2021

A Model to detect and prevent rogue DHCP attacks on wireless LAN communication.

Wachira, Fiona Njeri
School of Computing and Engineering Sciences
Strathmore University

Strathmore UNIVERSITY

A Model To Detect And Prevent Rogue DHCP Attacks On Wireless LAN Communication

Wachira, Fiona Njeri

Master of Science in Information System Security

2021

Declaration

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

Wachira, Fiona Njeri

Signature: _ _____

Date: _07-05-2021_____

Approval

This dissertation of Wachira, Fiona Njeri was reviewed and approved by:

Lecturer: Dr. Vitalis G. Ozianyi

Signature:

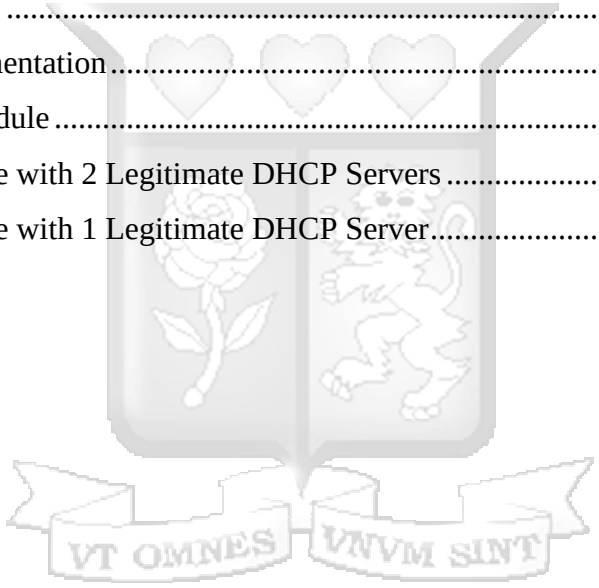
Date: 07-05-2021

Abstract

WLAN technology is a crucial component of computer networks. The use of Wi-Fi communication has grown due to the increasing population of end devices, which includes smartphones, tablets, laptops etc. This has significantly increased the number of internet users. When mobile hosts move from one network to another, they require new system configurations in order to communicate hence the use of WLAN. Dynamic Host Configuration Protocol (DHCP) supports automatic configuration of hosts. With respect to DHCP processes, one of the internal attacks that majorly affects WLAN security is rogue DHCP server. Due to the nature of DHCP communication, it is easy for an attacker to introduce a rogue DHCP server. This is possible since a client can receive DHCPOFFER messages from more than one DHCP server. To address this issue, the study proposes a custom IDS that detects rogue DHCP server attacks by monitoring and analysing DHCP transaction messages. The study implements an experimental design that involves setting up a test network containing both rogue and genuine DHCP servers. Packet characteristics of rogue DHCP are collected and analyzed to identify the parameters to be used by the IDS. To validate the proposed solution, the offered IP addresses by the rogue DHCP server are checked against the report generated by the IDS. From the findings, it was confirmed that the IDS has 100% detection rate since all offered IP addresses by rogue DHCP server were detected by the IDS.

Table of Contents

Table of Figures



List of Tables

List of Abbreviations/Acronyms

AP	-	Access Point
ARP	-	Address Resolution Protocol
BASE	-	Basic Analysis and Security Engine
BOOTP	-	Bootstrap Protocol
DAI	-	Dynamic ARP Inspection
DB	-	Database
DHCP	-	Dynamic Host Configuration Protocol
DNS	-	Domain Name System
DoS	-	Denial of Service
GUI	-	Graphical User Interface
IDS	-	Intrusion Detection System
IP	-	Internet Protocol
IPS	-	Intrusion Prevention System
IETF	-	Internet Engineering Task Force
LAN	-	Local Area Network
MAC	-	Media Access Control
MITM	-	Man-in-the-Middle
NIC	-	Network Interface Card
NIDS	-	Network Based Intrusion Detection System
NLA	-	Network Location Awareness
PGP	-	Pretty Good Privacy
PKI	-	Public Key Infrastructure
RFC	-	Request For Comments
TCP/IP	-	Transmission Control Protocol and the Internet Protocol
TDD	-	Test-Driven Development
UDP	-	User Datagram Protocol

UFW - Uncomplicated Firewall

UI - User Interface

Wi-Fi - Wireless Fidelity

WLAN- Wireless Local Area Network

WPA2 - Wi-Fi Protected Access

Chapter 1: Introduction

1.1 Background

A Local Area Network (LAN) is an interconnection of devices within a relatively small geographical location such as an office, home, campus etc. LANs are majorly divided into two namely: Fixed LAN and Wireless Local Area Network (WLAN). WLAN is a wireless computer network that links multiple devices within a limited geographical area (Tomar, 2020). WLAN technology uses radio waves to provide connectivity. WLAN are usually implemented as extensions to wired local area networks to provide enhanced user mobility (Caballero, 2014).

Wireless network has become popular in recent years and it is commonly being used in most public areas such as restaurants and malls (Deliang et al., 2016). Companies and institutions are investing in wireless networks to take advantage of mobile, real time access to information. University students use the Internet to support their social life through social networks such as Facebook, Twitter and WhatsApp. They also use the Internet for learning and researching through e-learning platforms. Most users may want to access the internet from anywhere in the world. The drive towards using wireless networks is caused by its striking important features like customizable security, reliability and availability with little downtime, and high performance.

However, when mobile hosts move from one network to another, they require new system configurations, including host IP address which is an identifier that uniquely identifies a device in a network, IP default gateway and IP of domain name servers. In order to support automatic configuration of hosts, several technologies such as Dynamic Host Configuration Protocol (DHCP) have been implemented. Request for Comments (RFC) 2131 and 2132 define DHCP as an Internet Engineering Task Force (IETF) standard based on the Bootstrap Protocol (BOOTP) (Kerravala, 2018). DHCP protocol is an application protocol that employs a connectionless service using the User Datagram Protocol (UDP). It is an internet protocol that automates assignment of IP addresses that devices use to communicate to each other in a network (Hendrickson, 2019). It runs on port 67 which is the destination port of a server and 68 which is used by the client (Yaibuates et al., 2016).

DHCP can be used to automatically assign IP addresses and configure other network parameters such as subnet mask, IP of default gateway and IP address of DNS servers. Research study by Yaibuates & Chaisricharoen (2020) states that without the use of DHCP it becomes imperative to

manually configure IP addresses, a process that is cumbersome and time consuming. Furthermore, manual configuration is susceptible to errors and this incorrect configuration will deny the devices from accessing the network and services (Yaibuates & Chaisricharoen, 2020).

When a device (client computer) joins a new network, it assumes the default IP 0.0.0.0 before being assigned a valid IP address. The DHCP transaction process from the moment a client joins a network to the time it configures itself with valid IP parameters is summarized below (A. K. Rajput et al., 2016).

- A DHCPDISCOVER is the first message; it is broadcast to the network by the client and is meant for one or more DHCP servers,
- All DHCP servers on the network (subnet) will respond to DHCP Discover with a DHCPOFFER; this contains an unused IP address and TCP/IP configuration parameters like subnet mask and default gateway (Shete et al., 2018),
- On receiving the Offer, the client sends a DHCPREQUEST , which is a broadcast message meant to formally request to use the network configuration parameters received in the DHCP Offer message.
- Finally the DHCP server sends a DHCPACK message; it is a broadcast message from the DHCP server acknowledging use of the issued IP address.

In addition to these, DHCP client and DHCP server exchange additional messages such as those meant for IP address release, negative acknowledgment of the IP address, decline of IP address and a request for local configuration parameters. These are DHCPRELEASE, DHCPNAK, DHCPDECLINE and DHCPINFORM respectively (Vondráček et al., 2018).

With respect to DHCP processes, WLAN security is majorly affected by three internal attacks: rogue Access Point (AP), rogue DHCP server and unauthorized clients using static IP addresses. In a network, DHCP security is usually the most overlooked (CIOReview, 2016). DHCP is vulnerable to several attacks such as DHCP starvation attack, DHCP rogue server attack and malicious DHCP client attack (Younes, 2016).

Since DHCP servers interact with end devices at the point of entry to a network, rogue DHCP servers lead to a crop up of many cyber-attacks. According to research by Shete et al. (2018), these attacks include Man-in-the-Middle (MITM) attack, sniffing. Further, rogue DHCP servers allow an attacker to configure rogue Domain Name System (DNS) which could be used to design

phishing webpages that enable one to acquire confidential information (Shete et al., 2018).

In strengthening DHCP security, some of the solutions proposed are port security, and Dynamic ARP Inspection with DHCP snooping (Pradana & Budiman, 2021). Port security can be used to mitigate against spoofed MAC addresses while Dynamic ARP Inspection depends on DHCP snooping database (ritcsec, 2019). The database contains IP addresses of legitimate DHCP clients. However, the solutions proposed have still not addressed the presence of rogue DHCP servers.

1.2 Problem Statement

Due to the nature of DHCP communication, it is easy for an attacker to introduce a rogue DHCP server. This is possible since a client can receive DHCPOFFER messages from more than one DHCP server. If a rogue DHCP server is selected, then a client will be configured based on the configurations provided by the rogue server. Due to this, a DHCP client cannot know if the DHCP server assigning IP address is the legitimate DHCP server or a rogue DHCP server. There are available tools and mechanisms for detecting the presence of rogue DHCP servers in the network. However, these tools and mechanisms are inadequate. Some of them are only applicable in the LAN environment and others introduce complexity and traffic load on the DHCP protocol.

1.3 General Objective

The aim of this project is to develop a system that can detect and prevent rogue DHCP servers in WLAN.

1.4 Research Objectives

i. To investigate modes of launching rogue DHCP attacks on WLAN communication,
ii. To analyse the existing solutions used to detect and prevent rogue DHCP attacks on WLAN communication,
iii. To design, develop and test a model that detects and prevents rogue DHCP attacks on WLAN communication,
iv. To validate the performance of the developed model in detecting and preventing rogue DHCP attacks.

1.5 Research Questions

i. What are the modes of launching rogue DHCP attacks on WLAN communication?
ii. What are the existing solutions used to detect and prevent rogue DHCP attacks on WLAN communication?
iii. What approach can be used to design and develop a cost effective model that can detect and prevent rogue DHCP attacks on WLAN communication?
iv. To what extent does the model effectively detect and prevent rogue DHCP attacks?

1.6 Justification

The importance of this project is to secure the users in a WLAN environment. In a high traffic Wireless LAN, it becomes unreliable to depend on administrators to monitor network traffic and ensure requests and responses are issued by a valid DHCP server. A rogue DHCP server can provide incorrect network information which can lead to devastating attacks like MITM attack, sniffing or phishing. Through this, they can intercept private communication and redirect traffic to a rogue DNS server. The proposed model will provide reliability and availability by preventing rogue DHCP attacks on WLAN communication.

1.7 Scope and Limitations

The focus of this study will be on rogue DHCP servers facilitating rogue DNS servers. The study will not cover DHCP starvation attack, presence of rogue APs and rogue default gateways. The implementation will be on a Linux platform and testing will be done on a test bed within the Strathmore University network. The challenges that might arise range from network down time to network fluctuations. This might affect the accuracy during data collection.

Chapter 2: Literature Review

2.1 Introduction

The previous chapter introduced rogue DHCP attack as an important subject with regard to security in wireless network communications. This chapter presents a review on security in DHCP operations, with key focus on IEEE WLAN communications.

