A SERIAL NUMBER BASED AUTHENTICATION MODEL FOR A COMPUTER IN A WIRELESS LOCAL AREA NETWORK

JOHN CHEBOR

A Thesis Submitted to the Institute of Postgraduate Studies in Fulfillment of the Requirements for the Award of Doctor of Philosophy in Information Technology

KABARAK UNIVERSITY

NOVEMBER, 2022

DECLARATION

1. I do hereby declare that:

(i) This thesis is my own work and to the best of my knowledge it has not been presented for the award of a degree in any university or college.

(ii) That the work has not in-cooperated material from other works or a paraphrase of such material without due and appropriate acknowledgement

(iii) That the work has been subjected to processes of anti-plagiarism and has met Kabarak University 15% similarity index threshold

2. I do understand that issues of academic integrity are paramount and therefore I may be suspended or expelled from the University or my degree may be recalled for academic dishonesty or any other related academic malpractices.

Student Name: **John Chebor**

Student Signature …………………………………. Date …………………………………..

Admission Number **GDI/M/1087/09/14**

RECOMMENDATION

To the Institute of Postgraduate Studies:

This thesis entitled "**A Serial Number Based Authentication Model for a Computer in a Wireless Local Area Network**" written by **John Chebor** is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the research thesis and recommend it to be accepted in fulfilment of the requirements for the degree of **Doctor of Philosophy in Information Technology.**

Sign……………………………………. Date………………………………………

Prof. Simon Maina Karume
Department of Computer Science and Information Technology
Kabarak University

Sign ……………………………………. Date…………………………………………

Dr. Nelson Bogomba Masese
Department of Computer Science and Information Technology
Kabarak University

COPYRIGHT

ACKNOWLEDGEMENT

The journey of my research studies seemed long, tough and lonely as it is commonly said, but on the way, I realized that there were individuals that surrounded and kept me moving on to the end thus made the journey not so lonely after all. And so they deserve my heartfelt gratitude.

Just as the Bible says “Now to him who is able to do more than immeasurably all we ask or imagine, according to his power that is at work within us, to him be glory in the Church and in Christ Jesus throughout all generations, forever and ever! Amen” (Ephesians 3:20, 21, NIV), I would start my gratitude by forever being grateful to the almighty God for his peace, presence, and providence throughout my studies. There were many a time, I would feel discouraged, anxious and not being sure of what to do but God’s peace that surpasses all human understanding kept me going. Of course, without his presence, I would not have gone far. I believe strongly that God provided everything I needed for the study: the study idea, time, resources and supervisors, to mention but a few. In all these, I give all the glory and honour to God.

My special gratitude goes to the Kabarak University fraternity through the Dean of School of Science, Engineering and Technology, the Director of Institute of Postgraduate Studies and the Head of Department of Computer Science and Information Technology for their immense support in creating a conducive environment and ensuring that the study goes on as planned.

No words can describe appreciations to my family for their love, patience and moral support.

To my able supervisors, Prof. Simon Karume and Dr. Nelson Masese, I thank them very much for their undivided attention, prowess, guidance, and counsel throughout my research. You were truly the cornerstone and building blocks of my study.

Last but not least, I thank Mr. Reuben Sirma (ICT Systems Administrator, Kabarak University), Mr. Timothy Sawe, Mr. Andrew Kipkebut and Mr. Evans Chesang (Department of Computer Science and Information Technology, Kabarak University), Victor Rotok and friends (IT students, Kabarak University) for allowing me use their machines during the prototype test runs.

I might not mention all those who assisted me physically, morally, psychologically and mentally or in any other form, but to all, I say thank you, God bless and be with you throughout all your endeavors.

DEDICATION

To

My wife, Agnes Jemutai Rotich

and

My children, Emmanuel Kiptoo, Abigael Jerono, and Shadrack Ayabei

ABSTRACT

With today's technological evolution, wireless networks have become very common for organizations, homes and public places due to the numerous benefits that come with them, as compared with wired networks. One of the biggest challenge though, is on how to control the ever increasing and dynamic nature of devices that use such kind of a network. Network access control security service uses identification, authentication, authorization and accounting services in that order, are used to provide such needed security controls. Machine based authentication methods are token, IP address and MAC address methods with their corresponding token, IP address and MAC address identifiers. A MAC address, a physical network address that is used as basis for this study, has a copy of its value in the system software that can be spoofed and altered rendering the address not unique, not secure and unreliable. On the contrary, a computer's serial number is hard-coded in the system hardware only and therefore cannot be spoofed and altered making it unique, secure and reliable. The research, therefore, was aimed at designing a model that demonstrates how a computer's serial number can be used for authenticating a computer in a wireless local area network. In order to achieve the research objective, the study examined the inbuilt access and use of a computer's serial number prototype model as an alternative method of authenticating devices in a network. Design science research methodology that involved design and development, demonstration and model evaluation was employed. The Serial Number Based Authentication or SNAM model was therefore designed using state chart and flow chart diagrams based on dynamic programming, developed over evolutionary prototyping and test run on a static experimental design using Java Development Kit and MySQL platforms to demonstrate, as proof of concept, that a computer's serial number can be used to authenticate a computer in a wireless local area network. The SNAM model first registered computers so that on execution, unregistered computers were denied network access while registered computers were allowed access to the network, based on the computer's serial number. Allowed computer details are then displayed on an authentication interface to further either block a computer from the network if a need arises or allow the computer to continually use the network and its resources. From the test runs whose outcome were the binary values yes or no, it was found out that SNAM can actually allow or deny, enable or disable a computer in a network based on the computer's serial number. The researcher therefore, recommends that the prototype be scaled up, then adopted as a network device authentication method

Key Words: Computer's Serial Number, Authentication, Wireless LAN, Serial Number-Based Authentication

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

2FA	Two-Factor Authentication
3G	Third Generation
4G	Fourth Generation
ARK	Archival Resource Key
AP	Access Point
AUID	Allocation Unique Identifier
ATM	Automated Teller Machine
BYOND	Bring Your Own Device
CLI	Command-Line Interface
CHAP	Challenge Handshake Authentication Protocol
CMD	Command
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
DSR	Design Science Research
DHCP	Dynamic Host Configuration Protocol
DOI	Digital Object Identifier
EAP	Extensible Authentication Protocol
EAP-MD5	EAP-Message Digest 5
ER	Entity-Relationship Diagram
EUI	Extended Unique Identifier
FAST	Flexible Authentication via Secure Tunneling
GUI	Graphical User Interface
GSM	Global Systems of Mobile communication

HDLC	High-level Data Link Control
HTML	Hypertext Mark-up Language
IAAA	Identification Authentication Authorization Accounting
IdM	Identity Management
IAM	Identification and Access Management
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
ID	Identifier or Identification or Identity
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IFF	Identification of Friend or Foe
IJWMN	International Journal of Wireless & Mobile Networks
IMEI	International Mobile Station Equipment Identity
IoT	Internet of Things
ISBN	International Serial Book Number
ISNI	International Standard Naming Identifier
IP	Internet Protocol
JDBC	Java Database Connectivity
JDK	Java Development Kit
JVC	Java Virtual Machine
LAN	Local Area Network
Li-Fi	Light Fidelity
LCD	Liquid Crystal Display
MAC	Media Access Control
MFA	Multi-Factor Authentication

MPN	Manufacturer Part Number
MSN	Manufacturer Serial Number
NAC	Network Access Control
NACOSTI	National Commission for Science, Technology, Innovation, and Communication
NIC	Network Interface Controller
NIV	New International Version
OCR	Optical Character Recognition
OSI	Open Systems Interconnection
OTP	One-Time Password
PAP	Password Authentication Protocol
PAC	Protected Access Credential
PAM	Pluggable Authentication Module
PDA	Personal Digital Assistance
PBN	Product Batch Number
PHP	Hypertext Pre-processor
PIN	Personal Identification Number
RA	Router Advertisement
RADIUS	Remote Authentication Dial-In User Service
RDBMS	Relational Database Management System
RFID	Radio Frequency Identification
SCM	Supply Chain Management
SFA	Single Authentication Factor
SIMM	Single In-line Memory Module
SLAAC	Stateless Addresses Auto Configuration

SLIP	Serial Line Internet Protocol
SSN	Social Security Number
SSID	Service Set Identifier
SQL	Standard Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universally Unique Identifier
VIAF	Virtual International Authority File
WAP	Wireless Access Point
WEP	Wired Equivalent Privacy
Wi-Gig	Wireless Gigabit
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

OPERATIONAL DEFINITION OF TERMS

Identifier or Identity: The set of attribute values (characteristics) by which an entity (a device or a computer) is recognizable and that, within the scope of an identity manager's responsibility, is sufficient to distinguish that entity from any other entity.

Identification: The process of verifying the identity of a user, process, or device, usually as a prerequisite for granting access to resources in an Information Technology (IT) system.

Serial Number: A number that is put in a product/device in order to identify it.

Wireless Local Area Network: Or WLAN is a group of wireless networking devices within a limited geographical area, such as an office building, that exchange data through radio communications.

Attackability: The ability to attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity.

Collectability: The ability of an identifier to be captured from the existing available devices.

Reliability: The likelihood of a computer system or device continuing to function over a given period of time and under specified conditions.

Uniqueness: The quality of being one of a kind, that is, no two devices should have the same identifier.

Robustness: Robustness as described in this study is synonymous to reliability, validity, performance, dependability, predictability among other closely related terms. Robustness can be summarized as the ability of an identifier to remain invariant over time.

Spoofing: Spoofing is the process of falsifying one's identity or masquerading as some other individual or entity to gain access to a system or network or gain information for some other unauthorized purpose.

Network Access Control: Network access control (NAC) is an enterprise security solution used to assess, manage, enforce, and optimize security and authentication policies through different measures like endpoint security, user access authentication, and network security policies. NAC constitutes identification, authentication, authorization, and accounting (IAAA) services.

Authentication: Authentication, as defined by the Economic Times, (2022), refers to the process of recognizing a user's identity. Authentication is part of identity management (IdM) or commonly referred to as identity and access management (IAM).

Identity and Access Management: Identity and access management authentication (IAM) is a discipline that not only identifies, authenticates and authorizes to network and network resource access but also to the hardware and applications the users need access to.

Authorization: Authorization determines a user’s privileges or eligibility of accessing or executing a resource in the network.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

This chapter begins with the background information concerning wireless LANs, then proceeds to state the study problem, objectives, research questions, and scope of the study. Finally, the chapter winds up by presenting the assumptions, limitations and significance of the study.

1.2 Background Information of the Study

Wireless LANs (WLAN) also known as Wireless Fidelity (Wi-Fi) or 802.11 standards is a type of a local area network that allows users access network services using mobile devices (wireless stations) such as laptops, personal digital assistants, smart watches and even smartphones (Dordal, 2018). The wireless stations use a base station usually an access point (AP) or a hotspot as an entry point to the network services. Unlike wired LANs that use cables or wires as transmission media, WLANs uses radio wave frequencies to transmit information over the local area network.

According to Mareco, (2015) and Mohapatra, Choudhury & Das, (2014), the most currently deployed Wi-Fi technologies include Tri-band Wi-Fi___33 or WiGig or IEE802.11ad, Light Fi (LiFi), Advanced Enterprise WiFi, WiFi CERTIFIED AC, and Wi-Fi CERTIFIED Passpoint, in the order of their technological advancements. Something common about all these technologies is in the improvements in the speed of transmission in each subsequent technology. In addition to speed improvement, Advanced Enterprise WiFi allows users to login to a WiFi network using their social credentials. Wi-Fi CERTIFIED Passpoint ultimately allows online-sign up for mobile devices without a SIMM card.

WLAN, therefore, comes with a myriad number of benefits as compared to wired LANs, notably, mobility, rapid deployment, reduction in infrastructure and operational costs, flexibility, and scalability (Raji 2014; DHS, 2017; Wallace, 2018; Smith et al., 2019). Due to these benefits, hotspots are now virtually found everywhere; in enterprises, at homes, and in public places. Wireless devices such as laptops, personal digital assistants and even smartphones come with Wi-Fi features integrated in them. Despite the numerous benefits that come with wireless LANs, network insecurity has become a thorn in the flesh for proprietors of such kind of networks. Singh & Sharma, (2015), points out that an attacker simply needs to be within the range of the WLAN access point to intrude into the network as opposed to wired LANs where an attacker requires physical access to the LAN or remotely use porous firewall systems to gain access to the LAN. WLANs are therefore easily targeted by attackers through spoofing, denial of service, eavesdropping, man-in-the-middle, masquerading, message modification, message replay and traffic analysis (Stallings, 2011; Poremba, 2017; DHS, 2017). Wallace, (2018), describes the reasons for the threats as default configurations, network architecture nature, encryption weaknesses, and physical security.

The numerous benefits that come with WLAN have made enterprises to adopt bring your own device (BYOD) concept to allow employees use their own devices (Poremba, 2017). As such, it results to flexibility, increased employee satisfaction and productivity, and reducing infrastructural and service costs of the enterprise. Apart from the employees accessing and using enterprise networks, another group of stake holders such as visitors, vendors, and contractors at one point, if not all, can as well require network usage as they carry on their businesses with the enterprise.

Rise in internet of things (IOT) devices such smart devices, smart watches and smart phones, as well, further complicates WLAN challenges equation (Pierce, 2021). Allowing users to connect to the network with their own devices can pause as a security challenge as it becomes difficult for network administrators to control such kind of a network access and usage. It is therefore imperative that network administrators use network access control tools to control who should and who should not access the network. One of such kind of a control tool is the network access control (NAC) (Robb, 2022; Pierce, 2021; & Chiradeep, 2021).

According to Chiradeep, (2021), Network access control (NAC) is an enterprise security solution used to assess, manage, enforce, and optimize security and authentication policies through different measures like endpoint security, user access authentication, and network security policies. NAC constitutes authentication, authorization, and accounting (AAA) according to Lawson, (2017). The controls are further identified by Garska, (2016), as identification, authentication, and authorization (IAA) as the essential functions in providing the required services in a network.

Authentication, as defined by the Economic Times, (2022), refers to the process of recognizing a user's identity. In other words, it is a method or a mechanism of associating an incoming request with a set of identifying credentials (such password, username, PIN, MAC address, IP address) and compared with those in a database within an authenticating server to either allow or deny access to a network service or resource. Authentication methods are categorized based on what one is known for or knowledge based (such as a password or a PIN), what one has or possesses (token, certificate), who one is or inheritance (biometrics), where one is or location based or address based (MAC address, IP address) and when one is authenticating or time factor as well (Shacklett, 2021).

1.3 Problem Statement

Among the commonly used authentication methods (that is password, smart card, biometric, certificates, MAC address and IP address based authentication), password, smart card, biometric, certificate methods are used by users or people in a system (referred to as user authentication) for authentication. MAC address, IP address and at times certificates, on the other hand, are machine based authentication methods (Fredriksson, 2017). Further, password, smart card, biometric and certificates, form part of the upper OSI reference model layers issue that provide services to the end users while MAC address and IP address based authentication provide services in the Data Link and Network layers of the OSI reference model respectively (Williams, 2022). That results to a missing authentication method at the Physical layer of the OSI reference model.

A MAC address and an IP address that are machine based identifies, are referred to as indirect rather than direct identifies. This is due to the fact that both are pointers to addresses just as their name expressions suggest. Actually, whereas MAC addresses are used by messages to identify actual physical destination and source networks, IP addresses are used by information to be effectively routed from network to network in an internetwork (Kurose & Ross, 2013). This then calls for an identifier that points to the device itself or a direct identifier rather an indirect identifier as in the case.

Apart from the absence of identifiers at the physical layer of the OSI reference model, MAC addresses, which are used to identify the physical source and destinations addresses, are not universally unique across networks, they are easily spoofed and altered to aid in spoofing attacks. In addition, MAC addresses are coded on the network hardware and a copy is replicated in the operating system when the system starts, to facilitate network troubleshooting and configuration. This copy of the MAC address is the one that is usually spoofed. Furthermore, a device could have several network interfaces that results to several MAC addresses. A MAC address is only unique to an interface but not unique to the device. A MAC address is therefore not unique, unreliable and not secure.

It is for these reasons then that the study investigated how a serial number can be used for physical authentication of a computer in a wireless LAN

1.4 General Objectives

The main aim of the study was to design a model that demonstrates how a serial number can be used for physical authentication of a computer in a wireless LAN. The specific objectives were as follows;

1.5 Specific Objectives

i. To develop an algorithm that can obtain and use a remote computer's serial number in a wireless LAN

ii. To design a model that uses the computer's serial number to authenticate the computer in a wireless LAN

iii. To demonstrate how the computer's serial number can be used to authenticate the computer in a wireless LAN

iv. To evaluate the model that uses the computer's serial number to authenticate the computer in a wireless LAN

1.6 Research Questions

In order to achieve the aim of the study, the following research questions had to be answered

i. How can an algorithm that can obtain and use a remote computer's serial number in a wireless LAN be developed?

ii. How can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be designed?

iii. How can the model that uses the computer's serial number to authenticate the computer in a wireless LAN be demonstrated?

iv. How can the model that uses the computer's serial number to authenticate the computer in a wireless LAN be evaluated?

1.7 Scope of the Study

The study described in this thesis was conducted in a wireless LAN (WLAN) set up that constituted an authenticating server, an access point (AP), a client and connections. The choice of the set was based on the fact that the composition of WLAN in terms of infrastructure (mobile devices, access points, radio waves), in addition to the challenges associated with it, makes it prone to attacks as opposed to wired LAN that an attacker requires physical access to intrude into the system.

Although computers include laptops, tablets, desktops, and palmtops, the study focused on the use of laptops to start with. This was informed by the fact that laptops are portable general purpose computers in nature, readily available and therefore can easily be manipulated through the numerous applications they can handle. The research in addition focuses on use of a computer's serial number which is an attribute of a laptop unlike tablets and smart phones that use IMEI for as an identifier.

1.8 Significance of the Study

The study was intended to eventually benefit enterprises, homes and any other wireless LAN network proprietors that wish to identify, hence authenticate, authorize and control devices or users of their network based on a device serial number, rather than based on an IP address or a MAC address that can be spoofed, therefore not secure, unreliable and not unique. Wireless network experts (network and systems administrators) would also ultimately benefit in authenticating and controlling devices that use their network.

1.9 Assumptions of the Study

As in line with the objective of the study, the research focused on proving that a computer's serial number can be used to authenticate a device in a wireless LAN. It is for this reason then the study was carried out based on the following assumptions;

i. The study experimented only on laptops despite the fact that there a number of networks devices that include cellphones, tablets and even routers. The study recommends for development of a system that can accommodate other attributes apart from and are related to serial numbers used by other devices like IMEI in cellphones.

ii. The pilot result indicated that old computer models did not have their serial numbers encoded in the system hardware but had them tagged on the computer surfaces. Modern computers on the other hand have their serial numbers both tagged on their surfaces and encoded on the system hardware. The serial number on the hardware of modern machines, therefore, can be accessed and used for authentication, not for older machines that does not have their serial numbers encoded in the system hardware.

1.10 Limitations of the Study

Despite the fact that the expected results of the study were realized, the following limitations, however, were encountered during the research undertakings

i. The research required funding for laptops, routers and software resources needed during study test runs. But due to limited funding sources, some resources that could not be bought were sourced from friends and well-wishers.

ii. Due to time constraints, a number of parts or components of the study that could have been done were not carried out. They include such aspects as automating computer registration, improvement and deployment of the serial number based authentication model (SNAM), development of a system that caters for other network device identifiers such as IMEI, and integration of SNAM system into an access point, were done due to time constraints. The components were recommended for future undertakings in the recommendations section of the study.

iii. The other limitation encountered was that no much work had previously been done on a computers serial number based authentication. However, the study was based on other related identifiers based authentication such MAC address and IP address.

1.11 Thesis Layout

This thesis consists of a total of five chapters. Chapter one which is the introduction part contains the background information concerning wireless LANs, then proceeds to state the study problem, objectives, research questions and scope of the study. Finally, the chapter winds up by presenting the assumptions and significance of the study. Chapter two provide an overview of current device authentication techniques and technologies that concerns the study, discussions on network access controls, suitability of existing network device identifiers, suitability of MAC address as a network device identifier, suitability of computer' serial number as network device identifier, authentication protocols, authentication methods, the research gap and the conceptual framework. Chapter three focuses on the design and methodologies used to design and demonstrate the research. Chapter four which is the core part of the thesis focuses on design, implementation and testing the use of the serial number as an alternative method to using a MAC address. The last chapter five of the thesis provides conclusions, recommendations, and areas for further research on as far as the study is concerned.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter provides an overview of current device authentication techniques and technologies that concerns the study. Consequently, the chapter presents discussions on network access control, suitability of existing network device identifiers, suitability of MAC address as a network device identifier, suitability of computer' serial number as network device identifier, authentication protocols, authentication methods, the research gap and the conceptual framework.

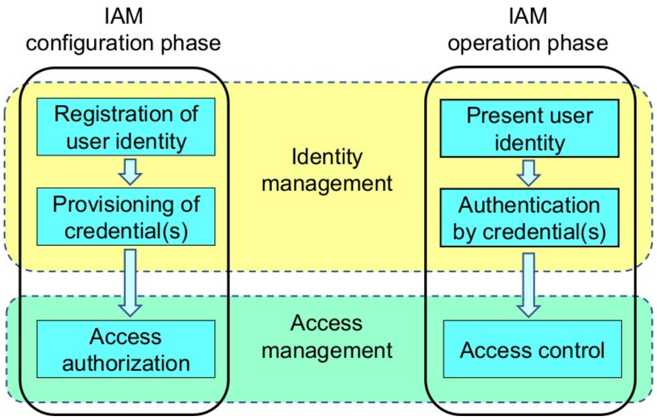
2.2 Network Access Control

As mentioned earlier in the introductory part of this study, authentication is part of the wider network access control (NAC) services needed to manage how users and machines access network and network resources (Robb, 2022; Pierce, 2021; Chiradeep, 2021). Other services NAC provides apart from authentication are identification and authorization (IAA). IAA ensures that only authenticated and authorized users can access a network and appropriate network resources using such a strong access control mechanism (Vital, 2019).

It is equally worthy to note that authentication is part of identity management (IdM) or commonly referred to as identity and access management (IAM), a discipline that not only identifies, authenticates and authorize to network and network resource access but also to the hardware and applications the users need access to (Strom, 2021). IAM is therefore put into configuration and operations phase as shown in the Figure 1 below.

Figure 1

Identity and Access Management Components. (Source: Wikipedia, 2022)



It first registers and gives authorization access rights or privileges to users in the configuration stage, then identifies, authenticates and controls user access to the network, applications or resources based on the previously authorized access rights in the operations stage.

IBM, (2022), defines identification as the ability to uniquely identify a user of a system or an application that is running in the system. Examples of identifiers for identification include a password, an ID, a card, biometrics, an email address, an IP address or a MAC address. Authentication on the other hand is the ability to prove that a user or an application is genuinely who that person or what that application claims to be, by adding another subject credentials. A password, for instance, verifies that the user is the owner of the username supplied when password authentication method is used. Other authentication methods include token based, certificates based, biometrics based, IP address based and MAC address based authentication methods. Authorization determines a user's privileges or eligibility of accessing or executing a resource in the network.

2.3 Suitability of Existing Network Device Identifiers

This section focuses on the suitability of current network device identifier at the OSI reference model data link layer namely the MAC address in relation to other identifiers at upper layers of the OSI reference model that include port numbers and IP addresses. But first, it describes identifiers and then describes the characteristics that an identifier should possess to be a good identifier that forms the basis of describing the suitability of an identifier. The section then looks at other identifiers in relation to the MAC address.

2.3.1 An Identifier

An identifier, according to Coulouris et al., (2012), is an attribute (or a combination of attributes) whose value distinguishes instances of an entity (device) type from another. Examples of such identifiers could be a code (identification number, serial number, ISBN). a name (domain name) or an address (IP, MAC or Port Number)

Identifiers can be categorized into various forms. But most importantly is whether they vary or remain invariant over their lifetime. Lavassani, Movahedi, & Kumar, (2006), lists unchangeable identifiers as Date of Birth, Mother's Maiden Name, Social Insurance Number (SIN) (depending on the governmental procedures), some of Membership Numbers (depending on organizational procedures), Tax Payers Identification Number (depending on the governmental procedures), Passport Number (depending on the governmental procedures) and Account Numbers/ID Numbers (depending on the organizational procedures). Changeable identifiers include Names, Newsgroup names on Usenet, Addresses, Telephone Numbers, PIN Numbers (password), Credit card information, Network Domain, Driver's License Information, Birth Certificate Information, SIN Number (depending on the governmental procedures), some of Membership Numbers (depending on organizational procedures), Tax Payers Identification Number (depending on the governmental procedures), government Passport Number (depending on the governmental procedures) and Account Numbers/ID Numbers (depending on the organizational procedures).

A more elaborate classification of identifiers was done by Leggott et al., (2016), basing on identifier landscape. Researcher Identifiers (that include Allocation Unique Identifier (AUID), Google Scholar ID and Virtual International Authority File (VIAF) also known as Author or Scholar Identifiers establish a unique identity for a given author or creator. Object Identifiers (Digital Object Identifier (DOI) and Universally Unique Identifier (UUID)) establish a unique identity for a specific digital object. Organizational Identifiers, for example International Standard Name Identifier (ISNI), establishes a unique identity for a specific organizational entity. Equipment Identifiers establishes a unique identity for a specific piece of equipment or subcomponents of such equipment.

2.3.2 Characteristics of a Good Device Identifier

An attribute or an identifier should possess the following properties as adopted from Boodoo-Jahangeer, (2010), for it to be a good identifier;

i. Universality: every device in the considered device-space should have the desired features
ii. Uniqueness: no two devices should have the same identifier
iii. Performance: the identifier should be invariant over time
iv. Collectability: it should be captured from the existing available devices

Danev et al., (2015) adds data-dependency or the ability of an identifier to be associated with the device data transmitted by the device, to the list of attributes presented

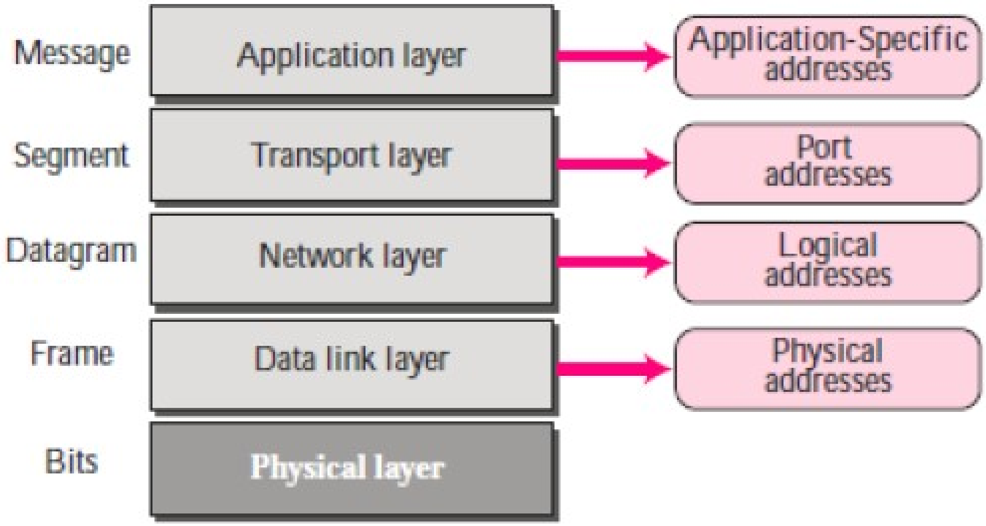
Leo, (2004), identifies the ease of use, usefulness, compatibility, privacy, security, normative beliefs and self-efficacy as some of the key factors that are important in technology acceptance models. In addition, Lavassani et al., (2010) cites ease of use of technology acceptance level of the system, security of identification factor practice represents the effects of using the identifier, and cost-benefit analysis represents the efficiency of the identifier. In other words, the characteristics of a good identifier can be summarized as ease of use, effectiveness, efficiency, and cost of the identifier.

2.3.3 Currently used Device Identifiers in a Network

A number of device identifiers have been proposed in the literature. Wang et al., (2016) presented the use of fingerprinting, a method that uses characteristics of network traffic to identify a device. Although this technique can be applied in thwarting attacks (Pang et al., 2007), it prefers passive features over active features (Xu et al., 2015). The most widely discussed identifiers in literature which also falls under the traditional methods of device identifiers are application specific addresses, port numbers, IP address and MAC address of a device. These identifiers operate at different levels of the OSI reference model as illustrated in Figure 2 below.

Figure 2

OSI reference model address layering. (Source: Jeevanesh, 2017)



These identifiers are discussed as follows:

i. Application-Specific Address

Application specific addresses are addresses that are designed for a specific application geared towards user-friendliness. Also referred to as persistent identifiers (Richards et al., 2011), the application-specific identifier is permanently assigned to an object. Examples of application-specific addresses include e-mail address such as deanset@kabarak.ac.ke and a universal resource locator (URL) such as kabarak.ac.ke. Whereas an e-mail address defines the recipient of an e-mail, a URL is used to find a document on the internet.

Such addresses or locators fundamentally play a crucial role in enabling internet users easily find information in the internet (Commer, 2013). This is more so as the internet has a huge amount of information which makes it difficult to find. Labelling the files or objects in a way of application-specific addresses, therefore, makes it easy to find a specific object or file. The addresses, however, get changed to the corresponding port and MAC addresses of the sending computer.

ii. Port Numbers

Port numbers are numbers on hosts or devices to identify sending and receiving processes. According to Lee, (2010), port numbers are usually captured when a system requests for a page with usernames and password details. Process identification in communication is crucial because a host could be running several applications. A port number is a 16-bit address represented by one decimal number. A Web server, for example, is identified by port number 80 and a mail server process is identified by port number 25 (Kurose & Ross, 2013).

Although port numbers can be used in conjunction with other numbers, namely, IP address and MAC address to identify a host, port numbers can pause as a threat to network security. Intruders can use a port scanner that listens to well-known port numbers so as to detect services running on a system, therefore can easily break into the system. Attackers identify services on an open port service running on a host and use it to exploit vulnerabilities (Canavan, 2012).

iii. IP Address

An IP address is a number assigned to a host or a router on the internet for identification and location of the device as stated by Tanenbaum, (2011). As far as the internet is concerned, an IP address is universally unique across networks; therefore, no two devices on the internet have the same IP address.

To ensure uniqueness and management of IP addresses, Internet Corporation for Assigned Names and Numbers (ICANN), an operator of the Internet Assigned Numbers Authority (IANA), distributes IP addresses in a hierarchical system (ICANN, 2011). It allocates IP address blocks to the Five Regional Internet Registries (RIRs) around the world roughly continental in size. The RIRs then allocates smaller IP address blocks to Internet Service Providers (ISPs) and any other network operators. And eventually, the IP address gets to the user via ISPs and other operators.

An IPv4, which is currently in use (Kurose & Ross, 2013), is composed of four dotted decimal notations (example of an IP address is 243.246.0.28) each part range between 0 and 255. Depending on the class or the subnet mask of the IP address, some parts (first) belong to network identification and the other parts (usually from the second part) belong to host identification. IPv4 was designed to use the 32-bit address space (Shay, 2004). This translates to 2^{32} or approximately four (4) billion addresses which are not sufficient for the global needs. Actually, the numbers were projected to have been exhausted by 2013 (IEEE-USA, 2009).

Although the ultimate solution to the exhaustion of IPv4 lies in the implementation of IPv6, a number of temporary solutions such as Network Address Translation (NAT) (Tanenbaum, 2011) and Dynamic Host Configuration Protocols (DHCP) (Shay, 2004) have been suggested. In NAT, an organization is assigned one public unique IP address and internal computers are assigned with internal IP addresses that are unique internally but not unique externally to the organization. Through NAT, internal devices use the public IP address to communicate to external devices rather than their private IP addresses.

A common way of conserving IP address used by network administrators in apart from organizational LANs, has also found use in residential and wireless LANs, is the use of Dynamic Host Configuration Protocol (DHCP) (Kurose & Ross, 2013). DHCP assigns devices IP addresses on demand, therefore two devices can use the same IP address but on different occasions. This then implies that an IP address cannot uniquely identify a device in a LAN.

The ultimate solution to IPv4 exhaustion problem is the implementation of IPv6. Just as it is with IPv4, IPv6 deals with addressing as well as internetwork interface identification on the internet (Nagaraj *et al*. 2010). Although the development of IPv6 itself has been completed, it is still undergoing refinement as well as ensuring that other related compatible protocols with IPv6 are developed. Furthermore, experimental trials on IPv6 have been going on for a while with caution to ensure that no such mistakes would be repeated as was with IPv4.

An IPv6 address is designed to use a 128-bit address, surprisingly, skipping the 64-bit address which could have been the case given that IPv4 uses 32-bit address. Of course, this is a case of the developers not wanting to repeat the mistakes of IPv4 particularly on the exhaustion of address space issue. As a result, IPv6 allows the use of 2^{128} which translates to a huge approximately 3.4×10^{38} undecillionth addresses or about 340 trillion, trillion, trillion addresses.

The 128-bit address is such long that the IPv6 address is expressed in hexadecimal notation with 8 sets of 16-bit and separated by colons (:) (Graziani, 2017). An example of IPv6 address is 2031:0000:130F:0000:0000:09C0:876A:130B. The address can be simplified by omitting the leading zeros (they are not significant) and representing consecutive zeros in continuous blocks by double colons (::) of which such double colons appear only once in the address. After simplification, the addresses stated in the example become 2031:0:130F:9C0:876A:130B. A closer look at the structure of the addresses indicates the address being composed of two logical parts: a 64-bit network prefix and a 64-bit host address part.

The 64-bit prefix, according to Graziani, (2017), is meant for routing data on the internet. The 64-bit prefix is further split into two sections of 48 bits and 16 bits. Whereas the first 48 bits are used for global network address reserved for routing over the internet, the second 16 bits section is used for subnets of internal networks that are controlled by network administrators.

The second 64-bit part is used to identify the node or the network interface of the attached node derived from the actual physical address (MAC address of the nodes interface) using IEEE's extended unique identifier (EUI-64) format, or using a randomly generated number or still, using a DHCPv6 assignment protocol.

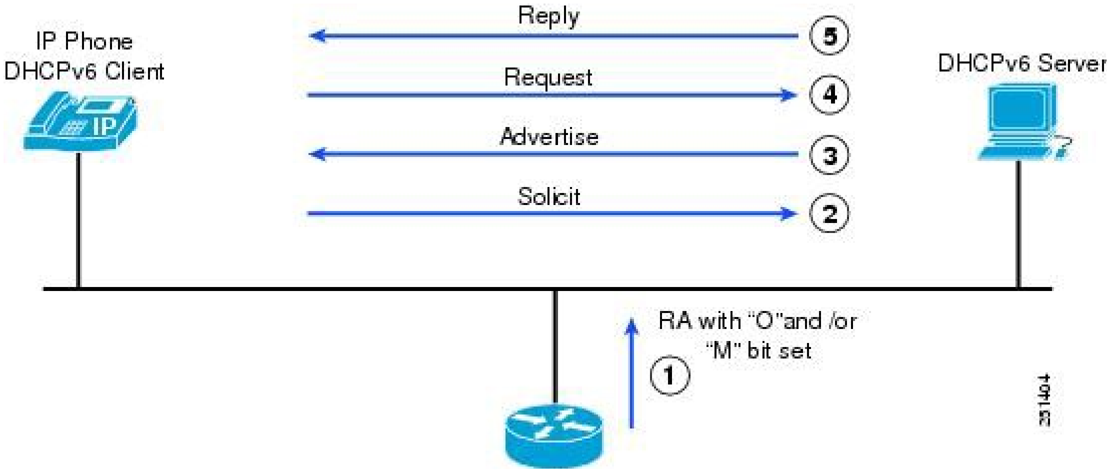
But of greater interest to this study is the configuration of IPv6 address. Graziani, (2017) points out three ways of assigning an IPv6 address to a device as using manual configuration, IPv6 stateless address auto configuration and DHCP assignment. In a manual configuration, the network administrator assigns the IPv6 address manually to a device. Although the method is preferred for assigning static addresses to devices such as router interfaces and some resource, it is not favorable for assigning to large number size devices such as computers. IPv6 is so complex such that it would be tedious and cost lots of overheads to network administrators if they were to configure devices manually.

The stateless auto configuration enables the network administrator to set up stateless addresses auto configuration (SLAAC) on an IPv6 router. The router sends router advertisements (RA) configurations on the links for the connected nodes to configure themselves with an IPv6 and routing parameters. In this method, the devices, obtain the IPv6 network prefix from the link-local router's RA create the IPv6 host ID based on the device MAC address and the EUI-64 format for the host IDs.

The other option of IPv6 assignment is for the devices to multicast to get IPv6 addresses and find DHCP servers. The client wishing to be assigned an IPv6 address first sends a request on the attached local networks to detect available DHCPv6 servers. The server then responds with requested information in a Reply message as illustrated in the Figure 3 below

Figure 3

IPv6 DHCP Messages (Source: Graziani, 2017)



The DHCPv6 client knows whether to use DHCPv6 based upon the instruction from a router on its link-local network. The default gateway has two configurable bits in its Router Advertisement (RA) available for this purpose:

i. O bit—when this bit is set, the client can use DHCPv6 to retrieve other configuration parameters (for example, TFTP server address or DNS server address) but not the client's IP address.

ii. M bit—when this bit is set, the client can use DHCPv6 to retrieve a managed IPv6 address and other configuration parameters from a DHCPv6 server.

iv. MAC Address

A MAC address also known as a LAN address or a physical address or still a link address is a number used to identify a network adaptor on a LAN. The size and format of the address vary depending on the type of the network. A MAC address in an Ethernet network, for example, is a 12-digit hexadecimal number with the following general and specific example of a flat structure

MM:MM:MM:SS:SS:SS example 00:1E:4C:CA:AB:29

The first three or a half parts of the structure contain the ID number of the adaptor's manufacturer. The second half or the last three parts represent the actual number of the adaptor as assigned by the manufacturer.

The address can either be a unicast, multicast or broadcast. While unicast MAC address enables the transmission of information from a sender to one single recipient, multicast involves one sender and a number of receivers and broadcast allows all signals to be received by all receivers in a network.

As Kurose & Ross, 2013 puts it "it is not hosts and routers that have link-layer addresses but rather their adapters (that is, network interfaces) that have link-layer addresses." In other words, a MAC address is used not by devices but by information to identify a particular physical network like an Ethernet network. A device with multiple interfaces has multiple corresponding MAC addresses. The management of MAC address space is as well, the prerogative of IEEE internet standard. This then implies that different adaptors from different manufacturing companies cannot have the same MAC address. Furthermore, the MAC address remains the same throughout the life span of an adapter.

As a device identification measure, access points can be configured to allow only recognized MAC address to access the network (Wallace, 2018). How is it possible? The network administrator simply keeps a record of all MAC addresses in a database server. The addresses are distributed to all APs that use them to filter out devices whose MAC addresses are not in the list in what is referred to as MAC address filtering (Singh & Sharma, 2015)

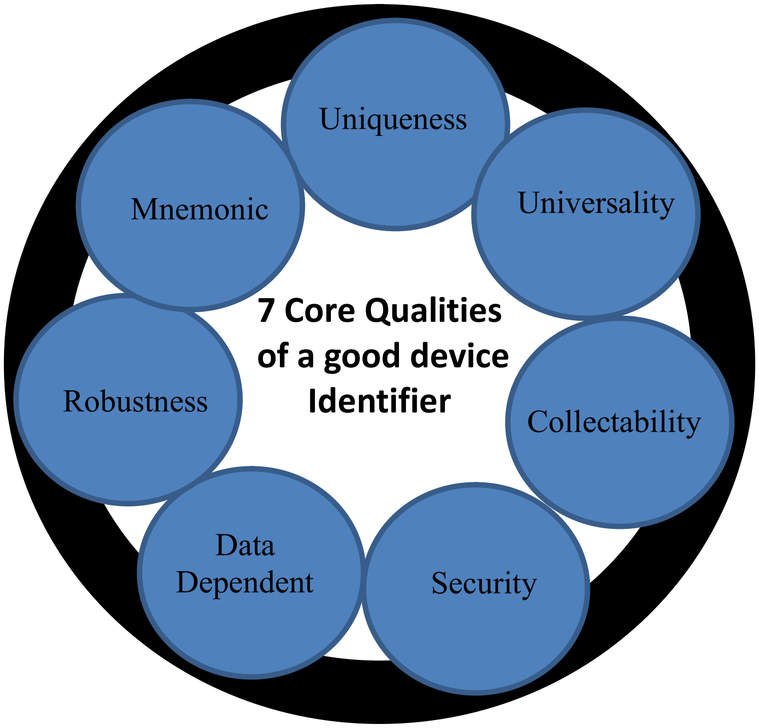
2.4 Suitability of a MAC Address as a Network Device Identifier at the Physical Layer

This section investigates the suitability of a MAC address as a physical layer identifier so that it can provide a basis of understanding the need of a computer's serial number as a physical network device identifier. As such, it presents a discussion on the three characteristics that required examination: security, robustness and uniqueness of a MAC address.

The characteristics that define the suitability of an identifier as were established in this study were uniqueness, universality, collectability, data dependent, security (availability, integrity, and confidentiality), robustness and mnemonics, best illustrated in Figure 4 below

Figure 4

Seven core qualities of a good device identifier (Source: Author)



As compared to the other identifier characteristics, security, robustness and uniqueness characteristics were established to compromise the suitability of MAC address as an identifier at the physical layer. These requirements were examined as follows:

2.4.1 Security of a MAC Address

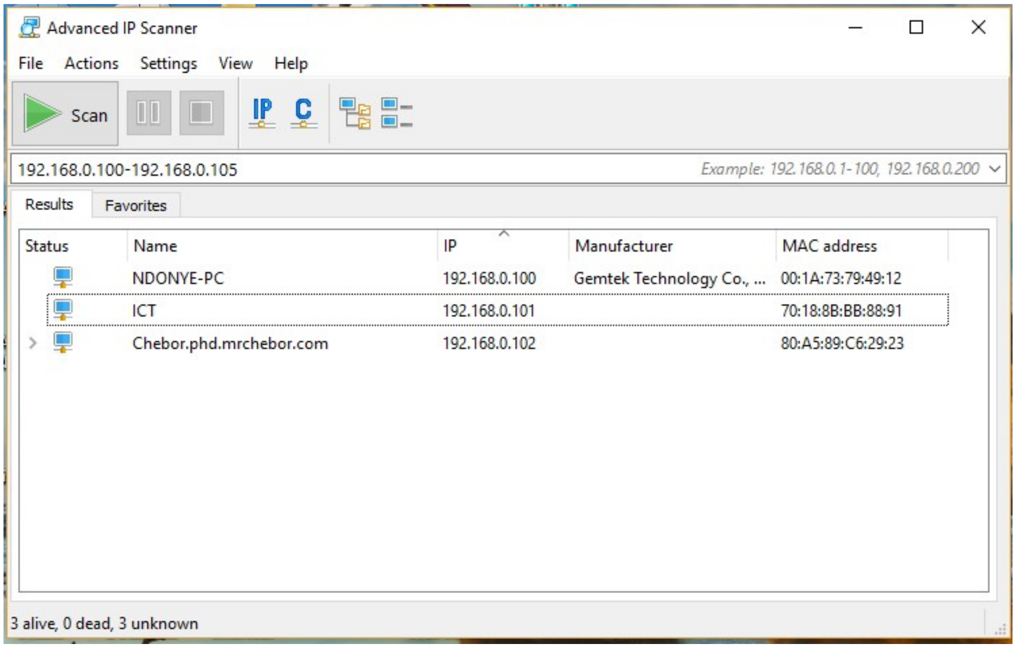
Availability, confidentiality and integrity aspects of security determine the suitability of a MAC address.

2.4.1.1 Confidentiality of a MAC Address

Confidentiality of an identifier is the degree of how the identifier can be disclosed to an unauthorized entity (Paulsen & Byers, 2019). Although measures have been put in place to protect the confidentiality of a MAC address by coding it into the network hardware, as a network device identifier, attackers would always have a way of getting it authoritatively. An Advanced IP Scanner tool, for instance, can be used by an intruder to access the MAC address of all devices connected to the network. To demonstrate this, the researcher created a test network of three computers and then the scanner was run from one of the computers. The results were presented as in Figure 5.

Figure 5

MAC address spoofing using Advanced IP Scanner



192.168.0.100-192.168.0.105

Example: 192.168.0.1-100, 192.168.0.200

Status	Name	IP	Manufacturer	MAC address
	NDONYE-PC	192.168.0.100	Gemtek Technology Co., ...	00:1A:73:79:49:12
	ICT	192.168.0.101		70:18:8B:BB:88:91
	Chebor.phd.mrchebor.com	192.168.0.102		80:A5:89:C6:29:23

3 alive, 0 dead, 3 unknown

As demonstrated in the results in figure 5 above, the scanner collected all the MAC addresses of the networked computers. This may result in the following flaws

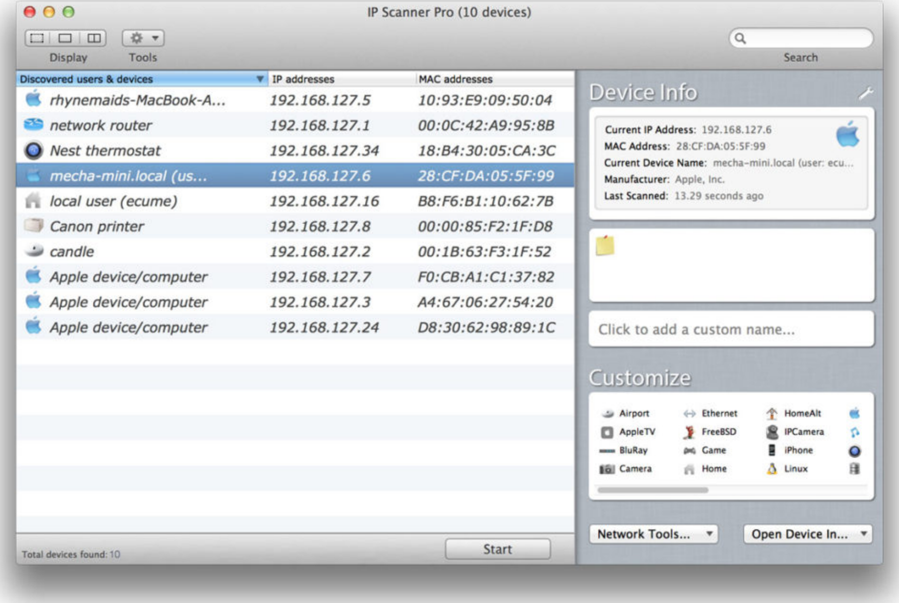
1. If the network is designed to use MAC filters to allow or block network access based on valid existing MAC addresses, then an attacker may use a MAC address spoofed using the advanced IP scanner to gain access to the network
2. When an attacker knows ones MAC address then they can use it to track a network user
3. If the valid MAC address device and the spoofed MAC address device log onto a network simultaneously, the addresses conflict with each other resulting in miscommunication and inconvenience on the valid device end.

In this demonstration, IP address and MAC addresses of the connected devices were fetched. The computer's serial number could not be fetched alongside the IP and MAC address due to its robust characteristics. There exist other spoofing tools apart from Advanced IP Scanner such as IP Scanner Pro 10, PCFinder, Angry IP Scanner, MAC Address Scanner, Colasoft MAC Scanner and Ipscan (Kumar, 2022), that can be used to spoof IP and MAC addresses

A freeware tool, IP Scanner for instance, allows a user to track network changes over time, for both active and inactive devices, as well whitelist or filter out trusted devices. Some of the captured details are illustrated in the Figure 6 below**.** PC Finder on the hand scans for an IP range and rerieves details such as host names and MAC addresses by prompting the user to key in desired IP address range. It is fast with minimum configurations. An exapmle of PCFinder screen shot is shown in the Figure 7 below.

Figure 6

MAC address spoofing using Advanced IP Scanner Pro. (Source: Larue-Langloise, R., 2022)



Discovered users & devices	IP addresses	MAC addresses
rhynemaids-MacBook-A...	192.168.127.5	10:93:E9:09:50:04
network router	192.168.127.1	00:0C:42:A9:95:8B
Nest thermostat	192.168.127.34	18:B4:30:05:CA:3C
mecha-mini.local (us...	192.168.127.6	28:CF:DA:05:5F:99
local user (ecume)	192.168.127.16	B8:F6:B1:10:62:7B
Canon printer	192.168.127.8	00:00:85:F2:1F:D8
candle	192.168.127.2	00:1B:63:F3:1F:52
Apple device/computer	192.168.127.7	F0:CB:A1:C1:37:82
Apple device/computer	192.168.127.3	A4:67:06:27:54:20
Apple device/computer	192.168.127.24	D8:30:62:98:89:1C

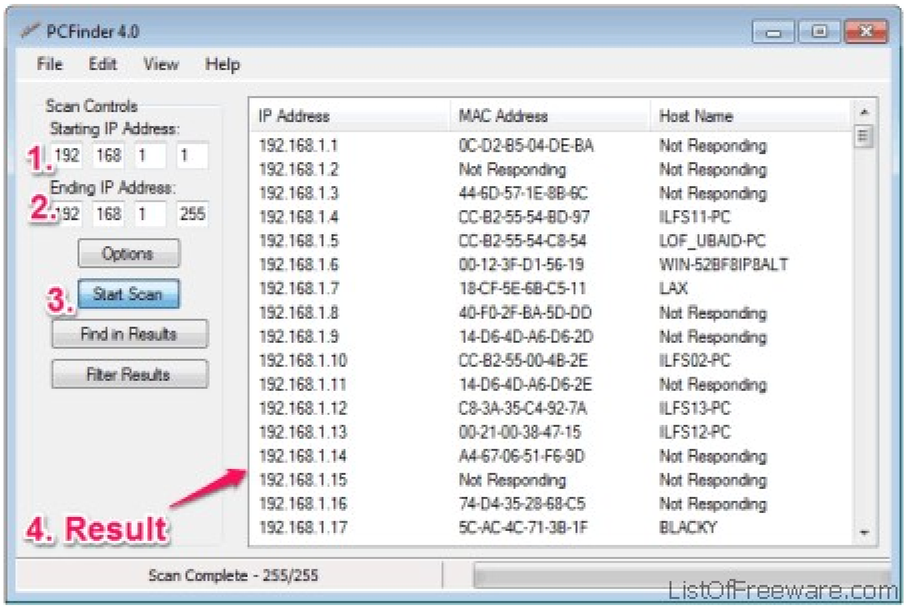
Current IP Address: 192.168.127.6
MAC Address: 28:CF:DA:05:5F:99
Current Device Name: mecha-mini.local (user: ecu...
Manufacturer: Apple, Inc.
Last Scanned: 13.29 seconds ago

Click to add a custom name...

Total devices found: 10

Figure 7

MAC address spoofing using PCFinder (Source: Popa, 2012)



Starting IP Address:
1. 192 168 1 1

Ending IP Address:
2. 192 168 1 255

3. Start Scan

4. Result

IP Address	MAC Address	Host Name
192.168.1.1	0C-D2-B5-04-DE-BA	Not Responding
192.168.1.2	Not Responding	Not Responding
192.168.1.3	44-6D-57-1E-8B-6C	Not Responding
192.168.1.4	CC-B2-55-54-BD-97	ILFS11-PC
192.168.1.5	CC-B2-55-54-C8-54	LOF_UBAID-PC
192.168.1.6	00-12-3F-D1-56-19	WIN-52BF8IP8ALT
192.168.1.7	18-CF-5E-6B-C5-11	LAX
192.168.1.8	40-F0-2F-BA-5D-DD	Not Responding
192.168.1.9	14-D6-4D-A6-D6-2D	Not Responding
192.168.1.10	CC-B2-55-00-4B-2E	ILFS02-PC
192.168.1.11	14-D6-4D-A6-D6-2E	Not Responding
192.168.1.12	C8-3A-35-C4-92-7A	ILFS13-PC
192.168.1.13	00-21-00-38-47-15	ILFS12-PC
192.168.1.14	A4-67-06-51-F6-9D	Not Responding
192.168.1.15	Not Responding	Not Responding
192.168.1.16	74-D4-35-28-68-C5	Not Responding
192.168.1.17	5C-AC-4C-71-3B-1F	BLACKY

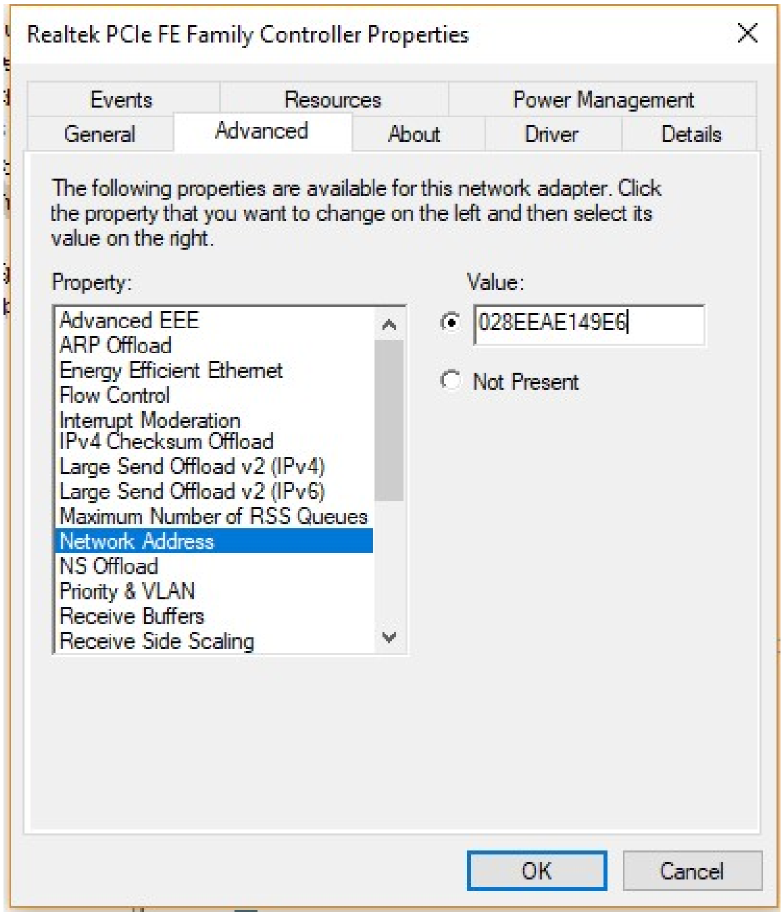
Scan Complete - 255/255

2.4.1.2 The integrity of a MAC Address

A MAC address is usually hard-coded or ‘burned’ into the network hardware; therefore, it is difficult to alter it in this case. However, due to some valid and invalid reasons, a copy of the MAC dress placed in the system software (Figure 8) can be altered as shown in Figure 9. Good reasons for changing a device MAC address include testing out networks for configurations, security applications or new protocols, workarounds and nefarious means, creating wireless connections to a network, changing the function of a single computer from a router to a computer and back to a router through sharing a single MAC address (Cardenas, 2003). It is prudent to note that a MAC address is primarily meant for communication identification. For whichever reasons in changing a MAC address, it leads to the conclusion that a copy of the MAC address in the system software can easily be modified by an attacker to suit the valid MAC addresses spoofed, rendering it not secure. To demonstrate this, the following procedure was used to alter the researcher’s computer MAC address and illustrated in Figure 8 below.

Figure 8

MAC address copy in system software



The following properties are available for this network adapter. Click the property that you want to change on the left and then select its value on the right.

Property: Network Address

Value: ● 028EEAE149E6

Procedure:

i. Open System Properties window

ii. Click on Device Manager

iii. Click on the plus sign preceding Network Adaptors list on the dialog that appeared

iv. Select the card whose MAC address is to be altered

v. Right click on the network adaptor and select Properties

vi. Click on the Advanced tab

vii. Click on Network Address option in the list provided

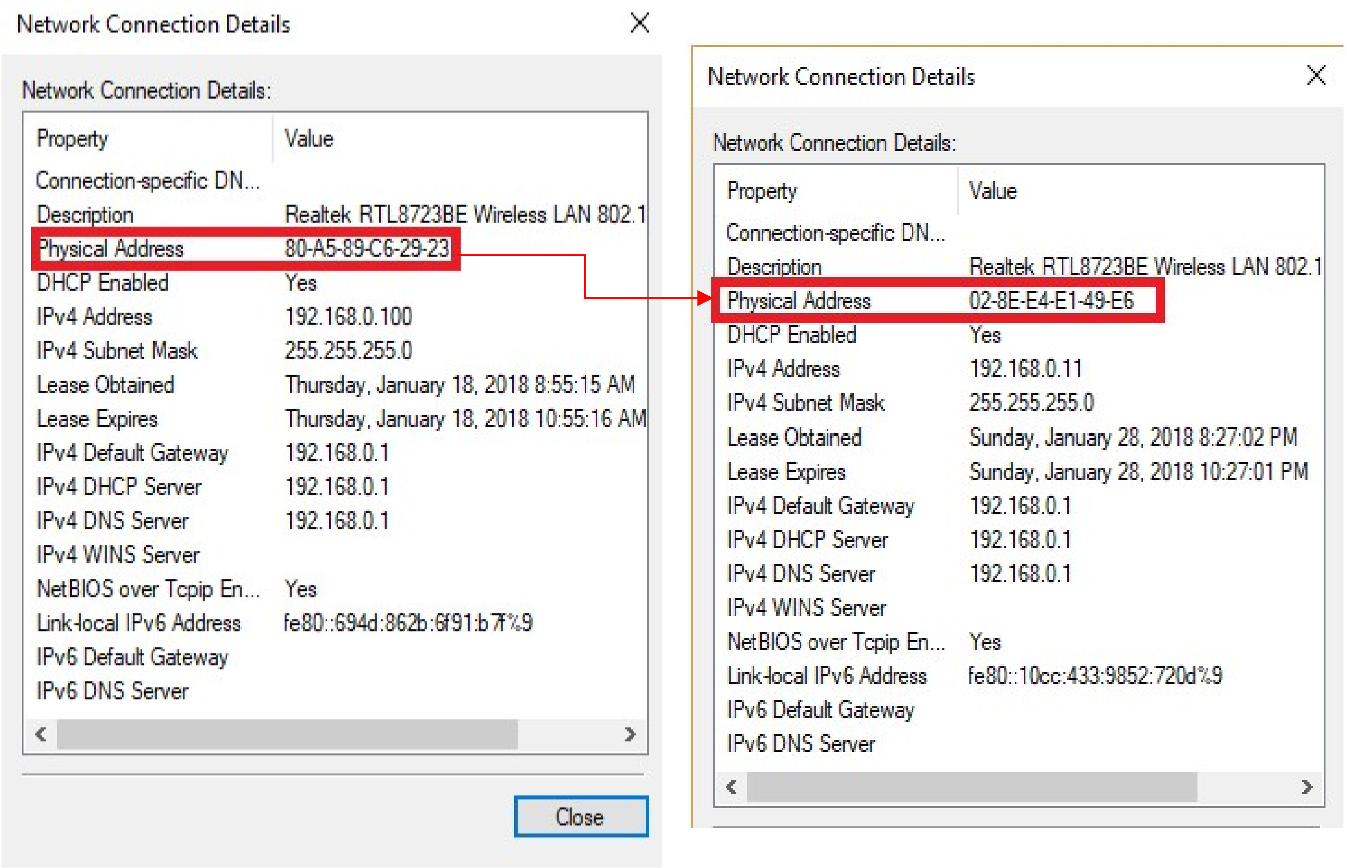
viii. Type the six-digit code in the Value field after selecting the radio button

ix. Click OK on the close icon on the dialog box.

This is further illustrated by the two figures in Figure 9 below. The first part of the figure shows the original MAC address 80-A5-89-C6-29-23 of the researcher's computer before it was altered to another MAC address 02-8E-E4-E1-49-E6 using the procedure described above.

Figure 9

MAC address before and after alteration



Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DN...	
Description	Realtek RTL8723BE Wireless LAN 802.1
Physical Address	80-A5-89-C6-29-23
DHCP Enabled	Yes
IPv4 Address	192.168.0.100
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Thursday, January 18, 2018 8:55:15 AM
Lease Expires	Thursday, January 18, 2018 10:55:16 AM
IPv4 Default Gateway	192.168.0.1
IPv4 DHCP Server	192.168.0.1
IPv4 DNS Server	192.168.0.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::694d:862b:6f91:b7f%9
IPv6 Default Gateway	
IPv6 DNS Server	

Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DN...	
Description	Realtek RTL8723BE Wireless LAN 802.1
Physical Address	02-8E-E4-E1-49-E6
DHCP Enabled	Yes
IPv4 Address	192.168.0.11
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Sunday, January 28, 2018 8:27:02 PM
Lease Expires	Sunday, January 28, 2018 10:27:01 PM
IPv4 Default Gateway	192.168.0.1
IPv4 DHCP Server	192.168.0.1
IPv4 DNS Server	192.168.0.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::10cc:433:9852:720d%9
IPv6 Default Gateway	
IPv6 DNS Server	

Both results were viewed through the following procedure:

i. Click on Control Panel then Network and Sharing Centre

ii. Click on any one of the active networks

iii. Click on the Details button on the general tab of the active network window

iv. Which leads to Network Connection Details shown in figure 9 above

2.4.1.3 Availability of a MAC Address

Availability aspect of security as defined earlier on (Paulsen & Byers, 2019), refers to the accessibility and usability of an identifier upon demand by an authorized user. Availability ensures that the identifier works properly and that its service is available to valid users when needed. In ideal scenarios, a MAC address is usually made available by having it encoded to the network hardware (as in figure 10) as well as having a copy in the operating system as illustrated in Figure 8. However, the effect of the possibility of altering a MAC address compromises the availability of a MAC address. Figure 10 below illustrates the list of MAC addresses the researcher's computer contains that were obtained from the operating system's system32 cmd file using the getmac command at the command line CMD interface.

