**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 15, No. 6, December 2025, pp. 6027~6042
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp6027-6042 

# Implementation of a network intrusion detection system for man-in-the-middle attacks

**Kennedy Okokpujie[1,2], William A. Abdulateef-Adoga[1], Oghenetega C. Owivri[1], Adaora P. Ijeh[3], Imhade P. Okokpujie[4,5], Morayo E. Awomoy[6]**
[1]Department of Electrical and Information Engineering, College of Engineering, Covenant University, Ota, Nigeria
[2]Africa Centre of Excellence for Innovative and Transformative STEM Education, Lagos State University, Lagos State, Nigeria
[3]Department of Computer and Information Science, College of Science and Technology, Covenant University, Ota, Nigeria
[4]Department of Mechanical and Mechatronics Engineering, Afe Babalola University, Ado Ekiti, Nigeria
[5]Department of Mechanical and Industrial Engineering Technology, University of Johannesburg, Johannesburg, South Africa
[6]School of International Service, American University, Washington, D.C., United States of America

**Article Info**

***Article history:***

Received Aug 31, 2024
Revised May 5, 2025
Accepted May 24, 2025

***Keywords:***

Address resolution protocol poisoning
Address resolution protocol spoofing
Intrusion detection systems
Man-in-the-middle attacks
Wireless network

**ABSTRACT**

Intrusion detection systems (IDS) are critical tools designed to detect and prevent unauthorized access and potential network threats. While IDS is well-established in traditional wired networks, deploying them in wireless environments presents distinct challenges, including limited computational resources and complex infrastructure configurations. Packet sniffing and man-in-the-middle (MitM) attacks also pose significant threats, potentially compromising sensitive data and disrupting communication. Traditional security measures like firewalls may not be sufficient to detect these sophisticated attacks. This paper implements a network intrusion detection system that monitors a computer network to detect Address Resolution Protocol spoofing attacks in real-time. The system comprises three host machines forming the network. Using Kali Linux, a bash script is deployed to monitor the network for signs of address resolution protocol (ARP) poisoning. An email alert system is integrated into the bash script, running in the background as a service for the network administrator. Various ARP spoofing attack scenarios are performed on the network to evaluate the efficiency of the network IDS. Results indicate that deploying IDS as a background service ensures continuous protection against ARP spoofing and poisoning. This is crucial in dynamic network environments where threats may arise unexpectedly.



***Corresponding Author:***

Kennedy Okokpujie
Department of Electrical and Information Engineering, College of Engineering, Covenant University
KM 10 Idiroko Road, Ota, Ogun State, Nigeria
Email: kennedy.okokpujie@covenantuniversity.edu.ng

## 1. INTRODUCTION

Wireless networks are integral to modern communication, providing extensive connectivity and convenience. However, their widespread adoption has led to increased security vulnerabilities, with cyber-attacks becoming more frequent and sophisticated, making detection more challenging. In response, organizations focus on reducing, detecting, and preventing these risks by implementing systems tailored to their specific network types, whether wired or wireless [1]. These systems serve as primary or secondary defense mechanisms against potential threats. Intrusion detection systems (IDS) are critical tools, designed to detect and prevent unauthorized access and potential threats within networks. While IDS is well-established in traditional wired networks, deploying them in wireless environments presents distinct challenges,

including limited computational resources and complex infrastructure configurations. These challenges necessitate innovative approaches to ensure adequate security measures.

The primary function of IDS is to continuously monitor network activities, identifying unusual patterns that may indicate security breaches. IDS are especially vital in environments where fully secure information systems are complex due to operational constraints or legacy infrastructure. By detecting and responding to potential threats, IDS plays a crucial role in maintaining the integrity of information systems and protecting sensitive data from unauthorized access. IDS can be categorized based on several factors, including detection methods, deployment strategies, and intended use [2]. For instance, signature-based IDS rely on known threat patterns, while anomaly-based systems detect deviations from normal behavior. Additionally, IDS can be deployed at various points in a network, such as host-based IDS, which monitors individual devices, and network-based IDS, which oversees broader network traffic. Each type of IDS offers tailored solutions to different security challenges within wireless networks, thereby ensuring data security. As wireless networks continue to evolve, the role of IDS will become increasingly important in safeguarding against emerging threats. Home networks are particularly vulnerable to cyberattacks due to the growing number of connected devices and the evolving tactics of attackers [3]. Packet sniffing and man-in-the-middle (MitM) attacks pose significant threats, potentially compromising sensitive data and disrupting communication. Traditional security measures like firewalls may not be sufficient to detect these sophisticated attacks [4].

This work implements a network intrusion detection system that monitors a computer network to detect real-time man-in-the-middle attacks. The system comprises three host machines forming the network. Using Kali Linux, a bash script is deployed to monitor the network for signs of address resolution protocol (ARP) poisoning. An email alert system is integrated into the bash script, running in the background as a service for the network administrator. Various ARP spoofing attack scenarios are performed on the network to evaluate the efficiency of the network IDS.

## 2. LITERATURE REVIEW

This section explores various technologies and strategies employed in developing a network IDS specifically tailored to detect and mitigate MITM attacks via ARP poisoning. The persistence of cyberattacks is not unusual [5]. Therefore, there are more attacks on organizations that can recognize risks and breaches. This suggests that organizations that are not as adept at spotting attacks as others may be unaware of an attack or being part of a botnet [6]. Additionally, an attack campaign using advanced persistent threat (APT) might continue stealing data without being noticed for months or years. This suggests that the attack frequency may be comparable to that of organizations that can identify high attack frequency [6].

Two significant scaling problems have arisen on the Internet in recent years as it has battled to maintain steady and unbroken growth [7]: i) The capacity to direct traffic throughout the ever-growing array of networks that make up the internet and ii) the ultimate depletion of IPv4 address space.

### 2.1. OSI model

The OSI model stands for open system interconnection. The OSI model is a reference for how applications communicate over a network. It describes the flow of packets and how they are transmitted. The work presented in this paper focuses on layer 3 (network layer) of the OSI model. The network layer is responsible for routing and forwarding packets between devices on different networks. It is responsible for providing logical addressing, routing, and congestion control. Figure 1 illustrates the OSI model and how packets are moved and transferred from the physical layer to the application layer.

### 2.2. Internet protocol address

Internet protocol address is abbreviated as IP Address. Every computing device connected to an IP network is given a unique identity known as an Internet protocol address. Since IP version 4 (IPv4), the most recent version of IP defines a 32-bit address; there are only $2^{32}$ (4,294,967,296) IPv4 addresses accessible. The limited supply of IP addresses will soon run out as new markets develop and a sizable fraction of the global population qualifies for IP addresses. The fact that some IP address space has yet to be allotted effectively exacerbates the address scarcity issue [8]. Every device on an internet service provider's (ISP) network is given an IP address to identify and communicate with other devices on the same IP network [8]. Two types of IP Addresses are currently and widely in use, clearly illustrated in Figure 2 [8].

An IP address is called static if it does not change after being assigned to a network element. In contrast, a dynamic IP address is momentarily issued to a computer or device on a network [9]. A dynamic IP address can be assigned to another networked device whenever it is not in use. Network components are assigned dynamic IP addresses via the point-to-point protocol over ethernet (PPPoE) or the dynamic host

configuration protocol (DHCP). This kind of IP address is used by the majority of ISPs on their networks. Static IP addresses are more expensive than dynamic ones [10].