2.2 DHCP Overview

According to definitions and descriptions of RFC 2131, the Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol (BOOTP), which provides the framework for passing configuration information to hosts on a TCP/IP network. The protocol relies on User Datagram Protocol (UDP). The UDP port number used by the server is 68 while the port number used by the client is 67 (Yaibuates et al., 2016). DHCP implements two mechanisms for IP address allocation. In the first mechanism, the network administrator assigns an IP address to an end device while in the second, an IP address is assigned to a client on a lease, that is, a limited period of time. The architecture of DHCP is composed of a DHCP server and a DHCP client (Yaibuates & Chaisricharoen, 2019). A DHCP server is responsible for allocating network configuration parameters to a DHCP client and a DHCP client runs a DHCP client program that wants to connect to a network.

2.2.1 Factors Influencing Use Of DHCP Protocol

A DHCP server is responsible for unique dynamic assignment of IP addresses in a network. This dynamic assignment eases the cumbersome nature of manual network administration. In addition to this, it reduces typographical errors that often lead to one IP address being issued to more than one network device. Such an error would result in a Denial of Service as the devices can no longer communicate in the network.

Using academic institutions as the focus of study, there is an enormous number of devices used within a university's network and all these devices need to communicate. It becomes almost impossible to allocate IP addresses manually, hence the need to use DHCP, which is automated. DHCP server dynamically assigns network configuration information for a specified lease

duration. The DHCP server maintains a database of assigned IP addresses and a mapping to the hosts that are allocated IP addresses. This enables devices to join a network and exchange messages with other devices in the network.

2.3 The DHCP Transaction Process

DHCP is an internet protocol that automatically assigns IP addresses and other network information to end devices that use the TCP/IP protocol suite. The network information includes client IP address, subnet mask, default gateway IP address, DNS IP address and IP lease duration (A. K. Rajput et al., 2016).

The transaction messages are illustrated in Figure 2.1.

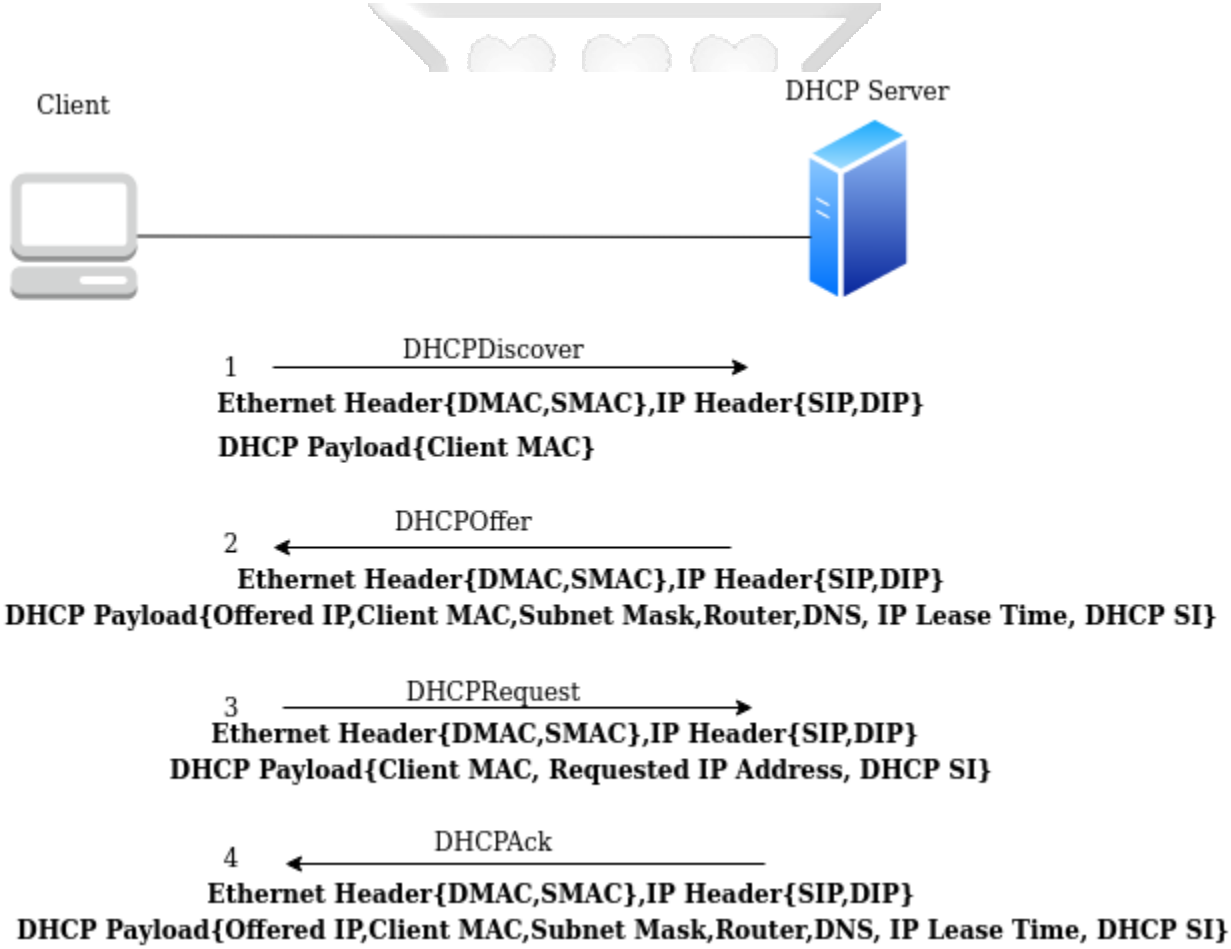


Figure 2. 1 Summary of DHCP Transactions

DHCP Discover: It is a broadcast message (Destination MAC Address=FF:FF:FF:FF:FF:FF) sent by the client device to a DHCP server when booting or entering a new network (A. K. Rajput et

al., 2016). It is a request message for an IP address. DHCP servers capture this information while other hosts on the network discard the message.

DHCP Offer: The DHCP server broadcasts a DHCP Offer message over the Ethernet network on receiving the DHCP client message**.** This message contains network information namely: client IP address, subnet mask, default gateway IP address, DNS IP address and IP lease time. This message is sent to all hosts in the same subnet including the client that sent DHCP Discover (Satapathy A., 2018).

DHCP Request: This is an echo message sent by the client containing information of the DHCP Offer message sent by the server as a confirmation. The message is usually a broadcast so as to notify the not selected servers to add the offered address to the reserved address (A. K. Rajput et al., 2016).

DHCP Acknowledgement: The DHCP server receives the DHCP request message from the client, it confirms the server identifier field and broadcasts a DHCP Ack to ensure the client can receive the message. During this instance, the DHCP server transfers all network configuration to the client. The client then configures their interface.

There are other special DHCP messages namely: DHCP Decline, DHCP Release, DHCP Renew and DHCP Negative Acknowledgement. **DHCP Decline** is a special message sent by a client to the server if the IP address is invalid or being used by another host in the network. **DHCP Negative Acknowledgement** is a special message sent by the server when the lease time for the client has expired.

2.4 Security Considerations of DHCP

During the design and development of DHCP, security was not considered a critical issue (Yaibuates & Chaisricharoen, 2020). DHCP is built on UDP and IP which are characteristically insecure (Droms, 2021). In addition to this, DHCP messages are transmitted in plain text and there is no authentication of DHCP messages. According to (Yaibuates & Chaisricharoen, 2020), this lack of security concerns has posed many network threats such as the setting up of rogue or unauthorized DHCP servers which acts as a DHCP server to provide network configurations to other legitimate client devices in the network. As a result, attackers might deny, capture, modify and analyze packets that contain private information or secret password that has been sent from a client (Yaibuates & Chaisricharoen, 2019).

2.5 DHCP Attacks

DHCP attacks are broadly classified into two: DHCP server attacks and DHCP client attacks.

2.5.1 DHCP Client attacks

A) DHCP Starvation Attack

A malicious client can gain unauthorized access in a network and launch a DHCP starvation attack. A DHCP starvation attack is considered a DoS attack (Yaibuates & Chaisricharoen, 2020). According to research study by Tripathi & Hubballi (2015), the attack can be launched by generating random MAC addresses and then sending a large number of DHCPDISCOVER messages using these MAC addresses. The study notes that the DHCP server releases a new IP address for every DHCP discover message received thus eventually running out of available IP addresses in the pool. However, it is not easy to launch a DHCP starvation attack using random MAC addresses in wireless networks due to an IEEE 802.11 WPA2 standard that requires an association phase of a wireless client with an AP (Tripathi & Hubballi, 2017).

2.5.2 DHCP Server attacks

A) DHCP Spoofing

DHCP spoofing is an attack that occurs when an attacker attempts to respond to DHCP requests by identifying as the valid DHCP server (Ornaghi & Valleri, 2003). According to Shete et al. (2018) research study, DHCP spoofing occurs after setting up a rogue DHCP server onto the network. The study notes that once an illegitimate server receives all the data packets and traffic, it can spoof responses sent by the legitimate DHCP server. The attack uses the ARP spoofing technique (Shete et al., 2018).

B) DHCP Lease Attack

IP lease time is among the network information issued by the DHCP Offer message. This manages the period of validity for each IP. A client sends a DHCP release packet to release the IP when it is no longer needed. In contrast with valid clients, hackers send forged DHCP Request packets and lease all the available IP's thus making free IP unavailable (Wang & Chen, 2017).

C) Rogue DHCP Server

A rogue DHCP server refers to an unauthorized server introduced to a network. An attacker can set up a rogue DHCP server in a network which can masquerade itself as the legitimate DHCP server. If the response time of the rogue is shorter than that of the legitimate DHCP server, client devices will receive DHCP Offer messages from the rogue DHCP server first. Through this, victim clients can be assigned invalid network configurations such as invalid IP Address, invalid DNS IP address and invalid Gateway IP address. Dinu & Togan (2014) research study states that if the provided configuration redirects to the wrong address, then it is a DoS attack as long as clients are not able to access the network. Further, the study notes that provision of fake gateway addresses by the attacker can redirect traffic to a controlled machine that can sniff traffic and realize MITM attack. In addition, they state that a more complex attack can be done by provision of fake DNS server addresses using a faithful copy of a real page (Dinu & Togan, 2014). Through this, the attacker can launch phishing attacks. To increase the chances of a successful Rogue DHCP server attack, DHCP client attacks involving scope exhaustion on the legitimate DHCP server are often conducted first. Rogue DHCP server attack could also be achieved through a sniffing attack. In this attack, the rogue DHCP server listens to all DHCP Discover messages sent by a client and responds with a DHCP Offer message. In most cases, the DHCP client accepts the first DHCP Offer message.

2.6 Ways of Launching Rogue DHCP Server Attack

According to research study by Mukhtar et al. (2012), the authors state that for a rogue DHCP server attack to occur, a DHCP starvation attack ought to have occurred prior to the rogue DHCP server attack. A DHCP starvation attack is considered a Denial of Service (DoS) attack as the attack prevents legitimate clients and devices from accessing the services in the network. Scapy, Gobbler and Yersinia are well-known tools used to launch DHCP starvation attacks (Yaibuates & Chaisricharoen, 2019).

After launching a DHCP starvation attack, the attacker sets up a rogue DHCP server. According to research by Tanceska et al. (2015), rogue DHCP server setup is done by setting up a network subinterface on the attackers machine to be used as a default gateway to reroute the clients. The study notes that an IP address is assigned to the new subinterface and IP forwarding enabled.

Further, the subinterface is set as the default gateway and default route (Tanceska et al., 2015).

2.7 Mitigating Rogue DHCP Server Attacks

Existing approaches to mitigate rogue DHCP server attack include DHCP snooping, use of digital certificates and authentication (Agarwal et al., 2019). The methods are discussed further below.

Secure DHCP System with User Authentication

Komori & Saito (2002) proposes a system composed of 8 sub-units that prevents unauthenticated users from obtaining an IP address. Legitimate DHCP clients ought to register in advance the ID and password with the authentication server. According to Dinu & Togan (2014) , the proposed system is complex to manage for a large number of clients given the use of shared secret between clients and the authentication server.