2.4.2 Robustness of a MAC Address

Robustness characteristic of an identifier refers to the ability of an identifier to function or continue functioning well in unexpected situations (Microsoft Corporation, 2002). Closely related to robustness characteristic of an identifier, are performance and reliability characteristics. The questions that lead to the conclusion on whether a MAC address is robust or not include; does the MAC address remain invariant over a period of time? Is a MAC address reliable? is it able to function as intended over a given period of time under specified conditions?

The answers to these questions are based on fact that the initial intention of encoding MAC address in the network hardware is to make it independent from the operating system supposedly rendering it permanent and hard to alter (Cardenas, 2003). However, due to some valid and invalid reasons, as stated earlier in the integrity of a MAC address section of this study, a copy of the MAC address in the operating system can be altered as shown in figure 9. Whether the MAC address is changed for good or bad reasons, it leads to the conclusion that a MAC address is not permanent and therefore unreliable.

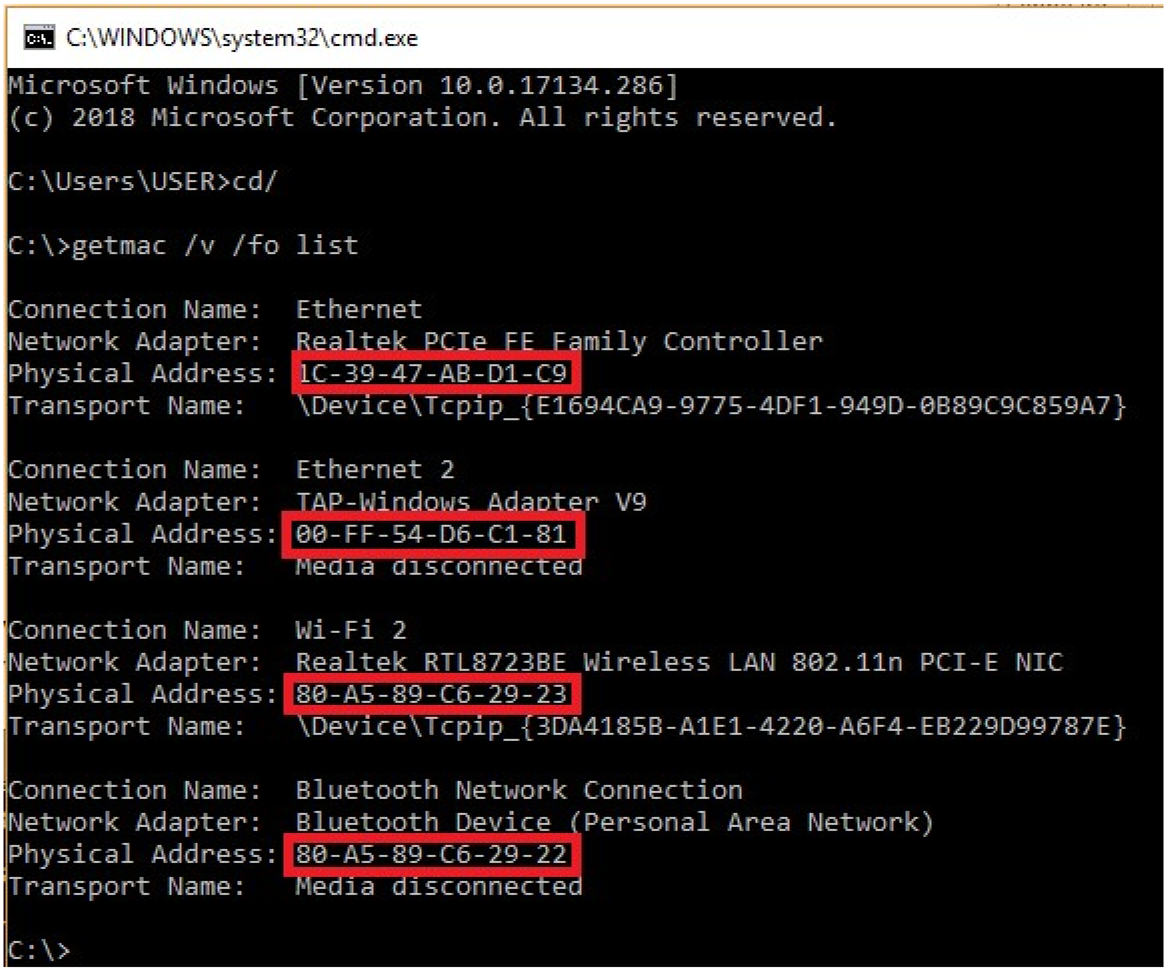
2.4.3 The Uniqueness of a MAC Address

The fact that the MAC address is assigned to each network interface controller (NIC) card by the manufacturer makes it unique only to that particular interface. Furthermore, vendors are given a range of MAC addresses that can be assigned to their products by the IEEE (Iwaya, 2015). This way, MAC address assignment is controlled in such a way that no different adaptors can have the same address even if they are from different manufacturers. However, a device can have more than one network interface card hence even though MAC address can actually uniquely identify a network interface, it doesn't necessary uniquely identify a device.

One case in mind is an instance where a networked computer could contain multiple interfaces. A computer for example, could have one interface for a Wi-Fi, another for Bluetooth and yet another for an Ethernet adaptor. As illustrated in Figure 10 below, a node then can contain several MAC addresses. In this particular case, for instance, the node in in the figure contains four interfaces with corresponding four MAC addresses. This was obtained by running the getmac command on the command prompt of the computer in question. The question is; which of the four can uniquely identify the computer? The answer is none of them. A device cannot have multiple identifiers. A MAC address, therefore does not uniquely identify a computer, rather, it can uniquely identify an interface.

Figure 10

One computer with a number of MAC addresses



```
Microsoft Windows [Version 10.0.17134.286]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>cd/

C:\>getmac /v /fo list

Connection Name:  Ethernet
Network Adapter:  Realtek PCIe FE Family Controller
Physical Address: 1C-39-47-AB-D1-C9
Transport Name:   \Device\Tcpip_{E1694CA9-9775-4DF1-949D-0B89C9C859A7}

Connection Name:  Ethernet 2
Network Adapter:  TAP-Windows Adapter V9
Physical Address: 00-FF-54-D6-C1-81
Transport Name:   Media disconnected

Connection Name:  Wi-Fi 2
Network Adapter:  Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC
Physical Address: 80-A5-89-C6-29-23
Transport Name:   \Device\Tcpip_{3DA4185B-A1E1-4220-A6F4-EB229D99787E}

Connection Name:  Bluetooth Network Connection
Network Adapter:  Bluetooth Device (Personal Area Network)
Physical Address: 80-A5-89-C6-29-22
Transport Name:   Media disconnected

C:\>
```

The possibility of a MAC address being spoofed is yet another case of a MAC address that makes it not to uniquely identify a device. If a device MAC address is altered for whatever reason, the likelihood of another device having the same address is imminent. As such, it cannot be assumed that a MAC address definitely identifies the device uniquely.

From the test runs and discussions on the nature of a MAC address as network device identifier, it can be concluded that a MAC address is not unique, not secure and not robust

2.4.4 Weaknesses of a MAC Address as a Network Device Identifier

Although MAC address filtering can go a long way in identifying and controlling devices in a network, it comes with a myriad number of challenges

One of the security flaws in using MAC filtering is spoofing (Lee, 2010). It comes from the fact that MAC addresses are compiled, maintained and distributed to all APs (Wallace, 2018). Wallace, (2018), further observes that the same MAC addresses are usually printed on the face of the card making it visible for spoofing. To add salt to an injury, Gill & Dahiya (2017), observes that MAC addresses are not encrypted thereby increasing the vulnerability of the address to spoofing.

Cardenas, (2018), gives a number of examples of good reasons why MAC addresses are allowed to be spoofed. One legitimate reason is for creating wireless connections to a network or changing the function of a single computer from a router to a computer and back again to a router through sharing a single MAC address. This is true in a case where if the network has only one public IP address, where one can only hook up one unit directly, but if there are two public IPs, the MAC addresses of the devices must be different. It then becomes easier to change the MAC addresses rather than the NIC cards. Apart from being used for troubleshooting network issues, system problems, testing network management tools and testing intrusion detection, other legitimate reason for MAC spoofing (Gardenas, 2018), is used in wireless network testing. This is so when a user just wants to spoof their own machine MAC address. If it is for illegitimate reason, then an intruder may change the MAC address of their station to enter a target network as an authorized user.

The compilation, maintenance, and distribution of MAC address to APs as stated by Wallace, (2018), apart from creating loopholes for spoofing, leads to another challenge: the limit of the number of MAC address that can be allowed in an AP. It is for this reason that MAC address filtering is not recommended for large wireless networks

Closely related to the problem of spoofing is that MAC address can be altered. The genesis of this weakness is from the fact that WLAN adaptors allow applications to set the MAC address therefore easy to spoof therefore allow unauthorized access to the network (Singh & Sharma, 2015). The flat structure of a MAC address as well enables intruders to use the easy to case numbers, alter their address to reflect a legitimate device, therefore can easily gain access to the network.

Rather than being associated with the device, the MAC address is usually linked to a particular system logic board (Apple Inc., 2011). While the address is invariant over the device life span, should the particular system logic board be replaced for whatever reasons, it changes. Devices relying on MAC address for identification would, therefore, be compelled to use other alternatives

The weakest point of using MAC address perhaps lies on its inability to uniquely identify a network device. As stated by Kurose & Ross, (2013), if a device has more than one interface, then the device has more than one MAC address.

In short, the following MAC address limitations can be noted from the discussions

i. A MAC address copy in the system software can be spoofed thereby pausing as security threat from intruders who may change the MAC address of their station to enter a target network as an authorized user

ii. A MAC address is not reliable or robust. It can be changed and remain variant over time due to spoofing

iii. A MAC address is not unique to a device, but unique to an interface card. A device could have several network interfaces that results to several MAC addresses.

2.5 Suitability of a Computer's Serial Number as a Network Device Identifier

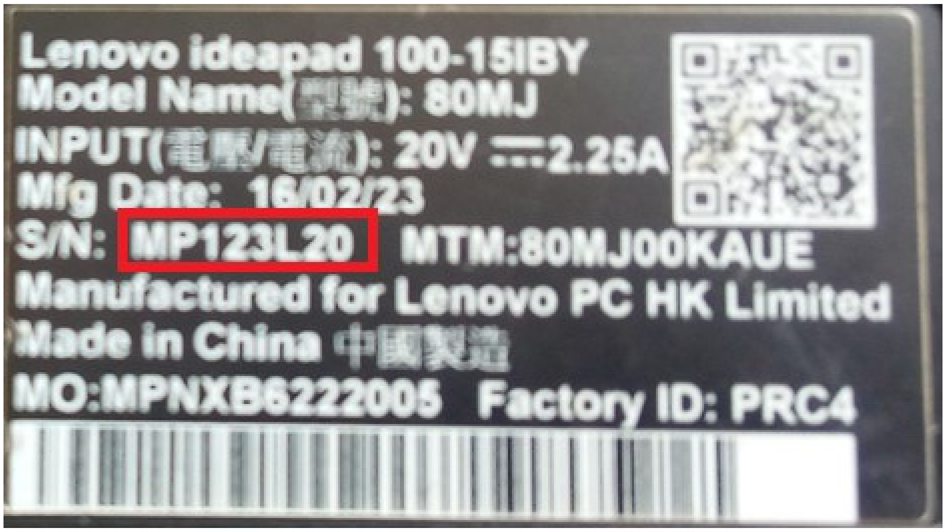
This section presents the suitability of a computer's serial number network device identifier on security, robustness and uniqueness factors as compared to a MAC address.

2.5.1 Computer's Serial Number Location

Just like in any other product, a computer has its serial number tagged as part of the serialization of the product. Perhaps the only extraordinary thing about a computer's serial number is that the number is placed strategically on the computer to simply frustrate snoopers from finding it to indicate the importance of securing the serial number. One would, therefore, more than often find it usually tagged beneath the computer or tagged somewhere beside it. Figure 11 below shows an illustration of a laptop model details that include the serial number in a tag.

Figure 11

Serial number tag on a laptop model



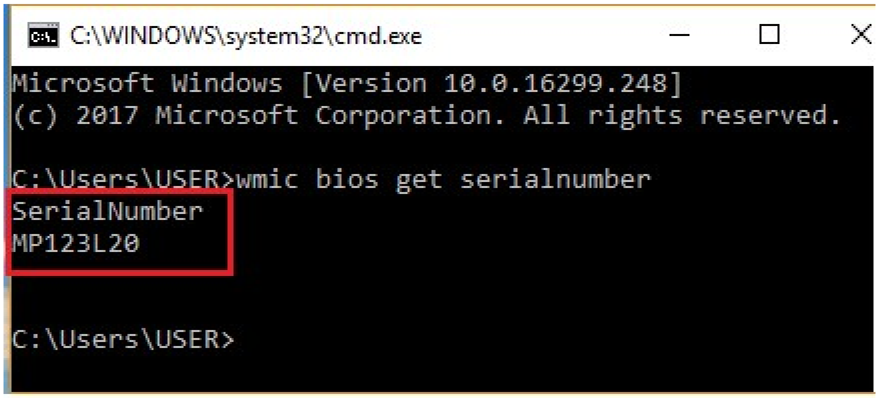
Lenovo ideapad 100-15IBY
Model Name(型號): 80MJ
INPUT(電壓/電流): 20V ⎓2.25A
Mfg Date: 16/02/23
S/N: MP123L20 MTM:80MJ00KAUE
Manufactured for Lenovo PC HK Limited
Made in China 中國製造
MO:MPNXB6222005 Factory ID: PRC4

Although tagging of computer serial number is the norm to serializing computers, it is a practice common to products including computers. This way, identifiers that use scanners such as bar code readers can be used to capture their identification details.

Alternatively, modern laptop models have their serial numbers coded into their basic input-output (BIOS) chips. This makes it possible for the identifier to be accessed using a software and so, it can be processed for a given desired function. The first line of accessing a computer's serial number is generally by running the command wmic bios get serial number at the systems command line interface. The serial number for the author's laptop, for instance, can be obtained as shown in Figure 12 below.

Figure 12

Serial number of a laptop obtained from system BIOS using the wmic command



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\USER>wmic bios get serialnumber
SerialNumber
MP123L20


C:\Users\USER>
```

The encoding of a serial number in a computer’s hardware rather than just tagging it on a surface makes it possible to access and manipulate the serial number. This way, it becomes possible to develop a model that can access the serial number internally and use it to authenticate the device in a network.

2.5.2 Uniqueness, Security and Robustness of a Computer’s Serial Number

The reason why a MAC address can be spoofed and therefore altered is because of a copy a MAC placed in system software for valid manipulation. Valid reasons for allowing a device MAC address to be changed include testing out networks for configurations, security applications or new protocols as stated earlier. A MAC address after all is mainly meant for communication.

However, the mere fact that a computer’s serial number is only hard-coded on the hardware without a copy in the system software alone renders it hard to be spoofed. This is demonstrated in figure 5, figure 6 and figure 7 where other computer’s details such as MAC address and IP address can be spoofed but not the computer’s serial number. The MAC and IP address values that can spoofed are the copies in the system software. If it cannot be altered, then a computers serial number is remains invariant, it is permanent, therefore robust. Therefore, a computer's serial number is secure, unique and robust.

It is only in some rare cases that the serial number can be altered. But this requires that a computer system has to turned off, any power lines disconnected, any static electricity discharged, computer case opened, disconnect the CMOS battery, wait for roughly 30 seconds (to completely ensure that the CMOS power is completely drained) then the process is done in reverse to revert back to original state (Derekyoung, 2017). This way, all original CMOS settings such as custom CMOS settings, BIOS password, time and date, as well as the motherboard serial number are lost. The system then generates a new system data that includes a serial number when booted.

It is worthy to note that serial numbers are assigned by manufacturers of computers according to their own standards. As such, the numbers come in different formats and therefore may only be unique to a specified manufacturer. Table 1 below shows various forms of serial numbers from some sampled laptops.

Table 1

Serial number formats by some manufacturers

#	Model	Name	Serial Number
1	Dell	TIMKORIR-PC	C2J9CN1
2	Dell	VINCE	3P077R1
3	Dell	SHEPHERD	DMV2FS1
4	Lenovo	CHEBOR	MP123l20
5	Lenovo	DESKTOP-400QDST	R90NFZBY
6	Lenovo	LAPTOP-DHDA4587	PF0WCLAK
7	HP	ADMINRG-5CSAP7G	MXL1422Q36
8	HP	DESKTOP-M4HV1N7	CNV4359PXT
9	HP	DESKTOP-088CEV5	CND6032KOK
10	Samsung	USER	JHA791HD900151
11	Samsung	BENJAA-PC	ZYN693NB100411
12	Samsung	VICK-PC	HTXP91RD710085A
13	Acer	EMANUEL	NUSGAEM0192191040A714
14	Acer	KLAW-014	PSVC1E90481460919B9600

The table illustrates the uniqueness quality of a serial number in the sphere it belongs to. In addition to this, it can be noted that serial numbers structure varies from model to model even within computers of the same model. Although the structure could greatly make it difficult for intruders to spoof for the values, it violates the mnemonic characteristic of an identifier as discussed earlier on in the study. Mnemonic factor of an identifier is crucial especially when it comes to memory allocation usage. A variable character or VARCAR does not optimize memory allocation unlike a standardized character set that uses CHAR datatype to optimize memory allocation usage. A remedy to mnemonic of a serial number is suggested in the recommendations section of the study

2.5.3 Role of Serial Numbers in Communication Devices Currently

The serial number also referred to as the manufacturer's serial number (MSN) is defined as a number assigned by a manufacturer to a unit of product for identification. It is this key role of device identification that leads to other tertiary roles of serial numbers namely device tracking, deterring device theft and counterfeiting, device quality control as well as activating aliases (Apple Inc., 2011).

Some examples of famous identifiers of products include International Serial Book Number (ISBN) for books, International Mobile Station Equipment Identity (IMEI) for Global Systems of Mobile (GSM) phones, Manufacturer Part Number (MPN) for products to their manufactures and Digital Object Identifier (DOI) for digital contents, such as e-book, a journal article or music (Wang, 2007). Identifiers defined by an organization include Personal Identification Number (PIN), Social Security Number (SSN) and Product Batch Number (PBN)

A device serial number plays a key role in device identification as noted by Apple Inc., 2011 and echoed by ANSI/NISO, (2013). Apart from using the serial number to identify devices, the serial number can be used to activate devices' aliases (Autodesk, 2010). Device alias are other attributes associated with the device that might include device details, device user details, device program details or even device parts details. Extra information about a device is critical in associating the identity of the device with these other attributes for a contextual description of a device.

When well-armed with the additional necessary information of a device, then it easy to asset-track and trace the system remotely (Apple Inc., 2011). This is mostly applicable in supply chain management (Lehtonen et al., 2008) and device inventory management.

Serial numbers are actually used as a measure to counteract counterfeiting. As Lehtonen et al., (2008) put it, manufactures of devices keep a list of valid device ID numbers in a secure online server so that the absence of a device ID from the list indicates a counterfeit.

2.6 Identification Details Capture Technologies

Major identification technologies began way back during the Second World War by use of identification friend or foe (IFF) (Lehtonen et al, 2008). From then on, advances in identification techniques took effect. This section discusses barcode readers, optic character recognition, biometric identification and radio frequency identification systems.

2.6.1 Bar Code Readers

Although it dates back to 1940s when it was first developed, but was later used after the Universal Product Code (UPC) was adopted (Su *et al*. 2007), bar code readers still find their applicability in auto-identification of goods in supply chain management (SCM) (McCathie & Michael, 2005). This includes the identification of items right from acquisition of raw materials through processing to delivery of such goods to the end user in what is commonly referred to as the point of sale.

A sensor in the bar code scanner detects the reflected light from the illuminating system, usually a red light, and generates an analog signal that is sent to a decoder. The decoder interprets the signal, validates it using a check digit and converts it to text.

Barcode readers are the simplest of all identifiers since they constitute a paper, ink and a scanner (McCathie, L. and Michael, K., 2005) and therefore cost less, easy to use, reliable and use line-of-sight. However, bar code readers come with a number of limitations. A bar code reader cannot be used to read text directly as only numbers can be coded. They are relatively fixed and therefore unsuitable for recording items whose details keep changing over time. Bar code readers as well can only be read and interpreted by machines but not human eyes

2.6.2 Optical Character Recognition

Mohammad, *et al*. 2014, defines optical character recognition (OCR) as a technology that converts a printed document or scanned the page into ASCII characters that a computer can recognize.

To totally perform its functionality, the OCR system constitutes four modules; image acquisition module, pre-processing module, and feature extraction module and pattern generation. The main function of the image acquisition is to obtain text image from a scanner or a pre-stored image file for processing. The feature extraction module is part that plays an important role as part of the system is the process of getting information about an object or a group of objects in order to facilitate classification.

Apart from converting text documents to some sort of digital presentation, Jesse, (2016), cites other examples where OCR is applied are as follows;

i. People who wish to scan in a document and have the text of that document available in a word processor

ii. Recognizing licenses plate numbers

iii. Post Office to recognize zip-codes

iv. Facial feature recognition (for security purposes)

v. Speech recognition

vi. A submarine who wishes to classify underwater sounds

Although OCR and bar code readers are both data capture methodologies, OCR is needed when information should be readable both to humans and to a machine

2.6.3 Biometric Identification System

Biometrics is defined as the science of identifying people using physiological features. Biometric identification system (BIS) then is identification based on biometrics. Although any physiological feature could be used for identification, the most generalized techniques include the automated recognition of fingerprints, faces, iris, retina, hand geometry, voice, and signature (Garzia-Luis *et al*. 2003). A biometric system basically operates by acquiring biometric details from an individual, extracts a feature set from the acquired details and compares the feature set against pre-recorded details of the individual in a template set in a database (Luis-Garzia *et al*. 2013). An individual simply inserts their smart card containing their biometric information in a reader at an identification point, interacts with the sensor (a camera or a scanner), the sensor acquires and processes the information and compares the information with the information on the card for verification. Through this, BIS can actually perform identification as well as verification of individuals.

Luis-Garzia *et al*. (2013), goes on to identify the components of BIS as the sensor, feature extractor, matcher and database modules. Whereas the sensor module captures the biometric details of an individual of interest (fingerprint sensor for instance), feature extractor module processes the extracted details to obtain a set of salient or discriminatory details. Matcher module on the hand is module used to compare extracted details with those stored in the database for recognition. This is the point where the user's claimed identity is confirmed (verified) and therefore the user's identity is established (identification) based on the matching details. System database module contains a list of biometric details of valid or enrolled individuals

Most identification methods such as RFID, barcodes, and OCR basically have their identification credentials tagged or labeled and therefore are based on what is known or owned rather than what they are. As such they are prone physical damages, attacks and can be forgotten. BIS is based on the subjects are. That is why BIS has found its usefulness in individual identification and verification in applications such as computer network login, electronic data security, e-commerce, internet access, ATMs, credit cards, physical access control, cellular phone, PDAs, medical records management systems, distance learning, national ID card, driver's license, border control, passport control, corpse identification, criminal investigation, terrorist identification and parenthood determination to name but a few applications.

2.6.4 Radio Frequency Identification

Radio Frequency Identification (RFID) is a technology that uses radio wave frequencies to automatically identify a person or an object. It comprises transponders, readers or interrogators and an online database also called back-end server (Lehtonen *et al.* 2008)

A number or an identifier for the object is implanted into a microchip (the transponder) that contains an antenna that transforms the identifier and any other implanted information to a reader usually within 5cm to 10m radius coverage. The reader then converts the signal into information that can be sent to a computer to be made usable. RFID thus provides unique and automatic identification of objects does use line-of-sight and transmit signals at a higher rate for a longer distance as compared to barcode readers (Robyns *et al*, 2017). That is why RFID is mostly useful in processing, manufacturing, supply chain management with the aim of fighting counterfeits in products (Lehtonen *et al.* 2008) and stock control applications.

However, RFID lack standards and regulatory requirements (McCathie & Michael, 2005), are mostly passive (draw power from the reader), cannot store passwords securely and are prone to tempering; they are just but tags. Attackers need only to read the product serial number on the tag and programming the same number into an empty tag thus end up reproducing a clone of the product. Furthermore, Lehtonen *et al.* (2008) associate this attack to other attacks such as side-channel attacks, reverse engineering and crypto-analysis, brute-force, physical, active attacks, and data theft. As a result, Juels *et al.* (2006), enlists cooperate espionage, competitive marketing, infrastructure, trust perimeter, action, association, location, preference, constellation, transaction, bread crumb and cloning as threats to use and deployment of RFIDs

2.7 Algorithm Design Paradigms

Software construction whether it is simple or complex requires approaches that would eventually result in an efficient solution. Such approaches include the greedy algorithm, backtracking, dynamic programming, and divide and conquer methods. Dunne, (2018), enumerates the benefits of such methods as:

i. They provide templates suited to solving a broad range of diverse problems.

ii. They can be translated into common control and data structures provided by most high-level languages.

iii. The temporal and spatial requirements of the algorithms which result can be precisely analyzed.

2.7.1 Greedy Algorithm

The greedy algorithm, as its name implies, is an algorithm that employs iterations in solution provision to problems. The criterion used at each iteration is to maximize the gain or the objective function (Moreno, 2012). This way, the greedy algorithm uses the shortest route or minimum spanning tree to arrive at a solution. Otherwise, a feasible solution can be added into the solution set as illustrated in the algorithm presented below (Balasubramanian & Viswanathan, 2004)