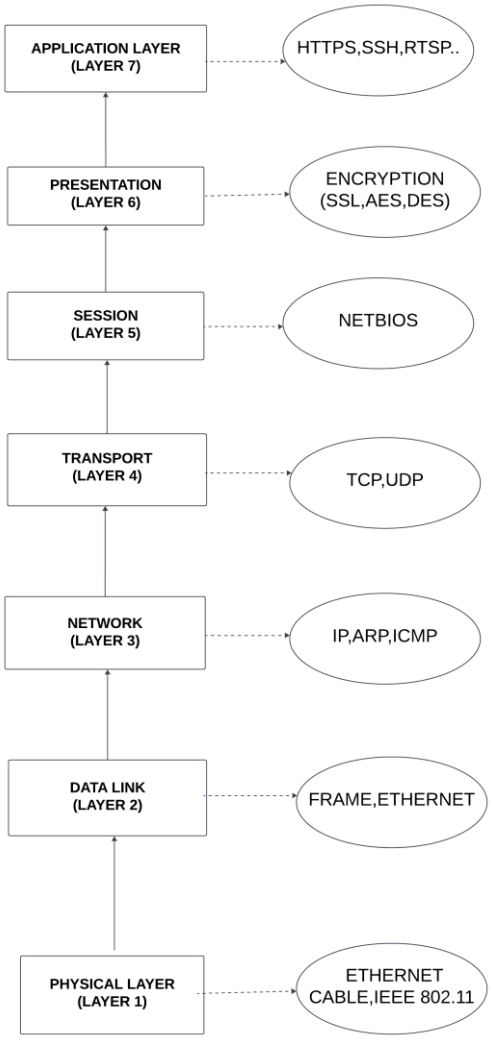



Figure 1. OSI model


Figure 2. Types of IP address

### 2.2.1. IP address format

An IP address in binary form, such as "00000001.10100000.00001010.11110000," is used by computer systems for communication and is then converted to a decimal format (e.g., 1.160.10.240) for user convenience [11]. An IP address is a 32-bit binary number represented as four sets of digits (usually referred to as octets) separated by dots. The IP address is divided into two sections: the Net ID, which identifies the

host's network and typically consists of the first three octets, and the Host ID, which identifies the specific host within that network and is represented by the fourth octet.

### 2.2.2. IPv4 address

An IPv4 address is used to identify all hosts within the same network. Figure 3 illustrates the three classes of IPv4 addresses, including their subnet masks and binary formats. An IPv4 address contains 32 bits, divided into four 8-bit segments separated by periods. IPv4 addresses are crucial in executing a man-in-the-middle attack, as the attacker must identify the target's IP address. Tools such as Netdiscover are often used during the information-gathering phase. Netdiscover is an active/passive ARP reconnaissance tool initially developed for wireless networks without DHCP servers. It can also be employed on switched networks, passively detecting online hosts or actively searching for them by sending ARP requests. There are two types of IPv4 addresses: Private IP Addresses are generated by a DHCP server and assigned within private networks, such as homes. The class of the IPv4 address determines the specific range, with most private networks using Class C, which supports up to 254 hosts with a subnet mask of 255.255.255.0. Figure 3 presents the three classes of private IPv4 addresses. An example of a Class C IPv4 address is 192.168.100.0. Public IP Addresses are used to identify an internet service provider (ISP), domain name, or organization. It serves as a unique identifier, enabling devices to communicate with each other and access online resources [12].

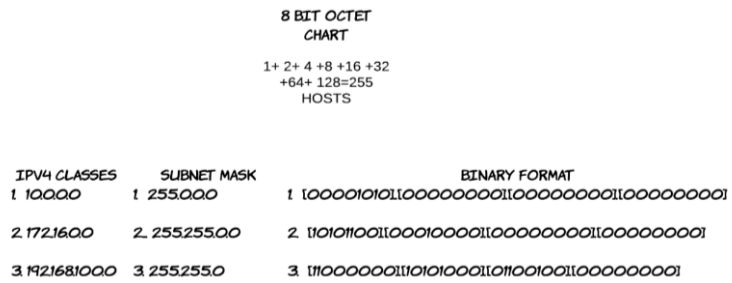



Figure 3. IPV4 address format

### 2.2.3. Subnet mask

A subnet mask is used to determine the network and host portions of an IPv4 address by masking the network address. The network address contains the information necessary for routing packets. In most private homes, a class C IPv4 address is used, which supports up to 254 hosts. Identification of each host is straightforward, as the first three octets remain constant while the fourth octet varies. In a Class C address such as 192.168.100.1 with a subnet mask of 255.255.255.0, the network ID is 192.168.100, while the last octet represents the Host ID. All devices on that network share the same network ID but have unique Host IDs [13].

## 2.3. Data transmission

Transferring data between networked devices requires adherence to specific protocols, which vary depending on whether the devices are directly or remotely connected. A critical factor in determining the data delivery method is whether the sender and receiver are within the same broadcast domain, often called virtual local area network (VLAN). Communication within the same broadcast domain uses switches and medium access control (MAC) addresses, while communication across domains requires a router and IP addresses.

### 2.3.1 Default gateway

The default gateway is the connection point between devices in different broadcast domains. It functions as a “doorway” that allows data to pass from one broadcast domain to another. Layer 3 devices like routers or multilayer switches typically act as the default gateway, facilitating communication beyond the local network [14].

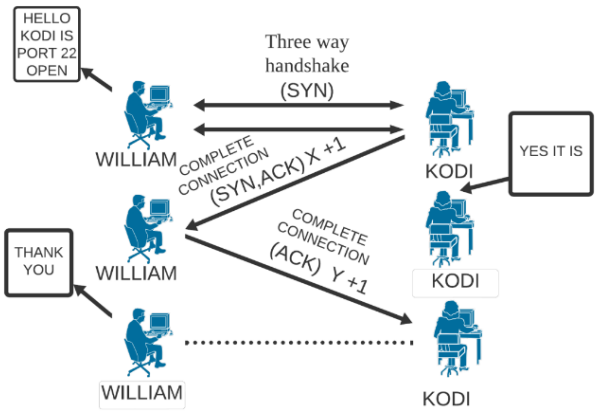
### 2.3.2 Transmission control protocol

The transmission control protocol (TCP) is a communication protocol that operates within the internet protocol (IP) model, which is structured into four layers: network access (link layer), network, transport, and application. These layers correspond to layers 1-2, 3, 4, and 5-7 of the OSI model represented

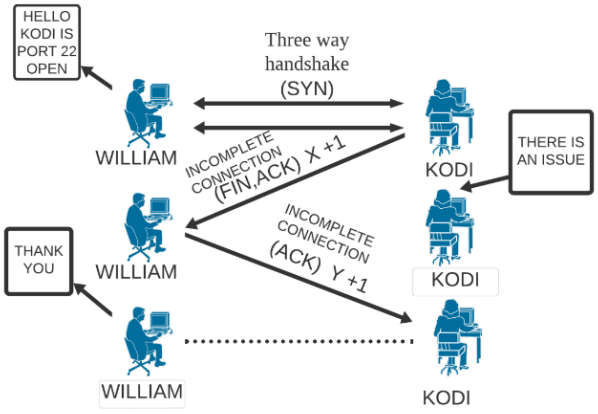
in Figure 1, respectively. While TCP is commonly used in network communications, it is often referred to as a layer 4 protocol in the OSI model due to its operation at the transport layer [14]. The TCP is a standard that establishes a strong connection between the client and a server and allows application programs to exchange data. TCP is very reliable and stable. For a TCP connection to go through, it has to undergo the process of a three-way handshake. The three-way handshake is a procedure used to establish and maintain a connection between a client and a host, involving specific TCP flags: SYN, ACK, FIN, RST, and PSH, illustrated in Figure 4.