A Unified Approach to Intra-domain Security

Shue et al. (2009) proposed a secure DHCP protocol based on the deployment of digital certificates in that both client and server need to have their own digital certificates installed before the assignment of configuration information commences. They further state that use of digital certificates and digital signatures attached to each message can guarantee the identities of clients and servers. The proposed technique introduces new DHCP options and additional communication between the authentication service and client, and assumes that in some networks DHCP packets are split and sent separately (Dinu & Togan, 2014).

Securing Network Location Awareness with Authenticated DHCP

The mechanism proposed by Aura et al. (2007) is based on use of public key cryptography to authenticate networks. Further, it ascertains that DHCP was chosen for NLA since it is the first protocol run by a node when entering a new network. However, the proposed method does not state how the public key or digital certificate of the server is transmitted (Dinu & Togan, 2014).

Design and Implementation of DHCP Using Symmetric Encryption

Yun Yang & Jia Mi (2010) proposes a technique based on shared keys that could counter DHCP starvation attacks. Their technique focused on the problem of MAC address spoofing during DHCP client authentication. They suggested the use of CPU ID and Disk value as parameters for

DHCP client authentication instead of MAC address. According to Duangphasuk et al. (2011), these values are static and can be intercepted over the network. Additionally, the technique is based on symmetric cryptography yet no information has been given with regards to updating the shared key between client and server. Encrypting a message using the same key in every transaction is vulnerable to traffic analysis (Duangphasuk et al., 2011).

Using Digital Certificates to Secure DHCP Communication

A study by Duangphasuk et al. (2011) proposes two solutions for securing DHCP: Secure DHCP with Digital Certificates and Secure DHCP with Shared Secrets. This method verifies messages sent by a client to ensure legitimacy. According to Wang & Chen (2017), this solution complicates DHCP implementation in server and client. The digital certificate size could exceed the DHCP message size (Dinu & Togan, 2014).

DHCP Authentication Using CHAP

The patent by (Graaf et al., 2011) describes a mechanism of authenticating a client using a RADIUS server. The authentication mechanism uses a shared secret between the DHCP client and authentication server. The DHCP client and the authentication server compute the DHCP server challenge response using a hash function (Graaf et al., 2011). The proposed method introduces additional communication between the DHCP server, client and authentication server.

DHCP Server Authentication Using Digital Certificates

(Dinu & Togan, 2014) proposed a technique that authenticates the DHCP messages and DHCP server using digital certificates and public key cryptography that can assist in preventing rogue DHCP server attack. However, this technique is vulnerable to DHCP starvation attack (Younes, 2017).

DHCP Message Authentication Module

(Dinu & Togan, 2015) proposes a scheme called DHCPAuth which was based on two trust models: PGP and PKI. These trust models have drawbacks which make them insecure (Younes, 2017).

Port Security

Port security restricts the number of MAC addresses attached to a switch's port. According to

Tripathi & Hubballi (2017), stealth DHCP starvation attack does not require MAC address spoofing. Thus, the attack can go undetected.

DHCP snooping can mitigate DHCP starvation attacks by checking the MAC address against the DHCP Discover message header (Tripathi & Hubballi, 2017). However, the study notes that it cannot detect stealth DHCP starvation attacks since the attack does not involve manipulation of the DHCP header.

DAI with DHCP Snooping determines validity of ARP messages by checking IP-MAC mappings on the DHCP snooping database (Tripathi & Hubballi, 2017). However, clients that have manually assigned IP addresses do not have mappings on the DHCP snooping database hence undetectable by DAI.

Table 2. 1 Comparison of DHCP Security Implementations

Model	**Strengths**	**Weakness**
Cryptographic Techniques	Mitigates identity spoofing based attacks including DHCP stealth starvation attack.	Implementation is cumbersome. Requires network administrator's intervention. Cryptographic techniques add complexity and traffic load by using third party modules like Authentication server.
Port Security	Limits the number of MAC addresses attached to a port. Prevents MAC address spoofing.	Port security cannot detect a malicious client with one specific MAC address.
DHCP Snooping	Detects manipulation of DHCP	Undetected attacks that do not require header

	message header.	manipulation like DHCP stealth starvation attack
Dynamic ARP Inspection with DHCP Snooping	Validates ARP message by checking valid IP-MAC binding corresponding to what is stored in the DHCP snooping database	Client machines with manually allocated IP addresses are undetected as the mappings are not contained in the snooping database. ARP requests sent by client machines mimic a new client who has joined the network e.g. IP "0.0.0.0"
DHCP Traffic Threshold Based Mechanism	Detects DHCP Requests exceeding set threshold	The DHCP messages generated by the attacker are less than the classical DHCP starvation attack.

Note. Adapted from "Detecting Stealth DHCP Starvation Attack using Machine Learning Approach" by N. Tripathi & N. Hubballi, 2017, *Journal of Computer Virology and Hacking Techniques, 14*.

Chapter 3: Research Methodology

3.1 Introduction

This chapter describes the methods used to address the research questions indicated in chapter one. To start with, it answers the question on the modes of launching rogue DHCP attacks on WLAN communication and the existing solutions used to prevent rogue DHCP attacks on WLAN communication. The chapter then proceeds to explore how data was collected and analyzed, research methodology, research design and ethical considerations. The research questions stated in chapter 1 were addressed as follows:

Research Question 1

What Are The Modes of Launching Rogue DHCP Attacks on WLAN Communication?

This research question has been addressed by the research study provided in the literature review in chapter 2. The research study shows the various scenarios by which rogue DHCP server attacks can be launched.

Research Question 2

What Are The Existing Solutions Used to Detect and Prevent Rogue DHCP Attacks on WLAN Communication?

This research question has been addressed by providing related work done by other researchers in the literature review in chapter 2. The various solutions provided include: use of digital certificates and encryption.

Research Question 3

What approach can be used to design and develop a model that can detect and prevent rogue DHCP attacks on WLAN communication?

This research question has been addressed by the system methodology which is presented later in the chapter. The system methodology shows the methods used by the researcher to meet the research objectives.

Research Question 4

To what extent does the model effectively detect and prevent rogue DHCP attacks?

This research question is meant to show the validation of the researcher's project. This has been tackled by a series of system tests indicated later in this chapter.

3.2 Research Design

The research employed an experimental setup. It involved setting up a test bed in a WLAN environment. The environment encompassed several components. These were one rogue DHCP server, two legitimate DHCP servers, DHCP clients, packet sniffer and an Intrusion Detection System for detecting rogue DHCP servers. Traffic generated from rogue DHCP server was distinguished from legitimate DHCP servers by observing the differentiated DHCP Acknowledgement unicast message from both servers. This was done by analysing the traffic received by DHCP clients from both DHCP servers.

3.3 Software Methodology

The chosen method for this study is Test-Driven Development. Test-Driven Development is an agile methodology that encompasses software development by repetitively turning the requirements of a system into a test case (Burris, 2017). According to a study by Burris (2017), TDD is divided into these steps:

1. Feature requirements
2. Write unit tests
3. Run test and if it succeeds, add another test by repeating step 2
4. Write a code that implements the unit test.
5. Run tests and if it fails, repeat step 4
6. Refactor code

Figure 3.1 below shows the TDD cycle:

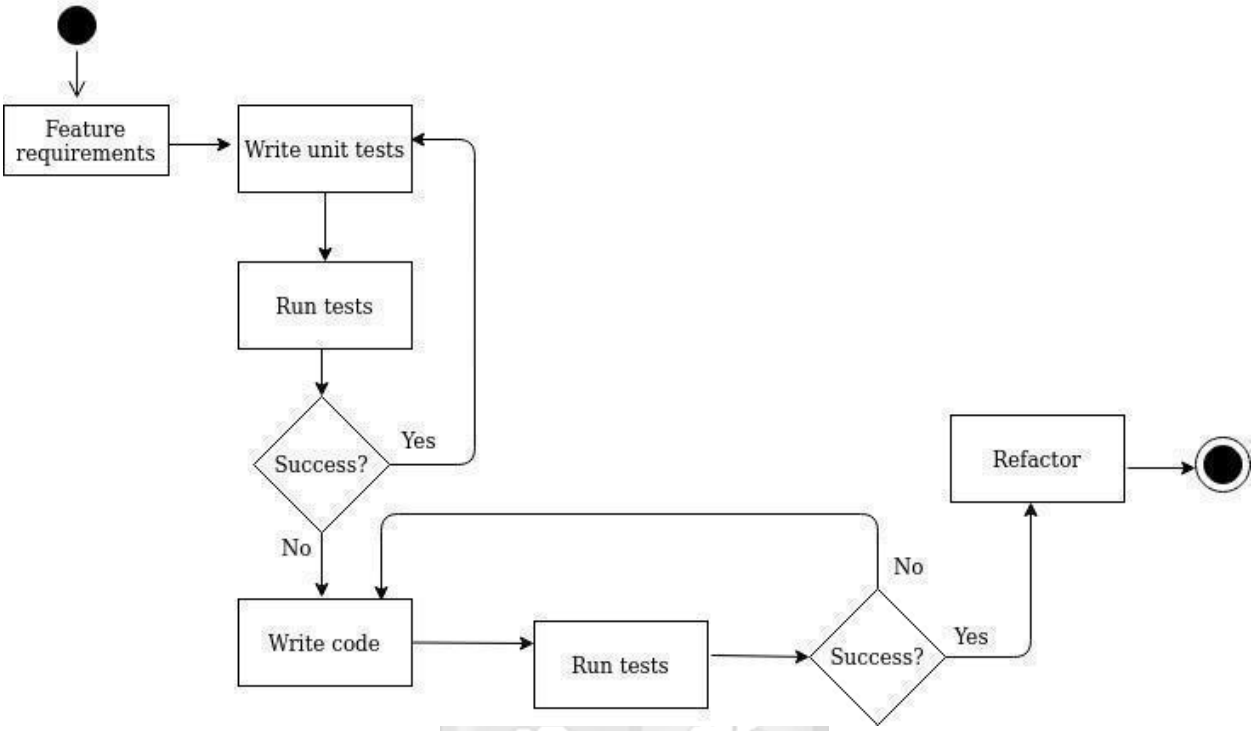


Figure 3. 1 TDD Life Cycle

3.3.1 Feature requirements

This process was used in identifying the system components for the experimental setup and system parameters. The system components were as follows: authorized DHCP servers, rogue DHCP server, DHCP clients, access point, packet sniffer. The components were assembled and integrated to ensure continuous end-to-end data flow. The system parameters were obtained from the differential DHCP transaction messages observed, examined and analysed between the authorized and rogue DHCP servers.

A feasibility study was done to ensure the requirements are specific and attainable. The requirements obtained were analysed for validity.

3.3.2 Write Unit Test

The researcher wrote functionality tests for each system component. The tests encompassed installation and configuration of each individual component in the experimental setup. The DHCP server test consisted of installation and configuration of both legitimate and rogue DHCP servers in a WLAN network. The WLAN network encompassed an Access Point (AP) and the test for this

was centered around ensuring wireless access to the network for both legitimate and rogue DHCP servers. The DHCP client test served the purpose of ensuring clients were assigned IP addresses from the DHCP servers in the WLAN network.

3.3.3 Run Test

The researcher did the functionality tests indicated in the previous phase to ensure each individual component was functioning as expected and can communicate with other components. DHCP servers and DHCP clients were launched in the WLAN network and the traffic was inspected to ensure the IP addresses issued to the clients were from the configured authorized and rogue DHCP servers. Wireshark, a packet sniffer was used to observe, examine and analyze the DHCP packets. It is from this analysis that the system parameters were obtained.