```
Algorithm Greedy (a, n)
// a ( 1 : n ) contains 'n' inputs
{
    solution = 0 // initialize the solution
    for i=1 to n
    {
        x = select (a)
        if feasible (solution, x) then
            solution = union(solution, x)
    }
    return solution
}
```

In the algorithm, the function select is used to select an input from a() and removes it. The selected input's value is assigned to 'x'. Feasible is a Boolean-valued function that determines whether 'x' can be included in the solution vector or not. The function Union combines 'x' with the solution and updates the objective function.

Dunne, 2018, sites examples that greedy algorithm works well such as minimum spanning tree, shortest distance in graphs, Knapsack problem, the coin exchange problem and Huffman trees for optimal encoding. In the Knapsack problem for example with a knapsack of capacity 10 and 4 items given by the < weight :value> pairs: <5:6>, <4:3>, <3: 5>, <3: 4>. The greedy algorithm will choose item1 <5:6> and then item3 <3:5> resulting in total value 11, while the optimal solution is to choose items 2, 3, and 4 thus obtaining total value 12.

Since the greedy algorithm is mainly used in solving optimization problems, they do not always give the best solution. In fact, Moreno, (2010), illustrates that a greedy algorithm can fail to determine a global maximum or minimum result because it could fall in a trap and converge to some local maximum.

2.7.2 Backtracking

Backtracking is an algorithm method used to solve problems with a large search space, by systematically trying and eliminating possibilities. It tries all possible combinations of things to optimize by exhaustive search (Moreno, 2010). At each iteration, possibilities that don't work are discarded and backtrack to other possibilities. A general algorithm for iterative backtracking is presented in the algorithm below (Balasubramanian & Viswanathan, 2004)

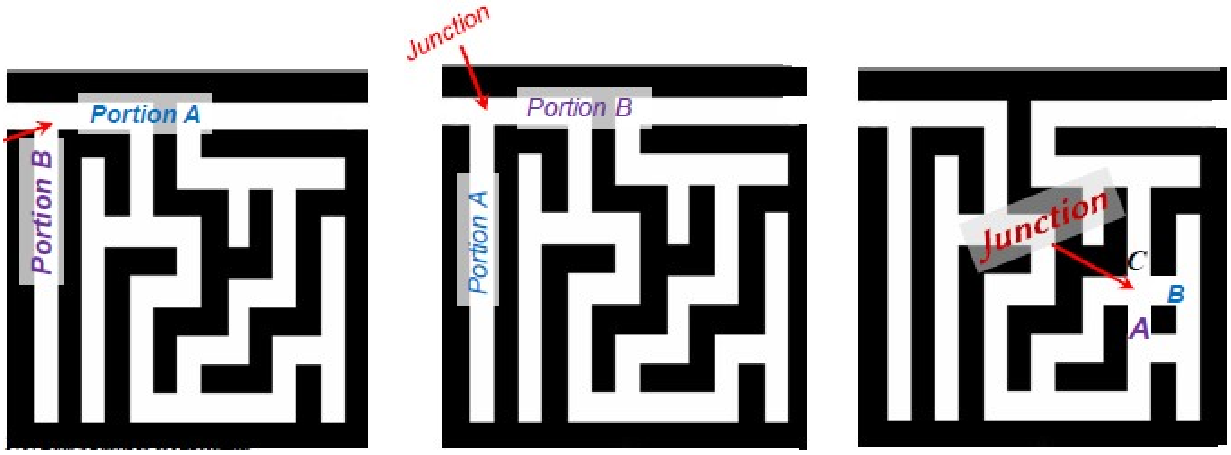
```
Algorithm backtrack (n)
// all solutions are generated in ( 1 : n )
{
k = 1
while ( $k \neq 0$ )
{
if (there remains an untried x(k) גּ⊂(T(x(1), x(2)…x(k-1)) and $B_k$(x(1), x(2)…x(k))is true then
{
if (x(1), x(2)…x(k)) is a solution then
write (x(1 : k))
k=k+1 //consider the next component
}
else
k=k-1 // backtrack to previous component
}
}
```

The function T from the algorithm yields a set of possible values that can be placed as the first component x_1 of the solution vector. Bounding function 'B' is used to check whether the component can be placed in this position or not. The variable 'k' continuously incremented and a solution vector is grown until either a solution is found or no, untried under the value of x_k remains. When 'k' is decremented, the algorithm must resume the generation of possible values of the k^{th} position that have yet been tried.

A good example as illustrated by Tanimoto, (2016), of backtracking through a maze is illustrated in the Figure 13 as follows:

Figure 13

Maze backtracking illustration (Source: Tanimoto, 2016)

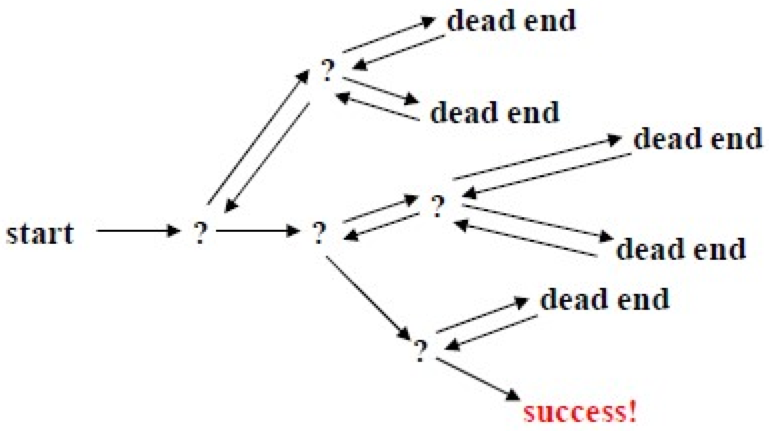


(a) (b) (c)

At some point, one would have two options on the direction to go as in Figure 13 (a). if one chooses portion B, then it leads to a dead end, therefore backtracks and uses portion B (Figure 13 (b)) to have a way out. The process is iterated at every junction in such a manner that if possibility taken reaches a dead end then backtracks, otherwise search progresses to the end.

Figure 14

Backtracking search tree (Source: Tanimoto, 2016)



If all choices made never lead out, then it implies that there is no solution to the problem.

The goodness of backtracking is that it ensures correctness by enumerating all possibilities as well as being efficient by never visiting a state more than once (Skiena, 2008). Furthermore, backtracking is simple to implement and easy of coding and therefore applicable in solving tactical problems. However, the overall runtime of the backtracking algorithm is slow, requires large memory locations for storing different state functions, thrashing problems and detects conflicts way too late (Ferdous, 2017). That is why it is not advisable to use it in solving strategic problems

2.7.3 Divide and Conquer

At times programming problems are too complex or large for them to be understood or solve once. It is at such instance that divide-and-conquer algorithm finds its application (Ferdous, 2017; Husssein, 2013 and Skiena, 2008). Divide and conquer method decomposes the entire problem into two or more disjoint problems recursively, solve them independently then combine the resulting solutions.

The general divide and rule algorithm below, for example, divides ‘n’ inputs into ‘k’ distinct subsets between 1<k<n yielding ‘k’ sub problems. The methods are then solved as well as get a method that combines them into a solution of the whole. If the sub problem is still large, then the divide and conquer can still be further applied. A general algorithm is presented below

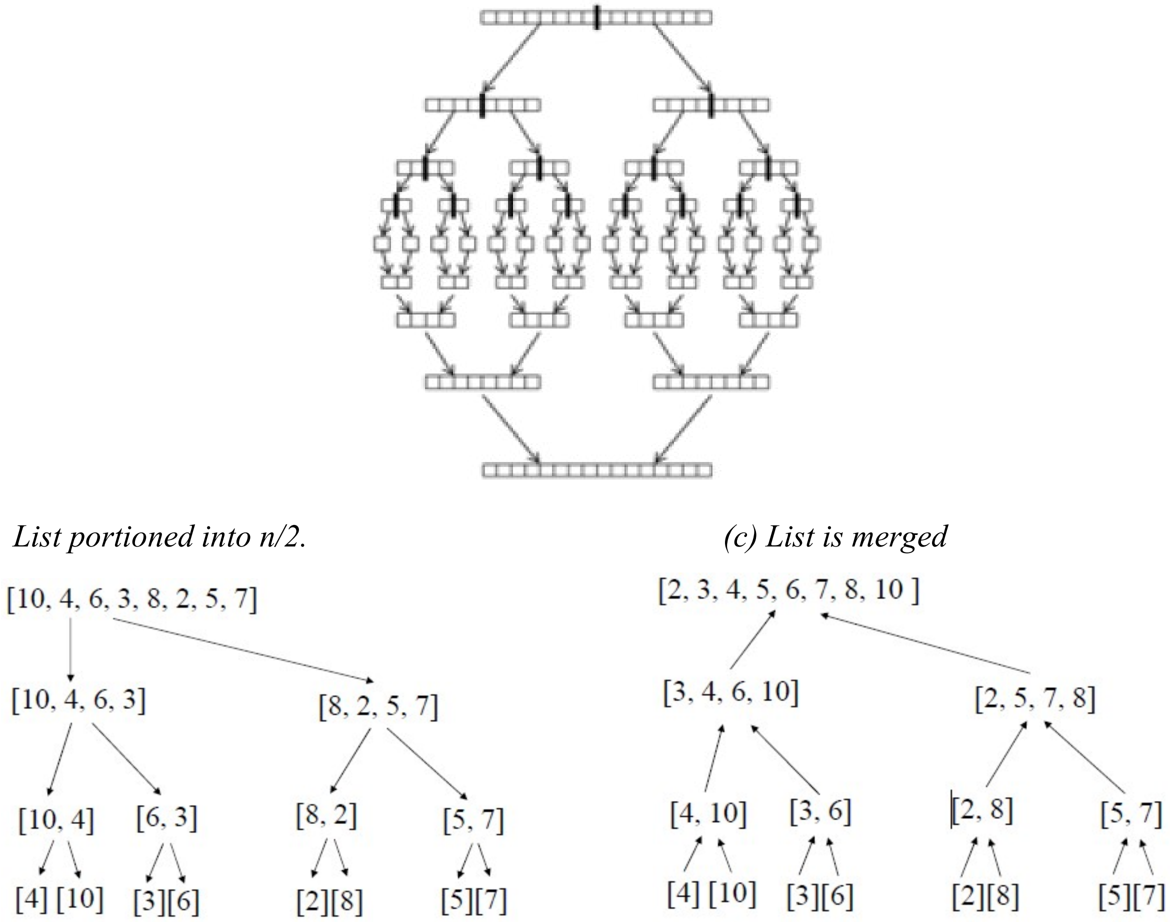
```
Algorithm DandC(P)
{
If Small (P) then return S(P) // problem is small
Else // problem is big
{
Divide P into small instances $P_1$, $P_2$, $P_3$ ... $P_k$
Apply DandC to each of these sub problems
Return combined (DandP($P_1$), ((DandP($P_2$) … (DandP($P_k$))
}
}
```

Whereas ‘P’ is the problem to be solved, small (p) is a Boolean value that determines whether the input size is small enough that the answer can be computed without splitting. If this is so, the function ‘F’ is invoked. Otherwise, the problem ’P’ is divided into smaller sub problems. These sub problems P1, P2, P3….Pk is solved by recursive application of D and C. Combine is a function that determines the solution ’P’ using the solution to the ‘k’ sub problems. If the size of ‘P’ is ‘n’ and the size of the ‘k’ sub problems are n_1, n_2 … n_k respectively.

Zhang, (2016) uses [10, 4, 6, 3, 8, 2, 5, 7] merge sort list as an example to illustrate how to divide and conquer algorithm works. The array is first divided into two halves, are recursively sorted and finally merged as shown in the illustration in Figure 15 below.

Figure 15

Divide and Conquer algorithm illustrated (Source: Zhang, 2016)



(a) generalized merge sort

(b) List portioned into n/2.

(c) List is merged

Examples, where divide and conquer applies, include insertion sort, topological sorting, binary tree traversals, computing the longest path in a binary tree, computing Fibonacci numbers, reversing a queue and Warshall's algorithm. This divide and conquer can solve complex problems as well as utilize memory cache effectively. However, it is a complicated and slow algorithm method.

2.7.4 Dynamic Programming

The algorithm designs are meant to optimize solutions to various problems. Greedy algorithms, for example, make the best use of local decision at each stage and therefore efficient, but they do not guarantee global optimality (Skiena, 2008). Backtracking as well tries all possibilities and selects the best that optimizes results. However, this exhaustive search technique incurs overheads in terms of complexity.

Unlike in divide and conquer where the sub problems are divided into pairs, solved simultaneously and resolutions combined recursively, which can result into duplicate computations, dynamic programming optimizes solutions on a step by step basis recursively (Paramalways, 2009). Furthermore, Skiena, (2008), states that dynamic programming systematically searches all possibilities while storing results subsequently to avoid re-computing. The results of one step solve the problem of another consecutive step recursively. This results in the implication that the decision taken in one step is dependent upon an immediate previous or later decision steps.

Solutions vary from 1-dimensional, 2-dimensional, interval or tree dynamic programming (DP). According to Park, (2015), a DP problem is solved by defining the problem, writing down the recurrence that relates to the problem and finally recognizing and solving the base cases in that order.

As a result, dynamic programming is inherently the right method for optimization of combinations of ordered interdependent solutions. Character strings, rooted trees, polygons, and integer sequences are an example of instances that can effectively be solved by dynamic programming.

2.8 System Models

Systems under construction use models instead of experimenting on realities at their early stages of development due to the fact that realities are too complex to exactly reproduce them. One would, in fact, realize that much of the real complexities are actually irrelevant for a particular solution. A model, therefore, according to Feinberg, (2009), is a simplified representation or an abstraction of reality.

Modeling systems instead of constructing a real system are beneficial in many ways. Apart from model manipulation allowance, models enable time compression, low construction, analysis, and execution cost can model risks and uncertainties as well as model large and extreme complexities with finite solutions and finally enhances and reinforces learning and training (Problemistics, 2017). Despite the numerous benefits that come with modeling systems, Singh (2017), lists the limitations of models as

i. May not always be practical and therefore lead to delays in decision making
ii. Nature of inter-dependence is not defined
iii. Classification of the interrelationships has to be done properly so that it could be productive
iv. May not cater for contingencies or a particular style of functioning in the organization
v. May not be applicable to smaller organizations. The theory assumes that most of the organizations are big, complex and open systems

Models come in different types and forms. The widely used structures are iconic, mathematical and analog models

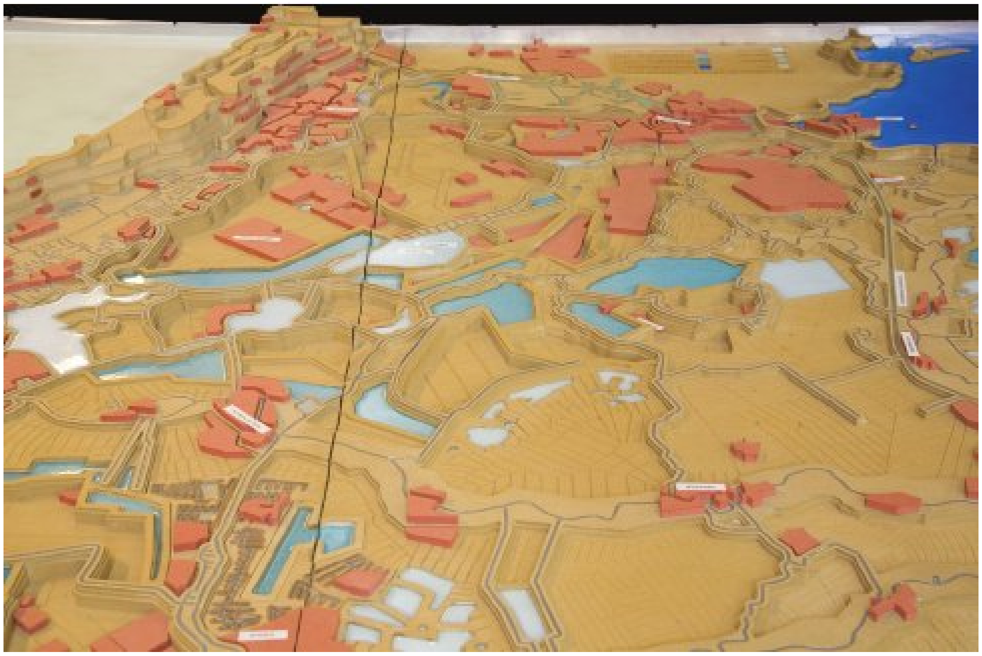
2.8.1 Iconic (Scale) Model

Also referred to as a scale model, the iconic model is a physical replica of a system usually represented by scaling the original system. Models of buildings are scaled down, reproduction models such as copies and prototypes or a working model have the same scale and models of such elements as an atom are scaled up (Problemistics, 2017). An iconic model may be in three-dimensional such as a building (scale model) and a car or a two-dimensional icon such as photos and drawings. This way, iconic or scale modeling finds its application in engineering, architecture, filmmaking, military command, salesmanship and hobby model building.

Figure 16 below illustrates an example of a spatial semi-iconic landscape model proposed to improve a water system

Figure 16

A spatial semi-iconic landscape model (Source: Nijhuis, 2011)

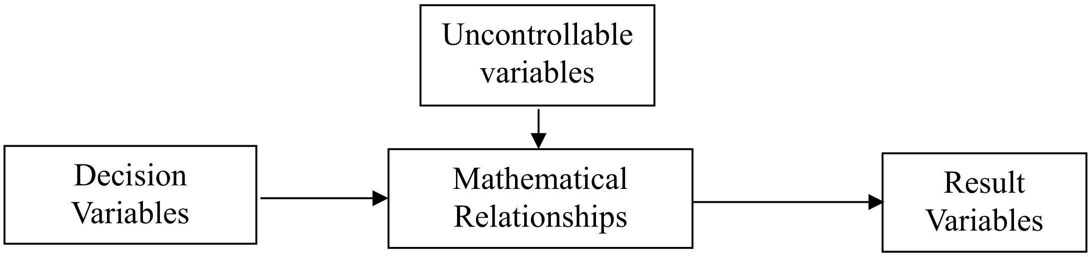


2.8.2 Mathematical (Quantitative) Model

Mathematical or quantitative models use mathematical relationships to represent realities. It involves the identification of system factors (variables), the establishment of equations describing the relation between the variables, simplification of the problem through assumptions and balancing the model simplification and the accurate representation of reality (Feinberg, 2009).

Figure 17

Mathematical model illustrated (Adapted from Feinberg, 2009)



A mathematical model constitutes the system variables, uncontrollable variables, result, variable and the mathematical relationship itself. System variables describe alternative courses of action input into the system that can be controlled. Uncontrollable or constant variables are the variables that are generally part of the system environment thus affect the result variables. Some of the uncontrolled variables constrain the system outcome and therefore referred to as the constraints. The result variables reflect the level of effectiveness of the system in the form of an output. Of course, the relationship between the system variables is realized through the mathematical relationship or a mathematical formula/expression.

Quantitative models are used when the reality is too abstract to be represented using iconic or analog models and when the variables factors (variables) of the system can be represented by a symbol that can be manipulated in a meaningful and fruitful manner (Problemistics, 2017). They can be manipulated easily for experimentation as well as predictions. That is why the model is mostly used in most management science analysis to aid decision making in business processes (Sindhuja, 2015).

2.8.3 Analogue (Symbolic) Model

Whereas the iconic model replicates a system and mathematical model uses expressions to represents realities, analog or symbolic models behaves like the real system but does not look like it (Feinberg, 2009). It just represents the entities of a system symbolically through the use of diagrams and illustrations.

Just like in iconic modeling, analog models can be built in two dimensions; two-dimensional visualization is applied in charts, diagrams, and graphical representations while three-dimensional visualization applies in natural or analog devices (Problemistics, 2017)

Marchese, (2013), structures analog models into context, interaction, structural and behavioral models. Each of these models constitutes one or two of activity, use case, sequence, class and state diagrams unified modeling language (UML) structures. UML is a widely accepted graphical notation for representing designs especially object-oriented systems (Bernstein & Braude, 2011)

i. Class Diagrams

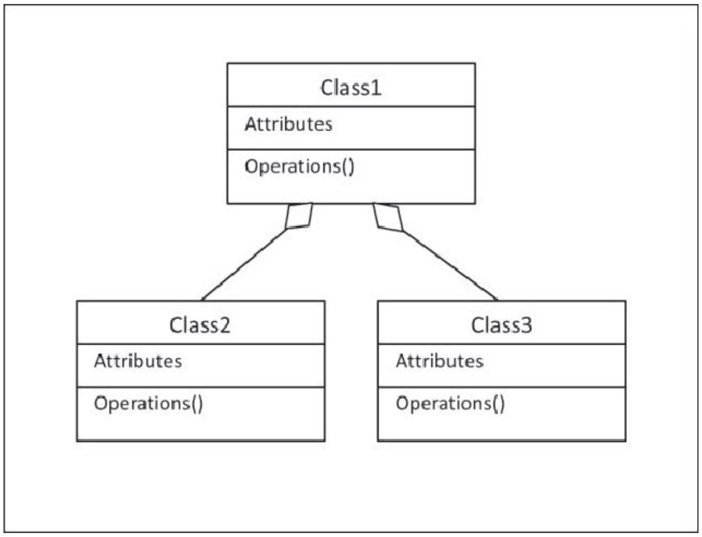
Class diagrams, in short, describe the structure of a system by exhibiting the system's classes, their attributes, and relationships among the system classes. A class according to Bruegge &

Dutoit, (2010), is an abstraction of a set of objects with the same attributes, operations, relationships, and semantics. The objects or entities could refer to people, things of concern to an organization or data. Apart from illustrating how different objects relate to each other, class diagrams, in equal measure, can be used to show the implementation of classes (Bell, 2012)

A class in the class diagram is represented by a rectangle divided horizontally into three sections. The upper part contains the class name, class attributes are listed in the middle section and class methods or operations are lined up at the lower section. Relationships between different classes are indicated with either a line with an arrowhead at the top pointing to a supper class for an inheritance, a solid line for class associations known by the associating classes or an arrow with an open arrowhead if the association is only known by one of the classes. Figure 18 below illustrates a generalized class diagram relating a class2 and class3 class to a supper class1 class details.

Figure 18

A class diagram (Source: Karasneh & Chaudron, 2013)



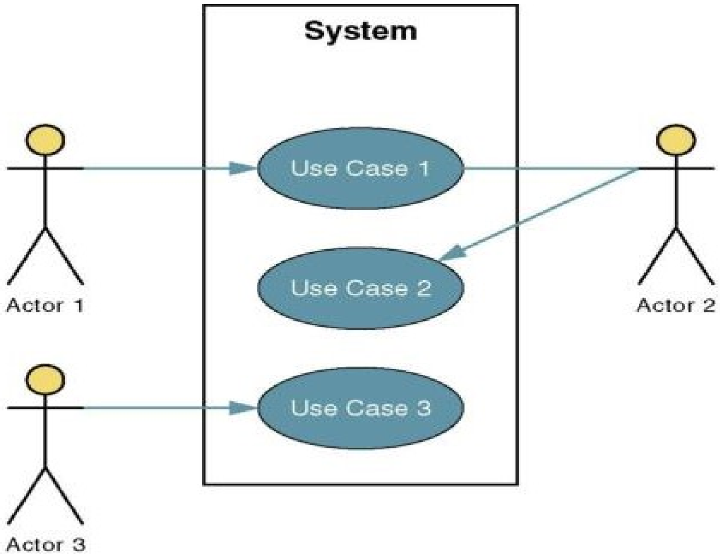
ii. Use Case Diagrams

Use case diagrams are used to decompose the problem or solution at hand into concrete functional units to show the interactions between a system and its environment (Marchese, 2013). This is the model that usually applied during requirements elicitation stage in software engineering.

Use case models use terms such as *actor* to denote a person any other system (or anything) that interacts with the system. Usually represented by an oval, the *use case* is a term used to denote a function, a workflow, a major process. It is the *use case* component that drives the actual design process. Bruegge & Dutoit, (2010) mentions that the identification of actors and use cases results in the definition of the boundary of the system in differentiating the task accomplished by the system and the tasks accomplished by the environment. Figure 19 illustrates a use case in which three actors external to a system acts on the system with three use cases.

Figure 19

A use case diagram (Source: Choi, 2014)



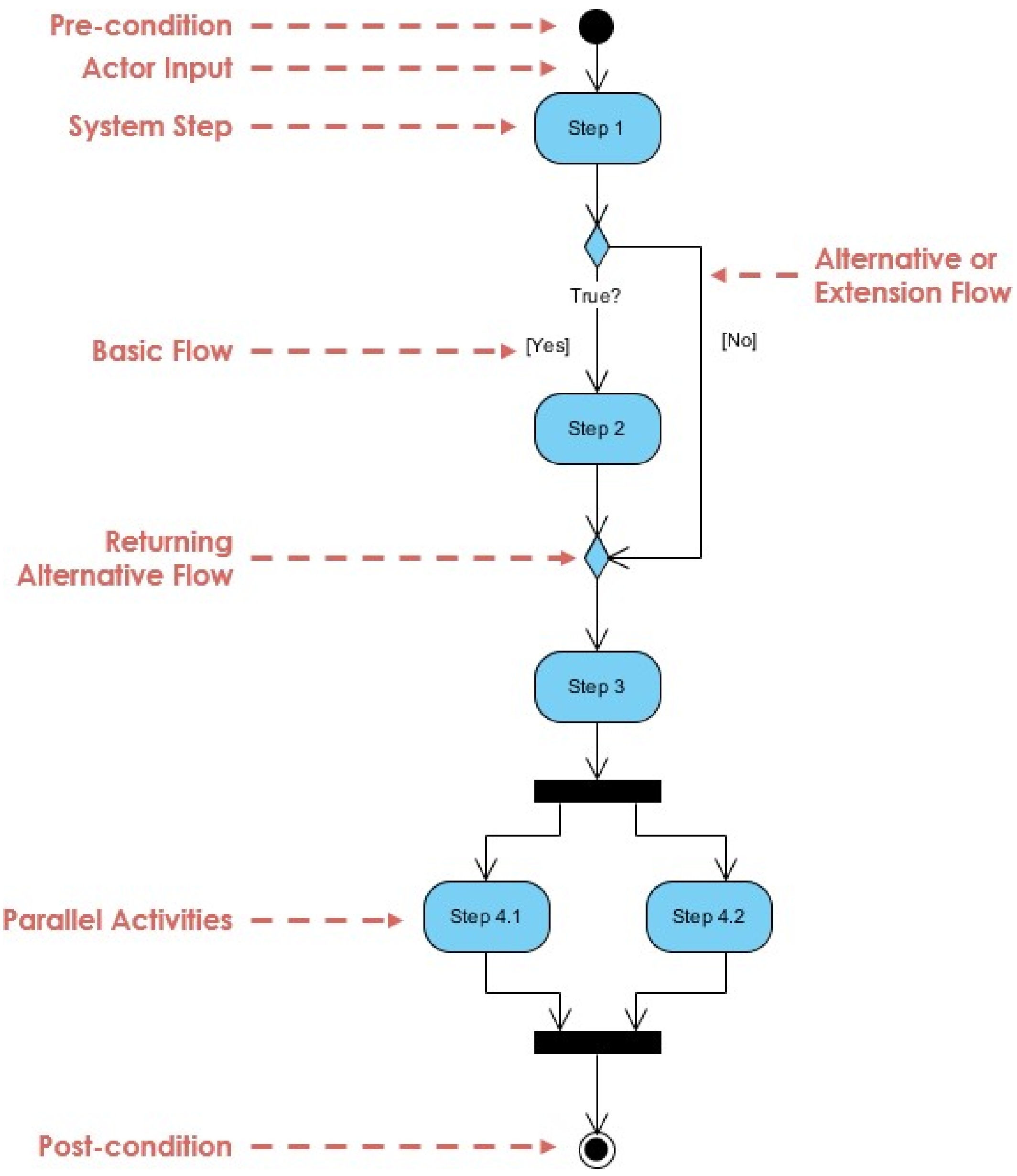
Use case generally allows models to be represented diagrammatically to provide an overview of the use case and in a more detailed textual form

iii. Activity Diagrams

Activity diagrams describe the behavior of a system in terms of activities going on (Bruegge & Dutoit, 2010). Activities are the elements that represent the execution of a set of operations triggered by either the completion of other activities, availability of objects or by external events. The activity diagram in Figure 20 illustrates how activities in an activity diagram progress from one step to another step.

Figure 20

A Basic Activity Diagram (Source: Visual-paradigm, 2018)



An activity diagram is used to represent control flows through flowcharts by.

i. Identifying candidate use cases, through an examination of business workflows
ii. Identifying pre and post-conditions for use cases
iii. Model workflows between/within use cases
iv. Model complex workflows in operations on objects

v. Model in detail complex activities in a high-level activity diagram

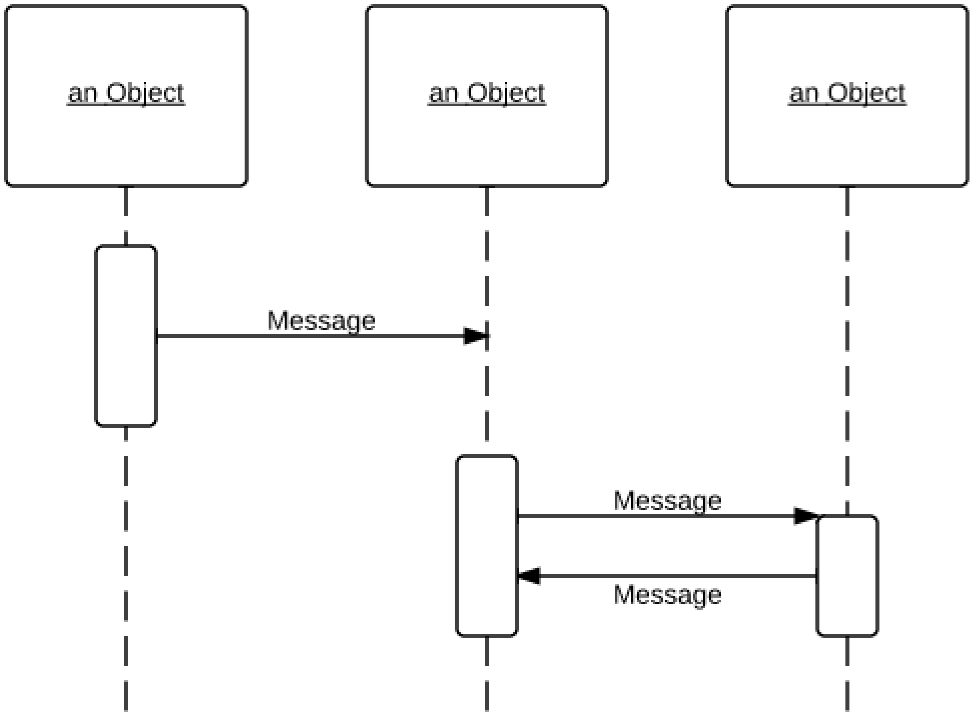
iv. Sequence Diagrams

Sequence diagrams model the interactions between the actors and the objects within the system. It shows the sequence of interactions that take place during a particular use case or a use case instance. It shows a representation of how events cause flow from one object to another as a function of time. Expressed in two dimensions, the vertical dimension of a sequence diagram indicates the sequence of a message in time order of occurrence while the horizontal dimension shows the object instances to which the messages are sent.

A sequence diagram is drawn by identifying the objects (class instances) and putting each class inside a rectangle across the top of the diagram. The name of the objects/class is expressed in the conventional object-oriented format as illustrated in the figure below. Message flows are either represented by an open arrow pointing to the receiving object or a dotted line with the arrowhead pointing back to the originating object. In both cases, the message labels are written above the line. Figure 21 below shows generalized sequence diagram of three objects and three messages.

Figure 21

A sequence diagram (Source: lucidchart, 2019)



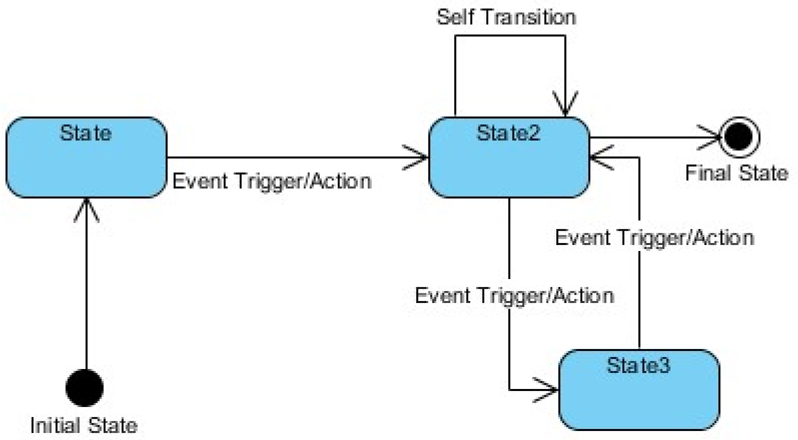
Sequence diagrams are mostly used to represent key classes and the events that cause the behavior to flow from class to class (Pressman, 2005)

v. State Diagram

State diagrams model the dynamic behavior of an individual object as a number of states and the transitions between these states (Bruegge & Dutoit, 2010). A state, in this case, represents a particular set of values for a given object. For a given state, a transition denotes a future state the entity can move to and the conditions associated with the change of the state

Figure 22

A state diagram at a glance (Source: Visual-paradigm, 2018)



The key concepts involved in a state diagram are the state, event, transition, and action. A state defines the condition of an object during its lifetime to satisfy certain conditions, perform some activity or wait for some event. An event refers to the specification of a significant occurrence, for example, a stimulus that can trigger a state transition. A transition is a relationship between two states indicating that an object in the first state will when a specified set of events and conditions are satisfied, perform certain actions and enter the second state. Action, on the other hand, is an executable and atomic computation that may include operations, creation or destruction of other objects, or the sending of signals to other objects.

State diagrams are useful when it comes to modeling a reactive (event-driven) system and animations (Temiz, 2014) or even in protocol state machine when using an interface for a class to perform functions such as open, close, query as in database access.

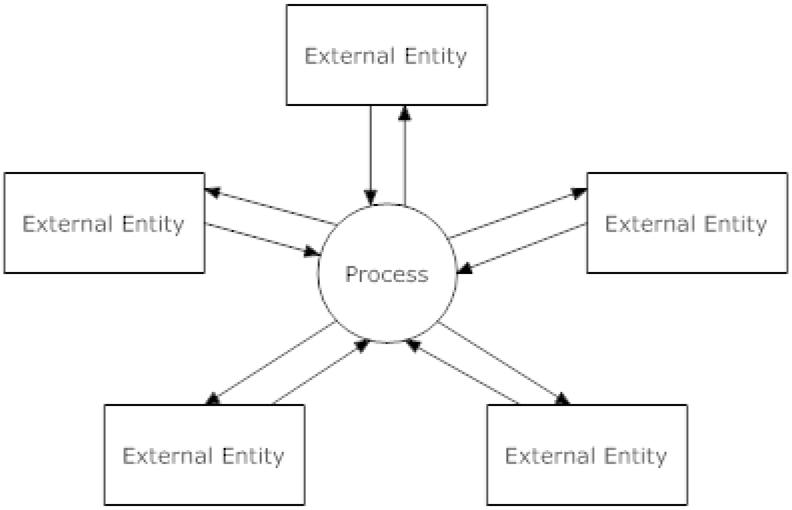
vi. Context Models

Marchese, (2013) state that context models are used to illustrate the operational context of the system by showing what lays outside the system boundaries. This is crucial because organizational concerns may affect the decision on where to position the system boundaries. And more so architectural models show the system and its relationship with other systems.

Context models are most appropriate during the early stages of requirements elicitation and analysis process (Sommerville, 2011) where the system boundaries are decided upon. This then requires the involvement of stakeholders to elaborate on the distinction between the system and non-system (environmental) requirements. Once decisions on the system boundaries are made, then an architectural model can be used to be part of the requirements analysis. Figure 23 below is an example of a generalized context diagram illustrating an overall system process in relation to five external entities.

Figure 23

A generalized context diagram (Source: SmartDraw, 2019)



In a nutshell, context models simply show the other systems in relation to the system in question, not how the system being developed is used in the environment. Other models such as activity diagrams can be used to define business processes in conjunction with the context diagram

vii. Interaction Models

According to Nieters, (2012), the interaction model is a design model that binds an application together in a way that supports the conceptual model of the target users. It defines how all of the objects and actions that are part of an application interrelate to mirror and support real-life application user. This way, the model enables designers, developers, and stakeholders to understand and explain how to navigate through the system to get information as well as perform the appropriate task. Using interaction models is a foundational requirement for and contributes to cohesive system architecture.

Modeling system-to-system interaction highlights the communication problems that arise and modeling component interaction help understand if a proposed system structure is likely to deliver the required system performance and dependability (Marchese, 2013). Use case diagrams and sequence diagrams may be used for interaction modeling.

viii. Structural Models

Structural models, according to Marchese, 2013, display the organization of a system in terms of the components that make up that system and their relationship. The models either show the structure of the system design in the static model state or show the organization of the system when executing in the dynamic model state. Structural models are able to provide a framework for detailed system modeling.

Most structural models have their own preferred set of system models. A class diagram is a good example of a model that is used in structural models.

Structural models have applied successfully in many large projects (Sommerville, 2011) because they use standard notations and ensure that standard design documentation is produced, therefore deliver at a significant low cost. However, they do not provide effective support for understanding or modeling nonfunctional system requirements, do not include guidelines to help users decide whether a method appropriate for a particular problem or not, produce too much documentation and the model produced is too detailed for users to understand.

ix. Behavioral Models

Behavioral models model the dynamic behavior of a system at execution time. They show what happens or what is supposed to happen when a system responds to a stimulus from its environment. The stimulus here can be thought of as data and an event (Marchese, 2013). This is because some data that has been processed by the system can arrive and some event can occur that triggers the system processing state. In the same way, events may have associated data. This makes the behavioral model be associated with two models; data-driven modeling (data models) and event-driven modeling (state machine models).

Data-driven models show the sequence of actions involved in processing input and generating an associated output. They are therefore crucial for many business systems whose systems are primarily driven by data processing. They are controlled by data input to the system, with relatively little external event processing. And more so they are useful during the analysis of requirements as they can be used to show end-to-end processing of in a system.

On the other hand, event-driven modeling shows how a system responds to both external internal events (Sommerville, 2011). State machine models are actually an integral part of real-time system designs due to their event-driven nature. The model assumes that at any time, is in one or a number of possible states that may cause transitions from one state to another.

2.9 Integrated Development Environments

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. Generally, IDEs consist of a source code editor, build automation tools and debuggers. In addition and depending on the IDE, some may or may not contain a compiler and an interpreter or both. There are a number of IDEs in existence, but the best ten in 2017 as listed in keycdn.com are Microsoft Visual Studio, NetBeans, PyChame, IntelliJ IDEA, Eclipse, Code:: Block, Aptana Studio 3, Komodo, RubyMine, and Xcode. But for the sake of this study, the first five IDEs were discussed as follows in reference to proximity LLC, (2017):

2.9.1 Microsoft Visual Studio

Microsoft Visual Studio is a premium IDE from Microsoft Corporation used to develop a wide range of programs ranging from web to mobile applications to video games. Programming languages supported by Microsoft Visual Studio IDE include ASP.NET, DHTML, JavaScript, JScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML and more (Eck, 2014).

The notable features of Microsoft Visual Studio IDE include:

i. A massive library of extensions that is always growing

ii. IntelliSense

iii. Customizable dashboard and dock able windows

iv. Straightforward workflow and file hierarchy

v. Insights for monitoring performance in real time

vi. Automation tools

vii. Easy refactoring and code snippet insertion

viii. Split screen support

ix. Error list that allows debugging while building

x. Approval checks when deploying apps via ClickOnce, Windows Installer or Publish Wizard

It's due to its flexibility that Visual Studio is greatly used by both students and professionals. However, Visual Studio is heavyweight, therefore it takes considerable resources to open and run so that making simple edits may be time-consuming on some devices (LLC, 2017).

2.9.2 NetBeans

A free and open source IDE, NetBeans is ideal for either editing existing projects or building projects from scratch. NetBeans as well boasts of a simple drag and drop interface that comes with a myriad of convenient project templates. Although NetBeans was primarily meant for Java programming language, it supports other programming languages such as C, C++, C++11, Fortran, HTML 5, Java, and PHP (Eck, 2014).

Apart from the intuitive drag-and-drop interface feature, other notable features that come with NetBeans include:

i. Dynamic and static libraries

ii. Multi-session GNU debugger integration with code assistance

iii. Allows for remote development

iv. Compatible with Windows, Linux, OS X, and Solaris platforms

v. Supports Qt Toolkit

vi. Supports Fortran and Assembler files

vii. Supports a number of compilers including CLang/LLVM, Cygwin, GNU, MinGW, and Oracle Solaris Studio

The only drawback with NetBeans is the consumption of lots of memory and therefore sluggish on some machines (LLC, 2017).

2.9.3 PyCharm

Armed with a combination of a *free community edition*, a 30-day free trial *professional edition* and an annual premium subscription, PyCharm is an IDE developed at Jet Brains. Its comprehensive code assistance and analysis make PyCharm the best IDE for Python programmers of all ability levels. PyCharm as well supports other programming languages such as AngularJS, Coffee Script, CSS, Python, HTML, JavaScript, Node.js, Python, TypeScript and template languages in addition to working on multiple platforms, such that anyone can use it (LLC, 2017).

And so the features that come with PyCharm include:

i. Compatible with Windows, Linux, and Mac OS

ii. Comes with Django IDE

iii. Easy to integrate with Git, Mercurial, and SVN

iv. Customizable interface with VIM emulation

v. JavaScript, Python and Django debuggers

vi. Supports Google App Engine

Despite all these features, PyCharm has some bugs such as the autocomplete features occasionally not working.

2.9.4 IntelliJ IDEA

IntelliJ IDEA is another IDE developed by Jet Brains that offers users a *free community edition*, a 30-day free trial for *ultimate edition* and an annual premium subscription. IntelliJ IDEA, which supports Java 8 and Java EE 7, comes with extensive tools to develop mobile applications and enterprise technologies for different platforms. It also supports AngularJS, CoffeeScript, CS, HTML, JavaScript, LESS, Node JS, PHP, Python, Ruby, Sass and TypeScript programming languages. When it comes to cost, IntelliJ is a real deal due to the massive list of features that comes with it (LLC, 2017).

The features include

i. Extensive database editor and UML designer
ii. Supports multiple build systems
iii. Test runner UI
iv. Code coverage
v. Git integration
vi. Supports Google App Engine, Grails, GWT, Hibernate, Java EE, OSGi, Play, Spring, Struts and more
vii. Deployment and debugging tools for most application servers
viii. Intelligent text editors for HTML, CSS, and Java
ix. Integrated version control
x. AIR Mobile supports Android and iOS devices

The only predicament with this IDE is that it comes with a learning curve, so it may not be the best for beginners. There are many shortcuts to remember, and some users complain about the clunky user interfaces

2.9.5 Eclipse

Eclipse is a free and flexible open source editor useful for beginners and professionals alike. Originally a Java environment, Eclipse now has a wide range of capabilities thanks to a large number of plug-ins and extensions. In addition to debugging tools and Git/CVS support, the standard edition of Eclipse comes with Java and Plugin Development Tooling. If that's not enough, there's plenty of other packages to choose from that include tools for charting, modeling, reporting, testing, and building GUIs. The Eclipse Marketplace Client gives users access to a treasure trove of plugins and information supplied by an expanding community of developers. Other programming languages supported by Eclipse include C, C++, Java, Perl, PHP, Python, Ruby and more (LLC, 2017)

The notable features are:

i. A plethora of package solutions allowing for multi-language support

ii. Java IDE enhancements such as hierarchical views of nested projects with customizable perspectives

iii. Task-focused interface including system-tray notifications

iv. Automated error reporting

v. Tooling options for JEE projects

vi. JUnit integration

While Eclipse is very versatile software, the many options may be intimidating to newcomers. Eclipse doesn't have all of the same features as IntelliJ IDEA, but it is an open source

2.10 Database Systems

Databases store information and its contents can be everything from product catalogs to repositories of customer information. For information to easily be accessed, used and understood, database management systems are required. Database management systems can help sort information as well as link databases to each other and provide reports about changes and trends in the information in databases. (Connolly & Begg, 2016)

There are a number of popular databases systems available. According to Arsenault, (2017), the most popular and used eight databases are Oracle 12c, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, MariaDB, DB2 and SAP HANA. Oracle 12c, MySQL and Microsoft SQL Server were, for the sake of the study, discussed as follows

2.10.1 Oracle 12c

It's no surprise that Oracle is consistently at the top of lists of popular databases. The first version of this database management tool was created in the late 70s, and there are a number of editions of this tool available to meet your organization's needs (Arsenault, 2017).

The newest version of Oracle, 12c, is designed for the cloud and can be hosted on a single server or multiple servers, and it enables the management of databases holding billions of records. Some of the features of the latest version of Oracle include a grid framework and the use of both physical and logical structures.

This means that physical data management has no effect on access to logical structures. Additionally, security in this release is excellent because each transaction is isolated from others.

Benefits

i. You'll find the latest innovations and features coming from their products since Oracle tends to set the bar for other database management tools.

ii. Oracle database management tools are also incredibly robust, and you can find one that can do just about anything you can possibly think of.

Limitations

i. The cost of Oracle can be prohibitive, especially for smaller organizations.
ii. The system can require significant resources once installed, so hardware upgrades may be required to even implement Oracle.

Oracle is therefore ideal for large organizations that handle enormous databases and need a variety of features.

2.10.2 MySQL

MySQL is one of the most popular databases for web-based applications. It is freeware, but it is frequently updated with features and security improvements. There are also a variety of paid editions designed for commercial use. With the freeware version, there’s a greater focus on speed and reliability instead of including a vast array of features, which can be good or bad depending on what you’re attempting to do (Arsenault, 2017).

This database engine allows one to select from a variety of storage engines that enable one to change the functionality of the tool and handle data from different table types. It also has an easy to use interface, and batch commands let one process enormous amounts of data. The system is also incredibly reliable and doesn’t tend to hog resources (Connolly & Begg, 2016).

Benefits

i. It’s available for free.

ii. It offers a lot of functionality even for a free database engine.

iii. There are a variety of user interfaces that can be implemented.

iv. It can be made to work with other databases, including DB2 and Oracle.

Limitations

i. You may spend a lot of time and effort to get MySQL to do things that other systems do automatically, like create incremental backups.

ii. There is no built-in support for XML or OLAP.

iii. Support is available for the free version, but one would need to pay for it.

MySQL is ideal for organizations that need a robust database management tool but are on a budget.

2.10.3 Microsoft SQL Server

As with other popular databases, one can select from a number of editions of Microsoft SQL server. This database management engine works on cloud-based servers as well as local servers, and it can be set up to work on both at the same time. Not long after the release of Microsoft SQL Server 2016, Microsoft made it available on Linux as well as Windows-based platforms (Arsenault, 2017).

Some of the standout features for the 2016 edition include temporal data support, which makes it possible to track changes made to data over time. The latest version of Microsoft SQL Server also allows for dynamic data masking, which ensures that only authorized individuals will see sensitive data (Connolly & Begg, 2016).

Benefits

i. It is very fast and stable.

ii. The engine offers the ability to adjust and track performance levels, which can reduce resource use.

iii. One is able to access visualizations on mobile devices.

iv. It works very well with other Microsoft products.

Limitations

i. Enterprise pricing may be beyond what many organizations can afford.

ii. Even with performance tuning, Microsoft SQL Server can gobble resources.

iii. Many individuals have issues using the SQL Server Integration Services to import files.

Microsoft SQL Server is ideal for large organizations that use a number of Microsoft products

2.11 Authentication Protocols

Authentication protocols or network authentication protocols are well defined, industry based standard ways of confirming the identity of a user when accessing network resources (Dooley, 2015; Latze, 2010). Commonly used protocols are Password Authentication Protocol (PAP), Challenge Hand-shake Authentication Protocol (CHAP) and Extensible Authentication Protocol (EAP). The protocols manage and communicate authentication credentials such as username, password, IP address and port numbers between a client (supplicant) and a server (authentication server) through an access point (authenticator). Supplicant, authenticator and authentication server form part of Institute of Electrical and Electronics Engineers 802.1X (IEEE802.1X) standard that authenticates users. The authentication credentials are managed and stored in the authentication Remote Authentication Dial-In User Service (RADIUS) server that uses a Remote Authentication Dial-In User Service (RADIUS) protocol to transmit the credentials. Point to

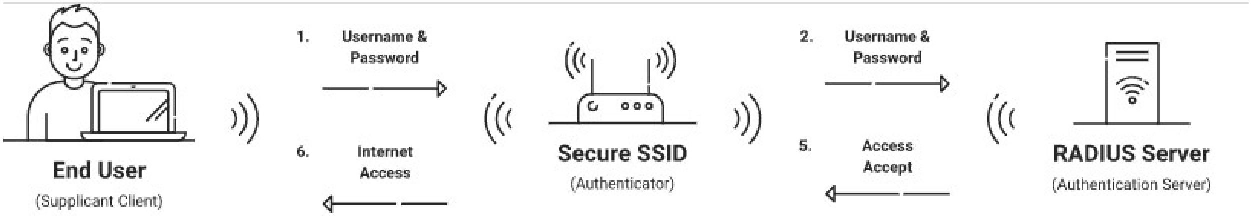
Point Protocol (PPP) establishes connection between the supplicant and the authentication server for credentials communication to take place.

2.11.1 IEEE802.1X Protocol

Huawei, (2019), defines 802.1X as a port-based network control and authentication protocol that prevents unauthorized clients from connecting to a LAN through publicly accessible ports unless they are not properly authenticated. Although it is primarily used in wired Ethernets, IEEE802.1X is equally applied in Wi-Fi networks via established virtual ports for authorized access (Intel, 2021; Rajesh, 2010). The protocol uses a RADIUS server and corresponding RADIUS protocol to validate user credentials thus allow or disallow network access. As a standard, IEEE802.1X constitutes three components; the supplicant, the authenticator and the authentication server. The Figure 24 below illustrates the relationship between the components and authentication messages flow.

Figure 24

IEEE802.1X message authentication flow. (Source: Intel, 2021)



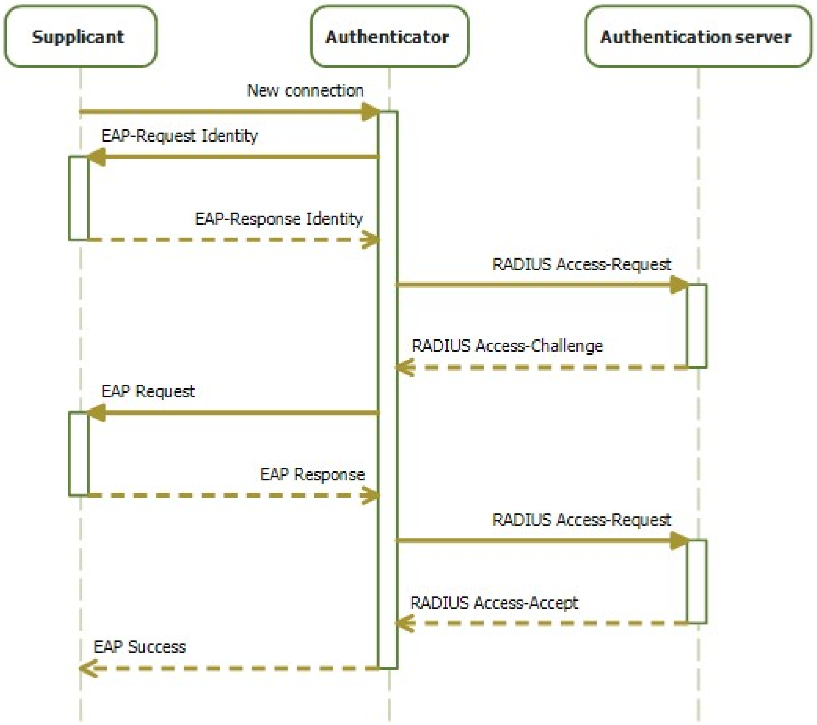
Whereas Extensible Authentication Protocol (EAP) or any other authentication protocol is used to exchange authentication messages between supplicant and authenticator, RADIUS communicates authentication messages between the authenticator and the authentication server.

Supplicant combines both the device and software that collects authentication credentials from an end user and forwards them for authentication process. Authenticator (access point), on the other hand, collects the credentials from the supplicant and relays them to the authentication server for verification. Authentication server validates the credentials sent by the supplicant based on the details stored in the RADIUS server to either allow or disallow network access (Fruhlinger & Snyder, 2021).

IEEE802.1X authentication is triggered by either a client sending an EAPoL (or any other authentication protocol) start packet, or a client sending a DHCP, ARP or any other packet or the access device sending an EAP request or identity packet. Either way, Huawei, (2019), enumerates IEEE802.1X authentication process as follows and illustrated in the Figure 25 below.

Figure 25

Sequence diagram of the 802.1X standard progression. (Source: Bauer, 2021)



The typical authentication procedure consists of:

1. The authenticator sends an "EAP-Request or Identity" packet to the supplicant as soon as it detects that the link is active (e.g., as the supplicant system associates with the access point).
2. The supplicant sends an "EAP-Response or Identity" packet to the authenticator, which passes it on to the authentication (RADIUS) server.
3. The authentication server sends back a challenge to the authenticator, such as with a token password system. The authenticator unpacks this from IP and repackages it into EAPOL and sends it to the supplicant. Different authentication methods will vary this message and the total number of messages. EAP supports client-only authentication and strong mutual authentication; only strong mutual authentication is considered appropriate for wireless networks.
4. The supplicant responds to the challenge via the authenticator, which passes the response on to the authentication server.
5. If the supplicant provides proper identity, the authentication server responds with a success message, which is passed to the supplicant. The authenticator now allows access to the LAN, though this can be restricted, based on attributes that come back from the authentication server. For example, the authenticator might switch the supplicant to a particular virtual LAN or invoke a set of firewall rules

2.11.2 Remote Authentication Dial-In User Service

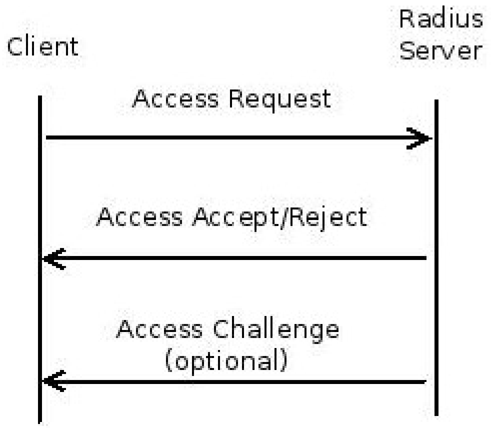
Remote Authentication Dial-In User Service (RADIUS) is a network protocol used to authenticate and authorize user access to a network (Brenna, 2021). RADIUS was originally designed for dial-in user access, and has been extended to support additional access methods, such as Ethernet and Active Directory Service (ADS). As a protocol that runs at the application layer of the OSI reference model, RADIUS contains a RADIUS client component that communicates with the RADIUS server using either UDP or TCP protocols (Emihem, 2022). RADIUS functions together with EAP in such a manner that whereas Extensible Authentication Protocol (EAP) is used to exchange authentication messages between supplicant and authenticator, RADIUS communicates authentication messages between the authenticator and the authentication server access (Intel, 2021).

Based on a client/server model, RADIUS constitutes the RADIUS client and RADIUS server (Huawei, (2019). Apart from transmitting user credentials to the specifies RADIUS server for processing, the RADIUS client additionally, processes request based on the responses from the server. Examples of such request include permitting or rejecting user access requests. RADIUS server on the other hand, maintains user authentication and network service access information.

The process begins by a user first sending a request to access signal (that contains access credentials such as username, password, IP address and port number) to the network to the RADIUS server. The server checks its database against the user request credentials and response with either Access Accept or Access Reject signals to the client. The reject response is usually accompanied with a message that explain the reject actions. If the user is accepted, the server sends attributes such as IP address, Vendor-Specific information, session timeout and access lists to define the kind of access granted (Nadia, 2013). The RADIUS server may eventually request for further information before granting access in the form of an Access Challenge signal. All these are illustrated in the Figure 26 below.

Figure 26

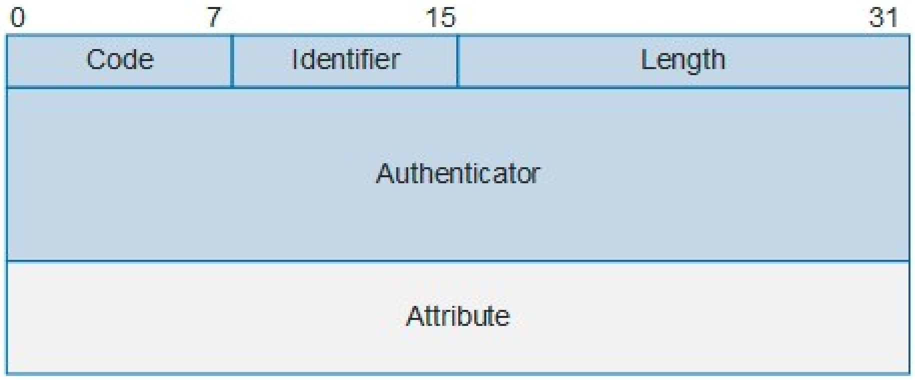
RADIUS message exchange process. (Source: Nadia, 2013)



Each RADIUS packet structure contains information as shown in the Figure 27.

Figure 27

RADIUS packet structure. (Source: Huawei, 2019)



The code field (1 byte long) indicates the type of the RADIUS packet that are either Access-Request, Access-Accept, Access-Reject, Accounting-Request and Accounting Response. The identifier field (1 byte long) is used to match response packets with request and to detect duplicate request packets. The length field (2 bytes long) indicates the length of the entire packet that include the code, the identifier, the authenticate, the attribute as well as the length fields.

Fields beyond the required length are padded and ignored by the receiver. Less length fields are dropped. The authenticator field (16 bytes long) is used to authenticate responses from the RADIUS server and encrypt user passwords. The attribute field (variable in length) contains multiple attributes that includes authentication, authorization and accounting information, each with Type, Length and Value sub fields. Type subfield indicates the type of attribute, Length indicates the length of the attribute in bytes and Value indicates the value of the attribute.

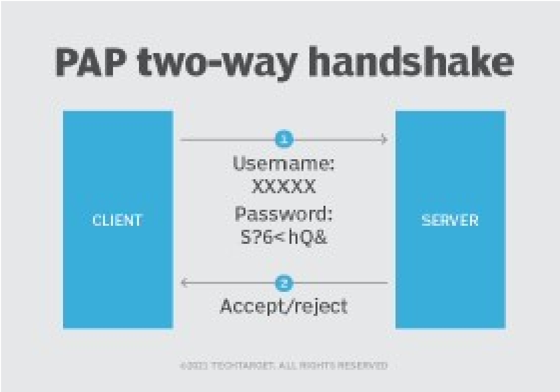
2.11.3 Password Authentication Protocol

Matkar et al., (2016), define Password Authentication Protocol (PAP) as a password-based authentication protocol used by a Point to Point (PPP) to validate network users. PPP is a data link network communication protocol used to establish a direct connection between two nodes, that's, the client and the server. Here, PAP sends user credentials (username and password) to the authentication server unencrypted as plain text by using a two-way handshake procedure (Prakash & Kumar, 2018).

According to Matkar et al., (2016), PAP works by client sending username and password to the server for verification, then the server response with accept/reject ack to the client. The Figure 28 below illustrates the process.

Figure 28

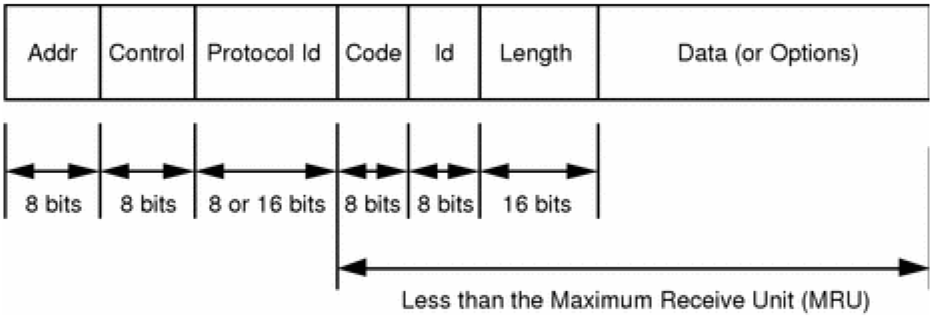
PAP authentication process. (Source: Froehlich & Tuomenoksa, 2020)



PAP frame structure as described in Oracle, (2010), contains Address, Control, Protocol Id, Code, Id and Data (or Options) fields to accomplish its purpose as illustrated in the Figure 29 below.

Figure 29

PAP frame format. (Source: Oracle, 2010)



Whereas the address and control fields, a one octet in length, and are part of the High-level Data Link Control (HDLC) framing, protocol id field identifies the type of information contained in the information field of the frame. Code field identifies the type of PAP frame and Id field carries an identifier used to match associated requests and replies. The length field indicates the total length of the PAP frame including the code, id, length, and data fields. The data field equally contains information associated with the authentication negotiation, in a format determined by the code field.

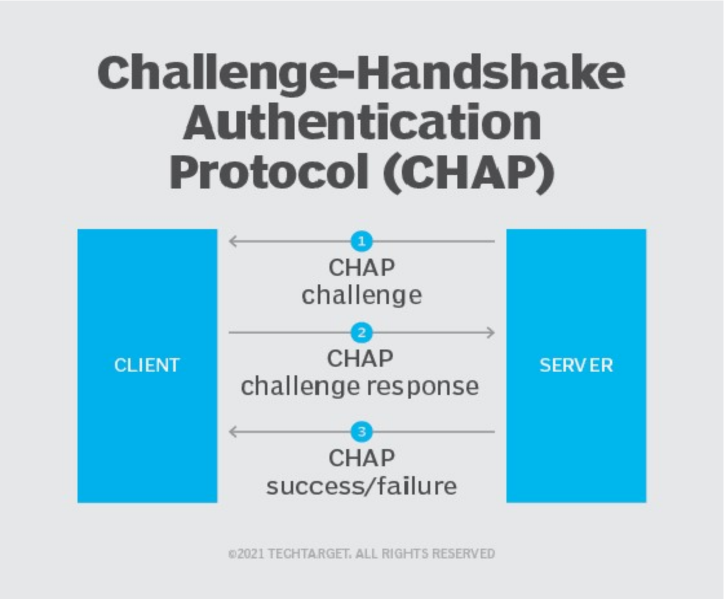
PAP is simple, uses relatively less computational resources, is compatible with different server types running on different OSs as well as can perform remote authentication. However, PAP is not secure; it is vulnerable to eavesdropping and man-in-the-middle based attacks, older and can overload the server due to sending sensitive credential repeatedly. As such, PAP is used on Serial Line Internet Protocol (SLIP) systems, can be used as a last resort if the remote server does not support stronger protocols such as CHAP or EAP and can be used in circumstances where plain text password must be available to simulate a login at the remote host (Orbitco, 2015).

2.11.4 Challenge-Handshake Authentication Protocol

Just like PAP, Challenge-Handshake Authentication Protocol (CHAP) authenticates PPP sessions and can be used with many virtual private networks (VPNs). However, CHAP protocol creates a unique challenge phrase for each authentication by generating a random string, then combines with a device hostnames using one-way hash functions. Both the server and the client run the password through the hash function along with a one-time password (OTP). This way, CHAP authenticates in such a way that secret information is not sent over the channel (Froehlich & Tuomenoksa, 2020), as shown in the Figure 30 below.

Figure 30

PAP authentication process. (Source: Loshin, 2021)


Challenge-Handshake Authentication Protocol (CHAP)
CLIENT
SERVER
1
CHAP challenge
2
CHAP challenge response
3
CHAP success/failure
©2021 TECHTARGET. ALL RIGHTS RESERVED

As soon as a link has been established, the authenticator sends “challenge” that includes a randomly generated challenge string to the client. The client then uses the password to create an encrypted one-way hash value based on the challenge string. Finally, the server decrypts the hash values and verifies it. If the strings match, the server responds with authentication success packet, otherwise, the server sends authentication failure (Froehlich & Tuomenoksa, 2020).

As compared to PAP, CHAP is more secure (it doesn’t send any other credentials other than the username in plaintext). It is also safe against replay attacks because the OTP changes periodically. However, CHAP is still vulnerable to eavesdropping on the packets because the payload is send in clear text. Intruders through eavesdropping, can take the MD5 hash, and brute-force the combination to determine the password (WorkOS, 2020).

Variants of CHAP include Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP) and MS-CHAPv2. MS-CHAP is the Microsoft implementation of CHAP. As compared to CHAP, MS-CHAP provides an authenticator-controlled password mechanism and defines failure codes returned in the Failure packet message field. MS-CHAPv2 supports two-way authentication to verify the identity of both sides of a point to point connection and provides separate cryptographic keys for transmitted and received data based on user password.

2.11.5 Extensible Authentication Protocol

Extensible Authentication Protocol (EAP) just like PAP and CHAP, is an authentication protocol that is used to transmit authentication credentials between the supplicant and the authentication server. However, just as its name suggests, EAP expands or extends the authentication methods used by PPP (Webster, 2021). EAP as well operates at the data link (layer 2) of the OSI reference model, thus ensures the elimination of duplicate retransmission of frames (Christian, 2021).

EAP is more referred to as framework rather than a protocol because it constitutes a number different authentication methods that can be used (WorkOS, 2020). So that during authentication process, each request or response between the server and client, a type for authentication is specified. This way, EAP methods protect a specific portal so that only users with an authentication key or password can get access. The methods therefore limit the number of users and help prevent network congestion, making the network faster and more secure. It supports various authentication methods that include token cards, smart cards, certificates, OTP, and public key encryption. That is why enterprises and network proprietors can adopt one of the methods for specific needs and guidelines (Webster, 2021). Some of the types include EAP-MD-5, EAP-TLS, EAP-PEAP, EAP-TTLS, and EAP-Fast.

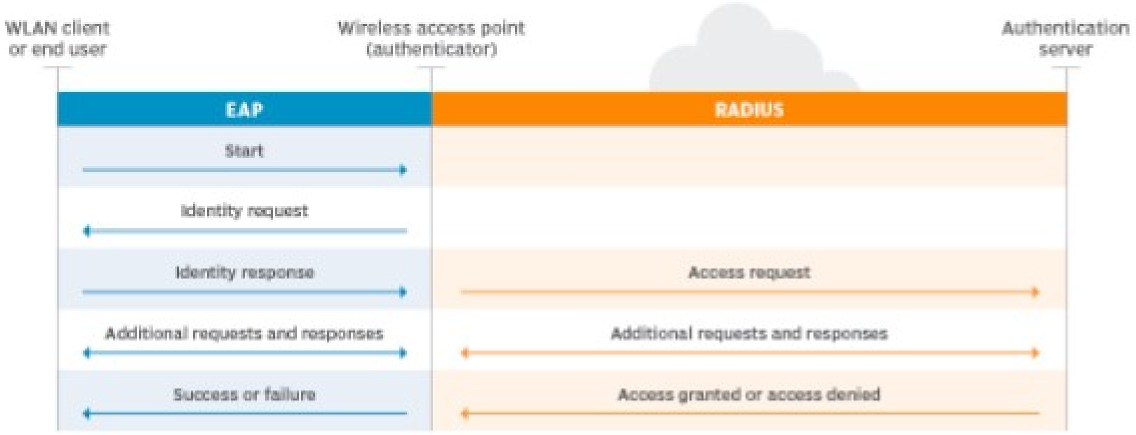
Almost like CHAP, for instance, EAP-Message Digest 5 (EAP-MD5) hash algorithm is used with an OTP to hide the password during authentication process. EAP-TLS is certificate-based authentication. Both the client and the server need to have a certificate. This renders EAP more secure if not the most secure methods, but deployment is difficult to manage due to its complexity. EAP- Flexible Authentication via Secure Tunneling(EAP-FAST) on the other hand does not use certificates, therefore authenticates with a Protected Access Credentials (PAC) managed by a separate authentication server. As such, EAP-FAST is complex but faster.

EAP generally functions as follows (Webster, 2021) as shown in the Figure 31 below

1. A user requests connection to a wireless network through an AP -- a station that transmits and receives data, sometimes known as a transceiver.
2. The AP requests identification data from the user and transmits that data to an authentication server.
3. The authentication server asks the AP for proof of the validity of the identification information.
4. The AP obtains verification from the user and sends it back to the authentication server.
5. The user is connected to the network as requested

Figure 31

802.1X authentication process. (Source: Webster, 2021)



2.12 Authentications Methods

Major authentication methods or technologies began way back during the Second World War by the use of identification of a friend or a foe (IFF) (Lehtonen et al., 2008). From then on, advances in authentication techniques took effect. The methods are categorized based on what one is known or knowledge based (password, PIN), what one has or possesses (token, certificate), who one is or inheritance (biometrics), where one is or location based/address based (MAC address, IP address) and when one is authenticating or time factor as well (Shacklett, 2021). This section discusses password, smart card, biometric, certificate MAC address, IP address and multi-factor (MFA) based authentication methods.

2.12.1 Password Authentication

Organizations, enterprises or Wi-Fi home owners require network resource control due to the number of users accessing their network. One such control method is the use of a password authentication that require the use of user name and password credentials. Passwords are the most commonly used authentication method because they are simple, cost-effective, ease of operation and practical (Sectona, 2021). One only requires to key in a combination of letters of the alphabet (both upper and lower case), digits, symbols, special characters and even space characters to form a strong password (Team, 2021).

The basic steps that password authentication process takes place as described by Mishra & Ali (2013) are

1. Prompt for user id and password
2. User enters user id and password
3. User id and password are validated
4. Authentication result back to the server
5. Inform user accordingly

Although passwords are supposed to be known only by the user, some users share their passwords with others. Some go to an extend of writing the password down making it easy for intruders to get them. The need of a password for every application weakens password authentication method as users find it difficult to remember them all, thus, prefer simple passwords that eventually are easy to guess. This way, password authentication is prone to snooping, phishing and brute-force-attacks (Johnson, 2021). Diaz, (2015) suggests the use of forward secrecy mechanism that ensures that keeps the password secure during transmission. The password as well needs to be protected against online attacks and database compromises.

A more elaborate way of enhancing security of password authentication is the implementation of a two-factor authentication (2FA) (N-able, 2020). Here, the user enters the user name, the password and a one-time code that has been sent to a physical device such as a cellphone or an email. Two-factor authentication is a great option for network proprietors and their users as it adds a layer of security that makes it difficult for hackers to crack.

All in all, simplicity of password authentication makes it possible to be used on various systems and protocols. It is the basis authentication mechanism in pluggable authentication module (PAM) for Unix as well as in may web based platforms that are usually encapsulated with TLS. A well-known implementation for password authentication is PAP that operates over PPP protocol (Diaz, 2015).

2.12.2 Token Based Authentication

Token based authentication method requires users to obtain a randomly and periodically computer-generated code (or token) before they're allowed access to a network. Each authentication generates a random and unique token value. There are two types of authentication tokens; challenge/response and time-based token (Mayan, 2020). Challenge/response token, a combination of techniques, is preprogrammed inside the authentication token, this value is kept secret and should be unique. In this token, the value becomes an encryption key. A time-based token, in which server needs to send any random challenge to the user. The goal behind this is to use the time as a variable input to the authentication process, in place of the random challenge.

Token authentication is typically used in conjunction with password authentication for an added layer of security in what is referred to as a two-factor authentication (2FA) (N-able, 2020). This way, token based authentication focuses on inheritance or something that one is rather than something that one knows or something one owns as in password and biometrics authentication methods respectively. It is like an added obstacle to make it difficult for intruders gain access if they had cracked password credentials.

The method is implemented on a device such as a Universal Serial Bus (USB) key fob or a smart card such that token information is protected on the device itself (Mishra & Ali, 2013). The device might as well store credentials such as passwords, digital signatures, certificates and private keys. An authentication token usually has a processor, liquid crystal display (LCD) for displaying outputs (random code values), a battery, a small keypad for entering information and a real time clock.

Token based authentication process take place through the following steps (Mishra & Ali, 2013);

1. Creation of a token
2. Use of the token
3. Token validation
4. Server returns an appropriate message back to the user

Token based systems are generally more expensive due to overheads incurred in purchasing devices, or the device can fall in the wrong hands, and can possibly be intercepted as the case in any other transmissions, but are more secure and cost-effective (SolarWinds Passportal, 2021; N-able, 2020).

2.12.3 Digital Certificates

Certificate-based or digital certificate authentication, as described by Mayan, (2020), identifies users, machines or devices using certificates. A digital certificate also known as a public key certificate, for that manner, is an electronic file that contains identification details about the holder, including the owners public key (owner's private key is stored virtually), along with authority's digital signature for verification with the authority (Gregersen, 2021). They are used to proof the ownership of a public key and issued only by a certification authority or a third party.

Digital certificate authentication can be costly and time consuming to deploy. They involve a third party and at times requires reenrollment in case a user is unable to access the certificate. On the other hand, digital certificates are stronger than as compared to password based authentication, and are applicable in scenarios that involve an external entity like a contractor, supplier or a consultant to a n enterprise (Johnson, 2021). This is so because the use the "have" something rather than "know" something, as in password based authentication method. Just like any other password-less authentication (biometrics), digital certificate authentication improves user experience (they are not many per user), no worry of password theft, provides solutions to in protecting brute-force attacks, strengthens an organizational cyber security posture and reduce running cost (Mehta, 2021)

Mishra & Ali, (2013), enumerates the steps taken by digital certificate based authentication as;

1. Creation, storage and distribution of digital certificates
2. Login in request (user to the server)
3. Server creates a random challenge
4. User signs the random challenge
5. Server returns an appropriate message back to the user

2.12.4 Biometric Authentication

Biometrics or biometric authentication is defined as the science of authenticating people using physiological features. Biometric identification system (BIS) then is identification based on biometrics. Although any physiological feature could be used for authentication, the most generalized techniques include the automated recognition of fingerprints, faces, iris, retina, hand geometry, voice, and signature (Garzia-Luis et al, 2003). A biometric system basically operates by acquiring biometric details from an individual, extracts a feature set from the acquired details and compares the feature set against pre-recorded details of the individual in a template set in a database (Luis-Garzia et al, 2013). An individual simply inserts their smart card containing their biometric information in a reader at an identification point, interacts with the sensor (a camera or a scanner), the sensor acquires and processes the information and compares the information with the information on the card for verification. Through this, BIS can actually perform identification as well as verification of individuals.

Luis-Garzia et al, (2013), goes on to identify the components of BIS as the sensor, feature extractor, matcher and database modules. Whereas the sensor module captures the biometric details of an individual of interest (fingerprint sensor for instance), feature extractor module processes the extracted details to obtain a set of salient or discriminatory details. Matcher module on the hand is module used to compare extracted details with those stored in the database for recognition. This is the point where the user's claimed identity is confirmed (verified) and therefore the user's identity is established (identification) based on the matching details. System database module contains a list of biometric details of valid or enrolled individuals

Password authentication is knowledge-based and token authentication is inheritance based authentication. As such they are prone physical damages, attacks and can be forgotten but biometrics authentication is possession based authentication. That is why BIS has found its usefulness in individual identification and verification in applications such as computer network login, electronic data security, e-commerce, internet access, ATMs, credit cards, physical access control, cellular phone, PDAs, medical records management systems, distance learning, national

ID card, driver's license, border control, passport control, corpse identification, criminal investigation, terrorist identification and parenthood determination to name but a few applications.

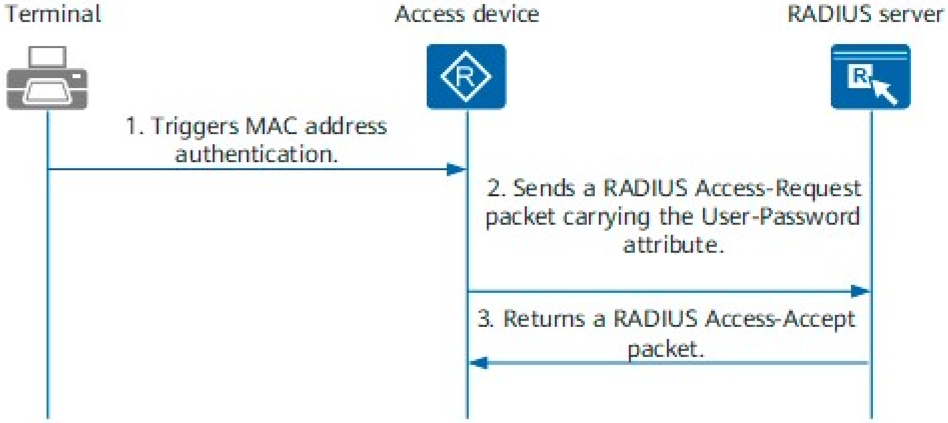
2.12.5 MAC Address Based Authentication

MAC address authentication is a port-based authentication method that allows or denies network access based on the MAC address credentials for machines such as IP phones, printers, and network attached storage devices. As a layer 2 OSI reference model issue, MAC address authentication solution uses RADIUS over IEEE802.1x framework rather than EAP (Cisco, 2018; Fredrisson, 2017). When a device connects to an access point (AP), the AP forwards the MAC address as the log in credential to the RADIUS server. With MAC-based authentication, the MAC address serves as both the username and the password. The RADIUS server consults the authentication server and sends back a RADIUS return attribute based on authentication results

The process of a MAC address authenticating a device according to Huawei, (2021), and illustrated in the Figure 32 below, is as follows:

Figure 32

MAC address authentication process. (Source: Huawei, 2021)


Terminal
Access device
RADIUS server
1. Triggers MAC address authentication.
2. Sends a RADIUS Access-Request packet carrying the User-Password attribute.
3. Returns a RADIUS Access-Accept packet.

1. The access device receives an ARP, DHCP or DHCPv6 packet from a terminal, which triggers MAC address authentication.

2. The access device generates a random value, arranges the terminal MAC address, shared key, and random value in sequence, and performs hash processing on them using the MD5 algorithm. It then encapsulates the user name, hash result, and random value into a RADIUS authentication request packet, and sends the packet to the RADIUS server for MAC address authentication.

3. Based on the received random value, the RADIUS server performs hash processing on the combination of the user MAC address, shared key, and random value in the local database using the MD5 algorithm. If the hash result is the same as that carried in the received packet, the RADIUS server sends an authentication accept packet to the access device, indicating that MAC address authentication of the user is successful. The user is then allowed to access the network.

MAC address filtering, also referred to as address control or address reservation, or still wireless MAC authentication allows or blocks traffic from a known machine or device depending on an organizational security policy to secure their networks. Although MAC filtering can go a long way in aiding secure a network to some extent, it is not as secure as such. As discussed in the weaknesses of a MAC address section of this study, MAC addresses are not encrypted (Gill & Dahiya, 2017), can be spoofed (Lee, 2010), it is not recommended for large wireless networks (Singh & Sharma, 2015; Watanabe et al, 2013) and can be altered (Apple, 2011). Furthermore, Kurose & Ross, (2013), raises the issue that a device can be attached to multiple network each with a corresponding MAC address interface. A device, for instance can be attached to an Ethernet port, a Wi-Fi, or a Bluetooth port, all of which cannot uniquely identify the device. A MAC address in a nutshell, is not unique, is not secure and unreliable

2.12.6 IP Address Authentication

As a subset of location-based authentication, IP address authentication is a traditional method of authenticating users that require network and resource access. Once a user logs onto a network, IP address authentication checks on their IP address and validates them against a list of allowed IPs or IP ranges. When a range of addresses is specified, the access point performs a logical and with the IP address entered in the IP address filter and the configured subnet. If an exact IP address is specified, the authentication method specifies a subnet mask so that only request from a client IP address is allowed or blocked, depending on what is configured in the filter (Servicenow, 2022).

This eliminates or reduces the use of user IDs and passwords, initial configuration and maintenance is simple, works well with static IPs, however, requires a separate remote authentication tool, and slow for dynamic IPs, (EBSCO, 2022)

But with dynamic nature of networks, that is, users and devices are mobile, plus users can use multiple devices from different locations, thus IPs might not correspond to their institutions. Although virtual private networks and proxy services have been fronted to remedy the drawback, they are complicated to manage have their own issues (Hoy, 2019). IP address range makes it easy to be spoofed, dynamic IP address allocation results to multiple users using the same IP, and the IP address keeps changing for the same device from location to location (Chad, 2017).

2.12.7 Multi-Factor Authentication

The issue of multi-factor authentication (MFA) came in due to limitations posed by password, token, certificates and biometrics based authentication methods. As such token authentication was thought to address the limitation of password authentication by using a token to establish an identity of a user. But token authentication still has a challenge since it uses a hardware that can be damaged or get lost, resulting to loss of information. In order to overcome password and token based authentication challenges, biometrics authentication method was introduced. The advantage of biometrics authentication is that it binds the user to their authentication credentials, yet an intruder can still copy biometrics attributes (Team, 2021).

MFA an authentication method that requires a combination of two or more independent authentication methods. The method adds multiple layers of security resulting to, apart from improved security, increases the confidence of users of such services (Maayan, 2020). The basis of MFA is single-factor authentication (SFA) that required authentication with only one type of authentication, mostly passwords. Two-factor authentication (2FA) or 2-step verification, require users to present two factors for authentication. One would, for instance, key in a password and prompted to confirm with a code (a one-time password (OTP)) sent via SMS authentication or via an email. Using a separate channel for confirmation, makes it difficult for intruders to intercept credentials.

As Karlinsky, (2021), puts it, security is no longer about either flatly granting or denying access based on a factor or two; but about granting a degree of access from a spectrum of possibilities, based on multiple data points and factors. This is exactly the case in MFA where a combination of login attempts, tokens, biometrics and SMS can be employed in accessing a network resource.

Although MFA increases defense against most account attacks as well as boosting confidence of users, the method is not without some pitfalls. Generated authentication codes from third parties might not be of any use if depended devices get lost or damaged (Maayan, 2020). IT manages equally find it cumbersome integrating multiple applications or systems

2.13 Research Gap

Following the discussion on currently employed identifiers and good characteristics of a device identifier, and authentication methods, three research gaps were observed; the need of an alternative identifier to MAC address, the need of additional identifier to the existing identifiers, the need for a direct, not an indirect identifier and consequently the need for an authentication method that uses a serial number to authenticate computers in a network.

The need of alternative identifier to MAC address arises from the fact that a MAC address has a number of weaknesses. A device could contain more than one network interface resulting to a number of MAC addresses for the same device rendering a MAC address not universally unique (Kurose & Ross, 2013). MAC address can additionally be easily spoofed (Wallace, 2018) and altered, and therefore aid in spoofing attacks. The fact then that a MAC address is not unique, not secure and not robust calls for the search of an alternative identifier, hence a corresponding authentication method, which is the focus of this study.

The other need arises from Figure 1 which demonstrates that a device can either be identified indirectly by an application-specific address at the application layer, a port address at the transport layer, an IP address at the network layer or a MAC address at the data link layer as far as OSI reference model layer is concerned (Jeevanesh, 2017). As far as the illustration is concerned, no identifier identifies a device at the physical layer.

In addition to the two key gaps discussed, the application specific, a port number, a MAC address, and an IP address are all addresses (Coulouris et al., 2012). Yet what is needed to actually identify, hence authenticate a device or any other entity is an attribute that directly describes the entity rather than an address which is location-based, therefore, keep changing. In other words, there is need of an identifier and consequently, a corresponding method that actually identifies the device rather than a pointer to the device. This is the reason then the study sought to design a model that demonstrates the possibility of using a computer's serial number as an authenticator.

Table 2

Research Gap

Author	Gap	Summary
Kurose & Ross, (2013)	Alternate identifier	MAC addresses can be attached to multiple interfaces MAC address can be spoofed and altered, therefore it's not secure, unreliable and not unique
Jeevanesh, (2017)	Additional layer identification	All other upper layers of the OSI (application layer uses application specific, transport layer uses port number, network layer uses IP address and data link layer uses MAC address) except the physical layer have identifiers
Coulouris et al., (2012)	Direct identifier	Existing identifiers are actually addresses; application specific, port number, IP and MAC addresses. They identify the computers in relation to their locations rather than the computers themselves

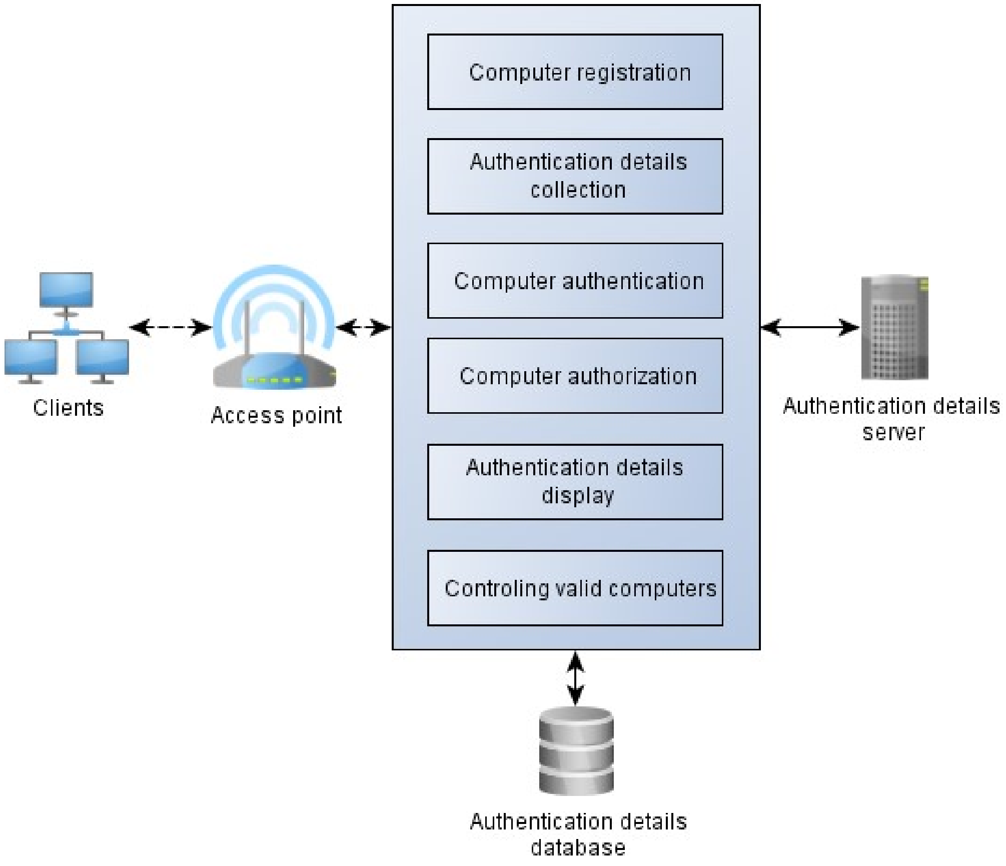
2.14 Conceptual Framework

The prototype conceptual framework in Figure 22 illustrates how the prototype modules used to demonstrate the use of a computer's serial number for authentication interact with other players of the system. Although the prototype was based on a client-server architecture, the application ran on the server side. The components of the prototype were the clients, access point, authentication server and authentication details database. The application itself is composed of the registration of new computer interface, the serial number, computer name and IP address details authentication collection module, the computer authentication module, computer authorization module, authentication details display interface and controlling of validly logged on computers interface.

Computers are first registered using the computer registration interface so that they can be allowed to log on to the network. The authentication details collection module is a code on the server that loads the serial number, name and IP address of client when client logs on to the network. The loaded number is validated with the valid serial numbers in the computer registration application by the computer authentication, a code running at the server as well. If the serial number exists in the registration application, the client is allowed access to the network and network resource (computer authorization); if not, access will be denied. Apart from displaying allowed user details, authentication details interface has a provision of disabling logged on computers if a need arises. The Figure 33 below summarizes the prototype conceptual framework.

Figure 33

Prototype based conceptual framework


Computer registration
Authentication details collection
Computer authentication
Computer authorization
Authentication details display
Controling valid computers
Clients
Access point
Authentication details server
Authentication details database

2.15 Chapter Summary

This chapter presented an overview of network access control services needed to manage how users access a network and its resources, followed by discussions on suitability of existing network device identifiers (Application-specific, Port number, IP address and MAC address) with a bias on MAC address and a computer's serial number as network device identifiers. From the discussions and test run presentations, it was found out that a MAC address is not secure, not unique and unreliable as opposed to a serial number that was found to be secure, unique and reliable. A discussion on IEEE802.1X, RADIUS, PAP, CHAP and EAP authentication protocols, then password, smart card, token, certificates, biometrics, IP address and MAC address based authentication methods were presented. Whereas password, smart card, biometric, certificate methods are used by users or people in a system (referred to as user authentication), MAC address and IP address authentication on the other hand, is machine authentication. The presentations indicated the need of a secure, unique and reliable, a direct, and a physical layer of the OSI reference model identifier with a corresponding authentication method. The chapter therefore wound up by proposing an authentication method as presented in the conceptual framework section. The section next presents the methodology used in carrying out the research.

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This chapter of the study presents discussion on how design science research (DSR), as an adopted research methodology, was employed in the study. It also presents pilot study, reliability, ethical consideration and research authorization that concerns the study.

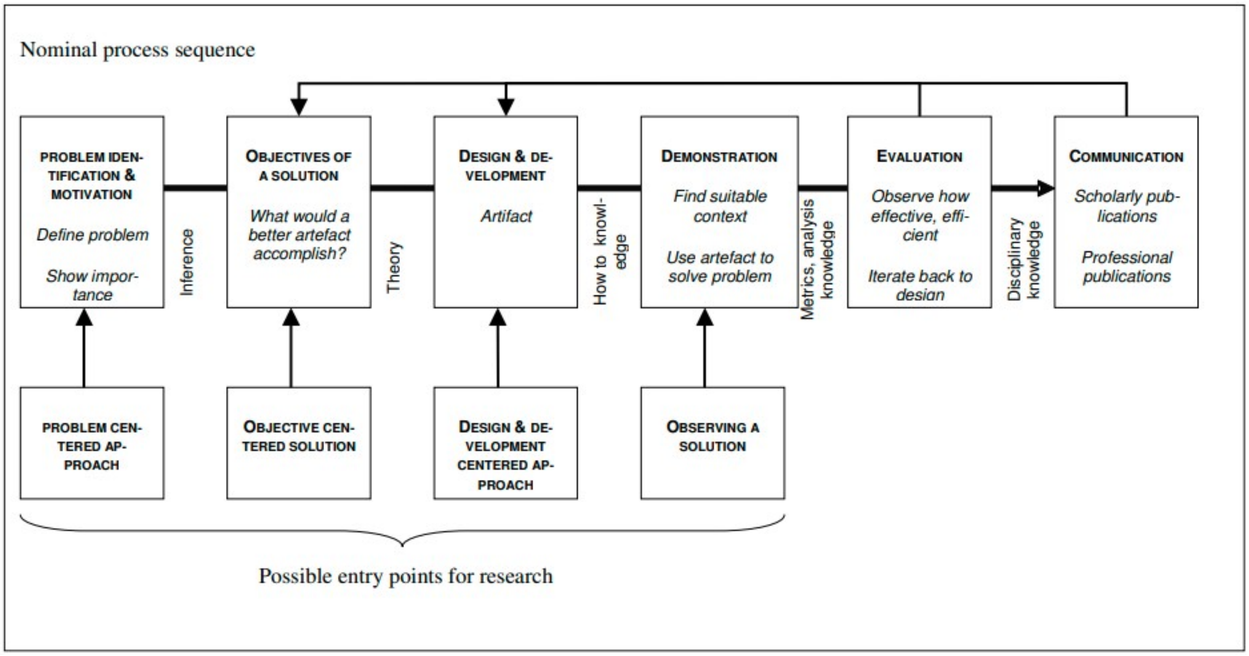
3.2 Design Science Research

As O'Leary, (2011), puts it; research design is the research master plan that shows how all major parts of a research study work together in an attempt to address the research question. In this study, the ultimate goal was to design a model that demonstrates how a computer's serial can be used to authenticate a computer in a wireless LAN. There was a need of designing a model, developing an algorithm, implementing and evaluating a program that demonstrates that a serial number can be used in authenticating a computer in the wireless LAN. The study therefore was carried out on a design science research method.

Design science research (DSR) as defined by Peffers et al., (2007), is an outcome based information technology research methodology. DSR is designed in such a way that it would be consistent with prior literature, it would provide a nominal process model for doing DSR research, and it would provide a mental model for presenting and appreciating DSR research in information systems. Lapão et al., (2017); Teixeira et al., (2017) and Peffers et al., (2007), presents the six steps that are required in DSR as (1) problem identification and motivation, (2) definition of the objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication as illustrated in the Figure 34 below.

Figure 34

Design science research process. (Source: Peffers et al., 2007)


Nominal process sequence
PROBLEM IDENTIFICATION & MOTIVATION
Define problem
Show importance
Inference
OBJECTIVES OF A SOLUTION
What would a better artefact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, analysis knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
OBJECTIVE CENTERED SOLUTION
DESIGN & DEVELOPMENT CENTERED APPROACH
OBSERVING A SOLUTION
Possible entry points for research

Peffers et al., (2007), clarifies that research don't have to necessarily follow the order given, rather it should include all of the steps stated.

3.2.1 Problem Identification and Motivation

Many a network administrators and network proprietors face numerous challenges in authenticating and controlling machines that use their network by their employees, clients, visitors, vendors or contractors. This is so especially at this time when enterprises have adopted the bring your device (BYOD) (Poremba, 2017), due to the many benefits that it comes with it. The challenge is more compounded by the rise in the use of internet of things (IoT) devices such as smart devices, smart watches and smart phones, as well (Pierce, 2021).

Authentication techniques such as password, smart card, biometric, certificate methods are used by users or people in a system to access a network and network resources. MAC address and IP address authentication on the other hand, are machine authentication method (Fredriksson, 2017). A MAC address authentication, for example, is not unique, not secure and not robust as part of requirements of the characteristics of a good identifier. A better alternative to MAC address lies in the use of a computer's serial number as a computer identifier and consequently, a corresponding authentication method.

3.2.2 Objectives of the Solution

The main objective of the study was to design a model that demonstrates how a serial number can be used for authenticating a computer in a wireless LAN. The following were the specific objectives;

i. To develop an algorithm that can obtain and use a remote computer's serial number in a wireless LAN

ii. To design a model that uses the computer's serial number to authenticate the computer in a wireless LAN

iii. To demonstrate how a computer's serial number is used to authenticate the computer in a wireless LAN

iv. To evaluate the model that uses the computer's serial number to authenticate the computer in a wireless LAN

In order to achieve the aim of the study, the following research questions had to be answered

i. How can an algorithm that can obtain and use a remote computer's serial number in a wireless LAN be developed?

ii. How can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be designed?

iii. How can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be demonstrated?

iv. How can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be evaluated?

3.2.3 Design and Development

The study adopted the use of state chart diagrams and flowcharts (Bruegge & Dutoit, 2010) as a theoretical basis for the model development. State chart diagrams relates the various fundamental states of the SNAM model starting from the preparation state, data collection, data processing, data storage and data display states of the model. The output of the later state is as a result of former states recursively. Dynamic programming design on flowcharts and pseudo codes were therefore used in overall and component parts of the model development.

For the purpose of demonstrating the intended functionality of the problem, namely using a serial number to authenticate a device in a network, an evolutionary prototyping was developed to proof the concept (Carter et al. 2001). Evolutionary prototyping method is used when an initial version of a system is developed using the best known and highly prioritized requirements. The initial version of the system is then evolved to a final product through a number of stages by consulting with different stakeholders.

3.2.4 Demonstration

In order to demonstrate the proof of concept that a computer's serial number can be used to authenticate a computer in a wireless LAN, the SNAM model was implemented using MySQL database and Java's IDE tools over evolutionary prototyping using a static group comparison pre-experimental design set up.

Apart from the need of using PoC to proof the overall concept that a computer's serial number can be used to authenticate a computer in a wireless LAN, the study as well had to use PoC to proof that other modules of prototype that culminates to the overall concept, can as well be executed. PoC according to Leurs & Duggan, (2018) and MacPherson, (2018) is an exercise to test design or assumption ideas. Software developers tend to use PoCs instinctively when experimenting with the technology. Leurs & Duggan, (2018), goes on to describe PoC as a performance-based method whose results need to be measurable so that they can be fed into a decision making process. The other concepts of the study, that correspond to the system modules, and require PoC were that;

1. A computer's authentication details (that is name, IP address and Serial number) can be collected
2. A registered computer can access a network
3. An unregistered computer cannot access a network
4. An already logged on or an allowed computer can be denied access to the network if a need arises

The PoC was actualized with the use of MySQL databases and Java's IDE tools. MySQL manages the storage and processing of the authentications details (that is name, IP address and Serial number). MySQL is a freely available open relational database management system (RDBMS) that uses the structured query language (SQL) platform. The choice of MySQL is based on the fact that apart from its popularity in storing and manipulating data, MySQL is simple, easy, fast, reliable and flexible (Nixon, 2018; Larusson, 2015). Therefore, very applicable in prototyping PoC projects. On the hand java's IDE NetBeans was used in interface construction and system codes. Java, as compared to other network or web-based programming language, was actually originally developed for network-based applications (Proinity, 2017; Harold, 2005). It is fast and easy to write programs in Java that manipulate data over a network that involves serves and clients. More so, NetBeans integrated development environment (IDE) is a Java programming environment integrated into a program application that provides graphical user interface (GUI) builder, a text or code editor, a compiler and or an interpreter and a debugger.

Apart from actualizing a PoC using MySQL and Java's IDE tools, a PoC cannot be complete without running a program to implement the PoC. This study adopted evolutionary prototyping as it enables proof the functionality of the system (Yang & Epstein, 2005) namely using serial a number to authenticate a computer in a wireless LAN. Furthermore, evolutionary prototyping is used when an initial version of a system is developed using the best known and highly prioritized requirements. The initial version of the system is then evolved to a final product through a number of stages by consulting with different stakeholders (wireless LAN proprietors, owners as well as experts) (Carter et al., 2001).

Part of the SNAM demonstration was to perform test runs to prove the model functionality. This could not have been possible without setting up an experiment. Here, static group comparison pre-experimental design, which is part of laboratory experiments method, which in turn is part of empirical evaluation techniques in model evaluation (Rossi & Siau, 2011) was adopted. According to Cash, Stancovic, & Storga, (2016), a static group comparison pre-experimental design is an experimental design that contains two experimental groups. One group is subjected to some activities and the other is not. The results are compared to determine whether change cause or do not cause an impact. Pre-experimental design generally as opposed to true experimental designs involves no random selection of samples in most of the cases.

In the case of this study, the experiment was conducted by setting up a network of two client computers (laptops), a server and an access point as illustrated in Figure 29 of this study. The server hosted the SNAM system that controlled how the clients accessed and made use of the network resources. Two clients were used, one for valid and another for invalid inputs so that the output results could compared. The model, for instance, required that only registered computers be allowed into the network when they log on to the network. One computer was registered and the other was not registered. On test running the prototype, the registered computers were allowed access while the unregistered computer was denied access to the network as illustrated in Figure 41 for unregistered computer and figure 44 for registered computers. The unregistered computer was registered and the application executed once again. Once the two clients were allowed access to the network services, one of the computers was disabled and the other not disabled to test the functionality of kicking a computer out of the network if need be while the other computer continually use the network. The expected outcomes in all the test runs were the binary value yes or no, that is, can the concept be executed as expected or it cannot.

3.2.5 Prototype Evaluation

Between goal-based, goal-free and criteria-based evaluation strategies in information systems (Cronholm & Goldkuhl, 2003), goal-based method was employed in evaluating SNAM prototype. With the focus of being on intended services and outcomes of a program, goal-based evaluation measures the extent to which a program attains its intended goals. It should be measurable and meet the requirements specification.

SNAM was evaluated on a static group comparison pre-experimental design over evolutionary prototyping set up. The setup constituted an access point, a server with the SNAM program, two clients; one for valid and the other for invalid SNAM configurations. Three independent evaluators drawn from Wi-Fi stakeholders evaluated different setups through tests runs based on the following goals;

1. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?
2. Can the system allow a registered computer access to a network?
3. Can the system deny an unregistered computer access to a network?
4. Can the system deny an already logged on computer to a network if need arises?

The responses were recorded in evaluation forms (Appendix VII, Appendix VIII and Appendix IX) and later analyzed as in Table 6

3.2.6 Communication

Apart from the initial results that were presented to selected stakeholders for evaluation, SNAM findings were equally presented to two Kabarak University International Conference on Computing and Information Systems. The first paper (Appendix X) "**A Serial Number Based Identification Model for a Computer in a Wireless Local Area Network**" was presented and captured in *Proceedings of the Kabarak University International Conference on Computing and Information Systems. 8th – 9th October 2018*. The second paper (Appendix XI) titled **"Towards a Unique, Secure, and Robust Wireless Local Area Network Device Identifier**" was presented and captured in the *Conference Proceedings of Kabarak University International Conference on Computing and Information Systems on 14th-15th October 2019.*

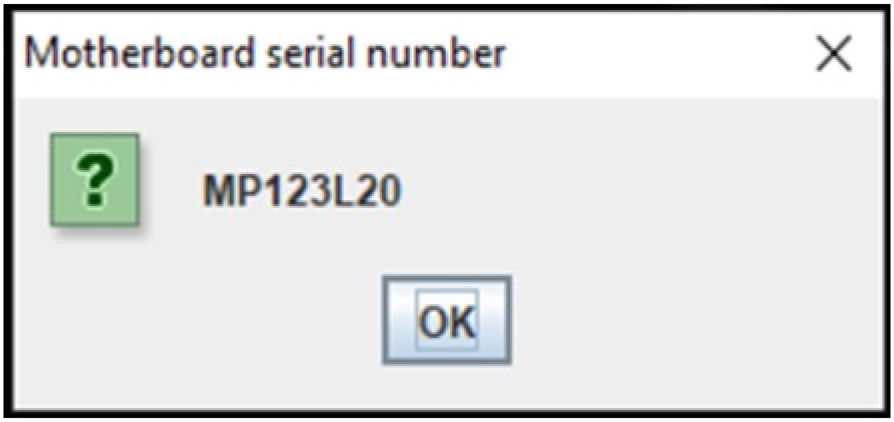
The results of the study were as well published in the *International Journal of Wireless & Mobile Networks (IJWMN)* journal, volume 14, issue number 4 of August 2022 (Appendix XII). The paper was titled "**PROTOTYPING A SERIAL NUMBER BASED AUTHENTICATION MODEL FOR A COMPUTER IN A WIRELESS LOCAL AREA NETWORK**"

3.3 Pilot Study

A pilot study was carried out prior to conducting the actual experiment with the aim of ensuring consistency of the actual experimental outputs regarding required characteristics of an identifier and serial number detail collection. Two sets of tests were carried out; one through a code using the wmic command line tool and the other using Advanced IP Scanner. Through wmic tool, it was found out that a computers serial number can actually be collected from the system as depicted in Figure 35 below with the corresponding code in Appendix V of this thesis. It was during this pilot that it was equally realized that some old computer models do not have their serial numbers encrypted in their hardware.

Figure 35

Pilot test outcome



Motherboard serial number

MP123L20

In the same way, two clients, a server, access points, and connections were as well set up for the sole purpose of piloting on spoofing hence determining the security characteristics of a MAC address. This in turn, demonstrated that a computer's serial number could not be spoofed. To achieve this, an Advanced IP scanner tool was installed and ran from the server. The results of the test are depicted in Figure 5, an indication that whereas a MAC address can be spoofed, a computer's serial number cannot.

3.4 Reliability of the Instrument

Reliability, according to Hawkins, (2016), is the degree to which an assessment instrument produces stable and consistent results either through test-retest, internal consistency or split-half reliability. Test-retest measurement was used in this study through test running the application on two clients for valid and invalid inputs respectively but using the same setup. The two out puts were compared to ensure that the out results to the correct expectation. The results were further confirmed by independent evaluators that evaluated the model on a similar set up using different computers. Furthermore, a pilot study was carried out, with the results in Figure 5 and Figure 8 screen shots, prior to the actual experiment, was carried out to ensure results consistency.

3.5 Ethical Consideration

The study involved the use wmic bios get command line tool to collect serial numbers and Advanced IP Scanner tool to collect and spoof a computers serial number, MAC address and IP addresses during the pilot stage of the study. SNAM prototype as well was developed and used to collect a remote computer's serial number, name and IP address details. A number of computers, many of which were borrowed from well-wishers, were involved during the study test runs. The researcher therefore sought permission and explanation to the computer owners about the test and details their computers would be subjected to. The stakeholders involved in the prototype evaluation were equally informed about what entailed the test runs and were assured confidentiality (Appendix VI) in the feedback and the evaluations that were carried out.

3.6 Research Authorization

Before the research was conducted, permission was sort from the National Commission for Science, Technology, Innovation, and Communication (NACOSTI) as part of requirements for carrying out the research. The letter for research authorization and the permit obtained are presented in Appendix II and Appendix III of this study respectively. But a letter for the introduction was first done by the postgraduate school as presented in Appendix I

3.7 Chapter Summary

This chapter of the study presented details on how design science research (DSR) was adopted for the study. Consequently, it presented information on problem identification and motivation, definition of the objectives for the solution, design and development SNAM model using state chart diagrams and flowcharts, demonstration of the model using MySQL and Java's IDE by PoC over evolutionary prototyping, evaluation by stakeholders on static group comparison pre-experimental design, and communication of the SNAM model during two international conference. Details about the pilot study, reliability, ethical considerations and research authorization were also presented. Chapter four below presents the results and discussions of the study.

CHAPTER FOUR

DATA ANALYSIS, RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the findings and results on the design, development, demonstration and evaluation of the serial number based authentication model (SNAM) perceived to solve the challenges of using other authentication methods, notably MAC address authentication, for authenticating computers in a network.

4.2 Algorithm Design for SNAM

The ultimate goal of the study was to demonstrate how a computer's serial number can be used to authenticate a computer in wireless LAN. Authentication details that included the serial number, the IP address, the computer's name and connection status had to eventually be displayed on a display interface for further manipulation. Apart from the display interface, other several sections of the system that included computer registration subsystem, authentication details collection module, the computer authentication module, computer authorization module, authentication details display interface itself and controlling valid computers module were designed to recursively culminate to the overall design. Before displaying and managing the authentication details on the authentication details display interface, for instance, fundamental computer authentication details, that is serial and name, have to be registered first, then pre-existing connected computers details are deleted to pave way for newly connected computers authentication details, then get connected computer IP address, name and serial number details of connected computers, collate computer authentication details (IP address, name and serial number), then allow registered computers access the network while denying network access to unregistered ones, then post valid computers details to the database, then retrieve and eventually display the details on the authentication details display interface, had to be designed and implemented in that order.

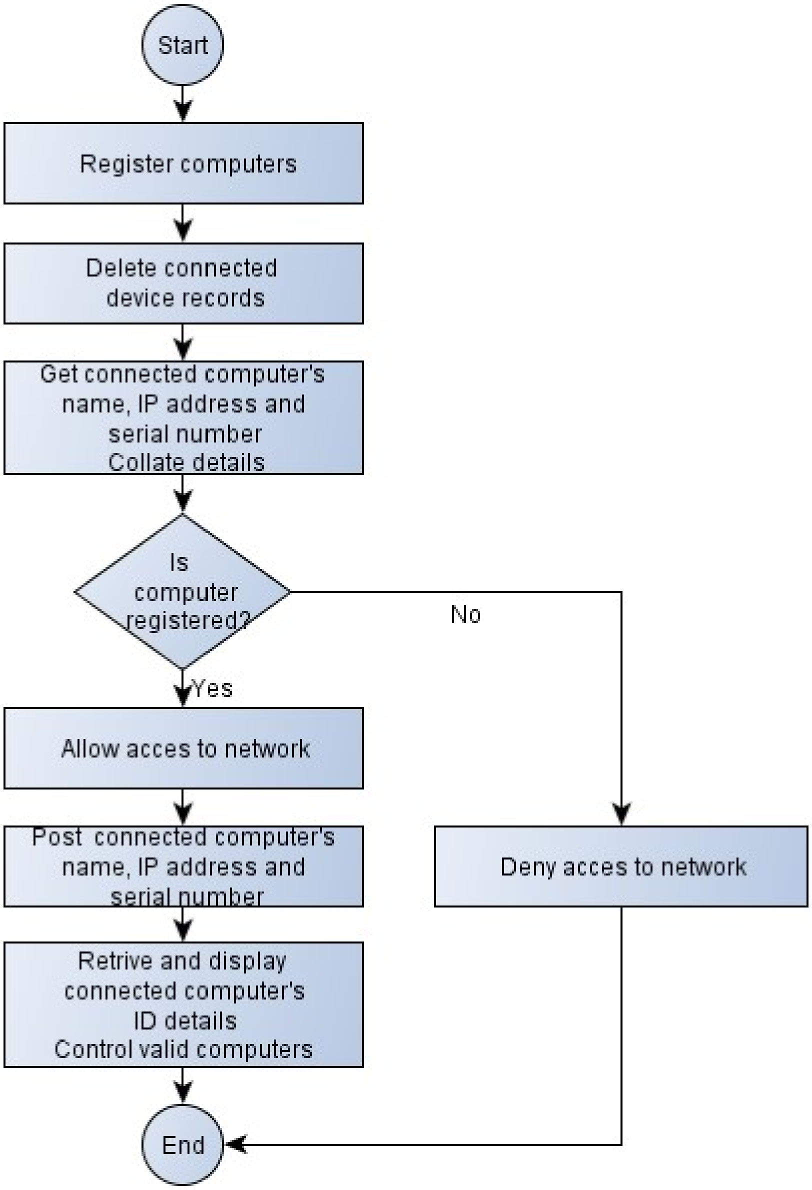
A dynamic programming design was therefore adopted during the algorithm design development phase of the study. This was informed by the fact that a dynamic programming algorithm optimizes solutions on a step by step basis recursively to the whole (SmartDraw, 2018; Visual-paradigm, 2018; Paramalways, 2009). In other words, the results of one step solve the problem of another consecutive step recursively. The decision taken in one step is dependent upon an immediate previous or later decision steps. As a result, dynamic programming is absolutely the right method for optimization of combinations of ordered interdependent solutions.

The algorithm with a corresponding flow chart (Figure 36) that depicts the system description is as follows:

