance detection of rogue APs and prevent occurrence of false positive alerts.

iii) NPcap or its equivalent should be installed on the computer where the system will run if the operating system is Window. A Linux equivalent should be installed if the developed system is to be run on a Linux environment.

7.3 Suggestions for Future Research

The researcher encourages other researchers to expound on the ideas expressed in this research and the rogue AP detection model developed to refine the ideas and the model. Future research should consider extending this model to include automatic elimination of the detected rogue APs on multivendor WLANs. They should consider implementing the system in other low-level languages such as C and C++. Researchers should advance on ideas from this research to study and develop models of managing multivendor devices in enterprises that adopt the BYOD concept. BYOD results in multivendor devices appearing on the network. All other areas and ideas that are in line with detection of rogue APs on multivendor WLAN that this research may have omitted should be addressed by future research done in the same or related area of research.

7.4 Contributions

This research is a continuation of research focusing wireless network security. It expounds more on rogue AP detection models and considers a new perspective (multivendor AP WLANs) which many past researchers have not explored. Many researchers are actively studying better ways of detecting and controlling rogue access points on wireless LANs. They may find this research useful especially if they are studying multivendor network environments and BYOD.

VT OMNES VNVM SINT

REFERENCES

Amiel, F., Villegas, K., Feix, B., & Marcel, L. (2007). Passive and Active Combined Attacks: Combining Fault Attacks and Side Channel Analysis. *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2007).* Vienna, Austria: IEEE. doi:10.1109/FDTC.2007.12

B, Y., G, C., J, W., & H, Y. (2009). Robust detection of unauthorized wireless access points. *Springer, Mobile Network Applications*, 508-522.

Bahl, P., Chandra, R., Padhye, J., Ravindranath, L., Singh, M., Wolman, A., & Zill, B. (2006). Enhancing the Security of Corporate Wi-Fi Networks using DAIR. *MobiSys.* Uppsala, Sweden: Microsoft Research.

Chirumamilla, M. K., & Ramamurthy, B. (2003). Agent Based Intrusion Detection and Response System for Wireless LANs. *ICC.* Anchorage Alaska, USA: IEEE.

Chirumamilla, M. K., & Ramamurthy, B. (2003). Agent Based Intrusion Detection and Response System for Wireless LANs. *IEEE International Conference on Communications* (pp. 492-496). Lincoln, NE, 68588-0115 U.S.A.: IEEE.

Chumchu, P., Saelim, T., & Sriklauy, C. (2011). A new MAC Address Spoofing Detection Algorithm using PLCP Header. *ICOIN 2011.* Kuala Lumpur, Malaysia.

Corbett, C., Beyah, R., & Copeland, J. (2006). A Passive Approach to Wireless NIC Identification. *International Conference on Communications.* Istanbul, Turkey: IEEE.

Economist. (2004, June). *A brief history of WiFi.* Retrieved from Economist: https://www.economist.com/node/2724397

Gast, G. M. (2005). *802.11 Wireless Networks: The Definitive Guide* (2nd ed.). (M. Loukides, & T. Collen, Eds.) Sebastopol, Carlifornia, USA: O'Reilly Media Inc.

Gopinath, K. N., & Hemant, C. (2009). *All You Wanted to Know About Wifi Rogue Access Points.* Retrieved from AirTight Networks Inc.: www.AirTightNetworks.com

Han, H., Sheng, B., Tan, C. C., Li, Q., & Lu, S. (2009). A Measurement Based Rogue AP Detection Scheme. *IEEE INFOCOM 2009 proceedings* (pp. 1-9). IEEE Communications Society.

Han, H., Sheng, B., Tan, C. C., Li, Q., & Lu, S. (2009). A Measurement Based Rogue AP Detection Scheme. *IEEE INFOCOM 2009 proceedings* (pp. 1592-1601). Williamsburg, VA USA: IEEE Communications Society.

Heale, R., & Twycross, A. (2015, January). Validity and reliability in quantitative studies. *Evidence-Based Nursing, 18*(3), 66-67.

Internet World Stats. (2017, December 31). *Internet World Stats Usage and Population Statistics.* Retrieved from Internet World Stats: https://www.internetworldstats.com/stats.htm

Jana, S., & Kasera, S. K. (2008). On fast and accurate detection of unauthorized wireless access points using clock skews. *14th ACM international conference on Mobile computing and networking, MobiCom 08* (p. 104115). New York, NY, USA: ACM.