3.3.4 Write code

The researcher wrote functional code based on the system parameters that were acquired after the packet analysis. The system parameters included the DHCP messages and the various DHCP option fields that distinguish the DHCP transaction messages discussed in the literature review. These option fields contain information about the DHCP server that issued the network configurations to the client. These formed the input that was used to detect and prevent the presence of the rogue DHCP servers in the WLAN network.

3.3.5 Run Test

The authorized and rogue DHCP servers were launched in the network. Also, the IDS was fired up. This was used to detect the presence of a rogue DHCP server in the WLAN network. When rogue DHCP server traffic was detected in the network, alerts were generated and displayed in the terminal. When a legitimate DHCP server was detected, information regarding the same was displayed in the terminal. In addition to this, a detailed report of the activities was generated. This confirmed that the test passed. If the test failed, code adjustments were made in order to meet the expected functionality.

3.3.6 Refactoring

At this stage, the researcher performed regression testing whereby previous tests were done to ensure no bugs were introduced during implementation. Also, the design module was improved

by removing duplicates and inconsistencies.

3.4. System Analysis

System analysis involved identifying ways to break down a large problem into small components which were further broken down into smaller components. According to research done by Waldo (2006), these small components undergo iterations until they reach a point where a problem can be solved on its own. The decomposition process that breaks down small components to smaller components, assists in coming up with the system components, and deciding what these components are and how they are interoperable is what is referred to as system design (Waldo, 2006). Once components of the platform were identified, system components were drawn from each of these components which facilitated one to describe the functionalities and come up with the system design of the research study.

3.5 System Testing

The following test cases were conducted to ensure that the system works as expected;

3.5.1 Usability Testing

The user interface of the model was developed using bash script. It was simple and easy to navigate thus allowing the user to input data and interact with the model efficiently.

3.5.2 Performance Testing

The purpose of this test was to ensure that the system can perform its functions and tasks as envisioned by the researcher.

Clients and legitimate DHCP servers: Two legitimate DHCP servers were set up and launched in a WLAN environment. One of the legitimate DHCP servers was set up by use of the ISC DHCP and the other was pre-installed in the Access Point (AP). ISC DHCP is a collection of software that implements all aspects of the DHCP suite namely: DHCP server, DHCP client and DHCP relay agent. In addition to this, the DHCP servers were assigned an address scope of twenty five addresses each. To simulate DHCP clients, twenty dummy clients were introduced by use of a python script based on the scapy module. Scapy is a packet manipulation program that can forge packets of a wide number of protocols among many other uses (Biondi & Scapy community,

2021). Once the clients were introduced, the Wireshark tool was used to observe DHCP transaction messages between the DHCP servers and clients. The DHCP transaction messages observed were: DHCP Discover, DHCP Offer, DHCP Request and DHCP Acknowledgement. Based on the setup, all clients were expected to receive IP addresses since the size of the address scope of each DHCP server was twenty five hence 50 in total.

Rogue DHCP server and two legitimate DHCP servers: One rogue DHCP server was introduced in the set up by installing the ISC DHCP server on a virtual machine installed in the host computer. The IP address of the rogue DHCP server was acquired by doing a non-intrusive network sweep on the network of the genuine DHCP servers in order to find gaps and get unused IP addresses. Also, an obfuscation attack was done that entailed assigning an IP address that looked similar in structure at first glance to an IP address assigned to a genuine DHCP server. The python script for generating dummy clients was launched again so as to determine the number of clients that received IP addresses from the rogue DHCP server. The time taken for each client to request an IP address was eight seconds. Wireshark was used to determine the number of DHCP servers that sent Offer messages. On observation, it was ascertained that three devices with different IP addresses sent offer messages.

One legitimate DHCP server, one rogue DHCP server: Another test was performed with one legitimate server and rogue DHCP server in order to compare the number of clients served by rogue DHCP server with relation to those served in a set up that involved two legitimate DHCP servers.

Detecting the parameters issued by rogue DHCP servers: This was done by using a configuration file from which an IP and MAC address of the legitimate DHCP server was entered. Then, the system detected a rogue DHCP server by comparing the IP and MAC address of DHCP servers responding to the DHCP client request with that included in the configuration file. IP and MAC addresses were used since they constituted the primary identity of DHCP servers in the network. If the parameters did not match, then it was an indication of a rogue DHCP server attack. This generated alerts from which a report was created afterwards. The report contained configuration file information which had the IP and MAC address of the legitimate DHCP server

; DHCP transaction messages between clients and DHCP servers; output of the IDS which indicated whether the server in operation is legitimate or rogue; and options of the DHCP server (subnet mask, lease time, DNS server information and IP address of the router). The options value provided more information that a DHCP client would needs to communicate in a network in addition to the IP address assigned by the DHCP server. For instance, the router provides a gateway to the internet. However, these options are not vital in the detection algorithm hence they were not factored in.

3.6 System Validation

Performance testing was done to ensure reliability, availability and consistent performance of the proposed model. The metric used for measuring the performance of the IDS was the detection rate. Detection rate was defined as the number of Offer messages sent by the rogue DHCP server to the number of Offer messages detected by the IDS. The IDS detected Offer messages sent by rogue DHCP servers and raised alerts. As for the legitimate DHCP server, the IDS provided an information message stating that the expected DHCP server was in operation. Another metric was the distribution of clients among DHCP servers. This was defined as the number of clients that accepted IP configuration parameters from each legitimate or rogue DHCP server.

3. 7 Ethical Considerations

This research involved an experimental setup in a test network, that is, a hotspot environment. The setup was an isolated environment to prevent interference from and to other subnets or networks. The attacks launched in the setup did not target other machines nor other networks.

Chapter 4: System Analysis and Design

4.1 Introduction

This chapter describes the steps taken by the researcher during the design phase of the system. It starts by looking into the requirement analysis, which involves a discussion of the functional and non-functional requirements. The chapter outlines the architecture design to detect and prevent rogue DHCP server attacks.

4.2 Requirement Analysis

Requirement analysis enables one to identify the appropriate information or resources that will meet the objectives set, which is, A Model to Detect and Prevent Rogue DHCP Attacks on WLAN Communication effectively. In order to attain this, each requirement ought to be broken down and defined clearly. A review of the flow of events as per interaction of each requirement is performed in order to assist in decision making of whether a requirement is needed or not. Requirement analysis enables creation of a development framework. System requirements can be categorized into two: Functional and Non-functional requirements.

4.2.1 Functional Requirements

These entail the system's functionality. They form the desired functionality that the user expects from the system. The functional requirements are as follows:

i. The system should be able to allow users to create custom rules on Snort.
ii. The system should be able to detect rogue DHCP server traffic on WLAN environments.
iii. The system should be able to store activity logs in a database.
iv. The system should allow BASE to interface with Snort to display the rogue DHCP server alerts on a web UI.

4.2.2 Non-Functional Requirements

The Non-Functional requirement are as follows:

i. The model should be accurate to increase reliability.
ii. The model should be scalable.
iii. The model should be always available. This will ensure that it is always checking the DHCP traffic during the client IP configuration process.

4.3 The System Diagram

The design comprises the system architecture, use case diagram and the System Sequence Diagram. These are presented in the below subtopics.

4.3.1 System Architecture

Figure 4.1 below shows the system architecture of the proposed model that detects and prevents rogue DHCP server attacks. As seen there are two DHCP servers: the first being the genuine DHCP configured in the AP and the second one is the rogue DHCP server. The rogue DHCP server is set up on a machine running Kali Linux operating system with an isc-dhcp-server package that provides a DHCP server environment. The AP provides the WLAN setup. The IDS runs on an Ubuntu machine that is wirelessly connected to WLAN so that it can capture all DHCP packets exchanged among the clients and the DHCP servers in the setup. Also, the IDS is implemented in python using the scapy library that features deep packet inspection capabilities. The clients depicted in the architecture represent desktops and laptops that have Wi-Fi cards.

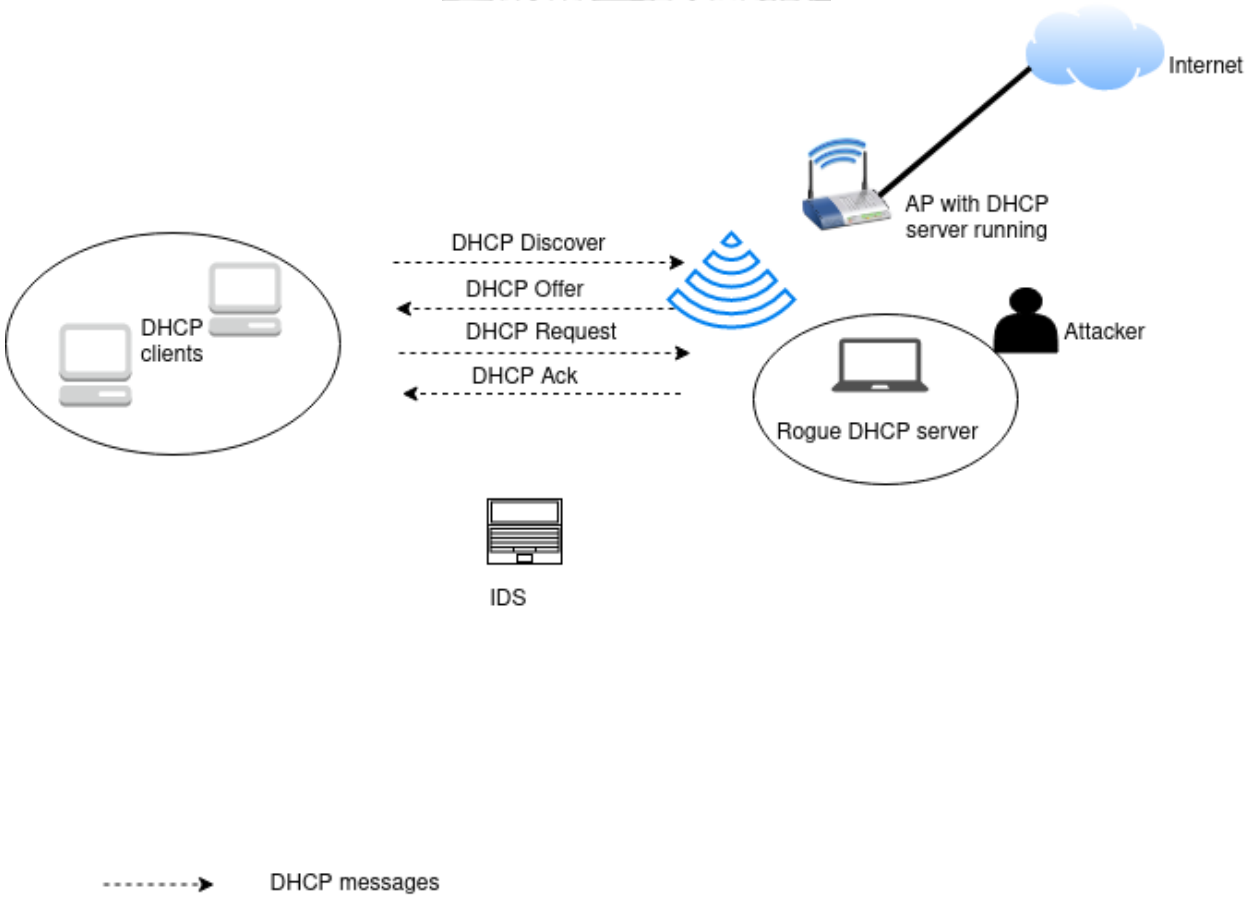


Figure 4. 1 System Architecture

From figure 4.1, all traffic exchanged between the DHCP clients and servers is monitored for rogue DHCP server attacks by the IDS. The traffic is then analyzed to identify IP and MAC address of a rogue DHCP server by comparing it to the DHCP server information entered in the configuration file. Based on this information, the IDS decides whether a rogue DHCP server attack is in progress or not.