```
1. Start
2. Register computers
3. Delete existing connected computers details
4. Get connected computer details
   4.1 Get the computer name
   4.2 Get computer raw IP address
   4.3 Get computer serial number
   4.4 Collate computer IP address, name and serial number
5. If computer is registered
   5.1 Allow computer connect to the network
   5.2 Post Connected Computer Details to the Database
   5.3 Retrieve and Display the Connected Computer Details from Database
   5.4 Control validly allowed computers
6. Else, deny computer network access
7. End
```

Figure 36

General flow chart for SNAM system


Start
Register computers
Delete connected device records
Get connected computer's name, IP address and serial number Collate details
Is computer registered?
No
Yes
Allow acces to network
Post connected computer's name, IP address and serial number
Deny acces to network
Retrive and display connected computer's ID details Control valid computers
End

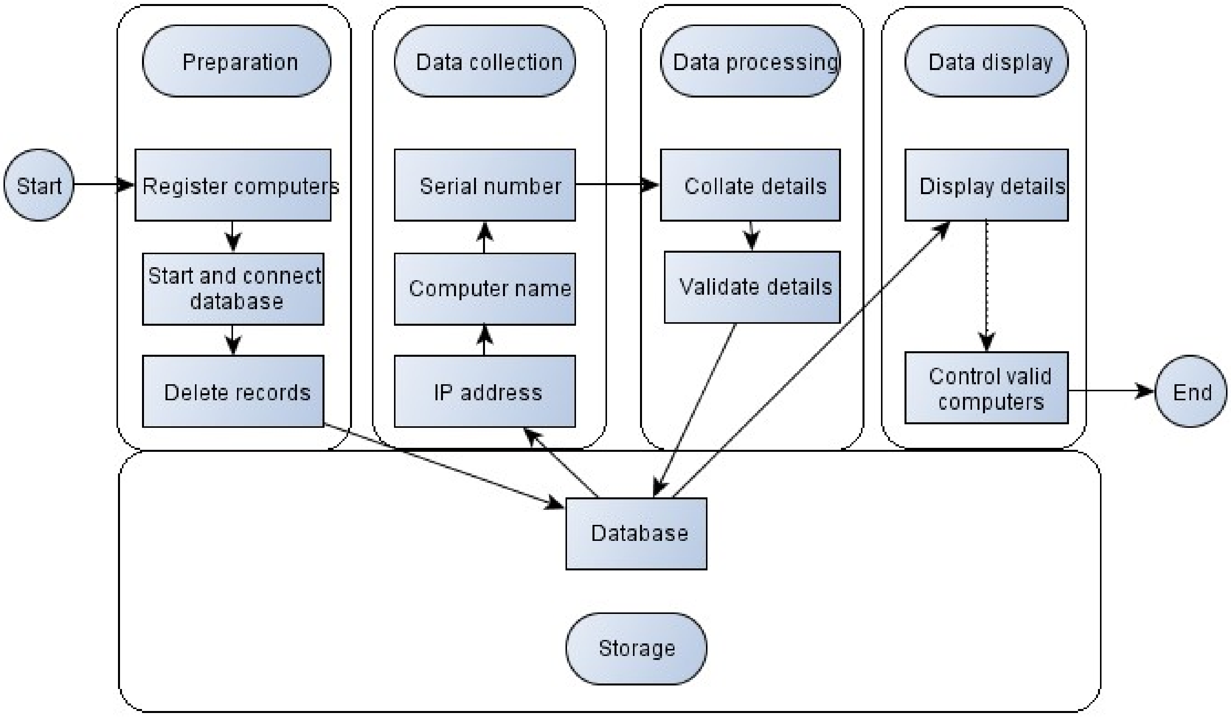
4.3 Serial Number Authentication Model Architecture

An information systems modelling method according to Rossi & Siau, (2011), is a method that uses a small of constructs to define the vocabulary of the method. The constructs include the concepts, ideas or images that are perceived to constitute the model for the purpose of organizing and representing knowledge of interest. Such models include data flow diagrams (DFD), entity-relationship (ER) diagrams and unified modelling language (UML) consortium.

The generalized model for authenticating computers based on the serial numbers comprise the data storage, data preparation, data collection, data processing and data display modules as demonstrated in Figure 37 below.

Figure 37

General model for a serial number based authentication system



The model is based on a state chart diagram model because it relates a number of states and the transitions between these states (Bruegge & Dutoit, 2010). Three basic components of a state diagram are initial state (represented by a black filled circle), transition (solid arrows) and state (a rounded rectangle). Management of the authentication details begins from when the computers are registered all the way, to when the computer is either allowed access or denied access to the network. All the SNAM components and much more are described in the section below.

4.3.1 The Storage Subsystem

The storage subsystem contains the structure that facilitates the storage and access of IP address, computer name and serial number details that are required for authentication. The database is located at the server, which is centralized in this case. Every time the program is run, the storage subsystem clears existing details in its structures to pave way for fresh details from connected devices.

The SNAM system contains one sytem_db database and two relations namely the condevices and computers tables. With only two columns; Serial Number and Computer Name, Computers table stores and manages registered computer registration details. The structure of the computers is as in Table 3 below.

Table 3

Computers system relation structure

#	Column Name	Data Type	Column Size	Primary Key	Indexed
1	Serial_Number	VARCHAR	50	NO	NO
2	Computer_Name	VARCHAR	50	NO	NO

Condevices table on the other hand, with four fields (Serial_Number, Computer_Name, IpAddress and Status, stores and manages connected computer details. Details of the relation is as illustrated in the Table 4 below

Table 4

Condevices system relation structure

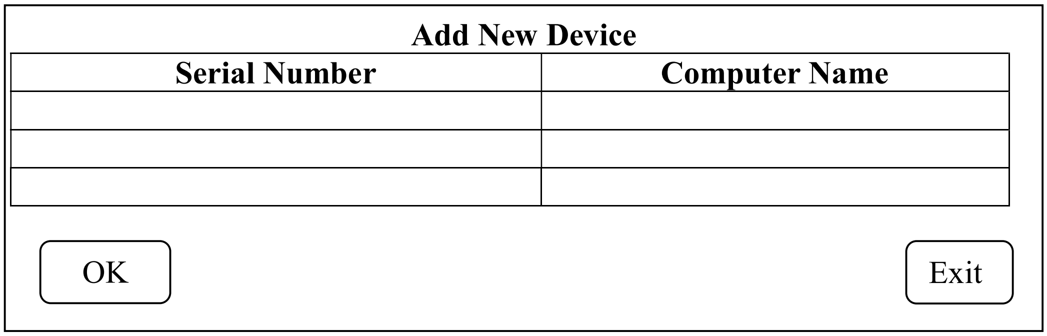
#	Column Name	Data Type	Column Size	Primary Key	Indexed
1	Serial_Number	VARCHAR	50	NO	NO
2	Computer_Name	VARCHAR	50	NO	NO
3	IpAddress	VARCHAR	50	YES	YES
4	Status	VARCHAR	50	NO	NO

4.3.2 The Data Preparation Module

Data preparation module involves computer registration interface, database connection, deleting of existing records and preparing the database for new records components. The registration interface enables addition of a new computer to the system so that it can later be validated upon to either be allowed or denied network access. Registered devices are validated to access network services while unregistered devices are denied entry to the LAN. Figure 38 below illustrates how the registration interface was designed.

Figure 38

Add New Devices registration interface design



Add New Device

Serial Number	Computer Name

The other item in the initialization component is the starting and connection to the database section. As a practice in any application that involves data management, a connection between the database and the application has to be established. In this program, the process is set to run every time the application is started. As soon as the database connection is established, the initialization subsystem embarks on deleting existing records to pave way for the details of the newly logged on computers. This way, the database is prepared to record details of newly connected computers.

4.3.3 The Details Collection Subsystem

The responsibility of the data collection module is to fetch the IP address, computer name and serial number details necessary of the computer identification process. The IP address, computer, and serial number details are both from the local host computer and other computers connected to the network.

4.3.4 The Details Collation Module

As part of the other subsystems, the details collation module presents and combines the IP address, computers name and serial number and stores them in the database for subsequent processes to be effected. It first resolves the hostnames and IP addresses, then performs IP address format conversions (from string to byte) and ultimately combines the three details. In addition, the hostname, IP address and serial details are as well posted to the database for the manipulation or validation purpose.

4.3.5 The Validation Subsystem

One of the paramount component of the SNAM system is the validation subsystem. Here, connected computers are validated against the registered ones with the purpose of either allowing or denying the computers access to the network services and resources. While registered computers are allowed access, unregistered computers are denied access to the network

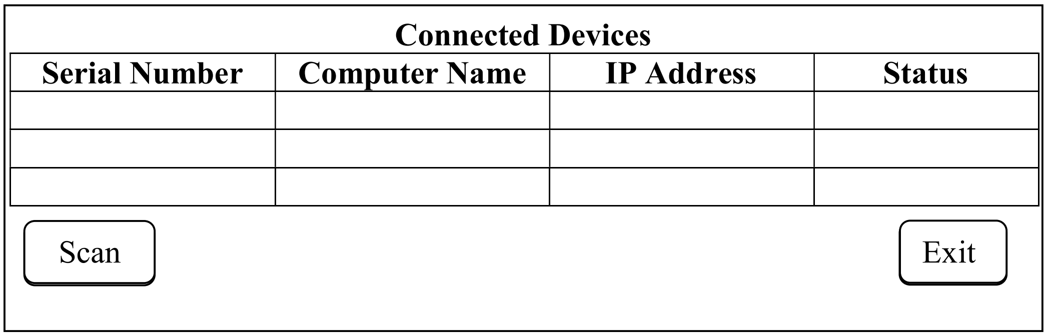
4.3.6 The Display Subsystem

The ultimate detail for authentication is the serial number record. However, for the number to be used as an authenticator, then it must be displayed on an interface. The display subsystem is designed therefore to access, apart from the serial number, the name, and IP address details that are essential for device authentication.

After the authentication details are collected, an interface had to be created that would eventually allow the authentication details to be displayed. The sketch of the form was designed to include the serial number, computer name, IP address and status details table as well as include a scan and an exit button as in Figure 39 below.

Figure 39

Authentication details interface sketch



Connected Devices

Serial Number	Computer Name	IP Address	Status

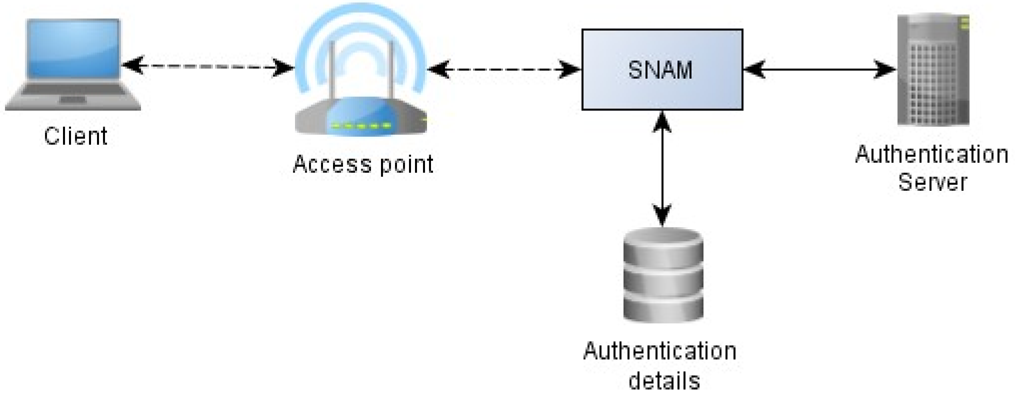
The nature of the system prompted the choice of Java NetBeans in developing the interface. Java, as compared to other network or web-based programs, was actually originally developed for network-based applications (Harold, 2005). It is fast and easy to write programs in Java that manipulate data over a network that involves servers and clients. NetBeans integrated development environment (IDE) is a Java programming environment integrated into a program application that provides graphical user interface (GUI) builder, a text or code editor, a compiler and or an interpreter and a debugger.

4.4 SNAM Model Components

In this section, the components of the model are presented. The model involved the client (transponder), an access point (AP), the application itself and the server. Just as in any other computer related issues, components of a system function in such a way that all should work together for a common goal, without which the whole system cannot function. The interconnection of these components is shown in Figure 40 below.

Figure 40

The SNAM system components



4.4.1 The Client

Laptops were used as clients for the purpose of the study in demonstrating how a serial number can be used as an inbuilt identifier for devices in a network. Apart from tagging serial numbers, laptops and modern computers, in general, have their serial number coded on to their system board making it accessible by a program.

4.4.2 The Access Point

Just as a requirement in a WLAN, an access point (AP) or wireless access point (WAP) is a valuable component of the SNAM model. An AP allows Wi-Fi devices to be connected to a wired LAN as well as within the WLAN. As such, communication between the server and clients logged onto the networks cannot take place without an AP.

Although three different APs were used to test-run the prototype for comparison purposes, the main one was a DP-LINK 3G/4G router device. The major configurations made on the AP included;

i. Address configuration of IP address: 192.168.0.1, subnet mask: 255.255.255.0

ii. Service set identifier (SSID) name: CheborAP with a Password

iii. Wireless security login credentials username: admin and password: admin

iv. DHCP basic configurations were done as follows: DHCP server: enable, Start IP Address; 192.168.0.100, End IP Address: 192.168.0.200, Address Lease Time: 120 minutes and Default Gateway: 192.168.0.1

4.4.3 The Authentication Server

The authentication server component plays a key role in the SNAM system. It primarily hosts the application used to identify connected computers based on their serial numbers. Here, connected computers can be authenticated and are controlled as well, a network administrator responsibility.

The reason for a must to have a server is derived from its DNS capabilities. In other words, the serial number based authentication is a child process of the IP address process. Since the computer name is also required as part of the identification process, the server as a DNS comes in as a resolver of IP address to a hostname and vice versa.

4.4.4 The SNAM Model Database

The ultimate goal of the SNAM system was to authenticate a device logged on a network using its serial number. The serial number, together with other authentication details namely the hostname and IP address had to be stored in a database. It is from the database that the details can easily be managed and manipulated.

MySQL was the choice database in the SNAM system. MySQL is a freely available open relational database management system (RDBMS) that uses the structured query language (SQL) platform. The choice of MySQL is based on the fact that apart from its popularity in storing and manipulating data, MySQL is simple, easy, fast, reliable and flexible (Nixon, 2018). Therefore, very applicable in prototyping simple projects

4.5 SNAM Implementation and Testing

As explained earlier in the serial number authentication model architecture section, the nature of the system prompted the choice of Java NetBeans in developing the interfaces and system modules. Java, as compared to other network/web-based programs, was actually originally developed for network-based applications (Harold, 2005). It is fast and easy to write programs in Java that manipulate data over a network that involves serves and clients. NetBeans integrated development environment (IDE) is a Java programming environment integrated into a program application that provides graphical user interface (GUI) builder, a text or code editor, a compiler and or an interpreter and a debugger.

In the same way, MySQL was the choice database in the SNAM system. MySQL is a freely available open relational database management system (RDBMS) that uses the structured query language (SQL) platform. The choice of MySQL is based on the fact that apart from its popularity in storing and manipulating data, MySQL it is simple, easy, fast, reliable and flexible (Nixon, 2018). Therefore, very applicable in prototyping simple projects

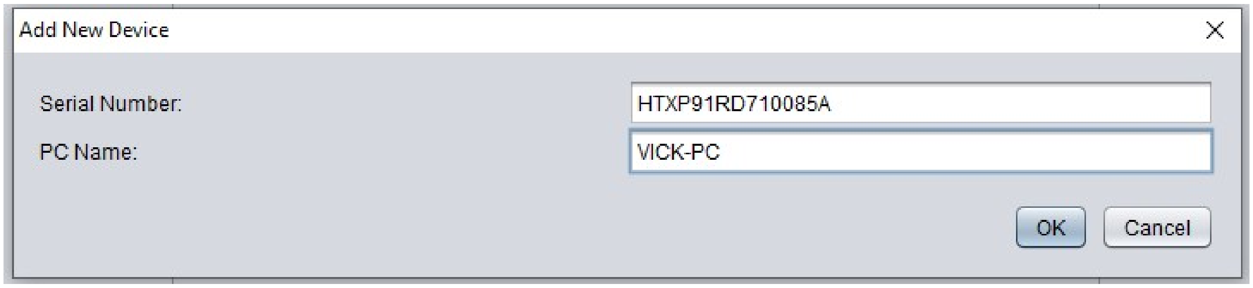
The tasks carried out under this subsection were as follows:

4.5.1 New Computers Registration

In respect to registration interface design in Figure 28 above, a registration interface was created that contained two fields (Serial Number and PC Name) and two buttons (OK and Exit) as shown in Figure 41 below.

Figure 41

Computer registration interface



Serial Number: HTXP91RD710085A

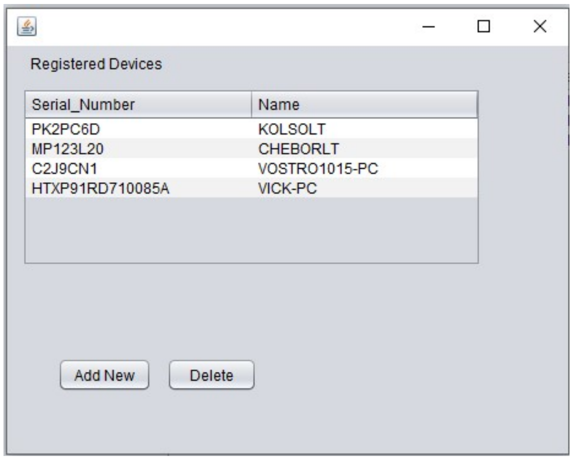
PC Name: VICK-PC

The interface is executed by either clicking New Device button on the Connected Devices interface in the details display interface in Figure 55 (i)

Or clicking on the Add New buton on the Registered Devices interface as shown in the registered computers display interface Figure 42 below.

Figure 42

Registered computers display interface



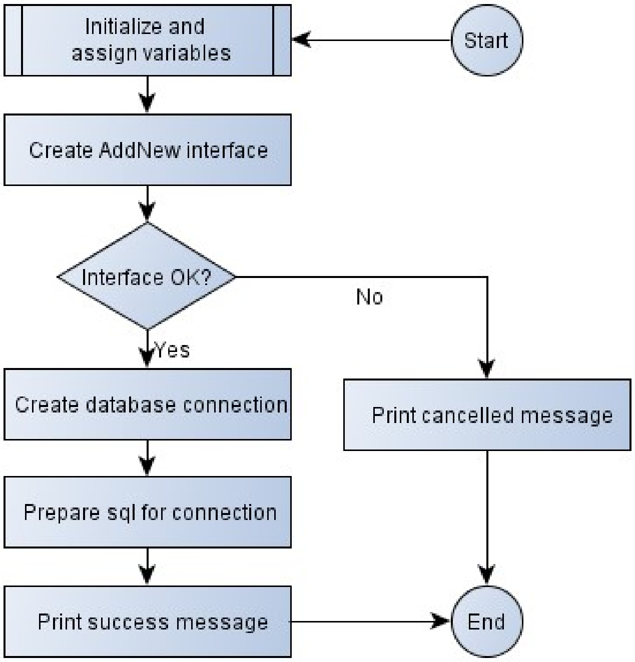
Registered Devices

Serial_Number	Name
PK2PC6D	KOLSOLT
MP123L20	CHEBORLT
C2J9CN1	VOSTRO1015-PC
HTXP91RD710085A	VICK-PC

The Registered Devices interface, apart from allowing new devices to be registered via the Add New Device button and displaying registered dcomputers, can equally enable delete a registered computer in case a need arises.

Figure 43

Computer registration flawchart



Following here below is the algorithm and code used to register computers and its corresponding flow chart (Figure 43) and code

The algorithm

```
1. Start
2. Initialize and assign variables
3. Create AddNew interface
4. If OK
   4.1 Create database connection
   4.2 Prepare sql for connection
   4.3 Print success message
5. Else print cancelled message
6. End
```

The code

```
public void addNew() {

    JTextField serialNumber = new JTextField(15);

    JTextField pcName = new JTextField(30);

    JPanel panel = new JPanel(new GridLayout(2, 2));

    panel.add(new JLabel("Serial Number:"));

    panel.add(serialNumber);

    panel.add(new JLabel("PC Name:"));

    panel.add(pcName);

    int reslt = JOptionPane.showConfirmDialog(null, panel, "Add New Device",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
```

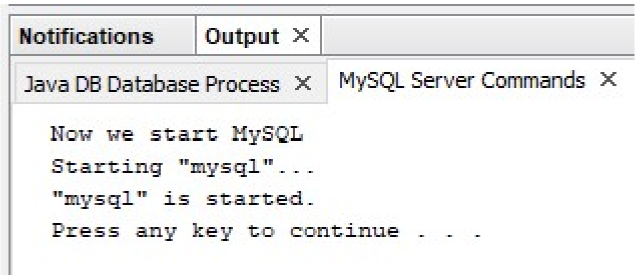
```
        if (reslt == JOptionPane.OK_OPTION) {

            //add to db

            try {

                con = ConnectDB.connect();//create a connection

                String sql = "INSERT INTO devices (Serial_Number,Name)values('" +
serialNumber.getText() + "','" + pcName.getText() + "')";

                pst = con.prepareStatement(sql);//prepare sql for execution

                pst.execute();

                System.out.println("saved succeed");

                data1();

//JOptionPane.showMessageDialog(sysInfo,"Successfully
saved","Record",JOptionPane.INFORMATION_MESSAGE);

            } catch (SQLException e) {

                System.out.println(e);

            }

        } else {

            System.out.println("cancelled");

        }

    }
```

4.5.2 Starting the Database

The database was designed as shown in Table 2. It contained one table with three fields namely: serial number, computer name, and IP address. The database was hosted in a server and as such, it has to be started every time the application runs since normally in such scenario the database stops when the server hosting it is off. Hence, it has to be started afresh as demonstrated in Figure 44 below

Figure 44

Starting MySQL database



```
Now we start MySQL
Starting "mysql"...
"mysql" is started.
Press any key to continue . . .
```

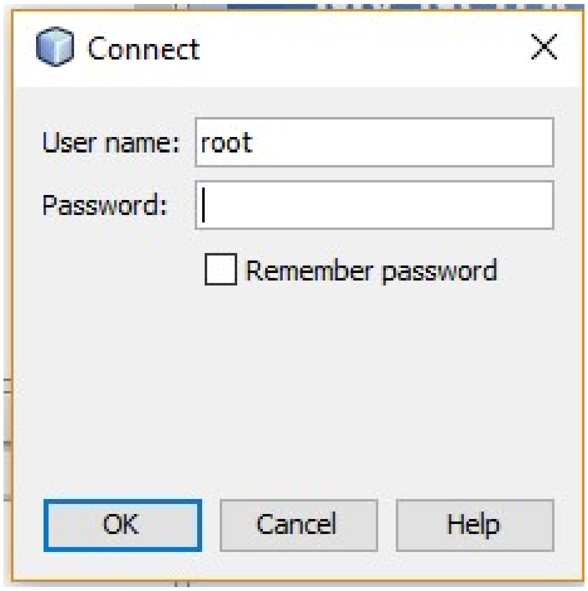
Once the database is started, it had to be connected to the system.as presented in the next section.

4.5.3 Database Connection

The system uses the Java Database Connectivity (JDBC) tool to create a connection between the MySQL database and the system application. Apart from allowing clients access data from the server, the JDBC as well provides techniques of querying and updating the computer name, computer serial number and IP address details.

Figure 45

Database Connection interface



User name: root

Password:

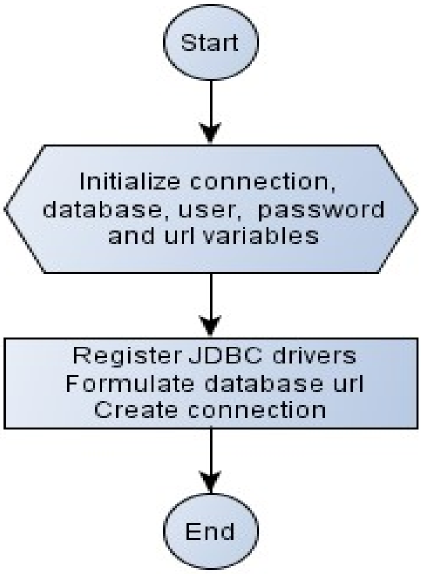
□ Remember password

The algorithm with the corresponding flowchart (Figure 46) used to develop this section was as follows:

1. Start
2. Initialize variables
3. Register JDBC drivers
4. Formulate database URL
5. Create a connection object
6. End

Figure 46

Database connection flowchart



The coding of the connection module involved

i. The inclusion of import statements to import JDBC packages required in the java code

ii. Registration of JDBC driver using the method Class.forName() to cause the JVM to dynamically load desired drivers' implementation into the memory so that it can fulfill the required JDBC requests. The Class.forName() method is common and allows driver registration configurable and portable.

iii. Database URL formulation was used to create a properly formatted address that points to the database. This is done after loading the driver. The three overloaded DriverManager.getConection() methods are DriverManager.getConnection("jdbc: mysql://localhost:3306/system_db"," root","") for String url, String user and String password

iv. Creation of a connection object that codes all the DriverManager object's getConnect() method to establish the actual databases connection bypassing the database url, a username*,* and a password

All this and more are illustrated in the code line below.

```
public class ConnectDB {
   Connection con = null;
   static String db = "AppServ";
   static String user = "Java";
   static String pass = "root";
   static String url ="jdbc: mysql://localhost:3306/";
   public static Connection connect () {
      try {
         Class.forName("com.mysql.jdbc.Driver");
         Connection                                                            con
=DriverManager.getConnection("jdbc:mysql://localhost:3306/system_db","root","");
         return con;
      }
catch (ClassNotFoundException | SQLException e)
{
         JOptionPane.showMessageDialog(null, e);
         return null;
      }
   }
```

Once connected the database has to be cleared of any previous records of formally connected devices so as to start storing details of currently connected devices. The procedure to delete existing records is presented next.

4.5.4 Deleting Existing Records

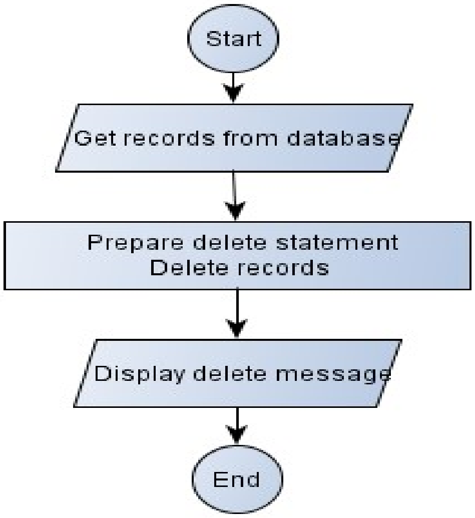
The program deletes all existing records to pave way for the records of the newly connected devices. However, before it deletes the records it first gets the records from the system then prepares a delete statement for delete confirmation then deletes the records. All these are done using the following algorithm and illustrated by the flowchart in Figure 47 below

The algorithm and corresponding flowchart (Figure 47) for this section is as follows;

1. Start
2. Get records from the database
3. Prepare delete statement
4. Delete records
5. Display record deleted message
6. End

Figure 47

Deleting existing records flow chart



The corresponding code was as follows:

```
public void recordDel() {
        try {
            String sqldel = "DELETE FROM condevices";
            PreparedStatement pdel = con.prepareStatement(sqldel);
            pdel.executeUpdate();
            System.out.println("Records Deleted");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

After clearing the database, it is now ready to store details of currently connected devices. To do this all the relevant details to be stored have to be collected for storage. The procedure to do this is presented next.

4.5.5 Getting the Computer's IP Address

The process of obtaining a computers serial number for authentication starts first from getting the computers IP address, then the computer's name, and eventually using the IP address to get the serial number. An IP address here acts as a stepping stone in obtaining the computers name but more so, the computers serial number.

This section first retrieves the raw IP address in string format but later has to be expressed in a numeric format. The algorithm used to extract the raw IP address is as follows with its corresponding Figure 48 below:

1. Start
2. Get IP address based on hostname

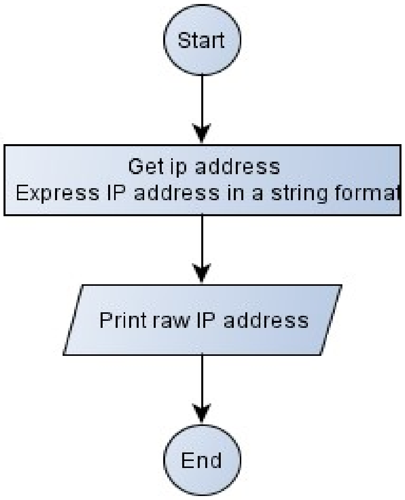
3. Express IP address in string format

4. Print the raw IP address

5. End

Figure 48

Getting the computer's IP address flow chart



The corresponding code follows:

```
public void getIP(String nm) {
    try {
      inetAddress = InetAddress.getByName(nm);
       ipAddress = inetAddress.getHostAddress();
                System.out.println(ipAddress);
    } catch (Exception e) {
             JOptionPane.showMessageDialog(null, e);
    }
}
```

4.5.6 Getting a Computer's Name

A computers name is one of the fundamental characteristics that describe a computer among other features such as the model, operating system, type. Apart from facilitating communication and troubleshooting, a computers name here is paramount in associating the identity of a computer with its serial number. The getCanonicalHostName() method in the subsequent code section is used to retrieve a computer's name passed to a variable Computer_name in the database.

```
Computer_Name = address.getCanonicalHostName();
```

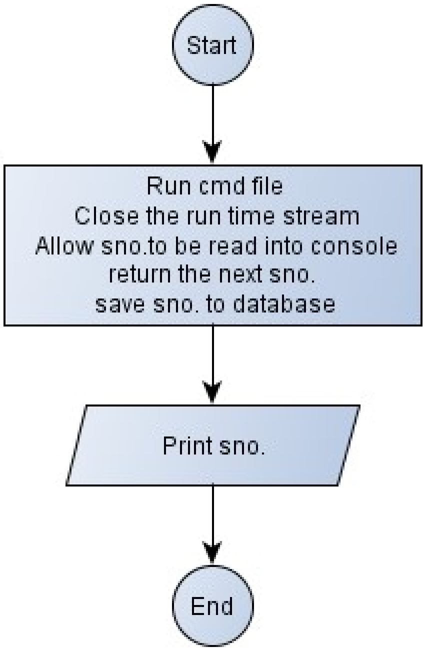
4.5.7 Getting the Computer's Serial Number

As stated in the computer's serial number location section of this study, the inbuilt serial number of a computer can be obtained using the command-line interface (CLI) command (CMD) wmic bios get serialnumber. The SNAM uses this command to load the serial number from the BIOS to the database for processing. The algorithm and the corresponding flowchart (Figure 49) were done as follows:

1. Start
2. Run cmd file
3. Close the run time stream
4. Allow serial number to be read into the console
5. Return the next serial number
6. Save the serial number in a database
7. Print the serial number
8. End

Figure 49

Getting the computer's serial number flowchart



This resulted in the following code:

```
public void getSerial() {
        try {
            Process process = Runtime.getRuntime().exec(
                    new String[]{"cmd.exe", "/c", "wmic", "/NODE:", ipoutput, "bios", "get",
"serialnumber"});
            process.getOutputStream().close();
            Scanner sc = new Scanner(process.getInputStream());
            String property = sc.next();
            serial = sc.next();
            System.out.println(property + ": " + serial);
        } catch (Exception e) {
        }
    }
```

Once the serial number is obtained, it has to be linked with the IP address. The procedure to do this is presented next.

4.5.8 Posting Connected Computer's Details to a Dataset

The system, as stated earlier relies on the computers IP address as a pointer to get the serial number. Of cause, an IP address here is essential in communicating the computers serial number. The serial number is meant merely for identifying the computer. It therefore calls for IP address and serial number association and the computers name to an extent. The Java network programming class called the inetAddress enables associate the details. This class is used with other networking java classes to manipulate the computer's hostnames and IP addresses (Harold, 2013).

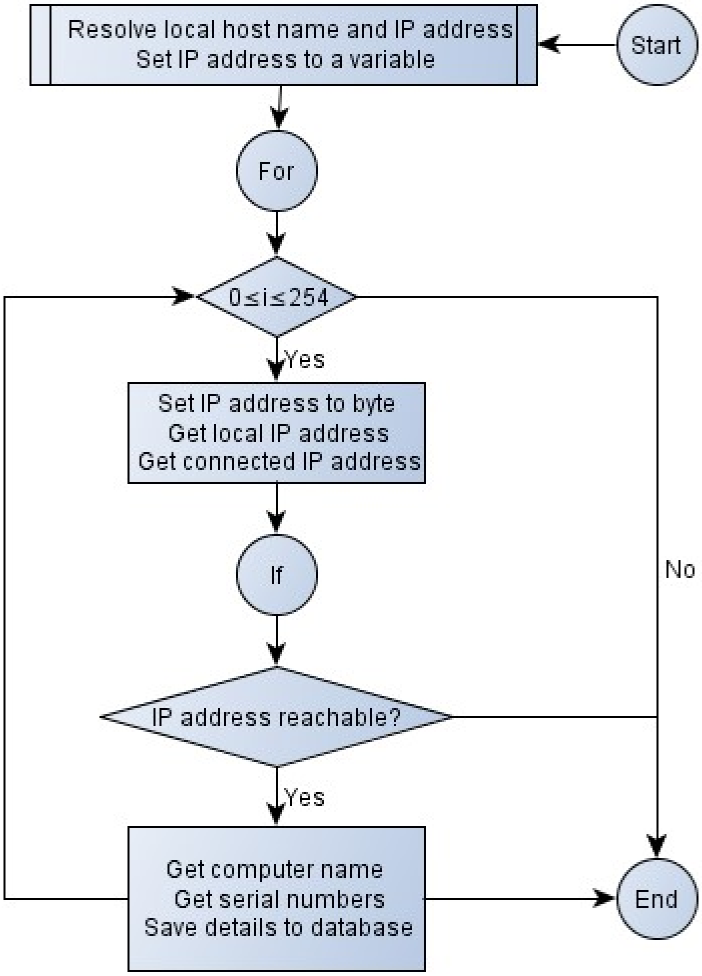
In this section, the function getCompName() was created to associate the serial number basing on the hostname and IP address of both the local and remote hosts. The class is enabled to throw an UnknownHostException error if an address is not found. The code running on the DNS then starts by connecting to the DNS to resolve local hostname from the IP address. This is followed by a variable creation for the IP address assignment. After representing the IP address in a byte format, the code gets the local IP address, the connected computer IP addresses, the connected computers serial numbers then post the details to a database. All these are illustrated in the algorithm and corresponding flow chart (Figure 50) below.

```
1. Start
2. Resolve local hostname and IP address
3. Assign the IP address a variable
4. For IP address between 0 to 254
   4.1 Set byte representation from a string representation of the IP address
```

```
    4.2    Get local IP address
    4.3    Get connected computers IP address
    4.4    If the address is reachable, then
        4.4.1    Set Computer name
        4.4.2    Get computer serial number
        4.4.3    Post IP address, Computer Name, and serial number to a dataset
5.  End
```

Figure 50

Posting details to dataset



The corresponding code is as follows:

```
    public void getCompName() throws UnknownHostException {
        InetAddress localhost = InetAddress.getLocalHost();
        byte[] ip = localhost.getAddress();
        for (int i = 0; i <= 254; i++) {
            try {
                ip[3] = (byte) i;
                InetAddress address = InetAddress.getByAddress(ip);//local ip address(server ip)
                ipoutput = address.toString().substring(1);//get the computer ipaddress
                if (address.isReachable(500)) {
                    Computer_Name = address.getCanonicalHostName();
                    getSerial();
                    System.out.println("COMPUTER NAME: " + Computer_Name);
                    System.out.println("COMPUTER IP: " + ipoutput);
String sqlins = "INSERT INTO condevices
(IpAddress,Computer_Name,Serial_Number)values('" + ipoutput + "','" + Computer_Name +
"','" + serial + "')";
                    pst = con.prepareStatement(sqlins);
                    pst.execute();
                    String sql1 = "SELECT IpAddress,Computer_Name,Serial_Number FROM
condevices";
                    PreparedStatement pst1 = con.prepareStatement(sql1);
                    ResultSet rs1 = pst1.executeQuery();
TblConnectedDevices.setModel(DbUtils.resultSetToTableModel(rs1));
                }
            } catch (Exception er) {
            }
        }
```

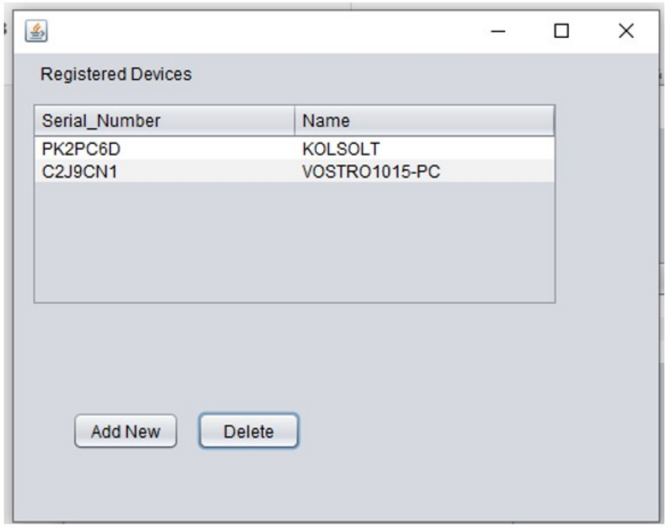
4.5.9 Allowing or Denying a Computer Connect to the Network

Allowing or denying a computer into the WLAN is one of basic requirement of network access controls. The use of a serial number as an identifier comes into play here due to its uniqueness characteristics as compared to other identifiers. The factor used to deny or allow a computer into a network in this study is whether a computer is registered or not. The study works on the assumption that a computer has to be registered first before accessing the network services and resources, otherwise, the computer would be denied.

This module was demonstrated by registering two computers in the data set leaving out one connected computer with serial number MP123L20 named CHEBORLT as shown in the Registered Devices Interface Figure 51 below.

Figure 51

Two registered computers



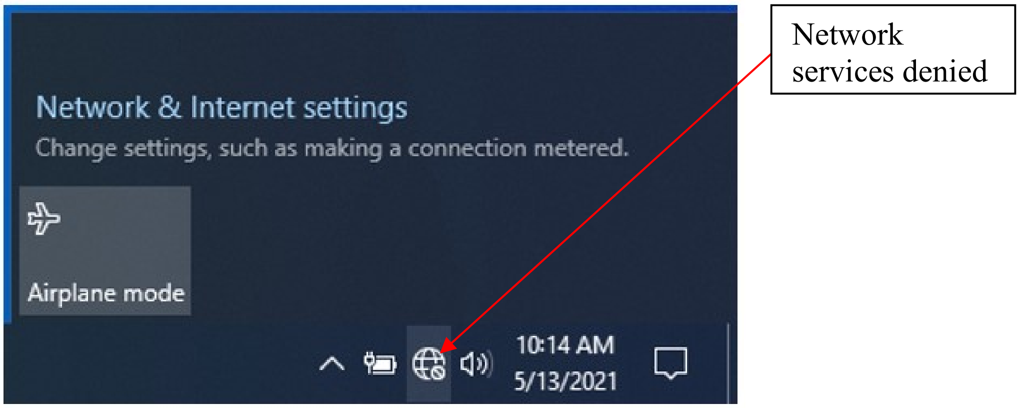
Registered Devices

Serial_Number	Name
PK2PC6D	KOLSOLT
C2J9CN1	VOSTRO1015-PC

On running the application, the network access icon cannot display the Wi-Fi connection interface for the unregistered CHEBORLT computer when clicked. This demonstrated that the unregistered computer cannot be connected to the network as displayed in the said computers network connection interface as shown in Figure 52 below.

Figure 52

Blocked unregistered CHEBORLT computer



Network & Internet settings

Change settings, such as making a connection metered.

The process of disallowing a computer access a network involved the creation of a code that first, calls and initiates connection to the system database. This is followed by preparing and executing registered computer details in the computer relation. At this point, the program loops through the details one by one as it keeps records of them in the memory space created. Ip addresses are then filtered to allow the display of IPv4 that was finally associated with serial number to ensure that only registered computers access the network services and resources.