As illustrated in Figures 4(a)-(b), the process of establishing a TCP connection occurs in three steps:

- Step 1: The client (William) initiates the connection by sending a SYN packet to the server (Kodi) with a random sequence number.
- Step 2: The server (Kodi) responds with a SYN-ACK packet, acknowledging the connection and synchronizing with the client using a sequence number of X+1.
- Step 3: The client (William) completes the connection by sending an ACK flag, acknowledging the server's SYN flag with a sequence number of Y+1.




(a)




(b)

Figure 4. TCP-based connection (a) client and server connecting and (b) client and server disconnecting

### 2.3.3. Switch and MAC address

A switch is a network device that facilitates communication between hosts via their MAC addresses, ensuring that packet information is not leaked to other hosts. It is an advanced version of a hub and contains several physical ports that store different host MAC addresses. The MAC address is a unique, physical, and permanent identifier used for communication between devices. Switches are critical in ARP spoofing, as they forward packets based on MAC addresses.

### 2.3.4. Address resolution protocol

The address resolution protocol (ARP) is a fundamental protocol in the internet protocol suite (TCP/IP) used to resolve an IP address into a MAC address. The address resolution protocol occurs in the network layer. It allows us to communicate with other hosts under the same network and the default gateway. The ARP maps IP addresses to MAC addresses, enabling data transmission between devices. Figure 5 demonstrates the two functions of the ARP. When host A needs to send data to Host B, it knows its IP address (172.16.3.2) but requires its MAC address. Host A sends an ARP request asking for the MAC address associated with the IP address 172.16.3.2. Host B responds with its MAC address (0800.0200.1111). Once host a receives this information, the data can be sent. ARP also maintains a cache of IP-to-MAC associations, eliminating the need for repeated ARP requests when sending additional data to the same host, thereby speeding up the data delivery process [15].

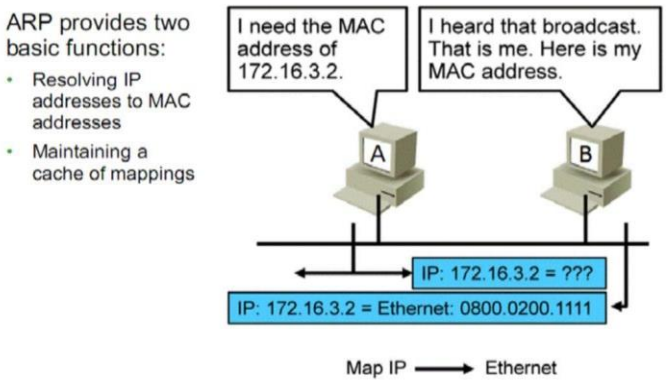



Figure 5. Address resolution protocol (ARP) request [15]

### 2.3.5. ARP spoofing

In ARP spoofing, an attacker sends fake ARP responses to manipulate the MAC address table on a switch. When a fake ARP response with a fraudulent MAC address is received, the switch updates its MAC address table with the incorrect information. As a result, the switch forwards packets to the attacker's machine, mistakenly identifying it as the original device. Figure 6 illustrates how an attacker poisons the ARP cache. By altering the MAC address table, the attacker can intercept traffic intended for the original device, enabling eavesdropping, malware injection, or the theft of sensitive information.

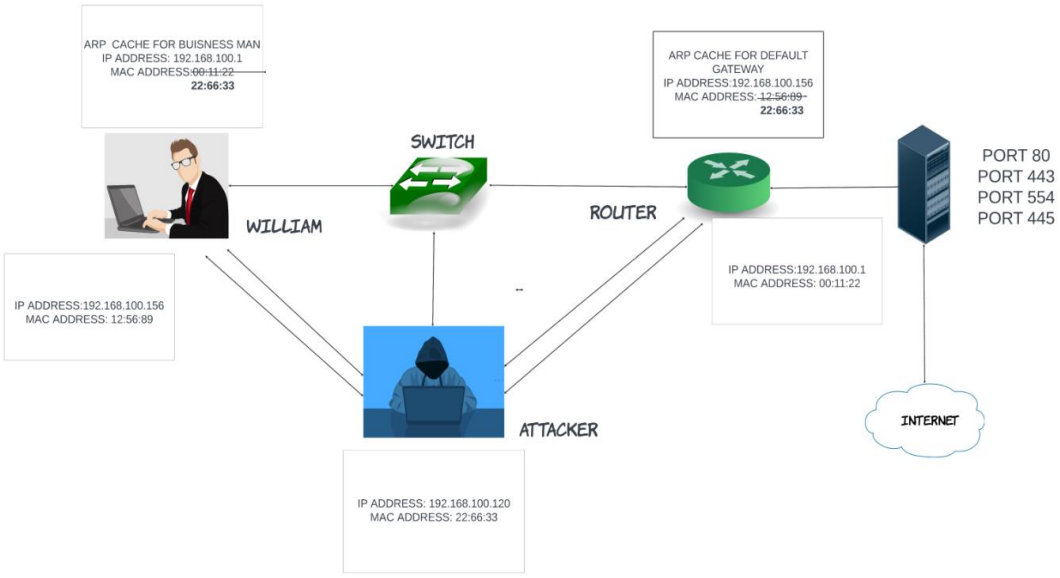



Figure 6. ARP spoofing

### 2.3.6. Internet control message protocol

Over the internet, network problems can be identified using the internet control message protocol (ICMP). Network devices employ the internet control message protocol, a network layer protocol, to identify problems with network connectivity. The primary purpose of ICMP is to ascertain whether or not data arrives at its destination on time. The ICMP protocol is frequently utilized on network equipment, including routers. Besides being useful for testing and error reporting, ICMP could also be employed in distributed denial-of-service (DDoS) assaults [16].

### 2.3.7. L3 header

The L3 headers are used to route data packets between networks. Frames encapsulate data for transmission over a physical medium, such as a network cable. Frames contain a data payload, which includes the information for that packet. L3 headers contain control information, such as routing information, checksums, and source and destination addresses. L3 headers have a standardized format that is defined by the protocol. Figure 7 shows the L3 header format.