The model takes response action to prevent the attack by blocking the IP address of the rogue DHCP server. This is done by an Uncomplicated Firewall (UFW) which is an inbuilt firewall in linux environments. It contains IPTABLES by which simple firewall rules such as blocking of the source IP address of the rogue DHCP server can be implemented (Kyambadde & Ngubiri, 2018). Figure 4.2 below shows the activity flow of the prevention mechanism.

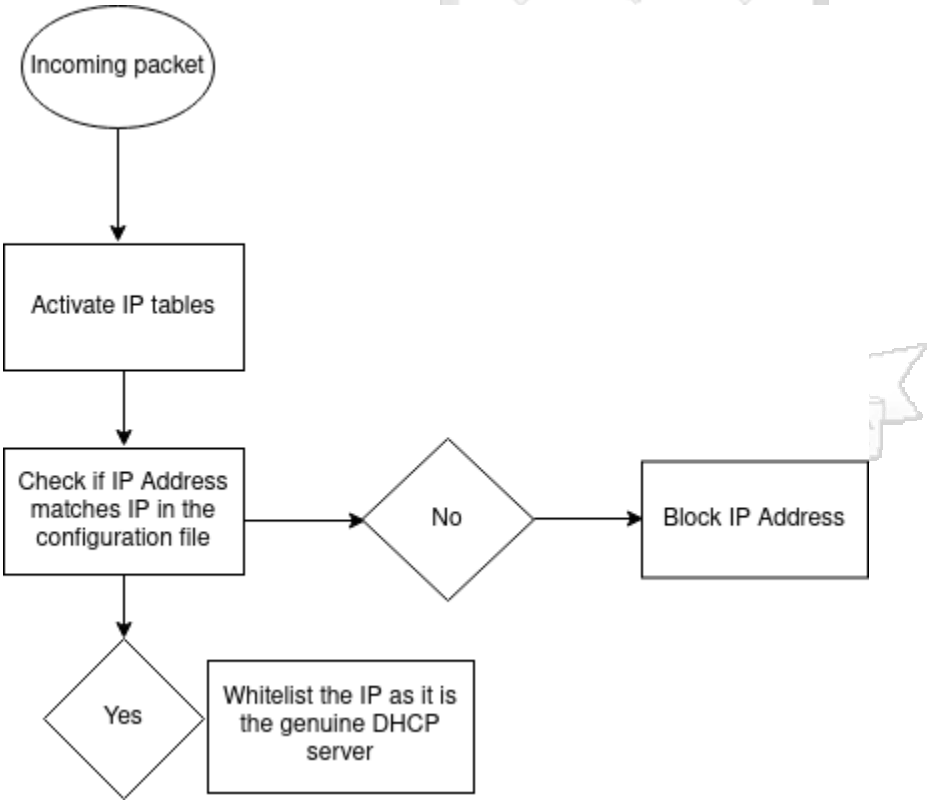


Figure 4. 2 Activity Flow of the Prevention Mechanism

4.3.2 Use Case Diagram

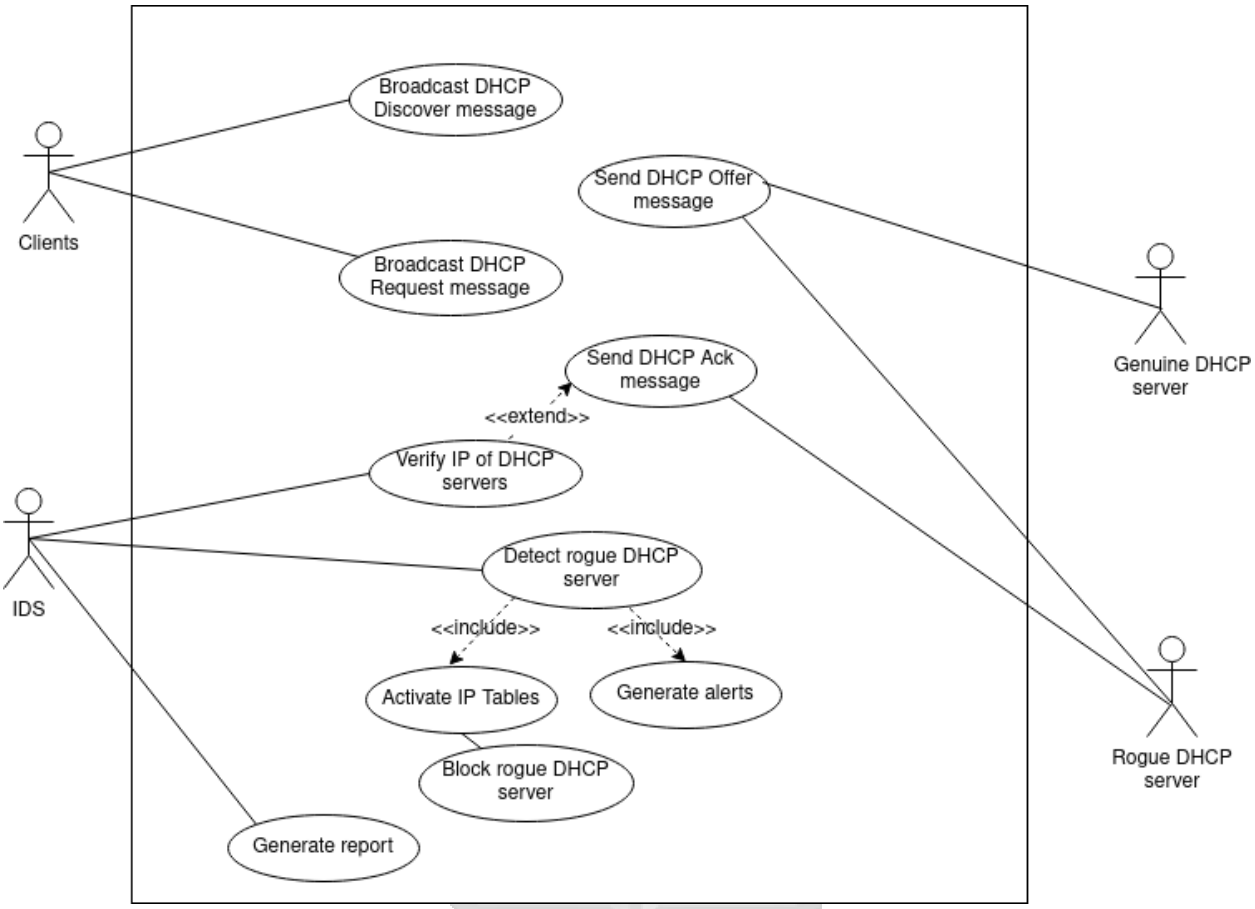


Figure 4. 3 Use Case Diagram

Figure 4.3 above shows the use case diagram. The clients initiate the DHCP transaction process by broadcasting the DHCP Discover messages. Both genuine and rogue DHCP server respond with DHCP offers. The client chooses one offer based on the offer that arrived first and broadcasts a DHCP request message. The server chosen responds with a DHCP acknowledgement message. All these messages are monitored and analysed by the IDS which makes the decision based on the server identification option field provided in the DHCP Ack message. If the chosen server is a rogue DHCP server, the IDS flags this as a rogue DHCP server attack and activates IP tables which block the IP address of the rogue DHCP server.

4.3.3 System Sequence Diagram

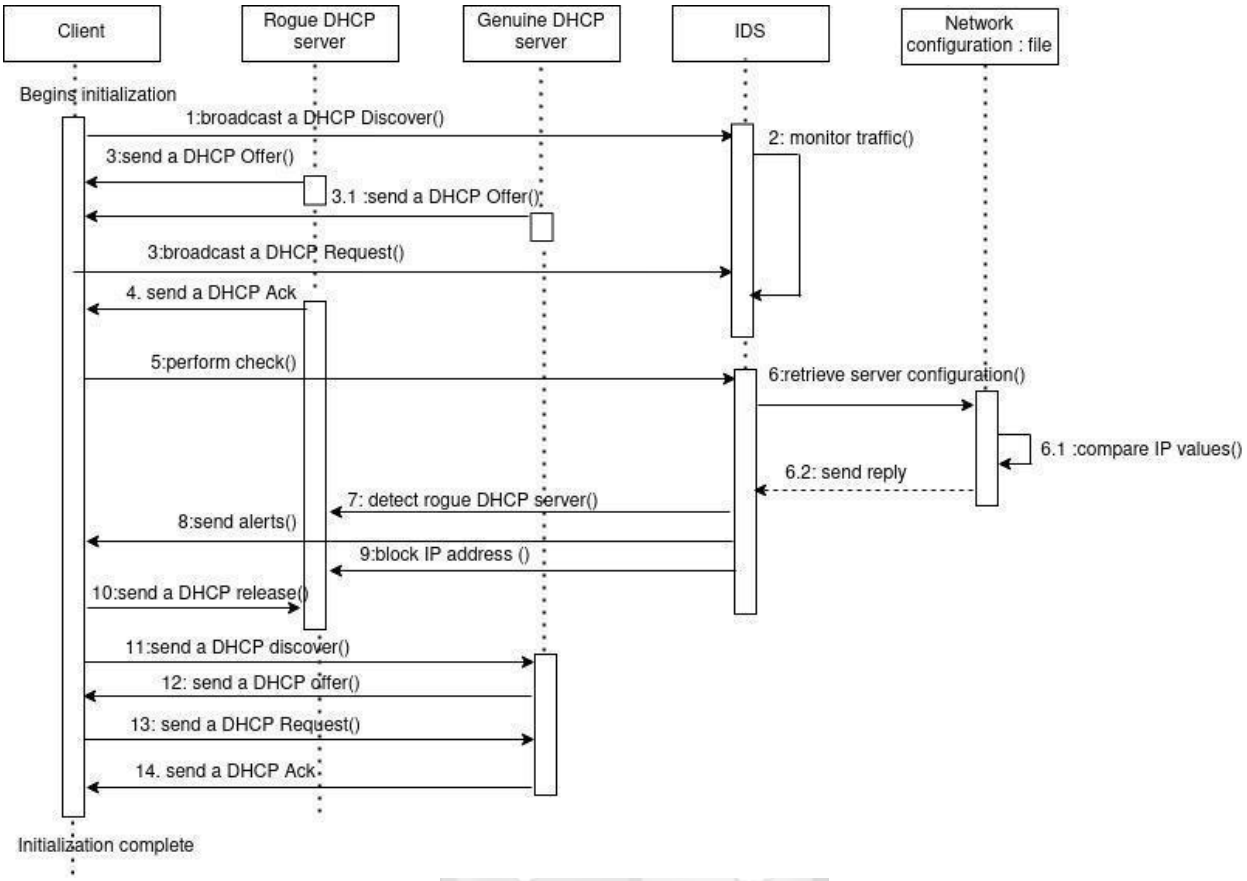


Figure 4. 4 Sequence Diagram

Figure 4.4 above demonstrates the DHCP transaction process and how the transaction traffic is monitored and analysed by the IDS. This is done by comparing the DHCP server configuration parameters in the configuration file with that of the servers offering IP addresses to the client. After which a rogue DHCP server is detected and blocked.

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter discusses the implementation of the model used to detect and prevent rogue DHCP servers over Wi-Fi networks. The chapter describes the integration of system components and configuration of the test network. In addition to this, it will include screenshots of the system interface and tests carried out in the system.