Below are corresponding algorithm design, flow chart model and the code used to implement the denying and allowing a computer to network resources and services function

The algorithm

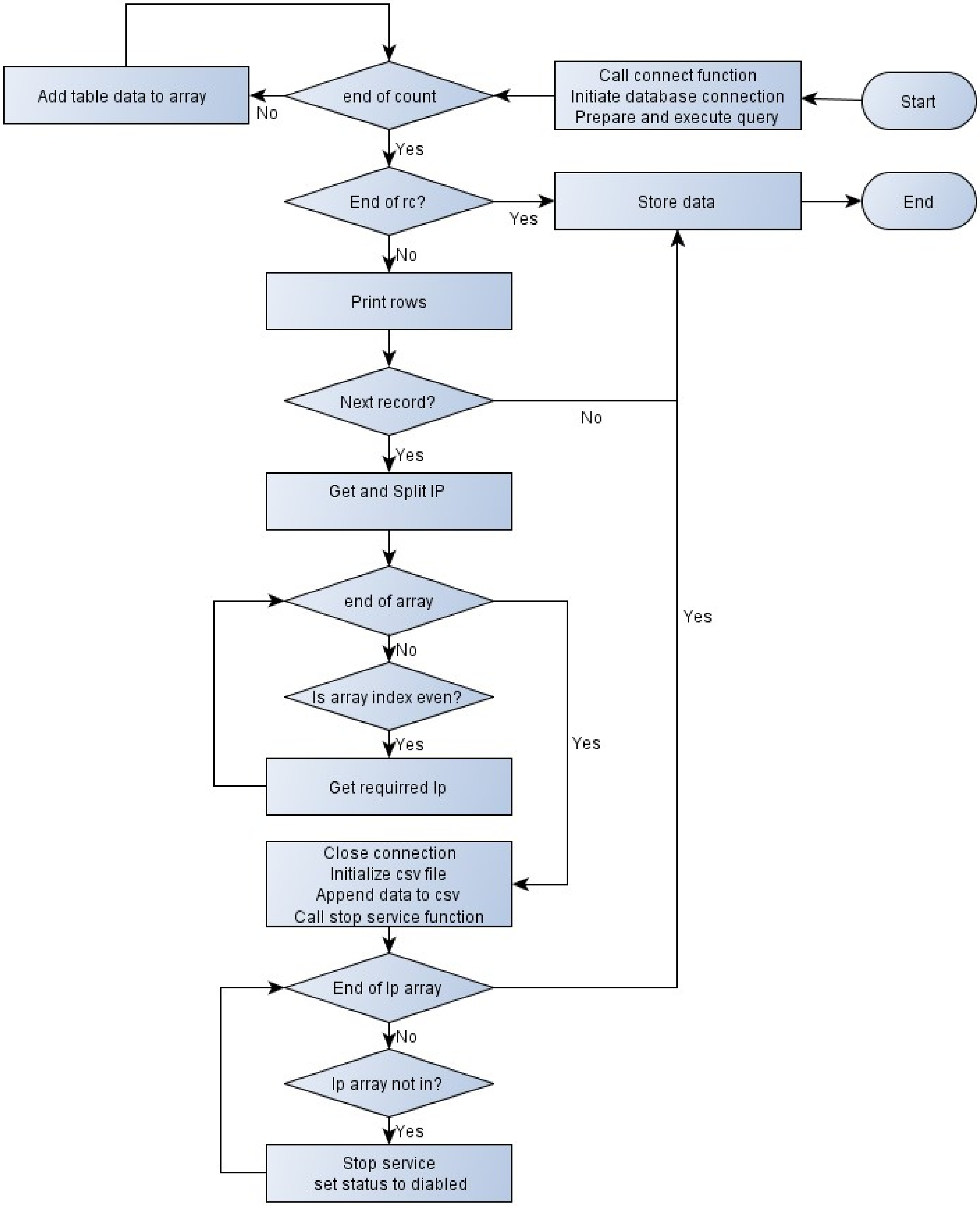
1. Start
2. Call connection function
3. Initiate database connection
4. Prepare and execute query

```
5.  While not at the end of data list
    5.1 Add table data to array
    5.2 End for loop
6.  While not at the end of the row
    6.1 Print and store rows one at a time
    6.2 Is there another record?
        6.1.1   Get and split Ip using a comma
        6.1.2   Else store data
    6.3 While not at the end of record array
        6.3.1   If index array is even
                Get required Ip
        6.3.2   End if
        6.3.3   End for loop
    6.4 Close connection
    6.5 Initialize csv file
    6.6 Append data to csv
    6.7 Call stop service function
    6.8 While not at end of Ip array
        6.8.1   If Ip is not null
            6.8.1.1 Stop service
            6.8.1.2 Set status to disabled
        6.8.2   End if
    6.9 End for loop
7.  End for loop
8.  Store data
9.  End
```

The chart

Figure 53

Allowing or denying a computer network access flowchart



The code

```
public void sysInitiate() {
    try {//this tries to select data from database in case of failure a sql exception error is caught
and prevents the program from crashing
      con=ConnectDB.connect();//calls the connect function from connectDB class that
initiates a database connection
          String sql = "SELECT c.serial_Number "
          + "FROM computers c "
          + "LEFT JOIN devices d "
          + "ON d.Serial_Number=c.Serial_Number "
          + "WHERE d.Serial_Number IS NULL";
      pst = con.prepareStatement(sql);//prepares query for execution
      rs = pst.executeQuery();//holds executed query results to result set variable
tblUnregistered.setModel(DbUtils.resultSetToTableModel(rs));//populate the data to table
      List<String> numdata1 = new ArrayList<String>();//create an array list
      int rc = tblUnregistered.getRowCount();//integer variable that stores no of rows in the
table
      for (int count = 0; count < rc; count++) {
        numdata1.add(tblUnregistered.getValueAt(count, 0).toString());//adds all table data to
array list
      }
      //print value one at a time
      for (int i = 0; i < rc; i++) {
        System.out.println(numdata1.get(i));
        //get the ip address of each serial and add to csv(coma separated values)
        String sql1 = "SELECT * FROM computers WHERE Serial_Number='" +
numdata1.get(i) + "'";
        pst = con.prepareStatement(sql1);
        rs = pst.executeQuery();
        if (rs.next()) {//if another record until over/check for next record in rs
          String ipA = rs.getString("IpAddress");//get ip
```

```
            String[] ipoutElements = ipA.split(",");//
            List<String> fixedLengthList = Arrays.asList(ipoutElements);
            ArrayList<String> listofstring = new ArrayList<String>(fixedLengthList);
            String iparray[] = new String[listofstring.size()];
            //String iparray[] = listofstring.toArray(new String[listofstring.size()]);
            int c;
            for (c = 0; c < listofstring.size(); c++) {//loop through the size of the array
                if (c % 2 == 0) {//gets element at array index that is even
                    iparray[c] = listofstring.get(c);//gets the ip
                    //System.out.println("elements stringsq:" + listofstring.get(c));
                } else {
                    //
                }

            }
           //close connection
            con.close();
            String csv = "infile.csv";//initilia
            syInfo.initCSV(csv);//calls initcsv function
            try ( // boolean alreadyExist = new File(csv).exists();
                    // CSVWriter writer = new CSVWriter(new FileWriter(csv));
                    CSVWriter writer = new CSVWriter(new FileWriter(csv, true), '\n') //append to
file
                    ) {

                writer.writeNext(str);//change to ip
                writer.flush();
            }
            syInfo.getUStop();
            for (int a = 0; a < iparray.length; a++) {
                if (iparray[a] != null) {
```

```
                          setStatus("Disabled", iparray[a].toString());//set status function
                        }
                    }
                    data();
                }
            }
        } catch (SQLException ex) {
Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
    Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

4.5.10 Posting Connected Computers' Details to the Database

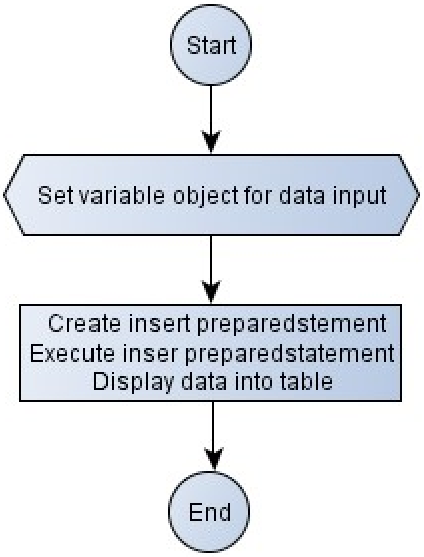
This section majorly deals with the module that inserts details into the database, then loads them onto the condevices table for storage management and later be collected for display. The algorithm, corresponding flow chart (Figure 54) and code is as follows:

1. Start
2. Set the variable object for data input
3. Create insert prepared statement
4. Execute a prepared statement
5. Get data into the table
6. End

The flowchart

Figure 54

Posting and Displaying connected computer’s details chart


Start
Set variable object for data input
Create insert preparedstement
Execute inser preparedstatement
Display data into table
End

The Code

```
String sqlins = "INSERT INTO condevices
(IpAddress,Computer_Name,Serial_Number)values('" + ipoutput + "','" + Computer_Name +
"','" + serial + "')";
                pst = con.prepareStatement(sqlins);
                pst.execute();
load them in the table
                String sql1 = "SELECT IpAddress,Computer_Name,Serial_Number FROM
condevices";
                PreparedStatement pst1 = con.prepareStatement(sql1);
                ResultSet rs1 = pst1.executeQuery();
TblConnectedDevices.setModel(DbUtils.resultSetToTableModel(rs1));
```

The stored details for connected devices need to be extracted and displayed on an interface for the user to know which devices are connected. The design and implementation of the display interface is presented next.

4.5.11 Retrieving and Displaying the Connected Computers Details on Interface

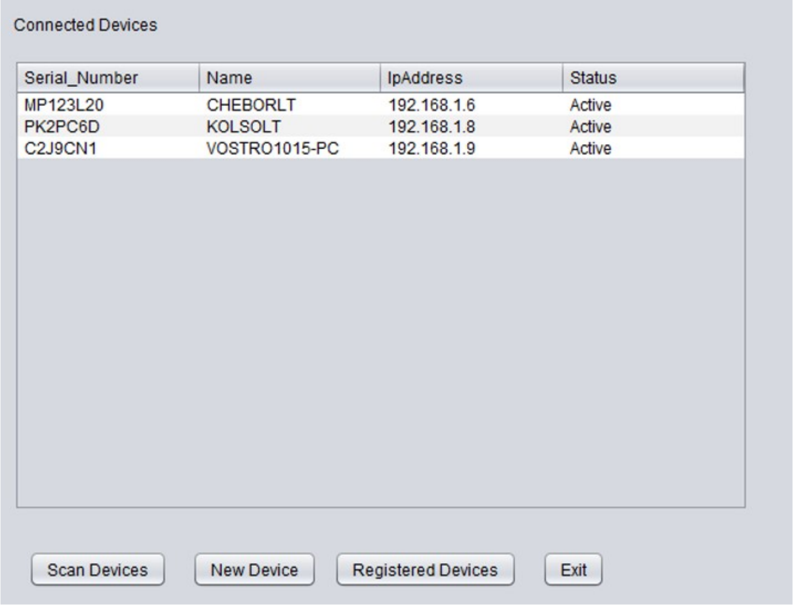
The centre bolt of the prototype was to have the valid allowed identification details displayed on an interface. It is from this display interface that computers can be managed by further disallowing them from being the network if a need arises. An administrator, for instance can decide, basing on some regulations, to disable a particular computer that was earlier allowed network access, from using the network services and resources.

Armed with four command buttons, Scan Devices, New Device, Registered Devices and Exit, the Connected Devices interface apart from displaying validly allowed computers, equally allows manipulation of the computers via the interface button. Scan Devices button runs through the program to displays connected computers once clicked. While Registered Devices button displays the list of registered computers, New Device Button links to AddDevice interface that allows new computers to be registered. Figure 55 below shows various detail displays using three different APs.

Figure 55

Valid computer details display using various APs

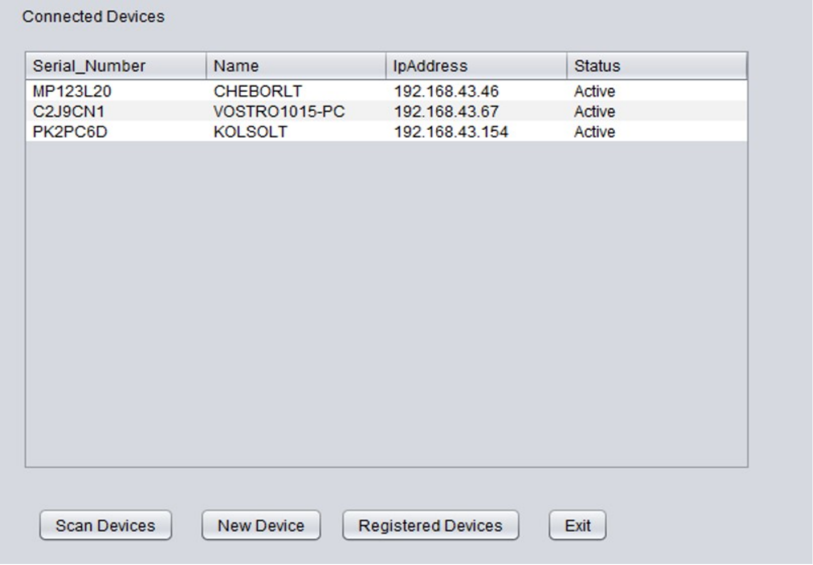
i. *Using an AP with an IP address 192.168.1.0*



Connected Devices

Serial_Number	Name	IpAddress	Status
MP123L20	CHEBORLT	192.168.1.6	Active
PK2PC6D	KOLSOLT	192.168.1.8	Active
C2J9CN1	VOSTRO1015-PC	192.168.1.9	Active

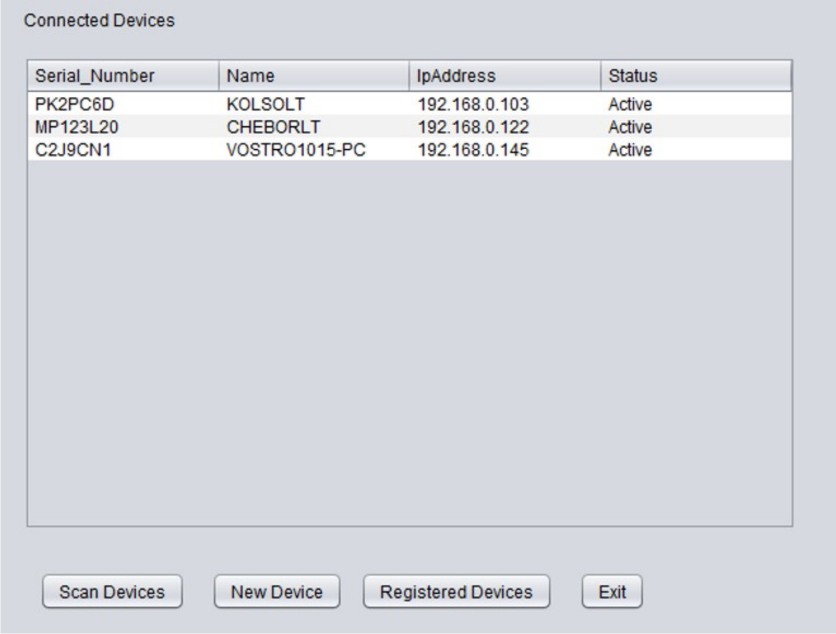
ii. *Using an AP with an IP address 192.168.43.1*



Connected Devices

Serial_Number	Name	IpAddress	Status
MP123L20	CHEBORLT	192.168.43.46	Active
C2J9CN1	VOSTRO1015-PC	192.168.43.67	Active
PK2PC6D	KOLSOLT	192.168.43.154	Active

iii. Using an AP with an IP address 192.168.0.1



Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.0.103	Active
MP123L20	CHEBORLT	192.168.0.122	Active
C2J9CN1	VOSTRO1015-PC	192.168.0.145	Active

The three displays from the three APs for the same computers gives an indication that IP addresses can change but computer serial numbers do not change. This way, a computers serial serves well as a networked computer identifier.

To display the identification details, this module first gets the identification details from the computers relation, then loops through them one by one and finally displays them in Connected Devices interface. The algorithm, flow chart and code used to generate the Connected Devices and display identification details was as in Figure 56 below

The algorithm:

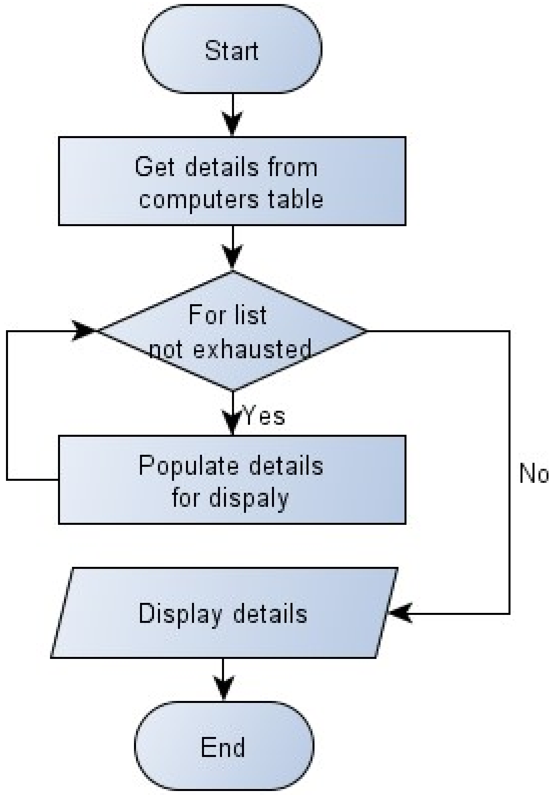
```
1.  Start
2.  Get details from computers table
3.  For list not exhausted
            Populate details for display
4.  Display details
```

5. End

The flowchart

Figure 56

Valid computer details display


Start
Get details from computers table
For list not exhausted
Yes
Populate details for dispaly
No
Display details
End

The code

```
public void data() {
        try {
            String sql = "SELECT Serial_Number,Name,IpAddress,Status FROM computers";
            pst = con.prepareStatement(sql);
            rs = pst.executeQuery();
tblDevice.setModel(DbUtils.resultSetToTableModel(rs));
            /**
             * gets the data from the database.computers
             *
```

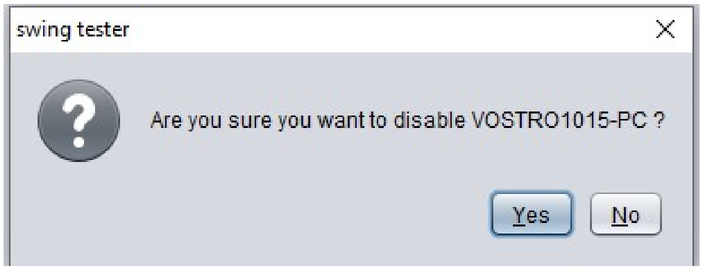
```
         */
        List<String> numdata = new ArrayList<String>();
        for (int count = 0; count < tblRegistered.getRowCount(); count++) {
            numdata.add(tblRegistered.getValueAt(count, 0).toString());
        }
        System.out.println(numdata);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

4.5.12 Denying a Computer Network Access

Apart from registering, validating and displaying valid computer authentication details, the SNAM model can as well be used to control valid computers by allowing them continue or are blocked from using the network. Blocking an already logged valid computer is the prerogative of a network administrator depending on their predetermined rules. If the administrator, for instance wants to block the computer with serial number C2J9CN1 name VOSTRO1015-PC, out of the network because of one reason or another, the administrator clicks on the computer row on the Connected Devices interface as in figure 55(i) above. The resulting feedback is as in the Figure 57 below

Figure 57

Disabling a computer prompt message



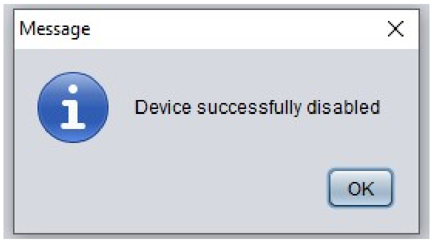
swing tester

Are you sure you want to disable VOSTRO1015-PC ?

Yes button is then clicked to block the computer, or else, the process is aborted if No button is clicked. The success message is the displayed as in the Figure 58 below for a successfully blocked computer.

Figure 58

Disabling a computer confirmation message

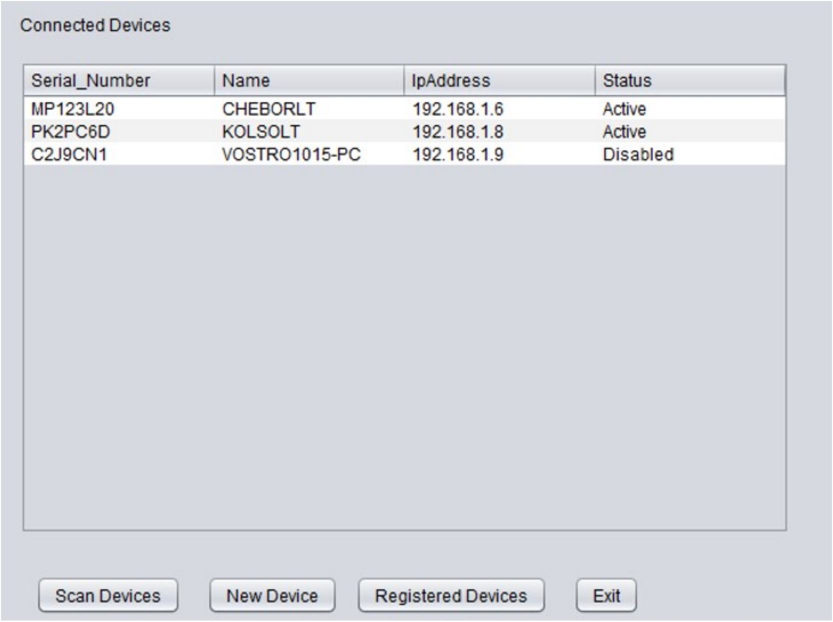


Device successfully disabled

Afterwhich the computer is blocked from accessing the network services and resources as indicated in the Figure 59 below

Figure 59

A disabled VOSTRO1015-PC computer



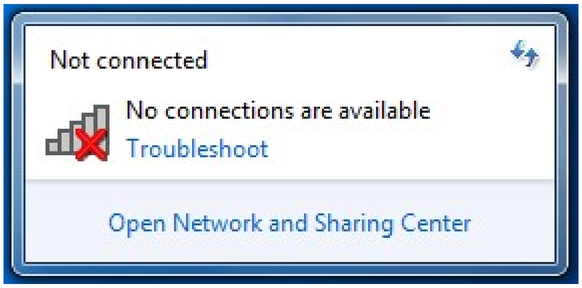
Connected Devices

Serial_Number	Name	IpAddress	Status
MP123L20	CHEBORLT	192.168.1.6	Active
PK2PC6D	KOLSOLT	192.168.1.8	Active
C2J9CN1	VOSTRO1015-PC	192.168.1.9	Disabled

Trying to access the network at this point from the computers end is futile as indicated in the disabled computers network interface as shown in the Figure 60 below

Figure 60

A blocked disabled computer network access interface



Not connected

No connections are available

The algorithm, flowchart (Figure 61) and code below made it possible for the module to block a valid computer from accessing a network

The algorithm

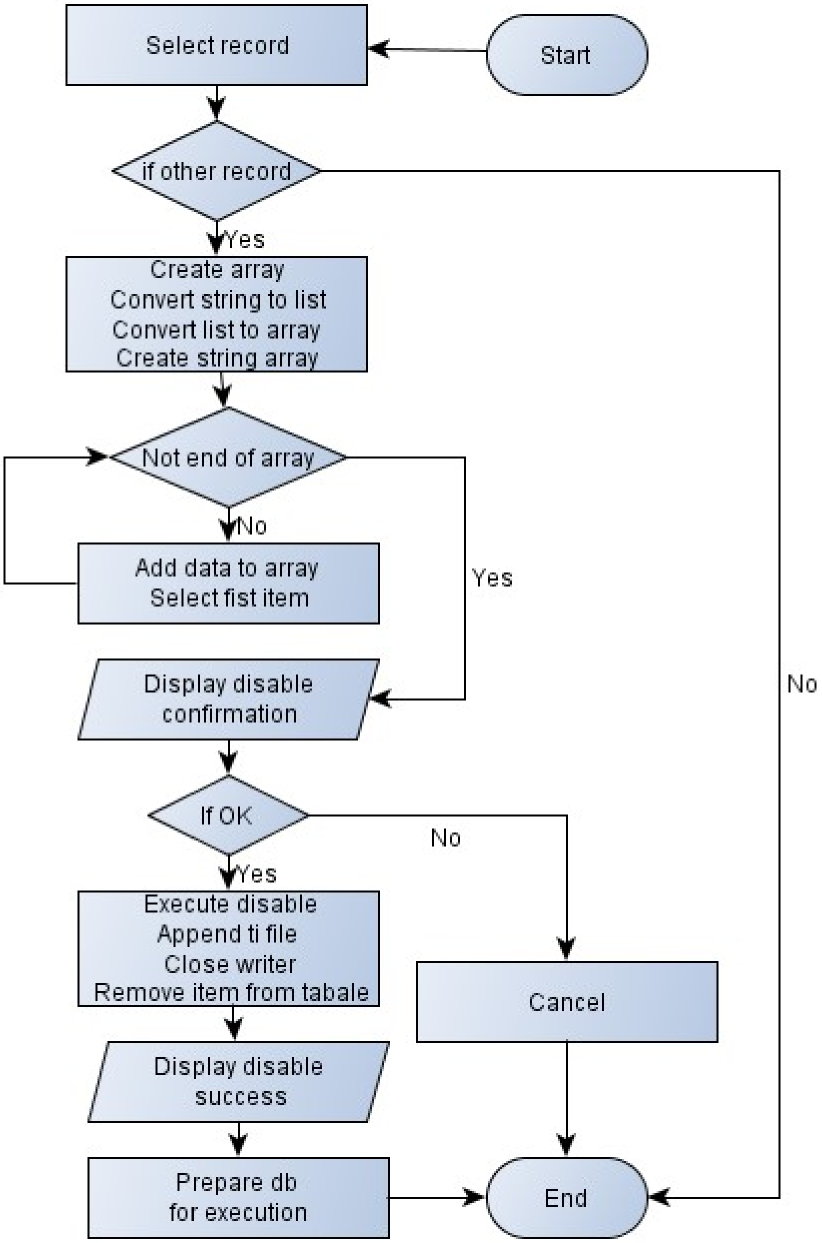
```
1.  Start
2.  Select record
3.  If other record is available
    3.1   Create record array
    3.2   Convert string to list
    3.3   Convert list to array
    3.4   Create string array
    4.    For not at the end of array
    4.1   Add record data to array
    4.2   Select first item in the array
    4.3   Display disable confirmation message
5.  If Okay
```

```
      5.1     Execute disable function
      5.2     Append to file
      5.3     Close the writer
      5.4     Remove computer from the table
      5.5     Display disable success message
      5.6     Prepare database for execution
6.  Else, cancel
7.  End if
8.  End if
9.  End
```

The flowchart

Figure 61

Disabling a computer flowchart



The code

```
private void tblDeviceMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
```

```
        try {//on click get selected row
            row = tblDevice.getSelectedRow();
            String table_click = (tblDevice.getModel().getValueAt(row, 0).toString());
            String sql = "SELECT * FROM computers WHERE Serial_Number='" + table_click +
"'";
            pst = con.prepareStatement(sql);
            rs = pst.executeQuery();
            if (rs.next()) {//is another rcord available
                String name = rs.getString("Name");
                String serial = rs.getString("Serial_Number");
                String ipout = rs.getString("IpAddress");
                String[] ipoutElements = ipout.split(",");//creates an array and splits from the comma
                List<String> fixedLengthList = Arrays.asList(ipoutElements);converts string to a List
                ArrayList<String> listofstring = new ArrayList<String>(fixedLengthList);converts
List to ArrayList
                String iparray[] = new String[listofstring.size()];creates a string array
                for (int i = 0; i < listofstring.size(); i++) {//loop through the arraylist and populate the
string array
                    iparray[i] = listofstring.get(i)//adding data to array
                }
                String[] str = {iparray[0], null};//select first element in array
                System.out.println("elements array:" + str[0]);
                        //this is used to show a confirmation dialog for disabling device
                int result = JOptionPane.showConfirmDialog(this, "Are you sure you want to disable "
+ name + " ?", "swing tester",
                        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
                if (result == JOptionPane.YES_OPTION) {
                            String csv = "stopOne.csv";
                    syInfo.initCSV(csv);//disables the selected computer
                    try (
              CSVWriter writer = new CSVWriter(new FileWriter(csv, true), '\n') //append to
file
              ) {
            writer.writeNext(str);//change to ip
            writer.flush();
            //close the writer
           }
          syInfo.getStopOne();
          //next remove disabled computer from the table
          JOptionPane.showMessageDialog(this, "Device successfully disabled");
          String st = "Disabled";
          con = ConnectDB.connect();//create a connection
          String sql2 = "UPDATE computers SET status=? WHERE  Name=?";
          pst = con.prepareStatement(sql2);//prepare sql for execution
          pst.setString(1, st);
          pst.setString(2, name);
          pst.executeUpdate();
          data();
        } else if (result == JOptionPane.NO_OPTION) {
          JOptionPane.showMessageDialog(this, "cancel");
        } else {
          //none selected
      }
    } catch (Exception e) {
      JOptionPane.showMessageDialog(null, e);
    }
  }
```

4.5.13 System Modules Execution Report Summary

The Table 5 below shows the summarized report on the status of system modules execution based on registration of computers, collection of a computers authentication details, allowing or denying a computers network access based on whether they are registered or not, and allowing a valid computer to either continue using the network or be denied network resource usage due one or another reason.

Table 5

System modules execution report

Functions	**Description**	**Execution Status (Yes/No)**	**Remarks**
New Computer registration	Check if a new computer can be registered	Yes	Are confirmed in the Registered Devices interface
Edit a Computer	Check if a registered computer can be deleted	Yes	Are confirmed in the Registered Devices interface
Collect Computer's Serial Number, IP address, Name	Check if authentication details can be collected from the system	Yes	Collected identification details are displayed on the Connected Devices interface
Collect Computer's authentication details with different access points (APs)	Check if authentication details can be collected from the system using different APs	Yes	Collected authentication details are displayed on the Connected Devices interface for three different APs
Allow a registered computer network access	Check if a registered computer is allowed network access	Yes	Registered computers are displayed in the Connected Devices interface
Deny an unregistered computer network access	Check if an unregistered computer is denied network access	Yes	Unregistered computers are denied network access as indicated in the affected computers network access status interface
Disable allowed computer	Check if an allowed computer can be disabled	Yes	This is confirmed by the disabled status of the computer in the Connected Devices interface and in the affected computers network access status interface

4.6 SNAM Model Evaluation

For validation purposes, proof of concept method on the cumulative essential functionalities of the prototype was evaluated using test runs as proposed by Diceus, (2020) based on goal-based evaluation method (Cronholm & Goldkuhl, 2003). With the focus of being on intended services and outcomes of a program, goal-based evaluation measures the extent to which a program attains its intended goals. It should be measurable and meet the requirements specification. Three selected stakeholders, that is, a network administrator, a Wi-Fi proprietor and a network expert were selected to perform the evaluations.

The prototype was fundamentally geared towards the ability to

1. Collect a computer's authentication details (that is name, IP address and Serial number)
2. Allow a registered computer access to a network
3. Deny an unregistered computer access to a network
4. Control an already logged on computer to a network if need arises

A set of three test runs were carried out using static group comparison pre-experimental design. Just as in the initial experimentation set up, each set had three computers and an access point. One of the computers, that was configured as server, was installed with the serial number authentication application. For comparisons purposes, the other two computers were configured as clients, one for valid and the other for invalid expected outcomes.

Each set was carried out independently by a wireless LAN stakeholder (that is, a network administrator, a Wi-Fi owner and an IT expert). Each participant was given instruction on how to run the system against a check list (Appendix VI, Appendix VII, Appendix IX) of the four test runs. They were expected to observe the behavior when the system is run and manipulated according to the fundamental questions that corresponds to the system functionalities and answer yes or no on the check list form given to them. The questions on the checklist were as follows;

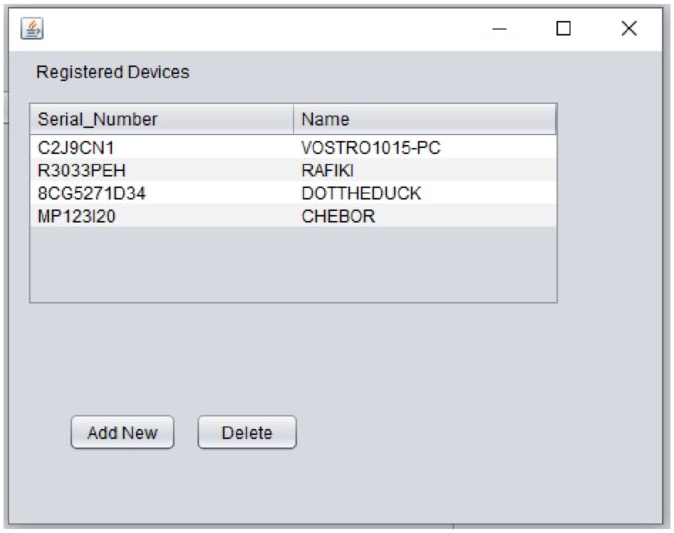
1. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?
2. Can the system allow a registered computer access to a network?
3. Can the system deny an unregistered computer access to a network?
4. Can the system deny an already logged on computer to a network if need arises?

4.6.1 Evaluator 1 Results

The evaluator ran the SNAM application on a setup of a server, two laptop clients and an access point. The application had a list of already registered computers (Figure 62) prior to running the program. Out of the three connected computers; the server (DOTTHEDUCK) and two other clients KOLSOLT and RAFIKI, KOLSOLT client had not been registered at the time of set up execution.

Figure 62

Computers that are already registered in the system



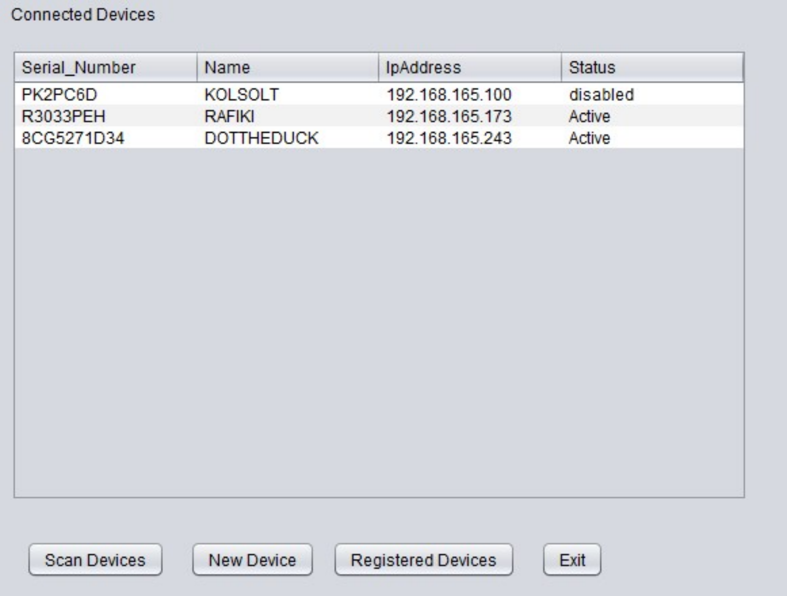
Registered Devices

Serial_Number	Name
C2J9CN1	VOSTRO1015-PC
R3033PEH	RAFIKI
8CG5271D34	DOTTHEDUCK
MP123I20	CHEBOR

As a result, KOLSOLT client was not allowed access to the network (Figure 63 and Figure 64). In contrast, RAFIKI client that was earlier registered, was allowed access to the network as shown in Figure 63, a clear indication that the prototype allows registered computers access to a network while unregistered computers are denied access to a network and its resources.

Figure 63

Blocked unregistered KOLSOLT computer

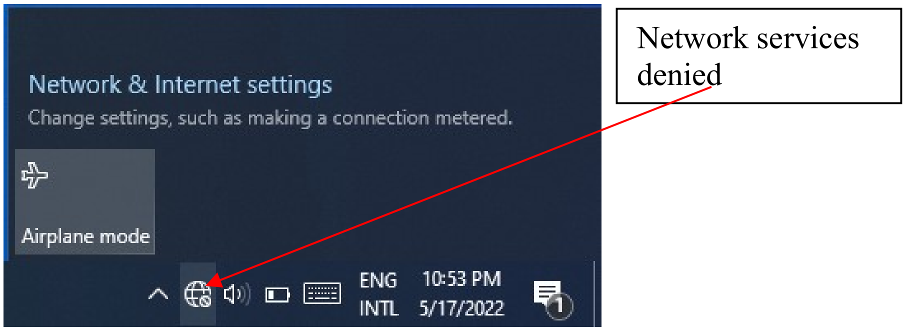


Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	disabled
R3033PEH	RAFIKI	192.168.165.173	Active
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

Figure 64

Blocked unregistered KOLSOLT computer Wi-Fi interface status



Network & Internet settings

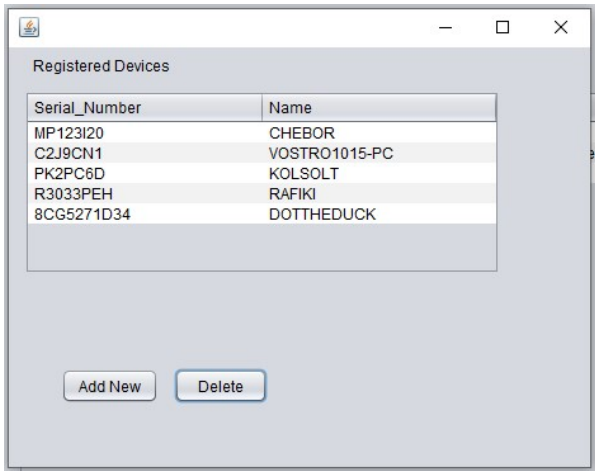
Change settings, such as making a connection metered.

Network services denied

KOLSOLT was then registered (Figure 65) and the application executed once again. This time, all the three connected computers were displayed as illustrated in Figure 65

Figure 65

Registered computers including earlier unregistered KOLSOLT



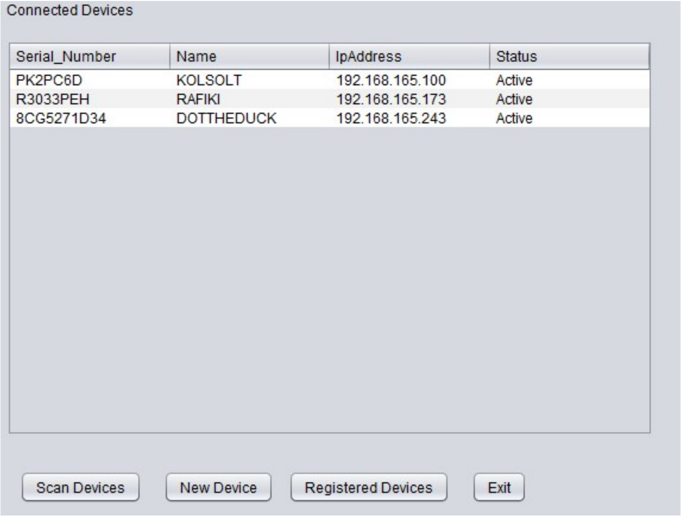
Registered Devices

Serial_Number	Name
MP123I20	CHEBOR
C2J9CN1	VOSTRO1015-PC
PK2PC6D	KOLSOLT
R3033PEH	RAFIKI
8CG5271D34	DOTTHEDUCK

The evaluator as well tested on whether or not, the system can deny network access to an already logged on computer in case a need arises. This was achieved by first clicking on the desired client (RAFIKI client computer in this case) on the Connected Devices interface on Figure 66 below.

Figure 66

KOLSOLT computer allowed access after registration



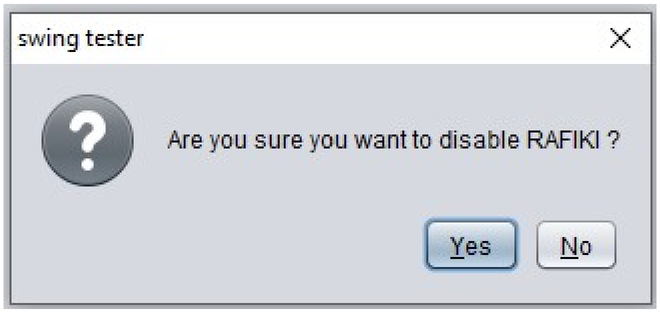
Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	Active
R3033PEH	RAFIKI	192.168.165.173	Active
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

The system prompts on the surety to disable the client from accessing the network Figure 67

Figure 67

Denying RAFIKI computer network access prompt message

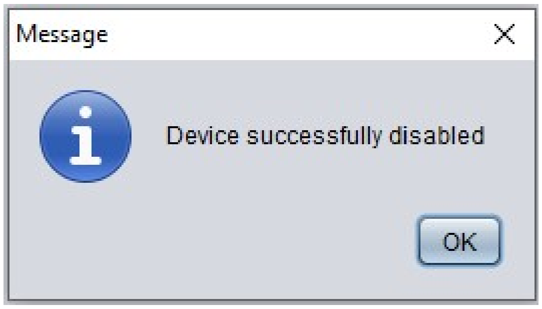


Are you sure you want to disable RAFIKI ?

On clicking YES button, the RAFIKI client is successfully denied network access as illustrated in system prompt in Figure 68 below.

Figure 68

Denying RAFIKI computer network access confirmation message

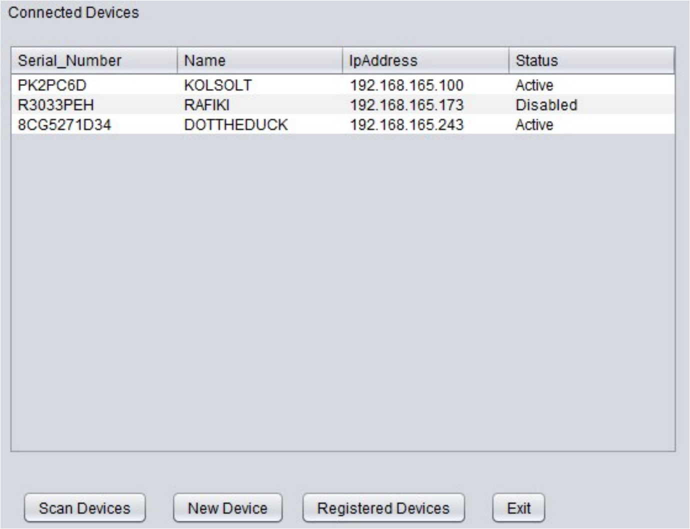


Device successfully disabled

The confirmation that RAFIKI client has been disabled, therefore, cannot access the network is illustrated in a screen shot in Figure 69 below

Figure 69

Denied RAFIKI computer network access



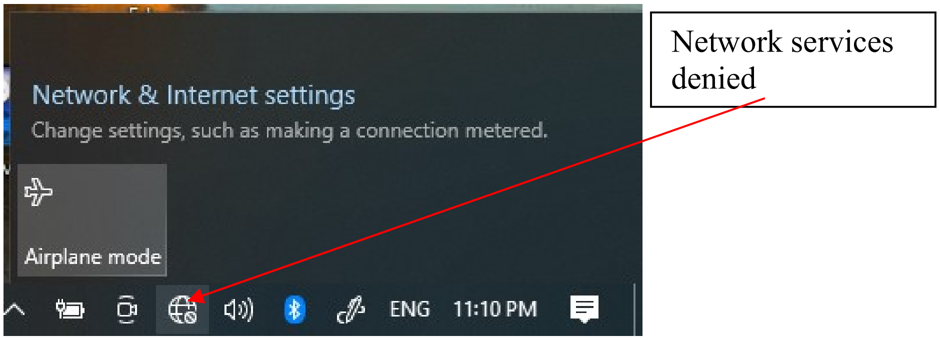
Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	Active
R3033PEH	RAFIKI	192.168.165.173	Disabled
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

The client Wi-Fi interface status as well indicates that the client cannot access the network as shown in the Figure 70 below

Figure 70

Blocked network denied KOLSOLT computer Wi-Fi interface status



Network services denied

Network & Internet settings
Change settings, such as making a connection metered.
Airplane mode
ENG
11:10 PM

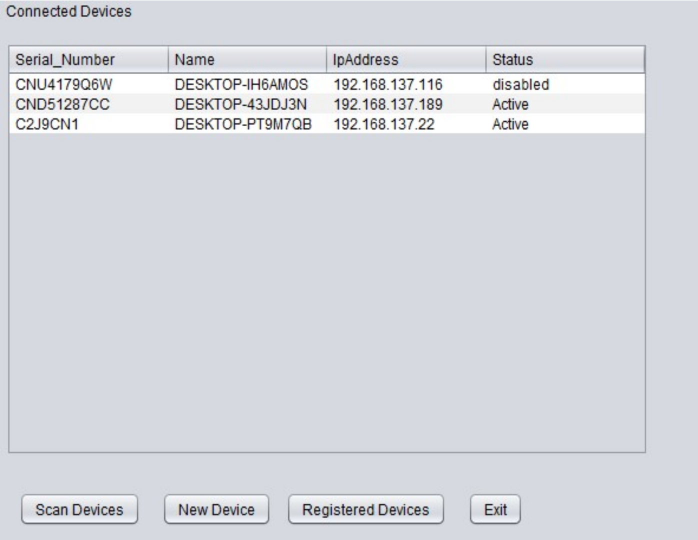
4.6.2 Evaluator 2 Results

A second evaluator using an access point with IP address 192.168.137.120, a server DESKTOP-PT9M7QB, a registered client DESKTOP-43JDJ3N and unregistered client DESKTOP-IH6AMOS repeated the evaluation process. On running the SNAM prototype, DESKTOP-

IH6AMOS client was denied access to the network as it was not one of the registered computers as illustrated in Figure 71.

Figure 71

Blocked unregistered DESKTOP-IH6AMOS computer



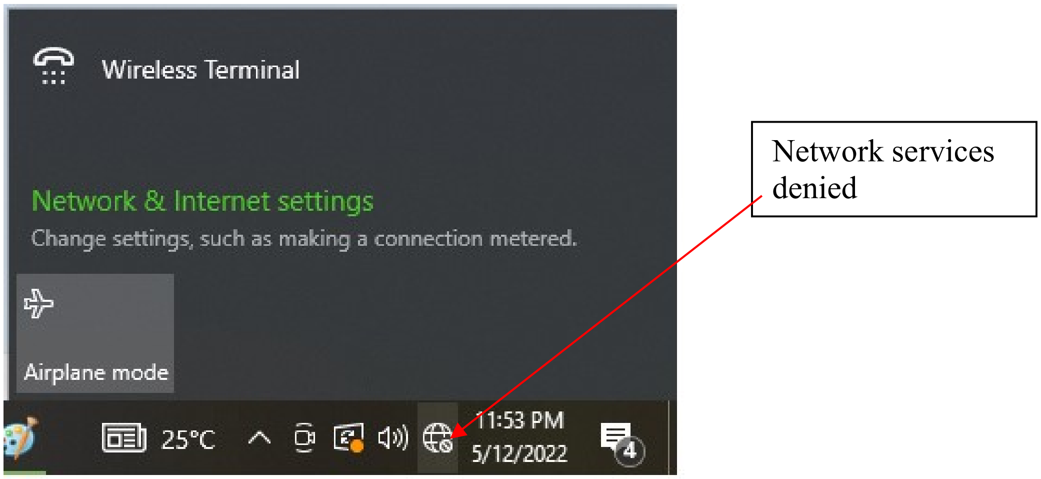
Connected Devices

Serial_Number	Name	IpAddress	Status
CNU4179Q6W	DESKTOP-IH6AMOS	192.168.137.116	disabled
CND51287CC	DESKTOP-43JDJ3N	192.168.137.189	Active
C2J9CN1	DESKTOP-PT9M7QB	192.168.137.22	Active

Further confirmation of **DESKTOP-IH6AMOS** client being denied access to the network is demonstrated by the client Wi-Fi status being inactive as shown in the Figure 72 below.

Figure 72

Blocked unregistered DESKTOP-IH6AMOS computer Wi-Fi interface



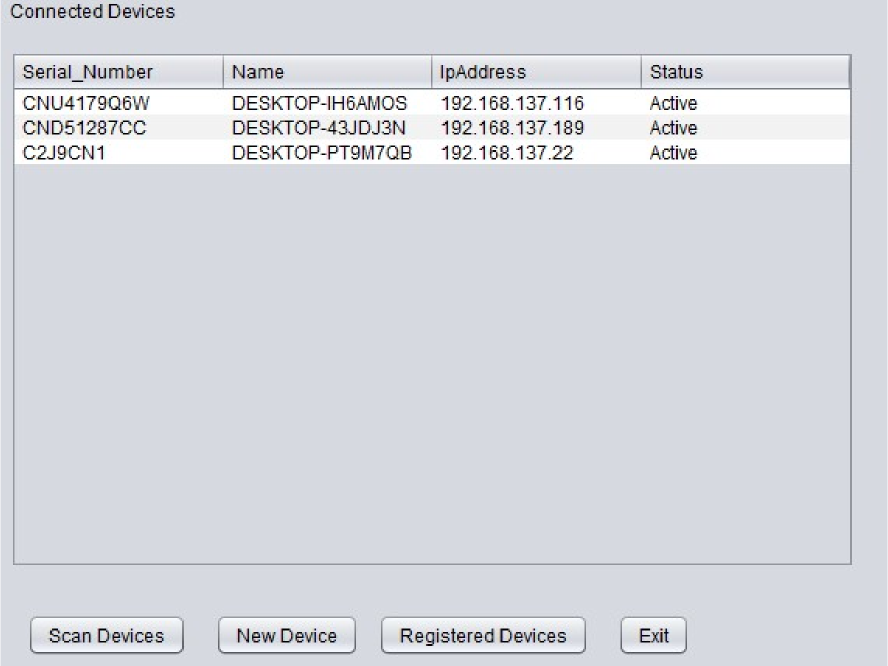
Wireless Terminal
Network & Internet settings
Change settings, such as making a connection metered.
Airplane mode
25°C
11:53 PM
5/12/2022
4

Network services denied

DESKTOP-IH6AMOS client was registered the application run once again. The result in the Figure 73 below indicated the DESKTOP-IH6AMOS client was eventually allowed access to the network, of course, after registration.

Figure 73

DESKTOP-IH6AMOS allowed access after registration

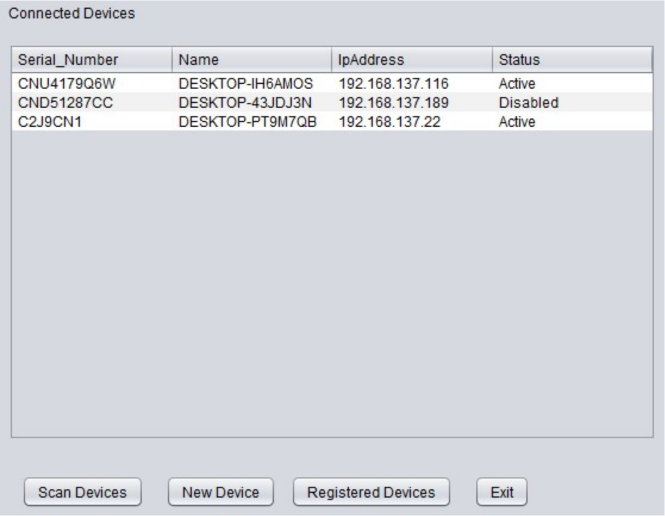


Serial_Number	Name	IpAddress	Status
CNU4179Q6W	DESKTOP-IH6AMOS	192.168.137.116	Active
CND51287CC	DESKTOP-43JDJ3N	192.168.137.189	Active
C2J9CN1	DESKTOP-PT9M7QB	192.168.137.22	Active

Just as in the case for the first evaluator, the second evaluator as well tested on whether or not, the system can deny network access to an already logged on computer in case a need arises. This was achieved by first clicking on the target client (that is DESKTOP-43JDJ3N client computer in this case) on the Connected Devices interface on Figure 73 above. On confirmation and accepting the process on denying DESKTOP-43JDJ3N network access, just as was done by the first evaluator, the client is denied network access as shown in screen shot Figure 74 below.

Figure 74

Denied DESKTOP-IH6AMOS computer network access



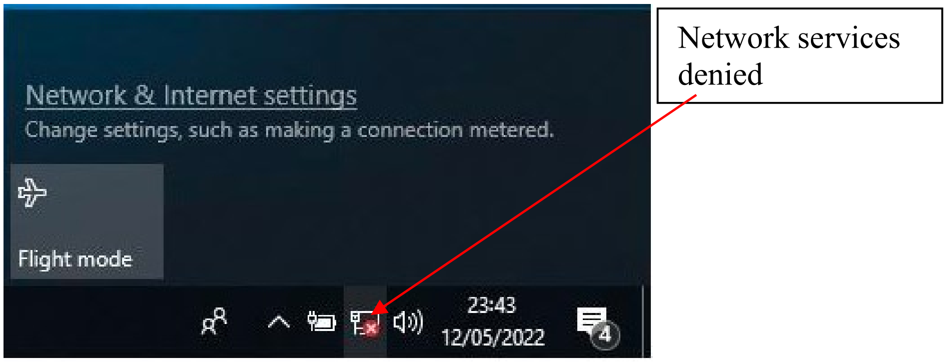
Connected Devices

Serial_Number	Name	IpAddress	Status
CNU4179Q6W	DESKTOP-IH6AMOS	192.168.137.116	Active
CND51287CC	DESKTOP-43JDJ3N	192.168.137.189	Disabled
C2J9CN1	DESKTOP-PT9M7QB	192.168.137.22	Active

And confirmed by the client Wi-Fi interface status as well, that indicated that the client cannot access the network as shown in the Figure 75 below.

Figure 75

Denied DESKTOP-IH6AMOS computer Wi-Fi interface status



Network & Internet settings

Change settings, such as making a connection metered.

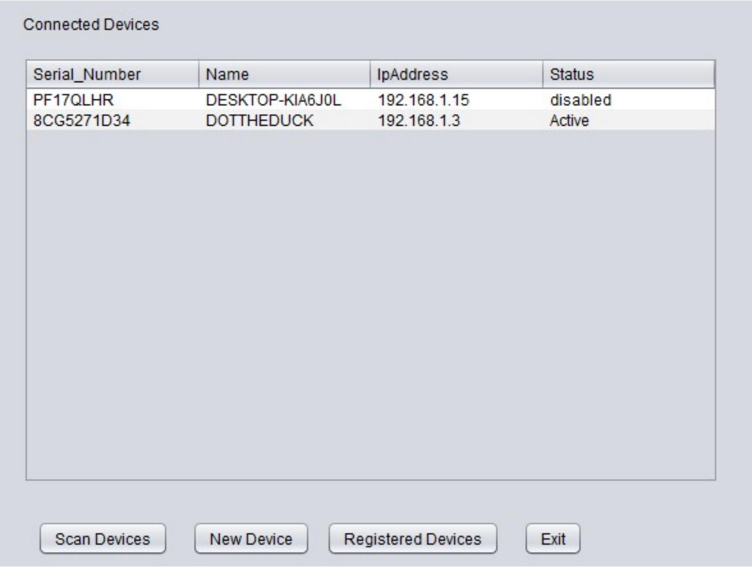
Network services denied

4.6.3 Evaluator 3 Results

Similarly, a third evaluator using an access point with IP address 192.168.1.1, a server VICK-PC, a registered client DOTTHEDUCK and unregistered client DESKTOP-KIA6J0L repeated the evaluation process once again. On running the SNAM prototype, DESKTOP-KIA6J0L client was denied access to the network as it was not one of the registered computers as illustrated in Figure 76 below.

Figure 76

Blocked unregistered DESKTOP-KIA6J0L computer



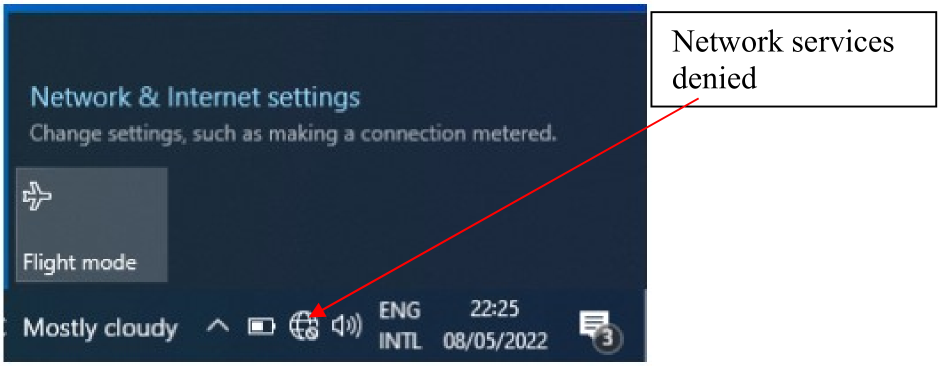
Connected Devices

Serial_Number	Name	IpAddress	Status
PF17QLHR	DESKTOP-KIA6J0L	192.168.1.15	disabled
8CG5271D34	DOTTHEDUCK	192.168.1.3	Active

A confirmation of denying the unregistered DESKTOP-KIA6J0L computer was further illustrated in disabled Wi-Fi status as shown in the Figure 77 below

Figure 77

Blocked unregistered DESKTOP-KIA6J0L computer Wi-Fi interface status



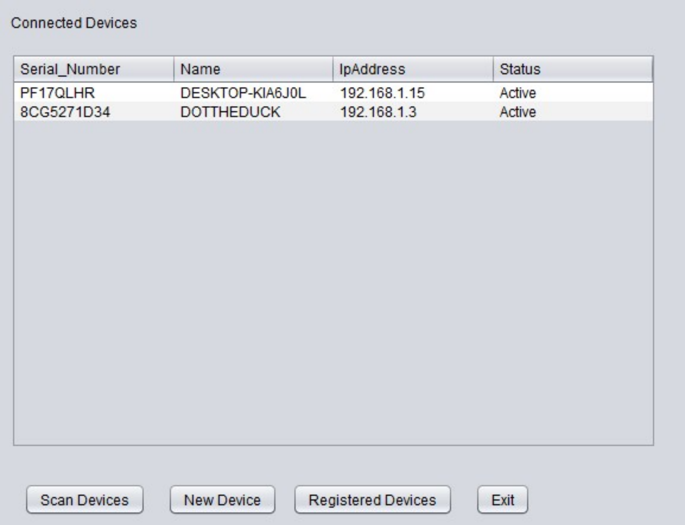
Network & Internet settings

Change settings, such as making a connection metered.

But on registration of the unregistered DESKTOP-KIA6J0L client, the computer was allowed access as illustrated in the Figure 78 below, on system execution.

Figure 78

DESKTOP-KIA6J0L client allowed network access after registration



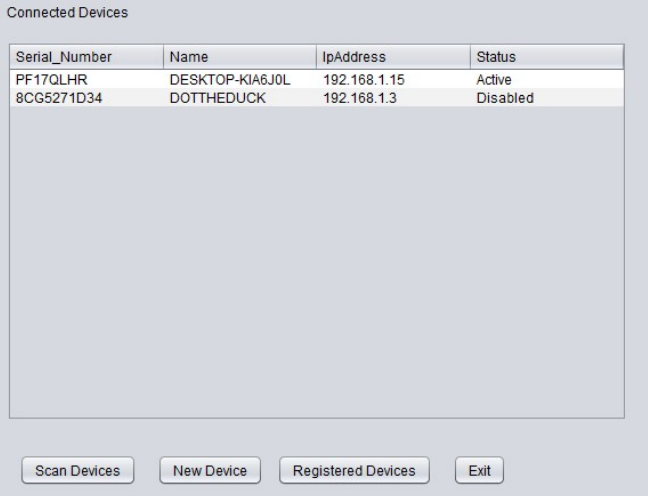
Connected Devices

Serial_Number	Name	IpAddress	Status
PF17QLHR	DESKTOP-KIA6J0L	192.168.1.15	Active
8CG5271D34	DOTTHEDUCK	192.168.1.3	Active

In the same way as in the case for the first two evaluators, the third evaluator as well tested on whether or not, the system can deny network access to an already logged on computer in case a need arises. This was achieved by first clicking on the target client (that is DOTTHEDUCK client computer in this case) on the Connected Devices interface on Figure 67 above. On confirmation and accepting the process on denying DESKTOP-43JDJ3N network access, the client was denied network access as shown in screen shot Figure 79 below.

Figure 79

DOTTHEDUCK client denied network access



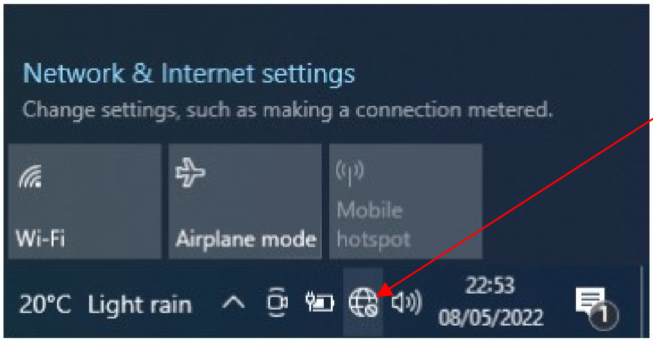
Connected Devices

Serial_Number	Name	IpAddress	Status
PF17QLHR	DESKTOP-KIA6J0L	192.168.1.15	Active
8CG5271D34	DOTTHEDUCK	192.168.1.3	Disabled

Denial of network access to DOTTHEDUCK client was as well confirmed by the client Wi-Fi interface status that indicated that the client cannot access the network as shown in the Figure 80 below

Figure 80

Denied DOTTHEDUCK computer Wi-Fi interface status



Network & Internet settings

Change settings, such as making a connection metered.

Network services denied

4.6.4 SNAM Evaluators Report Summary

The Table 6 below shows the three evaluators summarized report on the status of system modules execution based on collection of a computers authentication details, allowing or denying a computers network access based on whether they registered or not, and allowing a valid computer to either continue using the network or be denied network resource usage due one or another reason. The overall objective for the evaluation was to test run the prototype in order to proof that the prototype can collect a logged on computer's authentication details (that is name, IP address and Serial number, that it can allow a registered computer access to a network, that it can deny an unregistered computer access to a network and that it can deny an already logged on computer to a network if need arises. These objectives then culminate to the aim of the study that the prototype can use a serial number to authenticate a computer in a wireless LAN. The expected outcome was therefore either a yes or no. The summary of the results from the three evaluators as indicated in appendix were summarized as in the Table 6 below.

Table 6

Evaluators system modules execution report

#	Evaluation	Evaluator 1	Evaluator 2	Evaluator 3
			Yes/No	
1.	Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?	Yes	Yes	Yes
2.	Can the system allow a registered computer access to a network?	Yes	Yes	Yes
3.	Can the system deny an unregistered computer access to a network?	Yes	Yes	Yes
4.	Can the system deny an already logged on computer to a network if need arises?	Yes	Yes	Yes

The summary, as a way of confirmation, concluded that the SNAM prototype can collect a logged on computer's authentication details (that is name, IP address and Serial number), can allow a registered computer access to a network, can deny an unregistered computer access to a network and that it can deny an already logged on computer to a network if need arises. In turn, a serial number can therefore, be used to authenticate a computer in a wireless LAN

4.7 Chapter Summary

This chapter of the study presented the findings and results on the design, development, demonstration and evaluation of the serial number based authentication model (SNAM) used to authenticate computers in a wireless LAN. The model was designed and developed using dynamic programming algorithm, state and flow chart diagrams, then demonstrated using MySQL and Java's IDE tools over evolutionary prototyping on a static group comparison pre-experimental design set up test runs to proof the concept that a serial number can used to authenticate a computer in a wireless LAN. The model was then evaluated using goal based method, both by the researcher and selected stakeholders on a static group comparison pre-experimental design on similar set ups and test runs. It was found out that a computer's serial number can actually be used to authenticate a computer in a wireless LAN. The next chapter presents the study conclusions and recommendations.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter summarizes the study by presenting its conclusions from the key findings on design, and development, demonstration and evaluation of the SNAM prototype that collects and uses a computer's serial number to authenticate a computer in a wireless LAN. The chapter would not be complete without presenting recommendations, areas for further research that arose in the course of the study, as well as research contributions.

5.2 Conclusions

This section concludes on the questions the study sought to answer on the key findings, development and design, demonstration and evaluation of the prototype that uses a serial number to authenticate a computer in a wireless LAN.

5.2.1 Development of an Algorithm that can Collect and Use a Remote Computer's Serial Number

The ultimate goal result was to get registered computers' authentication details on an interface for manipulation and for further management if need be. Unregistered computers on the other hand, are denied access to a network. To actualize this ultimate functionality, several other modules were designed to recursively culminate to the overall goal. Before displaying and managing the authentication details on the authentication interface, computers that wish to use the network and its resources are first registered, pre-existing connected computers identification details are deleted to pave way for newly connected computers details, get connected computer IP address, name and serial details, collate computer IP address, name and serial number, allow registered computers access the network while denying network access to unregistered ones, post valid computers details to the database, retrieve and display them on the identification details interface modules were designed and implemented.

A dynamic programming design was therefore adopted during the algorithm design development phase for the study. This was informed from the literature review that a dynamic programming optimizes solutions on a step by step basis recursively to the whole. The results of one step solve the problem of another consecutive step recursively. This results in the implication that the decision taken in one step is dependent upon an immediate previous or later decision steps. As a result, dynamic programming is absolutely the right method for optimization of combinations of ordered interdependent solutions.

The question that the study aimed to answer was “how can an algorithm that can obtain and use a remote computer’s serial number in a wireless LAN be developed?” The answer from the study outcome showed that an algorithm that can obtain a remote computer’s serial number in a wireless LAN can be developed using dynamic programming algorithm design.

5.2.2 Design of a Model that can use a Computer’s Serial Number for Authentication

To better understand the system as well as ensure inclusion of all system components, SNAM model was put into preparation, data collection, data processing, data display, and storage subsections. Computer registration process, start and connect database process, and cleaning the database records process were placed under the preparation section. The details collection section ensures that the fundamental details for authentication (that is, IP address, computer name, and serial number) are factored in. Collation and validation of authentication details is done at the data processing part. The display section then is responsible for displaying authentication details as well as controlling valid computers if a need arises. Not all these will happen without a storage location that stores and manages the identification details.

To make the model complete, relationships between the processes and the sections were indicated using data flow arrows. A diagram that shows all the components and their relationships using a state chart diagram was used to model the system. Component parts of the system were equally illustrated using flowcharts based on their corresponding pseudocodes. The question then was "how can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be designed?" The conclusion to the question was that a model that uses the computer's serial number to authenticate the computer in a wireless LAN was designed using a state chart and flowchart diagrams

5.2.3 Demonstration of the Model that Uses a Computer's Serial Number for Authentication

The main aim of the study was to design a model that demonstrates how a serial number could be used to authenticate a device (computer) in a wireless LAN. After establishing the need of using a serial number rather than MAC address, as an alternative or rather an additional layer to the existing identifiers was established. Whereas a MAC address functions as an address or a locator of the device in a network, a serial number identifies the device in the network. A SNAM prototype had therefore, be developed to be used to demonstrate the fact that a serial number can actually be used as a device identifier. As a prototype, the model had to be simple and be able to prove the intended concept. This then called for the implementation of the model using an evolutionary prototyping on MySQL database and Java's IDE tools. From the test runs, it was proved that a serial number could actually be used as an identifier to identify hence authenticate a computer in a wireless LAN.

The answer then to the question of the study "how can a model that uses computer's serial number to authenticate the computer in a wireless LAN be demonstrated?" can be answered thus**:** a model that uses the computer's serial number to authenticate a computer in a wireless LAN can be demonstrated using a prototype on MySQL database and Java's NetBeans IDE tools.

5.2.4 Evaluation of the Model that Uses a Computer's Serial Number to Authenticate a Computer in a Wireless LAN

Apart from initial test runs carried out to test the essential functionalities of the model, other similar test runs were repeated by independent evaluators. Three stakeholders drawn from a network administrator, a Wi-Fi proprietor and a network expert were selected to perform the evaluations. With the aim of validating the SNAM model, the cumulative essential functionalities of the prototype were evaluated using test runs. Functionalities that were tested were whether the system can or cannot collect a logged on computer's authentication details (that is, computer's name, IP address and Serial number), can or cannot allow a registered computer access to a network, can or cannot deny an unregistered computer access to a network and can or cannot deny an already logged on computer to a network if need arises. The expected responses to such test runs were the binary values yes or no. Goal-based evaluation method was therefore adopted in evaluating the prototype. With the focus being on intended services and outcomes of the SNAM program, goal-based evaluation measures the extent to which the prototype attained its intended goals.

This then answered the research question "How can the model that uses the computer's serial number to authenticate the computer in a wireless LAN be evaluated?" Here, the model that uses the computer's serial number to authenticate the computer in a wireless LAN was evaluated using a goal-based evaluated method by independent evaluators.