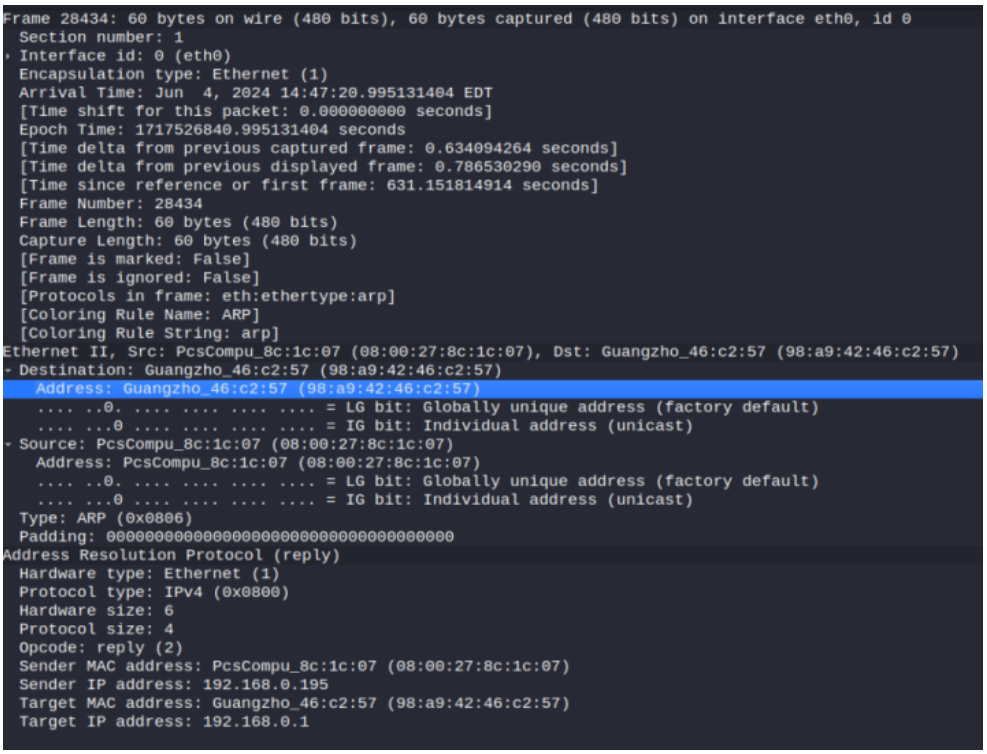



Figure 7. L3 header format

### 2.3.8. Packet transmission

For a packet to reach its destination, the system needs the source IP, destination IP, source MAC address, and destination MAC address [14]. An ARP request is issued if the destination MAC is unknown. The packet is considered complete and ready for forwarding once the ARP process resolves the MAC address. Fundamental principles of packet delivery include:

- MAC addresses do not leave their broadcast domain.
- ARP resolution is prioritized if necessary.
- Packets are delivered directly to the destination if both devices are in the same network.
- If the destination is in a different network, the packet is first sent to the default gateway before reaching the destination

Figure 8 demonstrates packet transmission within a network topology with four hosts. When Host 1 (IP: 10.10.10.1/24) pings Host 4 (IP: 10.10.10.4/24), an internet control message protocol (ICMP) echo request is generated.

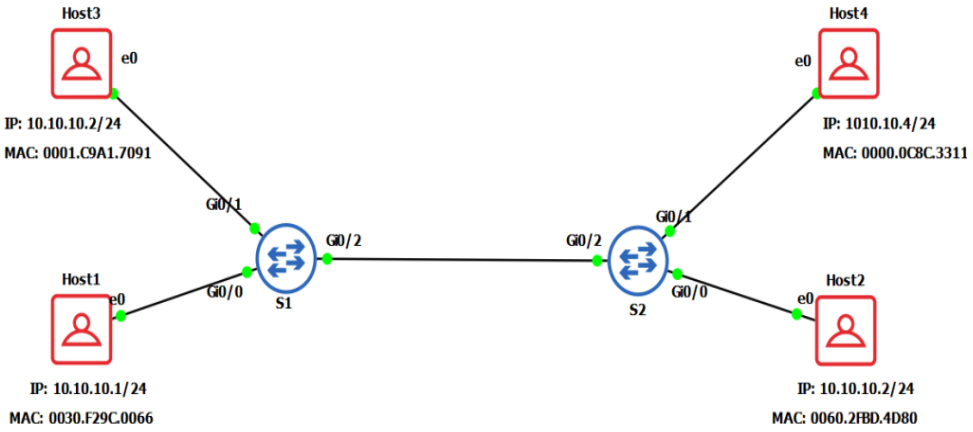



Figure 8. Packet transmission within a network topology with four hosts

The network performs AND operations with the subnet masks if both IP addresses are in the same subnet. The next-hop IP address is determined since both hosts are within the same network (10.10.10.0/24). If the ARP table does not contain the next-hop IP address, the packet is buffered while an ARP request is broadcasted. The switch (S1) checks for the source MAC address in its content addressable memory (CAM) table, adds an entry if none exists, and broadcasts the ARP frame to all ports in the same broadcast domain. Hosts that do not match the destination IP address discard the frame, while the correct host (Host 4) responds with its MAC address. The response is unicast back to Host 1, which updates its ARP table, extracts the buffered packet, and completes the transmission. Host 4 then replies to the internet control message protocol (ICMP) echo request, resulting in a successful ping.

### 2.4. Man-in-the-middle (MitM) attack

A man-in-the-middle (MitM) attack occurs when an attacker inserts themselves between a sender and a receiver, intercepting and altering traffic flow between the two communicating parties [17]. This allows the attacker to eavesdrop, modify, and retransmit data. Several methods can be used to execute a MitM attack, including DNS spoofing, ARP spoofing, and DHCP spoofing, with ARP spoofing being the primary focus of this paper.

MitM attacks pose significant risks, including identity theft, privacy breaches, and data loss. These attacks can be classified into two categories: those targeting the OSI model's upper layers (3-6) and those targeting the lower layers (1-2). Wi-Fi devices typically use omnidirectional antennas that broadcast data within a specific range, making them susceptible to packet sniffing by attackers who place their wireless adapters in “monitor mode.” encryption at various OSI model layers, such as HTTPS/SSL, VPNs, or Wi-Fi encryption protocols (WEP, WPA1, WPA2, WPA3), can mitigate packet sniffing. However, public Wi-Fi networks, often lacking layer two encryption, remain particularly vulnerable. MitM attacks exploit intermediaries (*e.g.*, routers) necessary for network message transmission. Attackers can hijack these intermediaries to intercept or alter data. MitM attacks can target multiple OSI layers: the Application layer (*e.g.*, DNS spoofing, SSL hijacking), the network layer (*e.g.*, ARP poisoning), and the data link layer (*e.g.*, antenna identity spoofing). Attackers typically launch MitM attacks on Wi-Fi networks by sniffing the Wi-Fi spectrum, creating a fake access point, and luring clients to connect to it, enabling the interception of data.

### 2.5. Review of related works

Aims to identify man-in-the-middle (MitM) attacks on wireless networks [17]. The research focused on the physical layer of the OSI model, examining electromagnetic activity to detect signal patterns indicative of such attacks. The study first evaluated the protection levels associated with Wi-Fi standards, followed by a detailed description of common MitM attack procedures on corporate, public, and private Wi-Fi networks. Based on their findings, the authors developed a MitM attack detection approach by identifying the signal characteristics emitted by fake access points. They successfully characterized MitM attacks on Wi-Fi networks and suggested that additional steps may be necessary for attackers to access a target network, depending on its security level. The study concluded that understanding these attack methods would aid in developing an intrusion detection system (IDS) focusing on the OSI model's physical layer [17].

A technical paper by Gangan [18] discussed the basics of MitM attacks in communication networks and their defense strategies. The paper highlighted that timing information can be used to identify attacks in

various real-time communication scenarios. The study also pointed out that DNS spoofing, session hijacking, SSL hijacking, and ARP cache poisoning are common causes of MitM attacks. MitM attacks apply to quantum cryptographic systems and peer-to-peer (P2P) networks. The paper emphasized that MitM attacks are a simple method of compromising quantum cryptography systems, especially if security measures are insufficient to verify the authentication of the two communicating parties [19], [20].

The study suggested that ARP poisoning can be mitigated using a shell script that monitors IP and MAC addresses in the ARP cache table. However, the author noted that this method is limited to Linux systems, and the frequent ARP queries generate substantial network traffic. The paper concluded with recommendations for network security measures, including security-focused network design, regular operating system upgrades, frequent updates of network devices, and routine installation of security patches [18], [21]–[24].

In the journal publication "Handling of man-in-the-middle attack in WSN through intrusion detection system," Mohapatra *et al.* [25] proposed a deep learning-based MitM-intrusion detection system (MITM-IDS) for attack detection, isolation, and node reconfiguration. The method trained nodes to recognize potential threats and demonstrated an 89.147% attack detection rate in simulations. The study's goal was to develop an attack-tolerant IDS. By evaluating throughput and packet loss, the authors validated the effectiveness of their proposed model. Packet loss was defined as the failure of a packet to reach its destination, while throughput was the successful transmission of packets within a set time. The findings revealed low packet loss and high throughput, supporting the authors' claim of having developed a robust network infrastructure.