5.2 Implementation Environment

5.2.1 Hardware Requirements

The requirements needed for the optimal performance of the model are:

1. Tp Link 3g/4g wireless n router tl-mr3420- Access Point with inbuilt DHCP server
2. 2 laptops with Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz
3. 12 Gigabyte (GB) Random Access Memory (RAM)
4. 250 GB Hard Drive Disk Storage
5. 2 HP desktop machines to act as clients

5.2.2 Software Requirements

The software requirements used were as follows:

1. Ubuntu 20.04 host Operating System - Running the IDS
2. Kali Linux 5.9.0 kali1-amd64 - Running on attacker's machine
3. ISC-DHCP-server v4.4.2-P1 - rogue DHCP server
4. Scapy v2.4.5.
5. Python 3.8.10
6. Bash version 5.0.17

5.3 System Components

5.3.1 User Input

The developed model contained sections of user input for instance, the creation of the configuration file which involved providing the IP address, MAC address of the legitimate DHCP server and the interface to be monitored. Figure 5.1 and 5.2 below shows a screenshot of some of the parameters that form the configuration file.

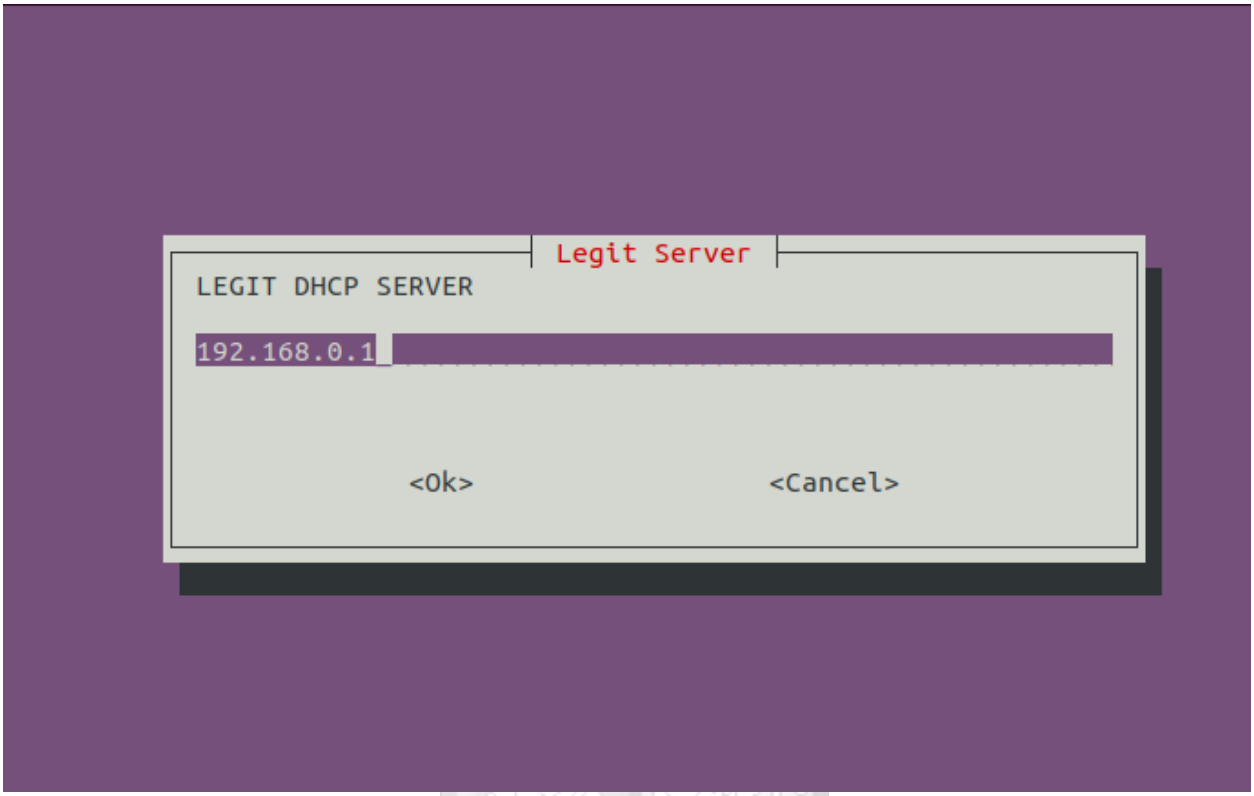


Figure 5. 1 IP Address of the Genuine DHCP Server

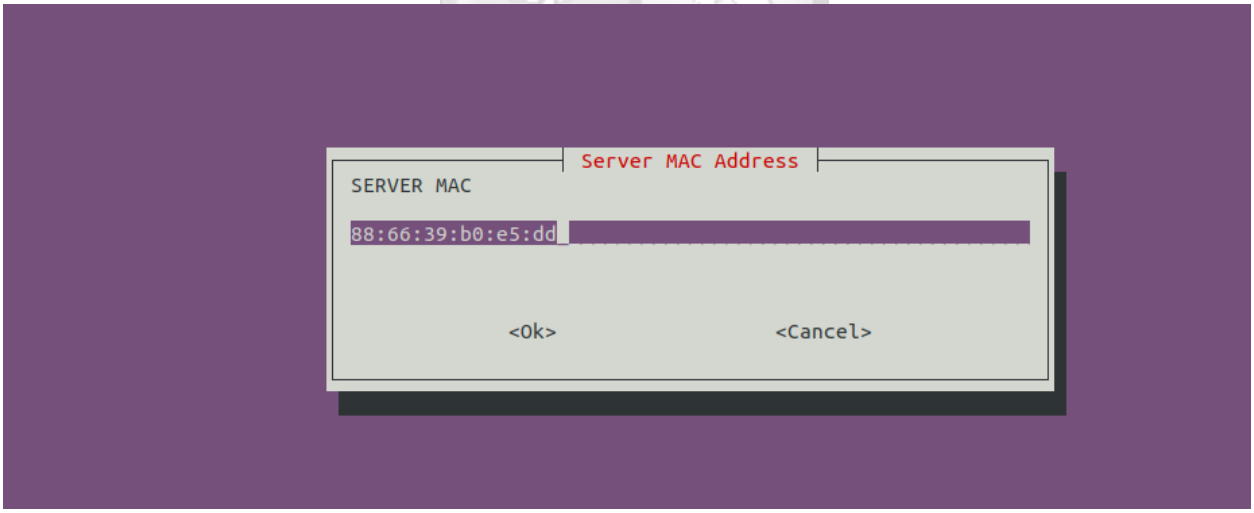


Figure 5. 2 MAC Address of the Genuine DHCP Server

5.3.2 User Interface

In order to enhance usability, a user interface was developed using bash script. The interface had a menu through which a user could easily navigate. Figure 5.3 below shows the user interface.

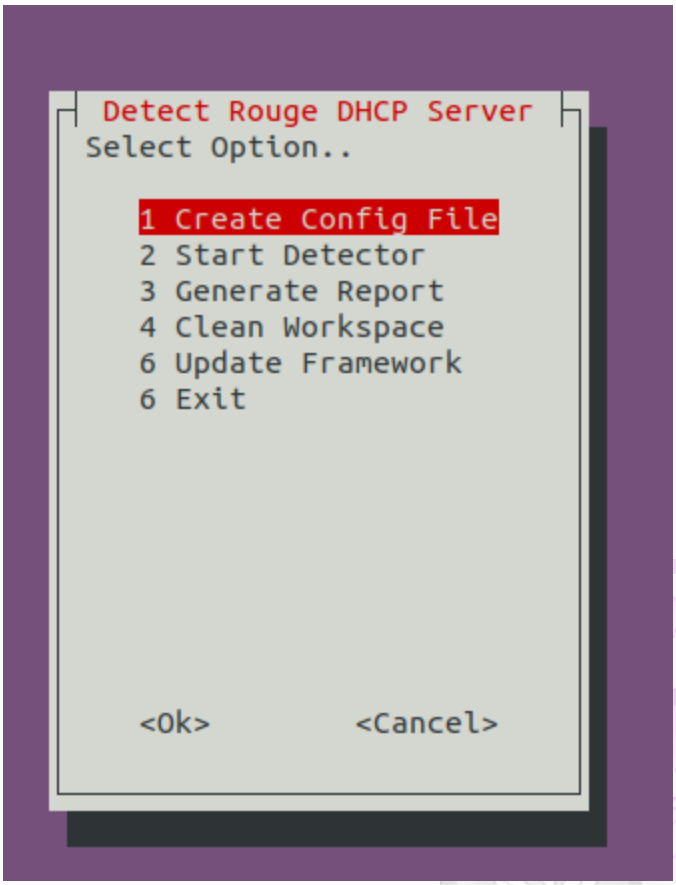


Figure 5. 3 User Interface

5.4 System Implementation

The implementation was done by building an IDS using scapy module which is a Python program and coupling it up with bash script for the user interface. To start with, the researcher installed version 20.04 of the Ubuntu Operating System. Then, Python3 and scapy were installed on the machine.

Afterwards, the researcher created DHCP packets using scapy and iterated through the option fields in order to retrieve the DHCP transaction messages discussed in the literature review. This was vital as the messages contained DHCP server information. Figure 5.4 below shows a code snippet of the scapy implementation.

```
elif DHCP in packet and packet[DHCP].options[0][1] == 2:
    print('---')
    print('New DHCP Offer')
    #print(packet.summary())
    #print(ls(packet))

    subnet_mask = get_option(packet[DHCP].options, 'subnet_mask')
    lease_time = get_option(packet[DHCP].options, 'lease_time')
    router = get_option(packet[DHCP].options, 'router')
    name_server = get_option(packet[DHCP].options, 'name_server')
    domain = get_option(packet[DHCP].options, 'domain')

    print(f"DHCP Server {packet[IP].src} ({packet[Ether].src}) "
          f"offered {packet[BOOTP].yiaddr}")

    print(f"DHCP Options: subnet_mask: {subnet_mask}, lease_time: "
          f"{lease_time}, router: {router}, name_server: {name_server}, "
          f"domain: {domain}")
```

Figure 5. 4 Scapy Implementation

This DHCP server information was used in the detection module. The detection module involved comparing the retrieved DHCP server information against the information contained in the configuration file. If the values matched, then it meant they were received from a genuine DHCP server. However, if the values did not match, it was an indication of a rogue DHCP server attack. Figure 5.5 below shows a code snippet of the detection module.

```
elif DHCP in packet and packet[DHCP].options[0][1] == 5:
    print('---')
    print('New DHCP Ack')
    #print(packet.summary())
    #print(ls(packet))

    subnet_mask = get_option(packet[DHCP].options, 'subnet_mask')
    lease_time = get_option(packet[DHCP].options, 'lease_time')
    router = get_option(packet[DHCP].options, 'router')
    name_server = get_option(packet[DHCP].options, 'name_server')
    #print(name_server)
    print(f"DHCP Server {packet[IP].src} ({packet[Ether].src}) "
          f"acked {packet[BOOTP].yiaddr}")

    if packet[IP].src != Legit_Server:
        print("[!] DHCP Attack Detected - Unexpected DHCP Server")
        print(packet[IP].src)
        # to do FW update
    elif (packet[Ether].src) != Legit_Server_MAC:
        print("[!] DHCP Attack Detected - Server has different MAC address")
        # to do FW update
        firewall_update(packet[IP].src)
    else:
        print("[+] Expected DHCP Server in Operation")
```

Figure 5. 5 Detection Module

5.5 System Testing

The following test cases were conducted to ensure that the system works as expected;

Clients and two legitimate DHCP servers: two legitimate DHCP servers were set up and launched in a WLAN environment. One of the legitimate DHCP servers was set up by use of the ISC DHCP and the other was pre-installed in the Access Point (AP). ISC DHCP is a collection of software that implements all aspects of the DHCP suite namely: DHCP server, DHCP client and DHCP relay agent. Both legitimate DHCP servers' IP address pool was configured to accommodate 25 clients each. The IP address of the first genuine DHCP server was 192.168.1.10 while the other was 192.168.1.36. The scope of IP address range of one DHCP server was from 192.168.1.11 to 192.168.1.35 while the other was from 192.168.1.37 to 192.168.1.601. To simulate DHCP clients, twenty dummy clients were introduced by use of a python script based on the scapy module. The DHCP transaction messages between the clients and DHCP servers were observed and analysed using the Wireshark tool. Twenty DHCP Discover messages and 2 DHCP Offers in response to the twenty Discover messages were observed from the Wireshark. This confirmed that the servers were well configured and the python script was working properly.