5.3 Recommendations

The study recommends the following;

5.3.1 Automating Computer Registration

Being one of the recommendations given in this study, automation of computer registration process would ensure registration of a computers details without the intervention of the network administrator. This would relieve the network administrator the tedious work of manually keying in the registration details one by one for a number of computers thus allowing the administrator to concentrate on other duties.

This could be achieved by creating a onetime registration form such that first time logins would be required to register before accessing the network and network resources. The details can be kept in a server for future reference and in validating computers to allow or deny network access any other time a computer wants to access the network. This way, subsequent logins would not require registration but would just be authenticated once they log on to the network.

5.3.2 Improvement and Deployment of SNAM

The ultimate goal of the study was to demonstrate that a computer's serial number could be used for authentication. The prototype was therefore developed with an emphasis on test runs to prove this concept. No much regard was considered for the usability design and actual implementation of the system in an ideal environment

It is with this regard therefore that the system is recommended to undergo through GUI enhancement, client Wi-Fi interface status feedback interaction on why the client is unable to access the network (inability to access a network because of registration or controlled reasons), user acceptability testing process and eventually deployment of the system

5.3.3 Development of a System that Caters for other Network Device Identifiers

Laptops and computers in general, like any other products are serialized with serial numbers for product differentiation. Other wireless network devices such as tablets and smartphones use IMEIs rather serial numbers for identification. The study, therefore, focused on laptops to argue its case that a serial number can be used for identification. Development of an overall model or system that caters for the varieties of identifiers, whether it is a serial number or an IMEI or any other related identifier, is recommended

5.3.4 Integration of SNAM into an Access Point

The SNAM system for the sake of the study was implemented in a server. However, due to the functions and nature of servers, it is recommended that the SNAM be implemented in an AP. Servers or DNS ideally, are meant to resolve domain names and IP addresses. In addition, there could be a number of servers communicating through the same AP that might require the SNAM system. Access points (AP) on the other hand is an ideal home to SNAM. Apart from an AP being the entry point to all devices connecting to the network, it also acts as good security control point in what is commonly referred to as identification, authentication, authorization, and accounting (IAAA).

5.3.5 Standardization of Computer's Serial Numbers

From the literature review discussion of the study, it was concluded that a computer's serial number satisfied the uniqueness, universality, collectability, data dependent, security (availability, integrity, and confidentiality) and robustness characteristics. The only predicament to serial numbering is on the mnemonic characteristics as demonstrated in table 1 of the study.

It is common practice that attributes, especially an identifier are set to follow a standardized format or structure within the sphere of the object. To name but examples book's ISBN, internet document's DOI and internet locater URL, all have a uniform global structure. It is, therefore, recommended that a computer's serial number be standardized. One of the benefits of standardizing an identifier is on optimum memory allocation usage that results in a more efficient algorithm.

5.4 Areas for Further Research

Apart from using a computer's serial number for authentication and authorization to a network and its resources, a computer's serial number can be leveraged on accounting and tracking computers in a network. Further research can therefore be carried out on accounting in a network access and management as well as tracking trails of computers usage. This is so because the study demonstrated that unlike other identifiers, serial numbers can be collected from the computers, are unique, and cannot be spoofed therefore are reliable. The suggestions are further discussed in the next section below.

5.4.1 Using a Computer's Serial Number for Network Accountability

As mentioned earlier, serial number authentication system will not be complete without leveraging it on the accountability component of network services provision. Accountability or auditing traces the device actions right from identification, authentication, and authorization as well as track the activities performed. Through accountability, a specific action performed can be traced based on the device identity (serial number). Once again, a serial number, due to its reliability, can be used as a base for network resource usage accounting. This way, the systems can prove who or what a given action was performed (no-repudiation).

5.4.2 Using a Computer's Serial Number in Tracking a Network Device

Apart from a further research suggestion on using a computers serial number to trace network resource usage, a research can also be carried out on how the computers serial number can be used in tracking a computer in a network. This could be most applicable in such areas as supply chain management, a device in inventory management and computer forensics

5.5 Research Contributions

The SNAM model cannot be used as a replacement for other device authentication methods. Rather it can be used as an alternative method to MAC address in device authentication. It can actually be added to the existing MAC address at data link, IP address at the network, port number at the transport, and application-specific address at application layers of the OSI reference layer model of identification, specifically, at the physical layer

5.5.1 Alternative Identification Method

SNAM can go a long way in providing an alternative identification hence authentication method to MAC address at the physical OSI reference model layer. The fact that a serial number is coded on the system BIOS without a copy in system software, makes it hard to be spoofed. As a result, it makes it unique, robust and more secure than a MAC address, therefore, providing a better alternative device identifier or identification, hence authentication to network devices.

5.5.2 Additional Identification Layer

A notable contribution of the study to research is on the additional layer to existing network device identifiers. The identifiers identified earlier on in the literature review section are an IP address, MAC address, port number, and application specific address. Of course, the identifiers are implicit as they are primarily used to identify devices indirectly through their locations rather the actual device.

Due to its functionality in process communication identification, port number identifiers are the transport layer of the OSI reference model. IP address identifier, of course, is a network issue as far as OSI layer model is concerned because it identifies devices on the internet. MAC addresses, on the other hand, identify devices in a particular network, therefore placed at the data link layer of the same model.

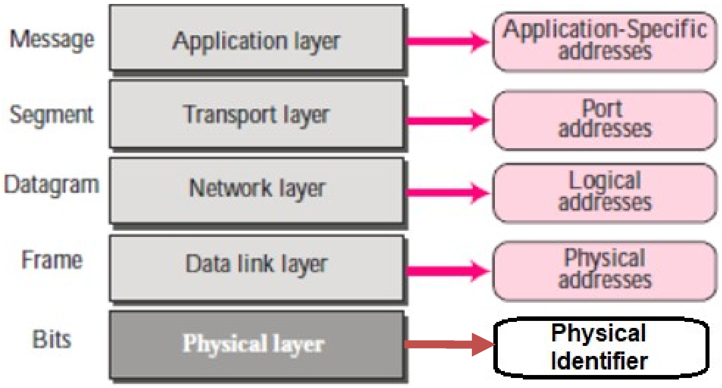
A serial number is hard-coded into the system board BIOS of a computer by the manufacturer of the product. Its association with pure hardware makes it be a physical layer issue in the OSI reference model. In this case, the serial number becomes an additional not only as a mere identifier to the existing port numbers, IP address, and MAC address identifiers but more so as an actual identifier to the product (computer).

In addition, modern computer networks are full of complexities because of internet exponential growth (Behringer, 2009). A particular concern is the tremendous growth of computer hardware capacities, software sizes and so their configurations. One way of simplifying this complexity is by layering these issues, despite the fact that layering comes with an overhead. This way, functions are devolved to each layer relieving other layers for their fundamental responsibilities that would have otherwise performed.

In this study, for instance, whereas the additional serial number identifies the actual product (computer), the MAC address is left to identify the network and IP address identifies the device in the internet, port numbers are left with the sole responsibility of identifying inter-process communication. This then results in a modification of the OSI reference model address layering in Figure 1 to modified Figure 81 shown below.

Figure 81

OSI reference model network address layering re-defined

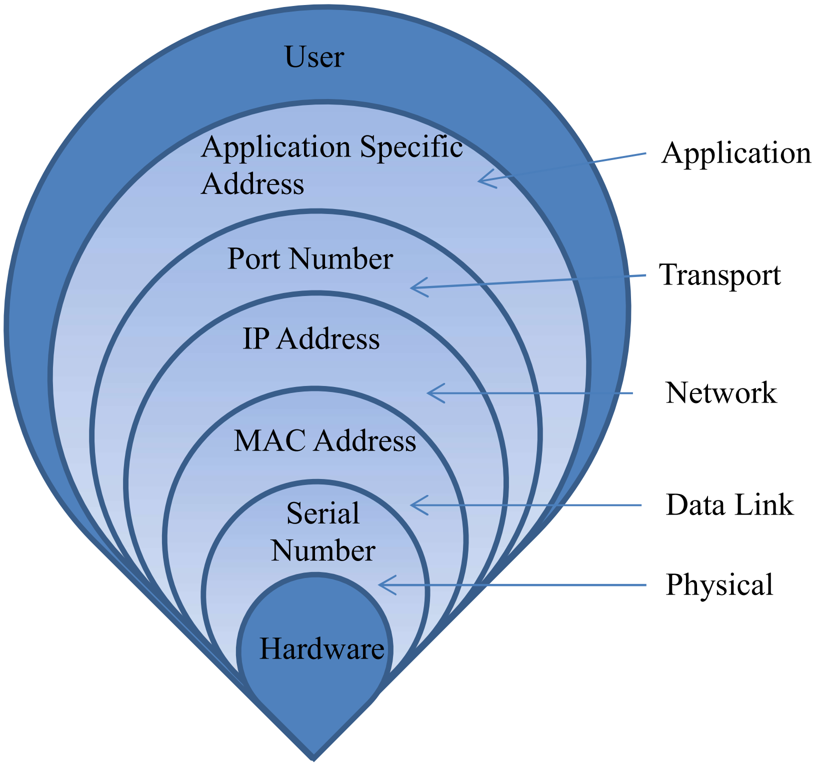


The layering of security issues is not something strange. Even in natural life scenarios, security is layered in such a way that the core business is protected from intruders or damage. The fragility of an egg, for instance, requires that apart from the shell, there are other shell layers namely the cuticle, inner membrane and the outer membrane for strength and protection (Solomon, 2010).

Because of serial number identifier addition to the existing identifiers, computer device identifier layering becomes more compact, stronger and more protected as illustrated in the Figure 82 below.

Figure 82

Computer device identifiers layering



Here, application specific address, port number, IP address and MAC address at application, transport network and data layer OSI reference model layers respectively, identify network devices based on their locations. A computers serial number, at the physical OSI reference model layer, becomes a network device identifier that actually identifies the device, not by location.

REFERENCES

ANSI/NISO. (2013, March 26). *Serial Item and Contribution Identifier (SICI).* https://www.niso.org/publications/z397-2013

Autodesk. (2010, September 15). *Activating Alias from a Serial Numbers*. http://www.autodesk.com/techpubs/aliasstudio/2010/index.html

Apple Inc. (2011, July 18). *Uniquely Identifying a Macintosh Computer*". https://developer.apple.com/library/mac/technotes/tn1103/_index.html

Balasubramanian, R. and Viswanathan, V. (2004). *Data structures and Algorithms.* 2nd Edition, R.K. Publishers

Balucanag, J. (2014, September 8). *Experimental Research*. https://www.slideshare.net/jobitonio/experimental-research-38847004

Bauer, F. (2021, September 9). *IEEE802.1X.* https://commons.wikimedia.org/w/index.php?curid=39448253

Behringer, M.H. (2009). Classifying network Complexity, *Cisco Systems*, conferences.sigcomm.org/co-next/2009/workshops/rearch/papers/Behringer.pdf

Bell, D. (2012). UML Basics, An Introduction to Unified Modeling Language, *Rational Software*, http://www.therationaledge.com/content/jun_03/f_umlintro_db.jsp

Bernstsein, M.E. & Braude, E.J. (2011). *Software Engineering Modern Approach, Second* Edition*.* Jon Wesley & Sons

Brenna, L. (2021, June 15). "*What is the RADIUS Protocol*?". https://jumpcloud.com/blog/what-is-the-radius-protocol

Bruegge, B. & Dutoit, H.A. (2010). *Object-Oriented Software Engineering, Using UML, Patterns, and Java*, International Edition. Pearson

Butterfield, A. & Ngondi, G.E. (2016). *A Dictionary of Computer Science,*7th Edition. Oxford University Press. http:///doi.org.10.1093/acref/9780199688975.001.0001

Canavan, J.E. (2012). *Fundamentals of Network Security.* Artech House

Cardenas, D.E. (2018). *MAC Spoofing: An Introduction* [White Paper]. GIAC Security Essentials Certification (GSEC). https://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315

Carter, R.A., Anton, A.I., Dagnino, A., Williams, L. (2001). Evolving beyond requirements creep: a risk-based evolutionary prototyping model. *Requirements Engineering, 2001, Proceedings, Fifth IEEE International Symposium on*, vol., no.(94-101)

Cash, P., Stancovic, T. and Storga, M. (2016). Experimental Research Design, Approaches, Perspectives, Applications, *Springer*, uploaded by Storga, M. on file:///C:/Users/JK/Downloads/Cash_Stankovic_Storga_EDR_2016.pdf

Chad, F. (2017, May 3). *IP Authentication vs. Username and Password Login*. https://www.proxykey.com/ip-authentication-vs-username-password-login/

Chiradeep, B. (2021, June 28). *What Is Network Access Control? Definition, Key Components, and Best Practices*. https://www.toolbox.com/it-security/network-security/articles/what-is-network-access-control

Choi, Y., (2014, July 12). *Use Case Diagram (UCD)*. https://www.slideserve.com/nikki/use-case-diagram-ucd

Christian. (2021, June 15). *Authentication Protocol: 802 dot 1x and EAP types for Wired and Wireless Authentication.* https://techdirectarchive.com/2021/06/15/authentication-protocol-802-dot-1x-and-eap-types-for-wired-and-wireless-authentication/

Cronholm, S. & Goldkuhl, G. (2003). Strategies for Information Systems Evaluation- Six Generic Types. *Electronic Journal of Information Systems Evaluation (EJISE), 6*(2), 1-13. http://www.vits.org/publikationer/dokument/434.pdf

Cisco. (2018, August 14). *Configuring MAC-Based Authentication on a Switch*. https://www.cisco.com/c/en/us/support/docs/smb/switches/cisco-350-series-managed-switches/smb5836-configuring-mac-based-authentication-switch.html

Coulouris, G., Dollimore, J., Kindberg, T. & Blair, G. (2012). *Distributed Systems, Concepts, and Design*, Fourth Edition. Addison Wesley

Danev, B., Zanetti, D. & Capkun, S. (2015). On physical-layer identification of wireless devices, *ACM Computing Surveys, 451(1), 1-29.* https://doi.org/10.1145/2379776.2379782

Derekyoung. (2017, February 10). *How to Change a BIOS Serial Number.* https://itstillworks.com/how-to-change-a-bios-serial-number-10394.html

DHS. (2017). *A Guide to Securing Networks for Wi-Fi (IEEE 802.11 Family*) [White Paper]. Retrieved from https://www.us-cert.gov/sites/default/files/publications/A_Guide_to_Securing_Networks_for_Wi-Fi.pdf

Diaz, J. (2015, January 22). *Password-based Authentication*. https://www.incibe-cert.es/en/blog/password-based-authentication

Diceus. (2022, May 18). *Top 5 steps to define crucial POC success criteria*. https://diceus.com/poc-success-criteria/

Dordal, P.L. (2018). *An introduction to Computer Networks*. Loyola University Chicago. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

Dooley, K. (2015, March 31). *An Introduction to Network Authentication Protocols*. https://www.auvik.com/franklyit/blog/authentication-protocols/

EBSCO. (2022, February 7). *What is IP Address Authentication*? https://connect.ebsco.com/s/article/What-is-IP-Address-Authentication?language=en_US

Economic Times. (2022, April 27). *What is 'Authentication'*. https://economictimes.indiatimes.com/definition/authentication

Eck, D.J. (2014). *Introduction to Programming Using Java,* SoHo Books. https://archive.org/details/javanotes6

Emihem. (2022, January 13). *RADIUS*. https://en.wikipedia.org/wiki/RADIUS

Fredrisson, J. (2017, May 3). *User and Machine Network Authentication*. https://www.wiresandwi.fi/blog/user-and-machine-authentication

Froehlich, A. & Tuomenoksa, M. (2020, July 13). *The differences between PAP and CHAP*. https://www.techtarget.com/searchnetworking/answer/Which-is-most-secure-CHAP-or-PAP

Fruhlinger, J. & Snyder, J. (2021, April 28). *802.1X: What you need to know about this LAN-authentication standard.* https://www.networkworld.com/article/2216499/wireless-what-is-802-1x.html

Garska, K. (2016, November 9). *Higher Education's Unique Identity and Access Management Challenges.* https://blog.identityautomation.com/higher-educations-unique-identity-and-access-management-challenges

Gill, S. & Dahiya, M. (2017). Protecting MAC Address Spoofing in IEEE 802.11 Using MATLAB. *International Journal of Advanced Research in Computer Science*, *8(*3*)*. 305-308. https://www.researchgate.net/publication/321864906_Protecting_MAC_Address_Spoofing_in_IEEE_80211_Using_MATLAB

Graziani, R. (2017). *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6.* 2nd Edition, Cisco Press

Gregersen, E. (2021, August 11). *Digital Certificate*. https://www.britannica.com/topic/digital-certificate

Harold, E.R. (2013). *Java Network Programming,* 4th Edition. O'Reilly Media

Hawkins, L. (2016). Reliability in Research. *Slideplayer.* https://slideplayer.com/slide/7016043

Hoy, M. B. (2019). An Introduction to RA21: Taking Authentication Beyond IP Addresses. *Medical Reference Services Quarterly*, *38*(1), 81-86. https://doi.org/10.1080/02763869.2019.1554370

Huawei. (2019, June 4). *Understanding 802.1X Authentication.* https://support.huawei.com/enterprise/en/doc/EDOC1100086527

ICANN. (2011). Beginner's Guide to INTRNET PROTOCOL(IP) ADDRESSES. *ICANN.* https://www.icann.org/en/learning/ip-addresses-beginners-guide-04mar11.en.pdf

IEEE-USA. (2009). *Next Generation Internet: IPv4 Address Exhaustion, Mitigation Strategies and Implications for the U.S.* [White Paper]. IEEE-USA. https://www.ieeeusa.org

Intel. (2021, October 28). *802.1X Overview and EAP Types.* https://www.intel.com/content/www/us/en/support/articles/000006999/wireless/legacy-intel-wireless-products.html

Iwaya, A. (2015, September 13). *How is the Uniqueness of MAC Addresses Enforced?* https://www.howtogeek.com

Jeevanesh, (2017). *Different types of addresses used in the TCP/IP protocol*. https://www.ques10.com/p/21477/discuss-the-different-types-of-addresses-used-in-t/

Johnson, K. (2021, December 14). *Use these 6 user authentication types to secure networks.* https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks

Karlinsky, E. (2021, April 9). *Two-Factor Authentication vs. Multi-Factor Authentication: What are the Risks*. https://www.okta.com/blog/2016/12/two-factor-authentication-vs-multi-factor-authentication-what-are-the-risks/

Karasneh, B. & Chaudron, M. R. V. (2013). Extracting UML models from images. *2013 5th International Conference on Computer Science and Information Technology*, *2013,* pp. 169-178. doi: 10.1109/CSIT.2013.6588776

Kumar, C. (2022, April 24). *11 Best IP Scanner Tools for Network Management*, https://geekflare.com/network-scanner/

Kurose, J.K. & Ross K.W. (2013). *Computer Networking: A Top_down Approach Featuring the Internet*, *6th Edition*. Addison Wesley

Larue-Langloise, R. (2022, February 3). *The 8 Best IP Scanners for Mac in 2022*. https://www.addictivetips.com/net-admin/best-ip-scanners-for-mac/

Larusson, J. (2015, December 15). *Doing Research and Development with Proof of Concepts*. http://researchnetwork.pearson.com/digital-data-analytics-and-adaptive-learning/research-development-proof-concepts

Latze, C. (2010). *Towards a Secure and User-Friendly Authentication Method for Public Wireless Networks*. Logos Verlag

Lavassani, K.M., Movahedi, B. & Kumar, V. (2010). Identification in Electronic Networks: Characteristics of e-Identifiers. *Eight International Conference on Electronic Commerce (ICEC)*. 2006, Fredericton, New Brunswick, Canada

Lawson, L. (2017, April 17). *Managed Services: A Security Problem and Solution.* https://www.esecurityplanet.com/network-security/managed-services.html

Lee, T. (2010). *Securing your Meru Network* [White Paper]. Meru Networks. https://www.martolvan.is/spuningar-og-svor/skjoel-og-fraeesluefni/meru-networks/meru-fraedhsluefni/1-meru-bpg-1/file

Leggott, M., Shearer, K., Ridsdale, C., Baker, D. & Barsky, E. (2016). Unique Identifiers: Current Landscape and Future Trends, *Research Data Canada IDs Working Group, Standards, and Interoperability Committee*. http://doi.org/10.5281/zenodo.557106

Lehtonen, M., Staake, T., & Michahelles, F. (2008). From identification to authentication–a review of RFID product authentication techniques. In *Networked RFID Systems and Lightweight Cryptography* (pp. 169-187). Springer Berlin Heidelberg

Leo, R.V. (2004). Predicting consumer intentions to use on-line shopping: the case for an augmented technology acceptance model**.** *Information and Management,* (41), 747-762

Leurs, B. & Duggan K. (2018, December 20). *POC, Prototype, Pilot, MVP- What's in a Name?* https://www.nesta.org.uk/blog/proof-of-concept-prototype-pilot-mvp-whats-in-a-name/, Loshin, P. (2021, September 29). *CHAP (Challenge-Handshake Authentication Protocol)*. https://www.techtarget.com/searchsecurity/definition/CHAP-Challenge-Handshake-Authentication-Protocol

Luis-Garzia, R., Albero-Lopez, C., Aughzout, O. & Ruiz-Alzola, J. (2013). Biometric Identification Systems. *Signal Processing* (83), 2539-2557

Maayan, G. D. (2020, September 25). *5 User Authentication Methods that can prevent the Next Breach*. https://www.idrnd.ai/5-authentication-methods-that-can-prevent-the-next-breach/

MacPherson, L. (2018). *5 Steps to a Proof of Concept for Successful Software Development*. https://designli.co/blog/5-steps-proof-concept-successful-software-develoment

Mareco, D. (2015). *5 Campus Wi-Fi Trends Every College CIO Needs to Follow*. https://www.securedgenetworks.com/blog/5-campus-wifi-trends-every-college-cio-needs-to-follow

Matkar, P. M., Bhiogade, P. V., Patle, P. P., Gangewane, P. & Shelkepment, F. M. (2018). A Review of Authentication Protocols. *International Journal on Recent and Innovation Trends in Computing and Communication, 4*(6), 220-226. ISSN: 2321-8169. http://www.ijritcc.org

Mehta, M. (2021, March 22). *The Pros and Cons of Passwordless Authentication*. https://sectigostore.com/blog/the-pros-and-cons-of-passwordless-authentication/

Meru Networks. (2009). *WLANs in Higher Education* [White Paper]. Meru Networks. http://www.sagetechs.com/

Microsoft Corporation. (2018). *Microsoft Computer Dictionary*, 6th Edition. Microsoft Press

Mishra, D. & Ali, H. (2013, July 28). *Authentication*. https://www.slideshare.net/AliRaw1/authentication-24695177

Mohapatra, S. K., Choudhury, R. R. & Das, P. (2014). The Future Directions in Evolving Wi-Fi: Technologies, Applications and Services, *International Journal of Next-Generation Networks* (IJNGN), *(6)*3, 13-22

Nadia. (2013, June 5). *What is Radius Authentication Protocol?* https://www.routerfreak.com/radius-protocol/

Nagaraj, S., Kishore, B., Rao, G.N. & Ramachandra, M. (2010). A Comparative study of IPv6 Statistical Approach, *International Journal on Computer Science and Engineering (IJCSE), (02)*04, 1367-1370

Nixon, R. (2018). *Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5,* 5th Edition. O'Reilly

N-able. (2020, February 6). *How does Token-Based Authentication Work?* https://www.n-able.com/blog/how-does-token-based-authentication-work

O'Leary, Z. (2011). *The Essential Guide to Doing Your Research Project*. SAGE publications

Oracle. (2010). *Password Authentication Protocol (PAP) Frames.* https://docs.oracle.com/cd/E19096-01/sol.ppp301/805-4018/6j3qil166/index.html

Orbitco. (2015, November 9). *PPP, PAP Explained with Examples*. https://www.orbit-computer-solutions.com/password-authentication-protocol-pap/

Pang, J., Greenstein, B., Gummadi, R., Seshan, S. & Wetherall, D. (2007). 802.11 user fingerprinting. *Proceedings of the 13th annual ACM international conference on Mobile computing and networking September 2007.* Pages 99–110. https://doi.org/10.1145/1287853.1287866

Paramalways. (2009). Dynamic Programming, *Technology*, https://www.slideshare.net,

Paulsen, C. & Byers, R. (2019). Glossary of Key Information Security Terms, NIST Interagency/Internal Report (NISTIR). *National Institute of Standards and Technology, Gaithersburg, MD,* [online]. https://doi.org/10.6028/NIST.IR.7298r3

Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *(24)*3, 45-77. https://doi.org/10.2753/MIS0742-1222240302

Pierce, M. (2021, May 12). *What is NAC? Network Access Control Explained*. https://www.securedgenetworks.com/blog/network-access-control

Popa, B. (2012, February 24). *Intuitive Monitoring*. https://www.softpedia.com/get/Network-Tools/Network-IP-Scanner/PCFinder.shtml

Poremba, S.M. (2022, January 24). *Network Access Control: Restricting and Monitoring Access to Your Network and Data*. https://www.esecurityplanet.com/network-security/network-access-control.html

Prakash, A. & Kumar, U. (2018). Authentication Protocols and Techniques: *A Survey, International Journal of Computer Sciences and Engineering, 6*(6), 1014-1020, E-ISSN: 2347-2693

Rajesh, K. (2010, October 6). *How IEEE 802.1x secures the Network edge*. https://excitingip.com/758/ieee-802-1x-elements-protocols-advantages/

Raji, M.O. (2014). Design and Implementation of Wireless Network. *Research Gate*, https://doi.org/ 10.13140/2.1.4578.4649

Richards K, White R, Nicolson N & Pyle R. (2011). A Beginner's Guide to Persistent Identifiers. *Copenhagen: GBIF Secretariat*. https://doi.org/10.35035/mjgq-d052

Robb, D. (2022, January 24). *Top Network Access Control (NAC) for 2022*. https://www.esecurityplanet.com/products/network-access-control-solutions/

Robyns, P., Bonné, B., Quax, P. and Lamotte, W. (2017). Noncooperative 802.11 MAC Layer Fingerprinting and Tracking of Mobile Devices. *Security and Communication Networks*. https://doi.org/10.1155/2017/6235484

Rossi, M. & Siau, K. (2011). Evaluation techniques for systems analysis and design modelling methods – a review and comparative analysis. *Information Systems Journal*, *21(*3), 249-268. https://doi.org/10.1111/j.1365-2575.2007.00255.x

Sectona. (2021, December 3). *The importance of Password-Based Authentication*. https://sectona.com/resources/password-based-authentication/

Servicenow. (2022, February 3). *IP Range Based Authentication*. https://docs.servicenow.com/bundle/sandiego-platform-administration/page/administer/login/concept/c_IPRangeBasedAuthentication.html

Shacklett, E. M. (2021, September 13). *Authentication*. https://www.techtarget.com/searchsecurity/definition/authentication

Shay, W. A. (2004). *Understanding Data Communication and Networks*, Third Edition. Thomson Learning

Singh, R. and Sharma, P.T. (2017). On the IEEE 802.11i security: a denial-of-service perspective. *Security Communication Networks*, *(8)*7, 1378–1407. http:/doi.org/10.1002/sec.1079

SmartDraw. (2019). *HOW TO DRAW DATA FLOW DIAGRAMS.* http://120.105.184.180/lwcheng/SSADM/

Smith, J. Gill, R. and Clark, A. (2006). On Securing Wireless LAN Access to Government Information Systems, In Mendis, Priyan and Lai, Joseph and Dawson, Ed. *Proceedings* *of 2006 RNSA Security Technology Conference – Recent advances in security technology. Canberra, Australia,* 440-453. http://eprints.qut.edu.au

SolarWInds Passportal. (2021, March 2). *Which password authentication method works for businesses?* https://www.passportalmsp.com/blog/which-password-authentication-method-works-best-businesses

Solomon, S.E. (2010). The eggshell: strength, structure, and function, *British Poultry Science*, *51*(1), 52-59, https:/doi.org/10.1080/00071668.2010.497296

Stallings, W. (2011). *Network Security Essentials, Applications and Standards*, Fourth Edition. Pearson Education

Strom, D. (2021, April 8). *What is IAM? Identity and access management explained.* https://www.csoonline.com/article/2120384/what-is-iam-identity-and-access-management-explained.html

Su X., Chu C, Prabhu B.S., & Gadh R. (2007). On the creation of Automatic Identification and Data Capture infrastructure via RFID, The Internet of Things: from RFID to the Next-Generation Pervasive Networked Systems. *Auerbach Publications*, Taylor & Francis Group, 24 pp. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.92.2317

Tanenbaum, A. S. (2011). *Computer Networks,* 4th Edition. Pearson Education

Team, M. (2021, May 06). *Passwords are so 90s... certificates are the future of users' authentication*. https://www.zealid.com/en/blog/passwords-are-so-90s...certificates-are-the-future-of-usersauthentication

Temiz, S. (2014). *UML Basics*. https://www.slideshare.net/SerdarTemi/3-uml

Unar, J.A., Seng, W.C. & Abbasi, A. (2014*).* A review of biometric technology along with trends and prospects. *ScienceDirect, 47*(8), 2673-2688. https://doi.org/10.1016/j.patcog.2014.01.016

Visual-paradigm. (2018). *How to Draw an Activity Diagram in UML.* https://www.visual-paradigm.com/tutorials/how-to-draw-activity-diagram-in-uml/

Vital, T. (2019, June 3). *Access Control: Identification, Authentication, and Authorization.* https://www.thomasvitale.com/access-control-authentication-authorization/

Wallace, K. (2018, April 06). *Wireless LAN Security*. https://www.kwtrain.com/blog/wlan-security

Wang, J. (2007). Digital Object Identifiers and Their Use in Libraries, *Serials Review*, *33*(3), 161-164

Wang, C., Gerdes, M.R., Guan, Y. and Kusera, K.S. (2016). *Digital Fingerprinting*. Springer. https://link.springer.com/book/10.1007/978-1-4939-6601-1

Watanabe, Y., Otani, M., Eto, H., Watanabe K. & Tadaki, S. (2013). A MAC address based authentication system applicable to campus-scale network. *2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2013, pp. 1-3.

Webner, E. (2021, May). *Extensible Authentication Protocol (EAP*). https://www.techtarget.com/searchsecurity/definition/Extensible-Authentication-Protocol-EAP

Williams, L. (2022, April 2). *OSI Model Layers and Protocols in Computer Network*. https://www.guru99.com/layers-of-osi-model.html

Wikipedia. (2022, April 6). *Identity Management*. https://en.wikipedia.org/wiki/Identity_management

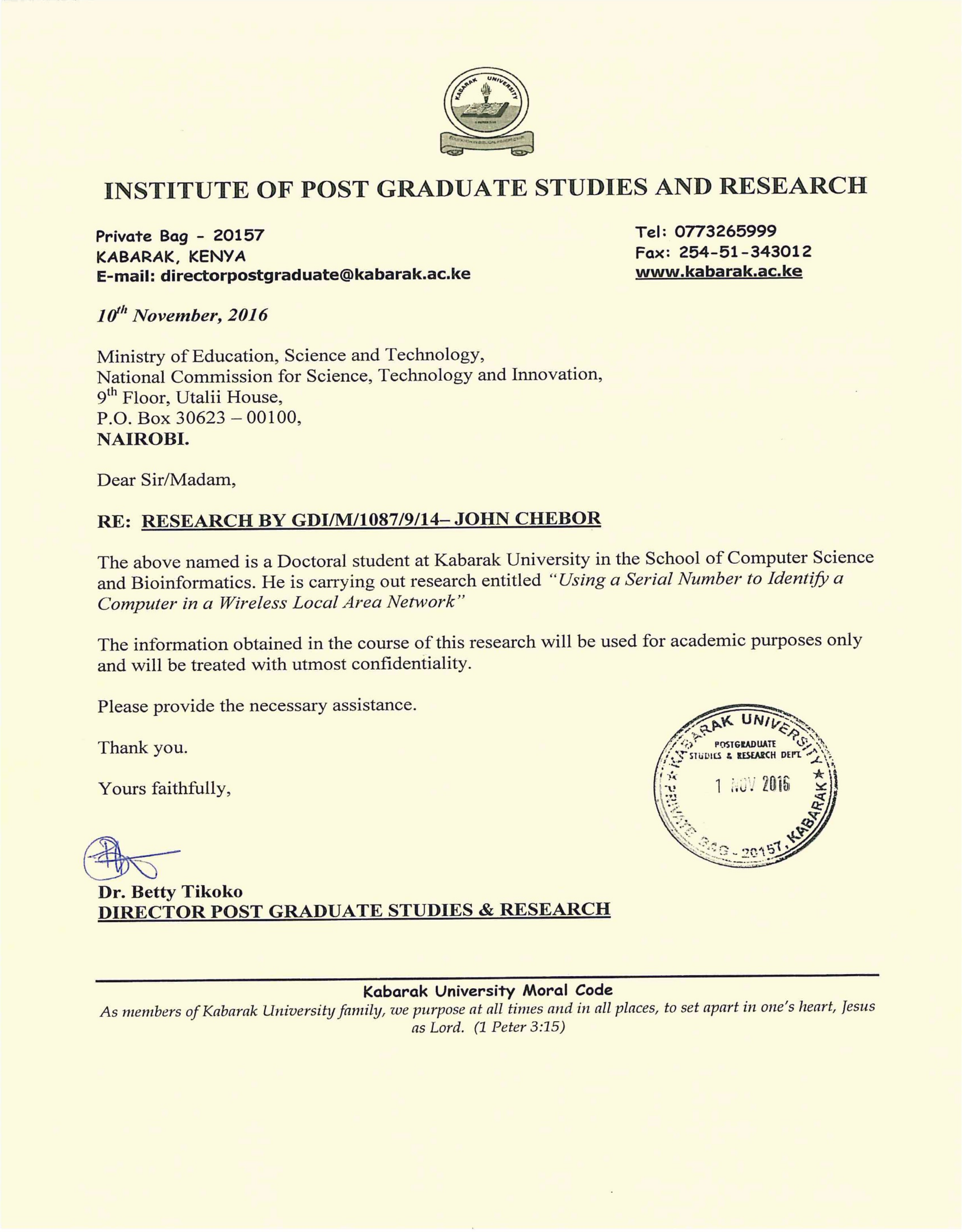
WorkOS. (2020, October 14). *Authentication Protocols: Your Guide to the Basics*. https://workos.com/blog/authentication-protocols-your-guide-to-the-basics

Xu, Q., Zheng, R., Saad, W. and Han, Z. (2015). Device Fingerprinting in Wireless Networks: Challenges and Opportunities. *IEEE Communications Surveys & Tutorials*, *18*(1), 94-104. https://doi.org/10.1109/COMST.2015.2476338

Yang M.C. and Epstein, J.D. (2005). A study of prototypes, design activity, and design outcome. *Design Studies, 26(*64), 649-669. https://doi.org/10.1016/j.destud.2005.04.005

APPENDIX I

Letter of Introduction



INSTITUTE OF POST GRADUATE STUDIES AND RESEARCH

Private Bag - 20157
KABARAK, KENYA
E-mail: directorpostgraduate@kabarak.ac.ke

Tel: 0773265999
Fax: 254-51-343012
www.kabarak.ac.ke

10th November, 2016

Ministry of Education, Science and Technology,
National Commission for Science, Technology and Innovation,
9th Floor, Utalii House,
P.O. Box 30623 – 00100,
NAIROBI.

Dear Sir/Madam,

RE: RESEARCH BY GDI/M/1087/9/14– JOHN CHEBOR

The above named is a Doctoral student at Kabarak University in the School of Computer Science and Bioinformatics. He is carrying out research entitled *"Using a Serial Number to Identify a Computer in a Wireless Local Area Network"*

The information obtained in the course of this research will be used for academic purposes only and will be treated with utmost confidentiality.

Please provide the necessary assistance.

Thank you.

Yours faithfully,

KABARAK UNIVERSITY
POSTGRADUATE STUDIES & RESEARCH DEPT.
1 NOV 2016
PRIVATE BAG - 20157, KABARAK

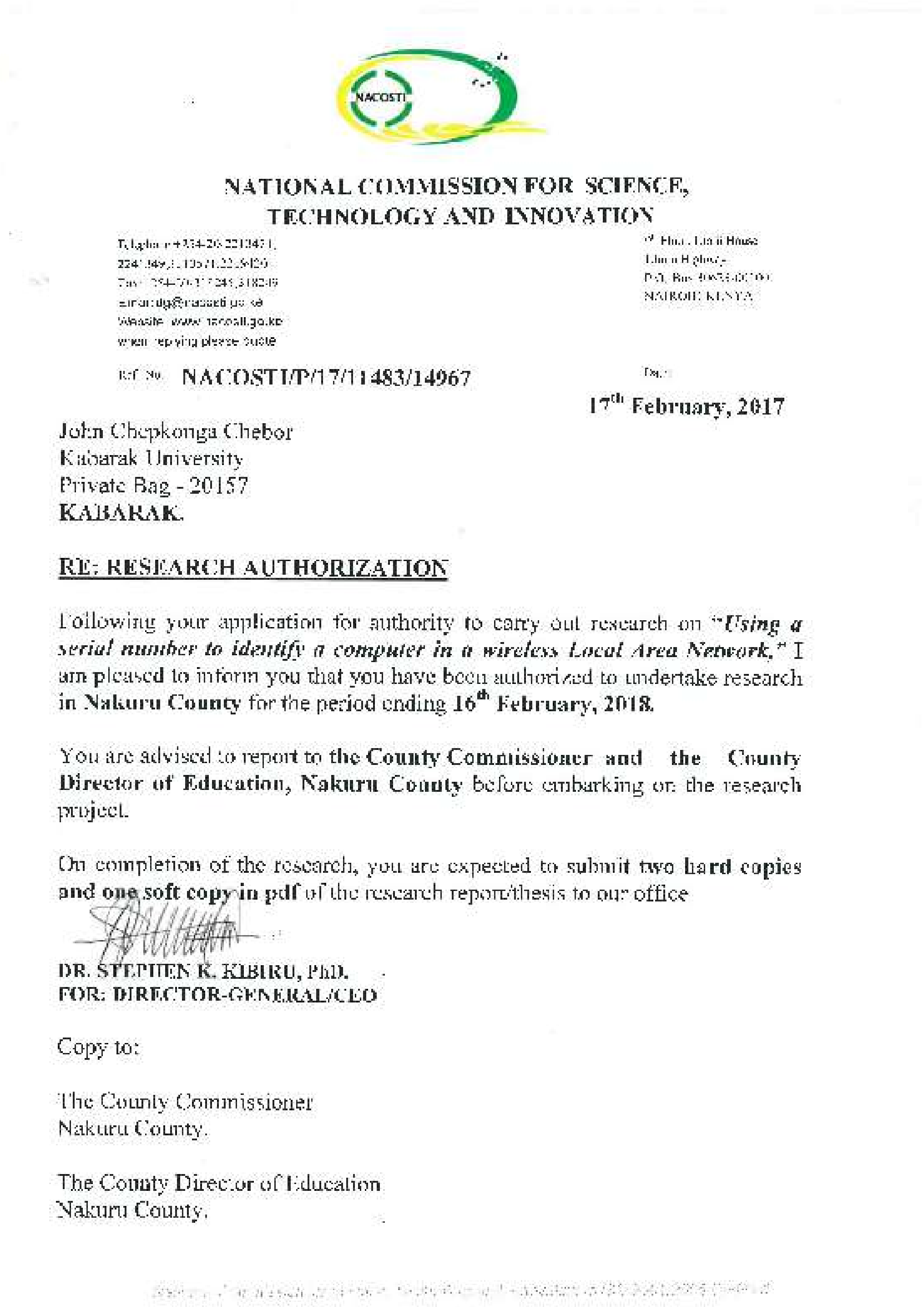
Dr. Betty Tikoko
DIRECTOR POST GRADUATE STUDIES & RESEARCH

Kabarak University Moral Code

As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, Jesus as Lord. (1 Peter 3:15)

APPENDIX II

Letter of Research Authorization



NATIONAL COMMISSION FOR SCIENCE,
TECHNOLOGY AND INNOVATION

Telephone:+254-20-2213471,
2241349,3310571,2219420
Fax:+254-20-318245,318249
Email: dg@nacosti.go.ke
Website: www.nacosti.go.ke
When replying please quote

9th Floor, Utalii House
Uhuru Highway
P.O. Box 30623-00100
NAIROBI-KENYA

Ref. No. **NACOSTI/P/17/11483/14967**

Date:

17th February, 2017

John Chepkonga Chebor
Kabarak University
Private Bag - 20157
KABARAK.

<u>RE: RESEARCH AUTHORIZATION</u>

Following your application for authority to carry out research on ***"Using a serial number to identify a computer in a wireless Local Area Network,"*** I am pleased to inform you that you have been authorized to undertake research in **Nakuru County** for the period ending **16th February, 2018.**

You are advised to report to **the County Commissioner and the County Director of Education, Nakuru County** before embarking on the research project.

On completion of the research, you are expected to submit **two hard copies and one soft copy in pdf** of the research report/thesis to our office

DR. STEPHEN K. KIBIRU, PhD.
FOR: DIRECTOR-GENERAL/CEO

Copy to:

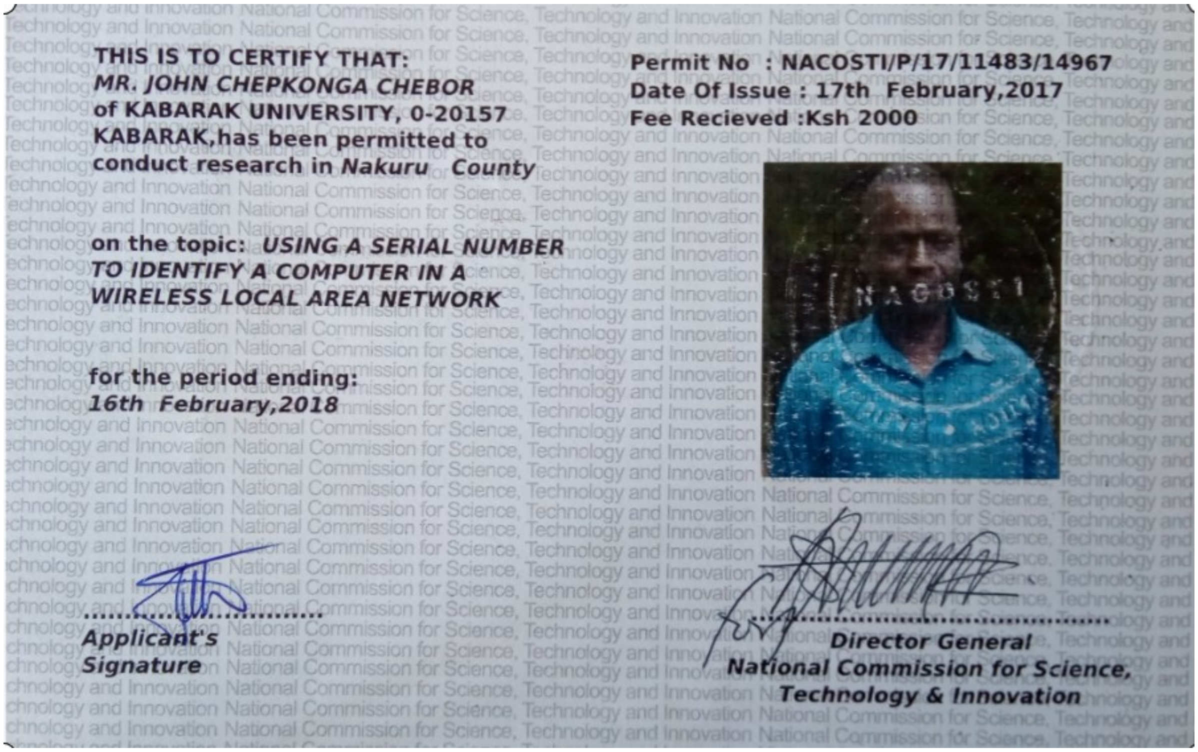
The County Commissioner
Nakuru County.

The County Director of Education
Nakuru County.

[illegible]

APPENDIX III

Research Permit



THIS IS TO CERTIFY THAT:
MR. JOHN CHEPKONGA CHEBOR
of KABARAK UNIVERSITY, 0-20157
KABARAK,has been permitted to
conduct research in *Nakuru County*

on the topic: *USING A SERIAL NUMBER TO IDENTIFY A COMPUTER IN A WIRELESS LOCAL AREA NETWORK*

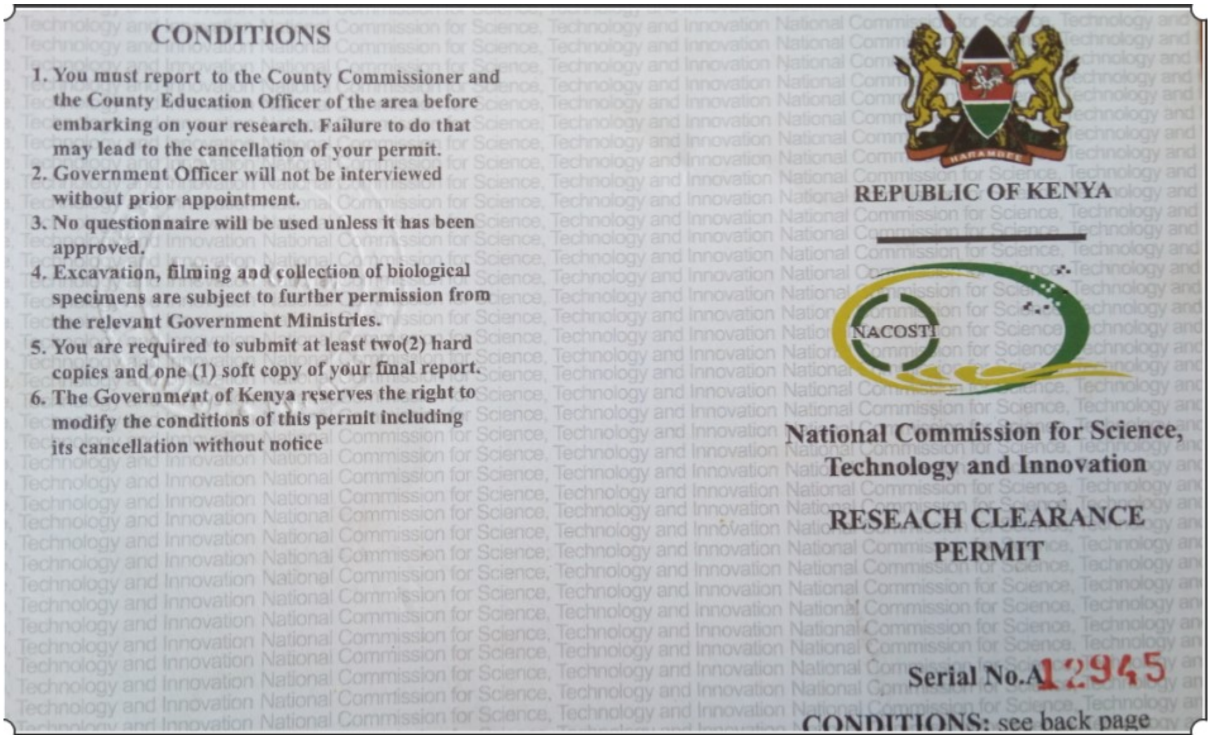
for the period ending:
16th February,2018

Permit No : NACOSTI/P/17/11483/14967
Date Of Issue : 17th February,2017
Fee Recieved :Ksh 2000

NACOSTI

....
Applicant's Signature

....
Director General
National Commission for Science, Technology & Innovation



CONDITIONS

1. You must report to the County Commissioner and the County Education Officer of the area before embarking on your research. Failure to do that may lead to the cancellation of your permit.
2. Government Officer will not be interviewed without prior appointment.
3. No questionnaire will be used unless it has been approved.
4. Excavation, filming and collection of biological specimens are subject to further permission from the relevant Government Ministries.
5. You are required to submit at least two(2) hard copies and one (1) soft copy of your final report.
6. The Government of Kenya reserves the right to modify the conditions of this permit including its cancellation without notice

HARAMBEE

REPUBLIC OF KENYA

NACOSTI

National Commission for Science, Technology and Innovation

RESEACH CLEARANCE PERMIT

Serial No.A 12945

CONDITIONS: see back page

APPENDIX IV

SNAM Source Code

//Computer Registration Code

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Admin
 */
public class Register extends javax.swing.JFrame {
    Connection con = null;
    ResultSet rs = null;
    PreparedStatement pst = null;
    public String Serial_Number, Name;
    /**
     * Creates new form Register
     */
    public Register() {
        initComponents();
        data();
    }
    public void data() {
        try {
            con = ConnectDB.connect();//create a connection
            String sql = "SELECT * FROM devices";
            pst = con.prepareStatement(sql);
            rs = pst.executeQuery();
            tblRegistered.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
            Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void SaveNew() throws SQLException {
        //Serial_Number = txt
        /**
         * save the values to the database
         *
         * //
         */
//        con = ConnectDB.connect();
//        String sql = "INSERT INTO computers (Serial_Number,Name)values('" + Serial_Number
+ "','" + Name + "')";
//        // String sql1 = "SELECT Serial_Number FROM computers WHERE Serial_Number = '"
+ txtSerial.getText() + "'";
//        pst = con.prepareStatement(sql);
//        pst.execute();
//        JOptionPane.showMessageDialog(null, "Successfully saved", "Record",
JOptionPane.INFORMATION_MESSAGE);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        tblRegistered = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        tblRegistered.setModel(new javax.swing.table.DefaultTableModel(
      new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
      },
      new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
      }
    ));
    jScrollPane1.setViewportView(tblRegistered);

    jButton1.setText("Add New");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
      }
    });
    jButton2.setText("Delete");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
      }
    });
    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(35, 35, 35)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(74, 74, 74)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
          .addComponent(jButton1)
          .addComponent(jButton2))
        .addGap(44, 44, 44))
    );
    jLabel1.setText("Registered Devices");
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 203,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
      .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(65, 65, 65))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
```

```
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
```

```
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Register().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable tblRegistered;
    // End of variables declaration
}
```

//Database Connection

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import java.sql.*;
import javax.swing.JOptionPane;
/**
 *
 * @author Gamer
 */
public class ConnectDB {
    //create a variable to establish the connection to the database
    Connection con = null;
    static String db = "system_db";
    static String user = "java";
    static String pass = "play";
    static String url ="jdbc:mysql://127.0.0.1:3306/";

    //create a method to connect the database
    public static Connection connect() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con
=DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/system_db","java","play");
            return con;
        } catch (ClassNotFoundException | SQLException e) {
            JOptionPane.showMessageDialog(null, e);
            System.exit(0);
            return null;
        }
    }
}
```

//Display Code

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import com.opencsv.CSVWriter;
import java.awt.GridLayout;
import java.awt.HeadlessException;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.UnknownHostException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author Gamer
 */
public class Display extends javax.swing.JFrame {

    Connection con = null;
    ResultSet rs, rs1 = null;
    PreparedStatement pst, pst1 = null;
    //String variable
    String address;
    String ipAddress;
    String Computer_Name;
    int row;

    /**
     * Creates new form Display constructor
     *
     * @throws java.net.UnknownHostException
     */
    public Display() throws UnknownHostException, IOException {
        initComponents();

        con = ConnectDB.connect();///creates connection
        jPanel1.setVisible(false);
        tblRegistered.setVisible(false);
        jScrollPane4.setVisible(false);
        setLocationRelativeTo(null);
        Del();
        syInfo.initCSV("infile1.csv");
        syInfo.getAddress();
        syInfo.getCompInfo();
        sysInitiate();
        data1();
        data();

        //set other components invisible
    }
    /**
     * compare and get unmatched data
     */
    public void sysInitiate() {

            String sql = "SELECT c.serial_Number "
                    + "FROM computers c "
                    + "LEFT JOIN registereddevices d "
                    + "ON d.Serial_Number=c.Serial_Number "
                    + "WHERE d.Serial_Number IS NULL;";

            pst = con.prepareStatement(sql);
            rs = pst.executeQuery();

            tblUnregistered.setModel(DbUtils.resultSetToTableModel(rs));
            List<String> numdata1 = new ArrayList<String>();
            int rc = tblUnregistered.getRowCount();

            for (int count = 0; count < rc; count++) {
                numdata1.add(tblUnregistered.getValueAt(count, 0).toString());
            }

            //print value one at a time
            for (int i = 0; i < rc; i++) {
                System.out.println(numdata1.get(i));
                //get the ip add of each serial and add to csv
            String sql1 = "SELECT * FROM computers WHERE Serial_Number='" +
numdata1.get(i) + "'";
            pst = con.prepareStatement(sql1);
            rs = pst.executeQuery();
            if (rs.next()) {
                String ipA = rs.getString("IpAddress");

                String[] ipoutElements = ipA.split(",");
                List<String> fixedLengthList = Arrays.asList(ipoutElements);
                ArrayList<String> listofstring = new ArrayList<String>(fixedLengthList);
                String iparray[] = new String[listofstring.size()];
                int c;
                for (c = 0; c < listofstring.size(); c++) {
                    if (c % 2 == 0) {
                        iparray[c] = listofstring.get(c);
                    } else {
                        //
                    }

                }

                String[] str = {iparray[0], null};
                System.out.println("elements fixed lenth :" + fixedLengthList);
                System.out.println("elements strings:" + listofstring.get(0));
                System.out.println("elements array:" + str[0]);

                String csv = "infile.csv";
                syInfo.initCSV(csv);
                System.out.println("*********************");
                for (int b = 0; b < iparray.length; b++) {
                    System.out.println(iparray[b]);
                }
                System.out.println(str.length + "is the length");
                System.out.println(str);
                System.out.println("**8*********************");

                try ( // boolean alreadyExist = new File(csv).exists();

                        CSVWriter writer = new CSVWriter(new FileWriter(csv, true), '\n') //append to
file
```

```
                        ) {
                    writer.writeNext(str);//change to ip
                    writer.flush();
                    //close the writer
                    // } else {
                }
                syInfo.getUStop();
                //next remove disabled computer from the table
                //update status of str in db

                for (int a = 0; a < iparray.length; a++) {
                    if (iparray[a] != null) {
                        System.out.println("db**********" + iparray[a]);
                        setStatus("Disabled", iparray[a].toString());
                    }
                }
                data();

            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
    }
}
public void setStatus(String status, String str) {
    try {
        con = ConnectDB.connect();//create a connection
        String query1 = "UPDATE computers SET status= 'disabled' WHERE  IpAddress=?";
         PreparedStatement pst1 = con.prepareStatement(query1);//prepare sql for execution
        pst1.setString(1, str);
        pst1.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
        ex.printStackTrace();
    }
}

/**
```

```
     * function to get all connected devices
     *
     */
    /**
     * function to get the ip address of a computer based on the computer name
     *
     */
    public void Del() {
        String delSQL = "delete from computers";
        try {
            Statement stmt = con.createStatement();
            stmt.executeUpdate(delSQL);
        } catch (Exception e) {
        }
    }

    public void addNew() {
        JTextField serialNumber = new JTextField(15);
        JTextField pcName = new JTextField(30);
        JPanel panel = new JPanel(new GridLayout(2, 2));
        panel.add(new JLabel("Serial Number:"));
        panel.add(serialNumber);
        panel.add(new JLabel("PC Name:"));
        panel.add(pcName);
        int reslt = JOptionPane.showConfirmDialog(null, panel, "Add New Device",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
        if (reslt == JOptionPane.OK_OPTION) {
            //add to db
            try {
                con = ConnectDB.connect();//create a connection
                String sql = "INSERT INTO devices (Serial_Number,Name)values('" +
serialNumber.getText() + "','" + pcName.getText() + "')";
                pst = con.prepareStatement(sql);//prepare sql for execution
                pst.execute();
                System.out.println("saved succeed");
                data1();
                //JOptionPane.showMessageDialog(sysInfo,"Successfully
saved","Record",JOptionPane.INFORMATION_MESSAGE);
            } catch (SQLException e) {
                System.out.println(e);
        }
    } else {
        System.out.println("cancelled");
    }
}

public void data1() {
    try {
        String sql = "SELECT * FROM registereddevices";

        pst = con.prepareStatement(sql);
        rs = pst.executeQuery();
        tblReg.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (SQLException ex) {
        Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * This method populates data to the table with all connected devices*
 */
public void data() {
    try {
        String sql = "SELECT Serial_Number,Name,IpAddress,Status FROM computers";
        pst = con.prepareStatement(sql);
        rs = pst.executeQuery();
        tblDevice.setModel(DbUtils.resultSetToTableModel(rs));
        /**
         * gets the data from the database.condevices
         *
         *
         * String sql1 = "SELECT * FROM condevices"; PreparedStatement pst1
         * = con.prepareStatement(sql1); ResultSet rs1 =
         * pst1.executeQuery();
         * tblDevice.setModel(DbUtils.resultSetToTableModel(rs1));
         *
         */
        List<String> numdata = new ArrayList<String>();
        for (int count = 0; count < tblRegistered.getRowCount(); count++) {
            numdata.add(tblRegistered.getValueAt(count, 0).toString());
            }
            System.out.println(numdata);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTabbedPane1 = new javax.swing.JTabbedPane();
        jTabbedPane2 = new javax.swing.JTabbedPane();
        jPanel1 = new javax.swing.JPanel();
        jScrollPane3 = new javax.swing.JScrollPane();
        tblUnregistered = new javax.swing.JTable();
        jScrollPane2 = new javax.swing.JScrollPane();
        tblRegistered = new javax.swing.JTable();
        Devices = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        tblDevice = new javax.swing.JTable();
        jLabel5 = new javax.swing.JLabel();
        btnScan = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        jScrollPane4 = new javax.swing.JScrollPane();
        tblReg = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setMaximumSize(new java.awt.Dimension(214748000, 2147483647));

        tblUnregistered.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane3.setViewportView(tblUnregistered);

        tblRegistered.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane2.setViewportView(tblRegistered);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(205, 205, 205)
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(206, Short.MAX_VALUE)))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 21, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(7, 7, 7)
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
        );

        tblDevice.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        tblDevice.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tblDeviceMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(tblDevice);
        if (tblDevice.getColumnModel().getColumnCount() > 0) {
            tblDevice.getColumnModel().getColumn(0).setResizable(false);
            tblDevice.getColumnModel().getColumn(1).setResizable(false);
            tblDevice.getColumnModel().getColumn(2).setResizable(false);
            tblDevice.getColumnModel().getColumn(3).setResizable(false);
        }
```

```
jLabel5.setText("Connected Devices ");

btnScan.setText("Scan");
btnScan.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      btnScanActionPerformed(evt);
   }
});

jButton2.setText("Exit");
jButton2.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton2ActionPerformed(evt);
   }
});

jButton3.setText("add New");
jButton3.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton3ActionPerformed(evt);
   }
});

jButton1.setText("Registered");
jButton1.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton1ActionPerformed(evt);
   }
});

tblReg.setModel(new javax.swing.table.DefaultTableModel(
   new Object [][] {
      {null, null, null, null},
      {null, null, null, null},
      {null, null, null, null},
      {null, null, null, null}
   },
   new String [] {
      "Title 1", "Title 2", "Title 3", "Title 4"
```

```
            }
        });
        tblReg.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tblRegMouseClicked(evt);
            }
        });
        jScrollPane4.setViewportView(tblReg);
        if (tblReg.getColumnModel().getColumnCount() > 0) {
            tblReg.getColumnModel().getColumn(0).setResizable(false);
            tblReg.getColumnModel().getColumn(1).setResizable(false);
            tblReg.getColumnModel().getColumn(2).setResizable(false);
            tblReg.getColumnModel().getColumn(3).setResizable(false);
        }
        javax.swing.GroupLayout DevicesLayout = new javax.swing.GroupLayout(Devices);
        Devices.setLayout(DevicesLayout);
        DevicesLayout.setHorizontalGroup(
            DevicesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
DevicesLayout.createSequentialGroup()
                .addContainerGap(36, Short.MAX_VALUE)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 544,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 544,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(31, 31, 31))
            .addGroup(DevicesLayout.createSequentialGroup()

.addGroup(DevicesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                    .addGroup(DevicesLayout.createSequentialGroup()
                        .addGap(59, 59, 59)
                        .addComponent(jLabel5))
                    .addGroup(DevicesLayout.createSequentialGroup()
                        .addGap(147, 147, 147)
                        .addComponent(btnScan)
                        .addGap(55, 55, 55)
                        .addComponent(jButton3)
                        .addGap(46, 46, 46)
                .addComponent(jButton2)
                .addGap(27, 27, 27)
                .addComponent(jButton1)))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        DevicesLayout.setVerticalGroup(
          DevicesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(DevicesLayout.createSequentialGroup()
            .addGap(0, 23, Short.MAX_VALUE)
            .addComponent(jLabel5)
            .addGap(18, 18, 18)

.addGroup(DevicesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILIN
G)
              .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 333,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE, 333,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(0, 24, Short.MAX_VALUE)

.addGroup(DevicesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
              .addComponent(btnScan)
              .addComponent(jButton2)
              .addComponent(jButton3)
              .addComponent(jButton1))
            .addContainerGap())
        );
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
          .addComponent(jTabbedPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
          .addGap(0, 0, Short.MAX_VALUE)
          .addComponent(Devices, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGap(0, 0, Short.MAX_VALUE)))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addComponent(jTabbedPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 467,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0))
      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
          .addGap(0, 38, Short.MAX_VALUE)
          .addComponent(Devices, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGap(0, 38, Short.MAX_VALUE)))
    );
    pack();
  }// </editor-fold>