## 3. METHOD

This section discusses the conceptual framework of the system developed in this work and the methods and processes for implementing the NIDS developed in this paper. The system implements a network intrusion detection system using Bash scripting to detect when an attack has occurred. Python scripting sends email notifications when a MitM attack is detected. The entire system is run as a background service for the network admin. The block diagram illustrated in Figure 9 presents an abstraction of the system. The conceptual framework for this work involves the interaction of three hosts within a network, identified as Host A (network administrator), Host B (Victim), and Host C (Attacker).

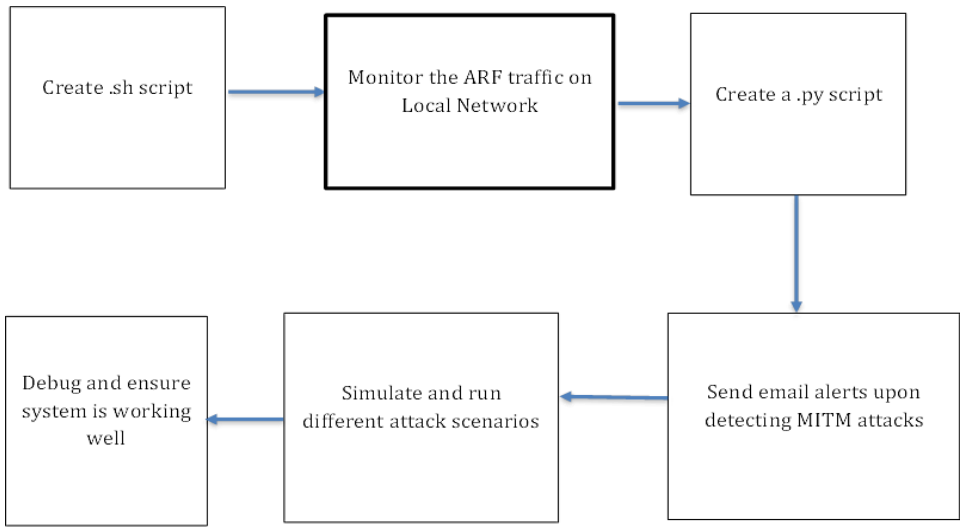



Figure 9. Block diagram of the system

The framework demonstrates how the ARP operates in the network and how the attacker can conduct an ARP spoofing attack to intercept communications between the victim and the router. Host B initiates the communication by sending an ARP request to determine the MAC address associated with the router's IP address (192.168.0.1). This request is broadcasted throughout the network, encapsulated in a frame that includes details such as Host B's IP address (192.168.0.3) and MAC address (06:09:08). The router receives the ARP request, verifies the matching IP address, and responds with an ARP reply. This reply provides the router's MAC address (00:15:18) to Host B, which is then stored in its ARP cache, linking

the router's IP address to its MAC address. An ARP spoofing attack is carried out by Host C (the attacker) by creating a fraudulent ARP reply, falsely claiming the IP address 192.168.0.1 (the router's address) but associating it with the attacker's MAC address (00:12:34). This spoofed ARP reply is sent to Host B, which subsequently updates its ARP cache, erroneously linking the router's IP address to the attacker's MAC address. Due to the attack, Host B's ARP cache is poisoned, resulting in future packets intended for the router being redirected to the attacker (Host C) instead. This enables the attacker to intercept, modify, and retransmit the data.

Figure 10 shows the ARP spoofing detection flowchart. A Bash script on Kali Linux is deployed on Host A, and this script serves as the IDS. This script periodically checks the MAC address associated with the router's IP address by querying the ARP table. If a discrepancy indicates a potential ARP spoofing attack, the script logs the event and triggers an alert. When an attack is detected, a Python script is integrated with the IDS to send email notifications via simple mail transfer protocol (SMTP). The IDS is automated as a background service, continuously monitoring the network to detect and respond to ARP spoofing attacks. Figure 11 illustrates the interactions between the hosts and the router, depicting the flow of ARP requests, replies, and the subsequent spoofing attack.