Clients, Two legitimate DHCP server and rogue DHCP server: A rogue DHCP server was introduced in the test WLAN environment described in the previous scenario. This was done by installing the ISC DHCP server on a Kali Linux virtual machine on an Ubuntu host machine. The ISC DHCP server was assigned IP address 192.168.1.110 which was an unused IP address from the network of the legitimate DHCP servers. This specific IP address was chosen intentionally because at first glance it looks similar to the IP address of the first genuine DHCP server configured in the previous setup. The reason for doing this is to obfuscate the change to suspecting users. This IP address was acquired by doing a non-intrusive network sweep on genuine DHCP servers' network in order to find gaps and unused IP addresses. The rogue DHCP server's pool was configured to accommodate 5 clients. The scope of the IP address range was from 192.168.1.111 to 192.168.1.115. The rogue DHCP server was rebooted in order to effect the configuration changes. After this, Wireshark tool was fired up in order to observe and analyse the changes in the network with the introduction of a rogue DHCP server. This step was repeated five times in order to get the distribution of clients among the DHCP servers and an approximate number of DHCP clients that received Offer messages from the rogue DHCP server. After a series of five repetitions, the results showed more than twice that 18 clients received IP addresses from genuine DHCP servers while 2 clients received IP addresses from the rogue DHCP server. The distribution of 18 clients among the 2 genuine DHCP servers was almost equal. In some iterations it was 9 each while in some it was 10 for one server and 8 in the other. These results showed that when there are more legitimate DHCP servers in the network compared to rogue DHCP servers, the less likely the danger of a rogue DHCP server attack. This is because there were more genuine DHCP servers processing DHCP clients requests thus the clients that received Offer messages from the rogue DHCP server were very few. Also, the rogue DHCP server's identity was returned as a payload in the DHCP Acknowledgement message. This justified the use of DHCP Ack as system parameter in the design of the system.

Clients, One legitimate and one rogue: Another experiment was done to test the distribution of clients in a set up that constituted one rogue DHCP server and one legitimate DHCP server. The test was also repeated five times by running the python script for generating dummy clients at each series test. The results from the test showed an increase in the number of clients issued IP addresses by a rogue DHCP server as compared to the setup that involved two legitimate DHCP servers. The distribution of clients between the legitimate DHCP server and rogue DHCP server was almost

half, that is, 10 clients each. Out of the 5 series of tests done, 2 showed that 11 clients received IP addresses from rogue DHCP server and 9 clients received IP addresses from the genuine DHCP server while 3 tests showed half distribution between the rogue and genuine DHCP server.

Detecting the parameters issued by rogue DHCP servers: This was done by using a configuration file from which an IP and MAC address of the legitimate DHCP server was entered. In addition to this, the IDS had an inbuilt sniffing function that enabled it to sniff all DHCP messages exchanged between the clients and the servers. Through sniffing, the IDS was able to monitor traffic and detect rogue DHCP servers by comparing offer and acknowledgement messages sent by the servers to the values contained in the configuration file. If the parameters didn't match, then it was an indication of a rogue DHCP server attack.

5.6 System Validation

Performance testing was done to ensure reliability, availability and consistent performance of the proposed model. The metric used for measuring the performance of the IDS was the detection rate. As the IDS was rule-based, it was done by querying the rogue DHCP lease file to determine the number of offer messages sent by the rogue DHCP server in comparison to those detected by the IDS. Table 5.1 below shows the offered IP addresses by rogue DHCP server based on the test that involved two legitimate DHCP servers and one rogue DHCP server..

Table 5. 1 Offered IP Addresses by Rogue DHCP Server

Offered IP Address	**Client Hostname**	**IP Address of (Offering) Rogue DHCP server**
192.168.1.112	Host A	192.168.1.110
192.168.1.114	Host B	192.168.1.110

The offered IP addresses were checked against the report generated by the IDS to ensure they were flagged. The report from the IDS confirmed that all IP addresses offered by the rogue DHCP server were detected by the IDS. This validated that the rule-based IDS had a 100% detection rate. Figure 5.6 and 5.7 below shows the IDS performance graph comparison between two legitimate DHCP servers.

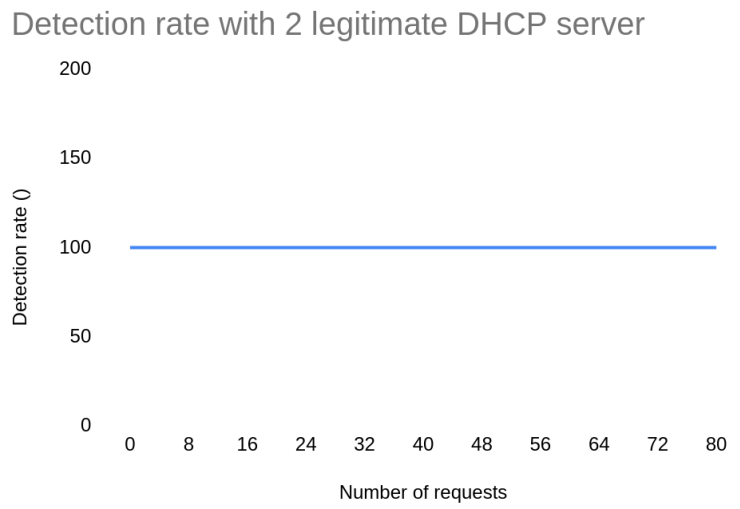


Figure 5. 6 Detection Rate with 2 Legitimate DHCP Servers

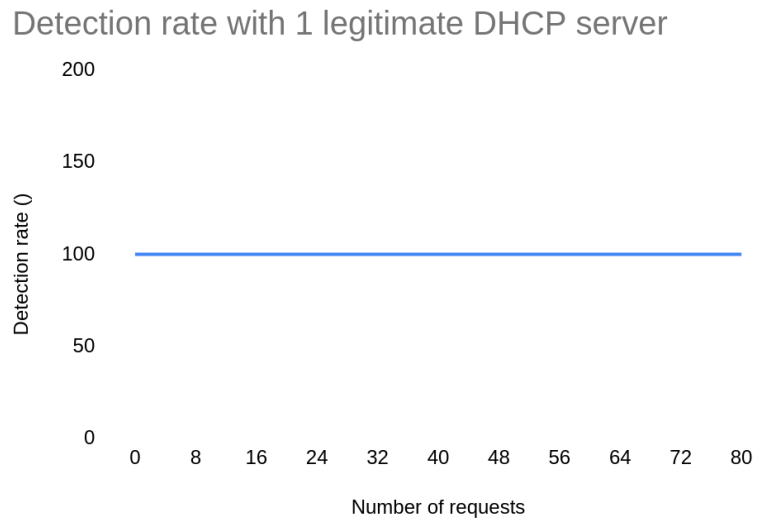


Figure 5. 7 Detection Rate with 1 Legitimate DHCP Server

Chapter 6: Conclusions and Future Work

6.1 Introduction

This chapter analyses the findings obtained from the previous chapter. It discusses the results obtained and correlates the findings against the research objectives. Moreover, the chapter looks at the conclusions, recommendations and future work.

6.2 Discussion

The first objective of this research was to investigate modes of launching rogue DHCP server attacks on WLAN communication. The research revealed that DHCP is vulnerable to several attacks and one of the attacks that majorly affects WLAN security is rogue DHCP server. Rogue DHCP server attacks can be launched through DHCP starvation which is a Denial of Service (DoS) attack and DHCP spoofing attack which is mostly achieved after launching a DHCP starvation attack. However, a rogue DHCP server can also be set up in the network directly without launching prior attacks.

The second research objective was to analyse the existing solutions used to prevent rogue DHCP attacks on WLAN communication. The literature revealed a lot of solutions that encompassed models that borrow from cryptography. These models were effective in detecting rogue DHCP server attacks; however, they introduced complexity in the performance of DHCP protocol structure and were also cumbersome to implement.

The third objective was to design, develop and test a model that detects and prevents rogue DHCP attacks. This objective was attained through designing, implementing and testing of the integrated system. It was developed mainly by integrating security system components such as NIDS. The tests conducted included the functionality and performance test in detecting rogue DHCP server traffic in WLAN environments.

The fourth research objective was to validate the performance of the developed model. Validation was attained by conducting several test cases of the model to evaluate the performance and ascertain that it was detecting rogue DHCP servers in the WLAN environment.

The dissertation brings into perspective the importance of systems security integration. The test

results collected in the above chapter, included a rogue DHCP server attack which was launched by the attacker using the guest Kali machine. The system was able to detect and prevent the rogue DHCP server attack and respond timely. This was made possible by use of the IDS. On detection, alerts were displayed on the terminal. This activated the IP tables that blocked the IP address of the rogue DHCP server.

6.3 Conclusion

The system was developed to monitor network activities and detect rogue DHCP server attacks in the WLAN environment. The experimental setup that involved integration of system components discussed in chapter four aided in coming up with a system that would achieve the objectives set in chapter one.

The literature review presented the different models that have been used to detect rogue DHCP server attacks. The models discussed centred their approach on cryptography, machine learning, just to mention a few. However, the models discussed increased complexity of DHCP protocol by introducing new parameters and increasing the number of messages exchanged. This aided in settling on the approach discussed in chapter three and coming up with a system that detects rogue DHCP server attacks without interfering with the DHCP protocol parameters.

6.4 Recommendations

This research recommends the use of a Linux environment as the base for which the system is built upon. This is because Linux is an open source platform that is secure. Permission and ownership of files can easily be assigned and reassigned.

Finally, the research recommends use of an Access Point (AP) that has port mirroring capabilities which provides better visibility of the DHCP transaction messages exchanged between clients and DHCP servers. This can be implemented on a

6.5 Future Research Work

This dissertation focused on detecting rogue DHCP server attacks. In the future, the researcher recommends a system that can counter both rogue gateway and rogue Access Point attacks in addition to the rogue DHCP server attack. This is because these network devices are vital during the IP configuration process.

Further, the researcher recommends researching on modes of detecting rogue DHCP server attacks on cloud instances especially with the current trend of most organizations migrating their businesses to the cloud.

References

A. K. Rajput, R. Tewani, & A. Dubey. (2016). The helping protocol "DHCP." *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 634–637.