  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    addNew();
  }
  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
  }

  private void btnScanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
        syInfo.getCompInfo();
    }

    private void tblDeviceMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        try {
            row = tblDevice.getSelectedRow();
            String table_click = (tblDevice.getModel().getValueAt(row, 0).toString());

            String sql = "SELECT * FROM computers WHERE Serial_Number='" + table_click +
"'";
            pst = con.prepareStatement(sql);
            rs = pst.executeQuery();
            if (rs.next()) {
                String name = rs.getString("Name");
                // txtName.setText(name);
                String serial = rs.getString("Serial_Number");
                String ipout = rs.getString("IpAddress");
                String[] ipoutElements = ipout.split(",");
                List<String> fixedLengthList = Arrays.asList(ipoutElements);
                ArrayList<String> listofstring = new ArrayList<String>(fixedLengthList);
                String iparray[] = new String[listofstring.size()];
                for (int i = 0; i < listofstring.size(); i++) {
                    iparray[i] = listofstring.get(i);
                }
                String[] str = {iparray[0], null};
                System.out.println("elements array:" + str[0]);
                int result = JOptionPane.showConfirmDialog(this, "Are you sure you want to disable "
+ name + " ?", "swing tester",
                        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
                if (result == JOptionPane.YES_OPTION) {
                    String csv = "stopOne.csv";
                    syInfo.initCSV(csv);
                    try ( // boolean alreadyExist = new File(csv).exists();
                            CSVWriter writer = new CSVWriter(new FileWriter(csv, true), '\n') //append to
file
                            ) {
                        writer.writeNext(str);//change to ip
                        writer.flush();
                        //close the writer
                    }
                    syInfo.getStopOne();
                    //next remove disabled computer from the table
                    JOptionPane.showMessageDialog(this, "Device successfully disabled");
                    String st = "Disabled";
                    con = ConnectDB.connect();//create a connection
                    String sql2 = "UPDATE computers SET status=? WHERE  Name=?";

                    pst = con.prepareStatement(sql2);//prepare sql for execution
                    pst.setString(1, st);
                    pst.setString(2, name);
                    pst.executeUpdate();
                    data();

                } else if (result == JOptionPane.NO_OPTION) {
                    JOptionPane.showMessageDialog(this, "cancel");

                } else {
                    //none selected
                }

            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
       Register register = new Register();
       register.setVisible(true);

    }

    private void tblRegMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
```

```
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Display.class.getName()).log(java.util.logging.Level.SEVER
E, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Display.class.getName()).log(java.util.logging.Level.SEVER
E, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Display.class.getName()).log(java.util.logging.Level.SEVER
E, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Display.class.getName()).log(java.util.logging.Level.SEVER
E, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    new Display().setVisible(true);
                } catch (UnknownHostException ex) {
                    Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
                } catch (IOException ex) {
                    Logger.getLogger(Display.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JPanel Devices;
    private javax.swing.JButton btnScan;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JTabbedPane jTabbedPane2;
    private javax.swing.JTable tblDevice;
    private javax.swing.JTable tblReg;
    private javax.swing.JTable tblRegistered;
    private javax.swing.JTable tblUnregistered;
    // End of variables declaration
}
```

//System Information

```
import com.opencsv.CSVReader;
import com.opencsv.CSVWriter;
import com.profesorfalken.jpowershell.PowerShell;
import com.profesorfalken.jpowershell.PowerShellNotAvailableException;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Reader;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Beastly
 */
public class syInfo {

    /**
     * global variables
     *
     */
    //  private static final String SAMPLE_CSV_FILE_PATH = "./outfile.csv";
    public static String serialNumber;
    public static String compName;
    public static String operatingSys;
    public static String sysVersion;
    public static String macAddress;
    public static String ipAddress;
    public static String subnet;
    public static String architecture;
    static String[] ipoutput = new String[2];
    static boolean n;
    static int count = 0;
    static String ipoutput1;

    static Connection con = null;
    static ResultSet rs = null;
    static PreparedStatement pst = null;
    static Statement stmt = null;

    /**
     * Main Test Method for this class
     *
     */
    /**
     * public static void main(String[] args) throws IOException {
     * //getCompInfo(); //getAddress(); }*
     */
    /**
     * function to get the ip address of connected devices and the hostnames
     *
     * @throws java.net.UnknownHostException
     */
    public static void getAddress() throws UnknownHostException, IOException {

        InetAddress localhost = InetAddress.getLocalHost();
        byte[] ip = localhost.getAddress();
        //System.out.println(ip + " is the ip add");
        for (int i = 1; i <= 20; i++) {
            try {
                ip[3] = (byte) i;
                InetAddress address = InetAddress.getByAddress(ip);//local ip address(server ip)
                ipoutput1 = address.toString().substring(1);
                // System.out.println("this is the address : "+address);
                //sendPingRequest(ipoutput1);

                if (address.isReachable(1000)) {
                    System.out.println(ipoutput1 + " is on the network");

                     ipoutput[count] = ipoutput1;
                   //System.out.println("count is at: " + count);
                   // System.out.println("this is final add: " + ipoutput[count]);
            if (ipoutput[count] != null ||ipoutput[count] != "") {
              System.out.println(ipoutput[count] +" is the ip array");
              toCSVFile();//check function no update to db
            }
            count++;
            // System.out.println("count is at: " + count);

            /**
             * save the values to the database
             *
             */
          }
        } catch (Exception er) {

        }

      }

    }

    public static void sendPingRequest(String ipad)
        throws UnknownHostException, IOException {
      InetAddress geek = InetAddress.getByName(ipad);
      System.out.println("Sending Ping Request to " + ipad);
      if (geek.isReachable(1000)) {
        n = true;
        System.out.println(n);
        count = 0;
        ipoutput[count] = ipoutput1;

        //System.out.println(ipoutput[count]);
        count++;

      } else {
        n = false;
      }

    }

    public static void initCSV(String csvName) throws IOException {
        // String csv = "infile1.csv";
        String csv = csvName;

        CSVWriter writer = new CSVWriter(new FileWriter(csv));

        String[] st = new String[]{"ComputerName"};
        writer.writeNext(st);

        writer.flush();
    }

    public static void toCSVFile() throws IOException {
        String csv = "infile1.csv";

                try ( // boolean alreadyExist = new File(csv).exists();
                CSVWriter writer = new CSVWriter(new FileWriter(csv, true),'\n') //append to file
                ) {
            writer.writeNext(ipoutput);
            writer.flush();
            //close the writer
        }
    }

    /**
     * this method executes a powershell script to get computer info and saves
     * it to an array file
     *
     */
    public static void getCompInfo() {
        try {

            PowerShell powershell = null;
            powershell = PowerShell.openSession();


powershell.executeScript("C:\\Users\\JK\\Documents\\NetBeansProjects\\appServ\\script2.ps1");
            //System.out.println(powershell.executeScript("./script2.ps1").getCommandOutput());

            try {
```

```
            /**
             * read all rows at once
             *
             */
            //Build reader instance
            CSVReader reader = new CSVReader(new
FileReader("C:\\Users\\Admin\\Documents\\NetBeansProjects\\appServ\\outfile.csv"), ',', '"', 1);

            //Read all rows at once
            List<String[]> allRows = reader.readAll();

            //Read CSV line by line and use the string array as you want
            for (String[] row : allRows) {
                System.out.println(Arrays.toString(row));

                compName = row[6];
                serialNumber = row[5];
                ipAddress = row[7];
                sysVersion = row[2];
                operatingSys = row[4];
                subnet = row[0];
                macAddress = row[3];
                architecture = row[1];

                try {
                    String st = "Active";
                    con = ConnectDB.connect();//create a connection
                    String sql = "INSERT INTO computers
(Serial_Number,Name,IpAddress,status)values('" + serialNumber + "','" + compName + "','" +
ipAddress + "','" + st + "')";
                    pst = con.prepareStatement(sql);//prepare sql for execution
                    pst.execute();
                    System.out.println("saved succeed");
                    JOptionPane.showMessageDialog(null,"Successfully
saved","Record",JOptionPane.INFORMATION_MESSAGE);
                } catch (SQLException e) {
                   // System.out.println(e.printStackTrace());
                   e.printStackTrace();
                }
            }
            /**
             * //Build reader instance //Read data.csv //Default seperator
             * is comma //Default quote character is double quote //Start
             * reading from line number 2 (line numbers start from zero)
             * CSVReader reader1 = new CSVReader(new
             * FileReader("outfile.csv"), ',', '"', 1);
             *
             * //Read CSV line by line and use the string array as you want
             * String[] nextLine; while ((nextLine = reader1.readNext()) !=
             * null) { if (nextLine != null) { //Verifying the read data
             * here //System.out.println(Arrays.toString(nextLine));
             * compName = nextLine[6]; serialNumber = nextLine[5]; ipAddress
             * = nextLine[7]; sysVersion = nextLine[2]; operatingSys =
             * nextLine[4]; subnet = nextLine[0]; macAddress = nextLine[3];
             * architecture = nextLine[1];
             *
             * }
             * }*
             */
        } catch (Exception e) {
            System.out.println("error: " + e);
            e.printStackTrace();
        }

    } catch (PowerShellNotAvailableException e) {
        System.out.println(e);
        e.printStackTrace();
    }

}

public static void getStop() throws SQLException {
    try {
        PowerShell powershell = null;
        powershell = PowerShell.openSession();

        // powershell.executeScript(ipoutput, address)

powershell.executeScript("C:\\Users\\Admin\\Documents\\NetBeansProjects\\appServ\\stopScrip
t.ps1");
```

```
        } catch (PowerShellNotAvailableException e) {
            JOptionPane.showMessageDialog(null, "powershell error" + e);
            throw new RuntimeException("Failed to execute Powershell(getStop)", e);
        }
    }

    public static void getUStop() throws SQLException {
        try {
            PowerShell powershell = null;
            powershell = PowerShell.openSession();

powershell.executeScript("C:\\Users\\Admin\\Documents\\NetBeansProjects\\appServ\\stopUScr
ipt.ps1");
        } catch (PowerShellNotAvailableException e) {
            JOptionPane.showMessageDialog(null, "powershell error" + e);
            throw new RuntimeException("Failed to execute Powershell(getUStop)", e);
           // e.printStackTrace();

        }
    }

    public static void getStopOne() {
        try {
            PowerShell powershell = null;
            powershell = PowerShell.openSession();

powershell.executeScript("C:\\Users\\Admin\\Documents\\NetBeansProjects\\appServ\\stopOneS
cript.ps1");
         } catch (PowerShellNotAvailableException e) {
            JOptionPane.showMessageDialog(null, "powershell error" + e);
            throw new RuntimeException("Failed to execute Powershell(getStopOne)", e);
        }
    }
}
```

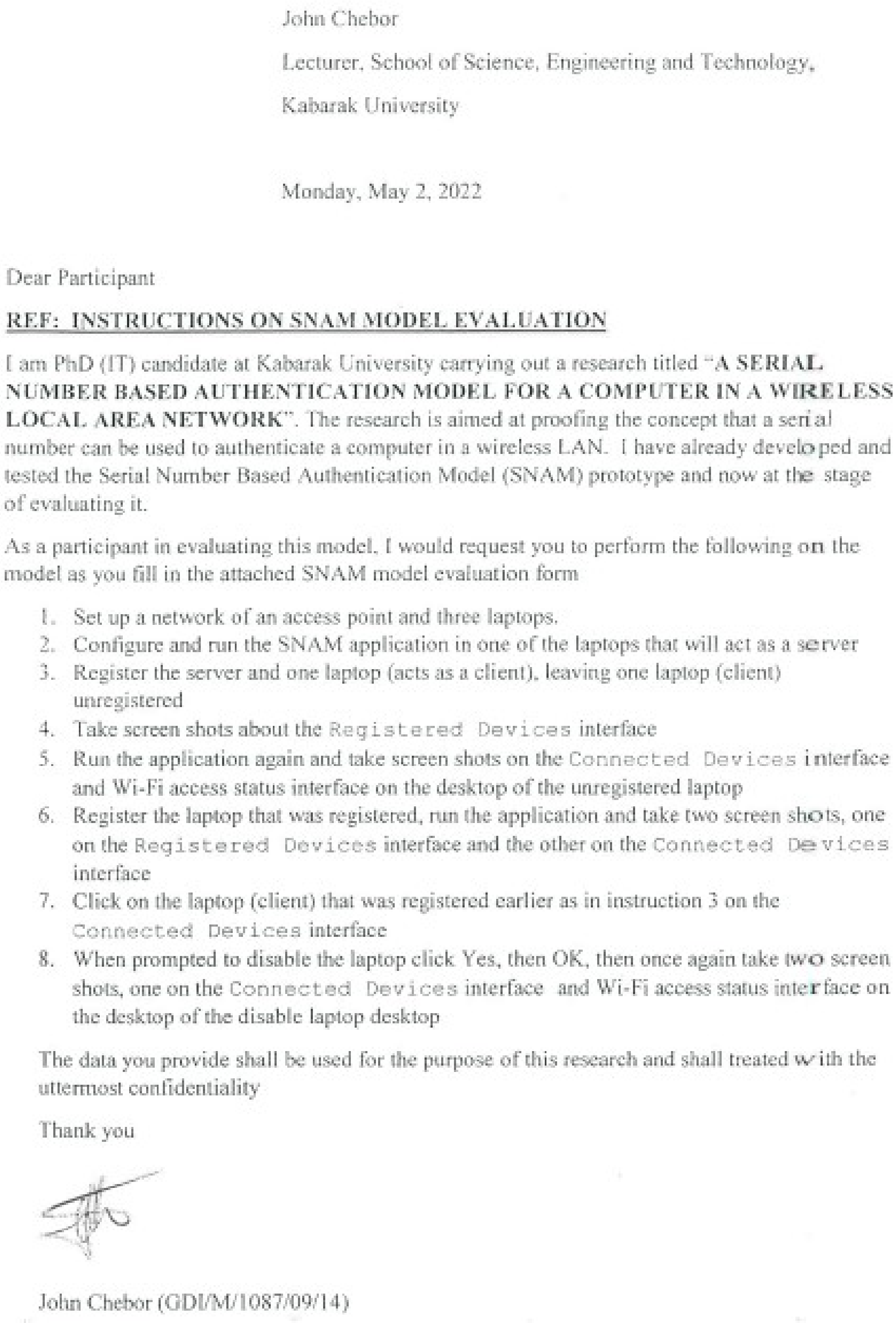
APPENDIX V

Pilot Test Code

```
import java.io.File;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class MiscUtils
{
private MiscUtils() { }

public static String getMotherboardSN()
{
String result = "";
try
{
File file = File.createTempFile("realhowto",".vbs");
file.deleteOnExit();
FileWriter fw = new java.io.FileWriter(file);

String vbs ="Set objWMIService = GetObject(\"winmgmts:\\\\.\\root\\cimv2\")\n"
+ "Set colItems = objWMIService.ExecQuery _ \n"
+ " (\"Select * from Win32_BaseBoard\") \n"
+ "For Each objItem in colItems \n"
+ " Wscript.Echo objItem.SerialNumber \n"
+ " exit for ' do the first cpu only! \n"
+ "Next \n";

fw.write(vbs);
fw.close();
Process p = Runtime.getRuntime().exec("cscript //NoLogo " + file.getPath());
BufferedReader input =
new BufferedReader
(new InputStreamReader(p.getInputStream()));
String line;
while ((line = input.readLine()) != null) {
result += line;
}
input.close();
}
catch(Exception e){
e.printStackTrace();
}
return result.trim();
}
          public static void main(String[] args){
            String cpuId = MiscUtils.getMotherboardSN();
            javax.swing.JOptionPane.showConfirmDialog((java.awt.Component)
                null, cpuId, "Motherboard serial number",
                javax.swing.JOptionPane.DEFAULT_OPTION);
          }
                                                  }
```

APPENDIX VI

Instructions on SNAM Model Evaluation Letter



John Chebor

Lecturer, School of Science, Engineering and Technology,

Kabarak University

Monday, May 2, 2022

Dear Participant

REF: INSTRUCTIONS ON SNAM MODEL EVALUATION

I am PhD (IT) candidate at Kabarak University carrying out a research titled "**A SERIAL NUMBER BASED AUTHENTICATION MODEL FOR A COMPUTER IN A WIRELESS LOCAL AREA NETWORK**". The research is aimed at proofing the concept that a serial number can be used to authenticate a computer in a wireless LAN. I have already developed and tested the Serial Number Based Authentication Model (SNAM) prototype and now at the stage of evaluating it.

As a participant in evaluating this model, I would request you to perform the following on the model as you fill in the attached SNAM model evaluation form

1. Set up a network of an access point and three laptops.
2. Configure and run the SNAM application in one of the laptops that will act as a server
3. Register the server and one laptop (acts as a client), leaving one laptop (client) unregistered
4. Take screen shots about the `Registered Devices` interface
5. Run the application again and take screen shots on the `Connected Devices` interface and Wi-Fi access status interface on the desktop of the unregistered laptop
6. Register the laptop that was registered, run the application and take two screen shots, one on the `Registered Devices` interface and the other on the `Connected Devices` interface
7. Click on the laptop (client) that was registered earlier as in instruction 3 on the `Connected Devices` interface
8. When prompted to disable the laptop click Yes, then OK, then once again take two screen shots, one on the `Connected Devices` interface and Wi-Fi access status interface on the desktop of the disable laptop desktop

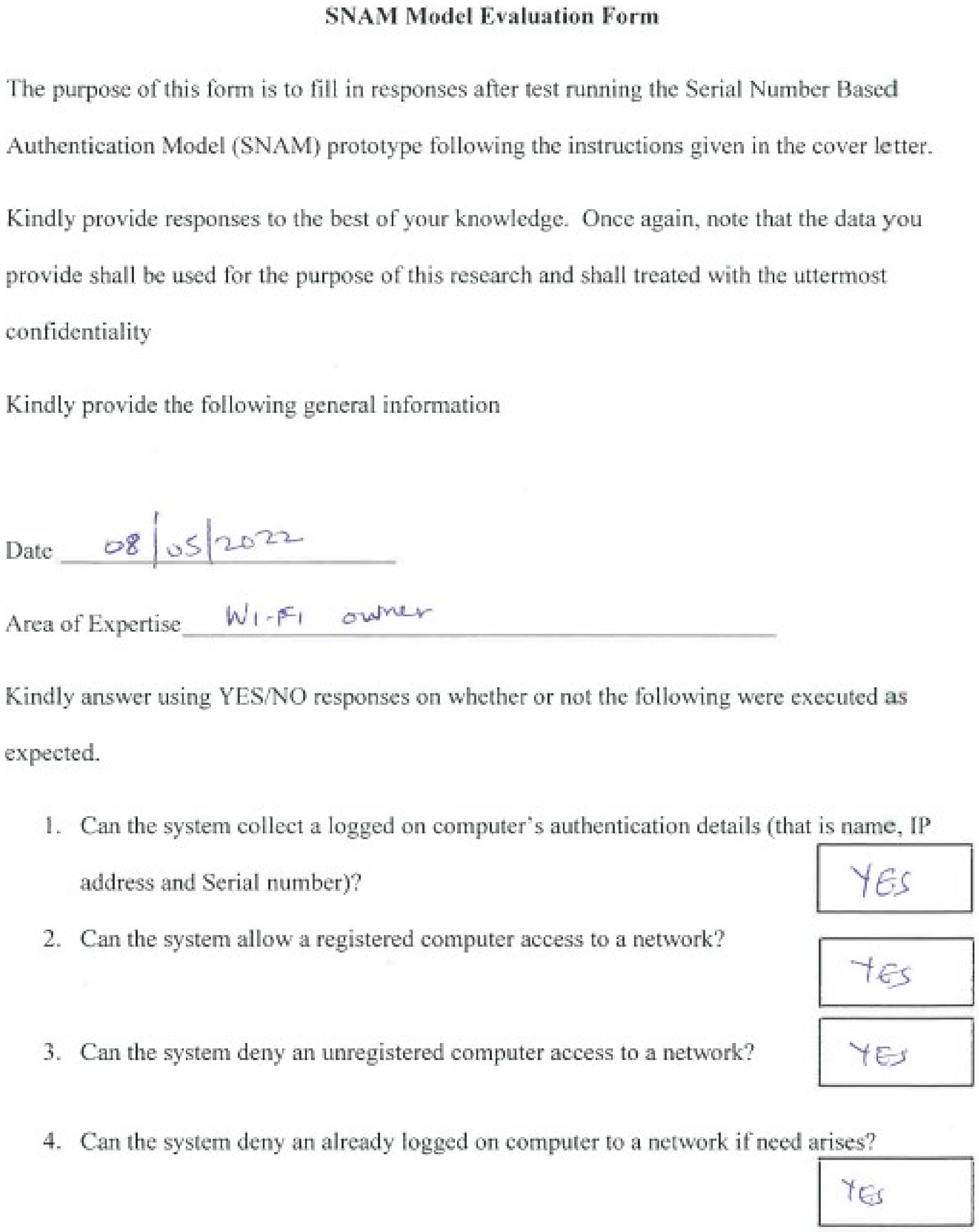
The data you provide shall be used for the purpose of this research and shall treated with the uttermost confidentiality

Thank you

John Chebor (GDI/M/1087/09/14)

APPENDIX VII

Evaluator 1 Feedback



SNAM Model Evaluation Form

The purpose of this form is to fill in responses after test running the Serial Number Based Authentication Model (SNAM) prototype following the instructions given in the cover letter.

Kindly provide responses to the best of your knowledge. Once again, note that the data you provide shall be used for the purpose of this research and shall treated with the uttermost confidentiality

Kindly provide the following general information

Date 08/05/2022

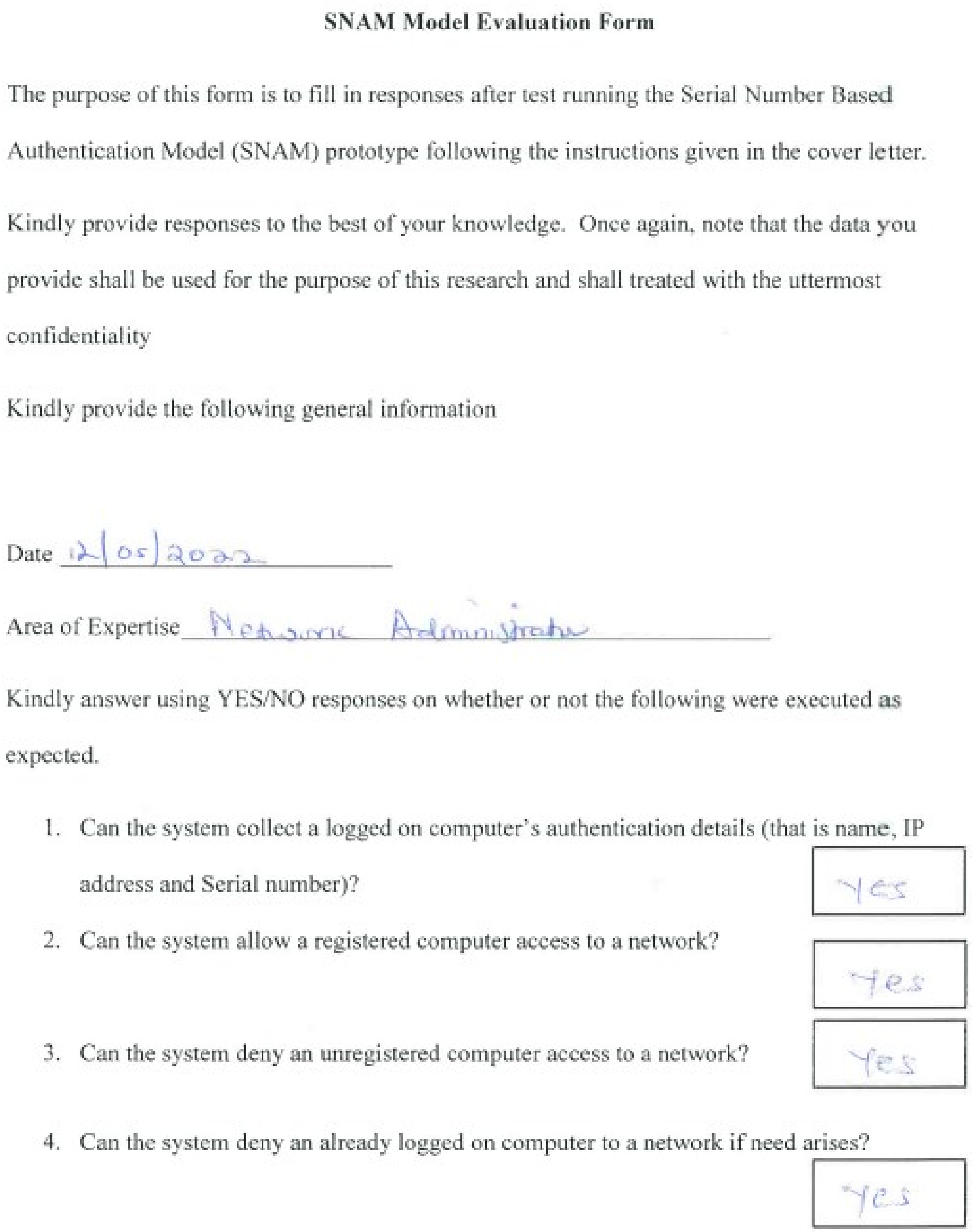
Area of Expertise WI-FI owner

Kindly answer using YES/NO responses on whether or not the following were executed as expected.

1. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)? YES
2. Can the system allow a registered computer access to a network? YES
3. Can the system deny an unregistered computer access to a network? YES
4. Can the system deny an already logged on computer to a network if need arises? YES

APPENDIX VIII

Evaluator 2 Feedback



SNAM Model Evaluation Form

The purpose of this form is to fill in responses after test running the Serial Number Based Authentication Model (SNAM) prototype following the instructions given in the cover letter.

Kindly provide responses to the best of your knowledge. Once again, note that the data you provide shall be used for the purpose of this research and shall treated with the uttermost confidentiality

Kindly provide the following general information

Date 12/05/2022

Area of Expertise Network Administrator

Kindly answer using YES/NO responses on whether or not the following were executed as expected.

1. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?

   Yes

2. Can the system allow a registered computer access to a network?

   Yes

3. Can the system deny an unregistered computer access to a network?

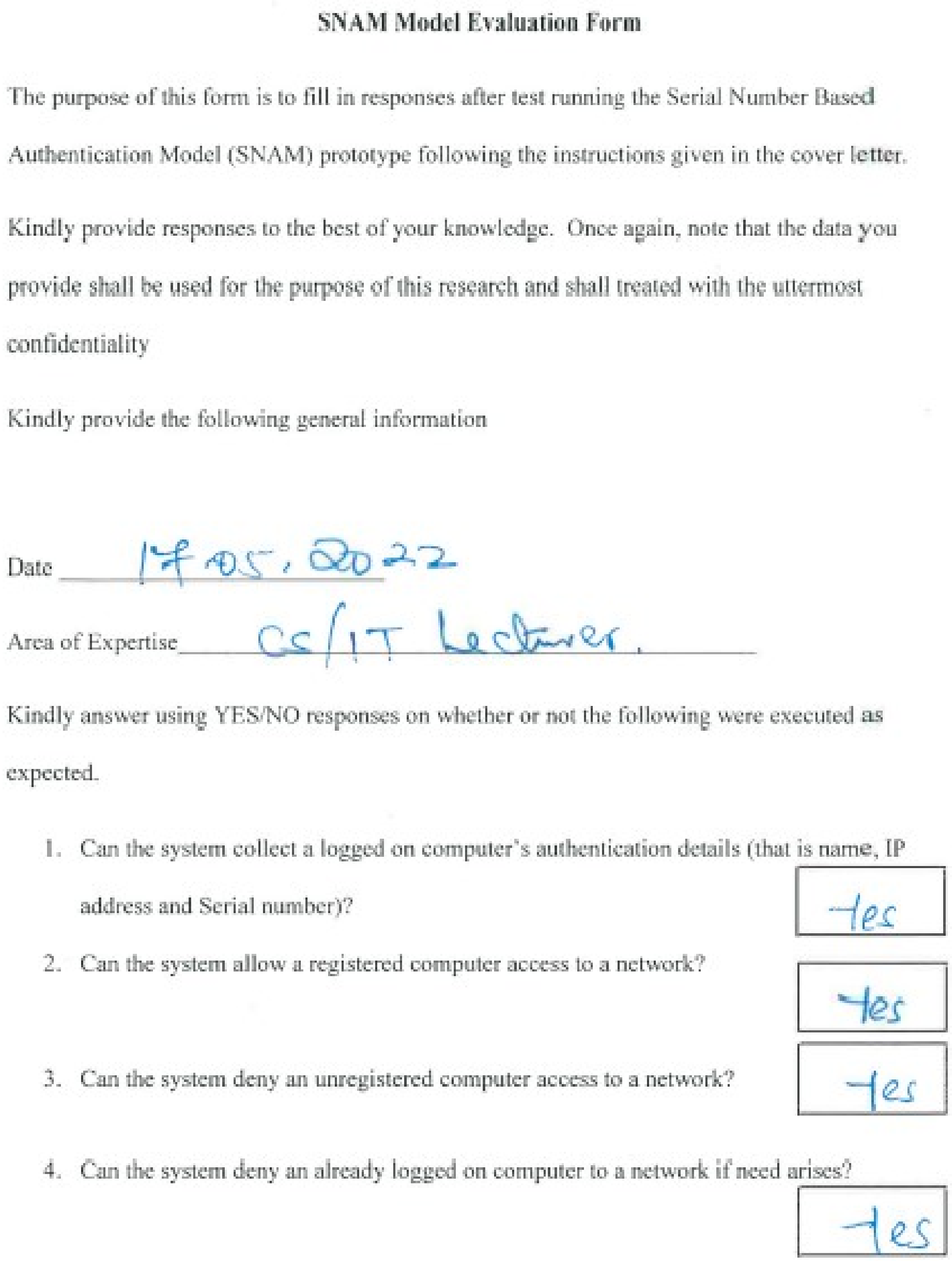
   Yes

4. Can the system deny an already logged on computer to a network if need arises?

   Yes

APPENDIX IX

Evaluator 3 Feedback



SNAM Model Evaluation Form

The purpose of this form is to fill in responses after test running the Serial Number Based Authentication Model (SNAM) prototype following the instructions given in the cover letter.

Kindly provide responses to the best of your knowledge. Once again, note that the data you provide shall be used for the purpose of this research and shall treated with the uttermost confidentiality

Kindly provide the following general information

Date 17 05. 2022

Area of Expertise CS/IT Lecturer.

Kindly answer using YES/NO responses on whether or not the following were executed as expected.

1. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)? Yes
2. Can the system allow a registered computer access to a network? Yes
3. Can the system deny an unregistered computer access to a network? Yes
4. Can the system deny an already logged on computer to a network if need arises? Yes

APPENDIX X

Conference Paper 1

Proceedings of the Kabarak University International Conference on Computing and Information Systems. 8th – 9th October 2018 Nakuru, Kenya

A Serial Number Based Identification Model for a Computer in a Wireless Local Area Network

John C. CHEBOR[1], Simeon M KARUME[2,] and Nickson M KARIE[3]

[1]*Kabarak University, P.O. Box Private Bag, Kabarak, 20157, Kenya*
Tel: +254 0721416894, Email: jchebor@kabarak.ac.ke

[2]*Laikipia University, P.O. Box 1100-20300, Nyahururu, Kenya*
Tel: +254 0722499397, Email: smkarume@gmail.com

[3]*, Private Bag No.4, Kwaluseni-M201 Swaziland*
Tel: +26876534638, Email:nickson.karie@gmail.com

Abstract: With today's technological evolution, wireless networks have become very common for organizations, homes and public places. Besides, wireless devices seem to fill our daily lives with wireless "hotspots" emerging almost everywhere both in offices, airports, cyber cafes, sports venues and even in coffee shops. For any device to be authenticated and authorized to use any of the wireless network services, it must first be identified. Once a device has been identified, it may then be authenticated and authorized to have access to the wireless network resources. One of the biggest challenges with implementing wireless networks, though, is implementing the identification of wireless devices. Apart from application specific addresses, port numbers and internet protocol (IP) addresses at application, transport and network layers respectively, devices in a network use media access control (MAC) addresses for identification at the data link layer. However, MAC addresses can be spoofed and altered thereby compromising the security and robustness and uniqueness qualities of the device identifier. On top of security and robustness issues, a network device as well could contain more than one network interfaces that results to number of MAC addresses for the same device, rendering the MAC address not unique. The research, therefore, was aimed at investigating how a computer's serial number may be used for physical identification of a computer in a wireless local area network (LAN). In order to achieve the research objective, the study examined the inbuilt access and use of a serial number prototype model as an alternative method of identifying devices in a network. The model was designed on a mixed research method using evolutionary prototyping and proof of concept methods through test runs and was found to actually identify a device in a network based on a computer's serial number. This was realized by developing a Serial Number Based Identification (SNAM) prototype on a Java Development Kit (JDK) and MySQL platforms. The SNAM model first collects and processes the computer's name and IP address then collects the computer's serial number and uses the IP address to process the computer's serial number for computer identification. Furthermore, the test runs indicate that a MAC address can actually be spoofed and altered rendering the MAC address not unique, insecure and unreliable. This was as a result of the fact that a computer's MAC address, apart from it being hard-coded in the hardware, it has a copy of the MAC address in the system software. On the contrary, a computer's serial number is hard-coded in the hardware only and therefore cannot be spoofed and altered making it unique, secure and reliable. The researcher recommends that the prototype be scaled up, then adopted as a network device identification method

Key Words: Computer, Serial Number, MAC address, Identifier, Identification, Serial Number-Based Identification, Wireless Local Area Network

APPENDIX XI

Conference Paper 2

Conference Proceedings of The Kabarak University International Conference on Computing and Information Systems, Edited by Christopher M. Maghanga and Moses Thiga. 14th-15th October 2019 Nakuru, Kenya. 28 | P a g e Available online http://ir.kabarak.ac.ke/handle/123456789/1

Towards a Unique, Secure, and Robust Wireless Local Area Network Device Identifier

John C. Chebor[1], Simeon M. Karume[2] and Nelson B. Masese[3]

[1]Kabarak University, P.O. Box Private Bag, Kabarak, 20157, Kenya Tel: +254 0721416894, Email: jchebor@kabarak.ac.ke
[2]Laikipia University, P.O. Box 1100-20300, Nyahururu, Kenya Tel: +254 0722499397, Email: smkarume@gmail.com
[3]Kabarak University, P.O. Box Private Bag, Kabarak, 20157, Kenya Tel: +254 0727171725, Email: NMasese@kabarak.ac.ke

Abstract

With today's technological evolution, wireless networks have become very common for organizations, homes and public places. For any device to be authenticated and authorized to use any of the wireless network services, it must first be identified then authenticated and authorized to have access to the wireless network resources. One of the biggest challenges with implementing wireless networks, though, is implementing the identification of the wireless devices. This study therefore examined uniqueness, security and robustness characteristics of MAC in relation to a device serial number in order to establish a suitable network device identifier. In order to achieve this, test runs through a proof of concept method by using Advanced IP Scanner and getmac command line tools. Advanced IP Scanner was used to determine the security, hence robustness of the identifiers while getmac was used to determine the uniqueness of the identifiers. The run tests indicated that a MAC address can actually be spoofed and altered rendering the MAC address not unique, insecure and unreliable. On the contrary, a computer's serial number is hard-coded in the hardware only and therefore cannot be spoofed and altered making it unique, secure and reliable. The researcher recommends that a study be conducted on how a device serial number can be used as network device identifier Key Words: Network device, MAC Address, Serial Number, Identifiers, Wireless Local Area Network

Key Words: Network device, MAC Address, Serial Number, Identifiers, Wireless Local Area Network

1. Introduction

Wireless LANs (WLAN) also known as Wireless Fidelity (Wi-Fi) or 802.11 standards is a type of a local area network that allows users access network services using mobile devices (wireless stations) such as laptops, personal digital assistants (PDA) and even smartphones [1]. The wireless stations use a base station usually an access point (AP) as an entry point to the network services. Unlike wired LANs that use cables or wires as transmission media, WLANs uses radio wave frequencies to transmit signals over the local area network.

WLAN comes with a number of benefits as compared to wired LANs, notably mobility, rapid deployment, reduction in infrastructure and operational cost, flexibility, and scalability [2, 3, 4]. Due to these benefits hotpots are now virtually found everywhere; in enterprises, at homes, and in public places. Wireless devices such as laptops, personal digital assistants and even smartphones come with WiFi features integrated into them. To cope up with large number of device usage, enterprises have ended up adopting the bring your own device (BYOD) to allow employees and other stakeholders such as vendors, visitors, customers and contractors access and use the network [5]. Rise in internet of things (IOT) devices such smart devices, smart watches and smart phones [6]. Despite the numerous benefits that come with wireless LANs, network insecurity has become a thorn in the flesh for proprietors of such kind of networks. [7], point out that an attacker simply needs to be within the range of the WLAN access point to intrude into the network as opposed to wired LANs where an attacker requires physical access to the LAN or remotely use porous firewall systems to gain access to the LAN. WLANs are therefore easily targeted by attackers through spoofing, denial of service, eavesdropping, man-in-the-middle, masquerading, message modification, message replay and traffic analysis [8, 5, 3].

Allowing users to connect to the network with their own devices can pause as a security challenge as it becomes difficult for network administrators to control such kind of a network access and usage. It is therefore imperative that network administrators use network access control tools to control who should and who should not access the network. One of such kind of a control tool is the network access control (NAC) [9, 6, 10]. NAC constitutes identification, authentication, authorization, and accounting (IAAA), in that order, according to [11], as the essential functions in providing the required services in a network. For authentication and authorization hence accounting to be realized, devices in a network must first be identified. According to [12], devices in a network can only be explicitly identified by their port numbers, IP address, and MAC address. But whereas MAC addresses are used by messages to identify actual physical destination and source network addresses, IP addresses are used by information to be effectively routed from network to network in an internetwork. On the other hand, port numbers are used specifically to identify processes running on sending and receiving stations [13].

Failure to adequately identify a device in a WLAN can lead progressively to ambiguity in authentication, authorization and subsequently accounting of network resource usage.

This paper, therefore, is aimed at investigating the suitability of a MAC address in relation to a computer's serial number as a wireless LAN device identifier. In order to achieve this goal, the study first establishes desirable characteristics of a network device identifier, then analyses the suitability of a MAC address in relation to a computer's serial number network device identifiers and finally recommends a suitable WLAN device identifier

2. Related Work

An identifier, according to [14], is an attribute (or a combination of attributes) whose value distinguishes instances of an entity type (device) from another. [14] further cites examples of identifiers as could be a code (identification number, serial number, ISBN), a name (domain name) or an address (IP, MAC, Port Number or an application specific address). An attribute should possess uniqueness, universality, collectability, security, data dependence, robustness and mnemonic [15, 16, 17] qualities to be a good identifier. Whereas uniqueness ensures that no two devices have the same identifier value, universality ensures that devices in the same space have an identifier, collectability is the ability of an identifier to be captured from existing systems, security ensures availability, integrity and confidentiality of an identifier, data dependence is the ability of an identifier to be associated with other device attributes, robustness or reliability or permanence is the ability of an identifier not to vary with time and mnemonic defines a standard and meaningful structure of the identifier.

Application specific addresses are addresses that are designed for a specific application geared towards user-friendliness. Also referred to as persistent identifiers [18], the application specifies identifier is permanently assigned to an object. Examples of application specific addresses include e-mail address and a universal resource locator (URL). Whereas an e-mail address defines the recipient of an e-mail, a URL is used to find a document on the internet. Such addresses or locators fundamentally play a crucial role in enabling internet users easily finds information on the internet. This is more so as the internet has a huge amount of information which makes it difficult to find. Labelling the files or objects in a way of application-specific addresses, therefore, makes it easy to find a specific object or file. The addresses, however, get changed to the corresponding port and MAC addresses of the sending computer.

Port numbers are numbers on hosts/devices that identify sending and receiving processes. According to [19], port numbers are usually captured when a system requests for a page with usernames and password details. Process identification in communication is crucial because a host could be running several applications. Although port numbers can be used in conjunction with other numbers, namely, IP address and MAC address to identify a host, port numbers can pause as a threat to network security. Intruders can use a port scanner that listens to well-known port numbers so as to detect services running on a system, therefore can easily break into the system [20].

An IP address, on the hand, is number assigned to a host or a router in the internet for identification and location of the device as stated by [21]. An IPv4, for instance, [13], is composed of four dotted decimal notations (example 243.246.0.28) each part range between 0 and 255. Depending on the class or the subnet mask of the IP address, some parts (first) belong to network identification and the other parts (usually from the second part) belong to host identification. IPv4 was designed to use 32-bit address space [22]. This translates to 2^{32} or approximately four (4) billion addresses which is not sufficient for the global needs. Actually, the numbers were projected to have been exhausted by 2013 [23]. A temporary solution of conserving IP address used by network administrators in apart from organizational LANs, has also found use in residential and wireless LANs, is the use of dynamic host configuration protocol (DHCP) [13]. DHCP assigns devices IP addresses on demand, therefore two devices can use the same IP address but on different occasions.

MAC address also known as LAN address or a physical address is a number used to identify a network adaptor on a LAN. As [24] puts it "it is not hosts and routers that have link-layer addresses but rather their adapters (that is, network interfaces) that have link-layer addresses." In other words, a MAC address is used not only by devices but also by information to identify a particular physical network like an Ethernet network. A device with multiple interfaces has multiple corresponding MAC addresses. Furthermore, the possibility of a MAC address being spoofed renders it not unique, variant and therefore unreliable.

Figure 1 below illustrates the four identifiers each at corresponding OSI reference layer. Its worthy to note from the figure that the OSI reference physical layer does not have an identifier.

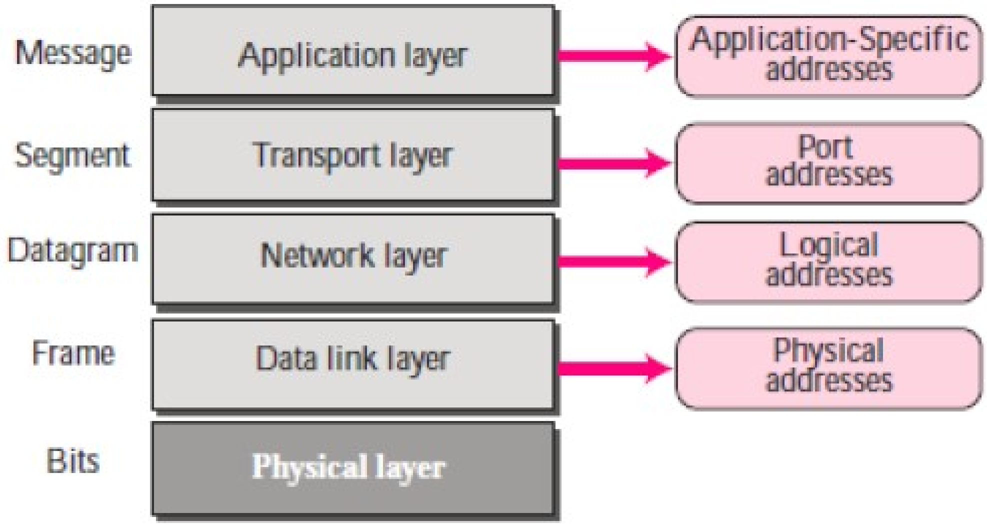


Figure 1: OSI network device address layering (Source: [25])

The other identifier that is of concern in this study is the computers serial number. Also referred to as the manufacture's serial number (MSN), a device serial number is number assigned by a manufacturer to a device for identification [26]. Older computer models, for instance, had their serial numbers tagged strategically either underneath or on the side to frustrate snoopers from getting it. Apart from tagging the serial numbers, modern computer models have their serial numbers hard-coded into the basic-input output (BIOS) chip on the system board [27]. This makes it possible for the identifier to internally be accessed (figure 8) by a program or a tool, and so, it can be processed for a given desired function.

The fact that a computer's serial number is hard-coded on the hardware without a copy in the system software alone renders it hard to be spoofed (figure 3). It then implies that the serial number in normal circumstances cannot be altered and therefore unique, secure and reliable. It is only in some rare cases that the serial number can be altered. But this requires that a computer system has to turned off, any power lines disconnected, any static electricity discharged, computer case opened, disconnect the CMOS battery, wait for roughly 30 seconds (to completely ensure that the CMOS power is completely drained) then the process is done in reverse to revert back to original state [28]. This way, all original CMOS settings such as custom CMOS settings, BIOS password, time and date, as well as the motherboard serial number are lost. The system then generates a new system data that includes a serial number when booted.

Apart from uniquely identifying a device, a serial number is suitable for device tracking, deterring device theft and counterfeiting, device controls and activating aliases [26]. However, serial numbers vary in formats according to their manufactures. Dell machines, for example, uses seven characters and HP machines use ten characters to express their serial numbers [27].

The table 1 below gives a summary of the application-specific, port number, IP address, MAC address and serial number identifiers. The summary is based on the identifier's key functions, the OSI layers they operate at, where it is located in the system and attributes (uniqueness, universality, collectability, security, data dependence, robustness and mnemonic).

Table 1: Device identifiers compared

Identifier/ Characteristic	**Application-specific address**	**Port Number**	**IP Address**	**MAC Address**	**Serial Number**
Key Function	Identifies a specific application	Identifies a host process	Identifies a device in the internet	Identifies a device in a network	Actual device identifier
OSI Layer	Application layer	Transport layer	Network layer	Data link layer	Physical layer
Location	Within an application	Within an application	In the system software	In the system logic board and a copy in the system software	In the system logic board
Availability	Accessible and usable	Accessible and usable	Accessible and usable	Accessible and usable	Accessible and usable
Integrity	Can be modified	Can be modified	Can be modified	Can be modified	Cannot be modified
Confidentiality	Can be modified	Can be modified	Can be modified	Can be modified	Cannot be modified
Robustness	Can change over time	Can change over time	Can change over time	Can change over time	Cannot change over time

Uniqueness	Unique to an application	Unique to an application	Unique to the Internet	Unique to a LAN interface	Unique to a device
Universality	Universal	Universal	Universal	Universal	Universal
Collectability	Collectable	Collectable	Collectable	Collectable	Collectable
Data dependent	Data dependent	Data dependent	Data dependent	Data dependent	Data dependent
Mnemonic	Mnemonic	Mnemonic	Mnemonic	Mnemonic	Not mnemonic

3. Key Findings and Discussions

3.1 Characteristics of a Network Device Identifier

The characteristics that define the suitability of an identifier as were established from the related work section of this study are uniqueness, universality, collectability, data dependent, security (availability, integrity, and confidentiality), robustness and mnemonics, best illustrated in figure 2 below

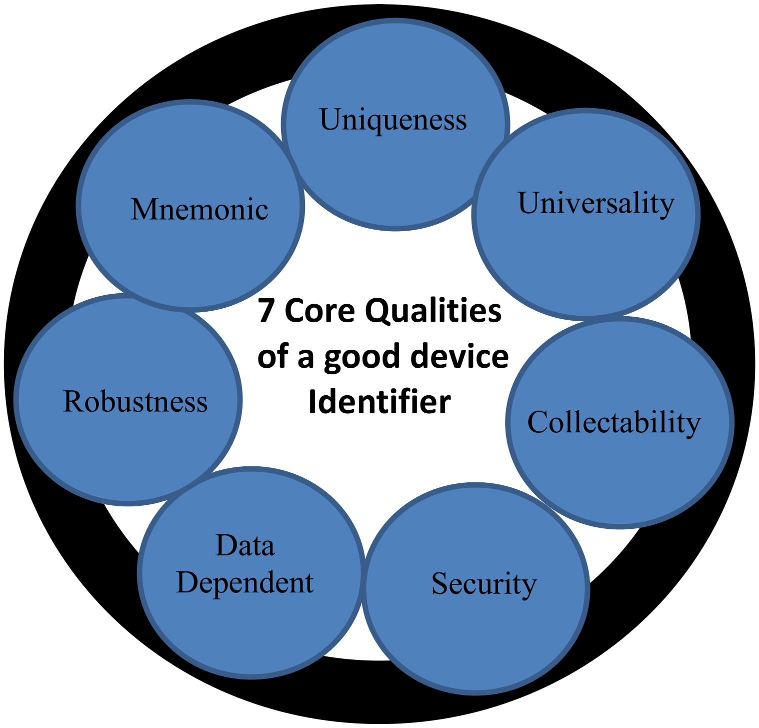


Figure 2: Seven core qualities of a good device identifier

As compared to the other identifier characteristics, security, robustness and uniqueness characteristics were established to compromise the suitability of MAC address as an identifier as indicated in the table 1 above. The three requirements were examined as in the next section using an `Advanced IP scanner` and `wmic` Windows tools in a Windows Operating system environment

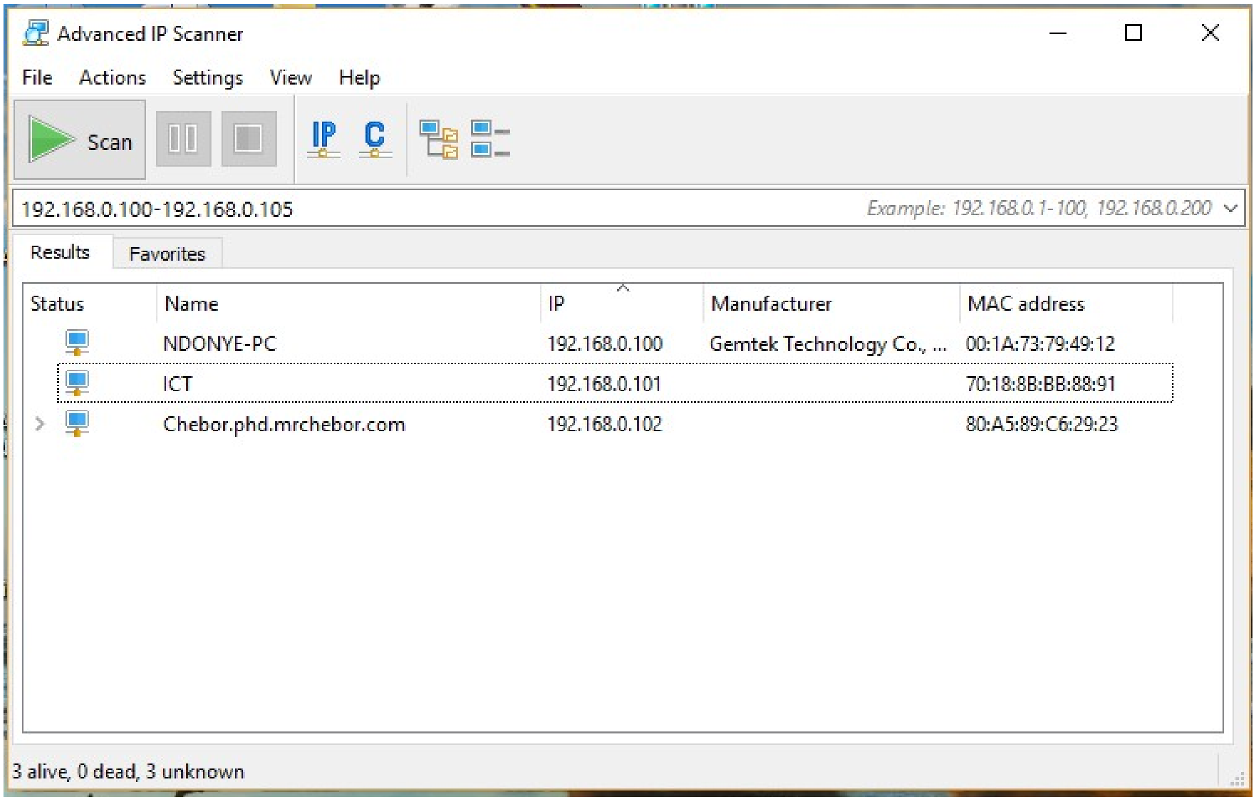
3.2 Suitability of MAC Address as Network Device Identifier

3.2.1 Security of a MAC Address

Availability, confidentiality and integrity aspects of security determine the suitability of a MAC address.

Confidentiality of a MAC Address

Confidentiality of an identifier is the degree to which an identifier can be disclosed to an unauthorized entity [29]. Although measures have been put in place to protect the confidentiality of a MAC address by coding it into the network hardware, as a network device identifier, attackers would always have a way of getting it without authorization. An Advanced IP scanner tool, for instance, can be used by an intruder to access MAC address of all devices connected to the network as demonstrated in in figure 3 below



192.168.0.100-192.168.0.105

Status	Name	IP	Manufacturer	MAC address
	NDONYE-PC	192.168.0.100	Gemtek Technology Co., ...	00:1A:73:79:49:12
	ICT	192.168.0.101		70:18:8B:BB:88:91
>	Chebor.phd.mrchebor.com	192.168.0.102		80:A5:89:C6:29:23

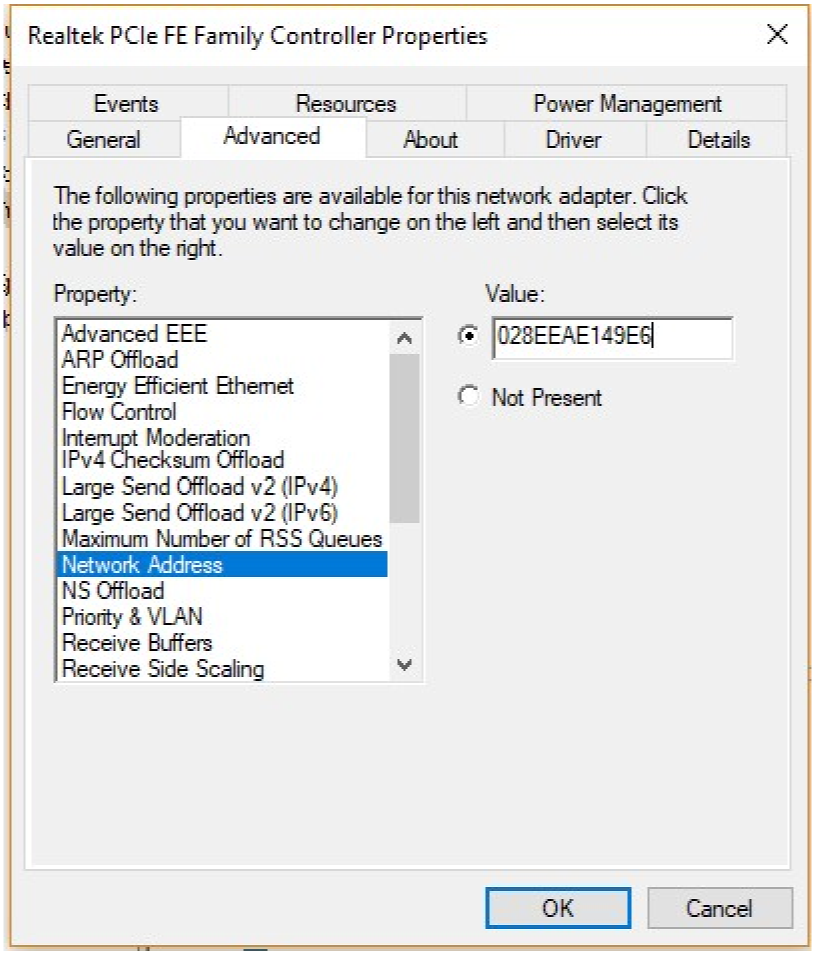
3 alive, 0 dead, 3 unknown

Figure 3: MAC Address scanning using `Advanced IP Scanner`

As demonstrated in the results in figure 3 above, the scanner collected all the MAC addresses of the networked computers. There exist other spoofing tools apart from `Advanced IP Scanner` such as `IP Scanner Pro 10`, `PCFinder`, `Angry IP Scanner`, `MAC Address Scanner`, `Colasoft MAC Scanner` and `Ipscan` [30], that can be used to spoof IP and MAC addresses. Either way, the demonstration indicates that a MAC address is not confidential.

The integrity of a MAC Address

A MAC address is usually hard-coded or 'burned' into the network hardware; therefore, it is difficult to alter it in normal cases. However, a copy of the MAC address in the system software, as illustrated in figure 4 below, can easily be modified by an attacker to suit the valid MAC addresses spoofed.



Realtek PCIe FE Family Controller Properties

The following properties are available for this network adapter. Click the property that you want to change on the left and then select its value on the right.

Property:

- Advanced EEE
- ARP Offload
- Energy Efficient Ethernet
- Flow Control
- Interrupt Moderation
- IPv4 Checksum Offload
- Large Send Offload v2 (IPv4)
- Large Send Offload v2 (IPv6)
- Maximum Number of RSS Queues
- Network Address
- NS Offload
- Priority & VLAN
- Receive Buffers
- Receive Side Scaling

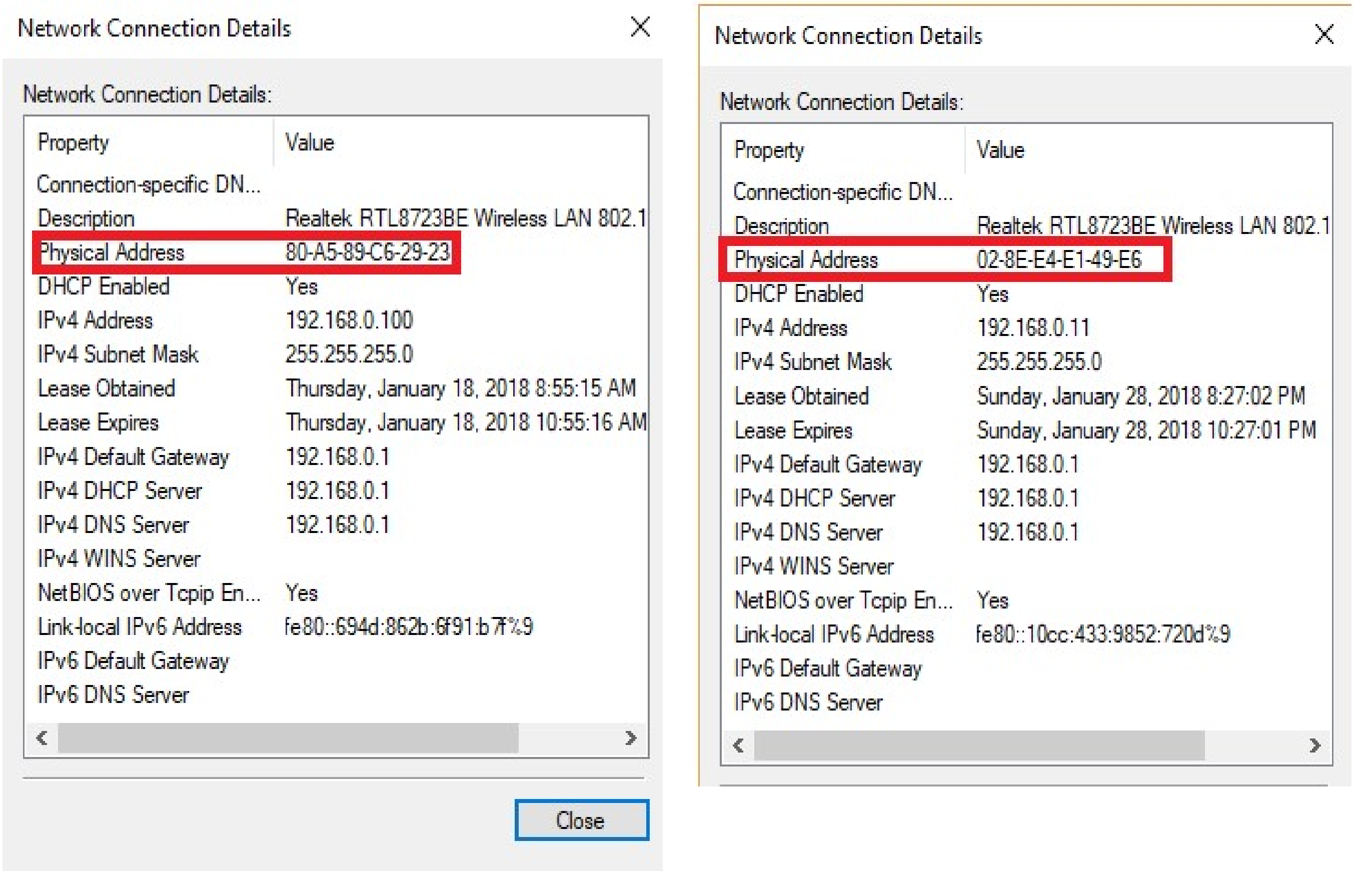
Value:

● 028EEAE149E6

○ Not Present

Figure 4: MAC address copy in system software

This is further illustrated by the two figures in figure 5 below. The first part of the figure shows the original MAC address `80-A5-89-C6-29-23` of the researcher's computer before it was altered using the procedure described above that correspond to figure 5 to `02-8E-E4-E1-49-E6`.



Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DN...	
Description	Realtek RTL8723BE Wireless LAN 802.1
Physical Address	80-A5-89-C6-29-23
DHCP Enabled	Yes
IPv4 Address	192.168.0.100
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Thursday, January 18, 2018 8:55:15 AM
Lease Expires	Thursday, January 18, 2018 10:55:16 AM
IPv4 Default Gateway	192.168.0.1
IPv4 DHCP Server	192.168.0.1
IPv4 DNS Server	192.168.0.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::694d:862b:6f91:b7f%9
IPv6 Default Gateway	
IPv6 DNS Server	

Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DN...	
Description	Realtek RTL8723BE Wireless LAN 802.1
Physical Address	02-8E-E4-E1-49-E6
DHCP Enabled	Yes
IPv4 Address	192.168.0.11
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Sunday, January 28, 2018 8:27:02 PM
Lease Expires	Sunday, January 28, 2018 10:27:01 PM
IPv4 Default Gateway	192.168.0.1
IPv4 DHCP Server	192.168.0.1
IPv4 DNS Server	192.168.0.1
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::10cc:433:9852:720d%9
IPv6 Default Gateway	
IPv6 DNS Server	

Figure 5: MAC Address before and after alteration

Availability of a MAC Address

Availability aspect of security as defined earlier on [29], refers to the accessibility and usability of an identifier upon demand by an authorized user. Availability ensures that the identifier works properly and that its service is available to valid users when needed. In ideal scenarios, a MAC address is usually made available by having it encoded to the network hardware (as in figure 6 below) as well as having a copy in the operating system as illustrated in figure 4 above. However, the effect of the possibility of altering a MAC address compromises the availability of a MAC address.

3.2.2 Robustness of a MAC Address

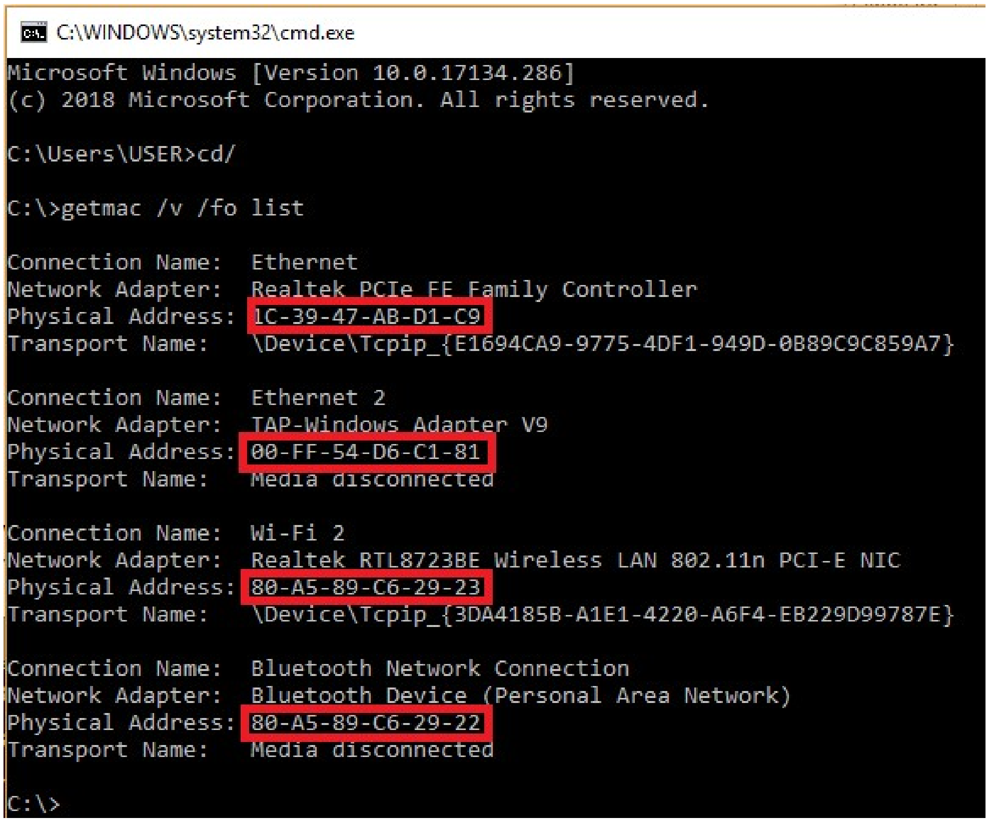
Robustness characteristic of an identifier refers to the ability of an identifier to function or continue functioning well in unexpected situations [31]. Closely related to robustness characteristic of an identifier, are performance and reliability characteristics. The questions that lead to the conclusion on whether a MAC address is robust or not include; does the MAC address remain invariant over time? Is a MAC address reliable? is it able to function as intended over a given period time under specified conditions?

The answers to these questions are based on fact that the initial intention of encoding MAC address in the network hardware is to make it independent from the operating system supposedly rendering it permanent and hard to alter [32]. However, due to some valid and invalid reasons, a copy of the MAC dress in the operating system can be altered as shown in figure 4. Good reasons for changing a device MAC address include testing out networks for configurations, security applications or new protocols, workarounds and nefarious means. For whichever reasons in changing a MAC address, it leads to the conclusion that a MAC address is not permanent and therefore unreliable.

3.2.3 The Uniqueness of a MAC Address

The fact that the MAC address is assigned to each network interface controller (NIC) card by the manufacturer makes it unique only to a particular interface. Furthermore, vendors are given a range of MAC addresses that can be assigned to their products by the IEEE [33]. This way, MAC address assignment is controlled such that no different adaptors can have the same address even if they are from different manufacturers. However, a device can have more than one network interface hence even though a MAC address can actually uniquely identify a network interface, it doesn't necessary uniquely identify a device.

One case in mind is an instance where a networked computer could contain multiple interfaces for Wi-Fi, Bluetooth and Ethernet adaptors. As illustrated in figure 6 below, a node can contain several MAC addresses. In this particular case, for instance, the node in question contains four interfaces with corresponding four MAC addresses. This was obtained by running the `getmac` command on the command prompt of the computer in question.