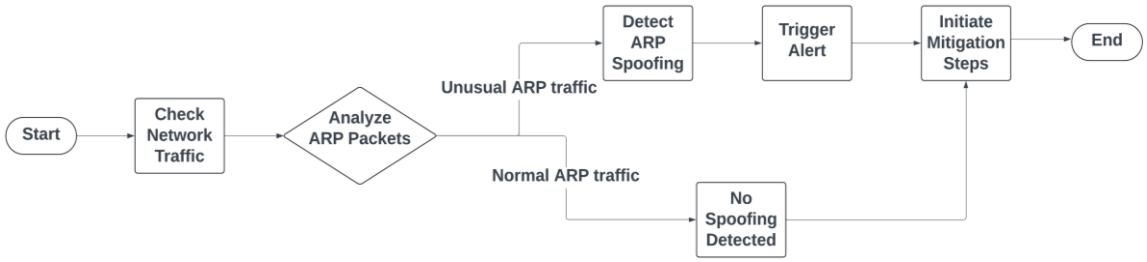



Figure 10. ARP spoofing detection process

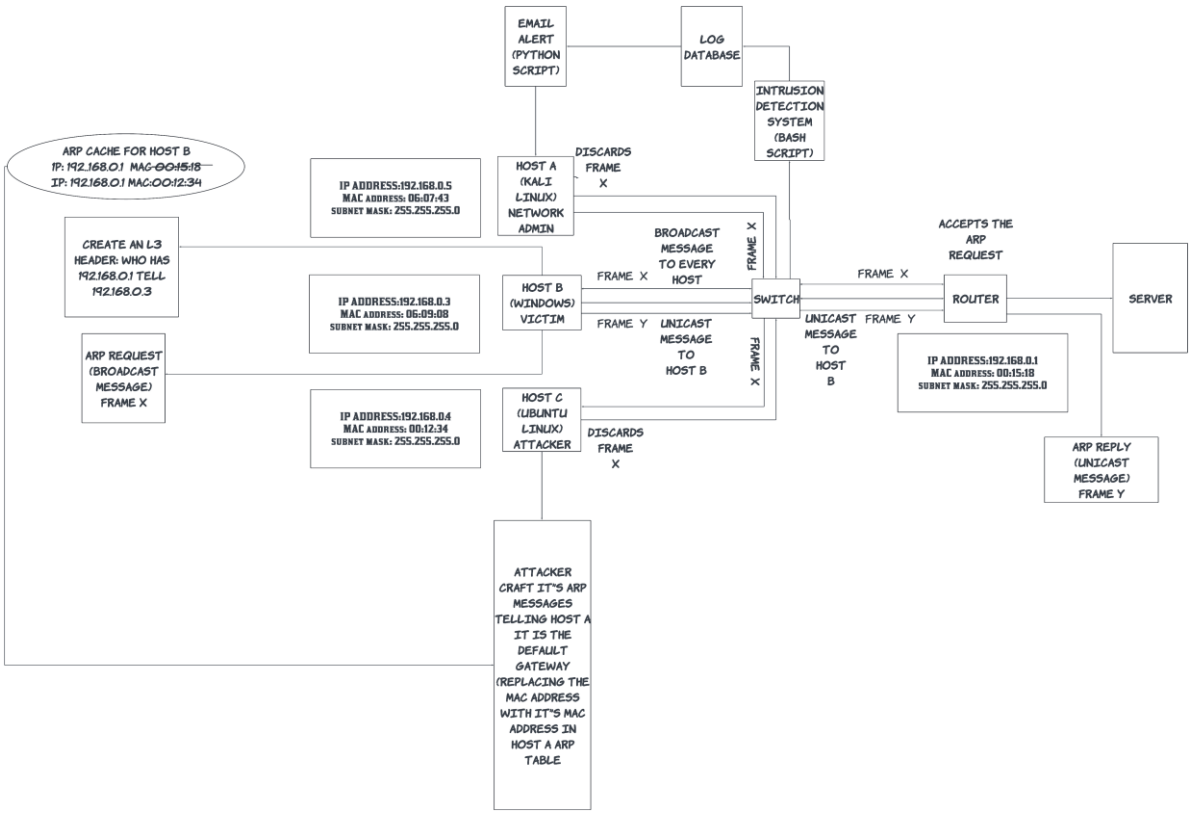



Figure 11. Conceptual framework of the system

The Bash script functions as a spoof detector by interacting with the operating system to execute commands and manage processes. It begins by identifying the router's IP address using the route -n command and obtaining the corresponding MAC address. The script then enters a loop, repeatedly checking the router's MAC address for a specified number of iterations (default: 500). If a change in the MAC address is detected during these checks, the script identifies it as a potential MITM attack. The results, including any detected changes, are displayed and saved to a file.

### 3.1. Bash script implementation

To implement the script, a file is created using the touch command and edited with nano. The script's interpreter is specified, allowing the Linux terminal to execute the commands. A usage function is included to guide the user on how to run the script with the correct arguments. The script ensures it is executed with root privileges and validates the provided network interface. A function is defined to retrieve the MAC address, and a welcome message introduces the spoof detector. The script then identifies the router's IP and stores the corresponding MAC address in a variable. The continuous loop monitors for any changes in the MAC address, with a 5-second delay between each check. If spoofing is detected, the script displays the altered MAC address and the detection date, saving the information to a results file. Finally, it warns the user if an attack is detected or confirms the absence of spoofing.

### 3.2. Python script implementation

The Python script serves as an SMTP server to send Results.txt to the network admin upon detecting an intrusion. The script imports necessary modules ($smtplib$, $email.mime.text$, $email.mime.multipart$), and the SMTP server's hostname is $mail.fxcapitalgrowth.site$. The script configures sender and receiver email addresses, reads Results.txt, and sends the content via email.

### 3.3. Integrating bash and python scripts

The Bash and Python scripts are integrated by including the command $(python3 mitm.py) within the Bash script. Both scripts must reside in the same directory to avoid file path errors. The Linux OS is preferred for its flexibility and suitability for network configurations and autonomous systems.

## 4. RESULTS

The study successfully implemented an ARP spoof detector using a bash script, which continuously monitored the ARP tables for any alterations in MAC addresses. The script logged the MAC addresses from the ARP tables and compared them in real-time with the current entries. Upon detecting a change, the bash script executed a Python script that sent an email alert to the network administrator. The email contained the IP address, the previous MAC address, and the altered MAC address of the compromised device, utilizing an SMTP server. The experimental setup required two Linux systems and one Windows system. The first Linux system acted as the network intrusion detection system, the second Linux system simulated the MITM attack, and the Windows system served as the victim. The scripts were made executable using the `$chmod + x$` command, which granted the necessary execution permissions. Upon execution, the Bash script first logged the router's IP address. It then monitored the associated MAC address using the `$get\_mac\_address()$` function, which retrieved the MAC address of the network gateway (eth0). A continuous while loop compared the router's MAC address against previous records, and any detected changes were saved locally in a `Results.txt` file. This triggered the execution of the Python script. Figures 12 and 13 illustrate the ARP table of the victim machine before and after the spoofing attack.

Initially, the ARP table mappings were correct, but after the attack, the table was poisoned, with the default gateway's MAC address being replaced by the attacker's MAC address. Figure 14 demonstrates the execution of the Bash script, where the router's IP and MAC addresses were retrieved. Figure 15 shows the successful detection of an ARP spoofing attack, resulting in a notification sent to the network administrator. The Python script read the `Results.txt` file and sent the detection report via an SMTP server.

The system was set up in an Oracle VM environment, with Kali Linux as the network intrusion detection system, Ubuntu as the attacker, and Windows as the host. Initial challenges arose due to the virtual machine's inability to detect the IP addresses of the host machine within the virtual environment. This was caused by network configuration constraints, which resulted in the virtual environment operating on a different virtual network card. To resolve this, the network card configuration was adjusted to a bridge network, allowing seamless communication between the host and the virtual machine. This modification was crucial for the effective functioning of the detection scripts.