Kagan, A. (2003, November 7). *How Things Work: WLAN Technologies and Security Mechanisms.* Retrieved April 2019, from SANS Website: https://www.sans.org/reading-room/whitepapers/wireless/things-work-wlan-technologiessecurity-%20mechanisms-1301

Kangsuk, C., Jiawei, S., Souhwan, J., Changmoon, H., Seongsoo, B., & Injang, J. (2012). A Scheme of Detection and Prevention Rogue AP using Comparison Security Condition of AP. In S. Seth (Ed.), *Proc. of the Intl. Conf. on Advances in Computer Science and Electronics Engineering* (pp. 306-306). Seoul, Korea: Universal Association of Computer and Electronics Engineers. doi:10.3850/978-981-07-1403-1 647

Kaoa, K. F., Liaob, I. E., & Lib, Y. C. (2009). Detecting Rogue Access Points Using Client-side Bottleneck Bandwidth Analysis. *Computers & Security, 28*(3-4), 144-152.

Kohno, T., Broido, A., & Claffy, K. (2005). Remote Physical Device Fingerprinting. *Transactions on Dependable Secure Computing, II*(2), 93-108.

Kung, D., & Lei, J. (2016). An Object-Oriented Analysis and Design Environment. *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET).* Dallas, TX, USA: IEEE. doi:10.1109/CSEET.2016.20

Lambert, A., McQuire, S., & Papastergiadis, N. (2013). *Assets.* (The University of Melbourne) Retrieved from Networkedsociety.unimelb.edu.au: https://networkedsociety.unimelb.edu.au/__data/assets/pdf_file/0007/1661317/Free-Wi-Fi-and-Public-Space.pdf

McGeehan, P. (2016, September). *Free Wi-Fi Kiosks Were to Aid New Yorkers. An Unsavory Side Has Spurred a Retreat.* Retrieved from The New York Times: https://www.nytimes.com/2016/09/15/nyregion/internet-browsers-to-be-disabled-on-new-yorks-free-wi-fi-kiosks.html

Mohd, A. O., Abdullah, Z. T., Zainal, A. S., Tan, S. Y., & Abdullah, S. A. (2012). A Study of the Trend of Smartphone and its Usage Behavior in Malaysia. *International Journal on New Computer Architectures and Their Applications, 2*(1), 275-286.

Monica, D., & Ribeiro, C. (2011). WiFiHop - Mitigating the Evil Twin Attack through Multi-hop Detection. *ESORICS 2011.* Leuven, Belgium.

Nikbakhsh, S., Azizah, B. M., Zamani, M., & Janbeglou, M. (2012). A Novel Approach for Rogue Access Point Detection on the Client-Side. *International Conference on*

Advanced Information Networking and Applications Workshops. Fukuoka, Japan: IEEE. doi:10.1109/WAINA.2012.108

Nikbakhsh, S., Manaf, A., Zamani, M., & Janbeglou, M. (2012). A Novel approach for rogue access point detection on the client side. *International conference on Advanced Information Networking and Applications workshops.*

P, B., R, C., J, P., L, R., M, S., A, W., & B, Z. (2006). Enhancing the Security of Corporate Wi-Fi Networks Using DAIR. *MobiSys.* Uppsala, Sweden.

Pesce, L. (2006). *Discovering Rogue Wireless Access Points Using Kismet and Disposable Hardware.* Washington, DC: SANS Institute.

Philbeck, I. (2017). *Documents.* Retrieved from Broadbandcommission: http://broadbandcommission.org/Documents/ITU_discussion-paper_Davos2017.pdf

Roth, V., Rieffel, E. G., Polak, W., & Turner, T. (2008). Simple and effective defense against evil twin access points. *First ACM Conference on Wireless Network Security, WISEC 2008.* Alexandria, VA, USA: Researchgate. doi:10.1145/1352533.1352569

Saracco, R. (2018, January 26). *Do you still remember the Waterfall Model?* Retrieved from sites.ieee.org: http://sites.ieee.org/futuredirections/2018/01/26/do-you-still-remember-the-waterfall-model/

Saracco, R. (2018, January 26). *Do you still remember the Waterfall Model?* Retrieved from IEEE Website: http://sites.ieee.org/futuredirections/2018/01/26/do-you-still-remember-the-waterfall-model/

Singh, S. (2017, June 28). *How to create wifi ssid finder using python and scapy*. Retrieved from BITFORESTINFO: http://www.bitforestinfo.com/2017/06/how-to-create-wifi-ssid-finder-using-python-and-scapy.html

Song, Y., Yang, C., & Gu, G. (2010). Who Is Peeping at Your Passwords at Starbucks? - To Catch an Evil Twin Access Point. *IEEE/IFIP DSN 2010.* Chicago, IL, USA: IEEE.

Sriram, S., Sahoo, G., & Agrawal, K. (2010). Detecting and eliminating Rogue Access Points in IEEE-802.11 WLAN - a multi-agent sourcing Methodology. *2010 IEEE 2nd International Advance Computing Conference (IACC)* (pp. 256-260). IEEE.

Sriram, V. S., Sahoo, G., & Agrawal, K. K. (2010). Detecting and Eliminating Rogue Access Points in IEEE-802.11 WLAN - A Multi-Agent Sourcing Methodology. *IACC.* Patiala, India: IEEE.

Srisalak, S., K, W., & A, P. (2009). Integrated Wireless Rogue Access Point Detection and Counterattack System. *International Conference on Information Security and Assurance.*

Tektronix. (2015). *Wi-Fi.* Retrieved from Nortelcoelectronics: www.nortelcoelectronics.se

VT OMNES VNVM SINT

University of Southern California. (2010). *Research Guide*. Retrieved from USC Libraries website: http://libguides.usc.edu/writingguide/researchdesigns

V, R., W, P., E, R., & T, T. (2014). Simple and effective defense against Evil twin access points. *WiSec08.* Alexandria, Virginia,USA.

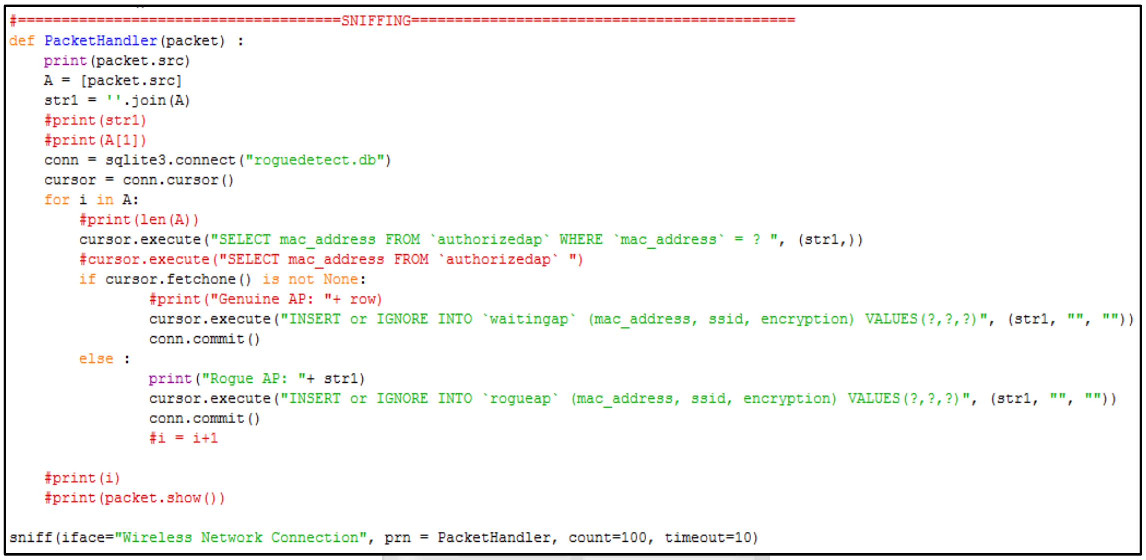
Vanjale, S., & Mane, P. B. (2014). A Novel approach for Elimination of Rogue Access Point in Wireless Network. *2014 Annual IEEE India Conference (INDICON)* (pp. 1-4). Pune: IEEE.

Wei, W., K, S., B, W., J, K., & D, T. (2007). Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs. *7th ACM SIGCOMM conference on Internet measurement.*

VT OMNES VNVM SINT

APPENDIX A

Code Snippets



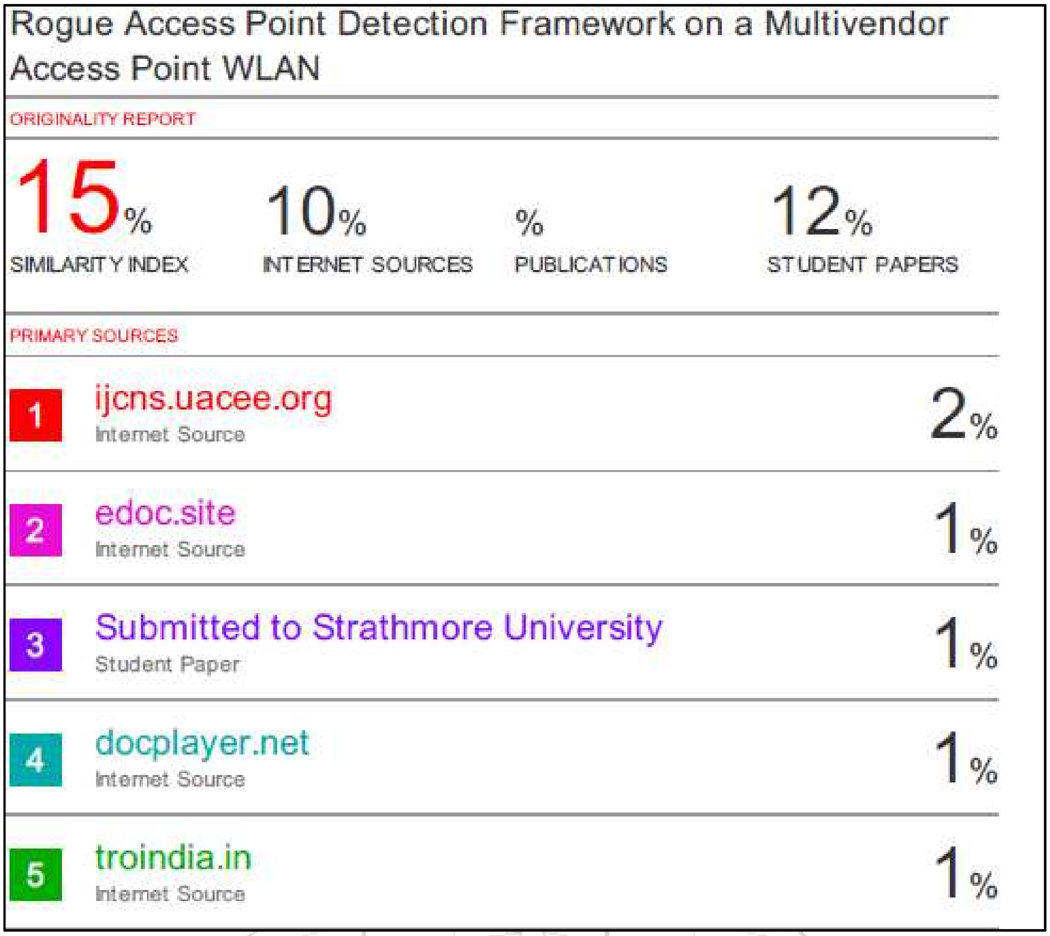
```
#=====================================SNIFFING============================================
def PacketHandler(packet) :
    print(packet.src)
    A = [packet.src]
    str1 = ''.join(A)
    #print(str1)
    #print(A[1])
    conn = sqlite3.connect("roguedetect.db")
    cursor = conn.cursor()
    for i in A:
        #print(len(A))
        cursor.execute("SELECT mac_address FROM `authorizedap` WHERE `mac_address` = ? ", (str1,))
        #cursor.execute("SELECT mac_address FROM `authorizedap` ")
        if cursor.fetchone() is not None:
                #print("Genuine AP: "+ row)
                cursor.execute("INSERT or IGNORE INTO `waitingap` (mac_address, ssid, encryption) VALUES(?,?,?)", (str1, "", ""))
                conn.commit()
        else :
                print("Rogue AP: "+ str1)
                cursor.execute("INSERT or IGNORE INTO `rogueap` (mac_address, ssid, encryption) VALUES(?,?,?)", (str1, "", ""))
                conn.commit()
                #i = i+1

    #print(i)
    #print(packet.show())

sniff(iface="Wireless Network Connection", prn = PacketHandler, count=100, timeout=10)
```

Figure 8.1: Sniffing python code

VT OMNES VNVM SINT

APPENDIX B

Originality Report



Rogue Access Point Detection Framework on a Multivendor Access Point WLAN

ORIGINALITY REPORT

15%	10%	%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijcns.uacee.org Internet Source	2%
2	edoc.site Internet Source	1%
3	Submitted to Strathmore University Student Paper	1%
4	docplayer.net Internet Source	1%
5	troindia.in Internet Source	1%

VT OMNES UNVM SINT