```
Microsoft Windows [Version 10.0.17134.286]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>cd/

C:\>getmac /v /fo list

Connection Name:  Ethernet
Network Adapter:  Realtek PCIe FE Family Controller
Physical Address: 1C-39-47-AB-D1-C9
Transport Name:   \Device\Tcpip_{E1694CA9-9775-4DF1-949D-0B89C9C859A7}

Connection Name:  Ethernet 2
Network Adapter:  TAP-Windows Adapter V9
Physical Address: 00-FF-54-D6-C1-81
Transport Name:   Media disconnected

Connection Name:  Wi-Fi 2
Network Adapter:  Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC
Physical Address: 80-A5-89-C6-29-23
Transport Name:   \Device\Tcpip_{3DA4185B-A1E1-4220-A6F4-EB229D99787E}

Connection Name:  Bluetooth Network Connection
Network Adapter:  Bluetooth Device (Personal Area Network)
Physical Address: 80-A5-89-C6-29-22
Transport Name:   Media disconnected

C:\>
```

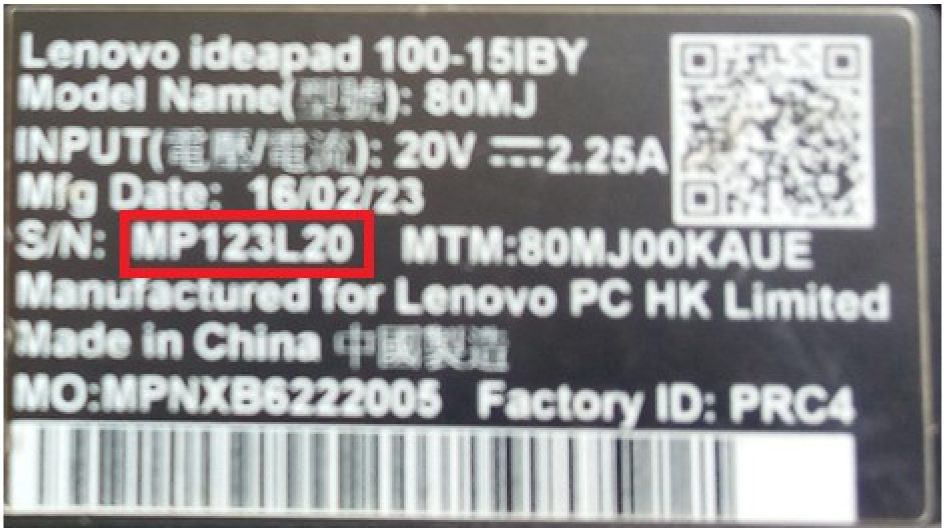
Figure 6: One computer with a number of MAC addresses

The possibility of a MAC address being spoofed is yet another case of a MAC address that makes it not to uniquely identify a device. If a device MAC address is altered for whatever reason, the likelihood of another device having the same address is imminent. As such, it cannot be assumed that a MAC address definitely identifies the device uniquely.

3.3 A Computer's Serial Number

3.3.1 Computer's Serial Number Location

Just like in any other product, a computer has its serial number tagged as part of the serialization of the product. Perhaps the only extraordinary thing is that the number is placed strategically on the computer to simply frustrate snoopers from finding it. One would, therefore, more than often find it usually tagged beneath the computer or staged somewhere beside it. Figure 7 below shows an illustration of a laptop model details that include the serial number in a tag.

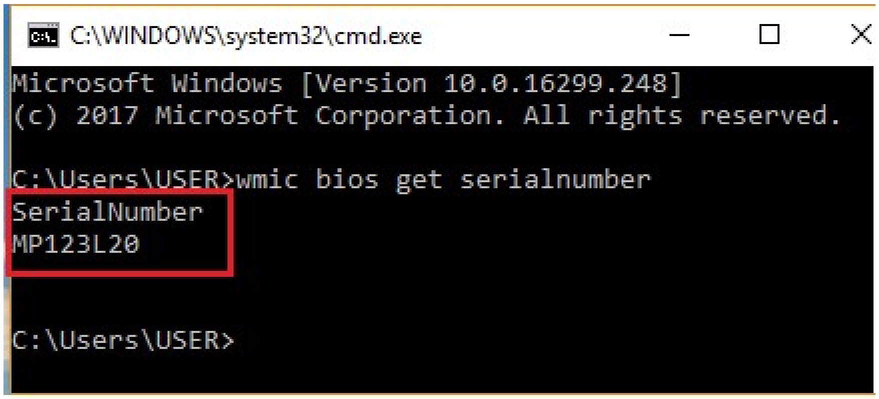


Lenovo ideapad 100-15IBY
Model Name(型號): 80MJ
INPUT(電壓/電流): 20V ⎓2.25A
Mfg Date: 16/02/23
S/N: MP123L20 MTM:80MJ00KAUE
Manufactured for Lenovo PC HK Limited
Made in China 中國製造
MO:MPNXB6222005 Factory ID: PRC4

Figure 7: Serial Number tag on a laptop model

Although tagging of computer serial number is the norm to serializing computers, it is a practice common to products including computers. This way, identifiers that use scanners such as bar code readers can be used to capture their identification details.

Alternatively, modern laptop models have their serial numbers coded into their basic input-output (BIOS) chips. This makes it possible for the identifier to internally be accessed and so, it can be processed for a given desired function. The first line of accessing a computers serial number is generally by running the command `wmic` bios get serial number at the systems command line interface. The serial number for the author's laptop, for instance, can be obtained as shown in figure 8 below.



```
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\USER>wmic bios get serialnumber
SerialNumber
MP123L20


C:\Users\USER>
```

Figure 8: Serial number of a laptop obtained from system BIOS using the `wmic` command

The encoding of a serial number in a computer's hardware rather than just tagging it on a surface makes it possible to manipulate the serial number. This way, a model that can access the serial number internally and use it to identify the device can be realized.

3.3.2 Characteristics of a Computer's Serial Number

The mere fact that a computer's serial number is hard-coded on the hardware without a copy in the system software alone renders it hard to be spoofed. It then implies that the serial number in normal circumstances cannot be altered and therefore unique, secure and reliable. It is only in some rare cases that the serial number can be altered. But this requires that a computer system has to turned off, any power lines disconnected, any static electricity discharged, computer case opened, disconnect the CMOS battery, wait for roughly 30 seconds (to completely ensure that the CMOS power is completely drained) then the process is done in reverse to revert back to original state [28]. This way, all original CMOS settings such as custom CMOS settings, BIOS password, time and date, as well as the motherboard serial number are lost. The system then generates a new system data that includes a serial number when booted.

4 Conclusions and Recommendations

The mere fact that a MAC address can be spoofed and altered affects its robustness and uniqueness attributes due to the fact that apart from it being hard-coded in the hardware, it has a copy of the MAC address in the system software. Uniqueness factor problem is more compounded due the possibility of multiple network interfaces attached to a computer that results to multiple MAC addresses for the same computer thus compromising the uniqueness quality of a MAC address as an identifier. On the other hand, a computer's serial number is hard-coded in the hardware only and therefore cannot be spoofed and altered making it unique, secure and reliable.

This study, therefore, recommends for a further research on how a serial number may be used for identifying a device in a wireless network. One way of realizing this recommendation could be by conducting a study towards an algorithm, a model and a prototype that can access a computer's number remotely and use it as an identifier. Subsequently, systems can as well be developed that would consequently use the computer's serial number as an identifier for authentication, authorization and accounting (AAA) to the network resources.

REFERENCES

[1] Dordal, P.L. (2018). *An introduction to Computer Networks*. Loyola University Chicago. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

[2] Raji, M.O. (2014). Design and Implementation of Wireless Network. *ResearchGate*. https://doi.org/10.13140/2.1.4578.4649

[3] DHS. (2017). *A Guide to Securing Networks for Wi-Fi (IEEE 802.11 Family)* [White Paper] https://www.us-cert.gov/sites/default/files/publications/A_Guide_to_Securing_Networks_for_Wi-Fi.pdf

[4] Wallace, K.. (2018). *Wireless LAN Security*, Kevin Wallace Training. https://www.kwtrain.com/blog/wlan-security

[5] Poremba, S.M. (2017, January 24). "*Network Access Control: Restricting and Monitoring Access to Your Network and Data".* https://www.esecurityplanet.com/network-security/network-access-control.html

[6] Pierce, M. (2021, May 12). "*What is NAC? Network Access Control Explained*". https://www.securedgenetworks.com/blog/network-access-control

[7] Singh, R. and Sharma, P.T. (2017). On the IEEE 802.11i security: a denial-of-service perspective, *Security Comm. Networks*; 8:1378–1407, DOI: 10.1002/sec.1079

[8] Stallings, W. (2011). *Network Security Essentials, Applications and Standards*, Fourth *Edition*. Pearson Education, New Jersey, USA

[9] Robb, D. (2022, January 24). "*Top Network Access Control (NAC) for 2022*". https://www.esecurityplanet.com/products/network-access-control-solutions

[10] Chiradeep, B. (2021, June 28). "*What Is Network Access Control? Definition, Key Components, and Best Practices*". https://www.toolbox.com/it-security/network-security/articles/what-is-network-access-control

[11] Lawson, L. (2017, April 17). "*Managed Services: A Security Problem and Solution".* https://www.esecurityplanet.com/network-security/managed-services.html

[12] Takahashi, D., Xiao, Y., Zhang, Y., Chatzimisios, P. and Chend, H. (2010). IEEE 802.11 user fingerprinting and its applications for intrusion detection, *Computers and Mathematics with Applications*, 60 (2010) 307_318

[13] Kurose, J.K. & Ross K.W. (2013). *Computer Networking: A Top_down Approach Featuring the Internet*, *6th Edition*. Addison Wesley

[14] Coulouris, G., Dollimore, J., Kindberg, T. and Blair, G. (2012). *Distributed Systems, Concepts, and Design*. Fourth Edition, Addison Wesley.

[15] Danev, B., Zanetti, D. and Capkun, S. (2015). **On physical-layer identification of wireless devices.** ***ACM*** *Computing Surveys, 451*(1), 1-29. https://doi.org/10.1145/2379776.2379782

[16] Lavassani, K.M., Movahedi, B. and Kumar, V. (2010). Identification in Electronic Networks: Characteristics of e-Identifiers, *Eight International Conference on Electronic Commerce (ICEC)*, 2006, Fredericton, New Brunswick, Canada

[17] Leo, R.V. (2004). Predicting consumer intentions to use on-line shopping: the case for an augmented technology acceptance model, *Information & Management*, 41, 747-762

[18] Aric, T. (2018, September 23). "Computer Network – Addressing (Port, Logical, Specific and Physical Address Basic Overview)". https://electronicsguide4u.com/computer-network-addressing-basic-overview/

[19] Lee, T. (2010). *Securing your Meru Network* [White Paper]. Meru Networks. https://www.martolvan.is/spuningar-og-svor/skjoel-og-fraeesluefni/meru-networks/meru-fraedhsluefni/1-meru-bpg-1/file

[20] Canavan, J.E. (2012). *Fundamentals of Network Security.* Artech House

[21] Tanenbaum, A. S. (2011). *Computer Networks, 4th Edition*, Pearson Education, USA

[22] Shay, W. A. (2004). *Understanding Data Communication and Networks*, *Third Edition*, Thomson Learning

[23] IEEE-USA. (2009). *Next Generation Internet: IPv4 Address Exhaustion, Mitigation Strategies and Implications for the U.S* [White Paper]. https://www.ieeeusa.org

[24] Kurose, J.K. & Ross K.W. (2013). *Computer Networking: A Top_down Approach Featuring the Internet*, *6th Edition*. Addison Wesley

[25] SlideToDoc. (2017). "*The OSI Model and the TCP/IP Protocol Suite".* https://slidetodoc.com/the-osi-model-and-the-tcpip-protocol-suite-3/

[26] Apple Inc. (2011, July 18). "*Uniquely Identifying a Macintosh Computer".* https://developer.apple.com/library/mac/technotes/tn1103/_index.html

[27] Omondi, F. (2017, March 1). "*How to Find your Computer Serial Number".* https://innov8tiv.com/find-computer-serial-number-windows-pcs/amp/

[28] Derekyoung. (2017, February 10). "*How to Change a BIOS Serial Number".* https://itstillworks.com/how-to-change-a-bios-serial-number-10394.html

[29] Paulsen, C. and Byers, R. (2019). Glossary of Key Information Security Terms. *National Institute of Standard and Technology (NIST)*, NISTIR 7298 Revision 3

[30] Kumar, C. (2022, April 24). "*11 Best IP Scanner Tools for Network Management*", https://geekflare.com/network-scanner/

[31] Corporation. (2018). *Microsoft Computer Dictionary*, 6th Edition. Microsoft Press, USA

[32] Cardenas, D.E. (2018). *MAC Spoofing: An Introduction,* [White Paper]. GIAC Security Essentials Certification (GSEC). https://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315

[33] Iwaya, A. (2015, September 13). "*How is the Uniqueness of MAC Addresses Enforced*?" https://www.howtogeek.com

APPENDIX XII

Journal Paper

International Journal of Wireless & Mobile Networks (IJWMN), Vol.14, No.4, August 2022

DOI:10.5121/ijwmn.2022.14403 27

PROTOTYPING A SERIAL NUMBER BASED AUTHENTICATION MODEL FOR A COMPUTER IN A WIRELESS LOCAL AREA NETWORK

John C. Chebor, Simon M. Karume, Nelson B. Masese and Andrew Kipkebut

Kabarak University, School of Science, Engineering and Technology, Nakuru, Kenya

ABSTRACT

With the increase of wireless LAN usage in homes and enterprises due to its numerous benefits, authenticating the ever increasing number of devices and their users has become a challenge to proprietors of such kind networks. A MAC address, a physical network address that is used as basis for this study, has a copy of its value in the system software that can be spoofed and altered rendering the address not unique, not secure and unreliable. On the contrary, a computer's serial number is hard-coded in the system hardware only and therefore cannot be spoofed and altered making it unique, secure and reliable. The research, therefore, was aimed at designing a model that demonstrates how a computer's serial number can be used for authenticating a computer in a wireless local area network. In order to achieve the research objective, the study examined the inbuilt access and use of a computer's serial number prototype model as an alternative method of authenticating devices in a network. Design science research methodology that involved design and development, demonstration and model evaluation was employed. A Serial Number Based Authentication prototype (SNAP) was therefore designed using state chart and flow chart diagrams based on dynamic programming, developed over evolutionary prototyping and test run on a static experimental design using Java Development Kit and MySQL platforms to demonstrate, as proof of concept, that a computer's serial number can be used to authenticate a computer in a wireless local area network. From the test runs whose outcomes were the binary values yes or no, it was found out that SNAP can actually allow or deny, enable or disable a computer in a network based on the computer's serial number. The researcher therefore, recommends that the prototype be scaled up, then adopted as a network device authentication method.

KEYWORDS

Computer's Serial Number, Authentication, Wireless LAN, Serial Number-Based Authentication

1. Introduction

Wireless LANs (WLAN) also known as Wireless Fidelity (Wi-Fi) or 802.11 standards is a type of a local area network that allows users access network services using mobile devices (wireless stations) such as laptops, personal digital assistants, smart watches and even smartphones (Dordal, 2018). The wireless stations use a base station usually an access point (AP) or a hotspot as an entry point to the network services (Romanov & Succi, 2018). Unlike wired LANs that use cables or wires as transmission media, WLANs uses radio wave frequencies to transmit information over the local area network.

WLAN, therefore, comes with a myriad number of benefits as compared to wired LANs, notably, mobility, rapid deployment, reduction in infrastructure and operational costs, flexibility, and scalability (DHS, 2017 and Wallace, 2018). Due to these benefits, hotspots are now virtually found everywhere; in enterprises, at homes, and in public places. Wireless devices such as laptops, personal digital assistants and even smartphones come with Wi-Fi features integrated in them. Despite the numerous benefits that come with wireless LANs, network insecurity has become a thorn in the flesh for proprietors of such kind of networks. Singh & Sharma, (2017), points out that an attacker simply needs to be within the range of the WLAN access point to intrude into the network as opposed to wired LANs where an attacker requires physical access to the LAN or remotely use porous firewall systems to gain access to the LAN. WLANs are therefore easily targeted by attackers through spoofing, denial of service, eavesdropping, man-in-the-middle, masquerading, message modification, message replay and traffic analysis (Stallings, 2011; Poremba, 2017; DHS, 2017). Wallace, (2018), describes the reasons for the threats as default configurations, network architecture nature, encryption weaknesses, and physical security.

The numerous benefits that come with WLAN have made enterprises to adopt bring your own device (BYOD) concept to allow employees use their own devices (Poremba, 2017). Apart from the employees accessing and using enterprise networks, another group of stake holders such as visitors, vendors, and contractors at one point, if not all, can as well require network usage as they carry on their businesses with the enterprise. Rise in internet of things (IOT) devices such smart devices, smart watches and smart phones, as well, further complicates WLAN challenges equation (Pierce, 2021; Elkhodr & Mufti, 2019; Elkhodr et al. 2016). Allowing users to connect to the network with their own devices can pause as a security challenge as it becomes difficult for network administrators to control such kind of a network access and usage. It is therefore imperative that network administrators use network access control tools to control who should and who should not access the network. One of such kind of a control tool is the network access control (NAC) (Robb, 2022; Pierce, 2021; & Chiradeep, 2021). NAC constitutes identification, authentication, authorization, and accounting (IAAA) according to Lawson, (2017) as the essential functions in providing the required services in a network.

Major authentication methods or technologies began way back during the Second World War by the use of identification of a friend or a foe (IFF) (Lehtonen et al., 2008). From then on, advances in authentication techniques that include password, smart card, biometric, certificate MAC address, IP address and multi-factor (MFA) based authentication methods took effect (Shacklett, 2021; Johnson, 2021 and Fredriksson, 2017). The methods are categorized based on what one is known for or knowledge based (password, PIN), what one has or possesses (token, certificate), who one is or inheritance (biometrics), where one is or location based or address based (MAC address, IP address) and when one is authenticating or time factor as well (Shacklett, 2021). Out of all the mentioned authentication methods, MAC address, IP address and at times certificates are machine based authentication methods (Fredriksson, 2017). A MAC address, a physical network address that is used as basis for this study, has a copy of its value in the system software that can be spoofed and altered rendering the address not unique, not secure and unreliable, making it not suitable for authentication. A computer's serial number, in contrast, is only hard-coded on the hardware without a copy in the system software alone renders it hard to be spoofed (Derekyoung, 2017).

This research, therefore, was aimed at designing a model that demonstrates how a computer's serial number can be used for authenticating a computer in a wireless local area network through answers to the research questions: (i) How can an algorithm that can obtain and use a remote computer's serial number in a wireless LAN be developed? (ii) How can a model that uses the computer's serial number to authenticate the computer in a wireless LAN be designed? (ii) How can the model that uses the computer's serial number to authenticate the computer in a wireless LAN be demonstrated? (iv) How can the model that uses the computer's serial number to authenticate the computer in a wireless LAN be evaluated? The paper was then structured as follows: Section 1 provides background information and study motivation. Section 2 deals with related work. Section 3 presents the design, development, demonstration and evaluation of the prototype perceived to address the study questions listed. Section 4 presents the conclusion derived from the study

2. Related Work

As a subset of location-based authentication, IP address authentication is a traditional method of authenticating computers that require network and resource access. Once a user logs onto a network, IP address authentication checks on their IP address and validates them against a list of allowed IPs or IP ranges. When a range of addresses is specified, the access point performs a logical and with the IP address entered in the IP address filter and the configured subnet. If an exact IP address is specified, the authentication method specifies a subnet mask so that only request from a client IP address is allowed or blocked, depending on what is configured in the filter (Servicenow, 2022). This eliminates or reduces the use of user IDs and passwords, initial configuration and maintenance is simple, works well with static IPs, however, requires a separate remote authentication tool, and slow for dynamic IPs, (EBSCO, 2022). But with dynamic nature of networks, that is, users and devices are mobile, plus users can use multiple devices from different locations, thus IPs might not correspond to their institutions. Although virtual private networks and proxy services have been fronted to remedy the drawback, they are complicated to manage have their own issues (Hoy, 2019). IP address range makes it easy to be spoofed, dynamic IP address allocation results to multiple users using the same IP, and the IP address keeps changing for the same device from location to location (Chad, 2017).

MAC address authentication, on the other hand, a port-based authentication method, allows or denies network access based on the MAC address credentials for machines such as IP phones, printers, and network attached storage devices. As a layer 2 OSI reference model issue, MAC address authentication solution uses RADIUS over IEEE802.1x framework rather than EAP (Cisco, 2018; Fredrisson, 2017). When a device connects to an access point (AP), the AP forwards the MAC address as the log in credential to the RADIUS server. With MAC-based authentication, the MAC address serves as both the username and the password. The RADIUS server consults the authentication server and sends back a RADIUS return attribute based on authentication results

MAC address filtering, also referred to as address control or address reservation, or still wireless MAC authentication allows or blocks traffic from a known machine or device depending on an organizational security policy to secure their networks (Andysah, 2017). However, due to a copy of a MAC address value in the system software (Cardenas, 2003), MAC addresses can be spoofed (Lee, 2010) and can be altered (Apple, 2011). Additionally, MAC addresses are not encrypted (Gill & Dahiya, 2017), it is not recommended for large wireless networks (Singh & Sharma, 2015; Watanabe et al, 2013) and furthermore, Kurose & Ross, (2013), raises the issue that a device can be attached to multiple networks each with a corresponding MAC address interface. A device, for instance can be attached to an Ethernet port, a Wi-Fi, or a Bluetooth port, all of which cannot uniquely identify the device. A MAC address in a nutshell, is not unique, is not secure and unreliable as required characteristics for identifiers (Developer, 2022)

3. Serial Number Based Authentication Prototype

The serial number based authentication prototype (SNAP) that was perceived to address the challenges of using MAC address authentication, for authenticating computers in a network. The model was designed, developed, demonstrated and evaluated based on design science research (DSR) as follows

3.1 Design and Development

As earlier stated, the ultimate goal of the study was to design a model that demonstrates how a computer's serial number, as an identifier, can be used to authenticate a computer in wireless LAN. As such, authentication details that included the serial number, the IP address and the computer's name had to eventually be displayed on a display interface for further manipulation or be controlled if need arises. To achieve this ultimate goal, the SNAP model was designed and developed to first, registers computers so that it can later be used as a yard stick to either allow or deny computers access to the network. Then upon executing the application, it deletes existing authentication details in order to pave way for newly logged on computers' authentication details. This is followed by system collecting IP addresses, computer names, and computers' serial numbers for all computers that log on to the network. The authentication details are then collated for the purpose of validating them, as a unit, based on their registration status. Registered computers are allowed access to the network and unregistered computers are denied network access. Once they are allowed network access, valid computer authentication details are displayed on a `Connected Devices` display interface. Apart from containing computer's `Serial Number`, computer's `Name` and computer's `IpAddress`, the `Connected Devices` display interface contains an additional `Status` column to allow the validly logged on computers to be controlled if need arises. All the fundamental components, their details and relationships are illustrated in the figure 1 below

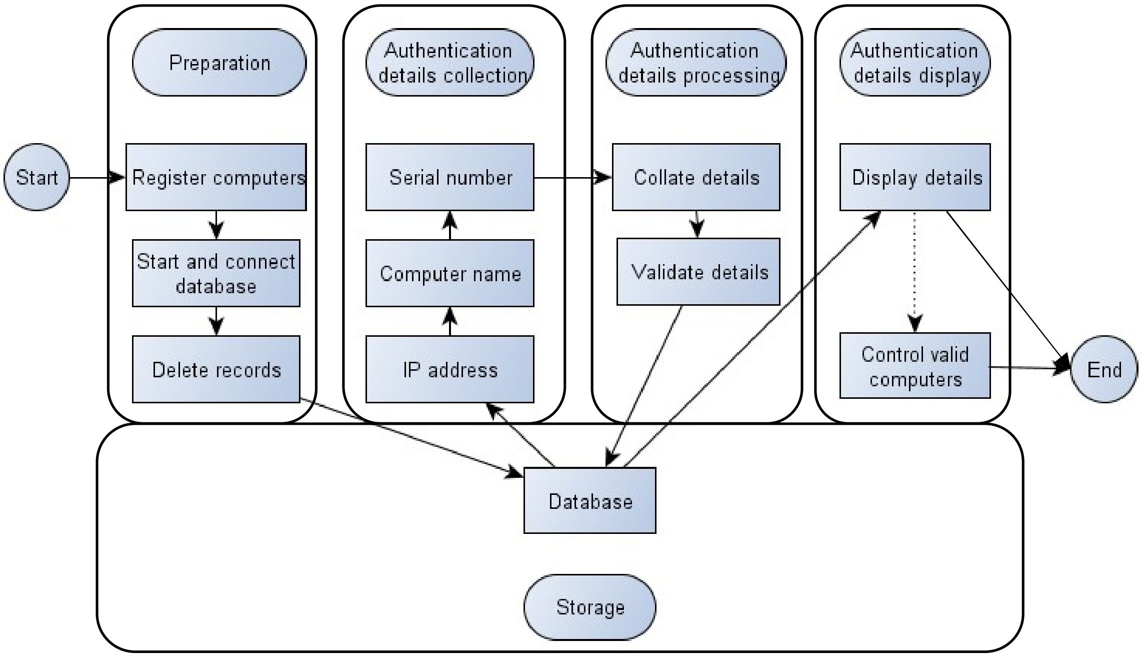


Figure 1: General model for a serial number based authentication system

A dynamic programming design was adopted during the algorithm design development phase of the study due to the fact that a dynamic programming algorithm optimizes solutions on a step by step basis recursively to the whole (SmartDraw, 2018; Visual-paradigm, 2018; Paramalways, 2009). In other words, the results of one step solve the problem of another consecutive step recursively. The algorithm with a corresponding flow chart (figure 2) that depicts the system description is as follows:

```
8. Start
9. Register computers
10.     Delete existing connected computers details
11.     Get connected computer details
    11.1 Get the computer name
    11.2 Get computer raw IP address
    11.3 Get computer serial number
    11.4 Collate computer IP address, name and serial number
12.     If computer is registered
    12.1 Allow computer connect to the network
    12.2 Post Connected Computer Details to the Database
    12.3 Retrieve and Display the Connected Computer Details
         from Database
    12.4 Control validly allowed computers
13.     Else, deny computer network access
14.     End
```

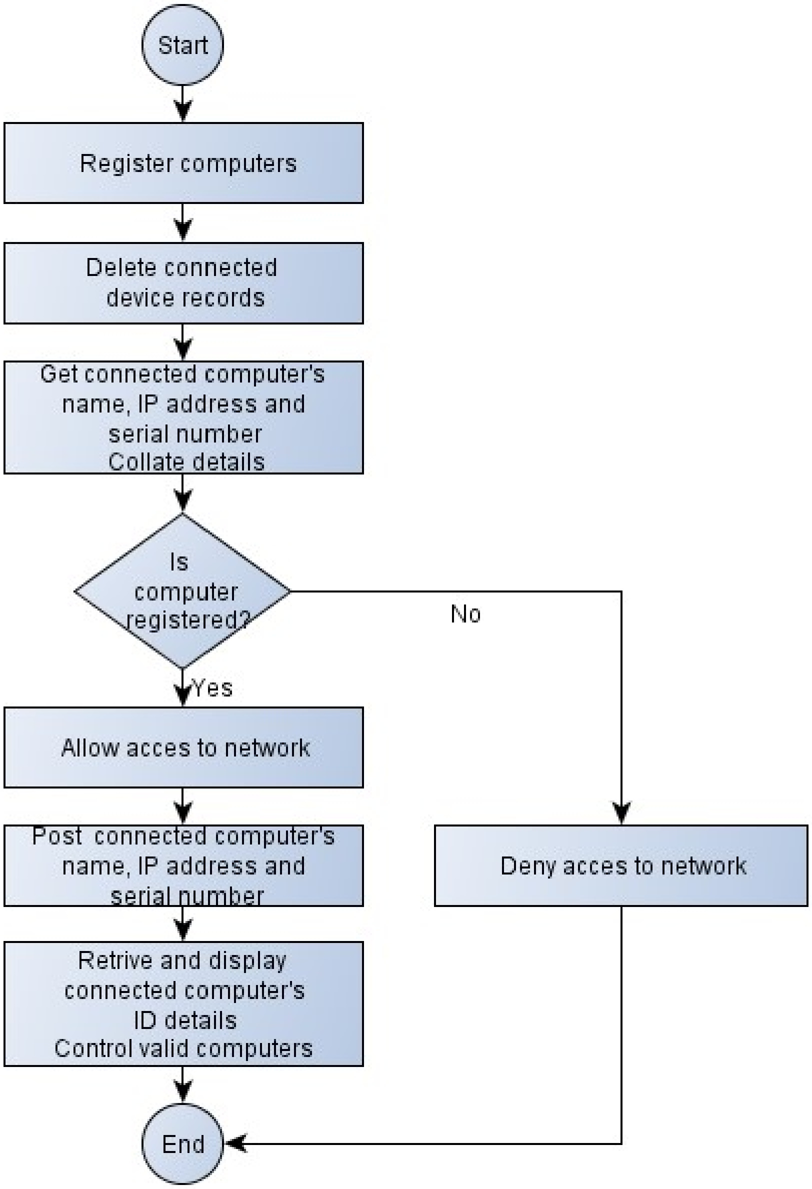

Start
Register computers
Delete connected device records
Get connected computer's name, IP address and serial number Collate details
Is computer registered?
No
Yes
Allow acces to network
Post connected computer's name, IP address and serial number
Deny acces to network
Retrive and display connected computer's ID details Control valid computers
End

Figure 2: General flow chart for SNAM system

3.2 Demonstration

In order to demonstrate the proof of concept that a computer's serial number can be used to authenticate a computer in a wireless LAN, the SNAP model was implemented using MySQL database and Java's IDE tools over evolutionary prototyping using a static group comparison pre-experimental design set up. The set contained authentication server, authentication details database, an access point, two clients, the SNAP application and their connections as depicted in figure 3 below.

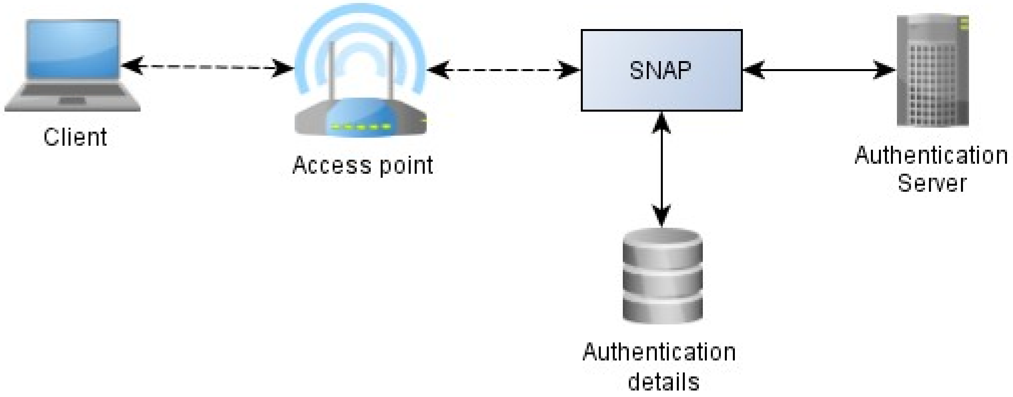
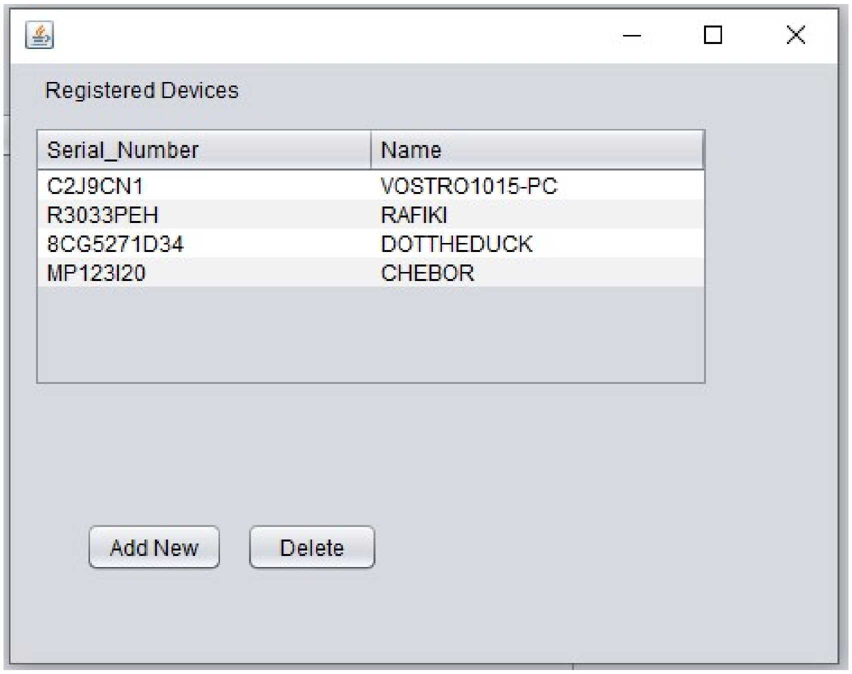


Figure 3: The SNAP system components

While the authentication server was installed with the SNAP application to perform authentication processing with whose details are stored and managed in the authentication database, the access point acted as a link between the clients and the server just as the case in a client-server architecture. Two clients were part of the set up so that one was configured for valid and the other for invalid expected outcomes. Apart from the need of using proof of concept (PoC) to prove the overall concept that a computer's serial number can be used to authenticate a computer in a wireless LAN, the set up as well was geared towards using PoC to prove that other modules of prototype that culminates to the overall concept, can as well be executed. PoC according to Leurs & Duggan, (2018) and MacPherson, (2018) is an exercise to test design or assumption ideas. The other concepts of the study, that correspond to the system modules, and required PoC were that;

5. A computer's authentication details (that is name, IP address and Serial number) can be collected
6. A registered computer can access a network
7. An unregistered computer cannot access a network
8. An already logged on or an allowed computer can be denied access to the network if a need arises

To begin with, four computers were registered as displayed in the `Registered Devices` interface shown the figure 4 below

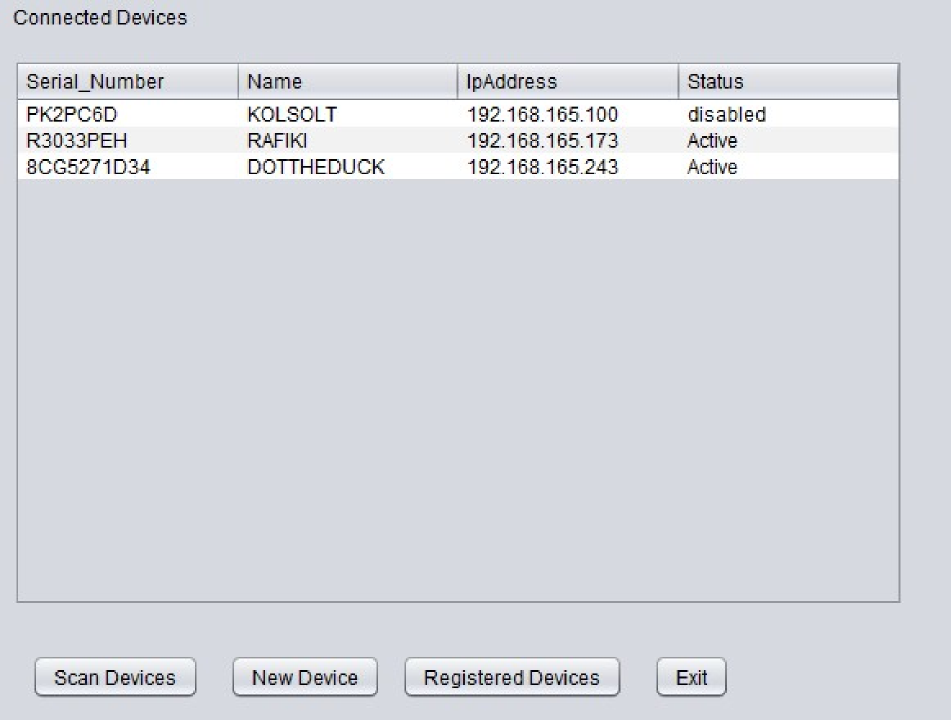


Registered Devices

Serial_Number	Name
C2J9CN1	VOSTRO1015-PC
R3033PEH	RAFIKI
8CG5271D34	DOTTHEDUCK
MP123I20	CHEBOR

Figure 4: Computers that are already registered in the system

During the demonstration, three computers, `KOLSOLT` (unregistered client), `RAFIKI` (registered client) and `DOTTHEDUCK` (server) were connected. On test running the SNAP application, only `RAFIKI` and `DOTTHEDUCK` computers were allowed access while the unregistered `KOLSOLT` computer was denied network access as illustrated by the disabled state of the `Status` column on the `Connected Devices` display diagram in the figure 5 below

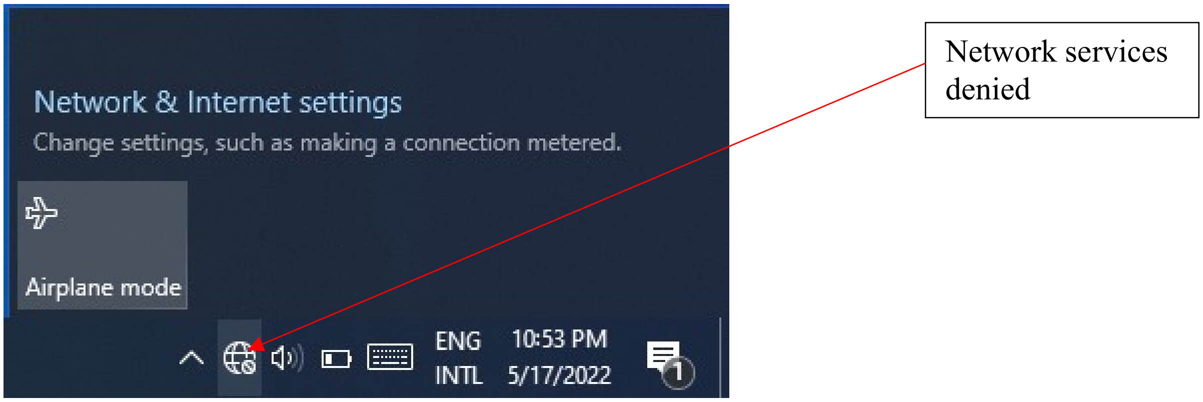


Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	disabled
R3033PEH	RAFIKI	192.168.165.173	Active
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

Figure 5: Blocked unregistered `KOLSOLT` computer

The Wi-Fi interface status for the `KOLSOLT` was equally disabled as indicated in the figure 6 below



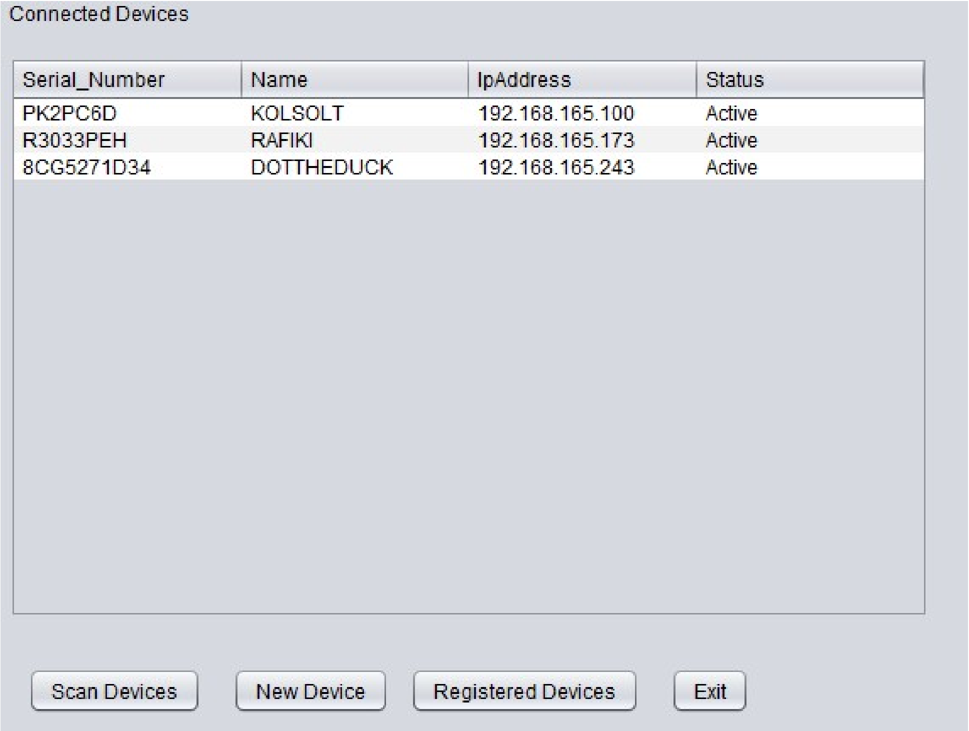
Network & Internet settings

Change settings, such as making a connection metered.

Network services denied

Figure 6: Blocked unregistered `KOLSOLT` computer Wi-Fi interface status

But once `KOLSOLT` was registered and the SNAP application executed once again, `KOLSOLT` was allowed network access together with other two registered computers as shown in the figure 7 below



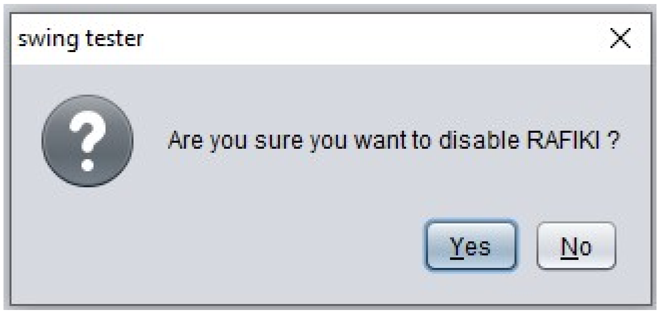
Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	Active
R3033PEH	RAFIKI	192.168.165.173	Active
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

Figure 7: `KOLSOLT` allowed access after registration

The other aspect of the prototype was to develop and test run a section that controls validly allowed computers if need arises, while allowing other computers to continually use the network. This was achieved by first clicking on the desired client (e.g `RAFIKI` client computer in this case) on the `Connected Devices` interface on figure 7 above.

The system prompts on the surety to disable the client from accessing the network figure 8

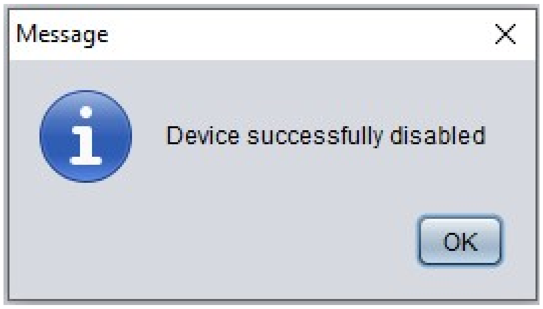


swing tester

Are you sure you want to disable RAFIKI ?

Figure 8: Denying `RAFIKI` computer network access prompt message

On clicking `YES` button, the `RAFIKI` client is successfully denied network access as illustrated in system prompt in figure 9 below.

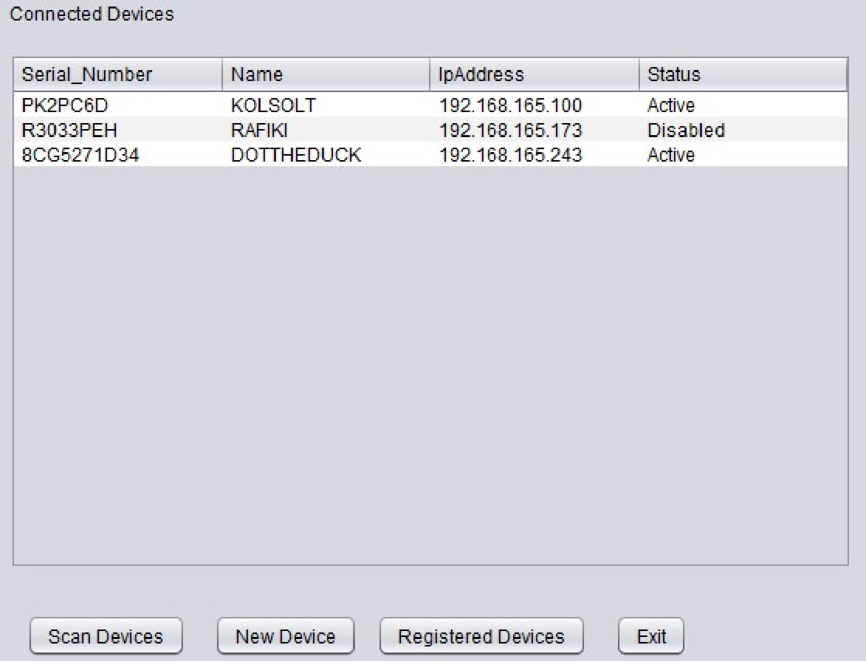


Message

Device successfully disabled

Figure 9: Denying `RAFIKI` computer network access confirmation message

The confirmation that `RAFIKI` client has been disabled, therefore, cannot access the network is illustrated in a screen shot in figure 10 below

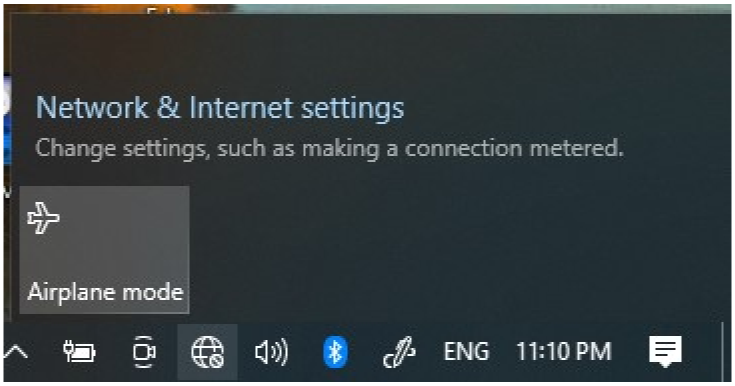


Connected Devices

Serial_Number	Name	IpAddress	Status
PK2PC6D	KOLSOLT	192.168.165.100	Active
R3033PEH	RAFIKI	192.168.165.173	Disabled
8CG5271D34	DOTTHEDUCK	192.168.165.243	Active

Figure 10: Denied `RAFIKI` computer network access

The client Wi-Fi interface status as well indicates that the client cannot access the network as shown in the figure 11 below



Network & Internet settings

Change settings, such as making a connection metered.

Figure 11: Blocked network denied KOLSOLT computer Wi-Fi interface status

A summary of the test runs on whether the prototype can or cannot register computers, collect computers authentication details, allow or deny a computers network access based on whether they are registered or not, and allow a valid computer to either continue using the network or be denied network resource usage due one or another reason were summarized as in table 1 below.

Table 1: System modules execution report

Functions	**Description**	**Execution Status (Yes/No)**	**Remarks**
New Computer registration	Check if a new computer can be registered	Yes	Are confirmed in the Registered Devices interface
Collect Computer's Serial Number, IP address, Name	Check if authentication details can be collected from the system	Yes	Collected identification details are displayed on the Connected Devices interface
Collect Computer's authentication details with different access points (APs)	Check if authentication details can be collected from the system using different APs	Yes	Collected authentication details are displayed on the Connected Devices interface for three different APs
Allow a registered computer network access	Check if a registered computer is allowed network access	Yes	Registered computers are displayed in the Connected Devices interface
Deny an unregistered computer network access	Check if an unregistered computer is denied network access	Yes	Unregistered computers are denied network access as indicated in the affected computers network access status interface
Disable allowed computer	Check if an allowed computer can be disabled	Yes	This is confirmed by the disabled status of the computer in the Connected Devices interface and in the affected computers network access status interface

3.3 Prototype Evaluation

For validation purposes, proof of concept method on the cumulative essential functionalities of the prototype was evaluated using test runs as proposed by Diceus, (2020) based on goal-based evaluation method (Cronholm & Goldkuhl, 2003). With the focus of being on intended services and outcomes of a program, goal-based evaluation measures the extent to which a program attains its intended goals. The prototype was fundamentally geared towards the ability to

1. Collect a computer's authentication details (that is name, IP address and Serial number)
2. Allow a registered computer access to a network
3. Deny an unregistered computer access to a network
4. Control an already logged on computer to a network if need arises

A set of three test runs were carried out using static group comparison pre-experimental design. Just as in the initial experimentation set up, each set had three computers and an access point. One of the computers, that was configured as server, was installed with the serial number authentication application. For comparisons purposes, the other two computers were configured as clients, one for valid and the other for invalid expected outcomes.

Each set was carried out independently by a wireless LAN stakeholder (that is, a network administrator, a Wi-Fi owner and an IT expert). Each participant was given instruction on how to run the system against a check list of the four test runs. They were expected to observe the behavior when the system is run and manipulated according to the fundamental questions that corresponds to the system functionalities and answer yes or no on the check list form given to them. The questions on the checklist that corresponded with the evaluation goals were as follows;

5. Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?
6. Can the system allow a registered computer access to a network?
7. Can the system deny an unregistered computer access to a network?
8. Can the system deny an already logged on computer to a network if need arises?

The summary of the results from the three evaluators as indicated in appendix were summarized as in the table 2 below.

Table 2: Evaluators system modules execution report

#	Evaluation	Evaluator 1	Evaluator 2	Evaluator 3
			Yes/No	
5.	Can the system collect a logged on computer's authentication details (that is name, IP address and Serial number)?	Yes	Yes	Yes
6.	Can the system allow a registered computer access to a network?	Yes	Yes	Yes
7.	Can the system deny an unregistered computer access to a network?	Yes	Yes	Yes
8.	Can the system deny an already logged on computer to a network if need arises?	Yes	Yes	Yes

4. Conclusion

From the discussions based on the design, development, demonstration and evaluation of a prototype that uses a computer's serial number to authenticate a computer in a wireless LAN, the following findings, that are actually geared towards answering the study questions, can be made thus (i) an algorithm that can obtain a remote computer's serial number in a wireless LAN can be developed using dynamic programming algorithm design, (ii) a model that uses the computer's serial number to authenticate the computer in a wireless LAN can be designed using a state chart and flowchart diagrams, (iii) a model that uses the computer's serial number to authenticate a computer in a wireless LAN can be demonstrated using a prototype on MySQL database and Java's NetBeans IDE tools and (iv) that the model that uses the computer's serial number to authenticate the computer in a wireless LAN can be evaluated using a goal-based evaluation method by independent evaluators. Furthermore, the test runs from the prototype set up indicated that the SNAM prototype could collect a logged on computer's authentication details (that is name, IP address and Serial number), can allow a registered computer access to a network, can deny an unregistered computer access to a network and that it can deny an already logged on computer to a network if need arises. In turn, a serial number can therefore, be used to authenticate a computer in a wireless LAN. It is then recommended that the prototype be scaled up so that it can adopted an authentication method.

5. References

Apple Inc. (2011, July 18). "*Uniquely Identifying a Macintosh Computer*". https://developer.apple.com/library/mac/technotes/tn1103/_index.html

Cardenas, D.E. (2018). *MAC Spoofing: An Introduction* [White Paper]. GIAC Security Essentials Certification (GSEC). https://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315

Chad, F. (2017, May 3). "*IP Authentication vs. Username and Password Login*". https://www.proxykey.com/ip-authentication-vs-username-password-login

Chiradeep, B. (2021, June 28). "*What Is Network Access Control? Definition, Key Components, and Best Practices*". https://www.toolbox.com/it-security/network-security/articles/what-is-network-access-control

Cisco. (2018, August 14). "*Configuring MAC-Based Authentication on a Switch*". https://www.cisco.com/c/en/us/support/docs/smb/switches/cisco-350-series-managed-switches/smb5836-configuring-mac-based-authentication-switch.html

Derekyoung. (2017, February 10). *"How to Change a BIOS Serial Number".* https://itstillworks.com/how-to-change-a-bios-serial-number-10394.html

DHS. (2017). *A Guide to Securing Networks for Wi-Fi (IEEE 802.11 Family*) [White Paper]. Department of Homeland Security (DHS). https://www.us-cert.gov/sites/default/files/publications/A_Guide_to_Securing_Networks_for_Wi-Fi.pdf

Diceus. (2022, May 18). "*Top 5 steps to define crucial POC success criteria*". https://diceus.com/poc-success-criteria/

Dordal, P.L. (2018). *An introduction to Computer Networks*. Loyola University Chicago. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

EBSCO. (2022, February 7). "*What is IP Address Authentication*". https://connect.ebsco.com/s/article/What-is-IP-Address-Authentication?language=en_US

Elkhodr, M. & Mufti, Z.B. (2019). On The Challenges of Data Provenance in the Internet Of Things. *International Journal of Wireless & Mobile Networks (IJWMN) (11)*3, 43-52

Elkhodr, M. Shahrestani, S. & Cheung, H. (2016). Emerging Wireless Technologies in The Internet Of Things: A Comparative Study. *International Journal of Wireless & Mobile Networks (IJWMN) (8)*5, 67-82

Johnson, K. (2021, December 14). *"Use these 6 user authentication types to secure networks".* https://www.techtarget.com/searchsecurity/tip/Use-these-6-user-authentication-types-to-secure-networks

Lawson, L. (2017, April 17). "*Managed Services: A Security Problem and Solution."* https://www.esecurityplanet.com/network-security/managed-services.html

Lee, T. (2010). *Securing your Meru Network* [White Paper]. Meru Networks. https://www.martolvan.is/spuningar-og-svor/skjoel-og-fraeesluefni/meru-networks/meru-fraedhsluefni/1-meru-bpg-1/file

Lehtonen, M., Staake, T., and Michahelles, F. (2008). From identification to authentication–a review of RFID product authentication techniques. In *Networked RFID Systems and Lightweight Cryptography* (pp. 169-187). Springer Berlin Heidelberg

MacPherson, L. (2018). "*5 Steps to a Proof of Concept for Successful Software Development*". https://designli.co/blog/5-steps-proof-concept-successful-software-develoment

Paramalways. (2009). Dynamic Programming, *Technology*, https://www.slideshare.net,

Pierce, M. (2021, May 12). "*What is NAC? Network Access Control Explained*". https://www.securedgenetworks.com/blog/network-access-control

Poremba, S.M. (2022, January 24). "*Network Access Control: Restricting and Monitoring Access to Your Network and Data*." https://www.esecurityplanet.com/network-security/network-access-control.html

Robb, D. (2022, January 24). "*Top Network Access Control (NAC) for 2022*". https://www.esecurityplanet.com/products/network-access-control-solutions/

Romanov, V. & Succi, G. (2018). WLAN BASED POSITIONING WITH A SINGLE ACCESS POINT. *International Journal of Wireless & Mobile Networks (IJWMN) (10)*3, 37-50

Servicenow. (2022, February 3). "*IP Range Based Authentication*". https://docs.servicenow.com/bundle/sandiego-platform-administration/page/administer/login/concept/c_IPRangeBasedAuthentication.html

Shacklett, E. M. (2021, September 13). "*Authentication*". https://www.techtarget.com/searchsecurity/definition/authentication

Singh, R. and Sharma, P.T. (2017). On the IEEE 802.11i security: a denial-of-service perspective. *Security Communication Networks*, *(8)*7, 1378–1407. http:/doi.org/10.1002/sec.1079

SmartDraw. (2019). "*HOW TO DRAW DATA FLOW DIAGRAMS".* http://120.105.184.180/lwcheng/SSADM/

Stallings, W. (2011). *Network Security Essentials, Applications and Standards*, Fourth Edition. Pearson Education

Visual-paradigm. (2018). "*How to Draw an Activity Diagram in UML*". https://www.visual-paradigm.com/tutorials/how-to-draw-activity-diagram-in-uml/

Wallace, K. (2018, April 06). "*Wireless LAN Security*". https://www.kwtrain.com/blog/wlan-security

Watanabe, Y., Otani, M., Eto, H., Watanabe K. & Tadaki, S. (2013). A MAC address based authentication system applicable to campus-scale network. *2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2013, pp. 1-3