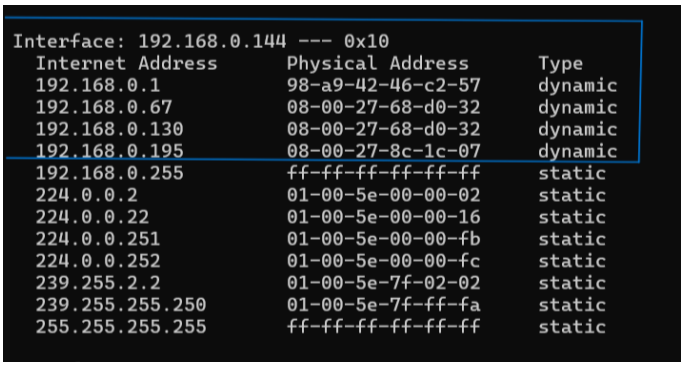



Figure 12. ARP table of victim machine before the attack

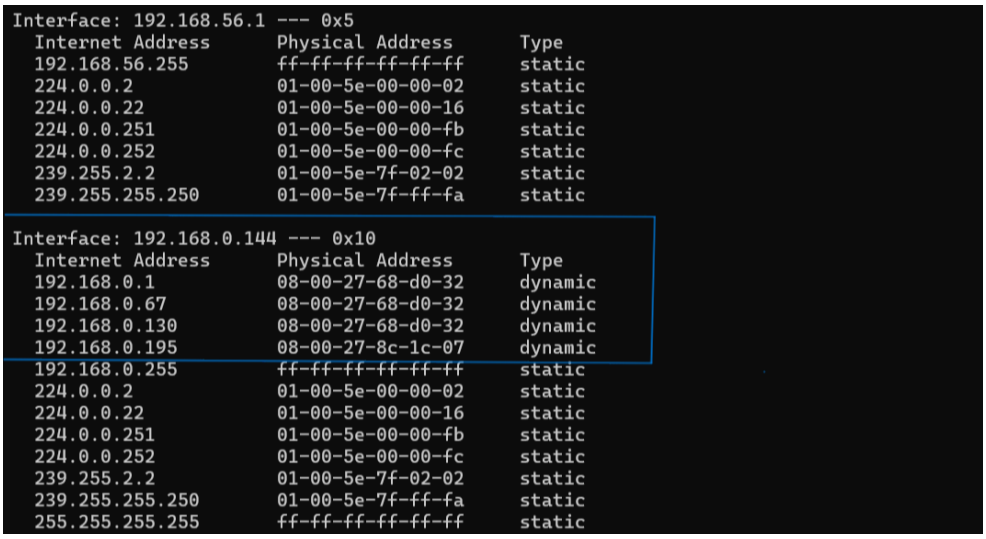



Figure 13. ARP table of victim machine after attack

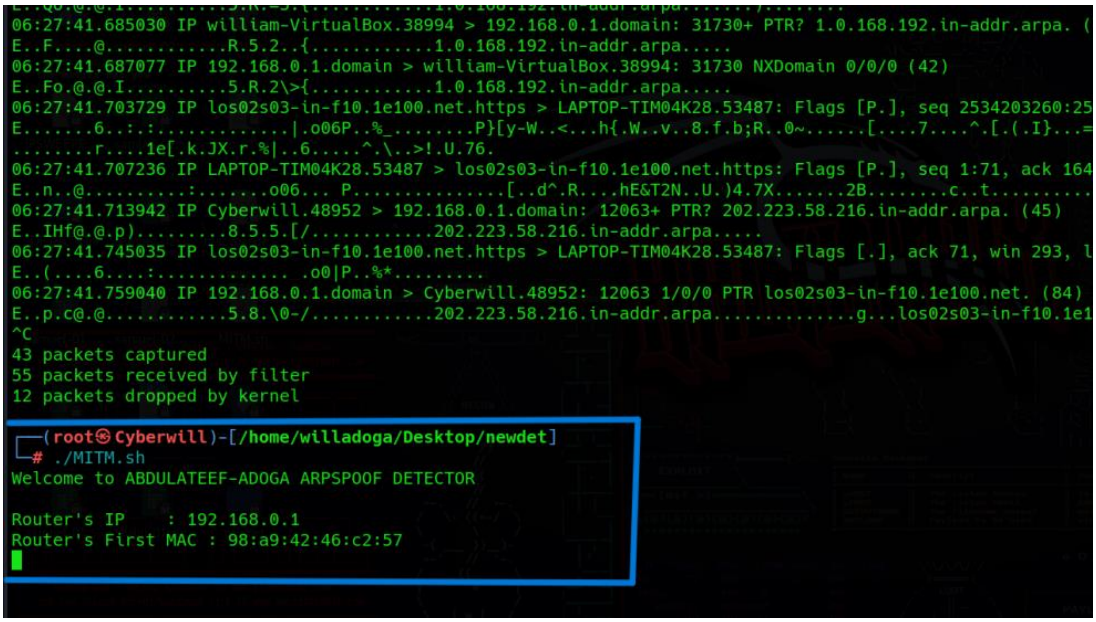



Figure 14. Bash script before detection

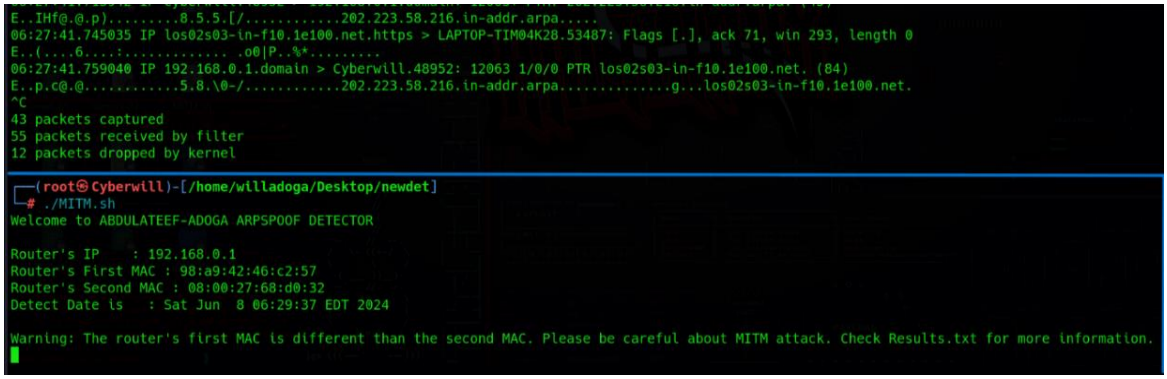



Figure 15. Bash script after ARP spoof detection

## 5. CONCLUSION

The system presented in this paper focuses on ARP spoofing attacks within a network, illustrating how attackers can exploit ARP to intercept communications. The sequence diagram effectively captures the interaction between network components, providing a clear visual representation of the attack process and the corresponding detection mechanism implemented by the IDS. This work underscores the importance of monitoring network traffic for anomalies. It highlights the role of IDS in safeguarding against man-in-the-middle attacks with a focus on ARP spoofing. The deployment of IDS as a background service ensures continuous protection, which is crucial in dynamic network environments where threats can arise at any moment. By integrating Bash scripting for detection and Python scripting for automated alerts, the approach outlined in this framework offers a robust solution for real-time detection and response to ARP spoofing attacks, contributing to the overall security of the network infrastructure.

## ACKNOWLEDGEMENTS

The authors acknowledge partial assistance from the Covenant University Centre for Research, Innovation, and Discovery (CUCRID), Ota, Ogun State, Nigeria.

## FUNDING INFORMATION

The authors acknowledge partial assistance from the Covenant University Centre for Research, Innovation, and Discovery (CUCRID), Ota, Ogun State, Nigeria.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Kennedy Okokpujie | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| William A. Abdulateef-Adoga | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | |
| Oghenetega C. Owivri | | | ✓ | ✓ | | | ✓ | | | ✓ | ✓ | | ✓ | ✓ |
| Adaora P. Ijeh | | | ✓ | ✓ | | | ✓ | | | ✓ | ✓ | | ✓ | ✓ |
| Imhade P. Okokpujie | | | | ✓ | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ |
| Morayo E. Awomoy | | | | ✓ | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, KO, Email: kennedy.okokpujie@covenantuniversity.edu.ng, upon reasonable request.

## REFERENCES


[1] Y. Kumar and V. Kumar, "A systematic review on intrusion detection system in wireless networks: variants, attacks, and applications," *Wireless Personal Communications*, vol. 133, no. 1. pp. 395–452, Nov. 2023, doi: 10.1007/s11277-023-10773-x.

[2] H. Debar, "An introduction to intrusion-detection systems," in *Proceedings of Connect*, 2000, pp. 1–18, Acesssed: May 24, 2024. [Online]. Available: https://www.researchgate.net/publication/228589845

[3] B. A. T. Mohit K. Raj and K. Jai, "Intrusion detection system," *International Journal of Technical Research and Applications*, vol. 5, no. 2. pp. 2320–8163, 2017.

[4] K. A. Scarfone and P. M. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST special publication, 2007.

[5] K. H. Günes Z.-H. A. Nur and M. I. Heywood, "Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets," *Proceedings of the Third Annual Conference on Privacy, Security and Trust*. pp. 1722–1723, 2006.

[6] J. Al-Mansor Mohammed and B. G. Kok, "Intrusion detection systems: principles and perspectives," *Journal of Multidisciplinary Engineering Science Studies*, vol. 4, no. 11, pp. 2266–2270, 2018.

[7] A. A. Anitha and L. Arockiam, "A review on intrusion detection systems to secure IoT networks," *International Journal of Computer Networks and Applications*, vol. 9, no. 1, pp. 38–50, 2022, doi: 10.22247/ijcna/2022/211599.

[8] P. Dedakia, "Network security: cyber-attacks & strategies to mitigate risks and threads," *Preprint*, pp. 1–4, 2020, doi: 10.13140/RG.2.2.30846.00320.

[9] C. Keturahlee, "An overview of intrusion detection and prevention systems," *arXiv preprint arXiv:2004.08967*, 2020.

[10] P. Ani, "Frequency of successful cyberattacks launched against organizations worldwide 2023," *Statista*. 2023, Accessed: May 25, 2024. [Online]. Available: https://www.statista.com/statistics/221394/successful-cyber-attacks-launched-against-businesses-worldwide/.

[11] C. Semeria, *Understanding IP addressing: everything you ever wanted to know*. NSD Marketing, 3Com Corporation, 1996.

[12] T. Wu, X. Xiao, L. Gong, Z. Liu, L. Wang, and C. He, "Research on IPv6 address state detection and management technology for new power internet of things," in *Proceedings of 2024 International Conference on Artificial Intelligence of Things and Computing, AITC 2024*, 2025, pp. 97–101, doi: 10.1145/3708282.3708300.

[13] M. Rouse, "Network identity," *Techopedia*, 2023. https://www.techopedia.com/definition/4029/network-identity-network-id-transmission-control-protocolinternet-protocol (accessed May 25, 2024).

[14] A. Thomas, "The packet delivery process: locally connected hosts," *Global Knowledge*. Accessed: May 25, 2024. [Online]. Available: http://d1wl9nui6miy8.cloudfront.net/media/965779/wp-the-packet-delivery-process-locally-connected-hosts.pdf.

[15] S. Gahlawat, "Packet flow within a network," *Linkedin*. Accessed: May 25, 2024. [Online]. Available: www.linkedin.com/pulse/packet-flow-within-network-sumit-gahlawat.

[16] Cloudflare, "What is the internet control message protocol (ICMP)." Accessed: May 25, 2024. [Online]. Available: https://www.cloudflare.com/en-gb/learning/ddos/glossary/internet-control-message-protocol-icmp/.

[17] A. Amoordon, C. Gransart, and V. Deniau, "Characterizing Wi-Fi man-in-the-middle attacks," in *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, Aug. 2020, pp. 1–4, doi: 10.23919/URSIGASS49373.2020.9232270.

[18] S. Gangan, "A review of man-in-the-middle attacks," *arXiv:1504.02115*, 2015.

[19] K. Okokpujie, C. G. Kennedy, K. Nnodu, and E. Noma-Osagha, "Cybersecurity awareness: Investigating students' susceptibility to phishing attacks for sustainable safe email usage in academic environment (a case study of a Nigerian leading university," *International Journal of Sustainable Development and Planning*, vol. 18, no. 1, pp. 255–263, 2023, doi: 10.18280/ijsdp.180127.

[20] O. J. Adeyemi, S. I. Popoola, A. A. Atayero, D. G. Afolayan, M. Ariyo, and E. Adetiba, "Exploration of daily Internet data traffic generated in a smart university campus," *Data in Brief*, vol. 20, pp. 30–52, 2018, doi: 10.1016/j.dib.2018.07.039.

[21] O. F. Isife, K. Okokpujie, I. P. Okokpujie, R. E. Subair, A. A. Vincent, and M. E. Awomoyi, "Development of a malicious network traffic intrusion detection system using deep learning," *International Journal of Safety and Security Engineering*, vol. 13, no. 4, pp. 587–595, 2023, doi: 10.18280/ijsse.130401.

[22] F. Gautama, E. Gunawan, and E. H. Yossy, "Development of a Linux based ARP poisoning detection system," in *ICBIR 2024 - 2024 9th International Conference on Business and Industrial Research, Proceedings*, 2024, pp. 333–338, doi: 10.1109/ICBIR61386.2024.10875789.

[23] D. Bruschi, A. Di Pasquale, A. Lanzi, and E. Pagani, "Ensuring cybersecurity for industrial networks: A solution for ARP-based MITM attacks," *Journal of Computer Security*, vol. 32, no. 5, pp. 447–475, 2024, doi: 10.3233/jcs-230023.

[24] D. R. Thomas, V. Prabhu, W. Nancy, G. Sowmiya, T. P. Adhithya, and V. Peroumal, "Detection and prevention of poisoning targets with ARP cache using scapy," in *Proceedings of the 2nd International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics, ICIITCEE 2024*, 2024, pp. 1–6, doi: 10.1109/IITCEE59897.2024.10467270.

[25] H. Mohapatra, "Handling of man-in-the-middle attack in WSN through intrusion detection system," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 1503–1510, 2020, doi: 10.30534/ijeter/2020/05852020.

## BIOGRAPHIES OF AUTHORS

**Kennedy Okokpujie** holds a Bachelor of Engineering (B.Eng.) in electrical and electronics engineering, a Master of Science (M.Sc.) in electrical and electronics engineering, a Master of Engineering (M.Eng.) in electronics and telecommunication engineering, a Master of Business Administration (MBA), and a Ph.D. in information and communication engineering, besides several professional certificates and skills. He is currently an associate professor in the Department of Electrical and Information Engineering at Covenant University, Ota, Ogun State, Nigeria. He is a member of the Nigerian Society of Engineers and a senior member of the Institute of Electrical and Electronics Engineers (SMIEEE). His research areas of interest include biometrics, artificial intelligence, and network security and management. He can be contacted at email: kennedy.okokpujie@covenantuniversity.edu.ng; kenjie451@gmail.com.

**William A. Abdulateef-Adoga** hold a bachelor's degree in information and communication engineering at Covenant University. His technical expertise includes proficiency with tools such as Burp Suite, Metasploit, Wireshark, Nmap and Nessus among others. He can be contacted at email: abdulateef-adoga.william@stu.cu.edu.ng.

**Oghenetega C. Owivri** holds a Bachelor of Engineering (B.Eng.) in electrical and electronics engineering and a Master of Engineering (M.Eng.) in computer engineering, among other professional certifications. He is an assistant lecturer in the Electrical and Information Engineering Department at Covenant University, Ota, Ogun State, Nigeria. His research areas of interest include blockchain, cryptography, software engineering, and cybersecurity. He can be contacted via email: oghenetega.owivri@covenantuniversity.edu.ng.

**Adaora P. Ijeh** is a postgraduate student currently pursuing a master's degree in computer science. She holds a bachelor's degree in computer science from the Federal University of Petroleum Resources, Effurun, Delta State, Nigeria. For her undergraduate research, she developed a web-based electronic management application aimed at improving administrative efficiency. Her current research interests include cybersecurity and artificial intelligence, with a focus on developing smart, data-driven systems. Miss Adaora is dedicated to advancing intelligent technologies through secure and innovative software solutions. She aspires to contribute to cutting-edge research in intelligent systems and secure software design, with the long-term goal of enhancing digital infrastructure in both developing and developed regions. She can be contacted at email: adaora.ijehpgs@stu.cu.edu.ng.

**Imhade P. Okokpujie** an associate professor at Afe Babalola University, holds a Ph.D. in mechanical engineering from Covenant University, focusing on nano-lubricants in advanced manufacturing. She has authored over 186 scholarly papers and a book on modern optimization techniques. Dr. Okokpujie has secured three research grants and received multiple awards, including Covenant University's Chancellor's Exceptional Researcher of the Year (2018, 2019) and Afe Babalola University's outstanding research award. She specializes in advanced manufacturing, nano-lubricants, and energy systems, and has been awarded the Global Excellence Stature Fellowship Grant at the University of Johannesburg. A registered COREN engineer and NSE member, she has held leadership roles in her institution, APWEN, and NSE, mentoring numerous students and young researchers. Dr. Okokpujie is passionate about girl-child education and women's development. She can be contacted at email: ip.okokpujie@abuad.edu.ng.

**Morayo E. Awomoy** is a dedicated advocate for social transformation, holding a master's degree in international peace and conflict resolution from American University in Washington, D.C. with a strong academic foundation and a passion for driving meaningful change, Morayo Awomoyi focuses on the intersection of peacebuilding, social justice, and community empowerment. Throughout her career, she has explored innovative approaches to addressing systemic inequalities, fostering inclusive dialogue, and promoting sustainable peace in diverse contexts. She brings a global perspective to local challenges, drawing on both theory and practical engagement in the field of conflict resolution through statistical analyses. Guided by the belief that lasting peace is rooted in equity and social change, she collaborates with communities, organizations, and policy-makers to cultivate environments where justice and human dignity thrive. She can be contacted at email: ma8161b@american.edu.