Agarwal, M., Biswas, S., & Nandi, S. (2019). Discrete event system framework for fault diagnosis with measurement inconsistency: Case study of rogue DHCP attack. *IEEE/CAA Journal of Automatica Sinica*, *6*(3), 789–806. https://doi.org/10.1109/JAS.2017.7510379

Aura, T., Roe, M., & Murdoch, S. J. (2007). Securing network location awareness with authenticated DHCP. *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, 391–402. https://doi.org/10.1109/SECCOM.2007.4550359

Biondi, P., & Scapy community. (2021, September 3). *Welcome to Scapy's documentation! — Scapy 2.4.5. Documentation*. https://scapy.readthedocs.io/en/latest/

Burris, J. W. (2017). Test-Driven Development for Parallel Applications. *2017 Second International Conference on Information Systems Engineering (ICISE)*, 27–31. https://doi.org/10.1109/ICISE.2017.20

Caballero, A. (2014). Chapter 1 - Information Security Essentials for IT Managers: Protecting Mission-Critical Systems. In J. R. Vacca (Ed.), *Managing Information Security (Second Edition)* (pp. 1–45). Syngress. https://doi.org/10.1016/B978-0-12-416688-2.00001-5

Deliang, C., Xing, L., & Qianli, Z. (2016). A comparative study on user characteristics of fixed and wireless network based on DHCP. *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, 327–330. https://doi.org/10.1109/ITNEC.2016.7560375

Dinu, D. D., & Togan, M. (2014). DHCP server authentication using digital certificates. *2014 10th International Conference on Communications (COMM)*, 1–6. https://doi.org/10.1109/ICComm.2014.6866756

Dinu, D. D., & Togan, M. (2015). DHCPAuth—A DHCP message authentication module. *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*, 405–410. https://doi.org/10.1109/SACI.2015.7208238

Droms, R. (2021). *Dynamic Host Configuration Protocol* (Request for Comments RFC 2131). Internet Engineering Task Force. https://doi.org/10.17487/RFC2131

Duangphasuk, S., Kungpisdan, S., & Hankla, S. (2011). Design and implementation of improved security protocols for DHCP using digital certificates. *2011 17th IEEE International Conference on Networks*, 287–292. https://doi.org/10.1109/ICON.2011.6168490

Graaf, K. de, Liddy, J., Raison, P., Scano, J. C., & Wadhwa, S. (2011). *Dynamic host configuration protocol (dhcp) authentication using challenge handshake authentication protocol (chap) challenge* (United States Patent No. US20110154440A1). https://patents.google.com/patent/US20110154440/en

Komori, T., & Saito, T. (2002). The secure DHCP system with user authentication. *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.*, 123–131. https://doi.org/10.1109/LCN.2002.1181774

Kyambadde, G. H., & Ngubiri, J. (2018). A Tool to Mitigate Denial of Service Attacks on Wired Networks. *Journal of Computing and Information Technology*, *5*, 102–107.

Ornaghi, A., & Valleri, M. (2003). *Man in the middle attacks*. 61.

Pradana, D. A., & Budiman, A. S. (2021). The DHCP Snooping and DHCP Alert Method in Securing DHCP Server from DHCP Rogue Attack. *IJID (International Journal on Informatics for Development)*, *10*(1), 38–46. https://doi.org/10.14421/ijid.2021.2287

ritcsec. (2019, April 28). Port Security Attack & Mitigation. *RIT Computing Security Blog*. https://ritcsec.wordpress.com/2019/04/27/port-security-attack-mitigation/

Shete, A., Lahade, A., Patil, T., & Pawar, R. (2018). DHCP Protocol Using OTP Based Two-Factor Authentication. *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 136–141. https://doi.org/10.1109/ICOEI.2018.8553753

Shue, C. A., Kalafut, A. J., & Gupta, M. (2009). A Unified Approach to Intra-domain Security. *2009 International Conference on Computational Science and Engineering*, *3*, 219–224. https://doi.org/10.1109/CSE.2009.204

Tanceska, B., Bogdanoski, M., & Risteski, A. (2015). Simulation Analysis of DoS, MITM and CDP Security Attacks and Countermeasures. In V. Atanasovski & A. Leon-Garcia (Eds.), *Future Access Enablers for Ubiquitous and Intelligent Infrastructures* (pp. 197–203). Springer International Publishing. https://doi.org/10.1007/978-3-319-27072-2_25

Tomar, D. (2020). A Network Architecture for secure traffic management for the Internet of

Things using Virtual Local Area Network. *International Journal of Computer Trends and Technology*, *68*, 11–14. https://doi.org/10.14445/22312803/IJCTT-V68I12P103

Tripathi, N., & Hubballi, N. (2017). Detecting Stealth DHCP Starvation Attack using Machine Learning Approach. *Journal of Computer Virology and Hacking Techniques*, *14*. https://doi.org/10.1007/s11416-017-0310-x

Waldo, J. (2006). *On System Design*. Sun Labs. https://scholar.harvard.edu/files/waldo/files/ps-2006-6.pdf

Wang, J., & Chen, Y. (2017). An SDN-based defensive solution against DHCP attacks in the virtualization environment. *2017 IEEE Conference on Dependable and Secure Computing*, 529–530. https://doi.org/10.1109/DESEC.2017.8073876

Yaibuates, M., & Chaisricharoen, R. (2019). STARVATION DELAYED DHCP SERVICE FOR ENABLING POOL RECOVERY. *Malaysian Journal of Computer Science*, 15–34. https://doi.org/10.22452/mjcs.sp2019no2.2

Yaibuates, M., & Chaisricharoen, R. (2020). A Combination of ICMP and ARP for DHCP Malicious Attack Identification. *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*, 15–19. https://doi.org/10.1109/ECTIDAMTNCON48261.2020.9090760

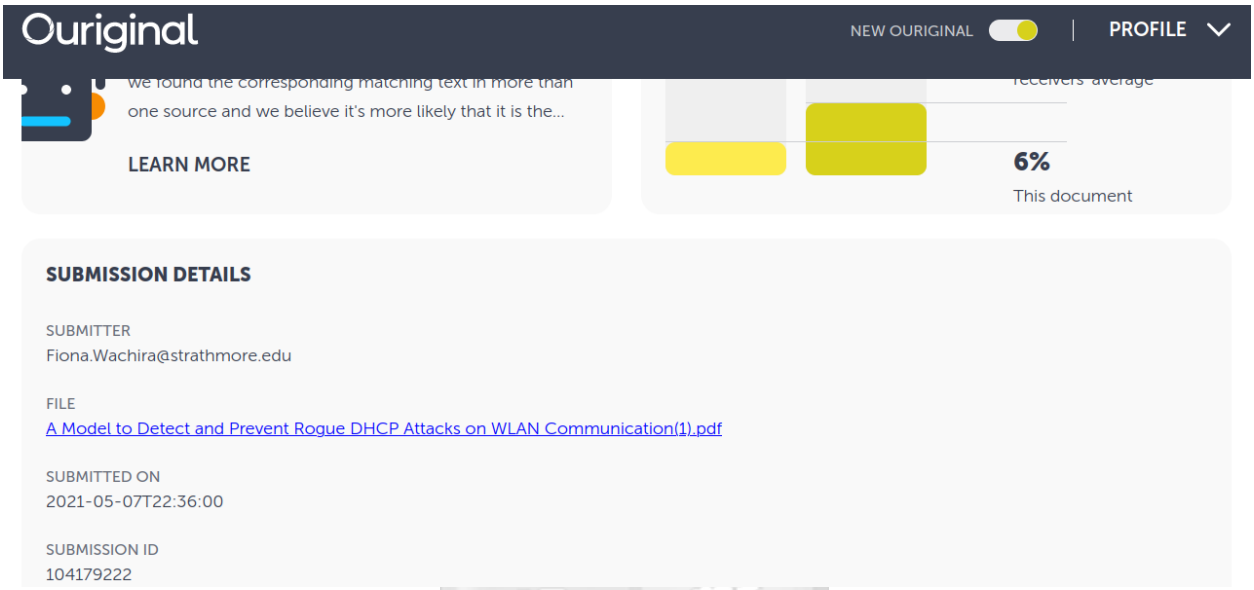
Yaibuates, M., Upra, R., & Chaisricharoen, R. (2016). *ICMP Based IP address Recovery Method for DHCP*. 267–271.

Younes, O. S. (2017). Securing ARP and DHCP for mitigating link layer attacks. *Sādhanā*, *42*(12), 2041–2053. https://doi.org/10.1007/s12046-017-0749-y

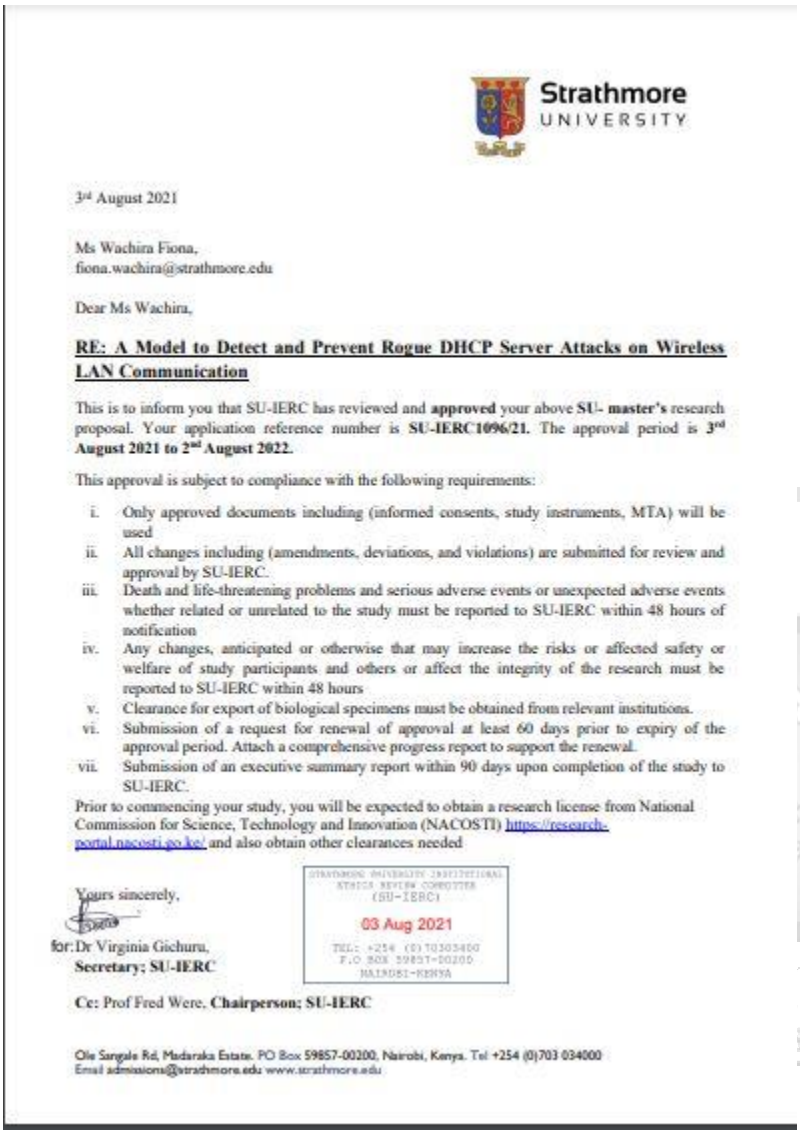
Yun Yang & Jia Mi. (2010). Design of DHCP protocol based on access control and SAKA encryption algorithm. *2010 2nd International Conference on Computer Engineering and Technology*, *6*, V6-264-V6-267. https://doi.org/10.1109/ICCET.2010.5486268

Appendices

Appendice A: Plagiarism Report



Appendice B: Ethical Approval



Strathmore UNIVERSITY

3rd August 2021

Ms Wachira Fiona,
fiona.wachira@strathmore.edu

Dear Ms Wachira,

RE: A Model to Detect and Prevent Rogue DHCP Server Attacks on Wireless LAN Communication

This is to inform you that SU-IERC has reviewed and **approved** your above **SU- master's** research proposal. Your application reference number is **SU-IERC1096/21.** The approval period is **3rd August 2021 to 2nd August 2022.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and also obtain other clearances needed

Yours sincerely,

for: Dr Virginia Gichuru,
Secretary; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
03 Aug 2021
TEL: +254 (0)703034000
P.O BOX 59857-00200
NAIROBI-KENYA

Cc: Prof Fred Were, **Chairperson; SU-IERC**

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu